Automatic Supervised Thesauri Construction with *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 Doctor of Philosophy in Computer Science

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

Abstract

Thesauri and similarly structured lexical resources are important tools for a variety of Natural Language Processing (NLP) applications. In recent years one resource in particular has become very widely used: *WordNet*. However, *WordNet* represents just one of many ways of organizing the English lexicon and is not necessarily the best suited tool for any particular task. Another thesaurus, less often used in NLP, is *Roget's Thesaurus*. Although it is of high quality and has been in development for a century and a half, its use has been limited. That is in no small part due to the fact that the only publicly available edition dates from 1911. In this thesis I propose and test methods of automatically updating the vocabulary of the 1911 *Roget's Thesaurus*. My hope is that introducing a more full and modern vocabulary will make *Roget's* more useful for many NLP tasks. Consequently, the goal for my thesis is twofold: (1) to automatically update *Roget's Thesaurus* and (2) to show how *Roget's* in its updated form compares to *WordNet* on a variety of tasks.

Throughout my thesis I attempt to use the existing *Roget's Thesaurus* as a source of training data in order to learn from *Roget's* for the purpose of enhancing *Roget's*. The updating of *Roget's Thesaurus* is done in two stages. In the first stage I develop a measure of semantic relatedness (MSR) that enhances existing distributional techniques. I add novelty to this process by using known sets of synonyms from *Roget's* to train a distributional measure to better identify near synonyms. In the second stage I use the new measure of semantic relatedness to find where in *Roget's* to place a new word. In this case I use existing words from *Roget's* as training data to tune the parameters of three methods for identifying where in *Roget's* to place a new word. Over 5 thousand new words and word-senses were added using this process.

Once I have updated *Roget's*, two kinds of evaluation are conducted. One evaluation is on my procedure for updating *Roget's Thesaurus*. This is accomplished by removing some words from the *Thesaurus* and testing my system's ability to reinsert these words in the correct location. Human evaluation of the newly added words is also performed. In it, the annotators must determine whether a newly added word is in the correct location. Their findings were that in most cases the new words were almost indistinguishable from those words already existing in Roget's Thesaurus.

The second kind of evaluation is to establish the usefulness of the enhanced *Roget's Thesaurus* by applying it to several known NLP problems. These problems include determining semantic relatedness between word pairs or sentence pairs, identifying the best synonym from a set of candidates and solving SAT-style analogy problems. One of two larger applications on which the various versions of *Roget's* are compared is a pseudo-word-sense disambiguation task, which could be extended to do real word-sense disambiguation or lexical replacement. The second application is the ranking of sentences from a document set for the purposes of building an extractive text summarization system. The updated *Thesaurus* consistently performed at least as well as or better than the original *Thesaurus* in all these applications.

Although the work in this thesis focusses on automatically adding new words to the *Thesaurus*, it is intended to be only the first step in updating *Roget's*. As future work, these additions ought to be examined by lexicographers, who will either confirm the placement of the newly added words, or move them to their correct location.

The main contributions of this work are the new supervised measure of semantic relatedness and generally the methodology for updating the vocabulary in *Roget's Thesaurus*. In addition to this, many of the problems used to evaluate the *Thesaurus* are solved using new methods or on new data sets, including pseudo-word-sense-disambiguation and sentence ranking for extractive text summarization. The end result of this thesis is a free, publicly available version of *Roget's Thesaurus* with an updated lexicon.

Acknowledgements

I would like to thank Dr. Stan Szpakowicz for supervising my thesis; Mario Jarmasz for providing the Java interface for, and insight into, *Roget's Thesaurus* and also for many of the data sets; Bartosz Broda for his help running *SuperMatrix* and measuring semantic relatedness; Terry Copeck for constructing the SCU-labelled corpus; Dr. Saif Mohamed for providing me with the NRC Emotion and Sentiment Corpus; and Dr. Vivi Nastase for devising the formula for solving analogy problems. I would also like to thank my manual annotators Oana Frunza, Chris Fournier, Anna Kazantseva and Richard Keelan for taking the time and effort to help me evaluate my work. Also thank-you to Mahesa Jeyakanthan and Martin Scaiano for helping design the manual annotation experiments. To my committee, Dr. Stan Matwin, Dr. Diana Inkpen, Dr. Jean-Pierre Corriveau and Dr. Vlado Keselj: thank you for the insightful comments and discussions on my thesis. I would also like to thank the Natural Sciences and Engineering Research Council of Canada and the University of Ottawa for supporting my research. Finally I want to thank my parents for all their support an inspiration during these many years of grad school.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Thesauri and other similarly organized lexical knowledge bases are useful resources for the Natural Language Processing (NLP) community and have played a role in many applications. *Roget's Thesaurus*, in existence for well over a century and a half, has been shown to be useful for some NLP applications, yet has not been used as widely as other similar resources. *WordNet* (Fellbaum, 1998) has become the default thesaurus that the NLP community turns to. This has largely been brought about by the fact that other similar resources like *Roget's* have not been publicly available in a suitable software package. It is important for NLP researchers to remember that *WordNet* represents just one of many methods of organizing the English lexicon and is not necessarily the best system available for every NLP task. By updating *Roget's Thesaurus* I hope to develop a competitive and up-to-date resource that will measure up to *WordNet* in terms of quality on a variety of NLP applications. In this thesis I describe and evaluate a number of variations on a novel method of updating the lexicon of *Roget's Thesaurus*.

There are many methods for learning to construct or enhance a thesaurus by clustering related words with the earliest work starting decades ago (Tsurumaru et al., 1986; Crouch, 1988; Crouch and Yang, 1992). In terms of updating an existing thesaurus, relatively few methods actually use the thesaurus in learning how to update itself. There are two primary ways in which I go about trying to "learn" a new resource from an old one. One is in learning a new Measure of Semantic Relatedness (MSR) between terms, the other is in tuning a system to place a word in the *Thesaurus*.

Measuring semantic relatedness between terms is often done using a *term-context* matrix constructed by counting the number of times a term appears in a given context. The vector distance between two terms can be used to determine their semantic relatedness.

Usually the weights in the *term-context* matrix are re-weighted based on the association between a word and the context that it appears in. This is essentially an unsupervised process as it does not benefit from any known synonym/non-synonym pairs. However, there is a plethora of such pairs available in resources such as *Roget's Thesaurus* and *WordNet*. I will demonstrate how supervised context weighting can be combined with unsupervised to create a more powerful and robust measure of semantic relatedness (MSR). By using sets of known synonyms from a particular resource, I am trying to create a "customized" MSR. Two thesauri may have somewhat different standards for what they call synonyms: some may be very closely related, while others may be related more loosely. In theory, using synonyms from these resources should cause the MSR to also have a tighter or looser concept of how much two words are related.

Adding new term to *Roget's Thesaurus* can be done in a variety of ways. A word could simply be placed in a resource next to its closest neighbouring term, or perhaps a certain number of terms should be used to identify where to place a new word. There are a number of parameters that need to be tuned when adding words to *Roget's* and, while this may not constitute supervision, I will take advantage of the structure of *Roget's* in order to best discover where to place new words in *Roget's*.

I will evaluate my methods of updating the lexicon on two versions of the *Thesaurus*, one from 1911 and one from 1987. The printed version of *Roget's Thesaurus* is periodically updated for new releases, but these releases are not easily available to NLP researchers and so have had little impact on the NLP community. Currently the 1911 version of *Roget's Thesaurus* is freely available through Project Gutenberg.[1] The other version that I will work with is the 1987 version from *Penguin's Roget's Thesaurus* (Kirkpatrick, 1987). An open Java API for the version of the 1911 *Roget's Thesaurus* and its updated versions, as well as the applications that are described in this thesis are available on the web under the name of *The Open Roget's Project*.[2] The API is built on the work of Jarmasz (2003).

A final evaluation of these updated thesauri will be conducted on several NLP applications. *Roget's* itself can be used as a MSR. Semantic relatedness between terms has been used as an evaluation method quite extensively in the past, as has identifying synonyms (Jarmasz and Szpakowicz, 2004). Measuring semantic relatedness between terms is not terribly useful on its own, but it has been a component in many other systems solving more interesting problems. Semantic relatedness can also be tested on a

[1] http://www.gutenberg.org/ebooks/22

[2] http://rogets.eecs.uottawa.ca/

pseudo-word-sense disambiguation task as in Weeds and Weir (2005). A pseudo-word is made by joining two different words, with similar distributions, together and the task of pseudo-word-sense disambiguation is to determine which of these two words actually belongs in a given context. This will demonstrate *Roget's* ability to determine contexts in which a given word sense appears. This could be an important component in a much larger word sense disambiguation system. If a word has multiple senses, but one of those senses has closely related words that regularly appear in the same context then one can guess that it is the correct sense of the word. Some research has been done on relatedness between sentences in Li et al. (2006) which can be used to evaluate *Roget's Thesaurus* at a sentence level. Sentence relatedness can also be used for sentence ranking in text summarization, which I will explore. These applications all are discussed in Chapter 6.

The process of updating *Roget's* is outlined in Figure 1.1. I will work with Wikipedia as a corpus and use the parser *Minipar* (Lin, 1998a) Essentially I start with raw text. It is parsed, and a word-context matrix is constructed. This matrix is then re-weighted in either a supervised or unsupervised manner. Using the term-context matrix for each word, its nearest synonyms are generated and a location in *Roget's Thesaurus* to place these words is deduced using the *Thesaurus* as a source of tuning data. This last step can be repeated multiple times to update the lexicon of *Roget's*.

1.1 History of *Roget's* Thesaurus

Peter Mark Roget, a physician, first started work on what would become *Roget's Thesaurus* in the early 1800s to categorize terms and phrases for his personal use in writing. The earliest manuscripts of the *Thesaurus* date back to 1805. In 1852 it was published for the first time and has gone through many revisions continuing to this day. Initially the upkeep of *Roget's Thesaurus* was put in the hands of Peter Mark Roget's son and then grandson. In 1952 the rights to the *Thesaurus* were sold off to Longmans who put out an edition in 1962. After that Penguin took over producing copies of the book (Kendall, 2008). Since then many successive versions of *Roget's* Thesaurus have been printed, including: Chapman (1977); Kirkpatrick (1987); Chapman (1992); Kirkpatrick (1998). Although the structure of *Roget's Thesaurus* has changed little throughout the years there has been quite a lot of change in content. The number of main concepts in *Roget's* has actually decreased over the years though the lexicon has increased by hundreds of thousands of words. By 2002 Penguin's *Roget's Thesaurus* contained nearly 25 times as many words as Dr. Roget's original manuscript (Kendall, 2008).

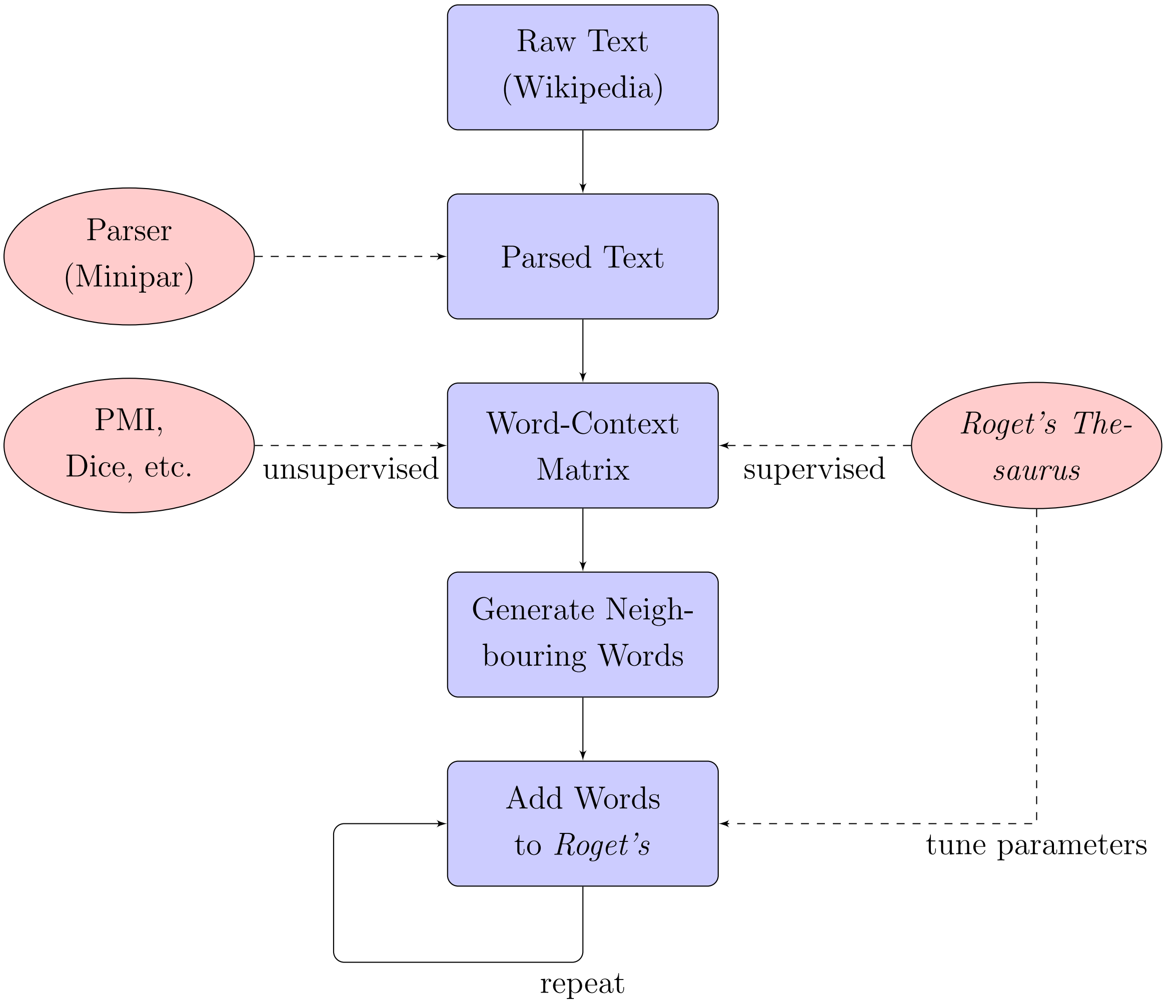

Figure 1.1: The process of adding new words to *Roget's Thesaurus*.

Some versions have been made available in other languages, including German and Spanish. The German versions were produced in the late nineteenth century while the Spanish version was produced during the twentieth century and has continued to be updated (Kendall, 2008).

Probably the most easily available version of *Roget's* is the 1911 edition, prepared by Micra Inc. and made public on Project Gutenberg. It appears to have been used in various Web versions of *Roget's*. The title *Roget's* is not trademarked, so anyone can use it when publishing their own version of the *Thesaurus*. That is why *Roget's* has become almost synonymous with the word *Thesaurus*.

In recent years this writing aid has been adapted with some success for use in Natural Language Processing. The 1987 version (Kirkpatrick, 1987) has been used for such problems as measuring semantic relatedness (Jarmasz and Szpakowicz, 2004) and building Lexical Chains (Jarmasz and Szpakowicz, 2003). An altered version of the 1911 *Roget's Thesaurus* called *FACTOTUM* was created and is publicly available (Cassidy, 2000). This was actually an attempt to make the framework for an ontology out of *Roget's*, but *FACTOTUM* does not appear to be widely used. It is described in more detail in Section 2.4.1. *Roget's* has also been used for word-sense disambiguation (Yarowsky, 1992), Text Summarization (Copeck et al., 2008, 2009; Kennedy and Szpakowicz, 2010a; Kennedy et al., 2010, 2011, 2012) and query expansion (Mandala et al., 1999). Some work has been done either translating *Roget's Thesaurus* into French (de Melo and Weikum, 2008) or aligning it with a French Thesaurus (Prince and Chauch, 2008).

1.2 Updating the Vocabulary of a Thesaurus and Similar Lexical Resources

The motivation for updating the 1911 *Roget's Thesaurus* is simple: the *Thesaurus* is old and outdated. Although some updating was done in preparing it for use in *FACTOTUM* (about 1000 words/phrases were added) this makes up about 1% of the *Thesaurus* contents that was added after 1911. Many new words and new senses of existing words do not appear in it, and this negatively effects its usefulness to NLP and indeed any user of the *Thesaurus* today. A system that automatically, or semi-automatically, updates the vocabulary of a thesaurus will also save countless man-days or even months of work in manually updating it. This line of research is not unique to *Roget's*. For example in Snow et al. (2006) *WordNet* is expanded with new words attached with hyponym links to existing synsets in the *WordNet* structure. Also Broda et al. (2008) have used similar techniques for building a Polish *WordNet*.

Adding new words automatically to any thesaurus is a difficult task. It will undoubtedly introduce many poorly placed words into *Roget's*. From a purely linguistic point of view this could be quite undesirable, but the purpose of this thesis is to make a resource that will benefit the NLP community not to replace human thesaurus builders. That is why always making perfect additions to *Roget's*, as a human annotator may do, should not be necessary to declare this a success. Since one of the biggest use of *Roget's* in NLP is as a database for semantic relatedness, this is one of the criteria on which I will determine the success of this project. If NLP problems that make use of a *Roget's* MSR improve with the updated lexicon then these updates will be a success. This is particularly true when evaluating the *Roget's* MSR on the newly added words. These experiments are reported in Chapter 6.

1.3 In This Thesis

1.3.1 Contributions From This Thesis

There are a number of contributions from this thesis:

- A supervised method of context weighting for measuring semantic relatedness and the software to run it.

- Experiments demonstrating the supervised MSR's effectiveness in ranking words in *Roget's Thesaurus* and also identifying emotional and sentimental words.

- Evaluation of several procedures for identifying where a word can be added to *Roget's Thesaurus*.

- A detailed comparison of the 1987 and 1911 versions of *Roget's* and comparisons with *WordNet* 3.0, exploring both their accuracies on various NLP tasks and the implications of their design for NLP algorithms.

- Several applications I present are novel, particularly the methods of applying *Roget's Thesaurus* for pseudo-word-sense disambiguation and sentence ranking for text summarization.

- Evaluation on previously established applications including word and sentence similarity, selecting the best synonym and identifying analogies.

- The updated *Roget's Thesaurus* itself, available as an open source Java API.

1.3.2 Chapters in This Thesis and Work Published So Far

This thesis is divided into 7 chapters including this introduction. Chapter 2 describes the structure of *Roget's Thesaurus* and compares the 1987 and 1911 versions against each other and with *WordNet*. Chapter 3 summarizes much of the literature on applications that make use of *Roget's Thesaurus* as well as work on building MSRs from large corpora. Chapter 4 describes my experiments building a supervised MSR. Chapter 5 outlines my experiments adding new words to *Roget's Thesaurus*. Chapter 6 reports on the evaluation of the original and updated thesauri on several applications pertaining to semantic relatedness, pseudo-word-sense disambiguation and text summarization. Chapter 7 concludes this thesis. Chapters 4, 5 & 6 all contain their own avenues for future work.

Portions of Chapters 2 and 6 have been published at ACL 2008 as (Kennedy and Szpakowicz, 2008). The description of the corpus for text summarization in Section 6.6.1 was published in Kennedy and Szpakowicz (2010b), while the system itself is published in Kennedy and Szpakowicz (2010a) also used in Copeck et al. (2009), Kennedy et al. (2010), Kennedy et al. (2011) and Kennedy et al. (2012). The results of the experiments comparing run times between *Roget's* and *WordNet* have been published in (Kennedy and Szpakowicz, 2012a). An early version of the supervised measure of semantic relatedness described in Chapter 4 can be found in Kennedy and Szpakowicz (2011) while the experiment from this thesis are published in Kennedy and Szpakowicz (2012b). This work was was also presented at the Canadian AI 2010 Graduate Symposium: parts of Chapter 1 are published in Kennedy (2010).

The following papers present the most significant results published from my thesis:

- Kennedy, A. and Szpakowicz, S. (2008). Evaluating Roget's Thesauri. In *Proceedings of ACL-08: HLT*, pages 416–424, Columbus, Ohio, USA. Association for Computational Linguistics.

- Kennedy, A. and Szpakowicz, S. (2010). Evaluation of a Sentence Ranker for Text Summarization based on Roget's Thesaurus. In *Proceedings of Text, Speech and Dialogue*, TSD 2010, pages 101–108, Brno, Czech Republic. Springer.

- Kennedy, A. and Szpakowicz, S. (2011). A Supervised Method of Feature Weighting for Measuring Semantic Relatedness. In *Proceedings of Canadian AI 2011*, pages 222–233, St. John's Newfoundland, Canada. Springer.

- Kennedy, A. and Szpakowicz, S. (2012). Supervised Distributional Semantic Relatedness. In *Proceedings of Text, Speech and Dialogue*, TSD 2012, Brno, Czech Republic. Springer.

More minor contributions can be found in:

- Kennedy, A. (2010). Automatically Expanding the Lexicon of Roget's Thesaurus. In *Proceedings of the Graduate Symposium at Canadian AI 2010*, pages 410-411, Ottawa, Ontario, Canada. Springer.

- Kennedy, A. and Szpakowicz S. (2012). Fast Semantic Relatedness: WordNet::Similarity vs Roget's Thesaurus. In Tiny Transactions on Computer Science, Volume 1.

Chapter 2

Thesauri and Lexical Ontologies

What does *Roget's Thesaurus* provide us that other NLP resources like *WordNet* do not? In this chapter I describe in detail the structure of *Roget's Thesaurus* and highlight the pros and cons of *Roget's* design. This also explains why I want to provide an up-to-date and NLP-friendly version of *Roget's*.

A thesaurus, like a dictionary, attempts to define the lexicon of a particular language. However, a thesaurus organizes words based on semantics, while a dictionary is organized alphabetically. A dictionary attempts to separate the different senses of a word through definitions. The senses and meaning of a word in a thesaurus are implied by the neighbouring terms (Kilgarriff and Yallop, 2000).

WordNet is often referred to as a lexical ontology (Saias and Quaresma, 2002; Alfonseca and Manandhar, 2002; Mann, 2002; Veale, 2003; Alfonseca, 2004; Simina and Barbu, 2004; Baek et al., 2008; Zheng et al., 2009; Aversano et al., 2010; Ofoghi and Yearwood, 2010). Resources like *Roget's* and *WordNet* probably should not be considered ontologies, though there are similarities between these resources and what is more traditionally accepted as an ontology. These similarities and differences are discussed in this chapter.

2.1 Description of *WordNet*

In this section I will briefly describe the structure of *WordNet*. The reason for describing *WordNet* is largely to provide a contrast to the design of *Roget's*. These resources both aim to organize the English lexicon, but they do it in quite different ways.

In *WordNet* words/phrases are grouped together in synsets. These synsets contain groups of words that are synonymous with each other. A definition for the words in the synset is also provided, sometimes with sample uses of the word. For example, there are two noun definitions of the word “thesis”:

1. (3) thesis – (an unproved statement put forward as a premise in an argument)
2. dissertation, thesis – (a treatise advancing a new point of view resulting from research; usually a requirement for an advanced academic degree)

In addition to this some frequency information is provided. The number of times that a given word appears in the SemCor (Fellbaum, 1998) corpus is included.

Each synset can be related to other synsets through a variety of semantic relationships. The most important relationship is HYPERNYMY/HYPONYMY which represents an IS-A relationship between synsets. This is used to create a hierarchy for both the verb and noun synsets. Nouns and verbs also have DERIVED FORMS available in *WordNet*. Other noun relationships include MERONYMY/HOLONYMY, SYNONYMY and ANTONYMY (SYNONYMS and ANTONYMS being available for all parts-of-speech). VERB FRAMES are available for verbs as are the relationships CAUSE TO and ENTAILMENT. Adjectives can have an ATTRIBUTE relationship with nouns. For adjectives and adverbs PERTAINYM relationships can also be retrieved from *WordNet*.

The design of *WordNet* is based around a theory of how the human mind stores concepts (Fellbaum, 1998). It has been under constant construction, *WordNet* 3.0 is the most recently released version available for download.[1] A comparison of *Roget’s* and *WordNet* can be found in Section 2.3.

2.2 Description of *Roget’s Thesaurus*

Several versions of *Roget’s Thesaurus* have been used in NLP research. Some of the more prominent ones in the literature include a 1911 version (Kennedy and Szpakowicz, 2008; Cassidy, 2000; Baumgartner and Waugh, 2002), a 1963 version (Old, 2002, 2004), and a 1987 version (Jarmasz and Szpakowicz, 2001a,b, 2003, 2004; Kennedy and Szpakowicz, 2007). There are also numerous Web sites with searchable versions of *Roget’s Thesaurus*,[2] many of which appear to be built on the 1911 versions. Much of that work (Baumgartner and Waugh, 2002; Old, 2002, 2004) focuses on describing the content and visualizing aspects of *Roget’s Thesaurus* and does not directly tackle any NLP applications.

[1] http://wordnet.princeton.edu/

[2] http://www.roget.org/

2.2.1 *Roget's* Hierarchy

This section describes the structure of *Roget's Thesaurus* and in doing so highlights the differences between the 1911 and 1987 versions.

A nine-level hierarchy makes up most of the structure of the *Thesaurus*. From top to bottom the hierarchy consists of:

- Class
- Section
- Sub-Section
- Head Group
- Head
- Part of Speech (POS)
- Paragraph
- Semicolon Group (SG)
- Word and Phrase

Classes, Sections Sub-Sections and Heads have names assigned to them. Head Groups are labelled with the head numbers or names of the heads it contains. Part of speech is represented by one of 8 different parts-of-speech found in the Thesauri. Paragraphs and Semicolon Groups are represented by the first word found in their grouping. It is worth noting that division of parts-of-speech happens quite low in the hierarchy, not at the very top as is the case in *WordNet*. I will define a *Roget's grouping* to be the set of words contained within an instance of one of these levels. For example a given Paragraph could be a *Roget's grouping* and so could a given Class. Usually the kinds of *Roget's groupings* I will talk about are either POSs, Paragraphs or SGs.

Classes

The *Thesaurus* has just 8 (though sometimes this is reduced to 6) classes containing the very highest-level divisions of information. The classes in the 1911 *Roget's Thesaurus* are:

- Abstract Relations

- Space
- Matter
- Intellect: formation of ideas
- Intellect: communication of ideas
- Volition: individual volition
- Volition: social volition
- Emotion, religion and morality

The 1987 *Thesaurus* has essentially the same classes although some are given different names. Some versions will merge the two *Intellect* and two *Volition* Classes into one each. Placing emotion, religion and morality so high in the *Thesaurus* are interesting choices, which could be considered to come from the 19^{th} century attitudes of the author.

Sections and Sub-Sections

These two categories represent further breaking down of concepts represented by the Class. In the following example I show the section "Existence" and its Sub-Sections.

- Existence
  - Abstract
  - Concrete
  - Formal
  - Modal

There appears to be an IS-A relationship between the Sub-Sections and the Section in this example since "abstract existence" → "existence". Sections and Sub-Sections do not all build an IS-A hierarchy. The following example demonstrates a much more complex relation: instead of KINDS OF "Causation" one can see a list of topics related to Causation in the Sub-Sections.

- Causations
  - Constancy of Sequence in Events

  - Connection Between Cause and Effect
  - Power in Operation
  - Indirect Power
  - Combinations of Cause

Head Groups and Heads

Heads in any *Roget's*-style thesaurus represent around 1000 concepts. The 1911 *Roget's* has 1044 while the 1987 *Roget's* contains 990. Examples of Heads include "existence", "evolution", "sculpture" and "presence".

Head Groups tend to contain opposite or complementary concepts. A Head Group is represented as a short list of Heads. Examples of opposites in Head Groups include:

- Presence – Absence
- Representation – Misrepresentation
- Marriage – Celibacy

An example of a Head Group with more than two heads appears in the Sub-Section for "Consecutive Order"; it contains "Beginning", "Middle" and "End". Occasionally a Head Group will contain just one Head; the Sub-Section "Absolute Relation" contains a Group with only the Head "Correlation".

An example of a head appears in Figure 2.1. It comes from *Roget's Thesaurus* 1911: Head #586 which groups terms pertaining to language.

Part of Speech (POS)

The part-of-speech level of *Roget's* hierarchy may be a little confusing: clearly no such set contains an exhaustive list of all nouns, verbs, etc. in English. I will write "POS" to indicate a structure in *Roget's* and "part-of-speech" to indicate the word category in general. Nouns, verbs, adjectives and adverbs are the four main parts-of-speech represented in a POS. Also included are interjections, which are usually phrases followed by an exclamation mark, such as "for God's sake!" and "pshaw!".

Class 5: Intellect: communication of ideas
Section 3: Means of communicating ideas
Sub-Section: Conventional means
Head Group: 586 Language
Head: 586 Language

N. *language*; *595* phraseology; *608* speech; tongue, lingo, vernacular; mother tongue, vulgar tongue, native tongue; household words; King's English, Queen's English; *589* dialect.

confusion of tongues, Babel, pasigraphie; *sign 576* pantomime; onomatopoeia; betacism, mimmation, myatism, nunnation; pasigraphy.

lexicology, philology, glossology, glottology; linguistics, chrestomathy; paleology, paleography; comparative grammar.

literature, letters, polite literature, belles lettres, muses, humanities, literae humaniores, republic of letters, dead languages, classics; genius of language; *scholar 516* scholarship.

VB. *592 express by words.*

ADJ. *lingual*, linguistic; dialectic; vernacular, current; bilingual; diglot, hexaglot, polyglot; literary.

PHR. *"syllables govern the world"*.

Figure 2.1: Sample of Head 586: Language from *Roget's Thesaurus* 1911

The 1911 version also contains phrases, prefixes and pronouns, absent from other versions of *Roget's*[3]. Phrases come from a variety of sources. There are well-known foreign-language phrases, for example:

- Le style, c'est l'homme
- Carpe diem

Other quotes are from famous people:

- For every action there is a reaction, equal in force and opposite in direction – Newton
- Ignorance never settles a question – Disraeli

Still other quotes come from fictional characters:

- Thou can'st not say I did it – Macbeth
- Go ahead, make my day! – Dirty Harry

Obviously the quote from Dirty Harry was added at a later date. Through a personal conversation with Patrick Cassidy I found it was added during the creation of *FACTOTUM*. Another example is "It's a long long way to Tipperary" which comes from a song by Jack Judge believed to have been written in 1912[4]. Other examples include "DNA virus" and "RNA virus" added to the Head for "Disease".

Paragraphs and Semicolon Groups

The Paragraph is the second smallest grouping in *Roget's*, while the Semicolon Group (SG) is the smallest. SGs are so named because they are separated with semicolons in *Roget's Thesaurus*. The concepts found in the Paragraph and the SG can be identified by looking at the first word in either grouping. The first SG in a Paragraph contains words most central to the concept expressed by the Paragraph. For example, in Figure 2.1 the first SG contains near-synonyms of the word *language*. The second SG {*phraseology*} relates to the manner in which words and phrases are used, while the third SG {*speech*}

[3] In the Project Gutenberg data of the 1911 *Roget's* there also are three prefixes ("tri-", "tris-" and "laevo-") and six pronouns ("he", "him", "his", "she", "her" and "hers").

[4] http://www.stalybridge.org.uk/jack_judge.htm

refers to the oral use of language. The SG {*mother tongue, vulgar tongue, native tongue*} contains phrases relating to the first language a person may learn. In the next SG, {*household words*} are part (MERONYMS) of a *language*. In the second last SG one can find {*King's English, Queen's English*} which names different kinds of English.

Another Paragraph starts with the SG {*confusion of tongues, Babel, pasigraphie*} – words and phrases related to not understanding language. In the third Paragraph in the noun POS, the first SG contains *lexicology*, the study of the lexical component of language; others contain words related to the study of language change *glossology*, or to the study of language in general *linguistics*. Of the related SGs, only {*King's English, Queen's English*} are kinds of languages, so they could be HYPONYMS of *language* (though it would make more sense for them to be HYPONYMS of "English Language").

Since there are fewer verbs and adjectives in Figure 2.1 it is harder to demonstrate the variety of relationships that can be found in these groupings. Nevertheless some relationships to the central concept of *Language* can be found. *Lingual* means related to language while *Dialectic* is related to debating opposing positions on some issue. *Bilingual* describes something that uses two languages. Although there are no adverbs and not enough verbs to show any relationships in Figure 2.1, these parts-of-speech also contain many varied relationships.

Also of interest is the relationship between Paragraphs and POS. Clearly "language" is the main concept of this Head, with the words/phrases in the first Paragraph being most central to the concept of language. The second paragraph covers misunderstanding of language, the third – the study of language, the fourth – things expressed in language.

For verbs the only concept is the act of using language (presumably in communication), for adjectives – linguistic qualities. The only phrase is a quote from George Bernard Shaw.

There are numerous relations, which can differ in a subtle manner. Identifying and labelling all the relations in *Roget's Thesaurus* is likely an extremely difficult process. These examples help demonstrate the variety of semantic relations available; not all of which are covered in *WordNet*. It is also worth noting that often the relation between words/phrases in the same SG is not synonymy, but rather a sort of close relatedness. Words/phrases in a SG are linked with the central theme of the Paragraph by a common relationship, be it HYPERNYMY or MERONYMY or anything else.

Hierarchy	**1911**	**1987**
Class	8	8
Section	39	39
Subsection	97	95
Head Group	625	596
Head	1044	990
Part-of-speech	3934	3220
Paragraph	10244	6443
Semicolon Group	43196	59915
Total Terms/Phrases	98924	225124
Unique Terms/Phrases	59768	100470

Table 2.1: Counts of each level of the hierarchy in the 1911 and 1987 Thesauri.

Words and Phrases

A Semicolon Group contains words and phrases. The length of a phrase in words is not limited. Most phrases in the Phrase POS are fairly long, averaging 4.6 words each in the 1911 *Thesaurus*, but phrases in other parts of the *Thesaurus* can be quite long too. Table 2.1 shows the counts of the nine groupings. The Unique Words/Phrases counts each word or phrase once, while Total Words/Phrases, counts each appearance of each word and phrase in the *Thesaurus*.

2.2.2 Content Comparison of the 1911 and 1987 Thesauri

Although the 1987 and 1911 Thesauri are very similar in structure, there are a few differences, among them, the number of levels and the number of parts-of-speech represented.

Table 2.2 shows the frequency of Paragraphs, Semicolon Groups and both total and unique words found in a given type of POS. *Total* is a count of all instances of all words/phrases, while *unique* only counts a particular word/phrase once. Many terms occur both in the 1911 and 1987 Thesauri, but many more appear in just one version or the other. Surprisingly, quite a few 1911 terms do not appear in the 1987 data, as shown in Table 2.3; many of them may have been considered obsolete and thus dropped from the 1987 version. For example “ingrafted” appears in the same semicolon group as “implanted” in the older but not the newer version. Some mismatches may be due to small changes in spelling, for example, “Nirvana” is capitalized in the 1911 version, but

POS	**Paragraph**		**Semicolon Group**	
	1911	1987	1911	1987
Noun	4495	2884	19215	31174
Verb	2402	1499	10838	13958
Adjective	2080	1501	9097	12893
Adverb	594	499	2028	1825
Interjection	108	60	149	65
Phrase	561	0	1865	0
Prefix	2	0	2	0
Pronoun	2	0	2	0
	Total Word		**Unique Words**	
	1911	1987	1911	1987
Noun	46308	114473	29793	56187
Verb	25295	55724	15150	24616
Adjective	20447	48802	12739	21614
Adverb	4039	5720	3016	4144
Interjection	598	405	484	383
Phrase	2228	0	2038	0
Prefix	3	0	3	0
Pronoun	6	0	6	0

Table 2.2: Counts of Paragraphs, Semicolon Groups, total words and unique words by their part of speech; I omitted prefixes and pronouns.

POS	**Both**	**Only 1911**	**Only 1987**
All	35343	24425	65127
Noun	18685	11108	37502
Verb	8618	6532	15998
Adjective	8584	4155	13030
Adverb	1684	1332	2460
Interjections	68	416	315
Phrases	0	2038	0
Prefix	0	3	0
Pronoun	0	6	0

Table 2.3: Counts of terms in either the 1911 or 1987 *Thesaurus*, and in both; I omitted prefixes and pronouns.

not in the 1987 version.

2.2.3 Word Senses in *Roget's*

Word senses in *Roget's Thesaurus* differ from word sense in dictionaries, or even those of *WordNet*. Whereas in dictionaries or *WordNet*, an attempt is made to separate each sense of a word, *Roget's* tries to indicate different facets of a word (Kilgarriff and Yallop, 2000). For example, the word “listless” appears in three heads “Boredom”, “Idleness” and “Apathy”. These are not different senses of the word, but rather indicate different aspects of what it means to be “listless”. In comparison *WordNet* gives two rather similar definitions for “listless”:

- marked by low spirits; showing no enthusiasm
- lacking zest or vivacity

Both definitions suggest feelings of idleness and apathy and to an extent, boredom. Another example is the word “Radio”, which appears in the following locations in the 1987 *Thesaurus*:

- Head: Power (160), paragraph: electronics
- Head: Sound (398), paragraph: sound

- Head: Information (524), paragraph: communicate
- Head: Publication (528), paragraph: publicity and publish
- Head: Communication (531), paragraph: broadcasting
- Head: Amusement (837), paragraph: amusement

Certainly “Amusement” and “Sound” are not distinct senses of “Radio”. Rather a radio is a source of amusement, and produces sound. Many of these instances of “Radio” are linked by cross-references.

Cross-References

Appearances of a word in *Roget's* do not necessarily correspond to a distinct word sense, but this does not mean *Roget's* has no ability to express word senses. There is frequent cross-referencing between words in the *Thesaurus*. Frequently a word will have a reference to a Head or Paragraph which also pertain to that sense of the word. For example the appearances of the word radio in the 1987 *Thesaurus* Heads for Power (160), Sound (398), Information (524), Publication (528) and Amusement (837) have references to the head Communication (531). Although the precise nature of these cross-references is not completely clear it would appear that most of these appearances of “radio” pertain to the same word sense. There should be no need to understand in perfect detail why each instance of a cross-reference is included or why some words are not linked. This would be analogous to requiring a user of *WordNet* to understand precisely how and why each sense of a word was chosen, which as seen above in the example for “listless”, is no easy task. At some level, word-senses do become subjective. For a more detailed discussion of *Roget's* cross references see Old (2009), and for an interesting discussion on word senses in general see Kilgarriff (1997).

There are several examples of cross references in Figure 2.1. For example the second and third SG of the first Paragraph are references to 595 *phraseology* and 608 *speech*.

2.2.4 The Index

Roget's Thesaurus contains an index, almost as large as the rest of the *Thesaurus*. The index in the book version tells which Head, POS and Paragraph that a word appears in. In the Java implementation of *Roget's* the index one can find the Class, Section, ..., Semicolon Group and position within the Semicolon Group in which each word can be

WordNet	***Roget's* Thesaurus**
Entity	Abstract Relations
Psychological feature	Space
Abstraction	Matter
State	Intellect: formation of ideas
Event	Intellect: communication of ideas
Act	Volition: individual volition
Group	Volition: social volition
Possession	Emotion, religion and morality
Phenomenon	

Table 2.4: Comparison of the top levels of the *WordNet* and *Roget's* hierarchies.

found. This index is used extensively for querying words in the *Thesaurus*, in fact some applications will only used the index.

2.3 Comparison with *WordNet*

WordNet organizes data quite differently from *Roget's Thesaurus*. The synsets of *WordNet* are most comparable to the SG in *Roget's* although the synset seems to contain closer synonyms than the SG. Explicitly labeled ANTONYM links indicate words of opposite meaning in *WordNet* while *Roget's* presents opposing concepts through the Head Group. The nouns and verbs of *WordNet* are organized into a hierarchy based on HYPERNYM/HYPONYM relations. The top level of the noun hierarchy in *WordNet* and *Roget's* are shown in Table 2.4. *Roget's* puts more emphasis on ideas and choices. Also each class in *Roget's* will contain a mixture of nouns, verbs adjectives and adverbs whereas only nouns will be found under the *WordNet* entries.

Synsets, and so words/concepts, in *WordNet* can appear at any level in the hierarchy, as opposed to only leaf nodes, as in *Roget's*. For example the word "entity" appears at the very top of the *WordNet* hierarchy, but at the bottom of the *Roget's* hierarchy with all other words. One other important difference is the presence of definitions, or glosses, in *WordNet* that define the terms in a synset and often gives examples of their uses. The dominant sense of the word "trip" has the following definition: a journey for some purpose (usually including the return); "he took a trip to the shopping centre". This

	WordNet	*Roget's* 1911	*WN*-1911 Overlap
Nouns	117798	29681	15307
Verbs	11529	15146	4527
Adjectives	21479	12723	6785
Adverbs	4481	3016	670

Table 2.5: Comparison of the overlap between *WordNet* and the 1911 version of *Roget's* Thesaurus.

	WordNet	*Roget's* 1987	*WN*-1987 Overlap
Nouns	117798	55818	22758
Verbs	11529	24612	5853
Adjectives	21479	21582	10349
Adverbs	4481	4143	1246

Table 2.6: Comparison of the overlap between *WordNet* and the 1987 version of *Roget's* Thesaurus.

differs from the implied semantics that come from neighbouring words in *Roget's*.

In terms of content overlapping there is a great deal that appears in both *Roget's* and *WordNet* but an even larger percentage is only found in one or the other. Table 2.5 and 2.6 show the overlaps between *WordNet* 3.0 and *Roget's* 1911 and 1987 respectively. From these tables one can see that *WordNet* 3.0's coverage of the noun part-of-speech is considerably greater than both versions of *Roget's Thesaurus*, but both versions of *Roget's* contains more verbs. In terms of adjectives and adverbs the 1987 *Roget's Thesaurus* is comparable to *WordNet* 3.0, while the 1911 *Roget's* has somewhat fewer. The actual overlap is not particularly high, rarely much more than 50% of a given part of speech. This is likely due to differences in phrases found in these two resources.

2.4 Ontology, Taxonomy and Classification

It is tempting to refer to *Roget's Thesaurus* as an Ontology as *WordNet* has occasionally been (Saias and Quaresma, 2002; Alfonseca and Manandhar, 2002; Mann, 2002; Veale, 2003; Alfonseca, 2004; Simina and Barbu, 2004; Baek et al., 2008; Zheng et al., 2009; Aversano et al., 2010; Ofoghi and Yearwood, 2010). Rees (2003) attempts to clarify the definition of three words: “classification”, “taxonomy” and “ontology” using the Merriam-Webster dictionary. Classification is: “systematic arrangement in groups or categories according to established criteria.” Rees provides an example of categorizing animals into groups of *tasty*, *edible* and *unedible*. Taxonomy means: “orderly classification of plants and animals according to their presumed natural relationships.” This could include categorizing words into classes of animals such as “mammals” or “carnivores”. Rees (2003) suggests that taxonomy applies to things that can be categorized in an IS-A hierarchy. Ontology is defined as: “a branch of metaphysics concerned with the nature and relations of being or a particular theory about the nature of being or the kinds of existents”, although another definition: “a specification of a conceptualization” is also provided. Hirst (2004) presents another discussion of the relationship between *WordNet* and Ontology. Some of the main differences between what is commonly known as an Ontology and *WordNet* are that an ontology should separate concepts in such a way that there is no overlap. In contrast, *WordNet* contains concepts for “error”, “mistake”, “blunder”, “slip”, “lapse” and “faux pas”, each of which overlap at least partially with each other. Hirst suggests that a lexical-ontology may not be possible outside of the lexicon for a very specific domain. An Ontology is not a linguistic object, while *WordNet* is. Ontology defines a set of concepts from a well specified domain; I do not believe this describes either *WordNet* or *Roget’s* very well.

Given these definitions *Roget’s* seems to fit as a “Classification” better than a “Taxonomy” or an “Ontology”. The divisions in *Roget’s Thesaurus* come with extremely broad and varied relationships between them. In the case of Figure 2.1 one can see that not every word in the Head for “Language” is a kind of language, only that these words have something to do with language. *WordNet* in contrast seems to meet the Rees (2003) definition of a Taxonomy. This is not in itself an advantage to either *Roget’s* or *WordNet* but merely a matter of definitions.

2.4.1 Previous Attempts to Modify *Roget’s Thesaurus* into an Ontology

The 1911 version of *Roget’s* Thesaurus has been built into another lexical ontology called *FACTOTUM* (Cassidy, 2000). Although *FACTOTUM’s* design is based on *Roget’s*, its structure has been changed significantly. The purpose of the work done in *FACTOTUM* was not to create an entire lexical semantic network for the English language, but to create the base for one that could be expanded to be used with many Natural Language Processing applications, and could be easily expanded by developers who required greater functionality. This was based on the assumption that it would take hundreds, or maybe thousands of person-years to create a completely lexical semantic resource and also on the assumption that *Roget's Thesaurus*, in its original state, was not adequate for most Natural Language applications.

Cassidy (2000) made two main changes to *Roget's*. The first is to modify the hierarchy so as to allow what he called "optimal inheritance". This was done by doubling the number of Head words, mostly by adding heads for technical subject matter. The second was to specify the relationship between the head word and the words contained in the Head.

The structure of *FACTOTUM* is designed to model every kind of semantic relationship between concepts. For example to indicate that an object S is "red" one could define relationships like: $red(X)$, or $is\_red(X)$, instead a relationship $has\_property(X) : redness$ will be used.

Concepts like BETWEENNESS can be represented as: $between(river) : leftbank + rightbank$ or $has\_relative\_value(dollar + cent) : 100$. Relationships can be modified with the addition of extra arguments, like $has\_part(bicycle) : wheel[num = 2]$ and $has\_property(gold) : color[val = yellow]$. Functional relationships are also represented: $has\_function(cannon) : propel[obj = shell]$. Predicates can contain modifiers:

- (no modifier) : by default
- & : sometimes
- ! : almost always
- !! : holds by definition

These can be used in situations such as indicating that similarity can sometimes be caused by imitation: $\&caused\_by(similarity) : imitation$. Negations can also be used, like in this definition of an acyclic graph:

$$\{\{has\_subtype(graph)\}\}\ \{\{not\_part\_of(cycle)\}\}acyclic\ graph$$

Contextual relationships are provided as well:

$$\{\{has\_subtype(render\ dry)\}\}\ \{[with\_object(corpse)]\}\ mummify$$

Clearly the implementation of *FACTOTUM* differs significantly from that of *Roget's* Thesaurus. Although it has been available for some years, it has not been widely used in research. One exception was an attempt to identify functional relations in *FACTOTUM* though this was not actually applying *FACTOTUM* to any NLP problems (O'Hara and Wiebe, 2003).

FACTOTUM was an attempt to create an Ontology out of *Roget's* Thesaurus by adding the ability to represent concepts and their attributes. I believe that altering *Roget's* structure in such a way has not proved beneficial. *Roget's* in its current form has been shown to be very valuable for many NLP tasks (Yarowsky, 1992; Jarmasz and Szpakowicz, 2003, 2004).

In my masters thesis I attempted to modify *Roget's Thesaurus* by adding HYPERNYM semantic relationships to the *Thesaurus* (Kennedy, 2007; Kennedy and Szpakowicz, 2007). These only make up a small sub-section of the relationships in *Roget's*, none the less they were able to use this information to improve *Roget's* accuracy at determining semantic relatedness and for solving SAT-style analogy questions.

2.5 Conclusion

Roget's may not strictly match the definition of either a taxonomy or an ontology, but this does not diminish its usefulness to NLP. That fact that *Roget's* does not rely on a rigid IS-A hierarchy means that it can easily express implicit relationships including those between parts-of-speech.

Roget's Thesaurus organizes data much differently that *WordNet*. Some of the main differences are:

- *WordNet* has explicit relationships, *Roget's* does not.
- *Roget's* mixes different parts-of-speech together much more than *WordNet*.
- *Roget's* has a fixed-depth hierarchy where words only appear at the leaf nodes, while the *WordNet* hierarchy is of variable depth and words can be found at every level.
- *Roget's* contains many famous quotes and phrases that *WordNet* does not.
- *WordNet* contains definitions, sample uses and frequency information that *Roget's* does not.

- *Roget's* has been developed over the last 200 years while *WordNet* has been developed over the last 30 years.

These may yet prove to be either strengths or weaknesses. They do show that *Roget's* is a very different resource from *WordNet* and explains why researchers should want to study it for NLP. Many of the observations from this chapter will influence how I go about adding new words to *Roget's Thesaurus*.

Chapter 3

Literature Review

This chapter describes previous research in three sections. Section 3.1 details previous work using *Roget's Thesaurus* for NLP as well as some other possible applications that could be used for evaluating the various versions of *Roget's* against each other and with *WordNet*. Section 3.2 discusses work done on updating the vocabulary of other lexical resources. Section 3.3 outlines methods and resources for clustering semantically related terms/phrases.

3.1 Applications of *Roget's Thesaurus*

There is a long history of using *Roget's* for NLP – for a brief and somewhat dated overview I would recommend Wilks (1998). Some of the earliest work includes using it as an interlingual for machine translation (Masterman, 1956, 1961). This section focuses mostly on more recent work done with *Roget's Thesaurus*.

3.1.1 Semantic Distance

The 1987 version of *Roget's* Thesaurus has been shown to be useful in a variety of applications. In Jarmasz and Szpakowicz (2004) a method of determining semantic similarity, called *SemDist*, between pairs of terms was proposed. The technique works by giving terms that appear closer together in the *Thesaurus* higher relatedness scores than those that are farther apart. A score of 16 was given to a pair of words in the same SG, 14 in the same Paragraph, etc.

In this thesis I repeat some of these experiments, so the methodology will be explained in more detail in Chapter 6. Here I will just describe the nature of the experiments.

Jarmasz and Szpakowicz (2004) tested a *Roget's* based measure of semantic relatedness on three data sets, containing word pairs with human-assigned similarity scores. Human-assigned scores vary between 0 (not related) and 4 (very related). The score assigned to each pair is the average of a set of human scores. The data sets are: Rubenstein and Goodenough (1965) (65 pairs), Miller and Charles (1991) (30 pairs) and Finkelstein et al. (2001)[1] (353 pairs). Spearman's correlation coefficient was used to determine correlation between *Roget's* 1987 and the human annotators.

Another set of tests from Jarmasz and Szpakowicz (2004) is to use *Roget's* 1987 to identify the best synonym for a word, from a set of candidates. Their system was tested on three different data sets: 80 questions taken from the Test of English as a Foreign Language (TOEFL) Landauer and Dumais (1997), 50 questions – from the English as a Second Language test (ESL) Turney (2001) and 300 questions – from the Reader's Digest Word Power Game (RDWP) Lewis (2001).

I replicate these experiments – in Section 6.2 – using both the 1987 and 1911 Thesauri and the updated versions. I also compare them against various semantic similarity measures using *WordNet* version 3.0.

3.1.2 Lexical Chains

In (Jarmasz and Szpakowicz, 2003) a method of creating Lexical Chains is described. Lexical Chains are sequences of words "identifying cohesive regions in a text". The lexical chains built by Jarmasz and Szpakowicz implemented a variation of a method first proposed by Morris and Hirst (1991) who manually constructed lexical chains with the 1911 *Thesaurus*.

There are four steps in building a lexical chain. First a set of candidate words is chosen. A candidate word is any word that appears in *Roget's*, excluding a set of stop words. Second an appropriate chain is found for each word. If a word is already found in an existing lexical chain then that chain is selected. If the word is not already found in a chain, then the chain with the most terms in the same *Roget's* Paragraph is selected. In the third step, the word is placed into the chain. The fourth and last step is to merge lexical chains and keep only the strongest ones.

Two criteria of evaluating these lexical chains, strength and quality, are discussed. Strength can be measured by how repetitious the words in the chain are, how long the chain is and how dense it is while quality is better measured through applications.

[1] http://www.cs.technion.ac.il/~ gabr/resources/data/wordsim353/wordsim353.html

Jarmasz and Szpakowicz (2003) propose malapropism detection or text summarization as a possible application based evaluation of the lexical chains, though no actual application-based evaluation is performed.

3.1.3 Word Sense Disambiguation

Some work on word sense disambiguation (WSD) has been done with *Roget's*. Yarowsky (1992) proposed a method of disambiguating words into whichever *Roget's* Head they appear in. This method works by discovering "salient" words that frequently co-occur in context with words from a given Head. These salient words are given probabilities of appearing with a word w from the given Head: $P(w|Head)$. The salient words need not be part of the *Roget's* Head or indeed any Head in *Roget's*. A word t is disambiguated into its Head, $category(t)$, in the following way:

$$category(t) = \underset{Head}{\arg\max} \sum_{w\ near\ t} log \frac{Pr(w|Head) \times Pr(Head)}{Pr(w)}$$

This is done for a window of 50 words to the left and right of t. This method was tested on 12 polysemous nouns: "star", "mole", "galley", "cone", "bass", "bow", "taste", "interest", "issue", "duty", "sentence" and "slug", and an accuracy of 92% was reported. One possible criticism is that the evaluation was done on an encyclopedia where these words are likely to only use one sense per article. Given that and the extremely large window used for contexts it seems probable that this algorithm would not work as well on a different domain.

In Kwong (1998a,b) a form of WSD is described where word senses in *Roget's* are mapped to senses in *WordNet*. This system is further described in Section 3.2.

3.1.4 Information Retrieval

Roget's Thesaurus has been used for the purpose of Information Retrieval (Mandala et al., 1999). A word w has a feature f if w and f appear in the same Head. Essentially this means that the features of a word w are all other words that appear in the same head with w, and w in turn will be a feature for every other word in that Head. A word is represented by a vector of its features. A similarity score is assigned to pairs of words w_1 and w_2 using the Dice coefficient of the two feature vectors $R(w_1)$ and $R(w_2)$:

$$sim(w_1, w_2) = \frac{2|R(w_1) \cap R(w_2)|}{|R(w_1)| + |R(w_2)|}$$

This measure, along with the Resnik (1995) measure for *WordNet* and three other distributional measures of relatedness extracted from a large text were evaluated for query expansion. All the words in *Roget's*, *WordNet* or the distributional thesauri were ranked according their similarity to the query terms. The terms with the highest similarity to query terms were used to expand the query. Of the techniques applied *Roget's* had the smallest improvement, though it was not far behind *WordNet*. Ultimately it was still a successful technique for query expansion.

3.1.5 Visualization of *Roget's Thesaurus*

There has also been related work on attempts to visualize *Roget's Thesaurus*. Old (2002) discusses spatial methods for the display of the content of *Roget's Thesaurus*. This line of work was continued in (Old, 2003) where overlap between propositions are represented in lattices. In (Old, 2004) a visualization technique called Formal Concept Analysis (Wille, 1981) is used to help decipher the complex relationships within *Roget's*. In all of these works visualization is the main tool and so does not really lend itself to automatically updating the lexicon of *Roget's*. Nonetheless it does show how computational techniques can be used to extract implicit information from the *Thesaurus*. In all cases work was done using the 1962 version of *Roget's Thesaurus*.

3.1.6 Other Languages

Some attempts have been made to either translate *Roget's* into French or align it with a French Thesaurus. In (de Melo and Weikum, 2008) *Roget's* and *WordNet* were translated into French. A union of three French-English dictionaries was used to come up with candidates for translation. A number of these samples were manually annotated and used to train an SVM for determining the correct translation. A variety of features were used some of which measured counts of how closely related the candidate translations were.

In (Prince and Chauch, 2008) *Roget's Thesaurus* was mapped to the French *Larousse Thesaurus*. Each word was represented by the set of Heads in which it appears in either *Roget's* or *Larousse*. A matrix was manually built to do mapping between the categories in these two thesauri. A set of possible translations for each word were taken from a bilingual dictionary and then the one whose vector is closest to the word being translated was selected as the correct translation.

