Nominal Taxonomies and Word Sense Disambiguation

Nuno MF Dionisio

A thesis submitted
for the
Degree of Doctor of Philosophy

School of Computing Science,
University of East Anglia, Norwich

May 31, 2004

©This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognize that its copyright rests with the author and that no quotation from the thesis, nor any information derived therefrom, may be published without the author's prior written consent.

Abstract

Word Sense Disambiguation (WSD) is a significant problem in Natural Language Processing (NLP). Current NLP research employs WSD to aid tasks such as Machine Translation, Information Retrieval, Content Analysis, Parsing and Speech Processing.

Semantic Similarity using lexical taxonomies is investigated, producing specialised WSD algorithms for the disambiguation of related noun groups. By creating semantic similarity measures based on notions of the "shape" of WordNet's lexical taxonomy (SBSMs) containing only layman terms, results are produced that significantly outperform existing state-of-the-art similarity measures in two tasks; firstly in matching human judgements, and secondly for disambiguating related noun-groupings. In the human judgement experiment, results are evaluated using Pearson and Spearman correlation coefficients. The best SBSM almost reaches the equivalent human performance producing coefficients of 0.90 and 0.86 respectively.

A WSD system is presented for disambiguating related nouns groups, producing 88% precision and 90% recall for labelling a subset of Wordsmyth with equivalent WordNet senses. These results improve those produced using alternative similarity measures, and when compared to the Wordsmyth experimental links to WordNet.

The SBSMs are used as part of a WSD system for disambiguating open-texts. The proposed WSD system makes use of partial-taggers to reduce senses at different stages of WSD. A final statistical component is investigated, using a new linguistically based definition of context. The SBSMs are used to match words according to similarity. Experiments with 11 highly polysemous words give promising results at 37.7% precision and recall for all words with an average polysemy of 22.1 senses, and 56.4% precision and recall for nouns with an average polysemy of 6 senses. Using a smaller test set of ambiguous contexts containing only test words produced 65.6% precision and recall for all words. This WSD is also used to reduce the costs of manual tagging of words, showing that a potential 60% reduction in cost is possible.

"There is no need to do more than mention the obvious fact that a multiplicity of languages impedes cultural interchange between the peoples of the earth, and is a serious deterrent to international understanding." (Weaver, 1949)

This thesis is dedicated to
my parents,
Alfredo & Felicidade Dionisio,
and to the rest of my family

Acknowledgments

I would like to thank Dr. Ian Marshall for his supervision throughout my PhD. He has shown great patience at the hardest times. It has been a privilege to be his first PhD. student.

I am also very grateful and indebted to the States of Jersey Education Committee for giving me opportunities in life, firstly go to University and secondly to allow me to study for a PhD. Without their funding, this opportunity would never presented itself to me, and I appreciate this even more in light of how little support many others around the world are given.

I would like to thank Dr. Robert Parks for kindly providing the Wordsmyth data used for evaluation purposes in section 4.5.2, and Dr. John Hutchins for his input and help with some of the references.

Thanks also to Dr. Robert Foxall, Dr. Éva Sáfár, Dr. Gavin Cawley, Dr. Richard Harvey and Prof. Ronan Sleep, who have found the time and courage to read parts of this thesis. Their comments have been greatly valued, and have helped to further improve the quality of this thesis.

I would like to thank my close friend (and soon to be Doctor) Effie Kostopoulou. She has given me the strength to work through some difficult periods over the past few years and has always been supportive at times of need.

I must also mention all my friends and colleagues that have offered support in the way of company during coffee breaks, for helping me relax by offering havens away from my studies, and for offering suggestions that have influenced the progress of my work. I choose not to mention these people by name in the risk of missing anyone out, but I am sure they all know who they are.

Last but definitely not the least, special thanks go to my parents and family who have been so supportive since the beginning of my studies. I would not have been able to achieve this work without them.