According to Kendall (2008), German and Spanish versions of *Roget's* have been compiled. These translations were done many decades ago and the process was not automated.

3.1.7 Proposed but Untested Methods

Kilgarriff (2003) proposed several method for evaluation of Thesauri based on known NLP problems. These were proposed as methods for evaluating *WASP-bench*, a tool for performing lexical disambiguation to aid in the construction machine translation systems (Kilgarriff and Tugwell, 2001).

Parsing

Two proposals involve parsing. The first is identifying prepositional phrase (PP) attachment. For example in the following two phrases "eat fish with a fork" and "eat fish with bones", a thesaurus could be used to determine that "with a fork" is attached to "eat" while "with bones" is attached to "fish" using some sort of semantic relatedness method. A second use is in identifying scope. For example in the sentence "old boots and shoes", "old" will likely refer to both "boots" and "shoes", while in the sentence "old boots and apples", "old" will likely apply only to "boots" and not "apples". Possibly a thesaurus could be used to identify phrases like "old boots" and "old shoes" and know which words "old" will be applied to. That said, it is not clear how often such information would be available in a thesaurus. I am uncertain that this will work with high accuracy, as it seems optimistic that a relation between "old" and "boots" would be found in any given thesaurus.

Anaphora Resolution

Another use for Thesauri is to bridge anaphora resolution. In the sentences "Maria bought a beautiful apple. The fruit was red and crisp." the proximity of the words "apple" and "fruit" can indicate that they are describing the same object, as long as one knows that an apple is a fruit. This would suggest a hypernymy relationship between the words, which is not made explicit in *Roget's Thesaurus*.

Measuring Text Cohesion & Word Sense Disambiguation

Text cohesion is the task of breaking down discourse into segments in such a way that each segment discusses a single cohesive topic. This could possibly be accomplished with the use of lexical chains (Jarmasz and Szpakowicz, 2003). Word sense disambiguation is the task of identifying which sense of a word is being used in a sentence. For example in the sentence "We caught a pike that afternoon." the word "pike" can be deduced to mean the fish since fish are often caught. A pike can also be a weapon, but this sort of pike is not generally caught. As seen already, Yarowsky (1992) conducted some experiments using *Roget's* for Word Sense Disambiguation. I perform some pseudo-word-sense disambiguation experiments in Section 6.5.

Ontology

The last described method of evaluating a thesaurus in (Kilgarriff, 2003) is to use it as an ontology. The purpose of an ontology in this case is to represent the meaning of some text and extract implicit information from this string. For example in the sentence "Fido is a cat", an ontology could extract information such as that cats are animals. It could also state that since cats are alive Fido must be alive too. Kilgarriff points out that since this is not a strictly linguistic use of the resource, it may be dangerous to use a thesaurus as an ontology. By the definitions of Ontology, Taxonomy and Classification described in Rees (2003) it does not appear safe to use *Roget's Thesaurus* as an Ontology (see Section 2.4). To an extent this was implemented in *FACTOTUM* (Cassidy, 2000).

3.2 Expanding Thesauri and Lexical Ontologies

As discussed before, Cassidy (2000) manually added around 1000 terms and phrases to the 1911 *Roget's Thesaurus* when constructing *FACTOTUM*. It is not always clear which words and phrases were added by Cassidy and which already appeared in *Roget's*. In this section I examine automatic methods of updating and expanding thesauri.

To my knowledge no work has been published on automatically enhancing *Roget's* Thesaurus with new terms. I have previously done some work on disambiguating relations in the 1987 *Roget's Thesaurus* (Kennedy and Szpakowicz, 2007). This work attempted to identify hypernymy-related terms within the same Paragraphs in the *Thesaurus* by identifying these pairs in other resources including *WordNet*, dictionaries like *LDOCE* (Procter, 1978) and extracted from corpora, particularly the *BNC* (BNC, 2007) using set patterns (Hearst, 1992). Other work on modifying *Roget's* mostly has focused on mapping it to thesauri in other languages or translating it (see Section 3.1.6).

3.2.1 Merging *Roget's* and *WordNet*

Kwong (1998a,b) does not quite attempt to expand any thesauri, but rather merges resources. *Roget's* 1987 word senses are mapped to *LDOCE* definitions. In this application *WordNet* is used as an intermediary for mapping between those two resources. This process is done in six steps.

1. First Kwong retrieves all the definitions of a given word w from *LDOCE*.

2. Second she collects all the synsets from *WordNet* that contained w and their corresponding glosses.

3. The third step is to build a matrix A where similarity scores are given between every synset from *WordNet* and every definition from *LDOCE*. The score is a weighted sum of the overlaps between *LDOCE* senses and *WordNet* synsets, HYPERNYMS of those synsets and their glosses.

4. The fourth step is like the first two, but involved selecting the Paragraphs from *Roget's* in which w appears.

5. Step five creates another similarity matrix B between *Roget's* and *WordNet* in the same fashion as step three, but using *Roget's* Paragraphs instead of *LDOCE* definitions.

6. In the last step, for each *LDOCE* sense i Kwong found $max(A(i,j))$ where j is a *WordNet* sense and then found $max(B(j,k))$, thus mapping the *LDOCE* sense i to the *Roget's* sense k.

This method was evaluated on a set of 36 nouns. The nouns were divided into 3 different sets, one containing high polysemy words (11 or more senses), medium polysemy (6-10 senses) and low polysemy (1-5 senses). Mappings from *LDOCE* to *WordNet* and from *WordNet* to *Roget's* were evaluated on this set. Mapping from *LDOCE* to *WordNet* for low polysemy words was about 65% accurate, for medium polysemy 66% accurate and for high polysemy words was 53% accurate. When mapping between *WordNet* and *Roget's* Kwong (1998a) found mappings for low polysemy words to be 79% accurate while mappings for both medium and high polysemy words were about 70% accurate.

Nastase and Szpakowicz (2001) experimented with a similar technique mapping *Roget's* 1987 to *WordNet* without the use of *LDOCE*. Evaluation of the technique was also expanded, this time tests were done on a set of 719 nouns. A precision of about 55% was found where nouns were correctly mapped between *Roget's* and *WordNet*, and 66% when allowing for ties.

3.2.2 Enhancing *WordNet*

In (Snow et al., 2006) a variety of new words were extracted from a corpus and added to *WordNet*. Many of the new terms are proper nouns which were discovered in a corpus using a machine learning system that discovers IS-A relationships using dependency paths generated by *Minipar* (Lin, 1998b) – described in Snow et al. (2005). A corpus was built using known hypernyms and non-hypernyms from *WordNet* and parsed using *Minipar*. A dependency paths is the paths through a parse tree connecting two words. Each kind of path was used as a feature and the value of that feature was how many of these paths connect two particular words. Experiments were carried out with Logistic Regression, Multinomial Naïve Bayes and Complement Naïve Bayes. The best system was a version of Logistic Regression which had an f-measure of 0.348. Lists of additional 10000, 20000, 30000 and 40000 words for *WordNet* 2.1 were discovered using this method were generated and made available.[2]

Another paper that focuses on expanding *WordNet* is (Pantel, 2005). In this paper semantic vectors are created for each word in *WordNet* by disambiguating contexts which appear with different senses of a word. The process of building these semantic vectors is described in (Pantel, 2003) and also in Section 3.3.2. The hypernym hierarchy of *WordNet* is used to propagate contexts where words may appear throughout the network. A word sense can then be represented by contexts from its semantic vector that are not shared with its parents. This research did not actually attempt to place new words into the resource, but rather evaluated it on existing words. To my knowledge no one has taken this method up as a means of actually updating *WordNet*. Additionally this technique was only examined for nouns and although presumably applicable to verbs, could not be used for adjectives or adverbs because they have no hypernym hierarchy in *WordNet*.

More recently experiments have been done using folksonomies to discover hypernym relationships which can be used to incorporate new words into *WordNet* (Zheng et al.,

[2] http://ai.stanford.edu/~rion/swn/

2008). Folksonomies are web services that allow users to freely annotate web sites or anything with strings of their choice. One such folksonomy was *Delicious* where users tag web pages. HYPERNYM/HYPONYM relationships can be extracted from these folksonomies by identifying tags that subsume other tags. In Zheng et al. (2008) this technique was put forward to discover HYPERNYM/HYPONYM relationships that could be added to *WordNet*. In their evaluation they produced 274 HYPERNYM/HYPONYM pairs where 192 (70%) were found in *WordNet*.

Not directly applicable but still relevant to this research is work on semi-automatically enhancing *WordNet* with sentiment (Esuli and Sebastiani, 2006) and affect (Strapparava and Valitutti, 2004) information. *SentiWordNet* (Esuli and Sebastiani, 2006) is an attempt to label synsets in *WordNet* 2.0 as objective, positive or negative. Three different scores are given for these three possibilities where all three sum up to 1.0 for each synset. A hand-labeled set of known positive and negative terms is used to train a classifier to identify which synsets in *WordNet* contain positive, negative or objective terms. A similar kind of enhancement is found in *WordNet Affect* (Strapparava and Valitutti, 2004). In *WordNet Affect* a set of synsets are labelled with one or more labels, often related to emotion. This is done by hand-building an initial set of words with these emotions and then using the relationships in *WordNet* to propagate these emotions to other synsets. This work was based on the *WordNet Domains* (Magnini and Cavagliá, 2000) which is a framework that allows one to augment *WordNet* by adding domain labels to synsets. Obviously these projects do not deal with adding new terms to a thesaurus, but they do highlight some of the more successful experiments for enhancing *WordNet*.

There has been a fair bit of work done on mining hypernym relationships from text. Although this sort of work is not necessarily focused on updating *WordNet*, as in (Snow et al., 2006), in most cases it could be applied to this task. In (Hearst, 1992) a set of six patterns were used to extract hypernyms from text. An example pattern is:

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

Hypernym relationships were extracted from Grolier Encyclopedia and although *WordNet* was not enhanced with these new relationships, it was used for evaluation. 57% of the relationships mined from Grolier in this way were found in *WordNet*. In (Sombatsrisomboon et al., 2003) this sort of method was expanded upon by using the Google API to search the Internet for hypernyms or hyponyms of a specific *term* by using the queries:

- *term* is a/an *

- * is a/an *term*

In (Morin and Jacquemin, 1999) pairs of known hypernyms were identified in a corpus for the purpose of discovering new patterns that are likely to yield even more hypernym relationships. Some work has been done on mining hypernyms in other languages including Swedish (Rydin, 2002), Dutch (Sang, 2007) and Japanese (Shinzato and Torisawa, 2004). This sort of work has also been replicated in searching for MERONYM/HOLONYM relationships (Girju et al., 2003, 2006).

Non-English Wordnets

A great deal of work has been done on wordnets in languages other than English. The Global WordNet Association[3] is a group that helps organize efforts to create wordnets for many languages. Their earlier goals were to use the Princeton *WordNet* as a starting point for *EuroWordNet* (Vossen, 1998) which was a project aimed at creating wordnets for all European languages. Some of these European wordnets include *BalkaNet* (Mititelu et al., 2006) which includes Bulgarian, Greek, Moldavian, Romanian, Serbian and Turkish into one *WordNet*. By contrast many language have multiple wordnets, for example Russian: *RussNet* (Azarova et al., 2002) and *Russian WordNet* (Balkova et al., 2004).[4] The Global WordNet Association also organizes bi-annual conferences where researchers can present and discuss research on constructing wordnets.

There are a few more noteworthy examples of non-english *WordNets* that I take some inspiration from. In Piasecki et al. (2009a) a tool called *WordNet Weaver* was presented for expanding the vocabulary of a wordnet; in particular it was applied to the construction of a Polish *WordNet*. The algorithm works in two phases. The first identifies a synset in which to place a new word, while the second phase connects possible candidate synsets. The *WordNet Weaver* does not actually add new words to the Polish *WordNet*, but rather suggests them to a linguist who selects which additions should be made. In Lemnitzer et al. (2008) semantic relationships between nouns and verbs were added to a German *WordNet*. Effectively they were adding verb-object relationships which they believe would be useful for applications including text summarization and anaphora resolution.

[3] http://www.globalwordnet.org/

[4] See www.globalwordnet.org/gwa/wordnet_table.htm for a full list of available wordnets

3.3 Measures of Semantic Relatedness

Measures of semantic relatedness (MSR) capture similarity in meaning between two words. These methods can generally be divided into two groups: those based on lexical resources such as *Roget's* or *WordNet* and those based on distributional similarity of words in a corpus. These methods are not quite mutually exclusive either. Some hybrid approaches have been attempted. In this section I describe relevant work on MSRs.

3.3.1 Resource-Based MSRs

MSRs between words in lexical resources tend to use edge distance between words as a central component of their MSR. Some examples of this are noted in the work of Jarmasz and Szpakowicz (2004), Pedersen et al. (2004), Leacock and Chodorow (1998), Wu and Palmer (1994) and Hirst and St-Onge (1998). In Jarmasz and Szpakowicz (2004) the number of edges between two words in *Roget's Thesaurus* was used to measure semantic relatedness, while Path (Pedersen et al., 2004) measures distance by the number of hypernym links between two words. Leacock and Chodorow (1998) proposed a measure similar to Path, only the distance is normalized by the depth of the *WordNet* hierarchy. As a result, these measures can only measure relatedness between two nouns or two verbs in *WordNet*, while the *Roget's* based measure can measure distances between any two words, noun, verb, adjective or adverb. The measure in Hirst and St-Onge (1998) makes use of every kind of semantic relation in *WordNet* when finding the path between two words. In Resnik (1995), Jiang and Conrath (1997) and Lin (1998a) different variations on information content were used to determine semantic relatedness. Another method Banerjee and Pedersen (2002) uses the overlap between two glosses in *WordNet* to determine semantic relatedness. This is an implementation of the Lesk algorithm. Although this could be used to measure relatedness between any pair of words in *WordNet* the package provided by Pedersen et al. (2004) only allows for similarity to be calculated between pairs of words with the same part-of-speech.

Another MSR using *Roget's* is described by Mandala et al. (1999) (see Section 3.1.4). This one is more comparable to Lesk than the other edge distances. I will not make use of this MSR, though it is useful to note. In Kennedy and Szpakowicz (2007) and Kennedy (2007) I proposed a method of measuring semantic relatedness that mixed that of Jarmasz and Szpakowicz (2004) with other information added to *Roget's*, hypernym relationship links. I will not make use of this measure of semantic relatedness in this thesis either.

3.3.2 Distributional MSRs

The idea for distributional MSRs comes from work in linguistics where it was hypothesized that a word can best be described by the context in which it appears (Harris, 1954; Firth, 1957). This work along with early work on vector space approaches for document similarity (Salton, 1971) make up much of the underlying theory in this section. A much more detailed description and analysis of distributional semantics for a variety of problems can be found in Turney and Pantel (2010). Mohammad and Hirst (2006c) also performed for a large survey of this area.

According to Kilgarriff and Yallop (2000) there are two main ways of measuring how two words are related. One is to find words that appear near each other frequently. These are called first order affinities, an example of this could be “Ottawa” and “Senators”. The second type of relationship is where one word frequently appears in the same contexts as another word. These are known as second order affinities and are more frequently used for determining when two words are synonyms. An example of this could be “Senators” and the abbreviation “Sens”.

Closely related words are easier to model with second order affinities but more loosely related words are easier to model with first order affinities (Kilgarriff and Yallop, 2000). In fact, combining these two methods is possible. The strength of the first order affinities can be used as feature values in a vector for determining second-order affinities.

The first step in creating a distributional MSR is to create a *word-context* matrix from which vectors representing words can be extracted. Commonly the cosine similarity between vectors representing two words can be used to measure their relatedness. In theory cosine similarity is one of only many techniques for measuring the similarity between two words. Jaccard (1901) and Dice (1945) are two other methods of measuring distance between two vectors that have been applied to word similarities. Still other methods have been proposed, including using some balance of precision and recall from the two vectors (Weeds and Weir, 2005). Nonetheless cosine similarity has proven successful and appears to be the most common method used.

The Word-Context Matrix

Before one can measure distances between vectors, one must first construct the *word-context* matrix. The first question is: what constitutes a *context*? Some of the earliest was the work of Crouch (1988): documents are used as contexts to measure relatedness between low-frequency words. The theory behind this is that low-frequency words that appear in related documents are more likely to be synonyms. Documents as contexts are no longer commonly used in this line of research; for the most part kinds of contexts can be divided into two groups, those that use neighbouring words found using a sliding window, and those that use syntactically related words determined using a parser. Some work where a sliding window was used includes Schütze and Pedersen (1997), Yoshida et al. (2003) and Mohammad and Hirst (2006b,a). Ruge (1997) used head/modifier relationships to discover synonyms. A noun is represented by a vector of other words that can modify that noun. If two words tend to have the same modifiers, then they are most likely semantically related. This idea that the semantics of a word can be best represented by the context in which it appears will be expanded on much more in this section. Other methods, such as SEXTANT (Grefenstette, 1994), Clustering By Committee (Pantel and Lin, 2002) and the work of Padó and Lapata (2007) work under similar principles.

In Yang and Powers (2008) a method of constructing thesauri from a parsed BNC (BNC, 2007) is described. The Link Parser (Temperley and Sleator, 1993) was used to obtain verb-adverb, noun-adjective, verb-object and verb-subject relationships. Each noun or verb was represented using a vector of the contexts in which it appears. These vectors were reduced to just 250 features using Singular Value Decomposition (SVD) (Deerwester et al., 1990; Landauer and Dumais, 1997). SVD is a technique for reducing the dimensionality of word vectors and also has been found to reduce noise. The distance between the word vectors is measured with cosine similarity.

Rychlý and Kilgarriff (2007) proposed and tested what they call a "fast" algorithm for building thesauri. A shallow parser was used to generate triples representing two words and a relationship between those words $\langle w, r, w' \rangle$. From this they identified the context $\langle r, w' \rangle$ of each word w and a score for the triple. Several heuristics were applied to speed up the algorithm. For example, if a context appears with more than 10,000 words then the context can be skipped, as these contexts are not very meaningful for similarity. The exact semantic similarity function is not explained in Rychlý and Kilgarriff (2007), though they reference Weeds and Weir (2005) and Curran (2003). This technique was used to generate large thesauri in several languages including Chinese, English, French, Italian, Japanese, Portuguese, Slovene and Spanish to be found on the Sketch Engine server.[5]

Curran and Moens (2002) used similar methods of clustering nouns together using relations from a parser. Four kinds of relationships were considered for the context of a

[5] http://www.sketchengine.co.uk

word:

- term is the subject of a verb
- term is the (direct/indirect) object of a verb
- term is modified by a noun or adjective
- term is modified by a prepositional phrase

Frequencies of these relationships were counted and a variety of systems were proposed for finding similarity between pairs of terms, represented with these relationships. These included cosine similarity (Salton and McGill, 1983), Dice (Dice, 1945) and Jaccard (Jaccard, 1901). Curran and Moens (2002) gather a "gold standard" of synonyms from *WordNet*, *Roget's* 1911 and Moby Thesaurus (Ward, 1996).[6] Ranked lists of synonyms were generated for seventy different test words.

Pointwise Mutual Information

The first measure I will describe in detail is Pointwise Mutual Information (PMI). PMI measures the association between two events. It has been used numerous times for measuring distributional similarity, starting with Hindle (1990). PMI differs from the information theory definition of Mutual Information, which is taken between two random variables rather than events. Pantel (2003) defines PMI between two events x, and y as $mi_{x,y}$:

$$mi_{x,y} = \frac{P(x,y)}{P(x)P(y)}$$

In a vector space model for a word or term e the vector is represented as $C(e) = (c_{e1},\ c_{e2},\ \ldots\ c_{em})$ where c_{ef} is the frequency of feature f and m is the number of features. A mutual information vector is the same vector, but where each feature is re-weighted with mutual information $MI(e) = (mi_{e1},\ mi_{e2},\ \ldots\ mi_{em})$. This way the features of the vector are given higher weight the more frequently they appear with the term e and less weight if they frequently appear with many terms. Pantel (2003) employed cosine similarity to determine the distance between two mutual information vectors.

This method of vector representation for determining semantic distance between words was put to use in an algorithm called Clustering by Committee (CBC) (Pantel and Lin, 2002). CBC works in three steps. First it finds the top k similar terms,

[6] http://icon.shef.ac.uk/Moby/

where k is a number in 10 .. 20. In phase two a tight cluster is created from these top k elements such that the intra-cluster similarity is high. This is repeated until no more committees can be built. In the third step terms not found in a committee can be assigned to one or more committees depending on the user's requirements.

I will make a great deal of use of this measure in my own work. It is a particularly good one and quite simple at the same time. In fact almost any measure of association could be applied to this sort of Measure of Semantic Relatedness (MSR), for example Z-score (Broda and Piasecki, 2008).

A variation on this methodology has been proposed and tested by Islam and Inkpen (2006). Rather than using dependency triples, all neighbouring words within a sliding window were used as contexts. The contexts were re-weighted using PMI and then the top β of these contexts were selected to represent a given word. A normalized sum of the overlapping words in each vector was taken to assign a similarity score.

Turney (2001) applied PMI to determining synonymy using the Web. PMI was used to calculate the relatedness between two words based on the count of Web articles in which they co-occur. This was used to decide whether a pair of words are synonyms. In Turney (2002) PMI was used to determine word sentiment by identifying contexts that tended to co-occur with words of positive or negative sentiment.

The work of Pantel and Lin (2002) was used as a tool for explaining word similarity in Vyas and Pantel (2008). They used the contexts in which words co-occur in order to explain how these words are similar. For example the entities "Palestinian-Israeli" and "India-Pakistan" are described with the following co-occurring terms/relations: *talks(NN), conflict(NN), dialogue(NN), relation(NN), peace(NN).*

Lin

The next measure I will discuss was proposed by Lin (1998a). Although I do not make use of this measure directly, it is an important and noteworthy one. This measure is based on a theory that the similarity between two words can be defined by the amount of information contained in their common contexts. Contexts of a word are the dependency triples $\langle w, r, w' \rangle$ where w is the word, w' is another word, and r is the dependency relation joining these two words. The frequency of a triple is denoted $||w, r, w'||$. A wild card $*$ can be used in place of a word or relation to indicate all matching words or relations. E.g. $\langle w, *, * \rangle$ will indicate every instance of the word w regardless of which relation and word are in its context.

In making the Lin measure three events are defined:

- A: a randomly selected word is w
- B: a randomly selected dependency type is r
- C: a randomly selected word is w'

P_{MLE} is a maximum likelihood estimation of a probability distribution.

$$P_{MLE}(B) = \frac{||*, r, *||}{||*, *, *||}$$

$$P_{MLE}(A|B) = \frac{||w, r, *||}{||*, r, *||}$$

$$P_{MLE}(C|B) = \frac{||*, r, w'||}{||*, r, *||}$$

The information content in $||w, r, w'||$ is defined as :

$$\begin{array}{rcl} I(w, r, w') & = & -log(P_{MLE}(B)P_{MLE}(A|B)P_{MLE}(C|B)) - (-log\ P_{MLE}(A, B, C)) \\ & = & log\frac{||w,r,w'||\times||*,r,*||}{||w,r,*||\times||*,r,w'||} \end{array}$$

It is worth noting that $I(w, r, w')$ is actually the mutual information between w and w'. $T(w)$ is then defined as the set of pairs (r, w') such that $log\frac{||w,r,w'||\times||*,r,*||}{||w,r,*||\times||*,r,w'||}$ is positive. Using this a similarity measure between two words is defined as follows:

$$sim(w_1, w_2) = \frac{\sum_{(r,w)\in T(w_1)\cap T(w_2)}(I(w_1, r, w) + I(w_2, r, w))}{\sum_{(r,w)\in T(w_1)} I(w_1, r, w) + \sum_{(r,w)\in T(w_2)} I(w_2, r, w))}$$

This method was tested by generating clusters of the most closely related words using the Lin measure, groups from *WordNet* and *Roget's Thesaurus.* Groups of the 10 closest words to a given query word were found and similarity scores were calculated based on how many words from these groups overlapped. The average similarity between corresponding entries was 0.21 for *WordNet* and 0.15 for *Roget's.* This was compared to a few other methods including cosine similarity and two other similarity functions reported in Hindle (1990). Lin's measure performed best. I perform a similar sort of evaluation using various recall points, other than simply the 10 closest words. This will be seen in Chapter 4.

Relative Feature Focus

Geffet and Dagan (2004) proposed Relative Feature Focus (RFF) as a method of enhancing Lin's measure (1998a). They claim that Lin's measure has a problem: high scores for words that are related but cannot be substituted for each other. This is partially due to the fact that the features in the Lin measure may contain very general terms that are not indicative of the word they are defining. An example is given that some terms with high weights to define "country" include "airspace" and "landlocked" which are quite specific to some countries, but also "destination" and "ambition" which are extremely general and need not apply to a country at all.

Relative Feature Focus attempts to re-weight features in such a way that those with strong focus to a terms meaning are ranked much higher. This is done by identifying features that appear in closely related terms (these terms are discovered as in (Lin, 1998a)). These features are then re-weighted giving them an RFF weight as follows:

$$RFF(w, f) = \sum_{v \in WS(f) \cap N(w)} sim(w, v)$$

where $WS(f)$ is the set of words with feature f and $N(w)$ is the set of neighbouring words to w. $sim(w, v)$ is the Lin defined similarity between two terms w and v. Similarities were re-computed by taking the top 100 features for each word and then re-computing Lin's sim function, this time using the RFF weight instead of mutual information MI.

For evaluation Geffet and Dagan randomly selected 30 nouns and found the top 40 most similar words creating a set of supposedly closely related word pairs. This set of word pairs was split in half and two judges labeled each pair as either being substitutable, or not. Substitutable means that in some context the two words can be used interchangeably. The results showed RFF outperforming Lin's measure by about 9-10%.

This method is interesting as it applies one method in conjunction with another in order to enhance the results. I will do something similar in Chapter 4.

Co-occurrence Retrieval Methods

A system for non-symmetric measures of distributional similarity was proposed by Weeds and Weir (2005). This comes from the observation that a word like "dog" can almost always be replaced by a hypernym such as "animal", whereas the reverse is not true. Distance between two words is determined using the precision and recall of the contexts in which both words appear. $D(w,c)$ is defined as the weight of a word w with a feature c and $F(w)$ is the set of features for that word. The set of true positives is their intersection: $TP(w_1,w_2) = F(w_1) \cap F(w_2)$, which can be abbreviated to just TP. Precision and recall are then defined as:

$$P(w_1,w_2) = \frac{\sum_{TP} D(w_1,c)}{\sum_{F(w_1)} D(w_1,c)}$$

$$R(w_1,w_2) = \frac{\sum_{TP} D(w_2,c)}{\sum_{F(w_2)} D(w_2,c)}$$

This was tested using a variety of different weighting functions including type-based, token-based, mutual information, weighted mutual information, t-test, z-test and log-likelihood ratio tests. Two versions of each function were used, one "additive" and one "difference-weighted". I will not explain the differences here, but recommend seeing (Weeds and Weir, 2005) for a more detailed explanation.

Precision and recall are then combined together into a single measure. Two functions are combined to do this. The first function is simply the f-measure or harmonic mean:

$$m_h(P(w_1,w_2), R(w_1,w_2)) = \frac{2 * P(w_1,w_2) * R(w_1,w_2)}{P(w_1,w_2) + R(w_1,w_2)}$$

The second function is a weighted average of precision and recall where β is the weight.

$$m_a(P(w_1,w_2), R(w_1,w_2)) = \beta P(w_1,w_2) + (1-\beta) R(w_1,w_2)$$

These two functions are combined into another weighted function where γ is the weight.

$$sim(w_1,w_2) = \gamma * m_h(P(w_1,w_2), R(w_1,w_2)) + (1-\gamma) * m_a(P(w_1,w_2), R(w_1,w_2))$$

This gives two free parameters β and γ that can be manipulated to tune a similarity measure.

Weeds and Weir (2005) test this system on two applications. Clustering related terms together was the first application, the second was pseudo-word-sense disambiguation. 1000 high frequency nouns and 1000 low frequency nouns were taken from *WordNet* and their distances from each other are found using the semantic relatedness measure proposed by Jiang and Conrath (1997). The best similarity function was found to be the additive mutual information for the high-frequency nouns and the additive t-test method for low-frequency nouns. The best variation for the high-frequency words had a score of 0.34, while the best system for the lower-frequency words had a score of 0.28.

A second evaluation technique they used was pseudo-word-sense disambiguation. In this task a large set of noun-verb pairs $\langle n, v \rangle$ are extracted from a corpus. A second verb is then added to make a noun-verb-verb triple $\langle n, v, v' \rangle$. The task is to determine which verb is more likely to take the noun as its direct object. Although this is an artificial task, it has become a common evaluation technique for work of this kind (Weeds and Weir, 2005). The k-nearest neighbours of n are given weighted votes to determine which verb is the correct one. The weight of each vote is the difference between frequencies of the verb in n and the nearest neighbour. This was tested using all the same variations of the functions described above. Once again the additive t-test method worked best for high frequency nouns and mutual information worked best for low frequency nouns.

One reason why this measure is of some significance is that Weeds and Weir (2005) identify that measure of association, be it PMI, or T-test are actually parameters of a semantic distance measure. Also to my knowledge this is the first measure to introduce a tuning phase, which I will expand upon. The various parameters of this measure can be adjusted in order to make it more suitable for a particular task. Even so they are just scratching the surface of what is possible. Once again the reader will see in Chapter 4 the influence this work has on my own.

Rank Weight Functions

Rank Weight Functions (RWF) is a method of weighting features based on their rank rather than based on the frequency with which those features occur (Piasecki et al., 2007; Broda et al., 2008). This process can be applied to a variety of different measures. The procedure is done on a matrix M of words w and features c in 5 steps:

1. Every entry in a matrix $M[w_i, c_j]$ is the number of times feature c_j appears with lexical unit w_i.

2. Weights are re-calculated according to some function f_w (for example PMI) such that: $\forall_c M[w_i, c] = f_w(M[w_i, c])$.

3. Features in the vectors $M[w_i, \bullet]$ are sorted in ascending order.

4. The k highest ranking features are selected; e.g $k = 1000$.

5. For each feature c_j assign a new value:$M[w_i, c_j] = k - rank(c_j)$.

This method was tested for building synsets for a Polish *WordNet* (Piasecki et al., 2009b). The experiments found that using RWF improved the accuracy of the synsets.

Broda et al. (2009) proposed an enhancement of RWF called Generalized Rank Weight Functions (GRWF). Unlike RWF, GRWF does not assume that features are linearly ordered. Instead the new score for each cell in the matrix is computed as $M[w_i, c_j] = f_{top}(M[w_i, \bullet]) - f_{por}(c_j)$ where f_{top} is the highest rank and $f_{por}(c_j)$ calculates the position of c_j. f_{top} can be set to k as is the case in RWF, or it can be set so that $f_{top}(M[w_i, \bullet]) = size(M[w_i, \bullet]) + 1$ so that the value of the best feature depends on the number of relevant features. $f_{por}(c_j)$ can either be set to the position of feature c_j as in RWF, or it can be adjusted to allow for ties in which position j can be occupied by several features.

Experiments were done using Lin, PMI and z-score feature weighting. These new measures were evaluated by using them to try and identify synonyms from *WordNet*. It was found that RWF improved scores for Lin and PMI over their unmodified scores, and GRWF improved even further over RWF. For the z-score tests were only done with RWF and GRWF, but the improvements found for GRWF over RWF were not consistent and depend very much on the value of k that was selected.

Much like RFF (Geffet and Dagan, 2004) this method works on top of another MSR with the goal of enhancing the MSR. This work also explores whether the score or rank of a lexical feature of some type is more important. Ultimately the authors found that rank was better than score. I will also perform such an experiment in Chapter 5 when determining whether the score or rank of neighbouring words is more important for identifying when to place a new word into *Roget's Thesaurus*.

Latent Semantic Analysis

A widely used method of reducing the dimensionality of a *term-context* matrix is Latent Semantic Analysis (LSA) (Landauer and Dumais, 1997). This technique effectively maps the vectors for each word into a lower dimensional space. The axes in the matrix are ranked by their variance and the top k axes are then selected as features for the new term vectors. This is accomplished using Singular Value Decomposition (SVD). In SVD a matrix X is decomposed into two orthogonal matrixes U and V and Σ is a diagonal matrix such that $X = U\Sigma V^T$. A new matrix $U_k\Sigma_k$ is created using the top k columns. Sometimes a parameter p is used to adjust the weights of the factors in Σ leading to a new matrix $U_k\Sigma_k^p$. The distance between the vectors in the new matrix are taken as the distance between words.

This method poses a few challenges. First of all there are the parameters k and possibly p which will have to be set, requiring an extra training phase. A second issue is that the actual values in the matrix become very hard to interpret. Rather than representing one contextual feature, they are representative of a weighted combination of all features. A third issue is that it changes the matrix from one being very sparse to one being extremely dense. This could actually make it more time-consuming to calculate relatedness using a function like cosine similarity, as zero entries could be ignored otherwise.

I will not experiment with LSA in my thesis, but it is possible that it could be applied in future work. LSA can be applied on top of other weighting methods; for example in Turney et al. (2011) LSA is used on top of a PMI weighted matrix.

Central to LSA's effectiveness is its ability to merge many features into a single feature in an unsupervised fashion. There are a number of other techniques to accomplish this, though usually with the aid of a large human-built resource. Gabrilovich and Markovitch (2007, 2009) describe a method of merging multiple features together by representing them with the categories from *Wikipedia*. The method, called Explicit Semantic Analysis (ESA), represents a word as the articles in which it is found in *Wikipedia*. The tf.idf scores of each word in an article are used to give weight to the article. This method can easily be used to represent entire text in addition to individual words. The authors' findings were that ESA outperformed LSA on a word similarity and text similarity data set. An interesting property of this work is that since *Wikipedia* contains cords – links between multiple language versions – it is possible to use it to build a cross-language measure of semantic relatedness (Hassan and Mihalcea, 2009).

Another method called Salient Semantic Analysis (SSA) (Hassan and Mihalcea, 2011) works similar to ESA but uses the linking structure of *Wikipedia* to a greater extent. In Radinsky et al. (2011) an enhancement on ESA is proposed to take temporal factors into account. This Temporal Semantic Analysis (TSA) uses time series information to alter the weighting of a concept vector. Mohammad and Hirst (2006b) mapped word features to their corresponding categories in *Macquarie Thesaurus*. All these methods aim to create sets of features that are richer in information than simple term co-occurrences.

Ensemble Methods

Curran (2002, 2003) explored an ensemble method for determining semantic relatedness. The idea was to re-rank synonyms based on scores assigned to them using a variety of different MSRs. Six different systems for discovering synonyms were combined using three techniques:

- Arithmetic mean
- Harmonic mean

- Mixture: calculates mean score for each synonym and then re-ranks using that mean

It was found that these ensemble methods outperformed each individual method. Experiments were also done using three best performing measures, but it was found that including the weaker measures still improved the accuracy of their ensemble.

Hagiwara et al. (2005) used Machine Learning to construct an ensemble for identifying synonyms. A variety of distributional and pattern-based methods of extracting synonyms from a corpus were used as features for machine learning. It was found that by combining heterogeneous sources of synonymy the authors were able to improve over any one given method. Of interest here is the emphasis on heterogeneity of the sources of synonymy. This method is analogous to the concept of Stacking from Machine Learning where multiple classifiers are trained and their output used as features in another classifier. The hope of such a method is that this meta classifier is superior to any individual classifier. Although I do not explore stacking to any great extent it has been observed that often such methods simply learn which classifier is the best and then use that ones decisions for classification (Witten and Frank, 2005). Perhaps this is why heterogeneous information is so important to the method of Hagiwara et al. (2005). In Chapter 4 I will experiment with merging distributional and thesaurus-based sources of synonymy. Such sources of synonymy information are quite heterogeneous.

The method described in Hagiwara et al. (2005) is actually a kind of supervised MSR. In this case the supervision is used in mixing multiple measures of synonymy. I will also create a supervised MSR, but in my case the supervision will happen at a much earlier stage, when re-weighting a *term-context* matrix.

Other Systems

Semantic Vectors[7] is a system built for determining semantic relatedness between words (Widdows and Ferraro, 2008). It works by applying a method called Random Projection to term-document matrices. One of the drawbacks of this sort of method is that the context is an entire document instead of just a few nearby words.

SenseCluster[8] is a program that separates instances of words into distinct senses (Purandare and Pedersen, 2004). Although it is not intended as a system for determining

[7] http://semanticvectors.googlecode.com/

[8] http://www.d.umn.edu/~tpederse/senseclusters.html

semantic relatedness between any pair of words, it does perform this task as part of its overall goal of finding word senses.

As mentioned earlier, *Sketch Engine* (Rychlý and Kilgarriff, 2007) was used to generate thesauri in a variety of languages. However, *Sketch Engine* is not free for use and so would be difficult to incorporate into a resource that is intended to be distributed freely.

More recently some work has been done harnessing multiple knowledge sources to create a single measure of semantic relatedness (Zhang et al., 2011). The basic idea presented is that merging semantic relatedness measures based on data from Wikipedia and *WordNet* can make for an even more powerful measure.

This has been just a short overview of work done on extracting thesauri from corpora. Work in this area is not limited to English either, in addition to the multi-language *Sketch Engine* (Rychlý and Kilgarriff, 2007), work for specific languages including Chinese (Tseng, 2002) and Japanese (Takenobu et al., 1995) has been carried out. In (Joubarne and Inkpen, 2011) distributional MSRs are built for English, French and German and compared on translations of the Rubenstein and Goodenough (1965) word-pair data set. For the most part these methods work by identifying the contexts in which a particular word appears. Vectors of these contexts are then used to determine semantic distance between pairs of words. In Section 3.3 I will describe a few particular systems in much more detail.

A much more detailed discussion of vector space models use in representing semantics can be found in Turney and Pantel (2010). Three kinds of vector space models (*term-document*, *word-context* and *pair-pattern*) are discussed. Different implementations of these models as well as applications of them are explored, including their use in generating thesauri.

3.3.3 Hybrid approaches to MSRs

Recently research in this area has moved in the direction of combining resource-based and distributional MSRs. There are a number of ways in which this can be accomplished. Discussed earlier, Weeds and Weir (2005) used external resources as a source of training data for a similarity function in a distributional method. Although they were only adjusting two parameters, it is an interesting starting point.

In Patwardhan et al. (2003) the Lesk algorithm for *WordNet* was enhanced using distributional techniques. In this method a distributional vector is created for each word in a definition in a *WordNet* gloss. The vectors for each word in a definition are combined into a single vector and then cosine similarity is used to determine relatedness between pairs of words. Effectively a word's definition is used as a source of related words from which a distributional co-occurrence vector is created. In this case the corpus used to find co-occurrence information was the actual set of *WordNet* glosses. This experiment thus does not directly make use of information from outside *WordNet*, but any corpus could be substituted, and it still uses distributional information to aid an otherwise resource-based measure of semantic relatedness. An extension of this methodology is discussed in (Liu et al., 2012) where a corpus of biomedical papers are used instead of glosses. The Unified Medical Language System (UMLS) (Bodenreider, 2004) and *WordNet* were used to provide definitions for each concept.

Another hybrid approach is that of Mohammad and Hirst (2006b,a) where, this time, information from a resource was used to enhance an otherwise distributional approach. In their method a word is represented by a vector of around 1000 features. Each feature corresponds to a category in *Macquarie Thesaurus* (similar to *Roget's Thesaurus*). A word is first represented by a vector of its neighbouring terms. These neighbouring terms are then mapped to the categories to which they belong, in *Macquarie Thesaurus*. This representation can then be enhanced by bootstrapping to disambiguate word senses, which can be repeated over many passes.

I will try something similar in my work by using both thesauri and distributional methods. I will try to use a thesaurus not to reduce dimensionality of a matrix, but rather to determine more appropriate feature weights.

3.3.4 Other Methods of Extracting Information From Text

Although not the same as updating an existing thesaurus, there is some relevant work on building thesauri from scratch. Several methods of measuring semantic relatedness from Section 3.3 were tested on generating thesauri automatically, and so will not be covered in this sub-section.

In Caraballo (1999) a method of automatically generating taxonomies is proposed. In this method noun contexts in a corpus are used to determine related nouns. Repeatedly, the closest two nouns are placed together in a node until a graph of nodes is formed. Hypernyms are extracted from text using the patterns proposed in Hearst (1992) and are used to determine which of the nodes in the graph are hypernyms of other nodes.

Some similar and more recent work on taxonomy construction can be found in (Kozareva and Hovy, 2010). A semi-supervised method of extracting hypernym/hyponym relationships from the Web was used to construct a hierarchy, which was evaluated against *WordNet*'s hypernym hierarchy. Given a starting concept, their algorithm harvests hypernyms, filters out noise and then organizes these hypernyms into a taxonomy. Hypernyms are mined from the Web using pattern-based bootstrapping approaches from (Kozareva et al., 2008) and (Hovy et al., 2009). This results in a directed graph of hypernym-related words. This graph is then turned into a tree, in part by eliminating cycles and redundant links. Some similar work on ontologizing semantically related word pairs can be found in (Pantel and Pennacchiotti, 2008).

Yamada et al. (2009) used distributional similarity to enhance a hypernym hierarchy that was originally constructed from Wikipedia. The technique, used to extract the Wikipedia hypernym hierarchy, is the same as that described by Sumida et al. (2008) for finding Japanese hypernyms. Hypernym candidates are extracted from the hierarchical layout of Wikipedia. An SVM – trained on features including POS tags and the appearance of morphemes – confirms these hypernyms. Two techniques based on identifying words that appear in similar contexts are tested for expanding this hypernym hierarchy. For each candidate word x its k most similar words $w_1 \dots w_k$ in the hypernym hierarchy are found. The weight of the similarity between x and w_i is used to vote for w_i's ancestors. There is a penalty for assigning weights too far up in the hierarchy. The hypernym with the highest weight is selected as the hypernym of x.

Wikipedia has been used in a variety of research related to building ontologies. In (Wu and Weld, 2008) a system called Kylin Ontology Generator (KOG) for building ontologies from Wikipedia info-boxes is discussed. One of the features of this system is the way in which it cleans up the info-boxes. It attempts to recognize duplicate entries, ignore rare cases, assign meaningful names and infer attribute types. This line of work is more similar to Cassidy (2000) in that they are taking a resource intended for an entirely different purpose and attempting to make an Ontology out of it. That said, it is not actually enhancing a lexical resource with new terms or relationships.

There is much research on mining Wikipedia for taxonomies (Ponzetto and Strube, 2007; Kassner et al., 2008) or semantic relationships (Sumida et al., 2008; Chernov et al., 2006). The structure of the resources mined from Wikipedia tend to be very different from that of *Roget's Thesaurus*, so does not influence my methodology for updating *Roget's* very much.

Some recent work has focused on measuring semantic relatedness between words using image data (Leong and Mihalcea, 2011a), and even measuring semantic relatedness between words and images (Leong and Mihalcea, 2011b). *ImageNet*[9] is a resource matching images to *WordNet* synsets. Features were extracted from these images and used to enhance existing semantic relatedness measures in (Leong and Mihalcea, 2011a). The work was expanded on in (Leong and Mihalcea, 2011b) to consider the relatedness between an image and a word. Words were represented by vectors of nouns found in the synsets gloss as well as visual codewords extracted from the images associated with that synset. Images were represented only as the codewords. *tf.idf* weighting was applied and cosine similarity was used to measure relatedness between words and images. This is a very interesting task, as finding the relatedness between items of two very different domains has not been widely tackled. In my future work in Chapter 4 I discuss proposed future work to measure relatedness between words in two different languages. Work measuring relatedness between images and words could act as a source of inspiration for some techniques to solve cross-language semantic relatedness.

3.3.5 Supervised Document Relatedness

Apart from Hagiwara et al. (2005) and Weeds and Weir (2005), not too much work has been conducted on introducing supervision to MSR between words, but there is still some work worth examining from similar areas of research. For example in Yih (2009) and Hajishirzi et al. (2010) a method of learning weights for short document similarity is proposed. In this case the weighting was done on a *term-document* matrix rather than a *term-concept* matrix. The documents in this case were short queries and the goal was to find the most similar queries. Using a set of known related queries, a loss function was learned that could re-weight the matrix to maximize similarity between known related queries and minimize similarity between unrelated queries. This method was found to perform better than TF.IDF at this task. I will use a similar idea, though instead of a loss function I will attempt to use measures of association to re-weight a matrix using known synonym/non-synonym pairs.

3.3.6 Composition and Text Representation

Although it is beyond the scope of this thesis, in this section I will remark on a recent work on composition. Representing the meaning of short texts using distributional representations has been a growing trend in NLP. One of the most straightforward means

[9] http://www.image-net.org/

of representing two or more words is to average their distributional vectors. In (Razavi et al., 2009), averaged co-occurrence vectors produced feature sets to represent short texts for the classification of dream descriptions and classification of medical abstracts. The co-occurrence representations are shown to outperform Bag-Of-Words representation of the same short documents.

Simply merging the distributional representation of two words does not take word order into account, and so is not really composition. Fortunately a number of methods of modelling composition have been attempted. Mitchell and Lapata (2010) compare and contrast a variety of models for composition. They also produced a *Rubenstein&Goodenough*-style data set for measuring similarity of compositions. However, their most successful model did not take word order into account. Baroni and Zamparelli (2010) used matrices to represent adjectives, while vectors represent nouns. When composing an adjective-noun pair, the noun vector is multiplied by the adjective matrix. This method is only tested on adjective-noun pairs. Turney (2012) discusses a dual model that applies relational similarity to the problem of composition. Rather than representing two words with a single vector, a composition is matched with other similar compositions.

3.4 Conclusion

The task of automatically enhancing a thesaurus is well founded. Some work has been done relatively successfully to enhance *WordNet* either with new terms (Snow et al., 2006; Zheng et al., 2008) or with added information such as sentiment and objectivity (Esuli and Sebastiani, 2006).

There are a variety of methods for measuring semantic relatedness using corpus statistics. Many of the most successful ones have been implemented in the *SuperMatrix* package. Since this package has already been used effectively on the Polish *WordNet*, I initially planned to conduct my experiments using this system. However as my work progressed I found that I needed to re-implement much of the functionality of this system to a point where I had implemented my own code based on a similar design to *SuperMatrix*. My system takes matrices formatted similar to those used in *SuperMatrix*. Employing a system similar to *SuperMatrix* for the purposes of enhancing *Roget's* will be the focus of Chapters 4 & 5 in this thesis proposal.

I have also described a variety of applications that make use of *Roget's*. Many of these applications can be applied to evaluating *Roget's* before and after the addition of new words. This will be the focus of Chapter 6 where many of the applications from this chapter as well as several new ones will serve as evaluation for the enhanced *Roget's Thesaurus*.

Chapter 4

Measuring Semantic Relatedness

In this chapter I describe how I use *Roget's Thesaurus* to enhance distributional measures of semantic relatedness (MSRs). I use sets of known related words in *Roget's Thesaurus* in order to learn a method of reweighting a *word-context* matrix.

Many MSRs use the context where a word appears to determine its meaning. Words which frequently appear in similar contexts are assumed to have similar meanings. Such MSRs usually re-weight contexts based on some measure of their importance, usually the association between the context and a term it appears with. One of the most successful of these measures is Pointwise Mutual Information (PMI). It increases the weight of contexts where a word appears regularly but other words do not, and decreases the weight of contexts where many words may appear. Essentially, it is unsupervised feature weighting. I present a method of supervised feature weighting. The method identifies contexts shared by pairs of words known to be semantically related or unrelated, and then uses a measure of association to weight these contexts by how often they contain closely related words. The method is very general and can use any thesaurus as a source of known synonym pairs and can be used with many measures of association, other than PMI. I will use the 1911 and 1987 *Roget's Thesaurus* and *WordNet* 3.0 as a sources of training data and will use the 1987 *Roget's Thesaurus* as evaluation data.

I compare this supervised weighting method with unsupervised methods and ultimately I combine supervised and unsupervised methods in order to create the best possible method. The paper (Kennedy and Szpakowicz, 2011) describes this methodology, though it is specific to using PMI as a measure of association and uses *SuperMatrix* (Broda and Piasecki, 2008). The method I present in this thesis is more general in terms of the association measures and the types of training data it can use. It has been accepted for publication in (Kennedy and Szpakowicz, 2012b).

The choice of the terms “supervised”, “unsupervised” and “combined” to name my systems may seem a bit strange at first, so it merits an explanation. The unsupervised and supervised systems learn slightly different things. The former learns weights for each word-context pair. The latter learns weights for each context, which is applied equally to every word-context pair with that context. They do not attemp to assign classes/clusters, in the way supervised and unsupervised machine learning systems do. In fact, the combined system, which makes use of both supervised and unsupervised learning might be more analogous to supervised machine learning. When I evaluate the “supervised” system, I am evaluating the component that makes use of training data in the “combined” system.

4.1 Goals of my Measure of Semantic Relatedness

My goal is to create a MSR which can be useful in adding new words to *Roget’s Thesaurus*, so I will make a few assumptions about how the *Thesaurus* should be modified. These will help to explain the choices of evaluation later in this chapter.

Although the number of terms, Paragraphs and Semicolon Groups changed quite a bit between the 1911 and 1987 Thesauri, the number of Heads remains approximately the same. In fact there are slightly more Heads in the 1911 version than in the 1987 version. For example the Head “Complexity” appears in the subsection for “Order” in the 1911 *Thesaurus*, but not the 1987 version. There are only a few such differences; just about every head in the 1987 version can also be found in the 1911 version. There are 8 different POSs represented in the 1911 *Roget’s Thesaurus*, but I plan to focus only on nouns, verbs and adjectives. Initially I had hoped to also add new adverbs but, as it will be seen in this chapter, this proved more difficult. Other parts parts-of-speech – labeled as phrases, interjections, prefixes and pronouns – appear in the 1911 *Thesaurus*, while only interjections are found in the 1987 *Thesaurus*. These other parts-of-speech tend to contain multi-word entries (interjections and phrases) or have small cardinality (prefixes and pronouns) and so can more easily be updated manually. For these reasons I do not intend to add to *Roget’s* any new categories other than Semicolon Groups and Paragraphs for nouns, verbs and adjectives.

I will aim to add new words to the *Thesaurus* in three different places:

- new word in an existing SG

- new word in a new SG in an existing Paragraph
- new word in a new SG in a new Paragraph

Based on these assumptions, evaluation of a new semantic distance measure should be useful at identifying words in the same POS, Paragraph and Semicolon Group.

There is one other thing to consider. In *Roget's* there are many cross-references between groupings. Ideally when adding new terms my system should consider whether or not to add cross-references between points in the *Thesaurus*. This may be difficult because it should require fairly accurate word sense disambiguation. However, since I do not make use of these cross-references in any of my *Roget's*-based applications – see Chapter 6 – I do not find this pressing. I consider adding cross-references to be outside the scope of this thesis and so it will be left for future work.

4.2 Building a Word-Context Matrix for Semantic Relatedness

Before I implement any MSR, I need to build a *word-context* matrix. I used Wikipedia[1] as a source of data and parsed it with *Minipar* (Lin, 1998a). Wikipedia is a good choice for a corpus because it is fairly well written and contains a current lexicon. It is continuously being updated, so new words like "iPhone", "google" or "reaganomics" can be found. This makes Wikipedia a very suitable corpus for this kind of work.

The choice of dependency triples instead of all neighbouring words allows only the use of contexts that most directly affect the meaning of the word. For example, if an adjective and two nouns appear within the same window, it may be beneficial to know which of the two nouns is actually modified by that adjective, rather than using the adjective as a context for both. In English this may be easy to approximate without a parser because sentences tend to follow a subject-verb-object pattern, but in freer word-order languages this could be even more beneficial. Additionally the parser provides information on the syntactic relations. Using dependency triples helps identify the most important contexts and explains how those contexts are related. Approximately 900 million dependency triples are generated by parsing Wikipedia, taking up approximately 20 GB. An example of these dependency triples can be seen in Figure 4.1. The problem I describe here is how to build a reliable *word-context* matrix from these triples.

[1] I used a dump of Wikipedia from August 2010.

```
fin C:i:V settle
settle V:s:N ignorance
settle V:mod-before:A never
settle V:subj:N ignorance
settle V:obj:N question
question N:det:Det a
```

Figure 4.1: Example of dependency triples taken from the quote "Ignorance never settles a question." by Benjamin Disraeli parsed in *Minipar*

Before describing how the cutoff is selected, I will outline how the matrices are built. Three matrices are built, one each for nouns, verbs and adjectives/adverbs. I used *Minipar* to create dependency triples $\langle w_1, r, w_2\rangle$ and then for each triple I generate two word-context pairs $(w_1, \langle r, w_2\rangle)$ and $(w_2, \langle w_1, r\rangle)$. When the word w_1 and w_2 are used as part of a context, they can be of any part-of-speech and all relationships r are considered. The direction of r is also retained. When w_1 and w_2 are the terms, they must be either noun, verb, adjective or adverb. In addition, they must be single words with no upper case letters, numbers or symbols. Generally only proper nouns are left in upper case by *Minipar*. With these constraints I was able to use 100% of Wikipedia when building the matrix for verbs and adjectives/adverbs, while only 50% was used for nouns. This limit was chosen both because it was the most data that could be held in a system with 4GB of RAM and because the left over data could be used in later evaluation.

Examples of triples are $\langle \mathit{time}, \mathit{mod}, \mathit{unlimited}\rangle$ and $\langle \mathit{time}, \mathit{conj}, \mathit{motion}\rangle$, where the word "time" appears in the context with the modifier "unlimited" and in a conjunction with "motion". The *word-context* matrix is constructed from these dependency triples. Each row corresponds to a word w, each column – to one of the contexts, c. That cell of the matrix records $\mathit{count}(w, c)$: how many times w is found in c. As my system learns either supervised or unsupervised weights, it changes the values in this matrix from straight counts to more appropriate weights. Each row in this matrix is essentially a vector representing a word. The distance between two words is the distance between their vectors.

In *Minipar* adjectives and adverbs are labeled with the same part-of-speech: "A". That is why I build three matrices for nouns, verbs and adjectives/adverbs. All words are in their lemmatized forms, giving us a total of 359,380 nouns, 9,294 verbs and 104,074 adjectives. The number of verbs appears to be extremely low, though if one looks at the number of verbs available in the index of *WordNet* 3.0 they will find that there are only around 11 thousand, a count that includes numerous phrases as well as single words.

4.2.1 Picking an Appropriate Cutoff

Both words and contexts that appear too infrequently tend to be unreliable. They are noisy and often the result of spelling errors. I can set thresholds for how many times a term and a context must be found to be considered reliable. One problem is, if I set that threshold too high I will lose a lot of words that could otherwise be placed into *Roget's Thesaurus*. For each of the three matrices constructed, I calculate a score for each cutoff value $c = 1..100$. For each cutoff I randomly select 100 words $x_1..x_{100}$, that appear c times in the matrix and find the 1000 most closely related words $y_1..y_{1000}$, each of which must also appear c times. From this ranked list of words, $y_1..y_{1000}$, I measure how many steps through the list are needed in order to find the first word y_j semantically related to x_i. That is, I iterate through the list of 1000 nearest neighbours $(y_1..y_{1000})$ to x_i until I find a word in the same *Roget's* Paragraph as x_i and record its rank. This score is averaged across all 100 random samples and then normalized by 1000 to give a score between 0 and 1. In some cases the list of randomly selected words $x_1..x_{100}$, may contain duplicates, particularly in the case of verbs where there are relatively few unique terms. Despite this there is a very obvious trend showing that the higher the cutoff c is, the sooner a known related word – in the same *Roget's* Paragraph – will be found in the list $y_1..y_{1000}$.

I graph both the above average rank score as well as a score for recall. The recall score is the percentage of words that appear in the matrix c or more times. I graph the results for nouns, verbs and adjective/adverbs in Figures 4.2, 4.3 and 4.4 respectively.

It appears that the scores for the average distance more or less mirror the recall measure. Another observation is that the curve for verbs is closer to a straight line. This happens because although there are less than 10 thousand verbs in my matrix, each verb tends to appear very frequently. Any cutoff I select will be somewhat arbitrary, so I selected values where the distance scores appear to plateau. In the case of nouns and adjectives I selected 35 as the cutoff, while for verbs I selected 10.

These cutoffs are for the number of times a word must be found, but it does not take into account how many times a context must appear. There are 2,463,001 contexts for nouns, 2,892,002 for verbs and 817,921 for adjectives/adverbs. Approximately 50% of these contexts appear only once. A context that appears only once is of limited use to any MSR since no pair of words could possibly share that context. They also tend to

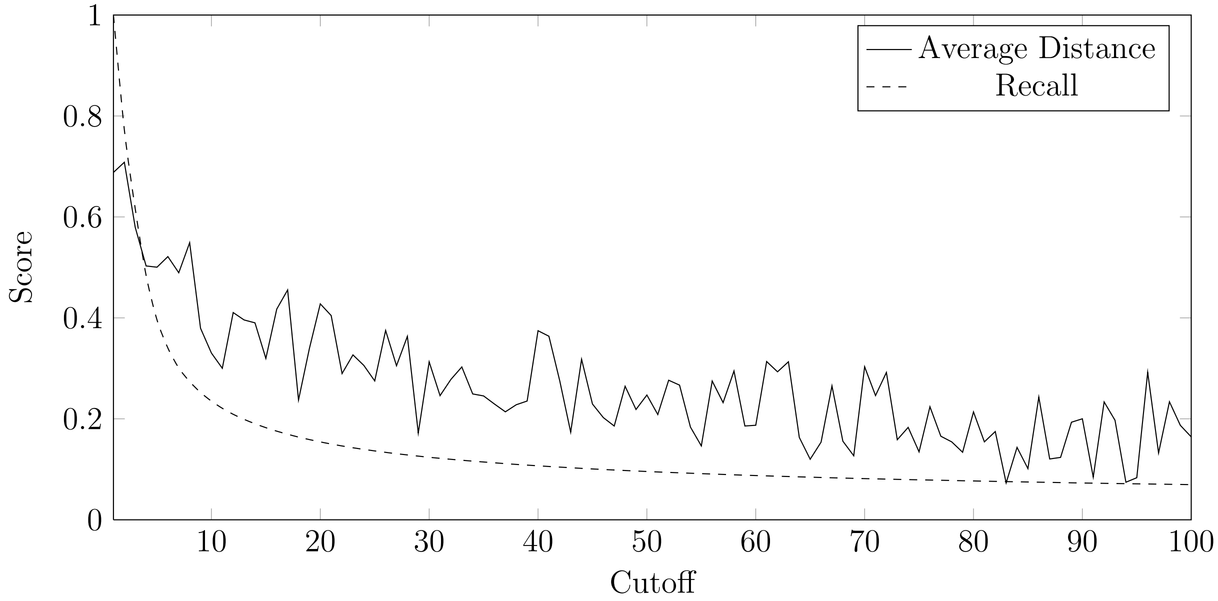

Figure 4.2: Average Distance and Recall of Nouns.

be noisy and anomalous, so I only included contexts that appear 2 or more times in the matrix. It may be possible to find a more suitable cutoff through experimentation, but I will leave that for future work as the cutoffs I've chosen appear to provide good results for my experiments.

4.2.2 The Final Matrices

Using these cutoffs, I created the three matrices. I report in Table 4.1 counts of the number of words and contexts in each of these matrices before and after the cutoff. To provide some context to these numbers I also include the number of entries in *WordNet* 3.0 for each part-of-speech, where those entries are single words – not phrases – with no numbers, symbols or upper-case letters – the same criteria I had for selecting words in the matrix. I also report the number of non-zero entries.

Table 4.1 shows that while the cutoffs I have chosen significantly reduce the number of words and contexts it does not ultimately reduce the number of non-zero entries in the matrix by a very large percentage. For example only 12% of nouns and 43% of noun contexts are kept, but the matrix retains 91% of the non-zero entries, while the reduction in size of the other two matrices are even less severe. I am left with much more dense

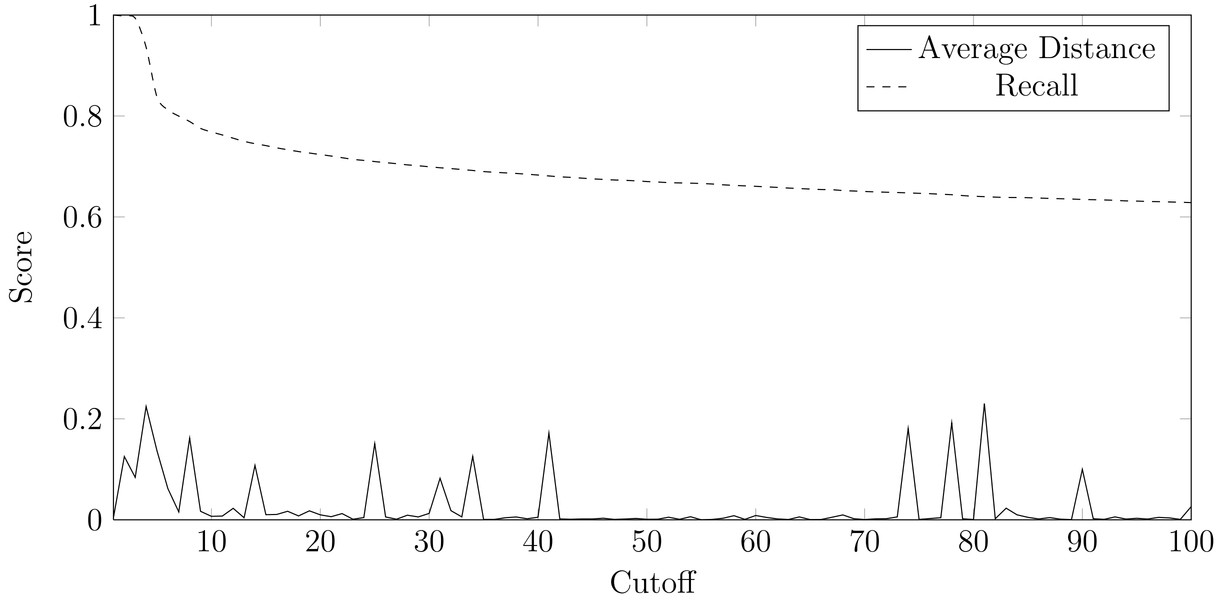

Figure 4.3: Average Distance and Recall of Verbs.

matrices which are richer in information.

Also in Table 4.1 I show counts of the number of words in each matrix and the number of words, by part-of-speech in *WordNet* 3.0. While my matrices contain fewer entries than *WordNet* 3.0, they are fairly close in terms of the count of words available. *WordNet* 3.0 may contain many infrequent or rare words as well as some that may not frequently be found in Wikipedia. Likewise, there are many words that one can find in these matrices that are not present in *WordNet* or *Roget's Thesaurus*.

4.3 Measures of Association

To measure semantic relatedness I implemented a variety of measures of association. These same measures are applied for both the supervised and unsupervised weighting. I used measures as defined in (Evert, 2004) and borrow much of his notation.

Measures of association measure the dependence between two random variables, or between two values of two random variables. Essentially I will use them to identify cases where two things are observed occurring together more frequently than they would be expected to. These measure of association are key to how I will re-weight the *word-context* matrix.

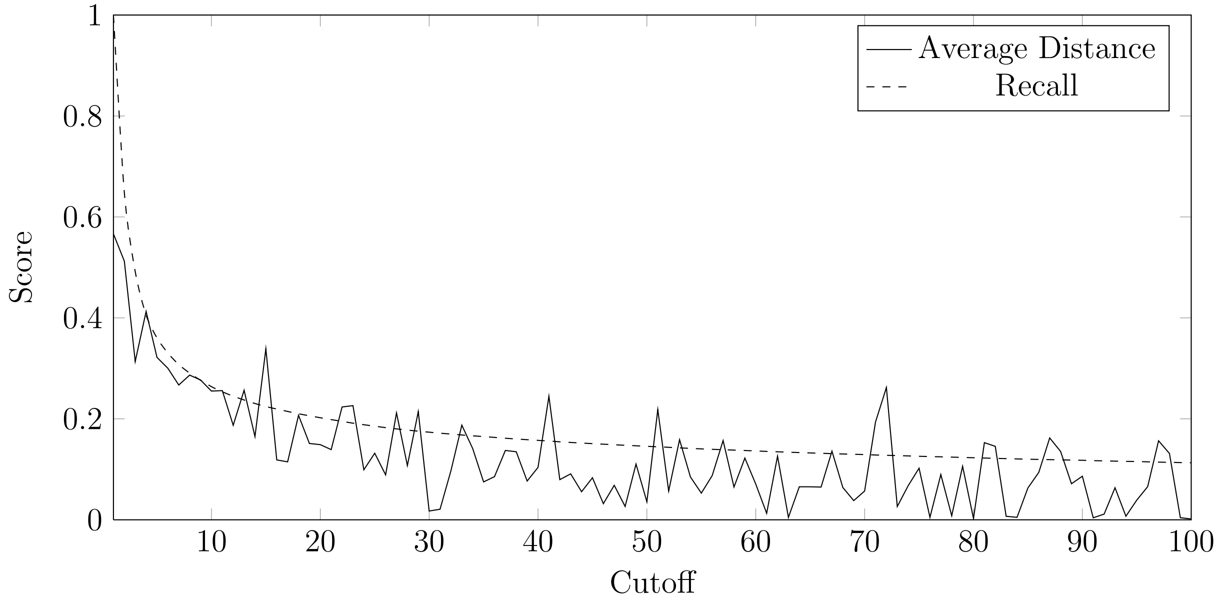

Figure 4.4: Average Distance and Recall of Adjectives and Adverbs.

$$
\begin{array}{llll}
 & y \in Y & y \notin Y & \\
x \in X & O_{0,0} & O_{0,1} & = R_0 \\
x \notin X & O_{1,0} & O_{1,1} & = R_1 \\
 & = C_0 & = C_1 &
\end{array}
$$

Figure 4.5: Observed Confusion Matrix.

4.3.1 Calculating Observed and Expected Values

I describe here a general method of creating confusion matrices for observed and expected values. This process and the functions applied here are used in both the supervised and unsupervised methods and so are described very generally. I count the number of co-occurring events in two discrete random variables, X and Y. I count the events $x \in X$, $x \notin X$, $y \in Y$ and $y \notin Y$. This makes up my observed confusion matrix shown in Figure 4.5. The matrix in Figure 4.5 is a 2×2 matrix, but any size of confusion matrix can be used. This process is described in Evert (2004).

From this matrix of observed counts I calculate values for the rows (equation 4.1), columns (equation 4.2) and total size of the matrix (equation 4.3). These are then used

POS	Matrix	Words	Contexts	Non-zero Entries	% non-zero
Noun (35 cutoff)	Full	359 380	2 463 001	30 994 968	0.0035%
	Cutoff	43 834	1 050 178	28 296 890	0.0615%
	(% of Full)	*(12.2%)*	*(42.6%)*	*(91.3%)*	
	WordNet	55 191	-	-	-
Verb (10 cutoff)	Full	9 294	2 892 002	26 716 709	0.0994%
	Cutoff	7141	1 423 665	25 239 485	0.2483%
	(% of Full)	*(76.8%)*	*(49.3%)*	*(94.5%)*	
	WordNet	8 429	-	-	-
Adj/Adv (35 cutoff)	Full	104 074	817 921	9 116 741	0.0107%
	Cutoff	17 160	360 436	8 379 637	0.1355%
	(% of Full)	*(16.5%)*	*(44.1%)*	*(91.9%)*	
	WordNet	21 504	-	-	-

Table 4.1: Counts of the number of rows, columns and non-zero entries for each matrix.

to calculate an expected value (equation 4.4) corresponding to every observed value in the matrix from Figure 4.5.

$$R_i = \sum_j O_{i,j} \tag{4.1}$$

$$C_j = \sum_i O_{i,j} \tag{4.2}$$

$$N = \sum_{i,j} O_{i,j} \tag{4.3}$$

$$E_{i,j} = \frac{R_i C_j}{N} \tag{4.4}$$

For observed value, $O_{i,j}$, in the confusion matrix (Figure 4.5) a corresponding expected value, $E_{i,j}$, is calculated. The expected value $E_{i,j}$ is calculated using values for the row R_i, the column C_j and a value for the whole matrix N. Expected values can also be represented in a confusion matrix – see Figure 4.6. These observed and expected values are then used to measure association between the random variables X and Y.

$$\begin{array}{cc} & \begin{array}{cc} y \in Y & y \notin Y \end{array} \\ \begin{array}{c} x \in X \\ x \notin X \end{array} & \begin{bmatrix} E_{0,0} & E_{0,1} \\ E_{1,0} & E_{1,1} \end{bmatrix} \end{array}$$

Figure 4.6: Expected Confusion Matrix.

4.3.2 Measures of Association

I experiment with six measures of association. These measures are Dice (equation 4.5), Pointwise Mutual Information (PMI) (equation 4.6), Z-score (equation 4.7), T-score (equation 4.8), χ^2 (equation 4.9) and Log Likelihood (equation 4.10).

$$Dice = 2\frac{O_{0,0}}{R_0 + C_0} \tag{4.5}$$

$$PMI = \log \frac{O_{0,0}}{E_{0,0}} \tag{4.6}$$

$$\textit{Z-score} = \frac{O_{0,0} - E_{0,0}}{\sqrt{E_{0,0}}} \tag{4.7}$$

$$\textit{T-score} = \frac{O_{0,0} - E_{0,0}}{\sqrt{O_{0,0}}} \tag{4.8}$$

$$\chi^2 = \sum_{i,j} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}} \tag{4.9}$$

$$\textit{Log-Likelihood} = 2 \sum_{i,j} O_{i,j} \log \frac{O_{i,j}}{E_{i,j}} \tag{4.10}$$

These measures can be broken down into three broad groups. Log Likelihood and χ^2 are measures that take into account all the observed and expected values derived from Figure 4.5. PMI, T-score and Z-score only take into account a single observed and expected value, $O_{0,0}$ and $E_{0,0}$ respectively, and so are measure the association for a particular value rather than for an entire random variable. Dice measures the overlap between two vectors as their harmonic mean and so never actually requires the calculation of any expected values.

4.4 Three classes of MSRs

Using these measures of association I can determine how dependent two events are. One could measure the association between a word and a context in which it appears, or the association between a context and a set of synonyms from some resource – in my case *Roget's* or *WordNet*. Measuring association and re-weighting a *word-context* matrix is really the first of two steps in measuring semantic relatedness. The second part is to find the relatedness score between two words in the *word-context* matrix. There are many established methods, including Jaccard (Jaccard, 1901), Dice (Dice, 1945) and the asymmetric measure of Weeds and Weir (2005). To measure relatedness I use cosine similarity (Equation 4.11).

$$cos(A,B) = \frac{A \bullet B}{\|A\|\|B\|} \tag{4.11}$$

I will only use cosine similarity to keep my evaluation consistent. Also, as I am measuring distance between word vectors, cosine similarity makes the most sense. I will leave any experimentation with other methods for future work.

4.4.1 Unsupervised Learning of Context Weights

Commonly when measuring semantic relatedness between pairs of words a measure of association is used to find the dependency between a word and a context. In the case of Figure 4.5 $x \in X$ is the appearance of a word w_i and $x \notin X$ is the non-appearance of that word, while $y \in Y$ is the appearance of a context c_j and $y \notin Y$ is the non-appearance of that.

- $O_{0,0}$ $[x \in X \wedge y \in Y]$: w_i is found in context c_j;
- $O_{1,0}$ $[x \notin X \wedge y \in Y]$: w_i is found in a context other than c_j;
- $O_{0,1}$ $[x \in X \wedge y \notin Y]$: a word other than w_i is found in context c_j;
- $O_{1,1}$ $[x \notin X \wedge y \notin Y]$: a word other than w_i is found in a context other than c_j.

From this, any of the described methods of association in Section 4.3.2 can be applied, though PMI appears to be the most common (Pantel, 2003; Turney and Pantel, 2010).

4.4.2 Supervised Learning of Context Weights

The supervised approach I describe here measures associations not between words and contexts but between pairs of words co-occurring in a context and pairs of words known to be synonyms in *Roget's Thesaurus* or *WordNet*. I calculate an association score for every context c_k. From Figure 4.5, $x \in X$ is the count of pairs of words that co-occur in context c, while $x \notin X$ is the count of word pairs where one word appears in c and the other does not. $y \in Y$ is the count of word pairs that are found to be near-synonyms, while $y \notin Y$ is the count of pairs of words that are not near-synonyms. To calculate the association, I count the following pairs of words w_i, w_j for each context c_k:

- $O_{0,0}$ $[x \in X \wedge y \in Y]$: $\langle w_i, w_j \rangle$ are synonyms and both appear in c_k;
- $O_{1,0}$ $[x \in X \wedge y \notin Y]$: $\langle w_i, w_j \rangle$ are not synonyms and both appear in c_k;
- $O_{0,1}$ $[x \notin X \wedge y \in Y]$: $\langle w_i, w_j \rangle$ are synonyms and only one appears in c_k;
- $O_{1,1}$ $[x \notin X \wedge y \notin Y]$: $\langle w_i, w_j \rangle$ are not synonyms and only one appears in c_k.

When taking these counts the number of times that a pair of words appear in c_k is also considered. The product of the counts of word pairs is used to give a score to each word pair, so that pairs of more frequent words will have a higher score. For example, if w_i and w_j appear 6 and 3 times respectively then this accounts for 18 pairs. I did this to account for cases where there are many infrequent and unrelated words but only a few closely related words with high frequency in c_k. This should still give those contexts a fairly high weight. A score $score(c_k)$ can be calculated for every context c_k, where the score is one of the measures of association.

Supervision of this sort could be done with either the 1911 or 1987 *Roget's Thesaurus*. In fact any list of synonyms would be sufficient for training, including *WordNet*. In theory the unsupervised method is more language-independent as it only requires a parser, or some method of finding contexts in a given language, while this supervised method requires a list of synonyms. That said, with the rise of non-English *WordNets*, I doubt this would be a major hindrance for any widely spoken language.

Calculating a score for all contexts is not trouble-free. For one, not all contexts will appear in the training data. To avoid this, I normalize every $score(c_k)$ so that the average of all the scores $score(c_1)..score(c_n)$ is 1; next, I assume that any unseen contexts also have a weight of 1; finally, I multiply the count of context c_k by $score(c_k)$ for every word in which c_k appears. Another problem is that some measure of association may give a negative score when the two events are less likely to occur together than by chance. In such situations I set $score(c_k)$ to zero. Another problem is that often the supervised matrix re-weighting is calculated with a fairly small number of true positives, so it may be difficult to get a very reliable score. The unsupervised matrix weighting, on the other hand, will use the distributions of a word and context across the whole matrix, so often will have more data to work with. It may, then, be optimistic to think that supervised matrix re-weighting will on its own outperform unsupervised matrix re-weighting. The more interesting experiments will be to see the effects of combining supervised and unsupervised weighting.

There are two variations on the supervised method that I explore. The first is to find a unique weight to be applied to every context $\langle r, w \rangle$. The second is to find a unique weight for every relationship r and then apply that weight to all contexts $\langle r, w_i \rangle .. \langle r, w_j \rangle$. To do this I use the same method as described above, only I combine the counts for contexts that share a common r. In this experiment, rather than learning contexts that are most appropriate for measuring semantic relatedness, I am learning which syntactic relationships best indicate semantic relatedness. The hypothesis behind this method is that the kind of relationship says more about a context than the word it is related to. Also as some of the contexts are very infrequent, appearing only twice in the *word-context* matrix, their scores are not always reliable. I will refer to these two methods methods as "relation" learning and "context" learning. In Section 4.5 my results will show which method works better.

4.4.3 Combined Supervised-Unsupervised Context Weights

Note that in the unsupervised learning system a measure of association is applied to every word-context pair and so a different weight is calculated for every spot in the *word-context* matrix. In comparison the supervised method finds association scores for every context. These are two very different scores and so it is natural that they may be complementary to each other. I attempt to combine these two methods by first weighting the matrix with the supervised score followed by using the unsupervised weighting method.

There are a variety of possible ways in which these two systems could be combined. I run the unsupervised method on top of the supervised methods, but the opposite is also possible. Likewise it could be possible to run the two independently on the unweighted matrix then apply the weights learned from the supervised method onto the unsupervised one. Another option would be to use some sort of voting system between the two. My intuition behind choosing this means of combining systems is that my proposed supervised method could be considered as a method of feature selection. Some contexts that rarely or never contain synonyms will have scores of 0 – or some extremely low score – and this will help the unsupervised system to identify good word-context pairs. Also, you the reader will see in Section 4.5 that the unsupervised method of context weighting is better than the supervised approach and so using the supervised approach as a pre-processing step seems sensible.

4.5 Evaluation

In this Section I will evaluate the unsupervised, supervised and combined methods of weighting a *word-context* matrix. I use 6 different measures of association that can be applied in an unsupervised system and in two supervised systems. There is also a baseline where there is no weighting at all applied. This gives me 19 differently weighted matrices to experiment with for each part-of-speech. I generate these matrices for nouns, verbs and adjectives/adverbs. Each of these configurations gives me a different MSR.

Once I have found the best supervised and unsupervised weightings, I try to combine them in the hope of creating an even more powerful weighting method. To select the best supervised and unsupervised weighting methods I have a tuning set to evaluate these systems. I use a separate testing set to compare the combined systems against the selected best supervised and unsupervised systems.

4.5.1 The Evaluation Dataset

I evaluate the measures of semantic relatedness by using them to generate lists of closely related words. As described in Section 4.1, I need to have MSRs which can identify words in the same POS, Paragraph and SG. These are three different evaluation criteria for my MSRs. For nouns I will evaluate the weighted matrices of sets of 1000 words from *Roget's Thesaurus*, while I use sets of 600 words for verbs and adjectives.

There are several steps I take in order to generate a good set that is representative of the *Thesaurus* as a whole. Since often Semicolon Groups are extremely small, I only selected words that appear in Semicolon Groups with 4 or more single words that are found in the noun *word-context* matrix. I did this to ensure that there would be a sufficient number of words to have a good evaluation on the SG. For each Paragraph, up to three words that meet these criteria are randomly selected – in some cases there may be no words that meet this criteria – then from this set of candidates 3000 words are randomly selected, which are divided into three sets of 1000. A word cannot be selected from a Paragraph if it was previously selected from another Paragraphs, thus giving me 1000 unique words. The idea is to make sure that I have as many words as possible from as many Paragraphs as possible, thus covering as many topics as possible.

For all evaluation I used the 1987 *Roget's Thesaurus*, since it is larger and newer. For each of the 1000 words I generate a long list of the nearest related words – up to 1000 – and perform evaluation at different levels of recall. The levels of recall are the top 1, 5, 10, 20, 50 and 100 closest words. Only words that are found in the 1987 *Roget's* are included in this evaluation. I include words in phrases for this part of the evaluation. The many words that are found in the *word-context* matrix but not present in the *Thesaurus* cannot be evaluated on.

When using this corpus on my supervised system, the evaluation words are not included in the training process. Essentially I treat the evaluation words as if they were not present in *Roget's* at all and then measure the MSRs ability to find potential neighbours for them.

The same process was used to create a set of words for testing verbs and adjectives, though in these cases there were not enough Paragraphs meeting the set criteria to create a dataset of 1000. Instead sets of 600 each were constructed. For adverbs the coverage is quite poor in *Roget's* and there were no Paragraphs that met the set criteria. For this reason I will not attempt to add adverbs to *Roget's* and it will be left for future work.

Three datasets are created for each of nouns, verbs and adjectives. One of these sets will be for tuning my system. As I will experiment with combining the best supervised and unsupervised systems it is preferable to have separate tuning and testing sets as selecting the best systems is in fact a tuning stage. The second dataset will be a testing set. The third dataset will be left for further experiments for adding new words to *Roget's Thesaurus* – see Chapter 5.

4.5.2 The Training Data

The training data itself consists of sets of known synonyms that appear in the matrix. These lists of synonyms can come from both versions of *Roget's* or from *WordNet*. In theory, these lists could come from any source. They need not be lists of synonyms but any sort of symmetric relationship that one wants to learn. Also, they could be of any language for which a training set can be constructed, though I will do all my work with

POS	Count	*Roget's* 1911	*Roget's* 1987	*WordNet* 3.0
Noun	SGs/synsets	5 569	18 094	8 885
	Words	19 126	56 422	22 251
Verb	SGs/synsets	2 209	7 279	3 718
	Words	7 859	23 413	10 161
Adjective	SGs/synsets	2 450	8 557	3 485
	Words	7 785	27 124	9 001

Table 4.2: Training Data Sizes.

English.

The amount of training data available depends on which resource is used for training. The 1911, 1987 *Roget's Thesaurus* and *WordNet* 3.0 all have different counts of SGs/synsets and counts of words. Counts of all these are shown in Table 4.2. These counts do not include synsets or SGs where just one word was found in the matrix. It is no surprise that the amount of training data available in the 1987 version of *Roget's* and *WordNet* 3.0 is much larger than for the 1911 *Thesaurus*. It is surprising that *Roget's* 1987 contained so much more data than *WordNet*. This is largely because many more synsets in *WordNet* had just one synonym than was the case for SGs in *Roget's*. SGs tend to be slightly larger and have a looser definition of synonymy than do synsets. These experiments will tests how the age of the training data affects the results, as the 1911 *Roget's* is much older than the 1987 version. Also the design of the data will be tested since *Roget's* and *WordNet* are designed so differently.

4.5.3 The number of experiments

I perform experiments using the 1911 and 1987 *Roget's Thesaurus* and also using *WordNet* 3.0 as sources of training data. The training data is taken from the SGs of *Roget's* and synsets from *WordNet*. Each of these can be used for learning weights for each context $\langle r, w\rangle$ or for each relationship r. These will be referred to as "context-XX" and "relation-XX", where XX is the name of the resource being used as training data. XX may also refer to the *Roget's grouping* from which neighbouring words are identified, be it SG, Paragraph or POS. There are 3 sources of training data, 2 different kinds of training – context and relation – 6 different measures of association, 3 parts-of-speech, and 3 *Roget's-groupings* in which to evaluate each measure. In addition to this, I have

6 different levels of recall at which to evaluate each measure. This gives a total of 1944 results to examine for my supervised systems alone, not including evaluation on the unsupervised weighting and the unweighted matrix. That is why most of the results are listed in Appendix A and only those most interesting and relevant to my work are presented here.

4.6 Tuning Data

As noted in Section 4.5.3, there are far too many experiments to show all of them. One of the most instantly interesting facts I discovered is that PMI outperformed every single other measure of association on almost every test. This is why I only show results for PMI, though if the reader would like to turn to Appendix A then they will see the complete results for all measure of association. Tables A.1 shows the unsupervised results, while Tables A.2, A.3 & A.4 show the supervised results for the 1911 *Roget's Thesaurus*, the 1987 version and *WordNet* 3.0 respectively. To help visualize how these measures compare against each other, I provide two graphs comparing them at identifying nouns in the same POS at the various levels of recall. Although these graphs illustrate only one of the many experiments I carried out, the graphs for other experiments would look similar when comparing the measures of association. Figure 4.7 shows results for the unsupervised MSRs, while Figure 4.8 shows results for the context-supervised MSRs trained with the 1911 *Roget's Thesaurus*.

4.6.1 Baselines

Perhaps before discussing the results further I should establish what the baselines for these experiments really are. The unweighted matrix will be a lower baseline for all experiments while unsupervised PMI weighting will be a higher baseline. In Figures 4.9, 4.10 and 4.11, I present the baseline results for evaluation at identifying words in neighbouring SGs, Paragraphs and POSs, for nouns, verbs and adjectives.

The low baseline is not spectacular, but it does show some interesting results already. Identifying words in the same Paragraph is only slightly more difficult than identifying words in the same POS, but it is a much bigger jump to go from Paragraph to SG. For all parts-of-speech, the average accuracy within the top 10 words, when evaluated at the POS level is at least 0.214. This would mean that on average there is at least 2 words in the top 10 that correctly indicate in which POS to place a word.

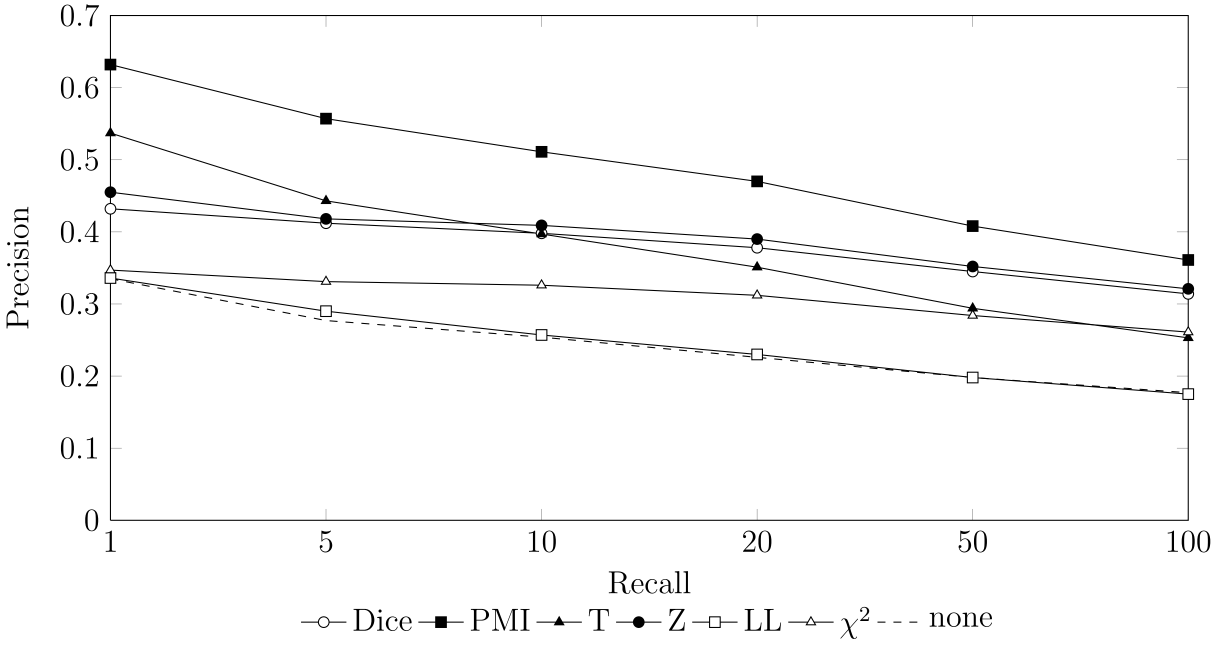

Figure 4.7: Scores for identifying nouns in the same POS, with the unsupervised MSR

The results for the unsupervised-PMI system will act as a higher baseline. The complete set of results for all unsupervised training systems is presented in Table A.1, but it is far too large to list here. My findings were that in all cases the PMI weighting was superior and so only these results will be provided here. Figures 4.9, 4.10 and 4.11 presents this higher baseline – where Training is listed as "PMI". Once again it would appear that identifying the Paragraph is a little more difficult than identifying the POS, while the SG is much more difficult than identifying the Paragraph. This time the results are much improved over the low baseline. When it comes to identifying words in the same SG, the scores are often double or triple that which can be seen in the low baseline Four or five of the words in the top 10 correctly identify a POS in which the word appears. This is a much better baseline against which to compare the supervised and combined systems.

If the reader goes to Appendix A and looks at Table A.1, there are a few interesting things that will be seen about the measures of association. First of all, PMI is consistently the best, usually followed by T-score. Z-score or Dice generally came in third and fourth, though the order between the two depended on the experiment, and χ^2 and Log Likelihood consistently were the two worst. Even for supervised learning this ordering

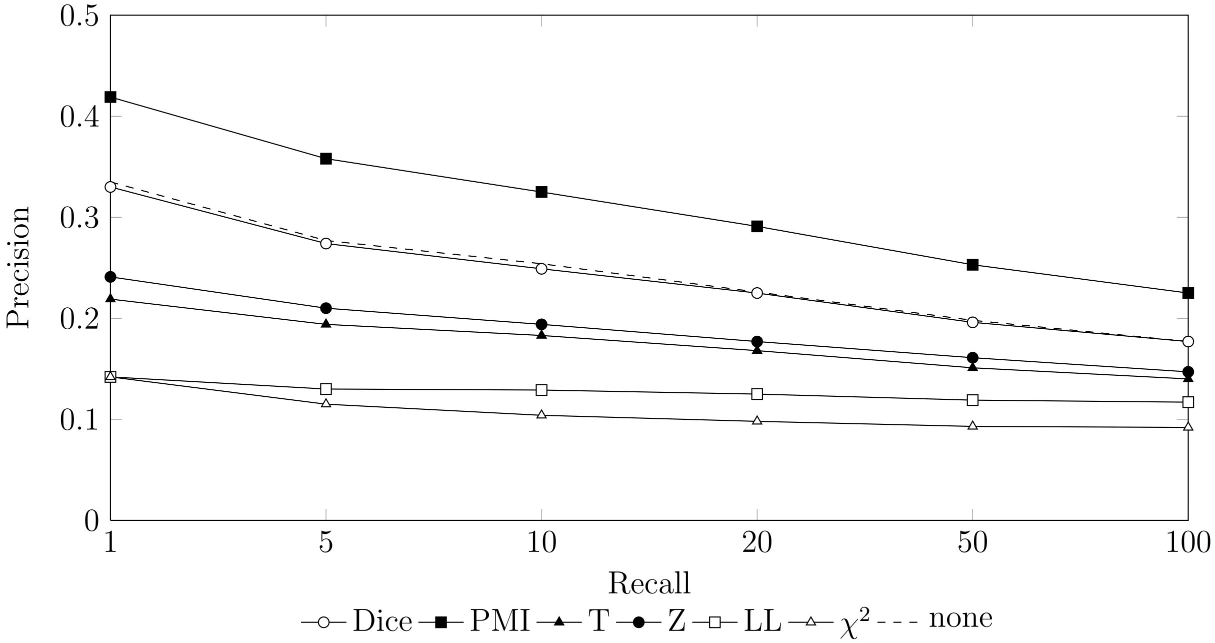

Figure 4.8: Scores for identifying nouns in the same POS, with the MSR supervised by context with *Roget's* 1911

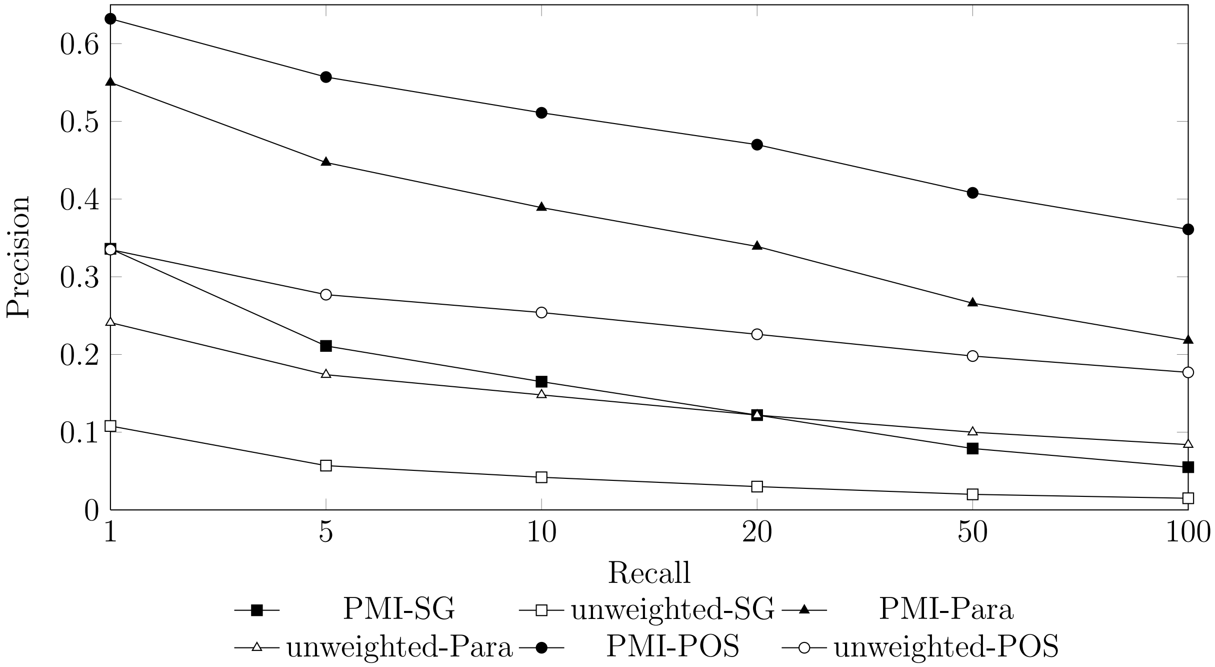

Figure 4.9: High and low baseline scores for nouns, using the PMI weighted and unweighted matrices respectively.

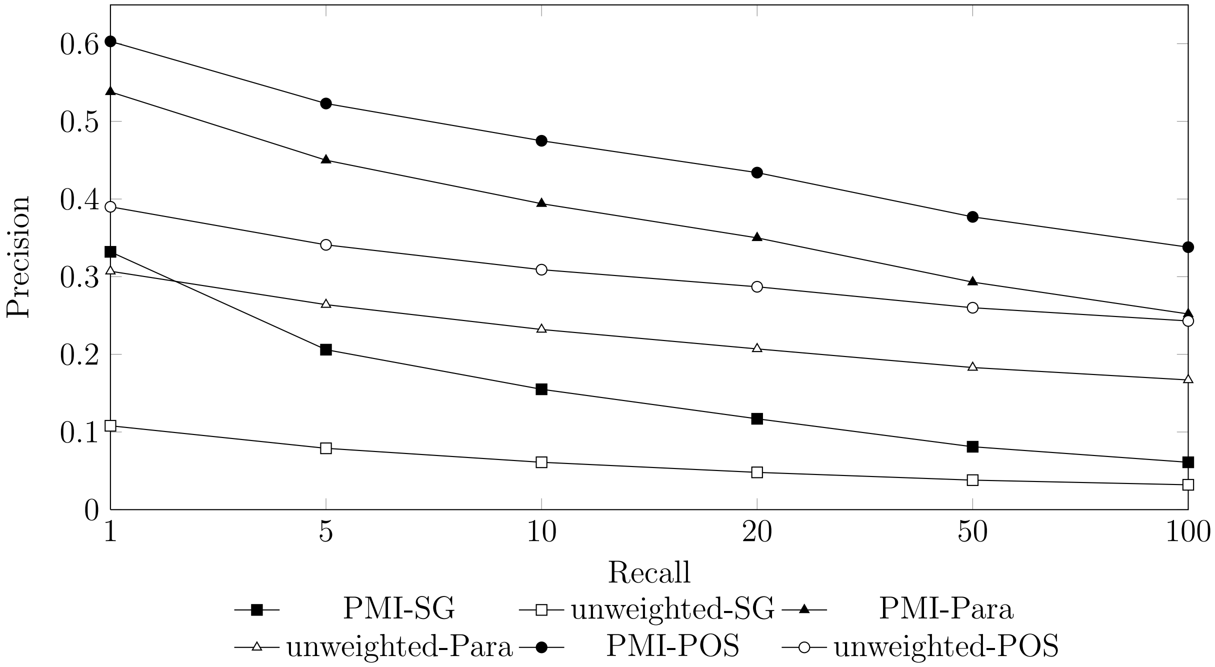

Figure 4.10: High and low baseline scores for verbs, using the PMI weighted and unweighted matrices respectively.

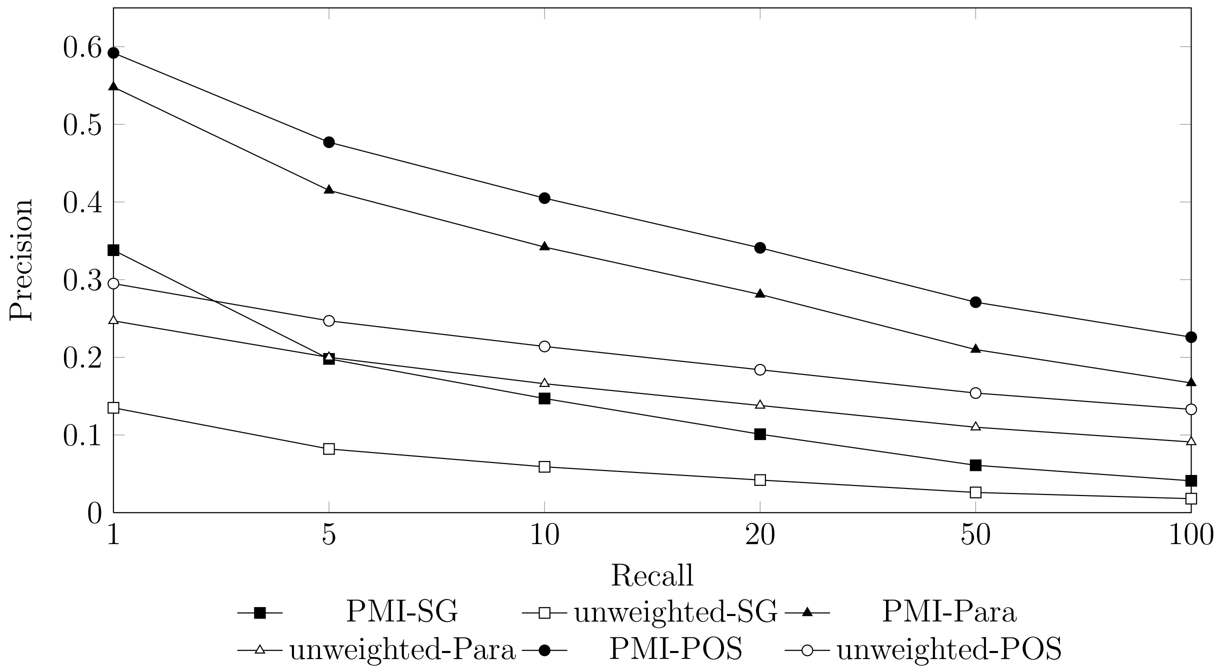

Figure 4.11: High and low baseline scores for adjectives, using the PMI weighted and unweighted matrices respectively.

was generally the case. In general χ^2 and Log Likelihood are not well suited to measuring these kinds of associations, particularly as they were outperformed by Dice. These results can be observed in Figure 4.7. This suggests an ordering to the three classes of association measures that I discuss in Section 4.3.2. Measures for values of random variables were best, followed by Dice – vector overlap – followed by measure of association between all values of a random variable. As a general rule, I would advise others to make PMI their go-to for measuring associations of this sort of problem.

4.6.2 Supervised MSRs

Before I explore more deeply the supervised results, I would like to comment on one of the key differences between the supervised and unsupervised systems. The unsupervised system finds a unique weight for every word-context pair, but the supervised system finds a unique weight only for every context. In fact when supervision is done at the relationship level, it is learning a unique weight for groups of contexts. As will be seen, this gives the unsupervised system a bit of an advantage. It will be difficult, if not impossible, for the supervised system to outperform the unsupervised one. The most

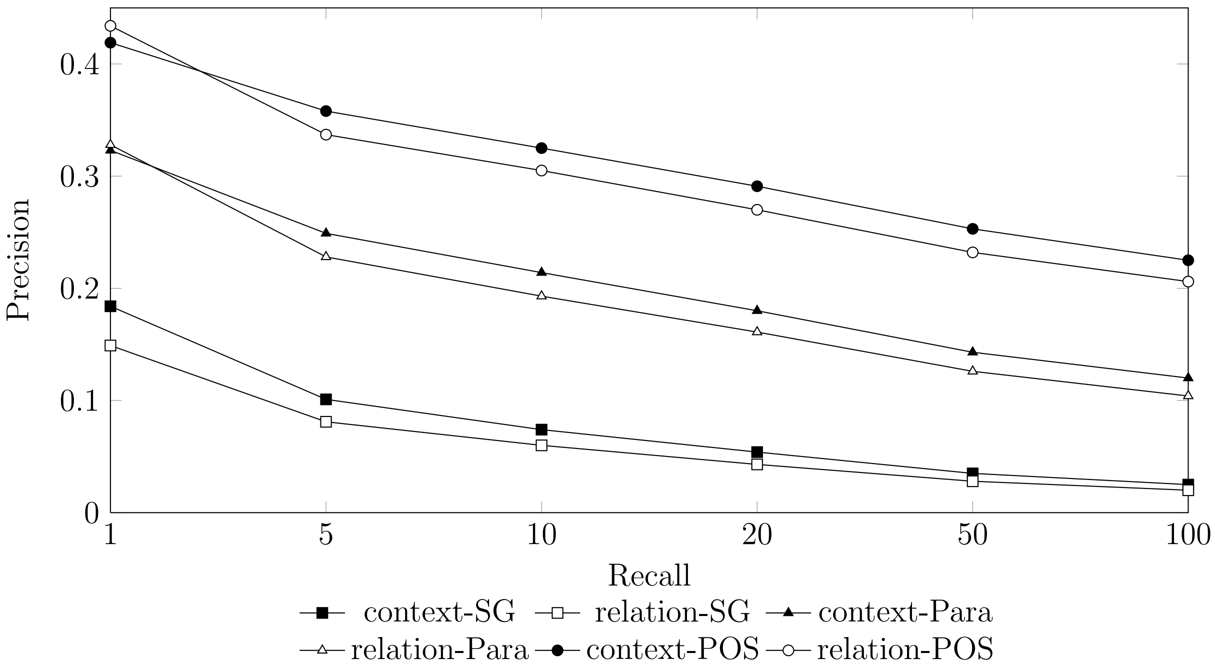

Figure 4.12: Context and relation scores for nouns, trained with *Roget's* 1911, using PMI as a measure of association

successful results will come by combining the systems. At this point it may be worth reminding the reader that – because the supervised and unsupervised systems are actually doing two different things – it is the "combined" system that will show the real benefits of supervision.

In terms of measures of association, it would appear that once again PMI was superior to the other measures. Dice appeared to be nearly equivalent to the unweighted baseline, while the other measures tended to fall below the unweighted matrix. T-score and Z-score performed similarly, while Log Likelihood and χ^2 were by far the worst. An example of these results can be seen in Figure 4.8.

The results for the supervised systems trained using *Roget's* 1911, 1987 and *WordNet* 3.0 can be found in Figures 4.12, 4.13, 4.14, 4.15, 4.16, 4.17, 4.18, 4.19 and 4.20 respectively. Results for supervision at the context level, as well as at the relation level, are presented. For the 1911 trained system – see Figures 4.12, 4.13 and 4.14 – training at the context level was clearly strongest for both nouns and verbs, though for adjectives supervision at the relation level was more frequently superior. One possible reason for this is that the adjective matrix is smaller than the verb and noun matrices. Although it has a relatively high number of words, the number of contexts it contains is much

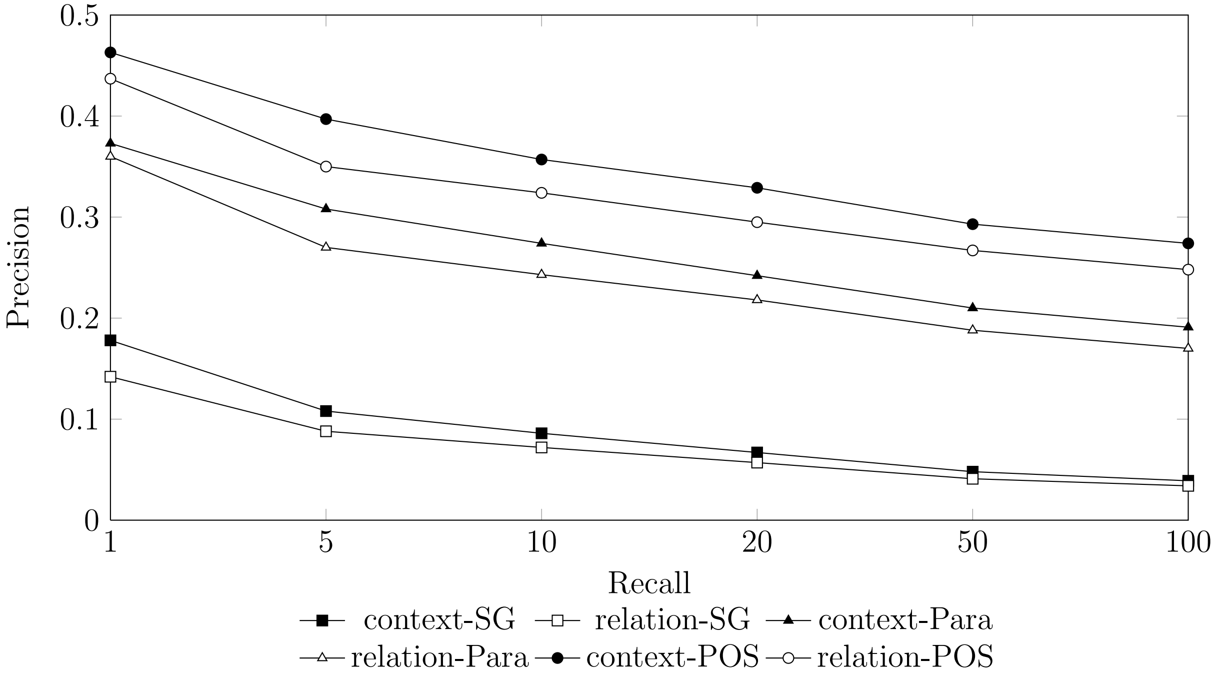

Figure 4.13: Context and relation scores for verbs, trained with *Roget's* 1911, using PMI as a measure of association

smaller than that of nouns or verbs. For nouns and verbs the improvement over the lower, unweighted, baseline was much larger than for adjectives, though none of these results meet that of the high baseline, unsupervised PMI weighting.

The results from the 1987 trained system – see Figures 4.15, 4.16 and 4.17 – show similar trends to the 1911 results. Also of interest is that the 1987 trained version tended to give worse scores when identifying nouns than the 1911 version, but was superior for verbs. For adjectives the superiority of training with the 1987 *Roget's vs* the 1911 *Roget's* is not so clear. As evaluation is done on the 1987 *Roget's Thesaurus*, it is a surprise that the 1911 version could outperform it on any measure. That said, the differences are small.

When conducting the same analysis on the *WordNet* 3.0 trained data – Figures 4.18, 4.19 and 4.20 – I found another interesting set of results. For nouns, once again the 1911 *Roget's Thesaurus* performs best while for verbs and adjectives generally the *WordNet*-trained data worked better. When compared agains the 1987 *Roget's Thesaurus*, *WordNet* 3.0 performed worse on verbs, though was usually comparable on nouns and adjectives. Again adjectives were better trained at the relationship level than the context

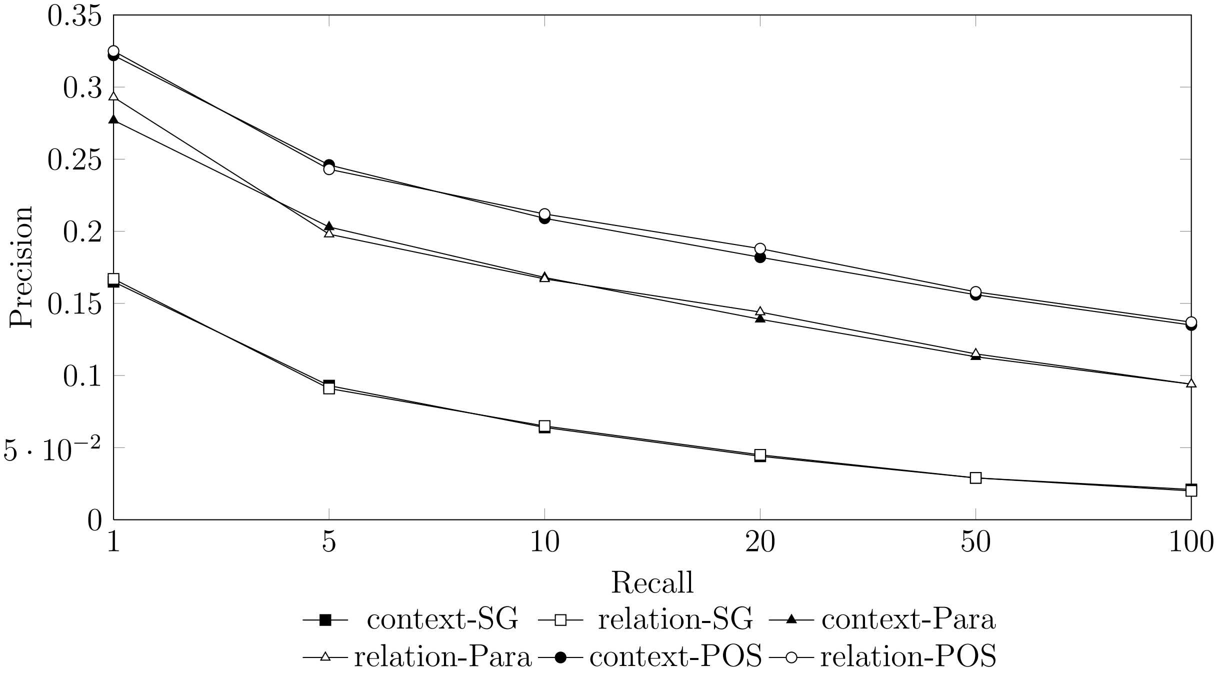

Figure 4.14: Context and relation scores for adjectives, trained with *Roget's* 1911, using PMI as a measure of association

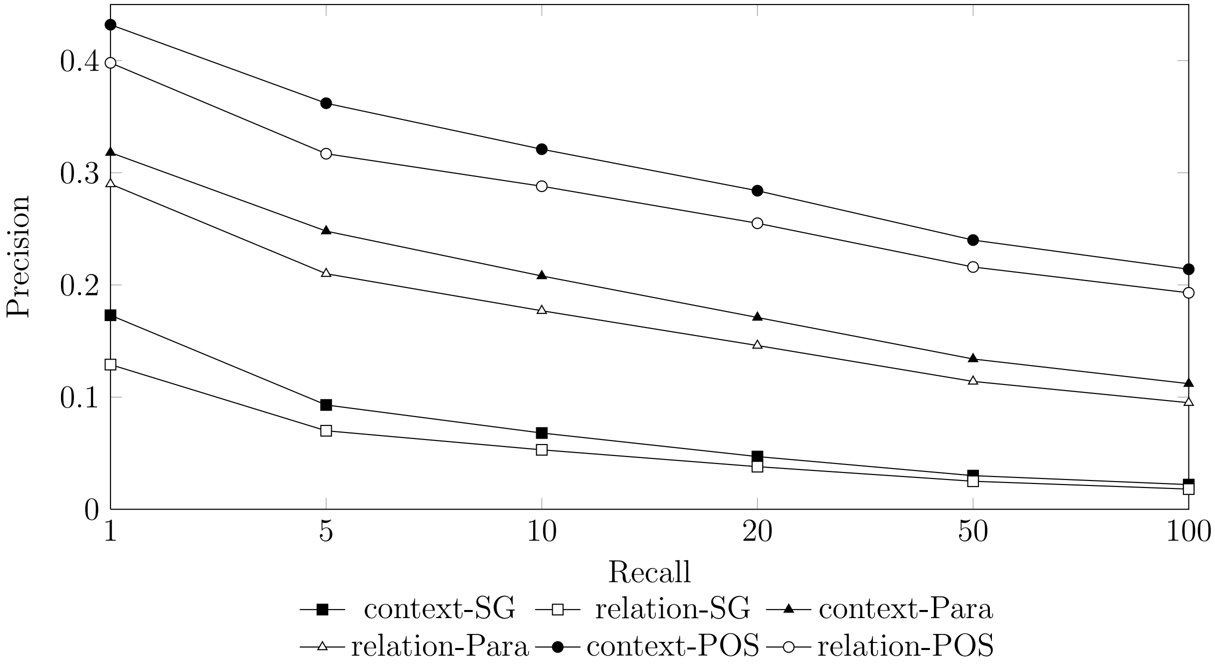

Figure 4.15: Context and relation scores for nouns, trained with *Roget's* 1987, using PMI as a measure of association

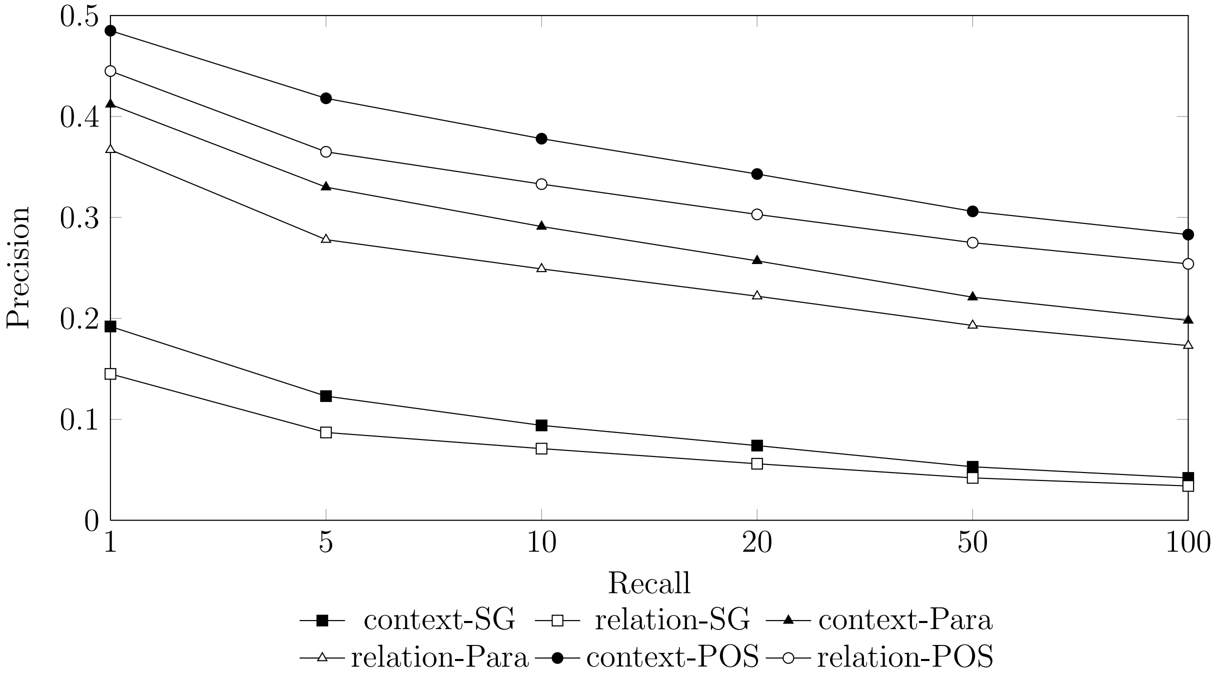

Figure 4.16: Context and relation scores for verbs, trained with *Roget's* 1987, using PMI as a measure of association

level.