Contents

List of Figures

List of Tables

Chapter 1

Overview

This chapter summarises the work presented in this thesis. Firstly, the motivations behind this work are presented, followed by a simple example of the kind of ambiguity of interest in language studied throughout the thesis. Lastly, a description of the overall organisation of the thesis is given.

1.1 Motivation

The most popular approaches in current machine translation research are based on the exploitation of statistical information from bilingual corpora (Brown et al., 1990; Hutchins, 1995; Berger et al., 1996; Turcato et al., 1999). By calculating statistical information about the translation of these texts, translation systems are able to determine the most likely translation of new sentences. A significant problem with such an approach is that it is currently only possible to build systems for a small number of languages due to the lack of available resources. The situation is even worse when a language does not have any resources at all or even an accepted written form, such as the various sign languages in the world (Veale et al., 1998). In order to be able to translate between such languages, it is necessary to take a completely different approach.

The aim of this thesis is to investigate two natural language sub-tasks which could be used as part of a larger linguistically based translation system, such as is shown in figure 1.1. In such an approach, a number of techniques are applied to the source text in order to remove all language specific facets, thus producing an interlingua representation of the original source text. This intermediate representation is used to create a

representation of the concepts expressed in the text in the target language.

Figure 1.1: Framework for a General Translation System

The focus of concern for this thesis is the investigation of tools for measuring semantic similarity and performing word sense disambiguation. In combination with further natural language processing (NLP) tools, such as a part-of-speech recognition tool, a syntactic parser and discourse reference structure generator, a system can be built to generate an interlingua representation from which a translation of the original text can be produced. By considering the semantic similarity of words, techniques can be developed to disambiguate semantically related word-groupings providing tools for linking different lexical resources and to allow words to be matched semantically. Matching words using semantic similarity allows the potential for statistical Word Sense Disambiguation (WSD) systems to gather adequate information from the existing limited resources, by making use of information from similar words to increase the amount of information available to disambiguate individual words. This also allows statistical WSD systems to disambiguate words or word senses for which no information was available in the resources available. WSD in turn allows for the disambiguation of concepts, and is especially important for translation as there are rarely one-to-one mappings from words to word senses between different languages. Therefore knowledge of the conceptual meaning of a word permits the correct lexical term to be selected for the translated text.

1.2 Example of Lexical Ambiguity

There are many different forms of ambiguity found in natural languages that pose problems to NLP tasks. Of these ambiguities, this thesis is particularly focused on ambiguities concerned with the semantic definitions of words, particularly of nouns. If one is to consider the definitions of a word within varying contexts, it is easy to find instances of uses of different word senses:

John deposited his money in the bank.

He was near the river bank.
I went to the bank.

In the first example, "bank" refers to a financial institute, whilst in the second example it refers to a slope. The last example remains ambiguous, and requires either further information for accurate disambiguation or, in the absence of further information, assumption of the most prominent sense for the word. The problem of disambiguation is typically more extreme as even in the case of "bank" one can find many further sense distinctions, for instance the lexical resource WordNet 1.6 (Miller et al., 1990; Fellbaum, 1998) cites 17 sense distinctions.

This thesis presents a study of lexical ambiguity, firstly concerned with the disambiguation of semantically related words, and then moving to the more general problem of disambiguating the senses of words in open texts.

1.3 Organisation of Thesis

This thesis is organised into six further chapters, grouped into four general sections as follows:

1. Description of important resources and tools used for the research.

   Chapter 2 introduces a number of tools that were used as the basis of the research and that are referred to throughout the thesis.

2. Research into measuring semantic similarity.

   Chapter 3 introduces some terminology and resources, and describes a number of the better known existing similarity measures. Finally, a short discussion is given about measuring similarity between verbs, and why the current techniques applied to nouns may not be as suitable for verbs. Chapter 4 discusses the difficulties posed by use of WordNet for similarity measures relying on the information contained within its lexical taxonomy. Given these difficulties, a number of axioms are defined to support discussion of the use of WordNet's lexical taxonomy for similarity between noun senses. From these axioms, a set of hypotheses are formulated describing how WordNet's lexical taxonomy can be used to measure similarity leading to the definition of a number of new similarity measures.