The matrix for adjectives was noticeably smaller than that for nouns and verbs. Although it is difficult to say for sure with these tests, it seems likely that smaller matrices benefit more from supervision at the relationship level, rather than at the context level. Since the amount of training data between adjectives and verbs is quite comparable I do not think it is for lack of training data that caused the adjective learning to work best at the relation rather than context level. Experiments could be conducted using many smaller matrices made out of nouns or verbs to test this theory and perhaps determine a threshold beyond which one kind of supervision surpasses the other. It may also be possible that this is a phenomenon unique to adjectives, or perhaps it is due to the fact that the adjective matrix contains both adjectives and adverbs, although adverbs were not included in the training process.

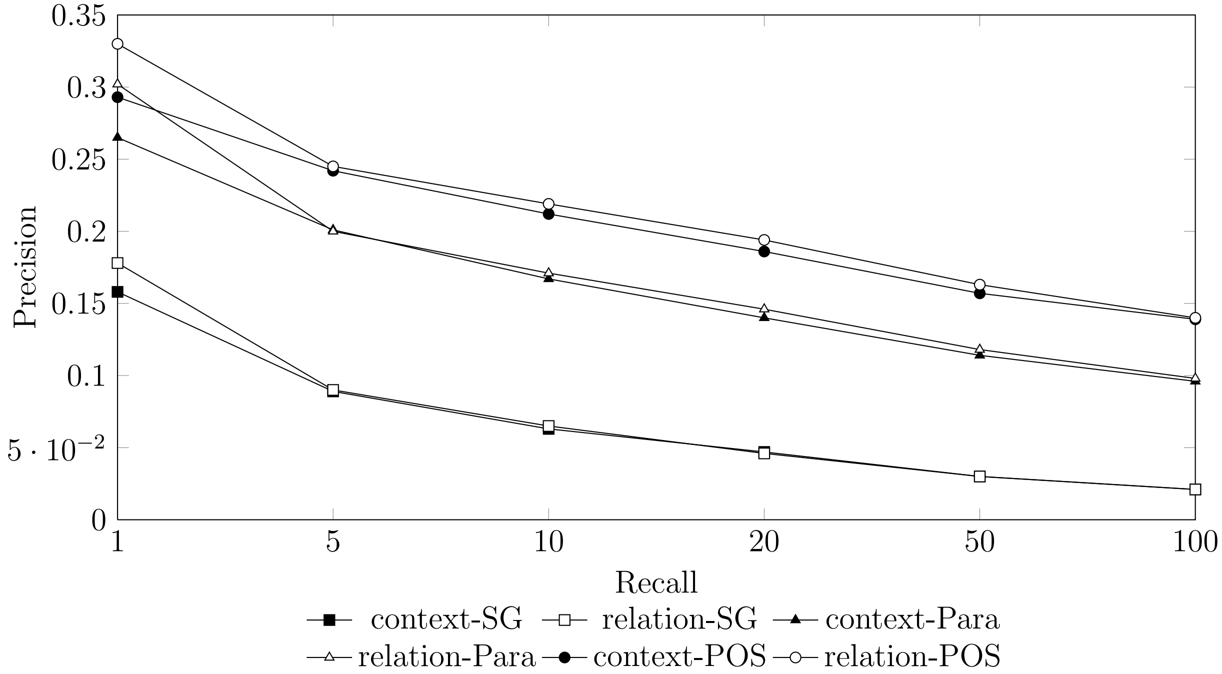

Figure 4.17: Context and relation scores for adjectives, trained with *Roget's* 1987, using PMI as a measure of association

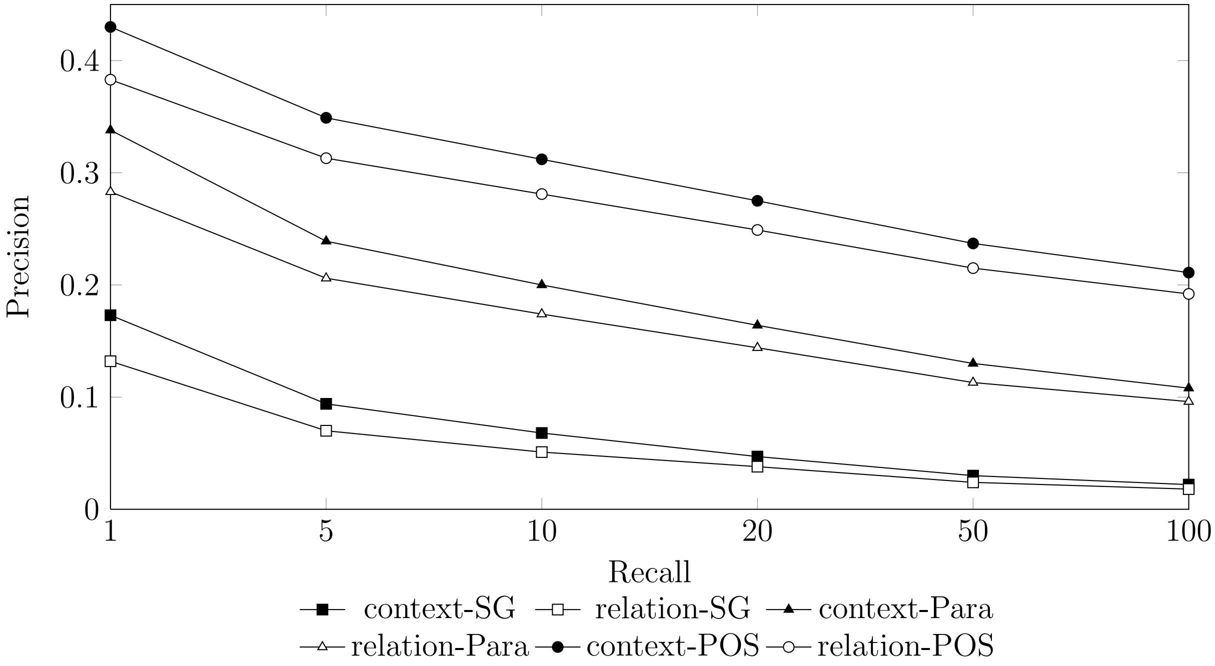

Figure 4.18: Context and relation scores for nouns, trained with *WordNet*, using PMI as a measure of association

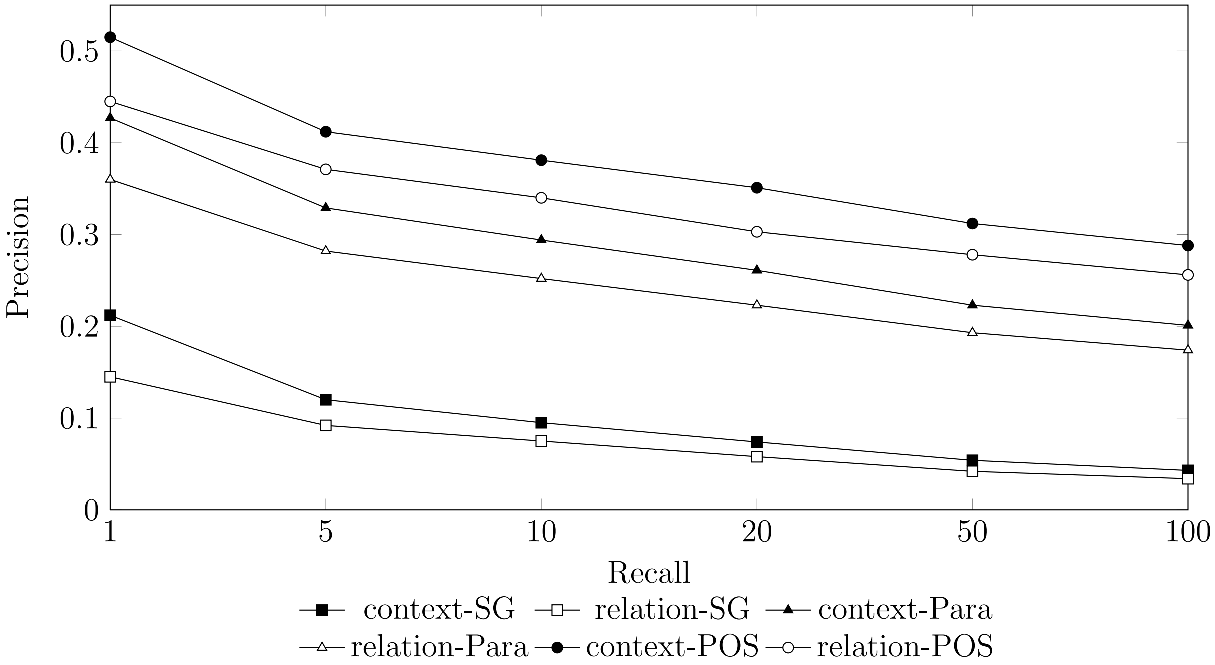

Figure 4.19: Context and relation scores for verbs, trained with *WordNet*, using PMI as a measure of association

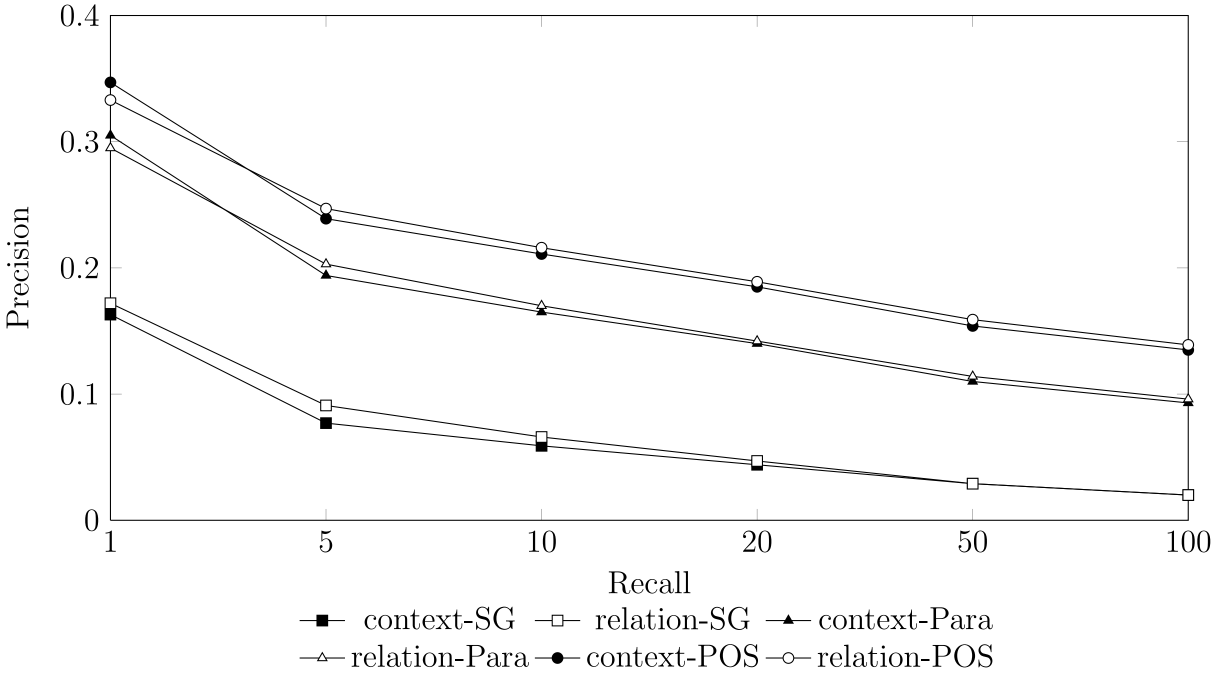

Figure 4.20: Context and relation scores for adjectives, trained with *WordNet*, using PMI as a measure of association

4.7 Test Data

I will now present and discuss the results on the test data. In the previous section I found that PMI weighting worked best for unsupervised weighting. For supervised weighting PMI-context worked best for nouns and verbs while PMI-relation worked better for adjectives. Here I will present the results for the supervised and unsupervised systems on the test set and will combine these systems. The combined system is essentially performing unsupervised weighting on top of a matrix that has already been re-weighted using the supervised weighting.

4.7.1 Unsupervised

The unsupervised results on the testing data are shown in Table 4.3. The results seen here are comparable to those found in the tuning dataset and so do not really need much discussion. These will be the low and high baselines against which the supervised and combined systems are compared.

4.7.2 Supervised

The results for supervision with the 1987 and 1911 *Roget's Thesaurus* and *WordNet* 3.0 are shown in Table 4.4. It clearly shows how the different systems compare against each other. For nouns the 1911 *Roget's Thesaurus* is generally superior to either the 1987 version or *WordNet* 3.0. When it comes to verbs *WordNet* performs marginally better and for adjectives *Roget's* 1987 was the best. That said, the choice of training data does not appear to greatly affect the outcome of this experiment, as weights derived from all three sources were comparable. I believe this suggests the 1911 version to be best as it produces comparable results with less training data.

Once again, Table 4.4 shows that this supervised system performs better than the lower baseline but not as good as the higher baselines shown in Table 4.3. This is no surprise as it mirrors what I found in the tuning data.

4.7.3 Combined

The results for the combined system are shown in Table 4.5. In this table I labeled in bold all cases where the combined system outperformed unsupervised PMI – the high baseline – by a statistically significant margin. When the supervised system had a

			Top	Top	Top	Top	Top	Top
POS	Group	Training	1	5	10	20	50	100
N.	SG	unweighted	0.104	0.060	0.042	0.033	0.022	0.016
		unsupervised-PMI	0.358	0.236	0.179	0.130	0.084	0.059
	Para	unweighted	0.262	0.185	0.155	0.136	0.109	0.092
		unsupervised-PMI	0.560	0.469	0.412	0.352	0.279	0.230
	POS	unweighted	0.376	0.296	0.262	0.239	0.207	0.186
		unsupervised-PMI	0.645	0.579	0.537	0.490	0.423	0.374
VB.	SG	unweighted	0.128	0.082	0.064	0.052	0.039	0.032
		unsupervised-PMI	0.302	0.206	0.162	0.126	0.086	0.065
	Para	unweighted	0.303	0.243	0.230	0.214	0.192	0.176
		unsupervised-PMI	0.513	0.445	0.407	0.358	0.304	0.264
	POS	unweighted	0.398	0.331	0.318	0.299	0.276	0.256
		unsupervised-PMI	0.582	0.526	0.487	0.444	0.396	0.357
ADJ.	SG	unweighted	0.133	0.080	0.058	0.044	0.028	0.020
		unsupervised-PMI	0.345	0.206	0.156	0.115	0.069	0.046
	Para	unweighted	0.272	0.207	0.176	0.153	0.116	0.095
		unsupervised-PMI	0.562	0.417	0.363	0.304	0.231	0.185
	POS	unweighted	0.317	0.259	0.224	0.205	0.163	0.139
		unsupervised-PMI	0.600	0.480	0.431	0.368	0.295	0.247

Table 4.3: Testing data evaluation results for identifying related words in the same *Roget's grouping*. These are baselines measured using only an unweighted matrix.

			Top	Top	Top	Top	Top	Top
POS	Group	Training	1	5	10	20	50	100
N.	SG	context-1911	0.171	0.102	0.077	0.056	0.037	0.026
		context-1987	0.176	0.104	0.072	0.050	0.033	0.024
		context-WN	0.172	0.096	0.071	0.049	0.032	0.024
	Para	context-1911	0.350	0.254	0.218	0.188	0.150	0.125
		context-1987	0.357	0.259	0.215	0.178	0.141	0.118
		context-WN	0.360	0.254	0.216	0.177	0.140	0.118
	POS	context-1911	0.440	0.363	0.330	0.303	0.262	0.233
		context-1987	0.456	0.376	0.334	0.296	0.252	0.223
		context-WN	0.466	0.370	0.333	0.291	0.252	0.224
VB.	SG	context-1911	0.187	0.109	0.087	0.067	0.049	0.039
		context-1987	0.213	0.123	0.097	0.074	0.054	0.043
		context-WN	0.202	0.125	0.098	0.077	0.054	0.044
	Para	context-1911	0.380	0.308	0.280	0.247	0.216	0.197
		context-1987	0.415	0.330	0.291	0.267	0.229	0.209
		context-WN	0.412	0.338	0.301	0.270	0.232	0.211
	POS	context-1911	0.468	0.394	0.368	0.334	0.303	0.283
		context-1987	0.480	0.418	0.382	0.356	0.318	0.299
		context-WN	0.482	0.426	0.393	0.365	0.324	0.303
ADJ.	SG	relation-1911	0.157	0.086	0.065	0.050	0.032	0.023
		relation-1987	0.148	0.083	0.067	0.051	0.033	0.024
		relation-WN	0.145	0.088	0.068	0.050	0.033	0.023
	Para	relation-1911	0.305	0.218	0.189	0.161	0.126	0.102
		relation-1987	0.298	0.218	0.193	0.165	0.130	0.107
		relation-WN	0.297	0.221	0.189	0.161	0.127	0.103
	POS	relation-1911	0.358	0.273	0.243	0.212	0.175	0.148
		relation-1987	0.357	0.277	0.250	0.217	0.179	0.153
		relation-WN	0.353	0.278	0.242	0.213	0.175	0.148

Table 4.4: Testing data evaluation results for identifying related words in the same *Roget's grouping*. These are baselines measured using only an unweighted matrix.

statistically worse performance than unsupervised PMI, I annotated it in italics. This was accomplished by taking every single word in the testing set as its own fold – 1000 folds for nouns and 600 folds for verbs/adjectives – and running a two tailed Student's T-test. A result was considered significant when $p < 0.05$.

One immediate observation is that this combined method is rarely better than unsupervised PMI at identifying words in the same SG. There are two cases where it is significantly worse for the 1911 *Roget's*, one case where it is worse and one better for the 1987 *Roget's* and five cases where it is worse for *WordNet*. That said, when measuring words in the same Paragraph or POS it is much more successful. In the case of nouns both versions of *Roget's* and *WordNet* are statistically improved when the top 20 or more nearest neighbours are counted for both Paragraph and POS. For verbs the improvement is not quite as pronounced, though in all cases either there was an improvement or no statistically significant difference. In the case of adjectives the improvement is much less noticeable. *WordNet* showed no statistically significant difference either way for adjectives while the 1911 and 1987 *Roget's* sporadically showed improvement at a few levels of recall.

Clearly the higher the recall level the easier it is to measure these differences as being statistically significant. When selecting just one word, only once was there a significant difference either way. I expect that when selecting in which POS or Paragraph to place a new word, the top 10 or 20 words will be most useful and so these will likely be the best levels of recall to examine.

If one looks at the three sources of training data separately, they can count how many times there was a statistically significant improvement, no improvement, or a statistically significant decrease. I will report this as a triple (improve/no change/decrease) in Table 4.6.

Both versions of the *Roget's*-trained combined methods are fairly consistent in showing improvement over the unsupervised methods. This is most clear for nouns and verbs, though there is some success for adjectives. When training with *WordNet*, it was not as successful. Probably the biggest difference is that the training was done using synsets not Semicolon Groups. Since the evaluation is done on Semicolon Groups from the 1987 *Roget's Thesaurus* it seems natural that any weighting trained with that resource, or a similar resource – the 1911 version – should outperform one trained using a different resource. It is, nonetheless, not obvious that learning from SGs should improve classification of words in the same Paragraph or POS.

POS	Group	Training	Top 1	Top 5	Top 10	Top 20	Top 50	Top 100
N.	SG	1911-comb	0.358	*0.225*	*0.175*	0.132	0.084	0.058
	Para	1911-comb	0.568	0.472	0.418	**0.361**	**0.286**	**0.234**
	POS	1911-comb	0.659	**0.588**	**0.548**	**0.501**	**0.431**	**0.382**
VB.	SG	1911-comb	0.310	0.207	0.163	0.124	0.086	0.064
	Para	1911-comb	**0.550**	0.456	0.414	0.362	0.307	**0.268**
	POS	1911-comb	0.605	0.533	**0.500**	**0.455**	**0.401**	**0.362**
ADJ.	SG	1911-comb	0.343	**0.209**	0.157	0.114	0.069	0.046
	Para	1911-comb	0.563	0.422	0.365	0.304	**0.232**	0.184
	POS	1911-comb	0.602	0.484	0.431	0.368	0.296	0.247
N.	SG	1987-comb	0.359	*0.229*	0.177	**0.134**	0.085	0.059
	Para	1987-comb	0.564	0.471	**0.419**	**0.365**	**0.285**	**0.234**
	POS	1987-comb	0.651	0.584	**0.549**	**0.501**	**0.430**	**0.381**
VB.	SG	1987-comb	0.308	0.211	0.167	0.127	0.087	0.064
	Para	1987-comb	0.525	**0.457**	**0.417**	0.362	0.305	**0.266**
	POS	1987-comb	0.588	**0.537**	**0.499**	**0.453**	0.399	**0.360**
ADJ.	SG	1987-comb	0.343	0.208	0.158	0.115	0.069	0.046
	Para	1987-comb	0.565	**0.421**	0.365	0.304	0.232	0.184
	POS	1987-comb	0.603	0.483	0.431	0.367	0.296	0.247
N.	SG	WN-comb	0.359	*0.222*	*0.173*	0.129	0.084	*0.058*
	Para	WN-comb	0.571	0.468	0.410	**0.357**	**0.284**	**0.232**
	POS	WN-comb	0.654	0.586	0.541	**0.495**	**0.430**	**0.380**
VB.	SG	WN-comb	0.323	0.209	0.161	0.125	*0.084*	*0.063*
	Para	WN-comb	0.522	0.450	0.408	0.359	**0.301**	**0.262**
	POS	WN-comb	0.587	0.531	**0.495**	**0.451**	0.395	0.356
ADJ.	SG	WN-comb	0.335	0.207	0.157	0.116	0.069	0.046
	Para	WN-comb	0.553	0.419	0.364	0.304	0.232	0.185
	POS	WN-comb	0.595	0.483	0.430	0.368	0.296	0.247

Table 4.5: Evaluating results for the combined measure with PMI. Significant improvements over unweighted PMI in **bold**, significantly worse results in *italics*

Resource	Noun	Verb	Adjective	All
Roget's 1911	(8/8/2)	(6/12/0)	(2/16/0)	(16/36/2)
Roget's 1987	(9/8/1)	(7/11/0)	(1/17/0)	(17/36/1)
WordNet 3.0	(6/9/3)	(4/12/2)	(0/18/0)	(10/39/5)

Table 4.6: Number of improved statistically improved/unaffected/decreased results for each source of training data.

4.7.4 Using these Measures to Enhance *Roget's*

From these results it appears that when trying to identify which POS or Paragraph in which to place a word, using the combined methods will work best, particularly for nouns and verbs. Supervised weighting provided some small improvement for adjectives as well, but the advantages are not so strong. When identifying words in the same Semicolon Group there does not seem to be a strong advantage or disadvantage to using the supervised or unsupervised methods. I will apply the supervised methods PMI-context for adding new nouns and verbs and PMI-relation for adding new adjectives to *Roget's*.

4.8 Other Things One Can Learn with Supervised Matrix Weighting

My method of leaning weights for semantic relatedness is, in theory, not limited to learning synonymy. It is actually a general method for learning some sort of relation between words in text. This section is peripheral to the main narrative of this thesis. It should cause no loss of continuity to skip directly to Section 4.9.

4.8.1 Learning Sentiment & Emotion

Research on sentiment analysis has been one of the more successful areas of NLP in the last 10 years, and more recently there has been a move towards studying emotion in text. A number of techniques have been applied to finding the sentiment or emotion of a word. One established method is that of Turney (2002) where PMI is used to learn association between a word and either positive or negative sentiment. Words that appear in the same contexts as words known to be negative are labeled as negative, while those more likely to appear in the context of a word that is positive are labeled as positive. I will not repeat these experiments, but rather use my supervised matrix weighting to learn a matrix where contexts are given higher weights if they tend to contain words that have the same sentiment or emotion.

4.8.2 Training Data

To learn emotion in words I used the NRC Emotion and Sentiment word list (version 0.5) as a dataset (Mohammad and Turney, 2012). This consists of lists of words annotated with the emotions and sentiment expressed by these words. Each word can be labeled with up to 8 emotions or have no emotion associated, likewise each word can be labeled with positive and negative sentiment or contain no sentiment. Rather than grouping words together based on closeness in meaning, they are organized into closeness in sentiment or emotion. In the experiments with *Roget's Thesaurus* and *WordNet* there were thousands of Semicolon Groups or synsets, while when working with sentiment or emotion there were just 2 or 8 classes respectively. Counts of the Emotions or Sentiment are as follows:

- Emotion: 2283
  - Joy: 353
  - Sadness: 600
  - Fear: 749
  - Surprise: 275
  - Disgust: 540
  - Anger: 647
  - Trust: 641
  - Anticipation: 439
  - No-emotion: 4808
- Sentiment: 2821
  - Positive: 1183
  - Negative: 1675

POS	Count	Sentiment	Emotion
Noun	Emo/Senti	2	8
	Words	1824	2834
Verb	Emo/Senti	2	8
	Words	705	1166
Adjective	Emo/Senti	2	8
	Words	1135	1479

Table 4.7: Training Data Sizes.

- No-sentiment: 4270

A word can be labeled with multiple emotions or sentiments, but they cannot be both emotional and non-emotional, or sentimental and non-sentimental. The words are not labeled with part-of-speech, though I will assume that if a word from this list is found in the noun matrix then it must have a noun sense, and the same for verbs and adjectives. I will not make use in these experiments of words labeled with no emotion or no sentiment. Counts of the number of words and number of groupings – emotions and sentiments – are shown in Table 4.7.

For evaluation I randomly selected 200 positive and 200 negative words for each sentiment for each part-of-speech, nouns, verbs and adjectives. The rest of the data is used for training. In the evaluation set only words that have one sentiment or one emotion are used. It is actually quite common for a word to express multiple emotions, but that would make evaluation more difficult. This was a limitation for emotion, particularly for verbs. I was able to take 30 words from each emotion for nouns and adjectives, but only 15 from each verb. This means the sentiment evaluation set contains 400 words and either 240 or 120 words for the emotion evaluation set.

4.8.3 The Experiments & Analysis

The results for the sentiment evaluation are in Table 4.8. Values for levels of recall after 1, 5 and 10 words are not shown as in all cases the average was 99% or higher. Instead I report the values for both positive and negative results with 20, 50 and 100 as the recall levels. One interesting observation is that the unsupervised PMI is not necessarily a higher baseline than the unweighted method. At some levels of recall PMI is actually worse. If two words appear in the same context they may be related in some way, yet

		Positive			Negative		
POS	Training	20	50	100	20	50	100
N.	none	**1.000**	0.888	0.507	0.998	0.847	0.477
	PMI	0.958	0.813	**0.611**	0.968	0.853	**0.697**
	relation	**1.000**	**0.903**	0.580	**1.000**	**0.952**	0.645
	relation-combined	0.958	0.814	**0.611**	0.965	0.851	0.696
	context	0.986	0.855	0.524	0.960	0.729	0.482
	context-combined	0.934	0.768	0.574	0.968	0.832	0.686
VB.	none	1.000	0.966	0.559	1.000	**0.999**	0.836
	PMI	1.000	0.951	0.653	1.000	0.997	0.902
	relation	1.000	0.990	0.599	1.000	**0.999**	0.852
	relation-combined	0.999	0.950	0.671	1.000	0.998	**0.923**
	context	1.000	**0.995**	**0.711**	1.000	0.998	0.855
	context-combined	1.000	0.932	0.622	1.000	0.998	0.915
ADJ.	none	0.994	0.893	0.565	0.993	0.958	0.824
	PMI	0.992	0.915	0.707	0.992	0.943	0.802
	relation	**0.998**	0.909	0.630	**0.998**	**0.964**	**0.844**
	relation-combined	0.993	**0.917**	**0.711**	0.992	0.939	0.787
	context	0.997	0.892	0.626	0.996	0.936	0.721
	context-combined	0.992	0.895	0.692	0.993	0.945	0.804

Table 4.8: Evaluating results for sentiment.

express different sentiment. As such it is more important to identify which contexts tend to contain words of the same sentiment.

In Table 4.8, one can see that for all three parts-of-speech the supervised method of weighting matrices at the relation level tended to work very well. In fact there is only one case where either the unsupervised matrix or the unweighted matrix was superior to any of the supervised or combined methods, that being evaluation on negative sentiment nouns at the 100 recall level. Even then it was only marginally better than the relation-combined method.

The results for evaluation on emotional words is shown in Table 4.9. The results for all 8 emotions are averaged together in this table to make it a bit more readable and save space. For full results see Appendix B, where Tables B.1 & B.2 show the complete results for the sentiment and emotion experiments. I do not include the Top 1 recall level in Table 4.9, as once again all systems had a score of 99% or higher.

Once again, it can be seen that at no point are the scores for unsupervised PMI actually the best. In two cases the scores for the unweighted matrix are best, though most often one of the supervised methods or combinations is superior. These results seem to suggest that learning at the relation level is better for identifying emotionally related words in the top 10, but at higher recall levels of 50 or 100 the combined method with supervision at the context level was superior. In either case there was a benefit from including supervised learning.

4.8.4 Discussion

An immediate observation is that it would appear that unsupervised-PMI is not consistently better than even the unweighted method. Learning a word's emotion and sentiment would appear to be two problems on which these supervised techniques have potential. Measuring association between a word and a context will naturally lend itself to finding synonyms, but if the problem changes somewhat to that of learning emotional similarity rather than similarity in meaning, then supervised learning brings much benefit.

Unfortunately the size of the training data and evaluation set means that I am somewhat limited when it comes to learning these kinds of relationships. Nonetheless, this suggests emotion and sentiment could be an interesting direction to take this work. These experiments show an attempt to customize a MSR so as to incorporate information other than synonymy. These MSRs have been customized to better identify words of the same

POS	Training	Top 5	Top 10	Top 20	Top 50	Top 100
N.	none	**0.999**	0.980	0.840	0.443	0.222
	PMI	0.980	0.924	0.802	0.541	0.330
	relation	0.998	**0.988**	**0.912**	0.536	0.275
	relation-Combined	0.980	0.925	0.800	0.541	0.331
	context	0.987	0.920	0.722	0.393	0.201
	context-combined	0.965	0.904	0.777	**0.543**	**0.339**
VB.	none	**1.000**	**1.000**	**0.976**	0.588	0.296
	PMI	**1.000**	0.995	0.961	0.676	0.382
	relation	**1.000**	**1.000**	0.968	0.624	0.321
	relation-Combined	**1.000**	**1.000**	0.963	0.702	**0.408**
	context-PMI	**1.000**	1.000	0.975	0.660	0.349
	context-combined	**1.000**	0.996	0.963	**0.705**	0.407
ADJ.	none	0.988	0.938	0.827	0.473	0.238
	PMI	0.980	0.935	0.860	0.589	0.321
	relation	**0.993**	**0.950**	0.823	0.470	0.236
	relation-Combined	0.983	0.939	0.868	0.595	0.324
	context	0.992	0.942	0.831	0.483	0.243
	context-combined	0.983	0.941	**0.871**	**0.610**	**0.335**

Table 4.9: Evaluating results for emotion.

emotion and sentiment.

I do not directly compare my results against (Turney, 2002), because it is beyond the scope of this thesis and this is already a fairly lengthy digression. That said, the method he proposes would appear to be more logical, particularly for sentiment. It makes sense to associate contexts with either positive or negative sentiment rather than associating a context with the likelihood of containing word pairs of the same sentiment, as I do. This is logical because sentiment identification is actually a 2-class problem, while identifying synonymy is an unlimited class problem – there are thousands of Semicolon Groups and synsets in *Roget's* and *WordNet*, and even then one could always invent new synonym groupings. But which method makes more sense for Emotion? The dataset would seem to suggest that Emotion learning is an 8-class problem, but this is simply not so. WordNet Affect contains words labeled with 6 emotions and some emotions – for example "jealousy", "awe" or "boredom" – are absent from this list. The set of emotions may not be a fixed-class problem. It would be possible to extend the methods of Turney (2002) to work for emotion, though I believe that limiting the possible emotions to a set of 8 would not necessarily be a good idea.

4.9 Conclusion

Of the measures of association that I experiment with, PMI appears to be the best for both supervised and unsupervised learning. In some ways, this should not be surprising, because PMI is the logarithm of the observed over the expected value. It effectively measures how unlikely, and so how significant, the observed value really is. Compared with the other measures of association, PMI would seem to have an intuitive advantage. Performing supervised learning at the context level works best for nouns and verbs, but for adjectives the best results came from measuring association at the level of relationships. I believe this is because the adjective matrix is smaller in size than the noun and verb ones. Most likely having a larger amount of data with which to populate the matrix makes it easier to come up with good results at low levels.

Ultimately the best results came by combining the strongest supervised and unsupervised methods. Supervised and unsupervised PMI combined together generally improved over the unsupervised PMI baseline. My experiments found that training using *Roget's Thesaurus* worked better than using *WordNet*, although this may be due to evaluation being conducted on *Roget's Thesaurus*. Nonetheless I have shown that supervised weighting is a general method for learning relatedness and could be applied to learning synonyms in *WordNet* in order to expand its lexicon. These MSRs have effectively been customized using either *Roget's* or *WordNet*.

My experiments with two versions of *Roget's Thesaurus* did not show that the older version from 1911 performed much worse that the 1987 version, despite the 1911 version having much less data and being older. Perhaps the 1911 *Roget's Thesaurus* has enough data so that increasing its size does not noticeably improve results.

4.9.1 Future Work

Obviously there are many avenues for future work. I have already found success learning semantic relatedness using *Roget's Thesaurus* and *WordNet*, and show some experiments learning emotion/sentiment using the NRC Emotion/Sentiment word lists. The same sort of matrix weighting could be used to learn antonymy or other semantic relations.

I have trained the matrix using data from just one source at a time. Another experiment might be to mix data from *WordNet* and *Roget's Thesaurus* to see if that improves the training process. This could be done either by simply mixing the two datasets, or it might be possible to use a mapping from *WordNet* and *Roget's* to expand either the Semicolon Groups, or the synsets. Work done in (Nastase and Szpakowicz, 2001) and (Kwong, 1998a) could be used.

I explored two different methods of training a system, learning at the relation level and learning at the context level. When learning at the context level, if for any reason a context could not be given a weight it was assigned an average weight. It might make sense to try combining these two methods, by using the relation weight in cases where no context weight could be assigned. Perhaps the relation weight could be used to given scores only to contexts that contain relatively few words.

Latent Semantic Analysis (LSA)

LSA is a popular method of reducing the dimensionality of a matrix (Landauer and Dumais, 1997). It works by identifying the dimensions along which the matrix has the highest variance and then ranks them in order. Often the top 250 to 1000 of these dimensions are used.

I did not experiment with LSA in this thesis, as it adds another layer between my supervised matrix re-weighting and the evaluation. Since my method of matrix re-weighting is key to the novelty in this thesis, I believe it should be evaluated directly, as I have done. LSA is a well established method and would not add anything new to my thesis.

That said, it may be possible to improve results by running LSA on top of the matrices as they are weighted now. I leave this for future work.

Another option would be to run LSA on the unweighted matrix and then attempt to use PMI in either its supervised or unsupervised fashion on top of it. This might be challenging as the supervised method of weighting a matrix relies on integer counts of the number of times pairs of words co-occur in the same context. To my knowledge this is not typically done with unsupervised PMI either.

One advantage of LSA is that its reduced vector space should make cosine similarity faster to calculate. This may not be the case in extremely sparse matrices as the one I am using. Using LSA every feature would have 250 ..1000 features compared to hundreds of thousands for my matrix, though with LSA almost every spot in the matrix will have a non-zero value. It is only necessary to compute the non-zero entries when measuring cosine similarity, so only words with more than 250 ..1000 non-zero features would benefit in terms of run time.

Additionally methods like RWF/GRWF (Broda et al., 2009) could be incorporated into this work, but I have not attempted that either. Other MSRs like Lin (1998a) could be combined with my supervised method. There is much possible work to be done in this area.

Cross-Language Semantic Relatedness

The kind of MSRs that I work with cannot be used to measure relatedness between words in two different languages, or between two parts-of-speech, as these words never appear in the same contexts. It may be possible to create a system that can measure semantic relatedness between pairs of words in different languages. Some work has been performed on this topic, but it is still largely uncharted territory (Mohammad et al., 2007; Haghighi et al., 2008; Hassan and Mihalcea, 2009).

I hypothesize that my work can be extended to identify translations of words across pairs of languages. To do this, I would find pairs of contexts (one from each language) that tend to contain translations. If a pair of contexts tends to contain words that are translations of each other, then a mapping can be produced between these contexts. Translations can be taken from a bilingual dictionary of some sort. The main challenge here will would be to determine how best to create a mapping. Mappings could be one-to-one or more likely many-to-many. The strength of a mapping could also be adjusted depending on the strength of the association between a pair of contexts. Precisely how to measure the association between two contexts is also an open problem, although

Pointwise Mutual Information would appear to be the obvious starting point. In some ways this project could be seen as the inverse of measuring semantic relatedness: rather than using contexts to discover related words I would use words to discover related contexts. Once a mapping has been established, it should be possible to measure semantic relatedness between words in the two languages. Effectively this would be a system that learns semantic relatedness between any pair of languages, without needing a large parallel corpus, only a bilingual dictionary.

For evaluation, a set of translations could be held out and the cross-language semantic relatedness measure evaluated by how often it finds the correct translation of these held out words. This could also be evaluated on parallel corpus discovery or one of many applications that could take advantage of this line of research. For example, in Machine Translation large parallel corpora are needed to determine how a sequence of words is translated into another language. A cross-language semantic relatedness measure could find an approximate translation for a word even if that word were not present in the parallel corpus. It could also be used to help in constructing a parallel corpus by determining if pairs of sentences in two different languages are expressing the same thing. Another application would be cross-language information retrieval, where one has a query written in one language and must find relevant documents written in many languages. Constructing multi-lingual thesauri is another task where a tool like this would be very useful. Also electronic tutoring systems designed to teach students a foreign language could take advantage of this sort of research.

Chapter 5

Adding Words to *Roget's Thesaurus*

In this chapter I explain how I go about adding new words to *Roget's Thesaurus*. In Chapter 4 I evaluated a variety of MSRs, but these measures alone do not tell me exactly where to put a word in *Roget's*. I will examine my methods on a sample set of held out words. Once I have tuned the best method I use it to add a large set of new words to *Roget's Thesaurus*. These word placements will be evaluated manually. I used the MSRs 1911-combine and 1987-combine for updating the 1911 and 1987 *Roget's Thesaurus* respectively. These measures are described in Chapter 4. For nouns and verbs, supervised context-level re-weighting is applied and the unsupervised re-weighting is done on top of that, in both cases using PMI. For adjectives the same process is applied, but this time using learning at the relation level before applying unsupervised re-weighting. Once again PMI is used for both. These measures were selected because they performed best when identifying words in the same POS and Paragraph. In terms of identifying words in the same SG, it was rarely any better or worse than using just the unsupervised PMI measure.

I have shown how MSRs can be used to determine if two words appear close to each other in *Roget's*, but this does not quite tell where to put a new word in *Roget's*. Currently the system indicates that a target word t is related to several words $w_1, ..., w_n$. The problem is, how many of these words should be included when making a decision. Should the neighbouring words be selected based on their relatedness scores or is rank preferable? The words $w_1, ..., w_n$ that are related to my target t could appear in a variety of different *Roget's groupings*. The possible set of *Roget's groupings* that t can be placed into is the set of *Roget's groupings* in which an appearances of $w_1, ..., w_n$ can be found. A *Roget's grouping* will only be considered if it contains w_i on its own and not as part of a phrase. I use only single words to place t in *Roget's* as single words are what was used in the matrices in Chapter 4. For evaluation I will consider a word t to be correctly added to a grouping if that grouping contains t either on its own or as part of a phrase.

5.1 Placing Words Into *Roget's* Thesaurus

I will evaluate a variety of systems for adding new words to *Roget's Thesaurus*. The baseline method places a word into the same POS, Paragraph and SG as its closest neighbour in *Roget's*. I will try to improve on this using multiple words to deduce a better location or multiple locations.

I will use the test set from Chapter 4 as a tuning set and use the third dataset for a final evaluation. All these results will be presented in this section. These evaluations are actually lower bounds on how well each system works since it is possible to discover new senses of each word. Section 5.3 reports experiments manually evaluating newly added words.

5.1.1 The Process of Adding New Words

I exploit the hierarchy of *Roget's Thesaurus* to find the best place to put new words. In this process I identify first the POS then the Paragraph and Semicolon Group in which to put a new word. Identifying the POS will effectively give me the correct Head as well since part-of-speech is determined by the parser. I will refer to the new word to be added to *Roget's* as the target word, or t. Other words already found in *Roget's* may be referred to as anchors. These anchors are used to find a good location to place the target word t.

I experiment with three different methods and a baseline for adding words to *Roget's Thesaurus*. For a baseline the target t is placed in the same POS, Paragraph and SG as w_i where w_i is the first word in the list that is found in *Roget's Thesaurus*. Since w_i may be polysemous t could be placed into multiple locations in *Roget's*. Often w_i will be w_1 if the first neighbour of t is found in *Roget's*. In table 5.1 this baseline is calculated using the MSRs for the combined weighting for *Roget's* 1911, 1987 on their respective thesauri. The results show one number for the count of POSs, Paragraphs and SGs that the target t was placed into as well as the precision of placing the word into the POSs, Paragraphs and SGs.

The three different ways to add words to *Roget's Thesaurus* rely on creating a long list of related words, to help identify where to place a new word. The first method is to use a nearest neighbour model. In this method X nearest neighbours are identified for each target word and if W of these X words appear in the same *Roget's grouping*, then the target word is placed into that grouping. The problem with this method is that it uses the same length of list for every word. It may be that some words have greater or fewer related words than others so it might be preferable to adjust the length of the list in such a way. This leads me to the second method.

The second method is similar, but uses the scores rather than the rank. Words with scores of value Y or higher are identified and if W of these words are in the same *Roget's grouping* then the target word is placed into that grouping. Although this provides a way of having differing lengths of lists, it is possible that similarity scores are dependent on the target word and so different scores may mean different levels of similarity depending on what the target word is. A word that appears in many common contexts may cause all its neighbours to have uniformly inflated similarity scores. To remedy this I try a third method.

The third method considers the relative scores. In this method I assume that the first similar word w_1 is a good anchor, then take all synonyms within $Z\%$ of the similarity score for w_1. This means that if w_i has a score of within $Z\%$ of w_1 then it can be used as an anchor of t for determining the correct *Roget's grouping.* Once again if W of these words in the same *Roget's grouping* have a relative score of Z or higher then the target word can be placed into that grouping as well.

One problem is how to optimize these measures. Each method has two parameters to optimize, W and either X, Y or Z. One logical method would be to evaluate F-measure based on the precision with which words are placed into the thesaurus and recall of the number of words from the test set that could actually be placed into *Roget's.* A second possible recall method would be to identify the number of places where a word appears in *Roget's* and see how many of them the word was placed into, but this measure has some problems. For one, rare senses are not well represented by the vectors in the term-context matrix, so synonyms for only the most dominant senses will be found.

Another problem is that an even balance of precision and recall may still yield many inaccuracies. I assume, therefore, that identifying the POS must have a higher weight given to precision than recall. I set a 3 to 1 ratio of precision to recall. This means that F-measure is evaluated using a F0.33 measure, rather than the more traditional F1 measure. Once the POS has been identified, the Paragraph and Semicolon Group will be identified using the F1 measure.

		1987				1911			
	Data	Words	P	R	F0.33	Words	P	R	F0.33
Noun	*Tuning*	*1000*	*0.281*	*0.486*	*0.293*	*817*	*0.232*	*0.296*	*0.237*
	Testing	1000	0.295	0.487	0.307	840	0.267	0.344	0.273
Verb	*Tuning*	*600*	*0.204*	*0.468*	*0.216*	*542*	*0.167*	*0.271*	*0.174*
	Testing	600	0.245	0.455	0.257	538	0.196	0.297	0.203
Adj	*Tuning*	*600*	*0.250*	*0.460*	*0.262*	*489*	*0.246*	*0.288*	*0.249*
	Testing	600	0.232	0.435	0.244	497	0.201	0.262	0.206

Table 5.1: Baseline results, identifying the POS of a word on the tuning and testing data.

The choice of F0.33 is somewhat arbitrary, but favouring precision over recall should mostly bring advantages. A high-precision system is more likely to place words in the correct grouping, in theory, at the cost of low recall. However, any method of adding new words to *Roget's* could be run in multiple passes which can be used to make up for the lower recall. Rather than attempting to add a lot of words in one pass, my method will add a smaller quantity of words over multiple passes. The choice of the 3 to 1 ratio can likely be substituted with a similar ratio, maybe 2 to 1 or 4 to 1, but that will have to be left for future work.

When using this method to actually add new words, sometimes it will be necessary to add new Paragraphs or Semicolon Groups. If a POS is identified but no Paragraph, then a new Paragraph will be created. Likewise if a Paragraph can be identified but no Semicolon Group is selected, then the word will be placed in a new Semicolon Group in the selected Paragraph.

5.1.2 Baseline

The results for the baseline experiments are shown in Table 5.1. The results are measured for the 1911 *Roget's* and the 1987 version. The 1911 version did not contain all the words for evaluation that the 1987 version did thus accounting for the differences in word counts. The results for this baseline experiment show a small advantage adding words to the 1987 *Thesaurus* over the 1911 version.

	1987		1911	
	X	W-POS	X	W-POS
Noun	26	10	10	4
Verb	22	7	6	3
Adjective	19	6	8	3

Table 5.2: Optimal values for parameter X, the number of nearest neighbours.

		1987				1911			
	Data	Words	P	R	F0.33	Words	P	R	F0.33
Noun	*Tuning*	*1000*	*0.746*	*0.267*	*0.633*	*817*	*0.613*	*0.171*	*0.488*
	Testing	1000	0.758	0.262	0.637	840	0.659	0.182	0.522
Verb	*Tuning*	*600*	*0.565*	*0.285*	*0.514*	*542*	*0.484*	*0.131*	*0.381*
	Testing	600	0.536	0.252	0.482	538	0.471	0.097	0.340
Adj	*Tuning*	*600*	*0.658*	*0.273*	*0.577*	*489*	*0.571*	*0.184*	*0.472*
	Testing	600	0.590	0.233	0.512	497	0.503	0.141	0.400

Table 5.3: Precision, Recall and F0.33-measure when optimizing for X

5.1.3 Tuning Parameters for Adding New Words

The parameters, optimized for F0.33, for the three non-baseline methods are shown in Table 5.2, 5.4 & 5.6. The results on the tuning and testing data can be found in Tables 5.3, 5.5 & 5.7.

When optimizing for the X nearest neighbours – Tables 5.2 & 5.3 – the results show a large improvement over the baseline – Table 5.1. The results for nouns were actually better on the testing dataset than the tuning one, though somewhat worse for verbs and adjectives. As with the baseline the results were better for the 1987 *Roget's Thesaurus* than the 1911 version. Generally about one third to half of the words found in the top X needed to be present in the same *Roget's grouping* in order to accurately select the right grouping.

Optimizing word placements with scores Y or higher are shown in Tables 5.4 & 5.5. In this case, the optimal scores were noticeably lower than when I optimized for X nearest neighbours in Table 5.3. The minimum score Y appeared to be lower for nouns than for verbs or adjectives, though more words were required in order to positively identify

	1987		1911	
	Y	W-POS	Y	W-POS
Noun	.08	15	.07	14
Verb	.09	9	.13	2
Adjective	.13	3	.1	4

Table 5.4: Optimal values for parameter Y, the minimal relatedness score.

		1987				1911			
	Data	Words	P	R	F0.33	Words	P	R	F0.33
Noun	*Tuning*	*1000*	*0.596*	*0.182*	*0.486*	*817*	*0.420*	*0.120*	*0.336*
	Testing	1000	0.507	0.160	0.417	840	0.367	0.110	0.297
Verb	*Tuning*	*600*	*0.477*	*0.078*	*0.316*	*542*	*0.211*	*0.096*	*0.189*
	Testing	600	0.573	0.062	0.313	538	0.234	0.063	0.184
Adj	*Tuning*	*600*	*0.529*	*0.122*	*0.396*	*489*	*0.480*	*0.084*	*0.326*
	Testing	600	0.421	0.103	0.322	497	0.274	0.066	0.209

Table 5.5: Precision, Recall and F0.33-mesure when optimizing for Y

	1987		1911	
	Z	W-POS	Z	W-POS
Noun	.82	4	.93	2
Verb	.89	3	.98	2
Adjective	.82	3	.91	2

Table 5.6: Optimal values for parameter Z, the relative score.

		1987				1911			
	Data	Words	P	R	F0.33	Words	P	R	F0.33
Noun	*Tuning*	*1000*	*0.643*	*0.190*	*0.519*	*817*	*0.468*	*0.200*	*0.413*
	Testing	1000	0.595	0.215	0.506	840	0.542	0.219	0.473
Verb	*Tuning*	*600*	*0.468*	*0.147*	*0.384*	*542*	*0.438*	*0.118*	*0.344*
	Testing	600	0.492	0.163	0.410	538	0.389	0.091	0.293
Adj	*Tuning*	*600*	*0.512*	*0.215*	*0.450*	*489*	*0.478*	*0.145*	*0.389*
	Testing	600	0.463	0.200	0.409	497	0.434	0.129	0.351

Table 5.7: Precision, Recall and F0.33-mesure when optimizing for Z

the *Roget's grouping*. This method is not as successful as simply selecting the X nearest neighbours. The third method also relies on scores, but uses the relative difference between a neighbour word and the nearest synonym (i.e. the closest neighbour). For verbs added to the 1911 *Roget's* there was actually no improvement over the baseline shown in Table 5.1. This is the least successful method that I tried.

Optimizing for the relative score Z is shown in Tables 5.6 & 5.7. In this case I found that most neighbouring words had to be within 80-90% of the closest neighbour in terms of score. This noticeably improves the results over simply selecting a hard score cut-off as seen in Table 5.5. Nonetheless it was not superior to the simple X nearest neighbours approach of Table 5.3. It would appear that rank is in many cases more important a feature for determining relatedness than score is. With this in mind, I applied the nearest neighbour function using X to find the best parameters for identifying the POS, Paragraph and SG. The parameter W shown in Tables 5.2, 5.4 & 5.6 was for the POS level, I will have three versions, W-POS, W-Para and W-SG for the POS, Paragraph and Semicolon Group respectively.

Table 5.8 shows the optimal values of X and W-POS, W-Para and W-SG. The

	1987				1911			
	X	W-POS	W-Para	W-SG	X	W-POS	W-Para	W-SG
Noun	26	10	5	2	10	4	3	3
Verb	22	7	4	3	6	3	2	2
Adjective	19	6	4	2	8	3	2	2

Table 5.8: Optimal parameters for X and W at the POS, Paragraph and SG levels.

same value of X was used for identifying groupings at the POS, Paragraph and SG levels. There is a fair bit of variance in the different measures, though X was always less than 25. The values of W-POS, W-Para and W-SG decrease as the groupings become smaller. To identify the correct Semicolon Group only 2 or 3 words were used. For the 1911 *Roget's Thesaurus*, the same number of words were used to identify the Paragraph as the Semicolon Group. More words could be used to identify the POS for the 1987 *Thesaurus* than for the 1911 version.