These measures are then evaluated against human judgements and semantically related groups of nouns. Results show that some of the newly created measures significantly outperform existing state-of-the-art similarity measures.

3. Research into a new approach for Word Sense Disambiguation.

   Chapter 5 introduces the field of WSD. The chapter starts by showing the interest that different sub-fields of NLP have for WSD in order to improve the results within those fields. This is followed by a brief summary of the history of important developments for WSD from early work in NLP. A selection of recent influential techniques is given. The chapter concludes by describing the gold-standard evaluation techniques for WSD. Chapter 6 discusses a new approach for WSD using a number of partial-taggers. The remainder of the chapter concentrates on the development of statistical classifiers for WSD based on the maximum entropy framework. This statistical approach uses a new definition of local context for words based-on the syntactic relationships between words in a sentence. A new set of maximum entropy features are also defined using this definition of local context, and utilising the most successful similarity measure from chapter 4 to match words and word senses given semantic similarity, instead of matching words using their word-form.

4. Conclusions.

   Finally, chapter 7 reviews the work presented in this thesis and the results for the systems evaluated. This is followed by a description of possible future work. The chapter concludes by listing the contribution the work of this thesis has made to the NLP field.

Chapter 2

Tools and Resources

A variety of tools and resources are introduced which are used in a number of different examples of systems described throughout this thesis, and that have been used to develop the similarity measures and Word Sense Disambiguation (WSD) systems described in chapters 4 and 6 respectively. Each tool provides specific functions for the systems produced for the work presented in this thesis:

- Machine Readable Dictionary (MRD) – WordNet
- Sense Annotated Corpus – Semcor
- Parser – Carnegie Mellon University's (CMU) Link Grammar Parser
- Custom Natural Language Processing (NLP) application

2.1 WordNet

WordNet (Miller et al., 1990) is a psycholinguistic lexicon widely used in a number of contemporary NLP systems. A psycholinguistic lexicon is a lexicon that models information in accordance with principles believed to govern the human lexicon memory. WordNet's growing influence in the field of WSD has been apparent over the last decade, and is now used for WSD almost to the exclusion of all other dictionaries. Its organisation of semantic information gives researchers one of the most compact and rich sources of information for such tasks as measuring word similarity, and in some cases has been used for full WSD of texts without the aid of further resources.

Fellbaum (1998) presents a full description of WordNet and a collection of papers describing some of the research performed using WordNet.

The major difference between WordNet and other lexical resources is the way in which lexical information is organised. As with most dictionaries, information is grouped into different grammatical categories (nouns, verbs, adjectives and adverbs), however rather than organising information primarily at the word-form level, WordNet organises information at the conceptual level. Each concept, referred to as a synset (set of synonyms) (Spark Jones, 1978), is then related to other concepts via a number of psycholinguistic relationships. Table 2.1 summarises the information available in WordNet 1.6. Note that in table 2.1, the figures for "Unique Strings" include both

Part-Of-Speech (POS)	Unique Strings	Synsets	Average Polysemy (Excluding Monosemos Words)
Noun	94474	66025	2.73
Verb	10319	12127	3.57
Adjectives	20170	17915	2.80
Adverbs	4546	3575	2.50
Total	121962	99642	

Table 2.1: WordNet v1.6 Summary

single words and word collocations contained in WordNet. Therefore, an entry like "breach of trust with fraudulent intent" counts as a unique string.