The precision, recall and F1 measure at the POS, Paragraph and SG level are shown in Tables 5.9 and 5.10 for the 1987 and 1911 *Thesaurus* respectively. These results show clearly that the F1 measure is highest when identifying the Paragraph level; this is largely because the POS level is optimized for the F0.33 measure. Once again the scores for the 1987 version tended to be better than those of the 1911 version. Most of the time it is possible to identify the correct POS with at least 40% accuracy or higher. The recall for the 1987 thesaurus was 0.233 or higher at the POS level. This is important because it indicates how many new words can be expected to be added to the *Thesaurus*. For the 1911 *Thesaurus* the results tended to be much lower, with scores from 0.097 to 0.182 on the test set. This number for verbs is very low, though for nouns and adjectives it is not nearly as bad, though still lower than the same results for the 1987 thesaurus.

5.2 Adding Words to the *Thesaurus*

In this section I apply the method described in Section 5.1.3 to adding new words to *Roget's Thesaurus*. When using this methodology, in practice, a few small modifications are needed. These modifications come from observations made about the *Thesaurus* in Chapter 2. First of all, I will only allow a word to be placed into a POS if it is not already present in either that POS or in another POS within the same Head Group. This reduces

			1987		
	Data	RG	P	R	F1
Noun	*Tuning*	*POS*	*306/410 (0.746)*	*267/1000 (0.267)*	*0.393*
	Tuning	*Para*	*225/402 (0.56)*	*189/267 (0.708)*	*0.625*
	Tuning	*SG*	*104/664 (0.157)*	*92/189 (0.487)*	*0.237*
	Testing	POS	304/401 (0.758)	262/1000 (0.262)	0.389
	Testing	Para	234/416 (0.562)	196/262 (0.748)	0.642
	Testing	SG	101/659 (0.153)	93/196 (0.474)	0.232
Verb	*Tuning*	*POS*	*227/402 (0.565)*	*171/600 (0.285)*	*0.379*
	Tuning	*Para*	*186/413 (0.45)*	*137/171 (0.801)*	*0.577*
	Tuning	*SG*	*34/129 (0.264)*	*32/137 (0.234)*	*0.248*
	Testing	POS	185/345 (0.536)	151/600 (0.252)	0.343
	Testing	Para	148/339 (0.437)	114/151 (0.755)	0.553
	Testing	SG	18/103 (0.175)	17/114 (0.149)	0.161
Adj	*Tuning*	*POS*	*227/345 (0.658)*	*164/600 (0.273)*	*0.386*
	Tuning	*Para*	*182/312 (0.583)*	*136/164 (0.829)*	*0.685*
	Tuning	*SG*	*75/381 (0.197)*	*63/136 (0.463)*	*0.276*
	Testing	POS	193/327 (0.59)	140/600 (0.233)	0.334
	Testing	Para	152/294 (0.517)	116/140 (0.829)	0.637
	Testing	SG	59/351 (0.168)	51/116 (0.440)	0.243

Table 5.9: Identifying best POS, Paragraph and Semicolon Group using optimized values for X and $W - POS$, $W - Para$ & $W - SG$. Using the F1 measure for evaluation on *Roget's* 1987.

			1911		
	Data	RG	P	R	F1
Noun	*Tuning*	*POS*	*157/256 (0.613)*	*140/817 (0.171)*	*0.268*
	Tuning	*Para*	*89/163 (0.546)*	*83/140 (0.593)*	*0.568*
	Tuning	*SG*	*31/62 (0.500)*	*29/83 (0.349)*	*0.411*
	Testing	POS	162/246 (0.659)	153/840 (0.182)	0.285
	Testing	Para	83/155 (0.535)	78/153 (0.510)	0.522
	Testing	SG	29/55 (0.527)	28/78 (0.359)	0.427
Verb	*Tuning*	*POS*	*76/157 (0.484)*	*71/542 (0.131)*	*0.206*
	Tuning	*Para*	*55/136 (0.404)*	*53/71 (0.746)*	*0.525*
	Tuning	*SG*	*24/86 (0.279)*	*24/53 (0.453)*	*0.345*
	Testing	POS	57/121 (0.471)	52/538 (0.097)	0.160
	Testing	Para	39/112 (0.348)	35/52 (0.673)	0.459
	Testing	SG	22/76 (0.289)	19/35 (0.543)	0.378
Adj	*Tuning*	*POS*	*109/191 (0.571)*	*90/489 (0.184)*	*0.278*
	Tuning	*Para*	*80/188 (0.426)*	*71/90 (0.789)*	*0.553*
	Tuning	*SG*	*23/107 (0.215)*	*22/71 (0.310)*	*0.254*
	Testing	POS	79/157 (0.503)	70/497 (0.141)	0.220
	Testing	Para	46/148 (0.311)	42/70 (0.600)	0.409
	Testing	SG	14/91 (0.154)	13/42 (0.310)	0.206

Table 5.10: Identifying best POS, Paragraph and Semicolon Group using optimized values for X and $W - POS$, $W - Para$ & $W - SG$. Using the F1 measure for evaluation on *Roget's* 1911.

the possibility of antonyms, which may be distributionally similar, being entered into the same POS. Within each POS I allow a word to be placed only into one Paragraph. I also do not allow for the same word to be added to multiple Semicolon Groups within the same Paragraph. In *Roget's Thesaurus* the same word can only appear twice in the same Semicolon Group if it is part of two different short phrases. At the Paragraph level it is extremely rare that the same word will appear twice. I also do not allow the same word to be added to two different Paragraphs in the same POS, though this does on rare occasion happen in *Roget's*.

The process that I have outlined can actually be applied iteratively. Once a new word has been added to the resource it can be used to help add even more words to *Roget's*. This is essentially a bootstrapping process and can be repeated several times. I will create two updated versions of each *Thesaurus*, one where only one pass is used to update the *Thesaurus* and one where five passes are used for updating *Roget's*. It will be interesting to see how many words can be added and how the different passes affect the quality of the updates.

To add new words to *Roget's* I consider each word in each matrix to be a target and then generate a list of the 100 nearest neighbours for each of these words.[1] Immediately I found a problem with this. Many of the most common words in each list tended to be unwanted words. For example: "he", "it", "his" and "one" were the top four most frequent words in the noun matrix. I decided to remove all stop words from these lists.[2] It was from these lists that I attempted to add new words to *Roget's*.

There are a number of different measures that are of interest when adding new words to the *Thesaurus*. For example, the number of times a target word t has sufficient X and W values to be placed in *Roget's*, regardless of whether it was already present. The second measure is the number of total words added to *Roget's Thesaurus*. The third measure is the number of unique words added to *Roget's Thesaurus*. This measure is likely to be similar to the number of total words added since most often a target word t is only added to a single location in *Roget's*. The last measure is the number of Heads that a new word was added to. It would be nice if words could be added to almost every Head, but this is not realistic. Many Heads contain a lot of short phrases and few individual words, making it difficult to add new words to these Heads. In addition, some

[1] Only the top X of these 100 nearest neighbours were used in identifying the best place to put a new word.

[2] I used a 980-element union of five stop lists, first used in Jarmasz (2003): Oracle 8 ConText; SMART; Hyperwave; lists from the University of Kansas and Ohio State University.

Heads may not contain any instances of a given part-of-speech, or in some cases only a few words. It is also possible that some Heads contain terms that are very broad in the meaning of its words. For example the Head for "Existence" in the 1911 *Thesaurus* is very broad in meaning. As a result it is impossible to add new words to every Head. The results can be seen in Table 5.11 for all five passes.

In addition to the five passes of new words added I also attempt a similar experiment adding random words. In this case I used exactly the same parameters when updating the versions of *Roget's* only when it came time to place a new word into *Roget's* the target word was replaced by a randomly selected word. The counts of total and unique words added, etc. can be seen in Table 5.12. The new word is substituted after a location is chosen but before it is checked to see whether the target word is actually found in that *Roget's grouping* or not. As a result the number of attempted placements is very close to the total number of words added, much closer than for the counts from Table 5.11.

Ultimately three updated version of the 1987 and three updated versions of the 1911 *Roget's Thesaurus* were created. These updated versions of *Roget's* will be referred to as *Roget's Thesaurus* 1911X1, 1911X5, 1911XR, 1987X1, 1987X5 and 1987R to indicate the year and the number of passes used to expand the *Thesaurus*. These new thesauri will be evaluated manually in Section 5.3 and through various Natural Language Processing applications in Chapter 6.

Other interesting statistics to consider are the total number of words, Paragraphs and Semicolon Groups added to each version of *Roget's*. Table 5.13 shows these statistics. Ultimately, for the 1911 *Roget's* up to 5,500 new words were added to 1911X5, while almost 9,600 were added to 1987X5. When adding words to the 1911 *Roget's*, approximately two-thirds of the new words were placed in a new SG, while about a quarter were added to a new Paragraph. For the 1987 *Roget's*, a little under half of the new words were placed in new SGs, while around one-fifth were added to new Paragraphs.

5.3 Manual Evaluation

To really verify the quality of the additions, I perform a manual evaluation. I considered a variety of tests before selecting one.

The first method I considered was to see how well a human can identify newly added words. In this test a person would be given a set of Paragraphs from *Roget's* and asked to identify which words they think were added automatically and which were originally part of the *Thesaurus*. The percentage of time when a person correctly identifies newly

Pass	Year	Part -of- Speech	Attempted Placements	Total Words Added	Unique Words Added	Mostly New Words	Completely New Words	Heads Affected
1	1987	Nouns	6755	1510	1414	175	98	206
		Verbs	2870	893	735	52	45	129
		Adj	3053	858	713	15	10	183
	1911	Nouns	3888	1259	1193	148	68	274
		Verbs	1069	407	378	22	19	133
		Adj	1430	539	480	18	16	198
2	1987	Nouns	8388	774	742	37	14	139
		Verbs	4335	747	653	23	16	92
		Adj	4412	612	549	4	4	114
	1911	Nouns	5315	762	719	65	13	164
		Verbs	1530	247	238	14	14	71
		Adj	2083	287	262	6	5	95
3	1987	Nouns	9213	499	478	16	6	88
		Verbs	5303	600	543	16	14	61
		Adj	5275	532	463	7	2	80
	1911	Nouns	6109	549	520	35	11	100
		Verbs	1761	147	142	6	6	36
		Adj	2393	205	191	5	4	57
4	1987	Nouns	9767	384	378	11	2	60
		Verbs	6068	523	496	11	9	49
		Adj	5926	451	404	6	6	55
	1911	Nouns	6652	417	395	20	5	76
		Verbs	1898	106	105	0	0	21
		Adj	2571	139	129	1	0	35
5	1987	Nouns	10210	330	324	12	2	49
		Verbs	6689	464	422	6	3	39
		Adj	6509	424	382	3	1	38
	1911	Nouns	7026	295	288	22	10	54
		Verbs	1979	76	74	0	0	14
		Adj	2710	119	115	1	0	22

Table 5.11: New words added after the 1^{st}, 2^{nd}, 3^{rd}, 4^{th} & 5^{th} passes.

Pass	Year	Part -of- Speech	Attempted Placements	Total Words Added	Unique Words Added	Mostly New Words	Completely New Words	Heads Affected
1	1987	Nouns	6755	6189	5007	3923	3593	306
		Verbs	2870	2238	1366	734	715	186
		Adj	3053	2631	1670	1547	1488	278
	1911	Nouns	3888	3718	3203	2736	2554	379
		Verbs	1069	946	759	468	465	195
		Adj	1430	1349	1051	952	926	276

Table 5.12: Random words added after one iteration.

Resource	New Paragraphs	New SGs	New Words
1911X1	633	1442	2209
1911X5	1851	3864	5566
1911R	1477	3803	6018
1987X1	653	1356	3261
1987X5	2063	4466	9601
1987R	1672	3731	11058

Table 5.13: New Paragraphs, SGs and words added to the updated versions of *Roget's Thesaurus*.

added words can be used to evaluate the additions. If a person is as likely to pick a word previously found in the *Thesaurus* as one newly added to the *Thesaurus*, then the additions would be indistinguishable from those already in *Roget's*. That would be an ideal outcome. This has a drawback that annotators may be more likely to select words whose meaning they do not know. This could be particularly true when evaluating on *Roget's* 1911, where there are many outdated words. Even the 1987 version has many words that are infrequently used today.

The second method of manual evaluation I considered was to show the annotator a newly added word and then ask them to assign the word to the correct location in the *Thesaurus*. An edit distance score could be assigned based on how many steps the new word is from its correct location. Moving the word to a different Semicolon Group in the same Paragraph would be 1 step, moving to a Semicolon Group in a different Paragraph would be 2 steps, etc. I would also measure how frequently the annotator needed to create a new Paragraph, or Semicolon Group for the word. Conversely, one could also measure the number of Semicolon Groups and Paragraphs that were automatically created but should not have been. Since effectively I would ask an annotator to place a word into the correct location in the *Thesaurus*, it would be preferable to limit moving a word to other locations within the same Head. If the word does not belong in the assigned Head, then it can be labelled as such instead of asking the annotator to place the word somewhere else in the *Thesaurus*, which would be quite difficult and time-consuming. Even so an annotator would need to read an entire Head before making a decision on how far off a word is from its correct location. This would be extremely difficult because larger heads, where most new words are added, could contain thousands of words. Identifying whether there is a SG more appropriate for a given word could take a fair bit of effort and it might be impossible to ask annotators to annotate enough to perform a meaningful evaluation. This evaluation would require a professional lexicographer.

The third strategy, the one I finally adopted, combines elements of both of the preceding methods. There are two parts to this annotation exercise. The first is to identify whether new words added to an existing SG or a new SG in an existing Paragraph are in the correct location. The annotator will be given the name of the Head, the part-of-speech and also the text of the Paragraph where the word has been added. The new term will be highlighted in red and underlined, while the other terms in the same SG will be in bold. The annotator will then be asked to decide whether the new word is in the correct SG, wrong SG but in the correct Paragraph, wrong Paragraph but in the correct Head or incorrect Head. The instructions given to the annotators, along with the

		Nouns	Verbs	Adjectives
1911X1	Word	755 (255)	336 (179)	485 (215)
	Paragraph	504 (218)	71 (60)	58 (51)
1911X5	Word	1740 (315)	804 (260)	1171 (289)
	Paragraph	1542 (308)	179 (122)	130 (97)

Table 5.14: Number of new words added to existing and new Paragraphs along with the number of samples selected.

results, can be found in Appendix C, Section C.1.

The second evaluation exercise is to determine whether a new word added to a new Paragraph is in the correct Head. As context, I provide the first word in every Paragraph in the same POS. In this case, it is too difficult to actually ask an annotator to determine which SG or Paragraph a new word would belong in, because some POSs are extremely large, containing thousands of words. Instead, I only ask whether the word is in the correct Head.

I only evaluate the additions to the 1911 *Roget's Thesaurus*, not the 1987 version. To limit the size of the Paragraphs I allow for no more than 250 characters, thus limiting the number of words that the annotators need look at. Evaluation is done on new words added to existing Paragraphs and new words added to new Paragraphs. This evaluation was completed by me and four volunteers. I chose enough samples to guarantee a 5% confidence interval at a 95% confidence level.[3] The total number of new words added to existing Paragraphs and new Paragraphs, along with the number of samples, appears in Table 5.14. I also included a high and low baseline, words already in *Roget's* and words randomly added to *Roget's*. There are enough samples from these baselines to guarantee a 5% confidence interval at a 95% confidence level if the samples from all three parts of speech are combined, though individually the confidence interval is greater than 5%.

Every new word in 1911X1 appears in 1911X5 because such a percentage of the samples needed to evaluate 1911X5 can be selected from the samples used to evaluate 1911X1. This means that I only need to evaluate a selection of the words from the 1911X5 thesauri not present in the 1911X1 edition. I randomly select words from the sample set for 1911X1 to make up the rest of the samples for the 1911X5 evaluation. Table 5.15 shows counts of how many words must be selected from passes 2-5 in order

[3] http://www.macorr.com/sample-size-calculator.htm

		Nouns	Verbs	Adjectives
1911X5	Word	1740 (179)	804 (152)	1171 (170)
reduced	Paragraph	1542 (213)	179 (74)	130 (54)

Table 5.15: Number of samples from 1911X5 added in passes 2-5.

to guarantee a 5% confidence interval at a 95% confidence level.

Random selection was made from each annotator's dataset: 40 tests for adding words to existing Paragraphs and 40 tests for adding words to new Paragraphs. These data points were added to all annotator's test set so that there would be an overlap of 200 samples for each experiment, on which to calculate inter-annotator agreement. I round up the number of samples needed to be divisible by five, as I will have five annotators – four volunteers and myself – for these experiments. Altogether each annotation task consists of 999 items, 547 tests adding words to existing Paragraphs and 452 tests adding words to new Paragraphs. In addition, each annotator was given 390 for each of known positive and negative examples in existing and new Paragraphs. The positive examples are words already present in *Roget's Thesaurus*, while the negative examples are words randomly placed in the *Thesaurus*.

5.3.1 Manual Annotation Results

The combined manual annotation results can be seen in Tables 5.16 and 5.17, for new words added to existing Paragraphs and new Paragraphs respectively. The results for each individual annotator can be found in Appendix C, Section C.2. Since I was one of the annotators, I will show in Appendix C the same results with my annotations excluded – see Section C.2, Tables C.13 and C.14. Generally I found that the results did not differ very much whether my annotations were included or excluded, and so I will only discuss results where all annotations are included. When only four annotators are considered, the confidence interval moves from 5% to the range between 5.7% and 8.1%. Most of the confidence intervals are quite close to 6% except when adding verbs and adjectives to new Paragraphs, where the confidence interval can range from 6.5% to 8.1%. The high and low baselines are labeled as "Positive" and "Negative" in these tables; a count and the proportion of results receiving each score are recorded.

A number of interesting observations can be taken from Table 5.16 where the annotators were evaluating words in an existing Paragraph. These results are summarized in

		Right	Right	Right	Wrong	
Task	POS	SG	Para	Head	Head	N/A
Positive	noun	117 (*0.6*)	20 (*0.103*)	22 (*0.113*)	21 (*0.108*)	15 (*0.077*)
	verb	59 (*0.562*)	14 (*0.133*)	10 (*0.095*)	16 (*0.152*)	6 (*0.057*)
	adjective	55 (*0.611*)	16 (*0.178*)	6 (*0.067*)	7 (*0.078*)	6 (*0.067*)
Negative	noun	6 (*0.031*)	2 (*0.010*)	20 (*0.103*)	144 (*0.738*)	23 (*0.118*)
	verb	9 (*0.086*)	2 (*0.019*)	18 (*0.171*)	73 (*0.695*)	3 (*0.029*)
	adjective	3 (*0.033*)	4 (*0.044*)	8 (*0.089*)	71 (*0.789*)	4 (*0.044*)
1911X1	noun	159 (*0.624*)	52 (*0.204*)	22 (*0.086*)	19 (*0.075*)	3 (*0.012*)
	verb	92 (*0.511*)	37 (*0.206*)	24 (*0.133*)	24 (*0.133*)	3 (*0.017*)
	adjective	135 (*0.628*)	44 (*0.205*)	17 (*0.079*)	17 (*0.079*)	2 (*0.009*)
1911X5	noun	181 (*0.576*)	59 (*0.188*)	44 (*0.140*)	25 (*0.080*)	5 (*0.016*)
	verb	107 (*0.412*)	45 (*0.173*)	53 (*0.204*)	52 (*0.200*)	3 (*0.012*)
	adjective	147 (*0.507*)	52 (*0.179*)	32 (*0.110*)	56 (*0.193*)	3 (*0.010*)

Table 5.16: Results of the Manual Evaluation for words added to existing Paragraphs.

Figure 5.1. In the case of positive examples one can see that around 60% of the time the annotators were able to correctly identify when a word belonged in the SG in which it was found. In all approximately 80-90% of the time the annotators agreed that the word was in the correct Head. One possible reason why annotators would believe the words belonged in different SGs, Paragraphs, etc. is that many of the words were difficult to understand. A high number of words that could not be labeled by the annotators fell into the Positive category. For the randomly assigned words the annotators tended to correctly identify that the words did not belong in that Head 70-80% of the time. For nouns there were a very large number of cases that the annotators could not answer. It would appear that words present in the *Thesaurus*, and those randomly added, are harder to determine the meaning of than those that were added using my methodology.

In terms of the quality of additions, for 1911X1, the distribution of scores in Table 5.16 is actually very close to that of the distribution for words already present in the *Thesaurus*. This would suggest that after one pass the words being added are nearly indistinguishable from those already in *Roget's*. This is very good news, as it confirms that my process of updating the lexicon has been successful. When looking at the *Thesaurus* updated with 5 passes, 1911X5 the distribution of scores suggests the additions were not as reliable. The scores are worse than for *Roget's* 1911X1, but still much closer

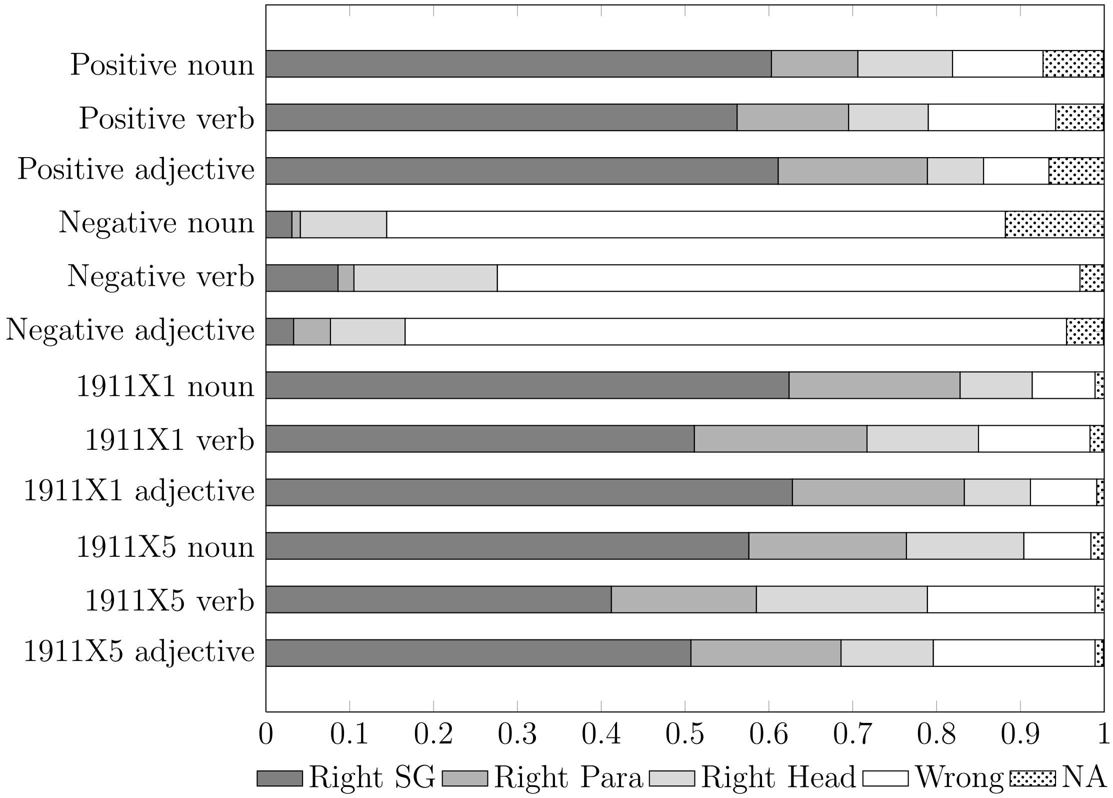

Figure 5.1: Evaluation on words added to an existing Paragraph in *Roget's* 1911.

Task	POS	Right Head	Wrong Head	N/A
Positive	noun	158 *(0.810)*	33 *(0.169)*	4 *(0.021)*
	verb	87 *(0.829)*	17 *(0.162)*	1 *(0.010)*
	adjective	75 *(0.833)*	14 *(0.156)*	1 *(0.011)*
Negative	noun	18 *(0.092)*	151 *(0.774)*	26 *(0.133)*
	verb	17 *(0.162)*	83 *(0.790)*	5 *(0.048)*
	adjective	13 *(0.144)*	74 *(0.822)*	3 *(0.033)*
1911X1	noun	189 *(0.859)*	27 *(0.123)*	4 *(0.018)*
	verb	50 *(0.833)*	10 *(0.167)*	0 *(0.000)*
	adjective	48 *(0.873)*	7 *(0.127)*	0 *(0.000)*
1911X5	noun	207 *(0.674)*	94 *(0.306)*	6 *(0.020)*
	verb	64 *(0.533)*	55 *(0.458)*	1 *(0.008)*
	adjective	61 *(0.616)*	37 *(0.374)*	1 *(0.010)*

Table 5.17: Results of the Manual Evaluation for words added to new Paragraphs.

to the positive baseline than the negative baseline. Multiple passes seem to increase the amount of error, but not by a large amount.

The results are a bit different when it comes to adding new words to new Paragraphs. These results are summarized in Figure 5.2. Once again the high and low baselines appeared to be fairly easy problems for the annotators, usually getting around 80% of the questions right. Also, a solid majority of the unknown words appeared in these two groups. The additions to 1911X1 also showed high scores, comparable to the high baseline, sometimes even exceeding it slightly. It may be that for the high baseline there were many words where the annotator was not aware of the sense being used and so mistakenly labeled it as incorrect.

This time when updating with 5 passes, the 1911X5 results clearly fall a fair distance from the scores for 1911X1. It would appear that multiple passes are adding considerable error to the *Thesaurus*, when these words are placed into new Paragraphs. This is in stark contrast to the results for adding words to existing Paragraphs, where the drop in scores between 1911X1 and 1911X5 was relatively small.

As noted earlier, the results when my annotations are included are not radically different from when they are excluded. There are a few differences though, which I will note here. In terms of placing words into existing Paragraphs the numbers are very close. The main difference comes when identifying negative examples (randomly added

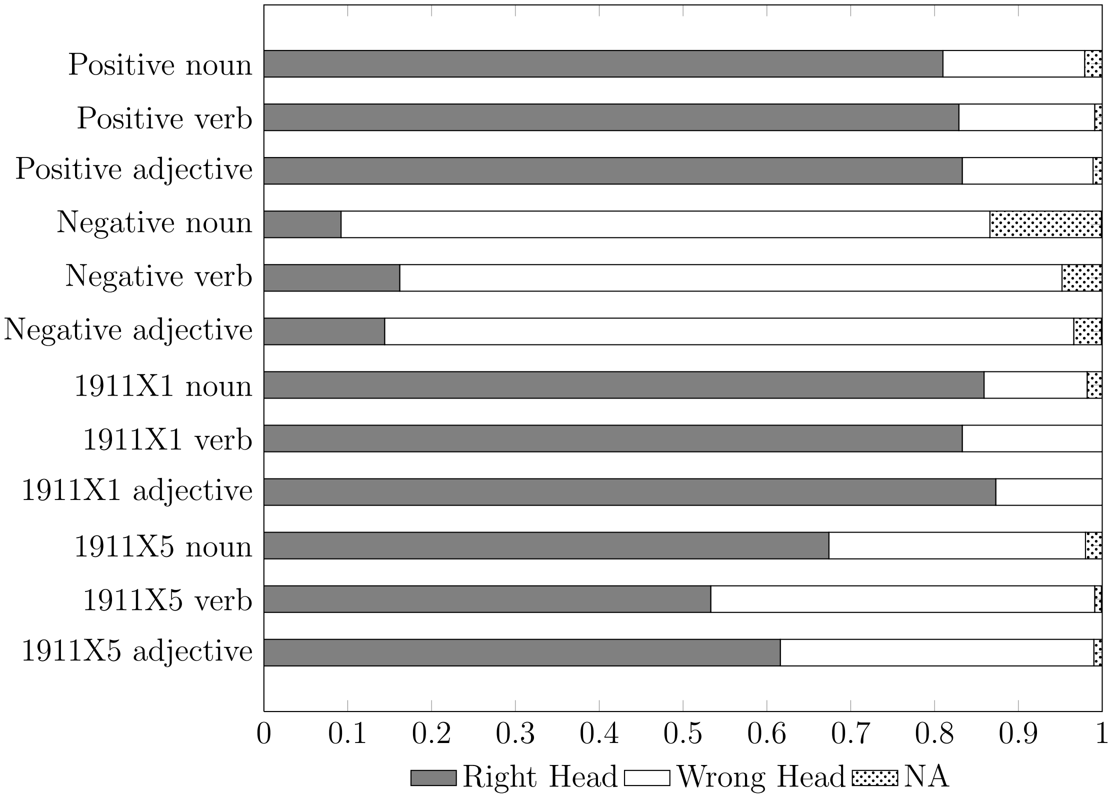

Figure 5.2: Evaluation on words added to new Paragraph in *Roget's* 1911.

words). The number of words correctly identified as incorrect increased when I was an annotator. In terms of adding words to new Paragraphs the biggest difference is that scores for words added after 5 passes were lower when I was not an annotator. All this would suggest that I had a somewhat easier time identifying negative examples, while I was slightly more lenient in approving new words in new Paragraphs.

5.3.2 Inter-Annotator Agreement

Each annotator was given a set of 200 examples that overlapped between their annotations. These overlapping sets were used to measure inter-annotator agreement. There are a number of criteria to consider when choosing an inter-annotator agreement measure. Firstly, I need a measure that will work for multiple annotators. Secondly, there can be missing data, as the annotators were instructed to leave difficult questions blank. Thirdly, the scores that the annotators give come from an ordered set representing words in the same SG, Paragraph, Head, or none of the above. All this lends itself naturally to a measure called Krippendorff's α (Krippendorff, 1980, 2004). Krippendorff's α is designed to work with a variety of kinds of data including nominal, ordinal and interval annotations. For my experiments I used ordinal.

Krippendorff's α was calculated for both experiments: adding words to existing Paragraphs and adding words to new Paragraphs. When adding words to an existing Paragraph I found a score of $\alpha = 0.340$, while when adding words to new Paragraphs the score was $\alpha = 0.358$. These scores are often considered a "fair" amount of agreement (Landis and Koch, 1977).

5.4 Conclusion

I have shown that it is possible to place words into the POS and Paragraph levels of *Roget's Thesaurus* with a high precision. Although it is more difficult to add new words at the Semicolon Group level, it was still relatively successful.

Using the methods discussed in this chapter, I have created six new versions of *Roget's Thesaurus*, three from 1987 and three from 1911. One of these versions is updated randomly, one is updated with one pass and another with five passes. In creating these updated versions of *Roget's*, I have proposed and tested three different systems for placing words in the *Thesaurus*. One of these is based on rank, one is based on score and one on a relative score. In all, I found that rank was easily the best method. Although it may be surprising that a rank-based method outperforms the score, there have been some other experiments that notice a similar phenomena. For example Broda et al. (2009) found that using the rank of a feature can actually improve over using the value of that feature in a MSR.

From the manual annotation experiments, my findings were that adding new words to existing Paragraphs could be done quite successfully. When it came to creating new Paragraphs, the results after the first pass showed great promise, but after 5 passes the results started to degrade.

In this chapter I evaluate my method of adding new words to *Roget's Thesaurus*, both automatically and through manual evaluation In Chapter 6 I will perform a task-based evaluation. The new and old thesauri will be evaluated on a variety of NLP tasks.

5.4.1 Future Work

Perhaps other more complex methods of adding new words to *Roget's Thesaurus* can be considered. For example mixing rank and score might be possible in order to create an even more accurate method of adding new words. It might be worth considering whether optimizing for F0.33 as a means of adding new words to the POS level of *Roget's Thesaurus* is truly best. Likewise other methods for identifying where in *Roget's* to place a word could be considered. In particular the method of Pantel (2005) could potentially be modified to work for *Roget's Thesaurus*.

Another problem I do not tackle is that of adding cross-references. If the same word appear in two places in *Roget's* then they often contain a cross-reference linking them together as one sense of the word. Since I do not deal with word sense disambiguation/discrimination, this work could be a considerable undertaking.

Another possible evaluation technique would be to use the 1987 *Thesaurus* as a source of new words that could be placed into the 1911 version. Although such a resource could not be released publicly (due to copyright difficulties), it might be interesting to see just how close the 1911 version comes to resemble the 1987 version. One issue with this sort of evaluation is that mapping between the two resources is not entirely trivial. The 1911 *Thesaurus* is not strictly a subset of the 1987 version and it actually contains more Heads. This said, because the 1911 version tends to contain either outdated terms, or those already present in the 1987 version, such a resource would have little to offer over the 1987 version.

The manual annotation has only been conducted on the 1911 version of *Roget's The-*

saurus. This was done because it was the only version that could actually be released to the public, and the annotation experiment was very expensive in terms of time. The updates to the 1987 version could be evaluated similarly. I expect that, since the 1911 version is both older and smaller, that the updates to the 1987 version should actually be more accurate. This would be in line with the automatic evaluation from Section 5.1, but it is yet to be proven manually.

Adding New Words to ***WordNet***

It should also be possible to adapt my methods of placing words in *Roget's* to work for *WordNet*. Instead of identifying words in the same POS, then Paragraph, then SG, groupings of words could be created from the hypernym hierarchy. I can see two possible ways of doing it. The first would be to pick a relatively high level within the hierarchy and classify each word into one or more of these synsets (much as I did with the POS level). A synset could be represented by all the words in the transitive closure of its hyponyms. Next, propagate the word down the hierarchy, as I do with the Paragraph and SG, but this time until it can go no farther, and then add it to the synset there.

One problem here is that this could not be applied to adjectives and also only takes a single kind of relationship into account when placing a word. Another option might be to create a neighbourhood of words for each synset, based on a variety of relations. A word could then be placed into a larger grouping of multiple synsets before it is determined which synset in particular it belongs to. If no synset in particular can be picked, then a new synset can be created with some sort of ambiguous link joining it to the other synsets in its neighbourhood. These neighbourhoods of words could have some degree of overlapping terms. In general it should be possible to find a cluster of synsets to which a new word likely belongs. That said, *WordNet* uses explicit semantic relations, so it will become necessary to find some way of labelling these, in cases where a word must be added to a new synset.

Adding Domains-Specific Words

Another direction to take this kind of research is to see how it will work with words of a particular domain. Most of the words in *Roget's Thesaurus* tend to be from everyday English, as opposed to, say, medical terms. The nearest synonyms of theses everyday words will be other everyday words, which could make it more difficult to actually add domain-specific terms to *Roget's Thesaurus*. That said, the trainable MSR described in Chapter 4 could be built using words of a particular domain. If domain-specific and non-domain-specific words could be grouped together as near synonyms, then this could be used to train a MSR that could be used to add domain-specific terms to *Roget's*.

Similar to adding domain-specific words is the challenge of adding extremely new words to *Roget's Thesaurus*. Very new words may not have close synonyms in *Roget's*, which is why I add words over multiple passes. It would be interesting to investigate how many passes are required before the word "iPhone" is added to the *Thesaurus*. For "iPhone" to be successfully added closely related phrases like "mobile phone" or "smart phone" would need to already appear in the *Thesaurus*. Other words like "cellular network", "texting", "Apple" or "twitter" would also be useful in choosing where to place a word like "iPhone". Examining how well my method works on domain-specific terms and extremely new terms will have to be left for future work.

Chapter 6

Evaluating the Resource

In this chapter I examine how the various versions of *Roget's Thesaurus* as well as *WordNet* 3.0 perform on several NLP applications. The problems selected are designed to evaluate *Roget's* on a diverse cross-section of NLP tasks. These tasks include semantic relatedness, synonym identification, sentence relatedness, analogy solving, pseudo word sense disambiguation and text summarization.

I make use of *WordNet* 3.0 and also *Roget's Thesaurus* 1911, 1911X1, 1911X5, 1911R, 1987, 1987X1, 1987X5 and 1987R – See Chapter 5. Although the updated versions of *Roget's Thesaurus* are larger than the original and new words have been added with relatively high accuracy, it does not guarantee that they will result in higher scores on any one application. Many of these applications only use very common words and so might not benefit too much from an expanded thesaurus. The purpose of these experiments is twofold: (1) to evaluate updates to *Roget's Thesaurus*, (2) to compare *WordNet* and *Roget's* from a practical NLP perspective.

6.1 *SemDist*: Word Relatedness

Relatedness can be measured by the closeness of the words or phrases in the structure of a thesaurus. A MSR can be constructed by counting the edge distance between pairs of words in *Roget's Thesaurus* (Jarmasz and Szpakowicz, 2004). Two terms which are the same word score 18, terms in the same Semicolon Group score 16, in the same Paragraph – 14, and so on. The score is 0 if the terms appear in different classes, or if either is missing from *Roget's*. Pairs of terms get higher scores for being closer together. I will treat each word as sets of its senses. When there are multiple senses of two terms A

and B, I want to select senses $a \in A$ and $b \in B$ that maximize the relatedness score. Morphological variations on the words A and B are also considered. I define a distance function:

$$semDist(A, B) = \max_{a \in A, b \in B} [2 * depth(lca(a, b))]$$

lca is the *lowest common ancestor* and *depth* is the depth in the *Roget's* hierarchy; a Class has depth 0, Section 1, ..., Semicolon Group 8, Word 9. If one thinks of the function as counting edges between concepts in the *Roget's* hierarchy, then it could also be written as:

$$semDist(A, B) = \max_{a \in A, b \in B} [18 - edgesBetween(a, b)]$$

The two original versions of *Roget's Thesaurus* and the six updated versions were compared with *WordNet* 3.0 on the three data sets containing pairs of words with manually assigned similarity scores: 30 pairs (Miller and Charles, 1991), 65 pairs (Rubenstein and Goodenough, 1965) and 353 pairs[1] (Finkelstein et al., 2001). Word pairs can be of any part-of-speech. I measure the correlation with Pearson's correlation and with Spearman's correlation.

I compare the results for the various versions of *Roget's Thesaurus* with a variety of *WordNet*-based semantic relatedness measures – see Table 6.1 & 6.2 – with both Pearson and Spearman correlation values. This table shows 10 measures including J&C (Jiang and Conrath, 1997), Res (Resnik, 1995), Lin (Lin, 1998a), W&P (Wu and Palmer, 1994), L&C (Leacock and Chodorow, 1998), H&SO (Hirst and St-Onge, 1998), Path (counts edges between synsets), Lesk (Banerjee and Pedersen, 2002), and finally Vector and Vector Pair (Patwardhan, 2003). The latter two work with large vectors of co-occurring terms from a corpus, so *WordNet* is only part of the system. I used Ted Pedersen's *WordNet::Similarity* software package (Pedersen et al., 2004). Unlike *Roget's Thesaurus*, this implementation of *WordNet*-based semantic relatedness does not allow for measuring distances between two different parts-of-speech. Every measure can be used on pairs of nouns and verbs, though only H&SO, Lesk and the two Vector methods can be applied to adjectives or adverbs.

The measure most similar to the *Roget's SemDist* method is the Path measure in *WordNet*. J&C, Res, Lin, W&P, L&C and Path can only measure relatedness between nouns and verbs,because they only make use of hypernym links. H&SO uses all available semantic relations in finding a path between two words. The Lesk and Vector methods

[1] http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/wordsim353.html

Method	Miller & Charles	Rubenstein & Goodenough	Finkelstein et. al
1911	0.811	0.724	0.343
1911X1	0.791	0.693	0.351
1911X5	0.790	0.676	0.344
1911R	0.805	0.721	0.333
1987	0.824	0.774	0.392
1987X1	0.818	0.788	0.390
1987X5	0.813	0.787	0.383
1987R	0.818	0.772	0.394
Path	0.752	0.783	0.466
J&C	0.473	0.575	0.314
Resnik	0.808	0.823	0.429
Lin	0.747	0.737	0.320
W&P	0.764	0.801	0.355
L&C	0.779	0.842	0.411
H&SO	0.667	0.726	0.423
Lesk	0.797	0.771	0.465
Vector	0.884	0.801	0.447
Vector Pair	0.510	0.580	0.415

Table 6.1: Pearson's coefficient values for three data sets on a variety of relatedness functions.

use glosses and so might be just as easily implemented using a dictionary. They need not take advantage of *WordNet*'s hierarchical structure.

The results for Pearson Correlation – Table 6.1 – do not show any version of *Roget's Thesaurus* to be the best. It is also observable that the correlation values do not necessarily improve for the updated thesauri. This can be largely attributed to the fact that these three data sets tend to contain fairly common words and so by updating the thesaurus no real benefit can be seen. Spearman's correlation has been considered more robust than Pearson's correlation on the task of word similarity (Gabrilovich and Markovitch, 2009), so I consider this the more important of the two measure. Since the change in scores for these data sets is relatively small, and never statistically significant

Method	Miller & Charles	Rubenstein & Goodenough	Finkelstein et. al
1911	0.683	0.600	0.291
1911X1	0.694	0.612	0.298
1911X5	0.684	0.610	0.304
1911R	0.659	0.590	0.282
1987	0.852	0.814	0.436
1987X1	0.849	0.833	0.442
1987X5	0.844	0.831	0.441
1987R	0.848	0.812	0.437
Path	0.701	0.788	0.369
J&C	0.691	0.588	0.160
Resnik	0.751	0.757	0.363
Lin	0.707	0.619	0.213
W&P	0.742	0.775	0.374
L&C	0.724	0.789	0.361
H&SO	0.757	0.784	0.380
Lesk	0.770	0.700	0.329
Vector	0.923	0.793	0.396
Vector Pair	0.659	0.703	0.322

Table 6.2: Spearman's coefficient values for three data sets on a variety of relatedness functions.

– evaluated with a test based on Fisher R-Z transformation[2] – at least I can say that the updates appear to be at worst neutral, if not helpful.

The Spearman correlation results – Table 6.2 – tended to show some improvement when new words were added to *Roget's Thesaurus*. This contradicts what was observed in Table 6.1 for Pearson's correlation. In reality these data sets are too small to say that these increases are statistically significant. Also of interest is that the 1987 *Roget's* performed better than any *WordNet*-based system on these tests. The 1911 version did not fare so well, but it was not statistically worse at $p < 0.05$ than the best *WordNet*-based method for any version of the 1911 *Thesaurus*. I did find that *Roget's* 1911X5 – the best 1911 version – was significantly worse than *Roget's* 1987 – the worst 1987 version.

[2] http://faculty.vassar.edu/lowry/rdiff.html

In general, there are few conclusions about which *WordNet* measure is best. There does not seem to be any single measure that is consistently the best, although the Path measure regularly performs well. Perhaps a simple MSR is sufficient for most relatedness problems, because the more complex methods do not always offer improvement.

6.1.1 Speed: *Roget's* versus *WordNet*

One of the most noticeable differences between the *WordNet* measures based on Pedersen's Perl package and the *Roget's* measures is the speed at which they run. In Table 6.3 I show the real time taken to calculate the semantic distance between all 353 pairs in Finkelstein et al. (2001). The measures based on *Roget's Thesaurus* were much faster than those based on *WordNet*. The time measurement was started after the *Roget's* or *WordNet* indexes and any other objects needed for a given similarity measure were loaded. This test is run on a Macbook Pro with a 2.4 GHz Intel Core 2 Duo processor and 4 GB 677 MHz DDR2 SDRAM. For this evaluation, only the distance between nouns was taken into account. Every system was run 5 times and the average of their times are reported. These tests do not eliminate the possibility that other *WordNet* based methods could be implemented faster, I only compare the Java implementation of *Roget's* and Pedersen's *WordNet::Similarity* Perl Module (Pedersen et al., 2004).

All versions of *Roget's* 1911 shows a distinct advantage over the *WordNet*-based measures, requiring around one eightieth of the time taken by the quickest *WordNet* measure (Lin, 1998a). The slowest *WordNet* measure (Hirst and St-Onge, 1998) takes close to an hour to run while both the 1911 and 1987 *Roget's* take less than one second once the index has loaded. I do not present times for loading the index in *Roget's* or for loading *WordNet* files, but they are in the order of a couple of seconds each. This difference in run time can largely be attributed to the design of the *Roget's* index file which when loaded can be used to determine exactly where in the *Thesaurus* any word appears. Therefore there is no need to access any other files or perform a non-constant number of operations. The distance between two words can be calculated by comparing 9 different numbers indicating the Class, Section, ..., Semicolon Group, Word number, where the words are found. By comparison many *WordNet*-based methods will require the system to start with two synsets and then traverse the hypernym hierarchies for an arbitrary length until finally finding a common ancestor. Even the Lesk-based methods will need to compare sets of words from the glosses, where the number of words cannot be perfectly predicted. H&SO requires random walks through the entire graph structure of *WordNet*,

Method	Time in seconds
1911	0.391
1911X1	0.392
1911X5	0.486
1911R	0.429
1987	0.591
1987X1	0.598
1987X5	0.619
1987R	0.598
Path	38.056
J&C	38.858
Resnik	38.218
Lin	32.445
W&P	66.243
L&C	38.583
H&SO	3466.929
Lesk	83.563
Vector	71.331
Vector Pair	129.124

Table 6.3: Time to perform semantic relatedness tests on Finkelstein et al. (2001).

explaining why it by far requires the longest to run. This means that *Roget's* makes it feasible to run applications that would require fast processing of massive amounts of semantic distance calculations. Examples of these sort of applications can be seen in Sections 6.5 & 6.6.

6.2 Synonym Identification

In this problem I am given a term q and I seek the best synonym s from a set of words C. I used the system from Jarmasz and Szpakowicz (2004) for identifying synonyms with *Roget's*. There are two steps. First I find a set of terms $B \subseteq C$ with the maximum relatedness between q and each term $x \in C$:

$$B = \{x \mid \operatorname*{argmax}_{x \in C} semDist(x, q)\}$$

Next, I take the set of terms $A \subseteq B$ where each $a \in A$ has the maximum number of shortest paths between a and q.

$$A = \{x \mid \operatorname*{argmax}_{x \in B} numberOfShortestPaths(x, q)\}$$

If $s \in A$ and $|A| = 1$, the correct synonym has been selected. Often the sets A and B will contain just one item. If $s \in A$ and $|A| > 1$, there is a tie. If $s \notin A$ then the selected synonym(s) are incorrect. If a multi-word phrase $c \in C$ of length n is found, it is replaced by each of its words $c_1, c_2..., c_n$, and each of these words is considered in turn. The c_i that is closest to q is chosen to represent c. When searching for a word in *Roget's* or *WordNet*, I look for all forms of the word. This is done by adding or removing possible suffixes of the word and searching for all variations in the *Thesaurus*. Words can be of any part-of-speech, though as noted in Section 6.1 only some *WordNet*-based methods allow for adjectives or adverbs and none can measure distance between two parts-of-speech.

I experiment with three frequently used data sets and later go on to generate a few new ones. The data sets I use are from the 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). TOEFL consists of 80 questions, while ESL has 50 and RDWP has 300.

The results of these experiments on the ESL data set appears in Table 6.4, TOEFL data set appears in Table 6.5 and RDWP data set appears in Table 6.6. "Yes" indicates

ESL								
Method	Yes	No	Ties	QNF	ANF	ONF	Precision	Recall
1911	27	20	3	0	3	3	57	57
1911X1	28	18	4	0	3	3	60	60
1911X5	29	16	5	0	3	3	62.67	62.67
1911R	26	20	4	0	2	3	56	56
1987	36	8	6	0	1	1	77.67	77.67
1987X1	36	8	6	0	1	1	77.67	77.67
1987X5	34	8	8	0	1	1	75.67	75.67
1987R	36	8	6	0	1	1	77.67	77.67
Path	30	9	11	4	4	10	72.83	69.00
J&C	30	16	4	4	4	10	65.22	62.00
Resnik	26	18	6	4	4	10	58.70	56.00
Lin	31	14	5	4	4	10	67.93	64.50
W&P	31	13	6	4	4	10	69.57	66.00
L&C	29	10	11	4	4	10	70.65	67.00
H&SO	34	12	4	0	0	0	71.33	71.33
Lesk	38	12	0	0	0	0	76	76.00
Vector	39	11	0	0	0	0	78	78.00
Vector Pair	40	10	0	0	0	0	80	80.00

Table 6.4: Synonym selection experiments for ESL.

TOEFL								
Method	Yes	No	Ties	QNF	ANF	ONF	Precision	Recall
1911	51	26	3	10	5	25	71.07	65.31
1911X1	53	23	4	10	5	25	74.64	68.44
1911X5	53	24	3	10	5	25	73.93	67.81
1911R	51	27	2	9	6	24	70.07	65.00
1987	59	14	7	4	4	17	80.70	77.92
1987X1	59	14	7	4	4	17	80.70	77.92
1987X5	58	14	8	4	4	16	80.04	77.29
1987R	59	14	7	4	4	16	80.70	77.92
Path	38	6	36	33	31	90	83.16	59.17
J&C	34	9	37	33	31	90	74.47	54.06
Resnik	37	6	37	33	31	90	81.91	58.44
Lin	33	6	41	33	31	90	74.47	54.06
W&P	39	5	36	33	31	90	85.64	60.63
L&C	38	6	36	33	31	90	83.16	59.17
H&SO	60	4	16	1	0	1	81.75	81.04
Lesk	70	9	1	1	0	1	88.61	87.81
Vector	69	10	1	1	0	1	87.34	86.56
Vector Pair	65	13	2	1	0	1	82.59	81.88

Table 6.5: Synonym selection experiments for TOEFL.

RDWP								
Method	Yes	No	Ties	QNF	ANF	ONF	Precision	Recall
1911	157	130	13	57	13	78	61.41	54.50
1911X1	159	129	12	57	13	77	62.04	55.00
1911X5	155	130	15	57	13	75	61.01	54.17
1911R	155	131	14	51	14	77	59.94	54.00
1987	198	85	17	22	5	17	72.15	68.69
1987X1	198	85	17	22	5	17	72.09	68.64
1987X5	196	82	22	22	5	17	72.15	68.69
1987R	198	85	17	22	5	17	72.15	68.69
Path	148	56	96	62	58	150	68.03	59.14
J&C	100	54	146	62	58	150	50.92	45.57
Resnik	114	72	114	62	58	150	55.85	49.47
Lin	94	46	160	62	58	150	49.98	44.82
W&P	147	66	87	62	58	150	66.04	57.56
L&C	149	58	93	62	58	150	67.82	58.97
H&SO	170	48	82	4	6	5	65.43	64.89
Lesk	220	73	7	4	6	5	74.77	74.11
Vector	216	76	8	4	6	5	73.65	73.00
Vector Pair	187	103	10	4	6	5	63.779	63.25

Table 6.6: Synonym selection experiments for RDWP.

correct answers, "No" – incorrect answers, and "Tie" is for ties. QNF stands for "Question word Not Found", ANF for "Answer word Not Found" and ONF for "Other word Not Found". Scores of precision and recall are also presented. Recall is the percentage of correctly answered problems over the entire data set, while precision is the percentage of correctly answered problems where the query word could be found in *Roget's* or *WordNet*. Overall, recall is probably the more important measure because it will give a score for the entire data set. It is the percentage of questions answered right, plus the percentage of unbroken ties normalized by the number of tied words. I used three data sets for this application: 80 questions taken from the Test of English as a Foreign Language (TOEFL) (Landauer and Dumais, 1997), 50 questions – from the English as a Second Language test (ESL) (Turney, 2001) and 300 questions – from the Reader's Digest Word Power Game (RDWP) (Lewis, 2001).

It is observable for the ESL and TOEFL data sets – Table 6.4 & Table 6.5 – that the expanded versions of the 1911 *Thesaurus* tend to do better than the original versions. On the Readers Digest problems – Table 6.6 – *Roget's* 1911X1 and 1987X1 do perform better than the original versions, though 1911X5 and 1987X5 perform noticeably worse. Even on this larger data set the differences are not so significant. Generally there was little change found by updating *Roget's*, though on the Readers Digest problems there was some noticeable improvement found for updated versions of the 1911 *Thesaurus*. Unfortunately as these words were neither the query words nor the correct answer, they did not contribute positively to the results on the data set.

Lesk and the Vector-based systems perform better than all other measures, including the versions of the *Roget's Thesaurus*. Even so, no other *WordNet* based system consistently outperformed the versions of 1987 *Roget's*. The versions of the 1911 *Thesaurus* were noticeably worse, but they still outperformed most *WordNet* based measures on the larger two data sets. In fact, six of the ten *WordNet*-based methods are consistently worse than the 1911 *Roget's Thesaurus*. One advantage of *Roget's Thesaurus* is that both versions often have fewer missing terms than *WordNet*, though Lesk, Hirst & St-Onge and the two vector-based methods had fewer missing terms than *Roget's*. This is because the other *WordNet* methods will only work for nouns and verbs.

6.2.1 Testing New Words Specifically

To test newly added words, I generate new synonym selection problems that specifically target the words newly added to *Roget's*. I take all words that appear in either the

1911 – Nouns								
Method	Yes	No	Ties	QNF	ANF	ONF	Precision	Recall
1911	0	98	0	98	0	0	0	0
1911X1	18	70	10	44	0	0	40.13	22.11
1911X5	30	45	23	0	0	0	39.63	39.63
1911R	3	93	2	88	0	0	39.98	4.08

Table 6.7: Evaluation on new data from 1911 Nouns using *WordNet* as a source of data.

1911 – Verbs								
Method	Yes	No	Ties	QNF	ANF	ONF	Precision	Recall
1911	0	27	0	27	0	0	0	0
1911X1	6	20	1	13	0	0	46.42	24.07
1911X5	11	14	2	0	0	0	44.44	44.44
1911R	0	27	0	26	0	0	0	0

Table 6.8: Evaluation on new data from 1911 Verbs using *WordNet* as a source of data.

1987X5 or 1911X5 but are not present in the original 1987 or 1911 versions, and use them as query words q for new problems generated using *WordNet*. I then find synsets in *WordNet* that contain at least one synonym s for q, where s is found in the non-updated version of *Roget's*. I then pick false synonyms $f1$, $f2$ and $f3$ from co-hyponym synsets to generate the problems, where $f1$, $f2$ and $f3$ are all found in the non-updated *Roget's*. I do this for both nouns and verbs.

Since the query word q may have morphological variations present in the non-updated version of *Roget's*, I do not use morphological variants or words found in phrases when solving these synonym problems. Four different versions of this problem are generated for the 1911 and 1987 *Roget's* using nouns and verbs. The linking structure for adjectives in *WordNet* makes it impossible to create a data set in this manner. Once again I present the final scores as precision and recall. Precision will be the score on questions that were possible to answer, i.e. excluding missing questions and recall will be the score over the entire data set.

See the results in Table 6.7 for 1911 nouns, Table 6.8 for 1911 verbs, Table 6.9 for 1987 nouns and Table 6.10 for 1987 verbs. The results found in all four tables are quite similar. Obviously a precision and recall of 0 is attained for the non-updated versions of

1987 – Nouns								
Method	Yes	No	Ties	QNF	ANF	ONF	Precision	Recall
1987	0	57	0	57	0	0	0	0
1987X1	11	38	8	18	0	0	38.03	26.02
1987X5	18	29	10	0	0	0	39.77	39.77
1987R	0	56	1	52	0	0	10.03	0.88

Table 6.9: Evaluation on new data from 1987 Nouns using *WordNet* as a source of data.

1987 – Verbs								
Method	Yes	No	Ties	QNF	ANF	ONF	Precision	Recall
1987	0	36	0	36	0	0	0	0
1987X1	5	27	4	20	0	0	41.67	18.52
1987X5	12	15	9	0	0	0	44.91	44.91
1987R	1	35	0	29	0	0	14.29	2.78

Table 6.10: Evaluation on new data from 1987 Verbs using *WordNet* as a source of data.

Roget's. The randomly updated versions did poorly as well. Versions that were updated after one pass had recall between 18% and 26%, while the versions updated after 5 passes had 40% or more. The random baseline is 25% if all of the questions can be answered. The thesauri updated with 5 passes all significantly beat this baseline. Significance was established with a Student's T-test where $p < 0.05$. The thesauri updated with 1 pass tended not to, though many of the problems they faced were unsolvable because q may not appear in *Roget's* 1911X1 or 1987X1.

In terms of recall, the improvement of the *Thesaurus* updated with 5 passes was significantly better (at $p < 0.05$) than for the thesaurus updated with one pass. In turn, the thesaurus updated with one pass was significantly better than the original thesaurus, at $p < 0.05$. The only exception to this was on the 1911 verb data set, where the improvement could only be measured as significant with $p < 0.065$. This is largely because the data set was fairly small. Another observation is that the randomly updated *Thesaurus* only once had a significant improvement over the original *Thesaurus*, in the case of the 1911 noun data set.

These results suggest that the words newly added in *Roget's* appear to be close to the correct location. They were more accurate at finding words in the same synset than words from synsets sharing a co-hypernym relationship. Generally the precision measure showed words added to the 1911X1 and 1987X1 thesauri to be approximately as accurate as, if not slightly more accurate than, those added in passes 2-5. The randomly updated *Thesaurus* did not perform as well, usually falling below the 25% baseline on the precision measure. The one noticeable exception to this is the results in Table 6.7, though it should be noted that it was evaluated on a very small sample.

6.3 Sentence Relatedness

The next experiment concerns sentence relatedness. I worked with a data set from (Li et al., 2006)[3]. They took a subset of the term pairs from (Rubenstein and Goodenough, 1965) and chose sentences to represent these terms; the sentences are definitions from the Collins Cobuild dictionary (Sinclair, 2001). Thirty people were then asked to assign relatedness scores to these sentences, and the average of these similarities was taken for each sentence.

Other methods of determining sentence relatedness expand term relatedness functions to create a sentence relatedness function (Islam and Inkpen, 2007; Mihalcea et al., 2006). I propose to approach the task by exploiting in other ways the commonalities in the structure of *Roget's* Thesaurus and of *WordNet*. I use the OpenNLP toolkit[4] for word segmentation and part-of-speech tagging.

I create a method of sentence representation that works by mapping the sentence into weighted concepts in either *Roget's* or *WordNet*. I mean a concept in *Roget's* to be any *Roget's grouping* while a concept in *WordNet* is any synset and hypernym synset. Essentially, a concept is a grouping of words from either resource. Concepts are weighted by two criteria. The first is how frequently words from the sentence appear in these concepts. The second is the depth (or specificity) of the concept itself. This is done with the assumption that concepts that appear high up in the hierarchy will be very general while those appearing farther down will be more specific.

6.3.1 Weighting Based on Word Frequency

Each word and punctuation mark w in a sentence is given a score of 1. If w has n word senses $w_1, ..., w_n$, each sense gets a score of $1/n$, so that $1/n$ is added to each concept

[3] http://www.docm.mmu.ac.uk/STAFF/D.McLean/SentenceResults.htm

[4] http://opennlp.sourceforge.net

Identifier	**Concept**	**Weight**
6	Words Relating to the Voluntary Powers - Individual Volition	2.125169028274
6.2	Prospective Volition	1.504066255252
6.2.2	Subservience to Ends	1.128154077172
8	Words Relating to the Sentiment and Moral Powers	3.13220884041
8.2	Personal Affections	1.861744448402
8.2.2	Discriminative Affections	1.636503978149
8.2.2.2	Ornament/Jewelry/Blemish [Head Group]	1.452380952380
8.2.2.2.886	Jewelry [Head]	1.452380952380
8.2.2.2.886.1	Jewelry [Noun]	1.452380952380
8.2.2.2.886.1.1	jewel [Paragraph]	1.452380952380
8.2.2.2.886.1.1.1	jewel [Semicolon Group]	1.166666666666
8.2.2.2.886.1.1.1.3	jewellery [Word Sense]	1.0
or	-	1.0
in	-	1.0
that	-	1.0
a	-	2.0
.	-	1.0

Table 6.11: "A gem is a jewel or stone that is used in jewellery." as represented using *Roget's* 1911.

in the *Roget's* hierarchy (Semicolon Group, Paragraph, ..., Class) or *WordNet* hierarchy that contains w_i. I weight concepts in this way simply because, unable to determine which sense is correct, I assume that all senses are equally probable. Each concept in *Roget's* Thesaurus and *WordNet* gets the sum of the scores of the concepts below it in its hierarchy.

I define the scores recursively for a concept c in a sentence s and sub-concepts c_i. For example, in *Roget's* if the concept c were a Class, then each c_i would be a Section. Likewise, in *WordNet* if c were a synset, then each c_i would be a hyponym synset of c. If c is a word sense w_i (a word in either a synset or a Semicolon Group), then there can be no sub-concepts c_i. When $c = w_i$, the score for c is the sum of all occurrences of the word w in sentence s divided by the number of senses of the word w – See Equation 6.1.

$$score(c, s) = \begin{cases} \frac{instancesOf(w,s)}{sensesOf(w)} & \text{if} \quad c = w_i \\ \sum_{c_i \in c} score(c_i, s) & \text{otherwise} \end{cases} \tag{6.1}$$

See Table 6.11 for an example of how this sentence representation works. The sentence "A gem is a jewel or stone that is used in jewellery." is represented using the 1911 *Roget's*.

A concept is identified by a name and a series of up to 9 numbers that indicate where in the *Thesaurus* it appears. The first number represents the Class, the second the Section, ..., the ninth the word. I only show concepts with weights greater than 1.0. Words not in the thesaurus keep a weight of 1.0, but this weight will not increase the weight of any concepts in *Roget's* or *WordNet*. Apart from the function words "or", "in", "that" and "a" and the period, only the word "jewellery" had a weight above 1.0. The categories labelled 6, 6.2 and 6.2.2 are the only ancestors of the word "use" that ended up with the weights above 1.0. The words "gem", "is", "jewel", "stone" and "used" all contributed weight to the categories shown in Table 6.11, and to some categories with weights lower than 1.0, but no sense of the words themselves had a weight greater than 1.0.

It is worth noting that this method only relies on the hierarchies in *Roget's* and *WordNet*. I do not take advantage of other *WordNet* relations such as meronymy, nor do I use any cross-reference links that exist in *Roget's* Thesaurus. Including such relations might improve this sentence relatedness system, but that has been left for future work.

6.3.2 Weighting Based on Specificity

To determine sentence relatedness, one could, for example, flatten the structures like those in Table 6.11 into vectors and measure their closeness by some vector distance function such as cosine similarity. There is a problem with this. A concept inherits the weights of all its sub-concepts, so the concepts that appear closer to the root of the tree will far outweigh others. Some sort of weighting function should be used to re-adjust the weights of particular concepts. Were this an Information Retrieval task, weighting schemes such as *tf.idf* for each concept could apply, but for sentence relatedness I propose an *ad hoc* weighting scheme based on assumptions about which concepts are most important to sentence representation. This weighting scheme will allow for a free parameter that must be tuned to determine its optimal value. This weighting scheme is the second element of the sentence relatedness function.

I weight a concept in *Roget's* and in *WordNet* by how many words in a sentence give weight to it. I re-weight it based on how specific it is. Concepts near the leaves of the hierarchy are more specific than those close to the root of the hierarchy. I define specificity as the distance in levels between a given word and each concept found above it in the hierarchy. In *Roget's* Thesaurus there are exactly 9 levels from the term to the class. In *WordNet* there will be as many levels as a word has ancestors up the hypernymy chain. In *Roget's*, a term has specificity 1, a Semicolon Group 2, a Paragraph 3, ..., a

Class 9. In *WordNet*, the specificity of a word is 1, its synset – 2, the synset's hypernym – 3, *its* hypernym – 4, and so on. Words not found in *Roget's* or in *WordNet* get specificity 1.

I seek a function that, given s, assigns to all concepts of specificity s a weight progressively larger than their neighbours. The weights in this function should be assigned based on specificity, so that all concepts of the same specificity receive the same score. Weights will differ depending on a combination of specificity and how frequently words that signal the concepts appear in a sentence. The weight of concepts with specificity s should be the highest, of those with specificity $s \pm 1$ – lower, of those with specificity $s \pm 2$ lower still, and so on. In order to achieve this effect, I weight the concepts using a normal distribution, where the mean is s and σ is the standard deviation:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left(-\frac{(x-s)^2}{2\sigma^2}\right)}$$

Standard deviation is set to 1.0 while the mean is a free parameter. Since the Head is often considered the main category in *Roget's*, I expect a specificity of 5 to be best, but I decided to test the values 1 through 9 as a possible setting for specificity. I do not claim that this weighting scheme is optimal; other weighting schemes might do better. For the purpose of comparing the 1911 and 1987 Thesauri and *WordNet* this method appears sufficient.[5]

With this weighting scheme, I determine the distance between two sentences using cosine similarity:

$$cosSim(A, B) = \frac{\sum a_i * b_i}{\sqrt{\sum a_i^2} * \sqrt{\sum b_i^2}}$$

6.3.3 Sentence Similarity Results

Once again, this system can be evaluated using either Pearson's or Spearman's correlation. The results for Pearson's correlation can be found in Figure 6.1, while Spearman's is shown in Figure 6.2. For comparison, I also implemented a baseline method that I refer to as Simple: I built vectors out of words and their count. These figures are a little bit difficult to decipher due to the large number of overlapping lines, so I produce graphs specific to the various versions of the 1911 *Thesaurus* and 1987 *Thesaurus* with Pearson's correlation in Figures 6.3 & 6.4 respectively and for Spearman's correlation in Figures 6.5 & 6.6 respectively.

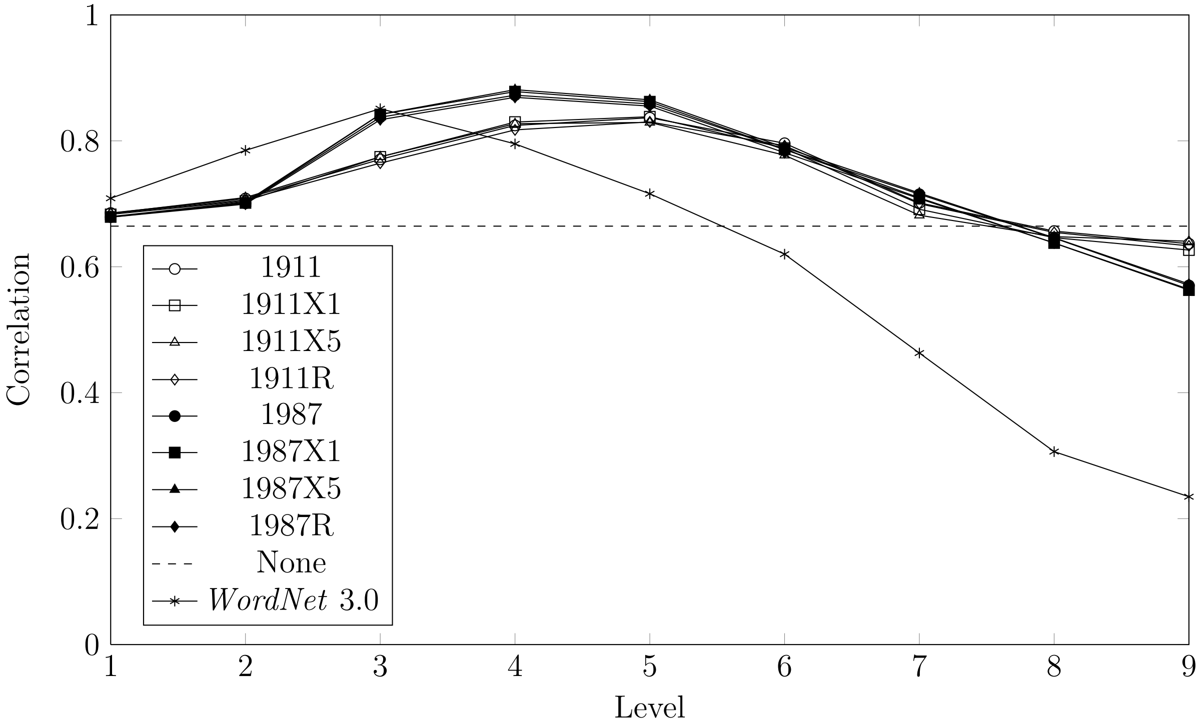

Figure 6.1: Pearson's correlation data for all eight systems.

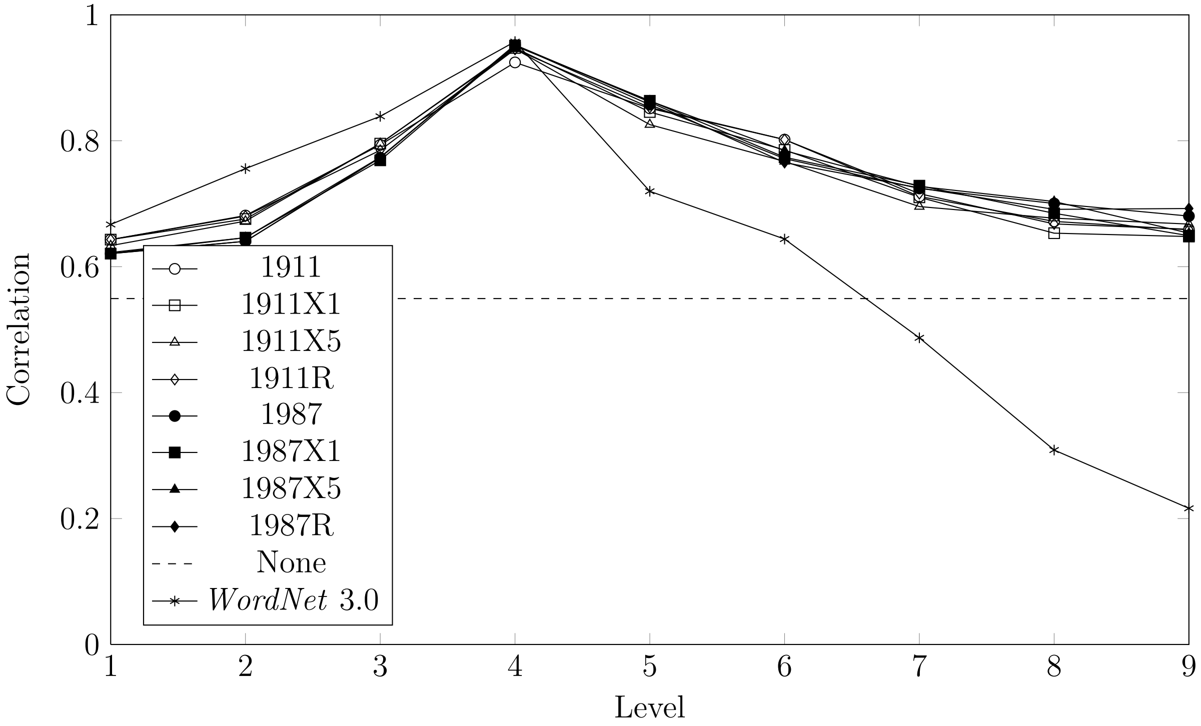

Figure 6.2: Spearman's correlation data for all eight systems.

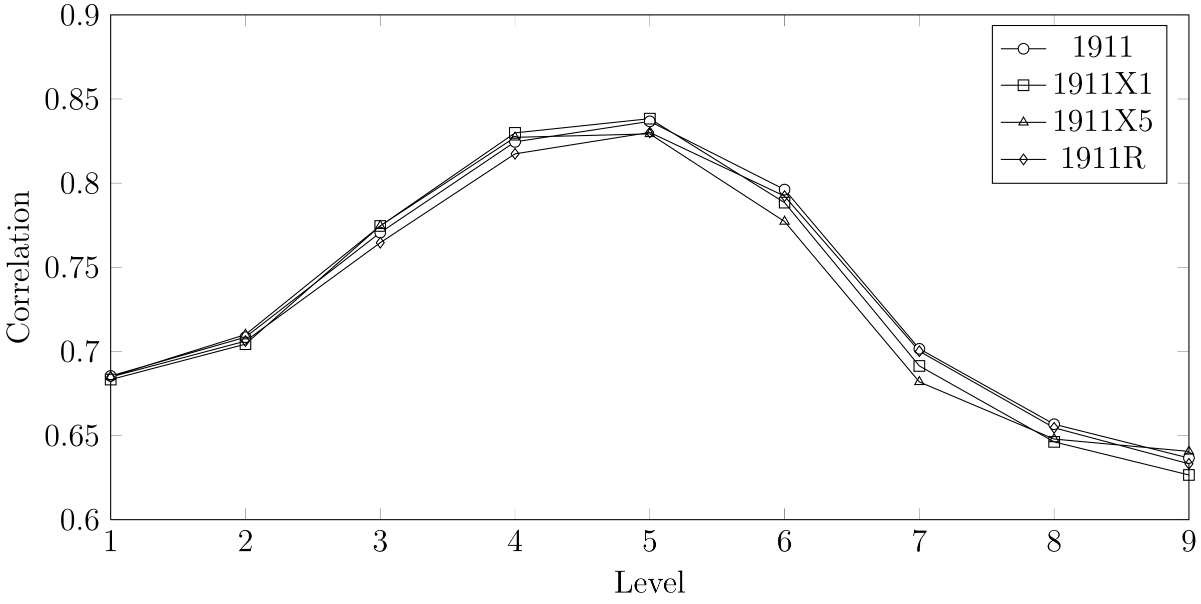

Figure 6.3: Pearson's correlation data for the 1911 *Roget's Thesaurus*.

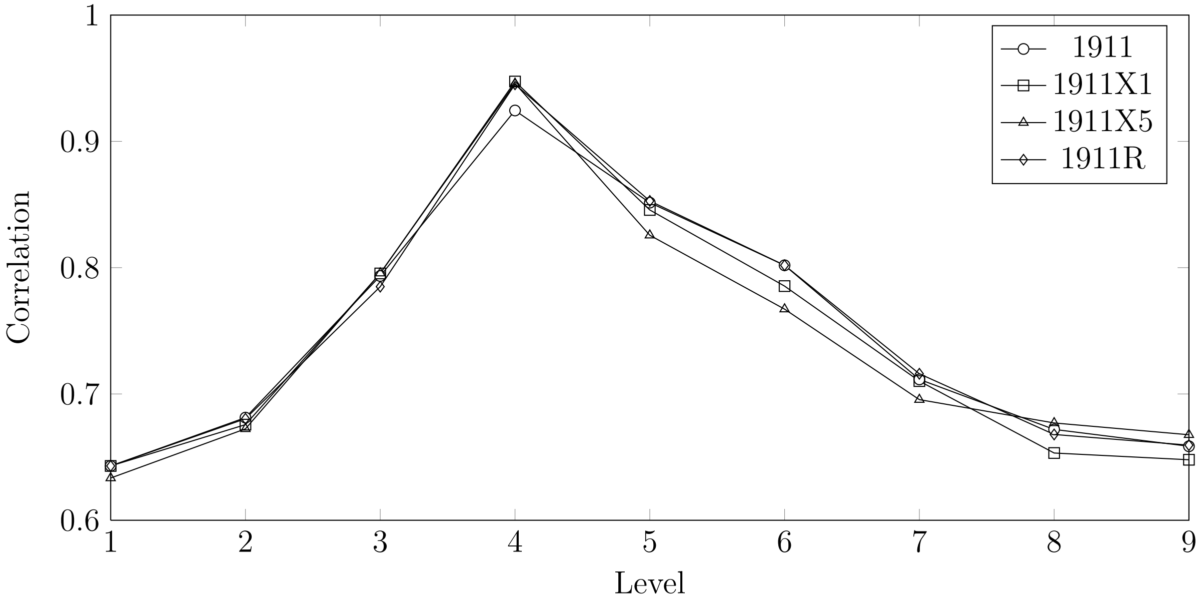

Figure 6.4: Spearman's correlation data for the 1911 *Roget's Thesaurus*.

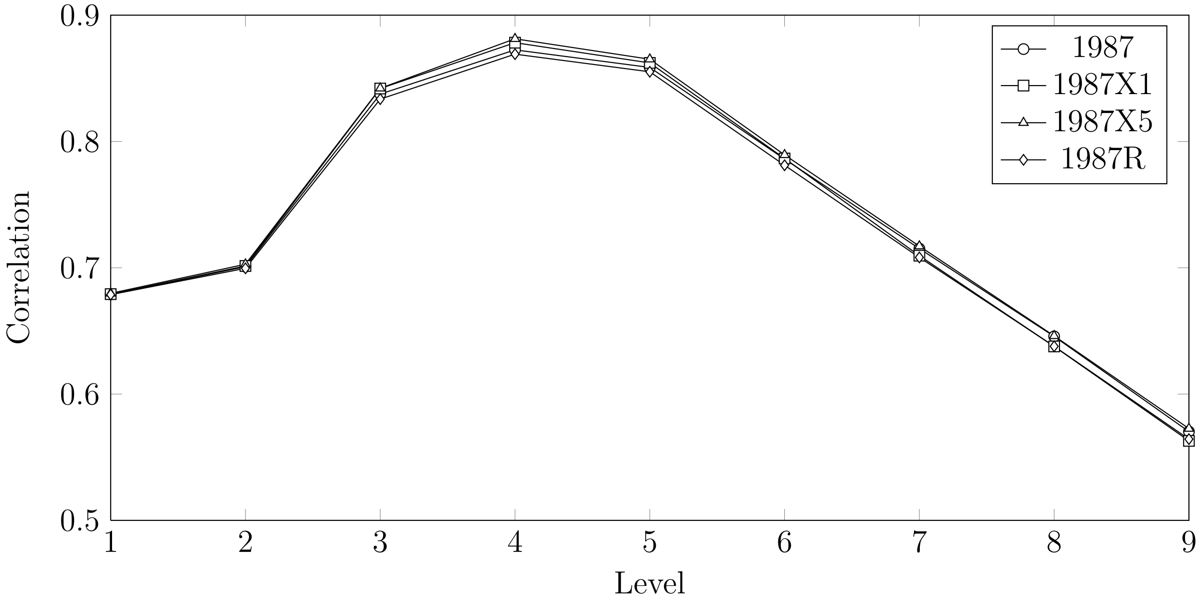

Figure 6.5: Pearson's correlation data for the 1987 *Roget's Thesaurus*.

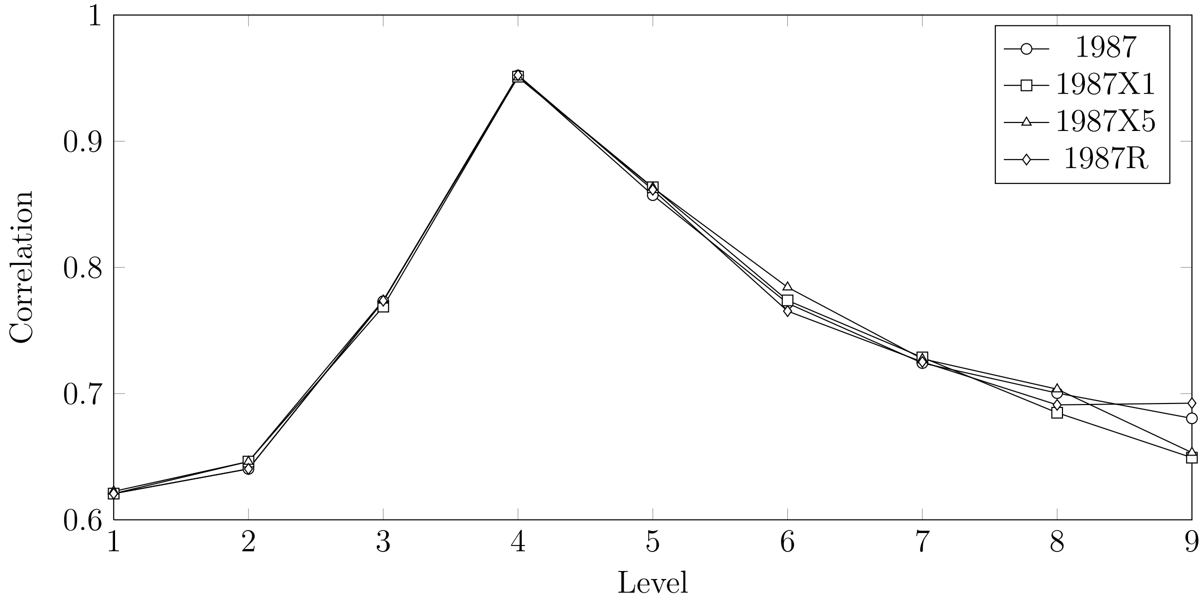

Figure 6.6: Spearman's correlation data for the 1987 *Roget's Thesaurus*.

The best correlation scores for both Pearson and Spearman are shown in Table 6.12. Interestingly the best results for *Roget's* tend to be seen at the POS level, though for the 1911 *Roget's* sometimes the Head level is preferable. In most cases the scores between these two groupings are fairly close. All methods outperform the baseline method by a fair margin. In terms of statistical significance for Pearson's correlation, the 1987 *Roget's* as well as 1987X1 and 1987X5 outperformed the simple baseline with $p < 0.05$.[6] No other systems significantly outperformed the baseline. For Spearman's correlation all systems outperformed the baseline. The difference between systems could not be established as significant on this relatively small data set. All enhanced versions of *Roget's* outperformed their original versions, though in some cases *Roget's* 1911X5 and 1987X5 did not outperform 1911X1 or 1987X1. *WordNet* performed comparably to the versions of *Roget's*, though on Spearman's correlation it appeared to perform best. Interestingly the scores for Spearman correlation were consistently much higher than Pearson's.

Several other methods have given very good scores on this data set. For the system

[5] For this problem I used the MIT Java *WordNet* Interface version 2.1.5, available at: http://www.mit.edu/~markaf/projects/wordnet/

[6] Established using http://vassarstats.net/rdiff.html

Method	Pearson	Level	Spearman	Level
1911	0.837	Head	0.924	POS
1911X1	0.838	Head	0.947	POS
1911X5	0.829	Head	0.946	POS
1911R	0.830	Head	0.945	POS
1987	0.873	POS	0.952	POS
1987X1	0.878	POS	0.951	POS
1987X5	0.881	POS	0.951	POS
1987R	0.869	POS	0.952	POS
WN 3.0	0.851	1^{st} hypernym	0.957	2^{nd} hypernym
Simple	0.665	-	0.549	-

Table 6.12: Optimal Pearson and Spearman correlations as well as the level of granularity within *WordNet* or *Roget's* at which that score was achieved.

in (Li et al., 2006), where this data set was first introduced, a Pearson's correlation of 0.816 with the human annotators was achieved. The mean of all human annotators had a correlation of 0.825, with a standard deviation of 0.072. The lowest-scoring human annotator had Pearson's correlation of 0.594, while the highest had Pearson's correlation of 0.921. To my knowledge, the best system on this particular data set is that of (Islam et al., 2012) where Pearson's correlation of 0.916 is achieved. Other systems tackling this data set have had scores between 0.756 and 0.895 (Liu et al., 2007; Islam and Inkpen, 2007; Feng et al., 2008; O'Shea et al., 2008; Ho et al., 2010; Hassan and Mihalcea, 2011). My best score with Pearson's correlation was 0.881, albeit with specificity as a parameter that was tuned to this data set.

Selecting the mean that gives the best correlation could be considered as training on test data. That said, were I simply to have selected a value somewhere in the middle of the graph, as was my original intuition, it would have given an unfair advantage to the versions of *Roget's* Thesaurus over *WordNet*. The 1987 *Thesaurus* once again performs better than the 1911 version and *WordNet*. Much like the benchmark from (Miller and Charles, 1991), the data set used here is not large enough to determine if any system's improvement is statistically significant.

6.4 SAT Analogies

Another class of problems that I attempt to apply *Roget's Thesaurus* to is that of solving Scholastic Aptitude Tests (SAT) style analogy problems. In an SAT analogy task, one is given a *target pair* $\langle A, B\rangle$ and then from a list of possible candidates they must select the pair $\langle C, D\rangle$ that is most similar to the *target pair.* Ideally the relation between the pair $\langle A, B\rangle$ and the relation between the pair $\langle C, D\rangle$ should be identical. For example:

Target pair	*word, language*
Candidates	*paint, portrait*
	poetry, rhythm
	note, music
	tale, story
	week, year

Although *Roget's* performs well on problems of semantic relatedness it is not clear just how well it will perform on tasks of identifying analogies, as relationships in *Roget's* are unlabelled. I will attempt two methods of solving this problem with both *Roget's* and *WordNet*. In the first method I will attempt to identify a few kinds of relations in *Roget's* and then apply them to identifying analogies. The second method will be to use semantic relatedness between the pairs $\langle A, B\rangle$, $\langle C, D\rangle$ and also $\langle A, C\rangle$ and $\langle B, D\rangle$ as a heuristic for guessing whether two word pairs contain the same relationship.

The dataset that I work with contains 374 analogy problems extracted from real SAT tests and preparation tests (Turney, 2005). Each problem contains a *target pair* $\langle A, B\rangle$, and several option pairs to choose from: $test_i = \langle X_i, Y_i\rangle$, i = 1..5. In evaluating this work I will consider seven different scores: correct, ties, incorrect, filtered out, precision, recall and equal-weighted F-score. Precision will be the accuracy over all the problems that were attempted, i.e. not filtered out, while recall will be the accuracy over the entire set. In the case of n-way tie, the correct answer counts as $1/n$ towards the precision and recall. I consider recall to be the most important measure, as it evaluates each method over the entire data set.

6.4.1 Matching Relations

Although *Roget's* contains no explicit semantic relations, there are a number of implicit ones that can be inferred from its structure. As seen in Chapter 2, near synonyms

System	Correct	Ties	Incorrect	Filtered	Precision	Recall	F-measure
1911	14	21	39	300	0.307	0.061	0.102
1911X1	15	23	39	297	0.321	0.066	0.110
1911X5	15	27	39	293	0.330	0.072	0.118
1911R	14	21	39	300	0.307	0.061	0.102
1987	18	85	81	190	0.271	0.133	0.179
1987X1	19	85	81	189	0.273	0.135	0.181
1987X5	21	85	81	187	0.278	0.139	0.185
1987R	18	86	80	190	0.271	0.133	0.179
WordNet 3.0	20	4	12	338	0.600	0.058	0.105

Table 6.13: Scores in the analogy problem solved by matching kinds of relations.

tend to appear in the same SG while near antonyms tend to appear in different Heads in the same Head Group. One can also infer a hierarchical relationship between two words if (1) they appear in the same Paragraph and one of them appears in the first SG, or (2) they appear in the same POS and one of them appears in the first SG of the first Paragraph. This gives us three relationships from *Roget's*, near-synonymy, near-antonymy and hierarchically-related. From *WordNet* I use all available semantic relationships and the transitive closure is included in the case of hypernymy/hyponymy.

In this method the analogy problem is solved by identifying a candidate analogy that contains the same relation. There is a problem that if the target pair is not found to have a relation of any sort between them then the problem cannot be answered. This experiment is interesting in that it will help to identify whether very specific semantic relations, like those in *WordNet*, are more or less useful than very broad relations, like those in *Roget's*. Table 6.13 shows the results; "filtered" shows the number of pairs which were not scored because no relation could be established between the words in the target or candidate pairs.

The *WordNet*-based method has high precision, but recall is low compared to the *Roget's* versions. Interestingly the precision and recall both increase as more words are added to *Roget's* for both the 1911 and 1987 versions. As I consider recall to be the most important method in this evaluation one can see that the most updated versions of *Roget's* 1911X and 1987X outperform *WordNet* by a fair margin. Although the original 1911 version of *Roget's* performed worse than *WordNet* in terms of f-measure, all other versions performed better. The existence of very specific semantic relations in *WordNet* did give it an edge in terms of precision, but it was only able to answer a few questions. This indicates that the relations between pairs in analogy tests are not only of the type encountered in *WordNet*. While the broader relations identified in *Roget's* appear to be less reliable and give lower precision, their recall is much higher.

6.4.2 Semantic Relatedness

The second method of solving analogy problems uses semantic relatedness as defined in 6.1 as a heuristic. Analogy problems have been solved in this way using a formula proposed in Turney (2006):

$$score(\langle A, B\rangle : \langle X_i, Y_i\rangle) = \frac{1}{2} * (sim_a(A, X_i) + sim_a(B, Y_i))$$

The highest-scoring pair $\langle X_i, Y_i\rangle$ is guessed to be the correct analogy. This method is based on an assumption that A and X_i should be closely related as are B and Y_i. An example illustrating the logic behind this is: $\langle carpenter, wood\rangle$ and $\langle mason, stone\rangle$.

In the formula above, sim_a is the attributional similarity, I will replace it with a semantic relatedness measure, either *SemDist* or one of the measures built on *WordNet*. To be general I will refer to this measure as rel. The *SemDist* semantic relatedness measure only gives scores of even numbers between 0 and 18 and so has a tendency to have a lot of ties. To alleviate this I use the following formula which incorporates a tie breaker based on the similarities between A and B and also between X_i and Y_i[7]:

$$score(\langle A, B\rangle, \langle X_i, Y_i\rangle) = rel(A, X_i) + rel(B, Y_i) + \frac{1}{|rel(A, B) - rel(X_i, Y_i)| + 1} \tag{6.2}$$

The tie-breaker $\frac{1}{|rel(A,B)-rel(X_i,Y_i)|+1}$ is used to select candidates $\langle X_i, Y_i\rangle$ that have a similar semantic relatedness to the target $\langle A, B\rangle$. I include another constraint, specifically that A and X_i must have the same part-of-speech, as do B and Y_i. Only one sense of each A, B, X_i and Y_i can be used in the calculation of Equation 6.2. By this I mean when calculating $rel(A, X_i)$ and $rel(A, B)$ the same sense of A is used.

I apply Equation 6.2 to the 374 analogy problems using all versions of *Roget's* and the *WordNet*-based semantic relatedness measures. The results are shown in Table 6.14. The $filtered$ column shows how many SAT problems could not be solved because at least one of the words needed could not be found in either *Roget's* or *WordNet*. Unfortunately

[7] This formula came from a personal communication with Dr. Vivi Nastase and was also used in (Kennedy and Szpakowicz, 2007)

System	Correct	Ties	Misses	Filtered	Precision	Recall	F-measure
1911	98	11	214	51	0.319	0.276	0.296
1911X1	98	17	208	51	0.329	0.284	0.305
1911X5	97	20	206	51	0.330	0.285	0.306
1911R	97	12	218	47	0.313	0.274	0.292
1987	101	35	232	6	0.318	0.313	0.316
1987X1	102	38	228	6	0.324	0.319	0.322
1987X5	102	39	227	6	0.325	0.320	0.323
1987R	103	34	233	4	0.320	0.316	0.318
Path	85	5	166	118	0.342	0.234	0.278
J&C	80	0	176	118	0.312	0.214	0.254
Resnik	91	16	149	118	0.385	0.263	0.313
Lin	82	3	171	118	0.325	0.222	0.264
W&P	90	1	165	118	0.354	0.242	0.287
L&C	91	4	161	118	0.363	0.249	0.295
H&SO	96	39	212	27	0.321	0.298	0.309
Lesk	113	0	234	27	0.326	0.302	0.313
Vector	113	0	234	27	0.326	0.302	0.313
Vector Pair	106	0	241	27	0.305	0.283	0.294

Table 6.14: Scores in the analogy problem solved using semantic distance function.

expanding the thesaurus did not reduce the number of filtered results. That said the precision and recall actually increased when more words were added to *Roget's*. Even so, these systems are well below the average human performance of 57%.

Overall my findings are that the updated *Roget's* 1987X5 performed better than any other measure examined. Even the updated versions of *Roget's* 1911 performed on par with the best *WordNet*-based measures.

6.5 Pseudo-Word-Sense Disambiguation

Pseudo-word-sense disambiguation, or pseudo-disambiguation, is a rather contrived task with the goal of evaluating the quality of a word-sense disambiguation system. The set-up for this task is to take two words and merge them into a *pseudo-word*. A word-sense disambiguation system then has the goal of identifying which of the two words in the pseudo-word actually belongs in a given context in which the whole pseudo-word appears. One advantage of experimenting with pseudo-word-sense disambiguation is that it will give me a chance to more accurately measure the amount of time each problem takes to run.

I use pseudo-word-sense disambiguation instead of real word-sense disambiguation for two main reasons. The first is that, to my knowledge, there is no word-sense disambiguation data set annotated with *Roget's* word senses and so one would have to be built from scratch. Worse still, to compare word-sense disambiguation systems built using *Roget's* and *WordNet* I would need a dataset labeled with senses from both. Pseudo-word-sense disambiguation gives me a fast way of building a dataset that can be used for evaluation of both *Roget's* and *WordNet* word-sense disambiguation systems.

A common variation on this task is to make triples out of a noun and two verbs and then determine which of the verbs takes the noun as its object. The aim here is to create a kind of verb disambiguation system that incorporates measures of semantic relatedness between nouns. In theory, this measure can help to indicate how well a system identifies contexts (verb object) in which a verb appears, which can be useful in real word-sense disambiguation. Others who have worked on different variations of pseudo-word-sense disambiguation include Gale et al. (1992); Schütze (1998); Lee (1999); Dagan et al. (1999); Rooth et al. (1999); Clark and Weir (2002); Weeds and Weir (2005); Zhitomirsky-Geffet and Dagan (2009). The methodology I use is similar to that of Weeds and Weir (2005).

Construction of the data set was done in 4 steps:

1. parse Wikipedia with *Minipar* (Lin, 1998a);
2. select all object relations and count the frequency of each verb-noun pair $\langle n, v \rangle$;
3. separate the noun-verb pairs into a training set (80%) and a test set (20%);
4. for each pair $\langle n, v \rangle$ in the test set find another verb v' such that v and v' have the same frequency; replace $\langle n, v \rangle$ with the test triple $\langle n, v, v' \rangle$.

This creates two data sets, one of which is a training set of noun-verb pairs $\langle n, v \rangle$. The second data set is the test set made up of noun-verb-verb triples $\langle n, v, v' \rangle$. Examples of such triples are $\langle task, assign, rock \rangle$ and $\langle data, extract, anticipate \rangle$. I select v' ensuring it appeared with equal frequency as $v \pm 1$, in addition to this I made sure that the pair $\langle n, v' \rangle$ does not appear anywhere in the training or test data. To reduce noise and decrease the overall size of the dataset, I removed all noun-verb object pairs which appeared less than five times from both the test and training set. This gave a test set of 3327 triples and a training set of 464,303 pairs. I only used half of Wikipedia to generate this data set, particularly the half that was not used in constructing the Noun matrix from Chapter 4.

To solve the pseudo-word-sense disambiguation task for each triple $\langle n, v, v' \rangle$, I find in the training corpus k nouns which are the closest to n. Every such noun m gets a vote: the number of occurrences of the pair $\langle m, v \rangle$ minus the number of occurrences of $\langle m, v' \rangle$. Any value of k could potentially be used. This means comparing each noun n in the test data to every noun m in the training set if these nous share a common verb v or v'. Such a computation is feasible in *Roget's*, but it takes a very long time for any *WordNet* measure.[8] To ensure that a fair value is selected, I divided the test set into 30 sets and use 29 folds to find the optimal value of k and apply it to the 30th fold.

The score for the pseudo-word-sense disambiguation task is typically measured as an error rate where T is the number of test cases:

$$Error\ rate = \frac{1}{T}\left(\#\ of\ incorrect\ choices + \frac{\#\ of\ ties}{2}\right)$$

Table 6.15 shows the results of this experiment. The improvement of *Roget's* 1911X1 and 1911X5 was statistically significant over that of the original 1911 version at $p < 0.05$, discovered using a Student's t-test. That said, the improvement on the updated 1987

[8] I ran these experiments on a different workstation from the experiments in Section 6.1 because I could not be without my laptop for such an extended period. I used an IBM ThinkCentre with a 3.4 GHz Intel Pentium 4 processor and 1.5GB 400 MHz DDR RAM.

Method	Error Rate	p-value	Relative Improvement	Time in seconds
1911	0.257	-	-	58
1911X1	0.252	0.000	1.9%	59
1911X5	0.246	0.000	4.3%	60
1911R	0.258	0.202	-0.6%	58
1987	0.252	-	-	135
1987X1	0.250	0.152	0.8%	135
1987X5	0.246	0.010	2.3%	134
1987R	0.252	0.997	0.0%	134
J&C	0.253	-	-	23,208
Resnik	0.258	-	-	23,112
Lin	0.251	-	-	19,840
W&P	0.245	-	-	38,721
L&C	0.241	-	-	23,445
H&SO	0.257	-	-	2,452,188
Path	0.241	-	-	22,720
Lesk	0.255	-	-	47,625
Vector	0.263	-	-	32,753
Vct Pair	0.272	-	-	74,803

Table 6.15: Pseudo-word-sense disambiguation error rates and run time.

version was not statistically significant for 1987X1 with $p \approx 0.15$, though it was significant for 1987X5. The 1911X5 version actually gave comparable results to the 1987 version.

The *Roget's*-based methods were actually comparable to the best *WordNet*-based methods. Of interest here is that the Vector-based methods actually performed much worse than any other method. On other problems they had fared quite well.

When it comes to the values of k, I found that $k = 0$ was by far most common. This means that the best way to perform pseudo-word-sense disambiguation is to select the nearest semantically related noun m taken as the object of either v or v'.

The CPU usage is perhaps the most pronounced difference with *Roget's*-based methods, running in a tiny fraction of the time that *WordNet*-based methods require. H&SO took around 28 days to run, showing that this measure simply is not an option for large-scale semantic relatedness problems. Even the fastest *WordNet*-based method – Lin – took around 5 and a half hours. This is over 300 times longer than *Roget's* 1911.

For all systems, a total of 193,192 word pairs must be compared. I also examine the number of necessary comparisons between word senses. If one resource contains a larger number of senses of each word it is measuring semantic distance on, then it will necessarily have to perform many more comparisons. The 1987 *Roget's* required nearly 120 million comparisons, the 1911 *Roget's* – 14.7 million comparisons, while the *WordNet*-based methods – only 3.5 million comparisons. Clearly the implementation of *Roget's* has a very strong advantage when it comes to run time.

6.6 Text Summarization

One of the hardest tasks in Natural Language Processing is text summarization: given a document or a collection of related documents, generate a short – often very short – text which presents only the main points of those documents. Text summarization has been a topic of research even in the earliest days of Artificial Intelligence (Luhn, 1958). There are many variations on this task. For example generic summarization, where there are no restrictions other than the required compression into the most salient points, or a query-driven summarization, where one seeks answers to one or more questions, or focus on the broad topic of the query. Language generation is quite a difficult task, for which no easily applicable tools exist in the public domain; in any event, generation would require the creation of a detailed formal model of the summary, itself a formidable task. That is why summarization systems usually rely on *extracting* a set of relevant sentences and then arranging them into a summary.

The Text Analysis Conference (TAC; formerly Document Understanding Conference, or DUC), organized annually by the National Institute of Standards and Technology (NIST), includes tasks in text summarization. In 2005-2007, the challenge was to generate 250-word summaries of news article collections of 20-50 articles. Summaries were to be built around a query – a few questions on the main topic of the collection and perhaps postulates for how to answer the questions. In 2008-2009 (after a 2007 pilot), the focus has shifted to creating *update summaries*. The document set is split into a few subsets. From each subset, a 100-word summary is generated. The subsets are ordered chronologically, and the goal is to exclude from a summary any information which can be found in a previous document set. For example, given subsets A_1, A_2 and A_3, a summary for A_1, $sum(A_1)$, will be generated normally, while $sum(A_2)$ must not contain any information found in document set A_1. Likewise $sum(A_3)$ should not contain information from document sets A_1 or A_2.

6.6.1 The Data Set

Manual summary evaluation[9] at DUC/TAC, financed by NIST, is an expensive but highly useful part of the exercise. It includes *pyramid evaluation*, outlined in Nenkova and Passonneau (2004), which begins with creating several reference summaries and determining what information they contain. A relevant element is called a Summary Content Unit (SCU), carried in text by a varying-size fragment, between a few words and a complete sentence. All SCUs, marked in the reference summaries, make up a so-called pyramid, with few frequent SCUs at the top and many rare ones at the bottom. In the actual pyramid evaluation, annotators use a custom-made tool to identify SCU occurrences in human written summaries. These human written summaries are often referred to as "peer" summaries. More SCUs mean more relevance for a peer summary; there may be redundancy if a SCU appears more than once. If a peer summary contains relevant information absent from reference summaries, the tool allows the creation of a new SCU. Two kinds of scores measure the quality of the summary after pyramid evaluation: the pyramid score (precision) and the modified pyramid score (recall) (Nenkova and Passonneau, 2004). Only modified pyramid scores are reported in TAC.

One of the primary advantages of pyramid evaluation is that it produces a fully annotated set of peer summaries. Assuming that TAC peers usually build extractive summaries, it becomes feasible to map the sentences from these summaries back to the

[9] See ⟨www.nist.gov/tac/2009/Summarization/update.summ.09.guidelines.html⟩.

<line>*As opposed to the international media hype that surrounded last week's flight, with hundreds of journalists on site to capture the historic moment, Airbus chose to conduct Wednesday's test more discreetly.* <annotation scu-count="2" sum-count="1" sums="0"><scu uid="11" label="Airbus A380 flew its maiden test flight" weight="4"/><scu uid="12" label="taking its maiden flight April 27" weight="3"/></annotation> </line>

<line>*After its glitzy debut, the new Airbus super-jumbo jet A380 now must prove soon it can fly, and eventually turn a profit.*<annotation scu-count="0" sum-count="3" sums="14,44,57"/> </line>

<line>*"The takeoff went perfectly," Alain Garcia, an Airbus engineering executive, told the LCI television station in Paris.*</line>

Figure 6.7: Positive, negative and unlabelled sentence examples for the query "Airbus A380 – Describe developments in the production and launch of the Airbus A380".

original corpus (Copeck and Szpakowicz, 2005). Many sentences in the corpus can be labelled with the list of SCUs they contain, as well as the score for each of these SCUs and their identifiers. Copeck et al. (2006) reported a mapping back to the original corpus of 83% and 96% of the sentences from the peer summaries in 2005 and 2006 respectively. A dataset has been generated for the DUC/TAC main task data in years 2005-2009, and the update task in 2007. This corpus indicates what useful information is included in a sentence and can be used to give sentences scores.

Figure 6.7 illustrates the format of the data. The example comes from the 2008 data set D0801; the goal was to build a summary around the query "Airbus A380 – Describe developments in the production and launch of the Airbus A380". The first sentence is tagged with the <annotation> tag indicating that it was used in at least one summary. This sentence appeared in exactly one summary, with ID 0. There are two SCUs. One, with ID 11, is "Airbus A380 flew its maiden test flight" with a weight of 4. The other, with ID 12, is "taking its maiden flight April 27" with a weight of 2. This is an example of a positive sentence with a weight of 6. The second sentence in Figure 6.7 is annotated but has a SCU count of 0. This means that the sentence was used – in three summaries numbered 14, 44 and 57 – but no SCU is contained in the sentences. Such sentences are negative examples. The third example in Figure 6.7 was not used in any summary, so it has no annotations. I call it an *unlabelled* sentence. The complete SCU-labelled corpus contains 19247 labelled sentences from a total set of 91658; Table 6.16 gives the number

Year	Pos	Neg	Unlabelled	% Labelled
2005	1187	1490	16176	14.2%
2006	988	1368	11642	16.8%
2007	937	975	10670	15.2%
2007-A	201	233	1580	21.5%
2007-B	178	285	955	32.7%
2007-C	164	289	912	33.2%
2008-A	1223	1140	8639	21.5%
2008-B	969	1519	7753	24.3%
2009-A	992	2075	7511	30.0%
2009-B	794	2241	6572	31.6%
Total	7633	11615	72410	21.0%

Table 6.16: Counts of the positive, negative and unlabelled SCU data.

of positive, negative and unlabelled sentences.

Parts of the SCU-labelled corpus have been used in other research. In Nastase and Szpakowicz (2006), the 2005 data are the means for evaluating two sentence-ranking algorithms. In Fuentes et al. (2007), a Support Vector Machine is trained on positive and negative sentences from the 2006 DUC data and tested on the 2005 data. The features include sentence position, lexical overlap with the query and others based on text cohesion.

In Katragadda et al. (2009), the SCU-based corpus is used to find a baseline algorithm for update summarization called Sub-Optimal Position Policy (SPP). This is an extension of Optimal Position Policy (OPP) (Lin and Hovy, 1997) where sentences are selected based on their location in a document. The SCU corpus from 2005-2006 was used for learning SPP, while the 2007 and 2008 data was used for testing.

In Katragadda and Varma (2009), the SCU-labelled corpus from 2005 - 2007 is used to identify whether summaries generated automatically tend to be query-focused or query-biased. A query-focused summary is one built to answer a query, while a query-biased summary is one that selects sentences with as much overlap with the query as possible. It turns out that words found in the query are much more likely to be repeated in machine-generated summaries than in human-made summaries making them query-biased.

6.6.2 Ranking Sentences

I compare my method of sentence ranking against a variety of baselines; these methods are described here.

Roget's **SemDist**

I use *SemDist* – see Section 6.1 – to find the distance between query words and words in the sentence being ranked. The function returns a score in the range 0..18 where 18 is the score when comparing a word with itself and 16 is the highest score between two different words.

A sentence is ranked by its similarity to the query, as determined using the *Roget's SemDist* function. The distance between each word w_j in sentence S is measured against each word q_i in query Q. For each sentence, S, a score, $score(S)$ is calculated, corresponding to the sum of the maximum score of any word in S to each query term q_i, after stop words have been removed[10].

$$score(S) = \sum_{q_i \in Q} max(SemDist(w, q_i) : w \in S)$$

This sum will give an overall weight to the sentence, representing its closeness to the query. $score(S)$ can then be used to rank sentences in order of relevance to the query. This system can in fact be implemented without the use of *SemDist*: just take each score to be either 0 or 18. I ran an experiment with this method as well, which I called Simple Match (SM), and the methods using all versions of *Roget's Thesaurus*. Stop words, as well as punctuation, are removed from both the queries and the sentences. This method tends to favour longer sentences: the longer a sentence is, the more chances it has that one of its words will have a high similarity score to a given word in the query q_i.

I did not experiment with *WordNet* based methods for summarization. Although in theory all of the *WordNet* based MSRs that I examined in the other evaluations could be applied here, as shown in Section 6.5 these measures could take an extremely long time to run. I found that the run time for this program on my laptop was approximately 8 minutes, which would mean that the fastest *WordNet* based method could take over 10 hours. I will leave experimenting with *WordNet* on this task for future work.

[10] I used a 980-element union of five stop lists, first used in Jarmasz (2003): Oracle 8 ConText; SMART; Hyperwave; lists from the University of Kansas and Ohio State University.

Term Frequency – Inverse Sentence Frequency (tf.isf)

Term Frequency – Inverse Document Frequency (tf.idf) has been widely used for document classification. In this system I rank sentences, not documents, so I talk of Term Frequency – Inverse Sentence Frequency (tf.isf). The query is also treated as a single sentence, regardless of how many sentences it actually contains. The term frequency of word t_i in sentence s_j is equal to the number of times the word is found, normalized by the number of words in the sentence. Inverse sentence frequency is the logarithm of the total number of sentences $|S|$ divided by the number of sentences s containing term t_i.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \qquad isf_i = log\frac{|S|}{|\{s : t_i \in s\}|}$$

The weight for t_i in sentence s_j is $weight(t_{i,j}) = tf_{i,j} * isf_i$. Again, stop words and punctuation are ignored. Cosine similarity is used to determine the distance between the query and each sentence. This is similar to what was done by Radev et al. (2004).

Other Baselines

I tested the various *SemDist* methods, the Simple Match method and the tf.isf method, against three other baseline methods. One method is simply to rank sentences based on the number of words it contains. The results of this baseline will be referred to as *Length*. The second method is to order the sentences randomly; I label this method *Random*. The last method, *Ordered*, is to not bother ranking the sentences on any criteria: sentences are selected in the order in which they appear in the data set.

Evaluation and Results

To determine how well the sentence ranker works I evaluate the ranked lists of sentences using Macro-Average Precision. This will give an overall score of how well the sentence ranker separates positive from negative sentences. I used Macro-Average instead of Micro-Average, because the score each sentence receives depends on the query it is answering, so scores are not comparable between document sets. Another method I considered was to measure precision and recall for some cut-off point. The problem is that any cut-off point I chose would be arbitrary and so would not be a good evaluation of the sentence ranker itself.

The calculation of average precision begins by sorting all the sentences in the order of their score. Next, I iterate through the list from highest to lowest, calculating the precision at each positive instance and averaging those precisions.

System	**2005**	**2006**	**2007**	**2007-A**	**2007-B**	**2007-C**
1911	0.582	0.560	0.653	0.702	0.618	0.605
1911X1	0.581	0.561	0.652	0.707	0.606	0.606
1911X5	0.584	0.560	0.652	0.705	0.606	0.605
1911R	0.583	0.558	0.655	0.710	0.617	0.604
1987	0.587	0.562	0.647	0.702	0.585	0.594
1987X1	0.589	0.562	0.648	0.701	0.584	0.593
1987X5	0.590	0.563	0.647	0.701	0.582	0.592
1987R	0.591	0.562	0.649	0.703	0.588	0.595
Simple Match	0.576	0.551	0.623	0.682	0.588	0.610
tf.isf	0.528	0.523	0.599	0.653	0.570	0.579
Length	0.578	0.535	0.607	0.681	0.497	0.591
Random	0.452	0.437	0.530	0.567	0.444	0.401
Ordered	0.431	0.464	0.547	0.588	0.460	0.451

Table 6.17: SCU Rankings for data from 2005-2007.

$$AveP = \frac{\sum_{r=1}^{N} Precision(r) \times rel(r)}{number\ of\ positive\ sentences}$$

$Precision(r)$ is the precision up to sentence r, and $rel(r)$ is a binary function: 1 if sentence r is positive, and 0 otherwise. I included only positive and negative sentences, ignoring unlabelled ones. The macro-average of the average precision is taken for every document in each document set and over all document sets, thus giving the macro-average precision. In Tables 6.17-6.18, I report the macro-average precision for every year – queries and documents, and then take the average over each of them for a given year. I report the average of the average precision for 2005, 2006, 2007, 2007 UpdatePilot, 2008 and 2009 – of the SCU data.