Although a number of relationships are shared across different grammatical categories, the relationships available are different between the grammatical categories, supporting the importance humans show in distinguishing between different relationships for different grammatical categories. The relationships available per category are as follows:

- General (Shared by all POS)
  - Synonymy
  - Antonymy
  - Familiarity

- Nouns
  - Hypernymy
  - Hyponymy
  - Co-ordinate terms
  - Meronymy
  - Holonymy
- Verbs
  - Hypernymy
  - Troponymy
  - Entailment
  - Causality
  - Sentence Frames
- Adjectives
  - “Value of”
  - Pertainym
- Adverbs
  - Pertainym

The remainder of this section briefly discusses the meaning of each of the relationships above.

2.1.1 Synonymy

Synonymy is the most important relation for WordNet. The word synonym is derived from the Greek word “syn-onoma” meaning “similar name”, and refers to the relationship between words with the same meaning. In general, this means that synonyms of a word can substitute for each other without changing the meaning of a phrase.

2.1.2 Hypernymy (is-a)

Hypernymy, coming from the Greek words "hyper" and "onoma" meaning "super name" or "general name", is a directional relationship defined between two concepts. A concept, a, is the hypernym of another concept, b, if b "is an" a. That is to say, a is a more general form of b, for instance a "cat" is a "feline". In general, most concepts have at most one hypernym, although examples can be found where concepts have more than one hypernym, such as a "person" is both a "life form" and a "causal agent" according to WordNet 1.6.

Throughout this thesis, references to noun hypernym taxonomies, or structures, shall be made. Hypernym taxonomies represent tree-like structures that are linked upwards. Each arc in the structure represents an asymmetric relationship, where traversing up the tree reflects traversing up hypernym relations. Figure 2.1 shows the hypernym taxonomy for person sense 1 according to WordNet. Note also the convention that is used throughout this thesis of referring to a particular sense of a word by using the notation <word>#<sense>.

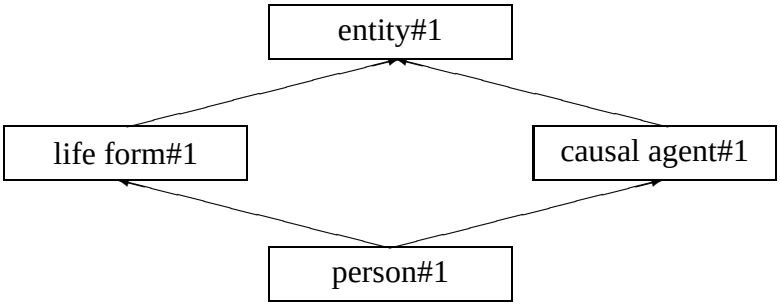


Figure 2.1: Multiple Hypernym Example

2.1.3 Hyponymy (kind-of)

Hyponymy is the inverse relation of hypernymy for nouns, therefore a concept, a, is the hyponym of another concept, b, if a "is a" b, in other words if a is a more specific form of b, or a "kind-of" b. In hypernym taxonomies, traversing down the arcs in the taxonomy is equivalent to traversing down hyponym relations.

2.1.4 Troponymy (way-of)

Troponymy is, similarly to hyponymy, the inverse relation of hypernymy, but in this case for verbs. The meaning of troponymy can also be seen in the meanings of the juxtaposed Greek words “tropos” and “onoma”, meaning a “way name” or “manner name”, and is used to describe a particular way or manner of doing something. For instance, to sprint is to run in a certain manner, therefore “sprint” is a troponym of “run”. Troponyms also reflect a type of entailment, as an action cannot be performed in a *certain manner* without also performing the original action.

2.1.5 Antonymy (opposite-of)

Antonyms, derived from “anti” and “onoma” in Greek meaning “opposite name”, are two words that can mean the opposite of each other, such as “true” and “false”, or “night” and “day”, however some care must be taken in its interpretation. It does not always follow that for two words, a and b, if a is an antonym of b, then not-a means b. Consider the antonyms “rich” and “poor”, or “black” and “white”. If someone is not rich, this does not automatically mean they are poor, and if something is not black, it does not mean that it is necessarily white, yet both pairs of words are still antonyms of each other.