The results show that the expanded lexicon did not consistently improve the results on these data sets. That said, there was some advantage found in using *Roget's Thesaurus*. The 1911 version of *Roget's* scores 5.2% higher than tf.isf or approximately 10% in terms of relative improvement. The improvement of all versions of *Roget's* over the *Random* and *Ordered* baselines is more noticeable, but the *Length* baseline performs very well. Nonetheless it can clearly be seen from these results that the *Roget's*-based methods perform better than the others. There are a total of 277 document sets in the

System	**2008-A**	**2008-B**	**2009-A**	**2009-B**	**Total**
1911	0.663	0.547	0.557	0.435	0.573
1911X1	0.663	0.545	0.556	0.433	0.572
1911X5	0.662	0.545	0.555	0.431	0.572
1911R	0.664	0.548	0.556	0.436	0.574
1987	0.661	0.548	0.549	0.437	0.571
1987X1	0.661	0.547	0.549	0.438	0.571
1987X5	0.661	0.547	0.549	0.437	0.571
1987R	0.660	0.549	0.548	0.437	0.571
Simple Match	0.639	0.533	0.540	0.428	0.558
tf.isf	0.590	0.494	0.506	0.390	0.521
Length	0.652	0.517	0.480	0.418	0.540
Random	0.551	0.455	0.366	0.326	0.445
Ordered	0.551	0.437	0.432	0.350	0.460

Table 6.18: SCU Rankings for data from 2008-2009.

whole data set, which is a suitably high number for determining whether the differences between systems are statistically significant. A paired t-test shows that the difference between the various *Roget's*-based methods is not statistically significant for $p < 0.1$, but the differences between these methods and the *Simple Match* and *tf.isf* methods are statistically significant at $p < 0.01$. This evaluation measure shows a clear advantage of these two methods over the other methods tested.

The *Roget's SemDist*-based method of sentence ranking could be implemented to work with any similarity measure, including those for *WordNet*. I have not yet implemented or tested *WordNet* for sentence ranking. One drawback of using *WordNet* would be the time it would take to run the system. The *SemDist* function is called many times and this program is not particularly fast even in *Roget's*. It may not be feasible to run this sort of program using *WordNet*-based semantic distance measures on a large scale. Section 6.5 demonstrates some of the problems with using *WordNet* for calculating large quantities of semantic distances.

6.7 Conclusion

This chapter shows a good quantity of applications that *Roget's* and *WordNet* can be used on. My findings were that in general the updated versions of *Roget's Thesaurus* performed on par, or better than the original versions. Tasks on which the updates were particularly useful were pseudo-word-sense disambiguation and analogy solving. Likewise the improved thesauri performed well on synonymy identification when the problems focussed on newly added words.

There were some tasks on which the updated thesauri did not improve results. On the text summarization task there was no real difference between the original and updated versions of *Roget's* or for that matter between the 1911 and 1987 versions themselves. Also, on the sentence relatedness data set of Li et al. (2006) no meaningful improvement could be measured. There are a number of possible reasons for this. One explanation why the enhanced lexicon did not help very much on these sentence relatedness exercises is that there are many words in these sentence contributing to the success or failure of the measures. Presumably more common words will already appear in the original versions of the *Thesaurus* and so the newly added words will have less impact. If one considers Zipf's law (Zipf, 1935, 1949) then most of the words added will be from the tail of the distribution and so will have less impact. If this is the case then the differences between the 1911 and 1987 versions of *Roget's* can be explained by slight shifts in the layout of the resource – number of Heads, for example – rather than the size of the lexicon.

In terms of semantic relatedness between words, there appeared to be a small improvement for Spearman's correlation for adding words, though these data sets are too small to say with statistical significance. For the problem of selecting synonyms I found that the improved thesauri 1911X1 and 1987X1 consistently showed improvement while 1911X5 and 1987X5 the improvement was dependent on the data set. All in all, though, *Roget's* 1911X5 and 1987X5 tended to be the best versions of *Roget's*.

In addition I perform a detailed comparison with *WordNet*. *WordNet* results are shown for all applications save text summarization. All the versions of *Roget's* performs quite well, even in comparison to *WordNet*. One of the most striking differences is simply the run time that is required for calculating semantic relatedness between my version of *Roget's* and the WordNet-Similarity Package (Pedersen et al., 2004). This is a testament to the fixed-depth hierarchy in *Roget's* which makes this fast processing of semantic relatedness a real possibility.

In general, *Roget's* will have an advantage over *WordNet* on tasks where relatedness between words of two different parts-of-speech would be useful. An interesting property of *Roget's* is that, using *SemDist*, two words that only appear in different *Roget's* Classes will have a similarity of 0. Thus 0 is the lowest semantic relatedness score and will apply to most word pairs. In comparison, *WordNet* has an arbitrary depth to its hierarchy, thus it is possible that *WordNet* can contain two words that are more distant and so more dissimilar than any other word pair. This would occur when there are two words, at leaf nodes in the hypernym hierarchy, who are further from the root than any other words and whose lowest common ancestor is the root of the hypernym tree. It does not make sense that any specific pair be the least related pair of words out of all word pairs. I mention this as an extreme example of why a fixed hierarchy may be preferred for measuring semantic relatedness over an arbitrary-depth hierarchy. However, the consequence of this is that when measuring relatedness between words that are distantly related *Roget's* may be preferable to *WordNet*.

6.7.1 Future Work

There are many possible applications of *Roget's Thesaurus* and *WordNet*. I have only shown a few of them. Some obvious applications would be to use *Roget's* for real word-sense disambiguation or lexical substitution. *Roget's* has already been used for the construction of lexical chains. Possibly such lexical chains could be applied to summarization or text segmentation as an evaluation criterion. Since *Roget's Thesaurus* contains a large number of opposing concepts it may be possible to apply it to lexical entailment as well.

NLP researchers are always on the hunt for newer and larger data sets on which to train and evaluate their experiments. Many of these experiments will require measuring semantic relatedness. That is why the need for fast semantic relatedness calculation will become more and more important in the coming years. A tool like *Roget's* can provide such fast semantic relatedness measure and so hopefully will become more widely used.

Chapter 7

Concluding Remarks

In this thesis I have described a method of automatically updating *Roget's Thesaurus* with new words. The process I developed is a two step process, where first, lists of semantically related words are generated and second these lists of words are used to identify where in the *Thesaurus* to place a new word. I have found that both of these steps can be enhanced by using the structure of *Roget's Thesaurus*. Each chapter in this thesis contains its own conclusions so this will be a summary of the conclusions from Chapters 4, 5 and 6.

When creating lists of related words I have proposed and evaluated a new technique for measuring semantic relatedness, that enhances both distributional methods using lists of known synonyms. This is described in Chapter 4. This has been shown to have a small, but statistically significant impact on the quality of the MSR. I believe that this system is effectively a type of machine learning as it seems to meet the definition proposed in Mitchell (1997).

The second step – described in Chapter 5 – is to actually add new words to *Roget's Thesaurus*. In this process I generate a list of neighbouring words and use them as anchors to identify where in *Roget's* to place the new word. This process benefits from tuning on the actual *Thesaurus*. The task here is to identify whether a word is a good anchor or not. I experiment with three methods, one using the rank, one using the similarity score and one using a relative similarity score. All in all, I found that rank was the best. The process of adding new words to *Roget's* is a hierarchical one. First the POS is identified, then the Paragraph, then the Semicolon Group. A new Paragraph, or Semicolon Group can be created as need be.

A manual evaluation of my methodology found that the words I added were almost indistinguishable from words already present in the *Thesaurus*. Even after multiple passes the words seemed to find fairly accurate placings. When adding words to a new Paragraph, after one pass the words were highly accurate, however this accuracy fell after multiple passes. In total I added up to 5500 words to the 1911 version and up to 9600 words to the 1987 version.

I also perform a sizeable application based evaluation – described in Chapter 6 – that is used to compare the original and updated *Roget's*. These tasks include semantic relatedness, synonym identification, sentence relatedness, analogy solving, pseudo word sense disambiguation and text summarization. Analogy solving, pseudo-word-sense disambiguation were two applications where the updates to the *Thesaurus* showed a noticeable improvement. I found that generally the additions to *Roget's Thesaurus* improved it for these tasks.

Overall, the goal of using *Roget's Thesaurus* as a source of training data to update itself was accomplished quite successfully. I was able at all stages to show improvements from this process. In Chapter 5 the additions to the thesaurus were shown to be comparable in quality to the words already in *Roget's*. The extrinsic evaluation of Chapter 6 showed more modest improvements, but because most of these tasks do not directly evaluate the new additions, I can still consider these results a success. In all, these experiments have been a success.

7.1 Future Work

Each section contains its own future work, but I will summarize it here. Much of the novelty comes from the new trained semantic relatedness method, and some of the most interesting avenues for future work will be on applying this measure to other problems. This measure represents a real attempt to create customizable semantic relatedness measures that are more useful to specific tasks, as opposed to the more general, catch-all methods that have been more traditionally used. I have attempted to apply this measure to some other tasks, including identifying emotionally related words. Exploring new methods of incorporating training data into semantic relatedness measures I believe is a logical next step for the research area in general.

I have attempted a number of methods of adding words to *Roget's Thesaurus*. These methods are not exhaustive and there may well be other superior ones. This work could also be adapted to adding new words to *WordNet* by identifying synsets, or possible groups of synsets where a target word's neighbours can be found. Although my methods appear specific to *Roget's*, it would not be difficult to move them to other resources.

Finally, I have applied *Roget's* to many NLP tasks, showing its value, particularly on problems of semantic relatedness. One of the key advantages of *Roget's* is the speed at which its API operates, particularly in comparison with *WordNet*. I will not predict which applications NLP researchers or developers will turn to next, but I will predict that they will require more and more processing of semantic relatedness. If this is the case, then *Roget's* will be a natural resource to turn to,because it is of comparable quality to *WordNet* but far superior in terms of how fast it can perform measures of semantic relatedness.

7.2 Software

The final product of this thesis is the actual *Thesaurus* itself, which is available in its original and updated forms via The Open Roget's Project.[1] The tool used for training the semantic distance measures is also available.[2] Both of these are available as Java packages.

[1] http://rogets.site.uottawa.ca

[2] http://www.site.uottawa.ca/~akennedy/Site/Resources.html

Bibliography

Alfonseca, E. (2004). Building phylogenetic lexical Ontologies. In *Proceedings of the 2nd International Semantic Web Conference, ISWC-2004*.

Alfonseca, E. and Manandhar, S. (2002). Extending a lexical ontology by a combination of distributional semantics signatures. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, EKAW '02, pages 1–7, London, UK. Springer-Verlag.

Aversano, L., Marulli, F., and Tortorella, M. (2010). Recovering traceability links between business activities and software components. In Varajão, J. E. Q., Cruz-Cunha, M. M., Putnik, G. D., and Trigo, A., editors, *CENTERIS (1)*, volume 109 of *Communications in Computer and Information Science*, pages 385–394. Springer.

Azarova, I., Mitrofanova, O., Sinopalnikova, A., Yavorskaya, M., and I., O. (2002). Russnet: Building a lexical database for the Russian language. In *Proceedings of the Workshop on Wordnet Structures and Standardization and How this affect Wordnet Applications and Evaluation*, pages 60–64.

Baek, S., Hwang, M., Chung, H., and Kim, P. (2008). Kansei factor space classified by information for Kansei image modeling. *Applied Mathematics and Computation*, 205(2):874–882.

Balkova, V., Suhonogov, A., and Yablonsky, S. (2004). Russian WordNet: From UML-notation to internet/intranet database implementation. In *Proceedings of the Second International WordNet Conference, GWC 2004*, pages 31–38.

Banerjee, S. and Pedersen, T. (2002). An adapted Lesk algorithm for word sense disambiguation using WordNet. In *Proceedings of CICLing 2002*, pages 136–145.

Baroni, M. and Zamparelli, R. (2010). Nouns are vectors, adjectives are matrices: representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1183–1193, Stroudsburg, PA, USA. Association for Computational Linguistics.

Baumgartner, J. L. and Waugh, T. A. (2002). Roget2000: A 2D hyperbolic tree visualization of Roget's Thesaurus. In *In Visualization and Data Analysis. Proceedings of SPIE. 2002*.

BNC (2007). The British National Corpus, version 3 (BNC XML edition). Distributed by Oxford University Computing Services on behalf of the BNC Consortium.

Bodenreider, O. (2004). The Unified Medical Language System (UMLS): Integrating biomedical terminology. *Nucleic Acids Research*, 32(Database-Issue):267–270.

Broda, B., Derwojedowa, M., Piasecki, M., and Szpakowicz, S. (2008). Corpus-based semantic relatedness for the construction of Polish WordNet. In Chair), N. C. C., Choukri, K., Maegaard, B., Mariani, J., Odjik, J., Piperidis, S., and Tapias, D., editors, *Proceedings of the 6th International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.

Broda, B. and Piasecki, M. (2008). SuperMatrix: a general tool for lexical semantic knowledge acquisition. Technical report, Institute of Applied Informatics, Wroclaw University of Technology, Poland.

Broda, B., Piasecki, M., and Szpakowicz, S. (2009). Rank-based transformation in measuring semantic relatedness. In *Canadian AI '09: Proceedings of the 22nd Canadian Conference on Artificial Intelligence*, pages 187–190, Berlin, Heidelberg. Springer-Verlag.

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.

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

Chapman, R. (1977). *Roget's International Thesaurus (4th ed.).* Harper and Row, New York.

Chapman, R. (1992). *Roget's International Thesaurus (5th ed.).* Harper-Collins, New York.

Chernov, S., Iofciu, T., Nejdl, W., and Zhou, X. (2006). Extracting semantic relationships between Wikipedia categories. In *Proceedings of the 1st International Workshop: SemWiki2006 - From Wiki to Semantics (SemWiki 2006), co-located with the ESWC2006 in Budva.*

Clark, S. and Weir, D. (2002). Class-based probability estimation using a semantic hierarchy. *Computational Linguistics*, 28(2):187–206.

Copeck, T., Inkpen, D., Kazantseva, A., Kennedy, A., Kipp, D., Vivi, N., and Szpakowicz, S. (2006). Leveraging DUC. In *HLT-NAACL 2006 - Document Understanding Workshop (DUC).*

Copeck, T., Inkpen, D., Kazantseva, A., Kennedy, A., Kunadze, A., and Szpakowicz, S. (2008). Update summary update. In *the First Text Analysis Conference (TAC 2008).*

Copeck, T., Kennedy, A., Scaiano, M., Inkpen, D., and Szpakowicz, S. (2009). Summarizing with Roget's and with FrameNet. In *First Text Analysis Conference (TAC 2009).*

Copeck, T. and Szpakowicz, S. (2005). Leveraging Pyramids. In *HLT/EMNLP - Document Understanding Workshop (DUC).*

Crouch, C. J. (1988). A cluster-based approach to thesaurus construction. In *SIGIR '88: Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 309–320, New York, NY, USA. ACM.

Crouch, C. J. and Yang, B. (1992). Experiments in automatic statistical thesaurus construction. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '92, pages 77–88, New York, NY, USA. ACM.

Curran, J. R. (2002). Ensemble methods for automatic thesaurus extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 222–229.

Curran, J. R. (2003). *From Distributional to Semantic Similarity.* PhD thesis, Institute for Communicating and Collaborative Systems School of Informatics University of Edinburgh.

Curran, J. R. and Moens, M. (2002). Improvements in automatic thesaurus extraction. In *Proceedings of the Workshop of the ACL Special Interest Group on the Lexicon (SIGLEX)*, pages 59–66.

Dagan, Ido, Lee, L., and Pereira, F. (1999). Similarity-based models of word cooccurrence probabilities. *Machine Learning Journal*, 34(1–3):43–69.

de Melo, G. and Weikum, G. (2008). Mapping Roget's Thesaurus and WordNet to French. In *Proceedings of the 6th Language Resources and Evaluation Conference (LREC 2008)*, Marrakech, Morocco. European Language Resources Association.

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.

Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302.

Esuli, A. and Sebastiani, F. (2006). SENTIWORDNET: A publicly available lexical resource for opinion mining. In *Proceedings of the 5th Conference on Language Resources and Evaluation (LREC 2006*, pages 417–422.

Evert, S. (2004). The statistics of word co-occurrences: word pairs and collocations. *Doctoral dissertation, Institut für maschinelle Sprachverarbeitung, Universität Stuttgart.*

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

Feng, J., Zhou, Y., and Martin, T. (2008). Sentence similarity based on relevance. In *Proceedings of International Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*, pages 832–839.

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.

Firth, J. R. (1957). A synopsis of linguistic theory 1930-55. *Studies in Linguistic Analysis (special volume of the Philological Society)*, 1952-59:1–32.

Fuentes, M., Alfonseca, E., and Rodríguez, H. (2007). Support vector machines for query-focused summarization trained and evaluated on pyramid data. In *ACL '07: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 57–60, Morristown, NJ, USA. Association for Computational Linguistics.

Gabrilovich, E. and Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based Explicit Semantic Analysis. *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 6–12.

Gabrilovich, E. and Markovitch, S. (2009). Wikipedia-based semantic interpretation for natural language processing. *Journal of Artificial Intelligence Research*, 34(1):443–498.

Gale, W. A., Church, K., and Yarowsky, D. (1992). Work on statistical methods for word sense disambiguation.

Geffet, M. and Dagan, I. (2004). Feature vector quality and distributional similarity. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 247, Morristown, NJ, USA. Association for Computational Linguistics.

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.

Grefenstette, G. (1994). *Explorations in Automatic Thesaurus Discovery.* Kluwer Academic Publishers, Norwell, MA, USA.

Haghighi, A., Liang, P., Berg-Kirkpatrick, T., and Klein, D. (2008). Learning bilingual lexicons from monolingual corpora. In *Proceedings of The Association of Computational Linguistics: Human Language Technologies*, pages 771–779, Columbus, Ohio. Association for Computational Linguistics.

Hagiwara, M., Ogawa, Y., and Toyama, K. (2005). Supervised synonym acquisition using distributional features and syntactic patterns. *Journal of Natural Language Processing.*, 16:59–83.

Hajishirzi, H., Yih, W.-T., and Kolcz, A. (2010). Adaptive near-duplicate detection via similarity learning. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 419–426, New York, NY, USA. ACM.

Harris, Z. (1954). Distributional structure. *Word*, 10(23):146–162.

Hassan, S. and Mihalcea, R. (2009). Cross-lingual semantic relatedness using encyclopedic knowledge. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009*, pages 1192–1201. ACL.

Hassan, S. and Mihalcea, R. (2011). Semantic relatedness using salient semantic analysis. In Burgard, W. and Roth, D., editors, *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*. AAAI Press.

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

Hindle, D. (1990). Noun classification from predicate-argument structures. In *Proceedings of the 28th annual meeting on Association for Computational Linguistics*, pages 268–275, Morristown, NJ, USA. Association for Computational Linguistics.

Hirst, G. (2004). Ontology and the lexicon. In Staab, S. and Studer, R., editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 209–230. Springer.

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.

Ho, C., Murad, M. A. A., Kadir, R. A., and Doraisamy, S. C. (2010). Word sense disambiguation-based sentence similarity. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 418–426, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hovy, E., Kozareva, Z., and Riloff, E. (2009). Toward completeness in concept extraction and classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, pages 948–957, Stroudsburg, PA, USA. Association for Computational Linguistics.

Islam, A. and Inkpen, D. (2006). Second order co-occurrence PMI for determining the semantic similarity of words. In *Proceedings of the International Conference on Language Resources and Evaluation*, pages 1033–1038, Genoa, Italy.

Islam, A. and Inkpen, D. (2007). Semantic similarity of short texts. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2007)*.

Islam, A., Milios, E., and Keselj, V. (2012). Text similarity using Google tri-grams. In Inkpen, D. and Kosseim, L., editors, *Advances in Artificial Intelligence - 24th Canadian Conference on Artificial Intelligence, Canadian AI 2012*, pages 312–317, Toronto, Ontario, Canada. Springer.

Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579.

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 NAACL workshop on WordNet and Other Lexical Resources Workshop*, page 186 188.

Jarmasz, M. and Szpakowicz, S. (2003). 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. (2004). Roget's Thesaurus and semantic similarity. In Nicolov, N., Bontcheva, K., Angelova, G., and Mitkov, R., editors, *Recent Advances in Natural Language Processing III: Selected Papers from RANLP 2003, Cur-*

rent Issues in Linguistic Theory, volume 260, pages 111–120. John Benjamins, Amsterdam/Philadelphia.

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

Joubarne, C. and Inkpen, D. (2011). Comparison of semantic similarity for different languages using the Google n-gram corpus and second-order co-occurrence measures. In *Proceedings of the 24th Canadian conference on Advances in artificial intelligence*, Canadian AI'11, pages 216–221, Berlin, Heidelberg. Springer-Verlag.

Kassner, L., Nastase, V., and Strube, M. (2008). Acquiring a taxonomy from the German Wikipedia. In Chair), N. C. C., Choukri, K., Maegaard, B., Mariani, J., Odjik, J., Piperidis, S., and Tapias, D., editors, *Proceedings of the 6th International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.

Katragadda, R., Pingali, P., and Varma, V. (2009). Sentence position revisited: a robust light-weight update summarization 'baseline' algorithm. In *CLIAWS3 '09: Proceedings of the 3rd International Workshop on Cross Lingual Information Access*, pages 46–52, Morristown, NJ, USA. Association for Computational Linguistics.

Katragadda, R. and Varma, V. (2009). Query-focused summaries or query-biased summaries? In *Proceedings of ACL-IJCNLP 2009 Conference Short Papers*, pages 105–108, Suntec, Singapore. Association for Computational Linguistics.

Kendall, J. C. (2008). *The Man Who Made Lists: Love, Death, Madness, and the Creation of Roget's Thesaurus.* G. P. Putnam's Son, New York.

Kennedy, A. (2007). Analysis and construction of noun hypernym hierarchies to enhance Roget's Thesaurus. Master's thesis, The University of Ottawa.

Kennedy, A. (2010). Automatically expanding the lexicon of Roget's Thesaurus. In *Proceedings of the Graduate Symposium at Canadian AI 2010*, pages 410–411, Ottawa, Ontario, Canada. Springer.

Kennedy, A., Copeck, T., Inkpen, D., and Szpakowicz, S. (2010). Entropy-based sentence selection with Roget's Thesaurus. In *Proceedings of the 3rd Text Analysis Conference (TAC 2010).*

Kennedy, A., Kazantseva, A., Inkpen, D., and Szpakowicz, S. (2012). Getting emotional about news summarization. In Inkpen, D. and Kosseim, L., editors, *Advances in Artificial Intelligence - 24th Canadian Conference on Artificial Intelligence, Canadian AI 2012*, pages 121–132, Toronto, Ontario, Canada. Springer.

Kennedy, A., Kazantseva, A., Mohammad, S., Copeck, T., Inkpen, D., and Szpakowicz, S. (2011). Getting emotional about news. In *Proceedings of the 4th Text Analysis Conference (TAC 2011)*.

Kennedy, A. and Szpakowicz, S. (2007). Disambiguating hypernym relations for Roget's Thesaurus. In *Proceedings of Text, Speech and Dialogue, 10th International Conference, TSD 2007*, pages 66–75. Springer.

Kennedy, A. and Szpakowicz, S. (2008). Evaluating Roget's Thesauri. In *Proceedings of ACL-08: HLT*, pages 416–424. Association for Computational Linguistics.

Kennedy, A. and Szpakowicz, S. (2010a). Evaluation of a sentence ranker for text summarization based on Roget's Thesaurus. In *Proceedings of Text, Speech and Dialogue, TSD 2010*, pages 101–108.

Kennedy, A. and Szpakowicz, S. (2010b). Towards a gold standard for extractive text summarization. In *Proceedings of Canadian AI 2010*, pages 51–62, Ottawa, Ontario, Canada. Springer.

Kennedy, A. and Szpakowicz, S. (2011). A supervised method of feature weighting for measuring semantic relatedness. In *Proceedings of Canadian AI 2011*, pages 222–233, St. John's, Newfoundland, Canada. Springer.

Kennedy, A. and Szpakowicz, S. (2012a). Fast semantic relatedness: WordNet::Similarity vs Roget's Thesaurus. In Bailis, P. and Sherry, J., editors, *In Tiny Transactions on Computer Science*, volume 1.

Kennedy, A. and Szpakowicz, S. (2012b). Supervised distributional semantic relatedness. In *Proceedings of Text, Speech and Dialogue, TSD 2012*. Springer.

Kilgarriff, A. (1997). I don't believe in word senses. *Computers and the Humanities*, 31(2):91–113.

Kilgarriff, A. (2003). Thesauruses for natural language processing. In *Proceedings of the 2003 International Conference on Natural Language Processing and Knowledge Engineering, 2003*, pages 5–13.

Kilgarriff, A. and Tugwell, D. (2001). WASP-bench: an MT lexicographers' workstation supporting state-of-the-art lexical disambiguation. In *Proceedings of the MT Summit VII*, pages 198–190.

Kilgarriff, A. and Yallop, C. (2000). What's in a Thesaurus? Technical Report ITRI-00-28, Information Technology Research Institute, University of Brighton. Also published in Proceedings of the 2nd Conference on Language Resources and Evaluation, pp. 1371-1379.

Kirkpatrick, B., editor (1987). *Roget's Thesaurus of English Words and Phrases*. Longman, Harlow.

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

Kozareva, Z. and Hovy, E. (2010). A semi-supervised method to llearn and construct taxonomies using the web. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1110–1118, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kozareva, Z., Riloff, E., and Hovy, E. (2008). Semantic class learning from the web with hyponym pattern linkage graphs. In *Proceedings of ACL-08: HLT*, pages 1048–1056, Columbus, Ohio. Association for Computational Linguistics.

Krippendorff, K. (1980). *Content Analysis: An Introduction to Its Methodology*. Sage, Beverly Hills, CA, 2 edition.

Krippendorff, K. (2004). *Content Analysis: An Introduction to Its Methodology*. Sage, Thousand Oaks, CA.

Kwong, O. Y. (1998a). Aligning wordnet with additional lexical resources. In *Proceedings of 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.

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.

Lee, L. (1999). Measures of distributional similarity. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 25–32, Morristown, NJ, USA. Association for Computational Linguistics.

Lemnitzer, L., Wunsch, H., and Gupta, P. (2008). Enriching GermaNet with verb-noun relations - a case study of lexical acquisition. In Chair), N. C. C., Choukri, K., Maegaard, B., Mariani, J., Odjik, J., Piperidis, S., and Tapias, D., editors, *Proceedings of the 6th International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.

Leong, C. W. and Mihalcea, R. (2011a). Going beyond text: A hybrid image-text approach for measuring word relatedness. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1403–1407, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Leong, C. W. and Mihalcea, R. (2011b). Measuring the semantic relatedness between words and images. In *Proceedings of the Ninth International Conference on Computational Semantics*, IWCS '11, pages 185–194, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Li, Y., McLean, D., Bandar, Z., O'Shea, J., and Crockett, K. A. (2006). Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering*, 18(8):1138–1150.

Lin, C.-Y. and Hovy, E. (1997). Identifying topics by position. In *Proceedings of the 5th conference on Applied natural language processing*, pages 283–290, Morristown, NJ, USA. Association for Computational Linguistics.

Lin, D. (1998a). Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics*, pages 768–774, Morristown, NJ, USA. Association for Computational Linguistics.

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

Liu, X., Zhou, Y., and Zheng, R. (2007). Sentence similarity based on dynamic time warping. In *Proceedings of the International Conference on Semantic Computing*, ICSC '07, pages 250–256, Washington, DC, USA. IEEE Computer Society.

Liu, Y., McInnes, B. T., Pedersen, T., Melton-Meaux, G., and Pakhomov, S. V. S. (2012). Semantic relatedness study using second order co-occurrence vectors computed from biomedical corpora, UMLS and WordNet. In Luo, G., Liu, J., and Yang, C. C., editors, *IHI*, pages 363–372. ACM.

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2:159–165.

Magnini, B. and Cavagliá, G. (2000). Integrating subject field codes into WordNet. In *Proceedings of LREC-2000, Second International Conference on Language Resources and Evaluation*, pages 1413–1418.

Mandala, R., Tokunaga, T., and Tanaka, H. (1999). Complementing WordNet with Roget's and corpus-based thesauri for information retrieval. In *Proceedings of the 9th conference on European chapter of the Association for Computational Linguistics*, pages 94–101, Morristown, NJ, USA. Association for Computational Linguistics.

Mann, G. S. (2002). Fine-grained proper noun ontologies for question answering. In *Proceedings of the 2002 workshop on Building and using semantic networks - Volume*

11, SEMANET '02, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.

Masterman, M. (1956). The potentialities of a mechanical thesaurus. *Machine Translation.*

Masterman, M. (1961). Translation. In *Proceedings of the Aristotelian Society*, pages 169–216.

Mihalcea, R., Corley, C., and Strapparava, C. (2006). Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence conference (AAAI 2006).* AAAI Press.

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

Mitchell, J. and Lapata, M. (2010). Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429.

Mitchell, T. (1997). *Machine Learning.* McGraw-Hill.

Mititelu, V. B., Bozianu, L., and Mihăilă, C. (2006). Romanian WordNet: New developments and applications. In *Proceedings of the Third International WordNet Conference, Jeju Island, Korea*, page 337347.

Mohammad, S., Gurevych, I., Hirst, G., and Zesch, T. (2007). Cross-lingual distributional profiles of concepts for measuring semantic distance. In *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 571–580. ACL.

Mohammad, S. and Hirst, G. (2006a). Determining word sense dominance using a thesaurus. In *EACL*. The Association for Computer Linguistics.

Mohammad, S. and Hirst, G. (2006b). Distributional measures of concept-distance: A task-oriented evaluation. In Jurafsky, D. and Gaussier, É., editors, *EMNLP*, pages 35–43. ACL.

Mohammad, S. and Hirst, G. (2006c). Distributional measures of semantic distance: A survey. Unpublished Manuscript.

Mohammad, S. and Turney, P. (2012). Crowdsourcing a word-emotion association lexicon. *Submitted to Computational Intelligence*.

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.

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.

Nastase, V. and Szpakowicz, S. (2006). A study of two graph algorithms in topic-driven summarization. In *Proceedings of the TextGraphs 2006, workshop at NAACL 2006*. Association for Computational Linguistics.

Nenkova, A. and Passonneau, R. J. (2004). Evaluating content selection in summarization: The pyramid method. In *HLT-NAACL*, pages 145–152.

Ofoghi, B. and Yearwood, J. (2010). Learning parse-free event-based features for textual entailment recognition. In Li, J., editor, *Australasian Conference on Artificial Intelligence*, volume 6464 of *Lecture Notes in Computer Science*, pages 184–193. Springer.

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

Old, L. J. (2002). Information cartography applied to the semantics of Roget's Thesaurus. In *Proceedings, 13th Midwest Artificial Intelligence and Cognitive Science Conference (MAICS'02)*.

Old, L. J. (2003). An analysis of semantic overlap among English prepositions in Roget's Thesaurus. In *Proceedings of the Association for Computational Linguistics SIG Semantics Conference (ACL-SIGSEM)*, pages 13–19.

Old, L. J. (2004). Unlocking the semantics of Roget's Thesaurus using formal concept analysis. In Eklund, P. W., editor, *ICFCA*, volume 2961 of *Lecture Notes in Computer Science*, pages 244–251. Springer.

Old, L. J. (2009). The semantic structure of Roget's Thesaurus cross-references. In *Proceedings of the SENSE Workshop on conceptual Structures for Extracting Natural language Semantics*.

O'Shea, J., Bandar, Z., Crockett, K., and McLean, D. (2008). A comparative study of two short text semantic similarity measures. In *Proceedings of the 2nd KES International conference on Agent and multi-agent systems: technologies and applications*, KES-AMSTA'08, pages 172–181, Berlin, Heidelberg. Springer-Verlag.

Padó, S. and Lapata, M. (2007). Dependency-based Construction of Semantic Space Models. *Computational Linguistics*, 33(2):161–199.

Pantel, P. (2005). Inducing ontological co-occurrence vectors. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 125–132. The Association for Computer Linguistics.

Pantel, P. and Lin, D. (2002). Discovering Word Senses From Text. In *KDD '02: Proc. eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 613–619, New York, NY, USA. ACM.

Pantel, P. and Pennacchiotti, M. (2008). Automatically harvesting and ontologizing semantic relations. In Buitelaar, P. and Cimiano, P., editors, *Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, volume 167 of *Frontiers in Artificial Intelligence and Applications*, pages 171–195. IOS Press.

Pantel, P. A. (2003). *Clustering by Committee*. PhD thesis, University of Alberta.

Patwardhan, S. (2003). Incorporating dictionary and corpus information into a vector measure of semantic relatedness. Master's thesis, University of Minnesota, Duluth.

Patwardhan, S., Banerjee, S., and Pedersen, T. (2003). Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the 4th International Conference on Intelligent Text Processing and Computational Linguistics*, pages 241–257.

Pedersen, T., Patwardhan, S., and Michelizzi., J. (2004). Wordnet::Similarity - Measuring the relatedness of concepts. In *Proceedings of the 19th National Conference on Artificial Intelligence.*, pages 1024–1025.

Piasecki, M., Broda, B., Marcińczuk, M., and Szpakowicz, S. (2009a). The WordNet weaver: Multi-criteria voting for semi-automatic extension of a wordet. In *Canadian AI '09: Proceedings of the 22nd Canadian Conference on Artificial Intelligence*, pages 237–240, Berlin, Heidelberg. Springer-Verlag.

Piasecki, M., Szpakowicz, S., and Broda, B. (2007). Automatic selection of heterogeneous syntactic features in semantic similarity of Polish nouns. In Matousek, V. and Mautner, P., editors, *TSD*, volume 4629 of *Lecture Notes in Computer Science*, pages 99–106. Springer.

Piasecki, M., Szpakowicz, S., and Broda, B. (2009b). *A WordNet from the Ground Up*. Wrocław University of Technology Press. `www.site.uottawa.ca/~szpak/pub/A_Wordnet_from_the_Ground_Up.zip`.

Ponzetto, S. P. and Strube, M. (2007). Deriving a large scale taxonomy from Wikipedia. In *AAAI'07: Proceedings of the 22nd national conference on Artificial intelligence*, pages 1440–1445. AAAI Press.

Prince, V. and Chauch, J. (2008). Building a bilingual representation of the Roget Thesaurus for French to English machine translation. In Chair), N. C. C., Choukri, K., Maegaard, B., Mariani, J., Odjik, J., Piperidis, S., and Tapias, D., editors, *Proceedings of the 6th International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.

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

Purandare, A. and Pedersen, T. (2004). Senseclusters - finding clusters that represent word senses. In McGuinness, D. L. and Ferguson, G., editors, *AAAI*, pages 1030–1031. AAAI Press / The MIT Press.

Radev, D. R., Jing, H., Styś, M., and Tam, D. (2004). Centroid-based summarization of multiple documents. *Information Processing Management*, 40(6):919–938.

Radinsky, K., Agichtein, E., Gabrilovich, E., and Markovitch, S. (2011). A word at a time: Computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 337–346, New York, NY, USA. ACM.

Razavi, A. H., Matwin, S., Inkpen, D., and Kouznetsov, A. (2009). Parameterized contrast in second order soft co-occurrences: A novel text representation technique in text mining and knowledge extraction. In *ICDMW '09: Proceedings of the 2009 IEEE International Conference on Data Mining Workshops*, pages 471–476, Washington, DC, USA. IEEE Computer Society.

Rees, R. v. (2003). Clarity in the usage of the terms Ontology, Taxonomy and Classification. In *Paper w78- 2003-432 in the Construction Informatics Digital Library.* Available on-line at http://vanrees.org/phd/Cib78ConferencePaper2003.

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

Rooth, M., Riezler, S., Prescher, D., Carroll, G., and Beil, F. (1999). Inducing a semantically annotated lexicon via EM-based clustering. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 104–111, Morristown, NJ, USA. Association for Computational Linguistics.

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

Ruge, G. (1997). Automatic detection of thesaurus relations for information retrieval applications. In *Foundations of Computer Science: Potential - Theory - Cognition, to Wilfried Brauer on the occasion of his sixtieth birthday*, pages 499–506, London, UK. Springer-Verlag.

Rychlý, P. and Kilgarriff, A. (2007). An efficient algorithm for building a distributional thesaurus (and other Sketch Engine developments). In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proc. Demo and Poster Sessions*, pages 41–44, Prague, Czech Republic. Association for Computational Linguistics.

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.

Saias, J. and Quaresma, P. (2002). Semantic enrichment of a web legal information retrieval system. In *In Trevor Bench-Capon, Aspassia Daskalopulu, and Radboud Winkels, editors, Legal Knowledge and Information Systems. IOS*, pages 11–20. Press.

Salton, G. (1971). *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Salton, G. and McGill, M. J. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill.

Sang, E. T. K. (2007). Extracting hypernym pairs from the web. In *ACL '07: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 165–168, Morristown, NJ, USA. Association for Computational Linguistics.

Schütze, H. (1998). Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.

Schütze, H. and Pedersen, J. O. (1997). A co-occurrence-based thesaurus and two applications to information retrieval. *Information Processing Management*, 33(3):307–318.

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.

Simina, M. and Barbu, C. (2004). Meta latent semantic analysis. In *Proceedings of the IEEE International Conference on Systems, Man & Cybernetics*, pages 3720–3724, The Hague, Netherlands. IEEE.

Sinclair, J. (2001). *Collins Cobuild English Dictionary for Advanced Learners*. Harper Collins Pub.

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 *Proceedings of the 2nd International Workshop on Active Mining (AM2003) (In Conjunction with the International Symposium on Methodologies for Intelligent Systems)*, pages 7–13.

Strapparava, C. and Valitutti, A. (2004). WordNet-Affect: an affective extension of WordNet. In Gavrilidou, M., Crayannis, G., Markantonatu, S., Piperidis, S., and Stainhaouer, G., editors, *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 1083–1086.

Sumida, A., Yoshinaga, N., and Torisawa, K. (2008). Boosting precision and recall of hyponymy relation acquisition from hierarchical layouts in Wikipedia. In *Proc. 6th International Language Resources and Evaluation (LREC)*, Marrakech, Morocco.

Takenobu, T., Makoto, I., and Hozumi, T. (1995). Automatic thesaurus construction based on grammatical relations. In *IJCAI'95: Proceedings of the 14th international joint conference on Artificial intelligence*, pages 1308–1313, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Temperley, D. and Sleator, D. (1993). Parsing English with a link grammar. In *Proc. 3rd International Workshop on Parsing Technologies*.

Tseng, Y.-H. (2002). Automatic Thesaurus Generation for Chinese Documents. *J. Am. Soc. Inf. Sci. Technol.*, 53(13):1130–1138.

Tsurumaru, H., Hitaka, T., and Yoshida, S. (1986). An attempt to automatic thesaurus construction from an ordinary Japanese language dictionary. In *Proceedings of the 11th coference on Computational linguistics*, COLING '86, pages 445–447, Stroudsburg, PA, USA. Association for Computational Linguistics.

Turney, P. (2001). Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the 12th European Conference on Machine Learning (ECML-2001)*, pages 491–502.

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

Turney, P. D. (2002). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. *CoRR*, cs.LG/0212032.

Turney, P. D. (2005). Measuring semantic similarity by latent relational analysis. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05)*, pages 1136–1141, Edinburgh, Scotland.

Turney, P. D. (2012). Domain and function: A dual-space model of semantic relations and compositions. *Journal of Artificial Intelligence Research (JAIR)*, 44:533–585.

Turney, P. D., Neuman, Y., Assaf, D., and Cohen, Y. (2011). Literal and metaphorical sense identification through concrete and abstract context. In *Proc. Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 680–690, Stroudsburg, PA, USA. Association for Computational Linguistics.

Turney, P. D. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

Veale, T. (2003). Pathways to creativity in lexical Ontologies. In Sojka, P., Pala, K., Smrž, P., Fellbaum, C., and Vossen, P., editors, *Proceedings of the 2nd International WordNet Conference—GWC 2004*, pages 220–225, Brno, Czech Republic.

Vossen, P., editor (1998). *EuroWordNet: a Multilingual Database with Lexical Semantic Networks*. Kluwer Academic Publishers, Norwell, MA, USA.

Vyas, V. and Pantel, P. (2008). Explaining similarity of terms. In Scott, D. and Uszkoreit, H., editors, *COLING (Posters)*, pages 131–134.

Ward, G. (1996). An improved method for deriving word meaning from lexical co-occurrence.

Weeds, J. and Weir, D. (2005). Co-occurrence retrieval: A flexible framework for lexical distributional similarity. *Computational Linguistics.*, 31(4):439–475.

Widdows, D. and Ferraro, K. (2008). Semantic vectors: a scalable open source package and online technology management application. In (ELRA), E. L. R. A., editor, *Proceedings of the 6th International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco.

Wilks, Y. (1998). Language processing and the thesaurus.

Wille, R. (1981). Restructuring lattice theory: An approach based on hierarchies of concepts. *Ordered Sets, Ivan Rival Ed., NATO Advanced Study Institute*, 83:445–470.

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

Wu, F. and Weld, D. S. (2008). Automatically refining the Wikipedia infobox ontology. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 635–644, New York, NY, USA. ACM.

Wu, Z. and Palmer, M. (1994). Verb semantics and lexical selection. In *Proceedings of the 32nd. Annual Meeting of the Association for Computational Linguistics*, pages 133 –138, New Mexico State University, Las Cruces, New Mexico.

Yamada, I., Torisawa, K., Kazama, J., Kuroda, K., Murata, M., De Saeger, S., Bond, F., and Sumida, A. (2009). Hypernym discovery based on distributional similarity and hierarchical structures. In *EMNLP '09: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 929–937, Morristown, NJ, USA. Association for Computational Linguistics.

Yang, D. and Powers, D. M. (2008). Automatic thesaurus construction. In Dobbie, G. and Mans, B., editors, *Proceedings of the Thirty-First Australasian Computer Science Conference (ACSC 2008)*, volume 74 of *CRPIT*, pages 147–156, Wollongong, NSW, Australia. ACS.

Yarowsky, D. (1992). Word-sense disambiguation using statistical models of roget's categories trained on large corpora. In *Proceedings of the 14th conference on Computational linguistics*, pages 454–460, Morristown, NJ, USA. Association for Computational Linguistics.

Yih, W.-T. (2009). Learning term-weighting functions for similarity measures. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, pages 793–802, Morristown, NJ, USA. Association for Computational Linguistics.

Yoshida, S., Yukawa, T., and Kuwabara, K. (2003). Constructing and examining personalized co-occurrence-based thesauri on web pages. In *WWW (Posters)*.

Zhang, Z., Gentile, A. L., and Ciravegna, F. (2011). Harnessing different knowledge sources to measure semantic relatedness under a uniform model. In *Proc. 2011 Conference on Empirical Methods in Natural Language Processing*, pages 991–1002, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Zheng, H., Wu, X., and Yu, Y. (2008). Enriching WordNet with Folksonomies. In *Proceedings of the 12th Pacific-Asia conference on Advances in knowledge discovery and data mining*, PAKDD'08, pages 1075–1080, Berlin, Heidelberg. Springer-Verlag.

Zheng, H.-T., Borchert, C., and Kim, H.-G. (2009). Exploiting corpus-related Ontologies for conceptualizing document corpora. *Journal of the American Society for Information Science and Technology*, 60:2287–2299.

Zhitomirsky-Geffet, M. and Dagan, I. (2009). Bootstrapping distributional feature vector quality. *Computational Linguistics*, 35(3):435–461.

Zipf, G. K. (1935). *The Psychobiology of Language*. Houghton-Mifflin, New York, NY, USA.

Zipf, G. K. (1949). *Human Behavior and the Principle of Least Effort*. Addison-Wesley.

Appendix A

Semantic Relatedness

Full results on the initial tuning data with all measures of association. Results are for the unsupervised MSR (Table A.1), supervised *Roget's* 1911 MSR (Table A.2), supervised *Roget's* 1987 MSR (Table A.3) and supervised *WordNet* MSR (Table A.4).

				Top	Top	Top	Top	Top	Top
POS	Group	Training	Association	1	5	10	20	50	100
noun	SG	Unsupervised	Dice	0.172	0.124	0.104	0.084	0.059	0.042
			PMI	0.336	0.211	0.165	0.122	0.079	0.055
			T-score	0.274	0.155	0.113	0.084	0.052	0.036
			Z-score	0.191	0.143	0.119	0.095	0.067	0.048
			LL	0.129	0.074	0.053	0.039	0.025	0.019
			χ^2	0.113	0.084	0.074	0.060	0.046	0.035
verb	SG	Unsupervised	Dice	0.198	0.153	0.128	0.102	0.074	0.060
			PMI	0.332	0.206	0.155	0.117	0.081	0.061
			T-score	0.257	0.161	0.122	0.091	0.062	0.046
			Z-score	0.200	0.154	0.129	0.104	0.076	0.061
			LL	0.125	0.080	0.063	0.049	0.039	0.031
			χ^2	0.127	0.094	0.080	0.068	0.054	0.045
adj	SG	Unsupervised	Dice	0.172	0.118	0.094	0.071	0.047	0.033
			PMI	0.338	0.198	0.147	0.101	0.061	0.041
			T-score	0.273	0.162	0.116	0.081	0.049	0.034
			Z-score	0.185	0.119	0.096	0.074	0.050	0.035
			LL	0.157	0.087	0.065	0.047	0.031	0.023
			χ^2	0.090	0.080	0.063	0.052	0.038	0.027

POS	Group	Training	Association	Top 1	Top 5	Top 10	Top 20	Top 50	Top 100
noun	Para		Dice	0.328	0.297	0.276	0.249	0.213	0.181
			PMI	0.550	0.447	0.389	0.339	0.266	0.218
			T-score	0.448	0.341	0.289	0.241	0.183	0.147
			Z-score	0.340	0.308	0.289	0.263	0.223	0.191
			LL	0.251	0.187	0.155	0.132	0.105	0.088
			χ^2	0.247	0.215	0.207	0.190	0.165	0.144
verb	Para	Unsupervised	Dice	0.400	0.372	0.347	0.318	0.279	0.250
			PMI	0.538	0.450	0.394	0.350	0.293	0.252
			T-score	0.462	0.369	0.324	0.279	0.231	0.198
			Z-score	0.402	0.361	0.344	0.321	0.284	0.258
			LL	0.302	0.237	0.207	0.188	0.167	0.150
			χ^2	0.292	0.271	0.255	0.244	0.222	0.207
adj	Para	Unsupervised	Dice	0.317	0.274	0.240	0.208	0.168	0.139
			PMI	0.548	0.415	0.342	0.281	0.210	0.167
			T-score	0.440	0.340	0.284	0.229	0.171	0.136
			Z-score	0.305	0.256	0.239	0.209	0.171	0.144
			LL	0.268	0.186	0.163	0.140	0.115	0.099
			χ^2	0.188	0.186	0.174	0.157	0.135	0.117
noun	POS	Unsupervised	Dice	0.432	0.412	0.398	0.378	0.345	0.314
			PMI	0.632	0.557	0.511	0.470	0.408	0.361
			T-score	0.537	0.443	0.397	0.351	0.294	0.253
			Z-score	0.455	0.418	0.409	0.390	0.352	0.321
			LL	0.336	0.290	0.257	0.230	0.198	0.175
			χ^2	0.347	0.331	0.326	0.312	0.284	0.261
verb	POS	Unsupervised	Dice	0.463	0.444	0.428	0.403	0.371	0.342
			PMI	0.603	0.523	0.475	0.434	0.377	0.338
			T-score	0.532	0.442	0.400	0.357	0.306	0.274
			Z-score	0.470	0.445	0.431	0.410	0.376	0.353
			LL	0.390	0.322	0.287	0.267	0.243	0.226
			χ^2	0.380	0.362	0.345	0.331	0.309	0.296
adj	POS	Unsupervised	Dice	0.372	0.331	0.298	0.266	0.224	0.193
			PMI	0.592	0.477	0.405	0.341	0.271	0.226
			T-score	0.472	0.394	0.341	0.285	0.224	0.186
			Z-score	0.337	0.310	0.293	0.265	0.227	0.200
			LL	0.305	0.230	0.208	0.186	0.162	0.143
			χ^2	0.228	0.238	0.226	0.210	0.186	0.168

Table A.1: Unsupervised Results

				Top	Top	Top	Top	Top	Top
POS	Group	Training	Association	1	5	10	20	50	100
noun	SG	1911-context	Dice	0.100	0.052	0.040	0.030	0.020	0.015
			PMI	0.184	0.101	0.074	0.054	0.035	0.025
			T-score	0.033	0.023	0.018	0.014	0.010	0.008
			Z-score	0.050	0.029	0.021	0.017	0.012	0.009
			LL	0.013	0.009	0.008	0.006	0.006	0.006
			χ^2	0.020	0.009	0.006	0.005	0.005	0.004
noun	SG	1911-relation	Dice	0.100	0.053	0.039	0.029	0.019	0.014
			PMI	0.149	0.081	0.060	0.043	0.028	0.020
			T-score	0.090	0.042	0.032	0.024	0.017	0.013
			Z-score	0.090	0.048	0.034	0.026	0.018	0.014
			LL	0.050	0.023	0.018	0.016	0.012	0.010
			χ^2	0.050	0.025	0.020	0.017	0.013	0.010
verb	SG	1911-context	Dice	0.103	0.067	0.056	0.046	0.036	0.030
			PMI	0.178	0.108	0.086	0.067	0.048	0.039
			T-score	0.088	0.068	0.058	0.046	0.036	0.029
			Z-score	0.125	0.072	0.057	0.046	0.036	0.030
			LL	0.065	0.049	0.039	0.036	0.030	0.026
			χ^2	0.057	0.048	0.041	0.034	0.028	0.026
verb	SG	1911-relation	Dice	0.098	0.070	0.054	0.044	0.035	0.028
			PMI	0.142	0.088	0.072	0.057	0.041	0.034
			T-score	0.103	0.068	0.057	0.049	0.037	0.030
			Z-score	0.102	0.066	0.056	0.048	0.036	0.030
			LL	0.095	0.063	0.051	0.041	0.032	0.027
			χ^2	0.088	0.057	0.049	0.039	0.031	0.027
adj	SG	1911-context	Dice	0.103	0.068	0.050	0.036	0.023	0.017
			PMI	0.165	0.093	0.064	0.044	0.029	0.021
			T-score	0.062	0.038	0.032	0.025	0.017	0.013
			Z-score	0.055	0.043	0.033	0.024	0.017	0.013
			LL	0.028	0.016	0.013	0.012	0.010	0.008
			χ^2	0.027	0.018	0.013	0.012	0.008	0.006
adj	SG	1911-relation	Dice	0.137	0.079	0.057	0.040	0.025	0.018
			PMI	0.167	0.091	0.065	0.045	0.029	0.020
			T-score	0.137	0.079	0.057	0.040	0.025	0.017
			Z-score	0.142	0.083	0.057	0.040	0.026	0.018
			LL	0.142	0.077	0.054	0.039	0.024	0.017
			χ^2	0.147	0.080	0.059	0.040	0.025	0.017

				Top	Top	Top	Top	Top	Top
POS	Group	Training	Association	1	5	10	20	50	100
noun	Para	1911-context	Dice	0.234	0.167	0.144	0.121	0.097	0.084
			PMI	0.323	0.249	0.214	0.180	0.143	0.120
			T-score	0.110	0.099	0.090	0.079	0.067	0.060
			Z-score	0.139	0.113	0.098	0.085	0.073	0.064
			LL	0.072	0.061	0.058	0.054	0.050	0.048
			χ^2	0.063	0.051	0.043	0.040	0.038	0.036
noun	Para	1911-relation	Dice	0.230	0.168	0.142	0.122	0.098	0.083
			PMI	0.328	0.228	0.193	0.161	0.126	0.104
			T-score	0.217	0.143	0.123	0.107	0.088	0.075
			Z-score	0.217	0.157	0.129	0.115	0.093	0.080
			LL	0.140	0.102	0.088	0.079	0.070	0.062
			χ^2	0.151	0.106	0.092	0.084	0.072	0.064
verb	Para	1911-context	Dice	0.270	0.244	0.218	0.197	0.171	0.156
			PMI	0.373	0.308	0.274	0.242	0.210	0.191
			T-score	0.273	0.230	0.213	0.193	0.174	0.160
			Z-score	0.300	0.237	0.216	0.194	0.176	0.161
			LL	0.245	0.211	0.190	0.179	0.165	0.152
			χ^2	0.213	0.206	0.189	0.176	0.162	0.152
verb	Para	1911-relation	Dice	0.280	0.242	0.217	0.196	0.170	0.155
			PMI	0.360	0.270	0.243	0.218	0.188	0.170
			T-score	0.305	0.234	0.221	0.203	0.176	0.159
			Z-score	0.297	0.232	0.218	0.201	0.175	0.157
			LL	0.265	0.218	0.199	0.180	0.161	0.148
			χ^2	0.272	0.209	0.191	0.177	0.158	0.147
adj	Para	1911-context	Dice	0.215	0.160	0.137	0.116	0.095	0.080
			PMI	0.277	0.203	0.168	0.139	0.113	0.094
			T-score	0.163	0.129	0.116	0.100	0.084	0.073
			Z-score	0.167	0.133	0.117	0.100	0.084	0.073
			LL	0.095	0.076	0.069	0.066	0.057	0.053
			χ^2	0.090	0.074	0.064	0.059	0.050	0.045
adj	Para	1911-relation	Dice	0.248	0.190	0.157	0.132	0.105	0.087
			PMI	0.293	0.198	0.167	0.144	0.115	0.094
			T-score	0.243	0.186	0.151	0.130	0.103	0.084
			Z-score	0.260	0.187	0.149	0.129	0.102	0.084
			LL	0.250	0.183	0.148	0.124	0.097	0.079
			χ^2	0.252	0.184	0.153	0.127	0.097	0.081

				Top	Top	Top	Top	Top	Top
POS	Group	Training	Association	1	5	10	20	50	100
noun	POS	1911-context	Dice	0.330	0.274	0.249	0.225	0.196	0.177
			PMI	0.419	0.358	0.325	0.291	0.253	0.225
			T-score	0.219	0.194	0.183	0.168	0.151	0.140
			Z-score	0.241	0.210	0.194	0.177	0.161	0.147
			LL	0.142	0.130	0.129	0.125	0.119	0.117
			χ^2	0.148	0.115	0.104	0.098	0.093	0.092
noun	POS	1911-relation	Dice	0.331	0.278	0.247	0.227	0.198	0.178
			PMI	0.434	0.337	0.305	0.270	0.232	0.206
			T-score	0.317	0.248	0.224	0.205	0.181	0.165
			Z-score	0.322	0.264	0.234	0.217	0.189	0.172
			LL	0.239	0.204	0.188	0.173	0.156	0.146
			χ^2	0.252	0.208	0.192	0.179	0.160	0.149
verb	POS	1911-context	Dice	0.337	0.321	0.294	0.272	0.245	0.228
			PMI	0.463	0.397	0.357	0.329	0.293	0.274
			T-score	0.347	0.303	0.284	0.270	0.249	0.235
			Z-score	0.368	0.311	0.294	0.274	0.251	0.234
			LL	0.332	0.292	0.268	0.258	0.243	0.228
			χ^2	0.298	0.288	0.267	0.255	0.239	0.227
verb	POS	1911-relation	Dice	0.340	0.321	0.292	0.271	0.244	0.228
			PMI	0.437	0.350	0.324	0.295	0.267	0.248
			T-score	0.388	0.323	0.307	0.289	0.255	0.237
			Z-score	0.382	0.324	0.305	0.285	0.254	0.236
			LL	0.352	0.304	0.282	0.259	0.238	0.223
			χ^2	0.367	0.295	0.276	0.257	0.236	0.222
adj	POS	1911-context	Dice	0.237	0.198	0.176	0.155	0.134	0.118
			PMI	0.322	0.246	0.209	0.182	0.156	0.135
			T-score	0.197	0.165	0.153	0.138	0.122	0.110
			Z-score	0.198	0.168	0.153	0.139	0.123	0.110
			LL	0.133	0.109	0.104	0.100	0.089	0.084
			χ^2	0.127	0.111	0.100	0.092	0.083	0.076
adj	POS	1911-relation	Dice	0.290	0.232	0.203	0.177	0.148	0.127
			PMI	0.325	0.243	0.212	0.188	0.158	0.137
			T-score	0.282	0.234	0.199	0.177	0.145	0.124
			Z-score	0.297	0.233	0.195	0.174	0.145	0.124
			LL	0.293	0.234	0.198	0.170	0.139	0.119
			χ^2	0.298	0.232	0.200	0.172	0.140	0.121

Table A.2: 1911 Supervised Results

				Top	Top	Top	Top	Top	Top
POS	Group	Training	Association	1	5	10	20	50	100
noun	SG	1987-context	Dice	0.089	0.053	0.041	0.029	0.020	0.015
			PMI	0.173	0.093	0.068	0.047	0.030	0.022
			T-score	0.028	0.017	0.013	0.010	0.008	0.007
			Z-score	0.034	0.020	0.016	0.013	0.010	0.008
			LL	0.010	0.006	0.006	0.005	0.004	0.004
			χ^2	0.013	0.007	0.005	0.005	0.004	0.004
noun	SG	1987-relation	Dice	0.095	0.051	0.039	0.028	0.019	0.014
			PMI	0.129	0.070	0.053	0.038	0.025	0.018
			T-score	0.078	0.039	0.029	0.021	0.014	0.011
			Z-score	0.080	0.041	0.030	0.022	0.015	0.012
			LL	0.039	0.023	0.017	0.014	0.010	0.008
			χ^2	0.044	0.023	0.020	0.015	0.011	0.009
verb	SG	1987-context	Dice	0.107	0.067	0.057	0.046	0.036	0.030
			PMI	0.192	0.123	0.094	0.074	0.053	0.042
			T-score	0.077	0.061	0.052	0.045	0.036	0.030
			Z-score	0.097	0.068	0.057	0.048	0.037	0.030
			LL	0.063	0.044	0.040	0.036	0.029	0.025
			χ^2	0.073	0.044	0.043	0.037	0.029	0.025
verb	SG	1987-relation	Dice	0.087	0.069	0.055	0.044	0.034	0.029
			PMI	0.145	0.087	0.071	0.056	0.042	0.034
			T-score	0.105	0.069	0.058	0.045	0.036	0.030
			Z-score	0.105	0.068	0.057	0.045	0.035	0.030
			LL	0.080	0.060	0.049	0.041	0.032	0.027
			χ^2	0.082	0.056	0.046	0.040	0.032	0.027
adj	SG	1987-context	Dice	0.102	0.058	0.044	0.031	0.022	0.016
			PMI	0.158	0.089	0.063	0.047	0.030	0.021
			T-score	0.053	0.031	0.026	0.021	0.016	0.012
			Z-score	0.060	0.034	0.025	0.023	0.016	0.013
			LL	0.033	0.019	0.015	0.012	0.009	0.007
			χ^2	0.042	0.025	0.018	0.014	0.010	0.008
adj	SG	1987-relation	Dice	0.125	0.071	0.049	0.035	0.022	0.016
			PMI	0.178	0.090	0.065	0.046	0.030	0.021
			T-score	0.140	0.075	0.054	0.038	0.024	0.017
			Z-score	0.137	0.077	0.057	0.039	0.025	0.017
			LL	0.115	0.071	0.052	0.037	0.023	0.016
			χ^2	0.123	0.072	0.054	0.038	0.024	0.017

				Top	Top	Top	Top	Top	Top
POS	Group	Training	Association	1	5	10	20	50	100
noun	Para	1987-context	Dice	0.222	0.163	0.143	0.120	0.096	0.082
			PMI	0.318	0.248	0.208	0.171	0.134	0.112
			T-score	0.072	0.073	0.070	0.063	0.056	0.052
			Z-score	0.100	0.081	0.077	0.070	0.061	0.056
			LL	0.047	0.047	0.045	0.042	0.040	0.039
			χ^2	0.058	0.047	0.042	0.039	0.038	0.037
noun	Para	1987-relation	Dice	0.232	0.160	0.138	0.119	0.096	0.081
			PMI	0.290	0.210	0.177	0.146	0.114	0.095
			T-score	0.182	0.130	0.112	0.100	0.079	0.068
			Z-score	0.191	0.137	0.119	0.103	0.083	0.071
			LL	0.119	0.094	0.081	0.073	0.063	0.057
			χ^2	0.136	0.100	0.090	0.078	0.065	0.058
verb	Para	1987-context	Dice	0.290	0.237	0.223	0.196	0.172	0.157
			PMI	0.412	0.330	0.291	0.257	0.221	0.198
			T-score	0.272	0.230	0.212	0.192	0.172	0.159
			Z-score	0.290	0.241	0.219	0.198	0.176	0.161
			LL	0.187	0.182	0.179	0.171	0.157	0.148
			χ^2	0.222	0.197	0.187	0.175	0.158	0.148
verb	Para	1987-relation	Dice	0.275	0.240	0.214	0.192	0.169	0.154
			PMI	0.367	0.278	0.249	0.222	0.193	0.173
			T-score	0.292	0.240	0.222	0.198	0.173	0.157
			Z-score	0.292	0.235	0.218	0.199	0.173	0.156
			LL	0.262	0.224	0.200	0.179	0.162	0.148
			χ^2	0.258	0.216	0.190	0.176	0.160	0.147
adj	Para	1987-context	Dice	0.215	0.154	0.131	0.112	0.092	0.079
			PMI	0.265	0.201	0.167	0.140	0.114	0.096
			T-score	0.165	0.120	0.107	0.096	0.080	0.071
			Z-score	0.167	0.124	0.103	0.095	0.081	0.071
			LL	0.085	0.075	0.068	0.062	0.054	0.048
			χ^2	0.088	0.087	0.074	0.064	0.055	0.049
adj	Para	1987-relation	Dice	0.235	0.181	0.151	0.126	0.100	0.085
			PMI	0.302	0.200	0.171	0.146	0.118	0.098
			T-score	0.237	0.180	0.152	0.131	0.102	0.086
			Z-score	0.237	0.179	0.150	0.130	0.103	0.085
			LL	0.218	0.178	0.147	0.125	0.097	0.081
			χ^2	0.223	0.180	0.147	0.125	0.099	0.081

				Top	Top	Top	Top	Top	Top
POS	Group	Training	Association	1	5	10	20	50	100
noun	POS	1987-context	Dice	0.325	0.273	0.249	0.222	0.194	0.174
			PMI	0.432	0.362	0.321	0.284	0.240	0.214
			T-score	0.181	0.165	0.156	0.145	0.133	0.126
			Z-score	0.209	0.174	0.165	0.155	0.140	0.131
			LL	0.119	0.107	0.102	0.100	0.098	0.098
			χ^2	0.120	0.107	0.098	0.095	0.093	0.093
noun	POS	1987-relation	Dice	0.328	0.269	0.243	0.221	0.195	0.175
			PMI	0.398	0.317	0.288	0.255	0.216	0.193
			T-score	0.287	0.233	0.213	0.195	0.170	0.155
			Z-score	0.302	0.244	0.221	0.201	0.174	0.158
			LL	0.220	0.191	0.173	0.163	0.148	0.136
			χ^2	0.246	0.199	0.184	0.168	0.150	0.139
verb	POS	1987-context	Dice	0.363	0.314	0.299	0.273	0.248	0.229
			PMI	0.485	0.418	0.378	0.343	0.306	0.283
			T-score	0.340	0.302	0.285	0.268	0.248	0.233
			Z-score	0.378	0.314	0.294	0.276	0.252	0.236
			LL	0.260	0.257	0.255	0.247	0.230	0.219
			χ^2	0.297	0.277	0.264	0.250	0.230	0.219
verb	POS	1987-relation	Dice	0.340	0.314	0.290	0.267	0.242	0.227
			PMI	0.445	0.365	0.333	0.303	0.275	0.254
			T-score	0.380	0.332	0.310	0.282	0.253	0.234
			Z-score	0.387	0.330	0.306	0.282	0.252	0.234
			LL	0.350	0.309	0.279	0.260	0.239	0.223
			χ^2	0.350	0.299	0.272	0.256	0.237	0.222
adj	POS	1987-context	Dice	0.258	0.198	0.174	0.154	0.131	0.117
			PMI	0.293	0.242	0.212	0.186	0.157	0.139
			T-score	0.205	0.159	0.146	0.134	0.118	0.107
			Z-score	0.203	0.164	0.145	0.136	0.119	0.109
			LL	0.122	0.114	0.102	0.095	0.086	0.078
			χ^2	0.143	0.122	0.109	0.097	0.087	0.079
adj	POS	1987-relation	Dice	0.280	0.228	0.200	0.171	0.142	0.125
			PMI	0.330	0.245	0.219	0.194	0.163	0.140
			T-score	0.280	0.229	0.198	0.177	0.144	0.124
			Z-score	0.278	0.224	0.196	0.175	0.144	0.124
			LL	0.262	0.230	0.195	0.171	0.139	0.119
			χ^2	0.267	0.227	0.194	0.171	0.140	0.120

Table A.3: 1987 Supervised Results

				Top	Top	Top	Top	Top	Top
POS	Group	Training	Association	1	5	10	20	50	100
noun	SG	WN-context	Dice	0.109	0.062	0.045	0.031	0.020	0.015
			PMI	0.173	0.094	0.068	0.047	0.030	0.022
			T-score	0.030	0.019	0.014	0.011	0.009	0.008
			Z-score	0.046	0.026	0.019	0.014	0.011	0.009
			LL	0.011	0.010	0.007	0.006	0.006	0.006
			χ^2	0.032	0.011	0.008	0.006	0.004	0.004
noun	SG	WN-relation	Dice	0.108	0.059	0.042	0.031	0.021	0.016
			PMI	0.132	0.070	0.051	0.038	0.024	0.018
			T-score	0.086	0.041	0.030	0.022	0.015	0.012
			Z-score	0.087	0.046	0.033	0.024	0.017	0.013
			LL	0.044	0.023	0.018	0.014	0.010	0.008
			χ^2	0.044	0.026	0.020	0.016	0.012	0.010
verb	SG	WN-context	Dice	0.098	0.066	0.055	0.047	0.037	0.030
			PMI	0.212	0.120	0.095	0.074	0.054	0.043
			T-score	0.092	0.063	0.054	0.046	0.036	0.030
			Z-score	0.117	0.075	0.063	0.051	0.039	0.031
			LL	0.060	0.039	0.036	0.031	0.026	0.023
			χ^2	0.053	0.040	0.035	0.031	0.025	0.023
verb	SG	WN-relation	Dice	0.093	0.069	0.054	0.044	0.034	0.028
			PMI	0.145	0.092	0.075	0.058	0.042	0.034
			T-score	0.107	0.070	0.057	0.046	0.035	0.030
			Z-score	0.103	0.068	0.054	0.044	0.034	0.029
			LL	0.095	0.060	0.049	0.040	0.033	0.027
			χ^2	0.087	0.055	0.046	0.039	0.032	0.027
adj	SG	WN-context	Dice	0.082	0.052	0.042	0.032	0.021	0.016
			PMI	0.163	0.077	0.059	0.044	0.029	0.020
			T-score	0.045	0.030	0.025	0.021	0.014	0.011
			Z-score	0.042	0.031	0.025	0.021	0.015	0.012
			LL	0.020	0.013	0.012	0.011	0.008	0.007
			χ^2	0.043	0.026	0.019	0.014	0.010	0.007
adj	SG	WN-relation	Dice	0.147	0.079	0.057	0.042	0.026	0.018
			PMI	0.172	0.091	0.066	0.047	0.029	0.020
			T-score	0.140	0.074	0.053	0.038	0.023	0.016
			Z-score	0.138	0.076	0.055	0.038	0.023	0.016
			LL	0.093	0.052	0.040	0.029	0.019	0.015
			χ^2	0.097	0.053	0.040	0.029	0.019	0.015

				Top	Top	Top	Top	Top	Top
POS	Group	Training	Association	1	5	10	20	50	100
noun	Para	WN-context	Dice	0.233	0.169	0.143	0.124	0.099	0.083
			PMI	0.338	0.239	0.200	0.164	0.130	0.108
			T-score	0.084	0.083	0.073	0.065	0.059	0.053
			Z-score	0.105	0.094	0.085	0.075	0.065	0.058
			LL	0.052	0.054	0.052	0.050	0.048	0.046
			χ^2	0.090	0.052	0.046	0.039	0.035	0.032
noun	Para	WN-relation	Dice	0.252	0.176	0.149	0.127	0.100	0.085
			PMI	0.283	0.206	0.174	0.144	0.113	0.096
			T-score	0.195	0.136	0.119	0.102	0.083	0.071
			Z-score	0.216	0.145	0.127	0.109	0.087	0.075
			LL	0.124	0.096	0.087	0.075	0.063	0.057
			χ^2	0.144	0.108	0.093	0.081	0.068	0.061
verb	Para	WN-context	Dice	0.280	0.239	0.217	0.199	0.175	0.158
			PMI	0.427	0.329	0.294	0.261	0.223	0.201
			T-score	0.255	0.222	0.212	0.193	0.174	0.160
			Z-score	0.283	0.241	0.222	0.205	0.181	0.164
			LL	0.183	0.177	0.169	0.158	0.145	0.138
			χ^2	0.177	0.170	0.164	0.152	0.142	0.134
verb	Para	WN-relation	Dice	0.265	0.240	0.216	0.192	0.169	0.154
			PMI	0.360	0.282	0.252	0.223	0.193	0.174
			T-score	0.277	0.239	0.218	0.197	0.172	0.157
			Z-score	0.282	0.230	0.214	0.192	0.171	0.156
			LL	0.267	0.225	0.199	0.181	0.162	0.148
			χ^2	0.275	0.206	0.187	0.174	0.158	0.147
adj	Para	WN-context	Dice	0.197	0.157	0.133	0.112	0.092	0.078
			PMI	0.305	0.194	0.165	0.140	0.110	0.093
			T-score	0.138	0.114	0.101	0.089	0.075	0.066
			Z-score	0.145	0.113	0.102	0.087	0.075	0.067
			LL	0.095	0.068	0.063	0.059	0.053	0.049
			χ^2	0.117	0.087	0.071	0.061	0.052	0.044
adj	Para	WN-relation	Dice	0.260	0.191	0.162	0.135	0.109	0.092
			PMI	0.295	0.203	0.170	0.142	0.114	0.096
			T-score	0.233	0.185	0.151	0.125	0.097	0.082
			Z-score	0.232	0.187	0.152	0.124	0.098	0.082
			LL	0.207	0.159	0.130	0.108	0.089	0.077
			χ^2	0.212	0.156	0.130	0.108	0.091	0.079

				Top	Top	Top	Top	Top	Top
POS	Group	Training	Association	1	5	10	20	50	100
noun	POS	WN-context	Dice	0.330	0.276	0.249	0.226	0.195	0.175
			PMI	0.430	0.349	0.312	0.275	0.237	0.211
			T-score	0.185	0.174	0.162	0.148	0.136	0.127
			Z-score	0.212	0.189	0.176	0.161	0.144	0.134
			LL	0.130	0.128	0.122	0.120	0.116	0.114
			χ^2	0.157	0.119	0.106	0.097	0.089	0.086
noun	POS	WN-relation	Dice	0.354	0.290	0.256	0.229	0.199	0.178
			PMI	0.383	0.313	0.281	0.249	0.215	0.192
			T-score	0.301	0.242	0.220	0.200	0.175	0.159
			Z-score	0.331	0.254	0.230	0.204	0.179	0.164
			LL	0.233	0.195	0.181	0.164	0.147	0.136
			χ^2	0.258	0.215	0.192	0.172	0.154	0.143
verb	POS	WN-context	Dice	0.352	0.317	0.294	0.276	0.251	0.231
			PMI	0.515	0.412	0.381	0.351	0.312	0.288
			T-score	0.337	0.295	0.288	0.271	0.249	0.234
			Z-score	0.367	0.316	0.300	0.283	0.256	0.239
			LL	0.247	0.243	0.237	0.227	0.214	0.207
			χ^2	0.252	0.240	0.228	0.217	0.209	0.200
verb	POS	WN-relation	Dice	0.327	0.316	0.293	0.269	0.244	0.226
			PMI	0.445	0.371	0.340	0.306	0.278	0.256
			T-score	0.370	0.325	0.302	0.279	0.252	0.235
			Z-score	0.380	0.320	0.298	0.274	0.249	0.234
			LL	0.340	0.308	0.281	0.260	0.237	0.223
			χ^2	0.360	0.292	0.268	0.252	0.234	0.223
adj	POS	WN-context	Dice	0.235	0.197	0.173	0.155	0.134	0.117
			PMI	0.347	0.239	0.211	0.185	0.154	0.135
			T-score	0.172	0.153	0.139	0.129	0.113	0.102
			Z-score	0.183	0.152	0.141	0.127	0.115	0.104
			LL	0.130	0.106	0.099	0.093	0.086	0.081
			χ^2	0.150	0.124	0.105	0.092	0.082	0.073
adj	POS	WN-relation	Dice	0.315	0.240	0.210	0.183	0.153	0.134
			PMI	0.333	0.247	0.216	0.189	0.159	0.139
			T-score	0.285	0.229	0.194	0.170	0.139	0.123
			Z-score	0.297	0.230	0.196	0.169	0.141	0.123
			LL	0.267	0.203	0.174	0.152	0.132	0.117
			χ^2	0.275	0.203	0.178	0.154	0.135	0.120

Table A.4: WordNet 3.0 Supervised Results

Appendix B

Emotion and Sentiment Evaluation

Full results for sentiment and emotional relatedness experiments. This shows the scores for every individual sentiment and emotion. Sentiment results are found in Table B.1 while emotion results are found in Table B.2.

			Top	Top	Top	Top	Top	Top
POS	Training	Sentiment	1	5	10	20	50	100
N.	none	positive	1.000	1.000	1.000	0.998	0.847	0.477
		negative	1.000	1.000	1.000	1.000	0.888	0.507
	PMI	positive	1.000	0.999	0.992	0.958	0.813	0.611
		negative	1.000	0.998	0.996	0.968	0.853	0.697
	relation	positive	1.000	1.000	1.000	1.000	0.903	0.580
		negative	1.000	1.000	1.000	1.000	0.952	0.645
	relation-combined	positive	1.000	0.999	0.993	0.958	0.814	0.611
		negative	1.000	0.998	0.996	0.965	0.851	0.696
	context	positive	1.000	0.999	0.995	0.986	0.855	0.524
		negative	1.000	1.000	0.996	0.960	0.729	0.482
	context-combined	positive	1.000	1.000	0.989	0.934	0.768	0.574
		negative	1.000	0.999	0.994	0.968	0.832	0.686
VB.	none	positive	1.000	1.000	1.000	1.000	0.966	0.559
		negative	1.000	1.000	1.000	1.000	0.999	0.836
	PMI	positive	1.000	1.000	1.000	1.000	0.951	0.653
		negative	1.000	1.000	1.000	1.000	0.997	0.902
	relation	positive	1.000	1.000	1.000	1.000	0.990	0.599
		negative	1.000	1.000	1.000	1.000	0.999	0.852
	relation-combined	positive	1.000	1.000	1.000	0.999	0.950	0.671
		negative	1.000	1.000	1.000	1.000	0.998	0.923
	context	positive	1.000	1.000	1.000	1.000	0.995	0.711
		negative	1.000	1.000	1.000	1.000	0.998	0.855

POS	Training	Sentiment	Top 1	Top 5	Top 10	Top 20	Top 50	Top 100
	context-combined	positive	1.000	1.000	1.000	1.000	0.932	0.622
		negative	1.000	1.000	1.000	1.000	0.998	0.915
ADJ.	none	positive	1.000	1.000	1.000	0.994	0.893	0.565
		negative	1.000	1.000	1.000	0.993	0.958	0.824
	PMI	positive	1.000	1.000	1.000	0.992	0.915	0.707
		negative	1.000	1.000	1.000	0.992	0.943	0.802
	relation	positive	1.000	1.000	1.000	0.998	0.909	0.630
		negative	1.000	1.000	1.000	0.998	0.964	0.844
	relation-combined	positive	1.000	1.000	1.000	0.993	0.917	0.711
		negative	1.000	1.000	1.000	0.992	0.939	0.787
	context	positive	1.000	1.000	1.000	0.997	0.892	0.626
		negative	1.000	1.000	1.000	0.996	0.936	0.721
	context	positive	1.000	1.000	1.000	0.992	0.895	0.692
		negative	1.000	1.000	1.000	0.993	0.945	0.804

Table B.1: Sentiment Similarity

POS	Training	Emotion	Top 1	Top 5	Top 10	Top 20	Top 50	Top 100
N.	none	anger	1.000	1.000	0.997	0.857	0.412	0.206
		anticipation	1.000	1.000	0.980	0.755	0.326	0.163
		disgust	1.000	1.000	1.000	0.883	0.386	0.193
		fear	1.000	1.000	1.000	0.993	0.713	0.359
		joy	1.000	0.993	0.950	0.748	0.339	0.170
		sadness	1.000	1.000	0.983	0.882	0.451	0.225
		surprise	1.000	1.000	0.933	0.623	0.251	0.125
		trust	1.000	1.000	1.000	0.978	0.668	0.336
N.	PMI	anger	1.000	0.980	0.933	0.842	0.629	0.449
		anticipation	1.000	0.993	0.927	0.812	0.475	0.239
		disgust	1.000	1.000	0.943	0.763	0.412	0.218
		fear	1.000	1.000	1.000	0.965	0.733	0.475
		joy	1.000	0.967	0.920	0.775	0.477	0.246
		sadness	1.000	0.960	0.897	0.792	0.595	0.391
		surprise	1.000	0.940	0.790	0.555	0.237	0.119
		trust	1.000	1.000	0.983	0.913	0.770	0.503

			Top	Top	Top	Top	Top	Top
POS	Training	Emotion	1	5	10	20	50	100
N.	relation	anger	1.000	1.000	0.993	0.947	0.543	0.272
		anticipation	1.000	1.000	0.990	0.838	0.375	0.188
		disgust	1.000	1.000	1.000	0.935	0.469	0.235
		fear	1.000	1.000	1.000	1.000	0.856	0.449
		joy	1.000	1.000	0.987	0.885	0.423	0.212
		sadness	1.000	1.000	0.993	0.938	0.549	0.284
		surprise	1.000	0.987	0.937	0.760	0.323	0.161
		trust	1.000	1.000	1.000	0.995	0.748	0.396
N.	relation-combined	anger	1.000	0.987	0.937	0.838	0.629	0.449
		anticipation	1.000	0.993	0.930	0.813	0.471	0.237
		disgust	1.000	0.993	0.937	0.752	0.405	0.214
		fear	1.000	1.000	1.000	0.965	0.739	0.479
		joy	1.000	0.973	0.920	0.775	0.476	0.245
		sadness	1.000	0.953	0.893	0.790	0.597	0.392
		surprise	1.000	0.940	0.803	0.555	0.239	0.119
		trust	1.000	1.000	0.983	0.913	0.771	0.510
N.	context	anger	1.000	1.000	0.950	0.688	0.366	0.188
		anticipation	1.000	1.000	0.970	0.823	0.414	0.207
		disgust	0.967	0.933	0.760	0.525	0.241	0.121
		fear	1.000	0.993	0.973	0.795	0.451	0.231
		joy	1.000	1.000	0.960	0.712	0.311	0.155
		sadness	1.000	0.987	0.930	0.757	0.427	0.224
		surprise	1.000	0.980	0.827	0.537	0.216	0.108
		trust	1.000	1.000	0.993	0.942	0.714	0.376
N.	context-combined	anger	1.000	0.947	0.897	0.782	0.597	0.443
		anticipation	1.000	0.980	0.920	0.818	0.523	0.274
		disgust	1.000	0.987	0.923	0.710	0.399	0.210
		fear	1.000	1.000	0.990	0.940	0.723	0.484
		joy	0.967	0.947	0.873	0.737	0.474	0.248
		sadness	1.000	0.953	0.903	0.813	0.607	0.406
		surprise	0.967	0.907	0.763	0.530	0.257	0.129
		trust	1.000	1.000	0.963	0.885	0.762	0.519
VB.	none	anger	1.000	1.000	1.000	1.000	0.832	0.433
		anticipation	1.000	1.000	1.000	0.997	0.525	0.263
		disgust	1.000	1.000	1.000	0.977	0.449	0.225
		fear	1.000	1.000	1.000	1.000	0.848	0.427
		joy	1.000	1.000	1.000	0.940	0.429	0.215
		sadness	1.000	1.000	1.000	0.963	0.509	0.255
		surprise	1.000	1.000	1.000	0.930	0.415	0.207
		trust	1.000	1.000	1.000	1.000	0.695	0.347

			Top	Top	Top	Top	Top	Top
POS	Training	Emotion	1	5	10	20	50	100
VB.	PMI	anger	1.000	1.000	1.000	0.983	0.844	0.559
		anticipation	1.000	1.000	1.000	0.993	0.663	0.332
		disgust	1.000	1.000	0.967	0.940	0.493	0.247
		fear	1.000	1.000	1.000	1.000	0.841	0.547
		joy	1.000	1.000	1.000	0.927	0.467	0.233
		sadness	1.000	1.000	1.000	0.987	0.773	0.408
		surprise	1.000	1.000	0.993	0.860	0.472	0.236
		trust	1.000	1.000	1.000	1.000	0.853	0.493
VB.	relation	anger	1.000	1.000	1.000	1.000	0.896	0.494
		anticipation	1.000	1.000	1.000	0.997	0.657	0.333
		disgust	1.000	1.000	1.000	0.863	0.413	0.207
		fear	1.000	1.000	1.000	1.000	0.863	0.447
		joy	1.000	1.000	1.000	0.987	0.459	0.229
		sadness	1.000	1.000	1.000	0.980	0.507	0.253
		surprise	1.000	1.000	1.000	0.917	0.429	0.215
		trust	1.000	1.000	1.000	1.000	0.764	0.390
VB.	relation-combined	anger	1.000	1.000	1.000	1.000	0.919	0.657
		anticipation	1.000	1.000	1.000	1.000	0.672	0.336
		disgust	1.000	1.000	1.000	0.950	0.504	0.252
		fear	1.000	1.000	1.000	1.000	0.892	0.610
		joy	1.000	1.000	1.000	0.893	0.433	0.217
		sadness	1.000	1.000	1.000	1.000	0.856	0.465
		surprise	1.000	1.000	1.000	0.877	0.493	0.247
		trust	1.000	1.000	1.000	0.983	0.845	0.481
VB.	context	anger	1.000	1.000	1.000	1.000	0.912	0.563
		anticipation	1.000	1.000	1.000	0.997	0.607	0.305
		disgust	1.000	1.000	1.000	0.930	0.444	0.222
		fear	1.000	1.000	1.000	1.000	0.891	0.487
		joy	1.000	1.000	1.000	0.967	0.480	0.240
		sadness	1.000	1.000	1.000	1.000	0.721	0.361
		surprise	1.000	1.000	1.000	0.903	0.405	0.203
		trust	1.000	1.000	1.000	1.000	0.816	0.411
VB.	context-combined	anger	1.000	1.000	1.000	0.993	0.867	0.575
		anticipation	1.000	1.000	1.000	0.987	0.703	0.355
		disgust	1.000	1.000	1.000	0.987	0.567	0.283
		fear	1.000	1.000	1.000	1.000	0.847	0.587
		joy	1.000	1.000	0.967	0.893	0.436	0.218
		sadness	1.000	1.000	1.000	0.997	0.829	0.455
		surprise	1.000	1.000	1.000	0.853	0.532	0.266
		trust	1.000	1.000	1.000	0.993	0.857	0.515

			Top	Top	Top	Top	Top	Top
POS	Training	Emotion	1	5	10	20	50	100
ADJ.	none	anger	1.000	0.993	0.967	0.920	0.528	0.264
		anticipation	1.000	0.993	0.933	0.653	0.262	0.131
		disgust	1.000	0.987	0.977	0.967	0.747	0.386
		fear	1.000	0.987	0.977	0.942	0.594	0.299
		joy	1.000	0.973	0.927	0.803	0.369	0.184
		sadness	1.000	1.000	0.987	0.910	0.526	0.263
		surprise	1.000	0.980	0.780	0.478	0.191	0.096
		trust	1.000	0.987	0.960	0.942	0.570	0.285
ADJ.	PMI	anger	1.000	0.973	0.953	0.888	0.658	0.357
		anticipation	1.000	1.000	0.907	0.702	0.295	0.147
		disgust	1.000	1.000	1.000	0.978	0.787	0.497
		fear	1.000	0.980	0.970	0.942	0.711	0.383
		joy	1.000	0.960	0.937	0.898	0.573	0.289
		sadness	1.000	1.000	0.953	0.900	0.639	0.331
		surprise	1.000	0.933	0.803	0.648	0.333	0.166
		trust	1.000	0.993	0.960	0.925	0.717	0.397
ADJ.	relation	anger	1.000	0.993	0.963	0.892	0.507	0.253
		anticipation	1.000	1.000	0.943	0.632	0.253	0.126
		disgust	1.000	1.000	0.983	0.945	0.743	0.380
		fear	1.000	1.000	0.983	0.943	0.536	0.268
		joy	1.000	0.987	0.920	0.802	0.370	0.185
		sadness	1.000	1.000	0.983	0.917	0.539	0.270
		surprise	1.000	0.967	0.847	0.523	0.210	0.105
		trust	1.000	1.000	0.973	0.932	0.600	0.300
ADJ.	relation-combined	anger	1.000	0.973	0.953	0.885	0.655	0.357
		anticipation	1.000	1.000	0.910	0.727	0.311	0.156
		disgust	1.000	1.000	1.000	0.980	0.789	0.497
		fear	1.000	0.980	0.970	0.948	0.718	0.386
		joy	1.000	0.973	0.940	0.910	0.579	0.292
		sadness	1.000	1.000	0.950	0.900	0.636	0.330
		surprise	1.000	0.940	0.820	0.668	0.338	0.169
		trust	1.000	0.993	0.970	0.928	0.731	0.405
ADJ.	context	anger	1.000	0.987	0.960	0.870	0.495	0.248
		anticipation	1.000	0.987	0.907	0.630	0.257	0.128
		disgust	1.000	1.000	1.000	0.950	0.599	0.301
		fear	1.000	0.987	0.977	0.955	0.560	0.280
		joy	1.000	1.000	0.970	0.870	0.513	0.257
		sadness	1.000	0.993	0.953	0.883	0.582	0.298
		surprise	1.000	0.980	0.800	0.538	0.219	0.110
		trust	1.000	1.000	0.970	0.952	0.638	0.319

POS	Training	Emotion	Top 1	Top 5	Top 10	Top 20	Top 50	Top 100
ADJ.	context	anger	1.000	0.980	0.957	0.905	0.679	0.376
		anticipation	1.000	0.987	0.920	0.757	0.335	0.168
		disgust	1.000	1.000	1.000	0.967	0.788	0.503
		fear	1.000	0.987	0.977	0.950	0.741	0.404
		joy	1.000	0.980	0.943	0.895	0.583	0.296
		sadness	1.000	0.987	0.957	0.912	0.685	0.352
		surprise	1.000	0.940	0.800	0.650	0.342	0.171
		trust	1.000	1.000	0.977	0.932	0.723	0.411

Table B.2: Emotional Similarity

Appendix C

Annotator Instructions

In this appendix I show the instructions given to each annotator in Section C.1 and presents the results from each individual annotator in Section C.2.

C.1 Instructions

If anything is unclear these instructions , please contact me before starting.

In this evaluation exercise you are presented with a word added to the 1911 edition of Rogets Thesaurus along with the context in which this new word appears. You are requested to indicate whether the word belongs in that context. Two kinds of evaluation will take place, to identify if a word is

- in the right Roget's Paragraph;
- in the right Roget's Head.

Here is an example of a Head: (See Figure C.1)

A Head contains three parts-of-speech (POS). A POSs has one or more Paragraphs. A Paragraph contains one or more Semicolon Groups (SGs), which are made up of words/phrases. SGs tend to contain the closest synonyms, while Paragraphs contain more loose groupings of related words. SGs are separated by a semicolon, while a paragraph ends with a period. There are six Paragraphs in the example above, each with multiple semicolon groups.

Head: 586 Language

N.

language; phraseology; speech; tongue, lingo, vernacular; mother tongue, vulgar tongue, native tongue; household words; King's English, Queen's English; dialect.

confusion of tongues, Babel, pasigraphie; pantomime; onomatopoeia; betacism, mimmation, myatism, nunnation; pasigraphy.

lexicology, philology, glossology, glottology; linguistics, chrestomathy; paleology, paleography; comparative grammar.

literature, letters, polite literature, belles lettres, muses, humanities, literae humaniores, republic of letters, dead languages, classics; genius of language; scholarship.

VB.

express by words.

ADJ.

lingual, linguistic; dialectic; vernacular, current; bilingual; diglot, hexaglot, polyglot; literary.

C.1.1 New Word in an Existing Paragraph

In the first exercise you are given a Paragraph from Rogets Thesaurus where a new word has been added. The Head name/number and the POS are also provided. You are asked to identify how close this new word is to being located in the right spot in Roget's. Specifically you will indicate if the word is in the correct SG, Paragraph, Head, or is in the wrong Head. You will assign a score as follows:

- 4 – word is in the correct SG
- 3 – word is in the correct Paragraph
- 2 – word is in the correct Head
- 1 – none of the above (wrong Head)

A word can be said to be in the correct SG if either it is very close in meaning to the other words in that SG, or if it is alone in a SG and no other SG would be an appropriate fit. A word is in the correct Paragraph if either it belongs in a different SG or in a new SG within that Paragraph. A word is in the correct Head if it has some conceivable relation to the words in the Paragraph and to the Head name, but at the same time clearly does not belong in the shown Paragraph. A word is in the wrong Head if it has either an opposite meaning to the concept represented in the Head or if it is completely irrelevant. Examples of each are shown below, with additional explanation in brackets.

In the examples below the new word is coloured red and underlined. Words in the same SG are bold. Each SG appears on its own line, except when a line wraps. A semicolon denotes the end of a SG.

Score	*Roget's* Paragraph

4
(word fits in this SG)
	Head 25: Agreement, noun
	
	fitness, aptness;
	relevancy;
	pertinence, pertinencey;
	sortance;
	case in point;
	aptitude, coaptation, propriety, applicability, admissibility, commensurability, *compatibility*;
	cognation.

Score	*Roget’s* Paragraph

4
(word fits in this Paragraph but not in a different SG)
	Head 25: Agreement, noun
	
	fitness, aptness;
	relevancy;
	pertinence, pertinencey;
	sortance;
	case in point;
	aptitude, coaptation, propriety, applicability,
	admissibility, commensurability, compatibility;
	cognation**.**

3
(appropriate for this Paragraph but not the SG)
	Head 25: Agreement, noun
	
	fitness, aptness;
	pertinence, pertinencey;
	sortance;
	case in point;
	aptitude, coaptation, propriety, applicability,
	admissibility, commensurability, *relevancy*;
	cognation.

2
(related to Agreement but not this Paragraph)
	Head 25: Agreement, noun
	
	fitness, aptness;
	relevancy;
	pertinence, pertinencey;
	sortance;
	case in point;
	aptitude, coaptation, propriety, applicability,
	admissibility, commensurability, *cooperation*;
	cognation.

1
(holds an opposite meaning)
	Head 25: Agreement, noun
	
	fitness, aptness;
	relevancy;
	pertinence, pertinencey;
	sortance;
	case in point;
	aptitude, coaptation, propriety, applicability,
	admissibility, commensurability, *disagreement*;
	cognation.

Score	*Roget's* Paragraph

1
(irrelevant)
	Head 25: Agreement, noun
	
	fitness, aptness;
	relevancy;
	pertinence, pertinencey;
	sortance;
	case in point;
	aptitude, coaptation, propriety, applicability, admissibility, commensurability, *snowflake*;
	cognation.

Table C.1: Sample questions and scores for evaluating a new word added to a previously existing Paragraph.

Your task is to assign scores (in the column ”Score”) to items like the following three. You do not need to explain the reason for your scores as in the examples above, simply enter a number from 1 to 4. If a decision is too hard or a word's meaning too obscure, please feel free to leave the score box blank.

C.1.2 New Word in a New Paragraph

In this exercise you are given a new word which is alone in a Paragraph. You are asked to decide whether it is in the correct Head. Since there is no context for this word, you are given a list of the first words from each Paragraph in that Head. Once again the Head name, number and POS are provided. You will assign scores as follows:

- 2 – the word is in the correct Head,
- 1 – the words is not the correct Head.

Examples with explanations are as follows: (See Table C.2)

Score	*Roget's* Paragraph
2 (closely related)	*Head 25: Agreement, noun* agreement.. / conformity.. / fitness.. / adaption.. / ***consent;***;
1 (opposite meaning)	*Head 25: Agreement, noun* agreement.. / conformity.. / fitness.. / adaption.. / ***disagreement;***;

Score	*Roget's* Paragraph
1 (irrelevant)	*Head 25: Agreement, noun* agreement.. / conformity.. / fitness.. / adaption.. / ***drunkenness;***;

Table C.2: Sample questions and scores for evaluating a new word added to a new Paragraph.

Your task is now to assign scores to items like the following four. Once again you do not need to explain the reason for your scores as in the examples above, simply enter a number from 1 to 4. Additionally if a question is too hard or a word's meaning too obscure please feel free to leave it blank.

C.2 Individual Annotator Results

This section contains the results for each annotator, denoted “Annotator X” where X is in 0..4. Results for words added to existing Paragraphs for all 5 annotators are found in Tables C.3, C.5, C.7, C.9 & C.11, while results for adding words to new Paragraphs are found in Tables C.4, C.6, C.8, C.10 & C.12. Combined results for Annotators 1..4 – excluding my own annotations – is shown in Tables C.13 and C.14.

		Right	Right	Right	Wrong	
Task	POS	SG	Para	Head	Head	N/A
Positive	noun	27 (*0.692*)	2 (*0.051*)	0 (*0.000*)	7 (*0.179*)	3 (*0.077*)
	verb	15 (*0.714*)	1 (*0.048*)	0 (*0.000*)	4 (*0.190*)	1 (*0.048*)
	adjective	9 (*0.500*)	4 (*0.222*)	1 (*0.056*)	3 (*0.167*)	1 (*0.056*)
Negative	noun	0 (*0.000*)	0 (*0.000*)	1 (*0.026*)	36 (*0.923*)	2 (*0.051*)
	verb	1 (*0.048*)	0 (*0.000*)	1 (*0.048*)	19 (*0.905*)	0 (*0.000*)
	adjective	1 (*0.056*)	1 (*0.056*)	1 (*0.056*)	15 (*0.833*)	0 (*0.000*)
1911X1	noun	34 (*0.667*)	6 (*0.118*)	4 (*0.078*)	7 (*0.137*)	0 (*0.000*)
	verb	21 (*0.583*)	4 (*0.111*)	2 (*0.056*)	9 (*0.250*)	0 (*0.000*)
	adjective	32 (*0.744*)	3 (*0.070*)	2 (*0.047*)	6 (*0.140*)	0 (*0.000*)
1911X5	noun	42 (*0.667*)	5 (*0.079*)	8 (*0.127*)	8 (*0.127*)	0 (*0.000*)
	verb	28 (*0.538*)	2 (*0.038*)	9 (*0.173*)	13 (*0.250*)	0 (*0.000*)
	adjective	36 (*0.621*)	3 (*0.052*)	5 (*0.086*)	14 (*0.241*)	0 (*0.000*)

Table C.3: Results for Annotator 0 on the Manual Evaluation for words added to existing Paragraphs.

Task	POS	Right Head	Wrong Head	N/A
Positive	noun	34 (*0.872*)	5 (*0.128*)	0 (*0.000*)
	verb	20 (*0.952*)	1 (*0.048*)	0 (*0.000*)
	adjective	17 (*0.944*)	1 (*0.056*)	0 (*0.000*)
Negative	noun	4 (*0.103*)	30 (*0.769*)	5 (*0.128*)
	verb	5 (*0.238*)	16 (*0.762*)	0 (*0.000*)
	adjective	8 (*0.444*)	10 (*0.556*)	0 (*0.000*)
1911X1	noun	37 (*0.841*)	7 (*0.159*)	0 (*0.000*)
	verb	12 (*1.000*)	0 (*0.000*)	0 (*0.000*)
	adjective	9 (*0.818*)	2 (*0.182*)	0 (*0.000*)
1911X5	noun	50 (*0.806*)	12 (*0.194*)	0 (*0.000*)
	verb	17 (*0.680*)	8 (*0.320*)	0 (*0.000*)
	adjective	17 (*0.850*)	3 (*0.150*)	0 (*0.000*)

Table C.4: Results for Annotator 0 on the Manual Evaluation for words added to new Paragraphs.

		Right	Right	Right	Wrong	
Task	POS	SG	Para	Head	Head	N/A
Positive	noun	9 (*0.231*)	6 (*0.154*)	11 (*0.282*)	3 (*0.077*)	10 (*0.256*)
	verb	9 (*0.429*)	7 (*0.333*)	1 (*0.048*)	2 (*0.095*)	2 (*0.095*)
	adjective	6 (*0.333*)	5 (*0.278*)	1 (*0.056*)	1 (*0.056*)	5 (*0.278*)
Negative	noun	0 (*0.000*)	0 (*0.000*)	3 (*0.077*)	27 (*0.692*)	9 (*0.231*)
	verb	0 (*0.000*)	1 (*0.048*)	5 (*0.238*)	13 (*0.619*)	2 (*0.095*)
	adjective	0 (*0.000*)	0 (*0.000*)	1 (*0.056*)	15 (*0.833*)	2 (*0.111*)
1911X1	noun	15 (*0.294*)	25 (*0.49*)	5 (*0.098*)	3 (*0.059*)	3 (*0.059*)
	verb	10 (*0.278*)	14 (*0.389*)	8 (*0.222*)	1 (*0.028*)	3 (*0.083*)
	adjective	14 (*0.326*)	18 (*0.419*)	7 (*0.163*)	2 (*0.047*)	2 (*0.047*)
1911X5	noun	15 (*0.238*)	29 (*0.460*)	12 (*0.190*)	3 (*0.048*)	4 (*0.063*)
	verb	7 (*0.135*)	17 (*0.327*)	18 (*0.346*)	7 (*0.135*)	3 (*0.058*)
	adjective	13 (*0.224*)	24 (*0.414*)	11 (*0.190*)	8 (*0.138*)	2 (*0.034*)

Table C.5: Results for Annotator 1 on the Manual Evaluation for words added to existing Paragraphs.

Task	POS	Right Head	Wrong Head	N/A
Positive	noun	37 (*0.949*)	0 (*0.000*)	2 (*0.051*)
	verb	16 (*0.762*)	5 (*0.238*)	0 (*0.000*)
	adjective	18 (*1.000*)	0 (*0.000*)	0 (*0.000*)
Negative	noun	1 (*0.026*)	33 (*0.846*)	5 (*0.128*)
	verb	4 (*0.190*)	15 (*0.714*)	2 (*0.095*)
	adjective	0 (*0.000*)	17 (*0.944*)	1 (*0.056*)
1911X1	noun	35 (*0.795*)	7 (*0.159*)	2 (*0.045*)
	verb	8 (*0.667*)	4 (*0.333*)	0 (*0.000*)
	adjective	9 (*0.818*)	2 (*0.182*)	0 (*0.000*)
1911X5	noun	49 (*0.790*)	8 (*0.129*)	5 (*0.081*)
	verb	17 (*0.680*)	8 (*0.320*)	0 (*0.000*)
	adjective	14 (*0.700*)	5 (*0.250*)	1 (*0.050*)

Table C.6: Results for Annotator 1 on the Manual Evaluation for words added to new Paragraphs.

		Right	Right	Right	Wrong	
Task	POS	SG	Para	Head	Head	N/A
Positive	noun	34 (*0.872*)	0 (*0.000*)	3 (*0.077*)	2 (*0.051*)	0 (*0.000*)
	verb	16 (*0.762*)	1 (*0.048*)	1 (*0.048*)	1 (*0.048*)	2 (*0.095*)
	adjective	11 (*0.611*)	3 (*0.167*)	2 (*0.111*)	2 (*0.111*)	0 (*0.000*)
Negative	noun	5 (*0.128*)	1 (*0.026*)	8 (*0.205*)	19 (*0.487*)	6 (*0.154*)
	verb	8 (*0.381*)	1 (*0.048*)	3 (*0.143*)	9 (*0.429*)	0 (*0.000*)
	adjective	1 (*0.056*)	2 (*0.111*)	2 (*0.111*)	11 (*0.611*)	2 (*0.111*)
1911X1	noun	47 (*0.922*)	3 (*0.059*)	0 (*0.000*)	1 (*0.02*)	0 (*0.000*)
	verb	23 (*0.639*)	4 (*0.111*)	6 (*0.167*)	3 (*0.083*)	0 (*0.000*)
	adjective	35 (*0.814*)	3 (*0.070*)	3 (*0.070*)	2 (*0.047*)	0 (*0.000*)
1911X5	noun	48 (*0.762*)	6 (*0.095*)	6 (*0.095*)	2 (*0.032*)	1 (*0.016*)
	verb	29 (*0.558*)	5 (*0.096*)	7 (*0.135*)	11 (*0.212*)	0 (*0.000*)
	adjective	40 (*0.690*)	5 (*0.086*)	6 (*0.103*)	7 (*0.121*)	0 (*0.000*)

Table C.7: Results for Annotator 2 on the Manual Evaluation for words added to existing Paragraphs.

Task	POS	Right Head	Wrong Head	N/A
Positive	noun	27 (*0.692*)	10 (*0.256*)	2 (*0.051*)
	verb	14 (*0.667*)	6 (*0.286*)	1 (*0.048*)
	adjective	13 (*0.722*)	5 (*0.278*)	0 (*0.000*)
Negative	noun	4 (*0.103*)	24 (*0.615*)	11 (*0.282*)
	verb	4 (*0.190*)	14 (*0.667*)	3 (*0.143*)
	adjective	2 (*0.111*)	14 (*0.778*)	2 (*0.111*)
1911X1	noun	38 (*0.864*)	4 (*0.091*)	2 (*0.045*)
	verb	8 (*0.667*)	4 (*0.333*)	0 (*0.000*)
	adjective	10 (*0.909*)	1 (*0.091*)	0 (*0.000*)
1911X5	noun	30 (*0.508*)	28 (*0.475*)	1 (*0.017*)
	verb	9 (*0.429*)	12 (*0.571*)	0 (*0.000*)
	adjective	6 (*0.316*)	13 (*0.684*)	0 (*0.000*)

Table C.8: Results for Annotator 2 on the Manual Evaluation for words added to new Paragraphs.

		Right	Right	Right	Wrong	
Task	POS	SG	Para	Head	Head	N/A
Positive	noun	26 *(0.667)*	6 *(0.154)*	1 *(0.026)*	6 *(0.154)*	0 *(0.000)*
	verb	11 *(0.524)*	3 *(0.143)*	2 *(0.095)*	4 *(0.190)*	1 *(0.048)*
	adjective	16 *(0.889)*	1 *(0.056)*	0 *(0.000)*	1 *(0.056)*	0 *(0.000)*
Negative	noun	1 *(0.026)*	0 *(0.000)*	2 *(0.051)*	32 *(0.821)*	4 *(0.103)*
	verb	0 *(0.000)*	0 *(0.000)*	2 *(0.095)*	19 *(0.905)*	0 *(0.000)*
	adjective	0 *(0.000)*	0 *(0.000)*	0 *(0.000)*	18 *(1.000)*	0 *(0.000)*
1911X1	noun	34 *(0.667)*	8 *(0.157)*	1 *(0.020)*	8 *(0.157)*	0 *(0.000)*
	verb	17 *(0.472)*	9 *(0.250)*	1 *(0.028)*	9 *(0.250)*	0 *(0.000)*
	adjective	25 *(0.581)*	13 *(0.302)*	2 *(0.047)*	3 *(0.07)*	0 *(0.000)*
1911X5	noun	37 *(0.597)*	11 *(0.177)*	2 *(0.032)*	12 *(0.194)*	0 *(0.000)*
	verb	18 *(0.346)*	15 *(0.288)*	3 *(0.058)*	16 *(0.308)*	0 *(0.000)*
	adjective	30 *(0.517)*	8 *(0.138)*	4 *(0.069)*	16 *(0.276)*	0 *(0.000)*

Table C.9: Results for Annotator 3 on the Manual Evaluation for words added to existing Paragraphs.

Task	POS	Right Head	Wrong Head	N/A
Positive	noun	28 *(0.718)*	11 *(0.282)*	0 *(0.000)*
	verb	17 *(0.810)*	4 *(0.190)*	0 *(0.000)*
	adjective	12 *(0.667)*	5 *(0.278)*	1 *(0.056)*
Negative	noun	5 *(0.128)*	33 *(0.846)*	1 *(0.026)*
	verb	2 *(0.095)*	19 *(0.905)*	0 *(0.000)*
	adjective	2 *(0.111)*	16 *(0.889)*	0 *(0.000)*
1911X1	noun	36 *(0.818)*	8 *(0.182)*	0 *(0.000)*
	verb	11 *(0.917)*	1 *(0.083)*	0 *(0.000)*
	adjective	10 *(0.909)*	1 *(0.091)*	0 *(0.000)*
1911X5	noun	47 *(0.758)*	15 *(0.242)*	0 *(0.000)*
	verb	14 *(0.560)*	10 *(0.400)*	1 *(0.040)*
	adjective	14 *(0.700)*	6 *(0.300)*	0 *(0.000)*

Table C.10: Results for Annotator 3 on the Manual Evaluation for words added to new Paragraphs.

Task	POS	Right SG	Right Para	Right Head	Wrong Head	N/A
Positive	noun	21 *(0.538)*	6 *(0.154)*	7 *(0.179)*	3 *(0.077)*	2 *(0.051)*
	verb	8 *(0.381)*	2 *(0.095)*	6 *(0.286)*	5 *(0.238)*	0 *(0.000)*
	adjective	13 *(0.722)*	3 *(0.167)*	2 *(0.111)*	0 *(0.000)*	0 *(0.000)*
Negative	noun	0 *(0.000)*	1 *(0.026)*	6 *(0.154)*	30 *(0.769)*	2 *(0.051)*
	verb	0 *(0.000)*	0 *(0.000)*	7 *(0.333)*	13 *(0.619)*	1 *(0.048)*
	adjective	1 *(0.056)*	1 *(0.056)*	4 *(0.222)*	12 *(0.667)*	0 *(0.000)*
1911X1	noun	29 *(0.569)*	10 *(0.196)*	12 *(0.235)*	0 *(0.000)*	0 *(0.000)*
	verb	21 *(0.583)*	6 *(0.167)*	7 *(0.194)*	2 *(0.056)*	0 *(0.000)*
	adjective	29 *(0.674)*	7 *(0.163)*	3 *(0.070)*	4 *(0.093)*	0 *(0.000)*
1911X5	noun	39 *(0.619)*	8 *(0.127)*	16 *(0.254)*	0 *(0.000)*	0 *(0.000)*
	verb	25 *(0.481)*	6 *(0.115)*	16 *(0.308)*	5 *(0.096)*	0 *(0.000)*
	adjective	28 *(0.483)*	12 *(0.207)*	6 *(0.103)*	11 *(0.19)*	1 *(0.017)*

Table C.11: Results for Annotator 4 on the Manual Evaluation for words added to existing Paragraphs.

Task	POS	Right Head	Wrong Head	N/A
Positive	noun	32 *(0.821)*	7 *(0.179)*	0 *(0.000)*
	verb	20 *(0.952)*	1 *(0.048)*	0 *(0.000)*
	adjective	15 *(0.833)*	3 *(0.167)*	0 *(0.000)*
Negative	noun	4 *(0.103)*	31 *(0.795)*	4 *(0.103)*
	verb	2 *(0.095)*	19 *(0.905)*	0 *(0.000)*
	adjective	1 *(0.056)*	17 *(0.944)*	0 *(0.000)*
1911X1	noun	43 *(0.977)*	1 *(0.023)*	0 *(0.000)*
	verb	11 *(0.917)*	1 *(0.083)*	0 *(0.000)*
	adjective	10 *(0.909)*	1 *(0.091)*	0 *(0.000)*
1911X5	noun	31 *(0.500)*	31 *(0.500)*	0 *(0.000)*
	verb	7 *(0.292)*	17 *(0.708)*	0 *(0.000)*
	adjective	10 *(0.500)*	10 *(0.500)*	0 *(0.000)*

Table C.12: Results for Annotator 4 on the Manual Evaluation for words added to new Paragraphs.

Task	POS	Right SG	Right Para	Right Head	Wrong Head	N/A
Positive	noun	90 (*0.581*)	18 (*0.116*)	22 (*0.142*)	14 (*0.090*)	11 (*0.071*)
	verb	44 (*0.524*)	13 (*0.155*)	10 (*0.119*)	12 (*0.143*)	5 (*0.06*)
	adjective	46 (*0.639*)	12 (*0.167*)	5 (*0.069*)	4 (*0.056*)	5 (*0.069*)
Negative	noun	6 (*0.038*)	2 (*0.013*)	19 (*0.122*)	108 (*0.692*)	21 (*0.135*)
	verb	8 (*0.095*)	2 (*0.024*)	17 (*0.202*)	54 (*0.643*)	3 (*0.036*)
	adjective	2 (*0.028*)	3 (*0.042*)	7 (*0.097*)	56 (*0.778*)	4 (*0.056*)
1911X1	noun	125 (*0.613*)	46 (*0.225*)	18 (*0.088*)	12 (*0.059*)	3 (*0.015*)
	verb	71 (*0.493*)	33 (*0.229*)	22 (*0.153*)	15 (*0.104*)	3 (*0.021*)
	adjective	103 (*0.599*)	41 (*0.238*)	15 (*0.087*)	11 (*0.064*)	2 (*0.012*)
1911X5	noun	139 (*0.554*)	54 (*0.215*)	36 (*0.143*)	17 (*0.068*)	5 (*0.02*)
	verb	79 (*0.380*)	43 (*0.207*)	44 (*0.212*)	39 (*0.188*)	3 (*0.014*)
	adjective	111 (*0.478*)	49 (*0.211*)	27 (*0.116*)	42 (*0.181*)	3 (*0.013*)

Table C.13: Results of the Manual Evaluation for words added to existing Paragraphs where my annotations are excluded.

Task	POS	Right Head	Wrong Head	N/A
Positive	noun	124 (*0.795*)	28 (*0.179*)	4 (*0.026*)
	verb	67 (*0.798*)	16 (*0.190*)	1 (*0.012*)
	adjective	58 (*0.806*)	13 (*0.181*)	1 (*0.014*)
Negative	noun	14 (*0.090*)	121 (*0.776*)	21 (*0.135*)
	verb	12 (*0.143*)	67 (*0.798*)	5 (*0.06*)
	adjective	5 (*0.069*)	64 (*0.889*)	3 (*0.042*)
1911X1	noun	152 (*0.864*)	20 (*0.114*)	4 (*0.023*)
	verb	38 (*0.792*)	10 (*0.208*)	0 (*0.000*)
	adjective	39 (*0.886*)	5 (*0.114*)	0 (*0.000*)
1911X5	noun	157 (*0.641*)	82 (*0.335*)	6 (*0.024*)
	verb	47 (*0.495*)	47 (*0.495*)	1 (*0.011*)
	adjective	44 (*0.557*)	34 (*0.43*)	1 (*0.013*)

Table C.14: Results of the Manual Evaluation for words added to new Paragraphs where my annotations are excluded.