2.1.6 Co-ordinate terms

Co-ordinate terms are concepts that share a common hyponym, for example the co-ordinate terms for “car#1” in WordNet, meaning a 4-wheeled motor vehicle, are other motor vehicles such as “motorcycle” or “truck”.

2.1.7 Meronymy (part-of)

The term meronym is derived from the Greek “meros” and “onoma” meaning “part name”. A meronym of a concept, a, is something that is “part of” a. For example, “engine” is a meronym of “car” because cars have engines, and “person” is a meronym of “people” because people consist of persons.

2.1.8 Holonymy

Holonymy is the inverse relation to meronymy. A concept a has a holonym b, if a is part of b. The term is derived from the Greek "holo" and "onoma" meaning "whole name".

2.1.9 Entailment

Entailment relations exist between verbs, such as "snore" entails "sleep" meaning that without sleep there would be no snoring. WordNet does not include troponyms in its entailment relations.

2.1.10 Causality

The causality relations describe, as it suggests, the result that an action, or verb, causes, such as "give" causes "have".

2.1.11 Sentence Frames

WordNet assigns at least one of 35 generic sentence frames to each verb sense, indicating the required verbal arguments and prepositional phrases, along with some basic semantic information about these arguments. In practice, the information available is very limited. FrameNet (Baker et al., 1998) is a project that had as an initial goal the task of producing more complex sentence frames for verbs than those provided by WordNet. Work is still currently being undertaken and sense mappings for entries are no longer one-to-one with WordNet.

2.1.12 Value of

The "value of" relation for adjectives links adjectives to nouns for which they can be a value. For instance the adjective "rich" can be a value referring to some kind of "financial condition", and "fast" is a value of "speed".

2.1.13 Pertainym

Pertainyms relate adjectives and adverbs to word senses they pertain or relate to. For instance the adjective “rural#2” pertains to the noun “country#5”.

2.1.14 Familiarity

The familiarity index of a word indicates its familiarity in day to day speech. Words that are more familiar are more likely to be found in examples of utterances or texts. In previous versions of WordNet, other lexicons were used to calculate the familiarity index for words. However, version 1.6 of WordNet simply considers the polysemy of a word as its familiarity index. This follows the largely accepted theory in linguistics that the more polysemous a word is, the more likely it is to be used (Zipf, 1945; Jastrezembski and Stanners, 1975; Jastrezembski, 1981).

2.2 Semcor

The Semcor corpus (Miller et al., 1994; Fellbaum, 1998) was selected to provide training and testing data for the purposes of the work on WSD presented in chapter 6. The corpus has been hand-annotated with word senses according to WordNet and contains POS tags produced by the Brill POS-tagger (Brill, 1992). The texts contained in the corpus are a subset of the Brown-corpus (Kucera and Francis, 1967; Francis, 1980; Francis and Kucera, 1982), and are split into 3 groups summarised in Table 2.2.

Group Name	Group Contents	What is tagged?
brown1	103 documents from the Brown Corpus	All Content Words
brown2	86 documents from the Brown Corpus	All Content Words
brownv	166 documents from the Brown Corpus	Only Verbs

Table 2.2: Summary of Semcor Texts

The Brown corpus consists of a number of texts from a range of topics and genres, therefore it can be assumed that the Semcor corpus also varies across a number of domains and genres. An earlier version of Semcor where words are tagged against WordNet 1.5 senses, as described in (Landes et al., 1998), only used the 103 document

collection from the Brown Corpus, together with a completely annotated version of Stephen Crane's novella "The Red Badge of Courage". The latter text is not contained in the latest version of Semcor tagged with WordNet 1.6 senses.

Annotation of the Semcor corpus was assisted with a tool called ConText. ConText allows users to view the senses of polysemous words and select an appropriate sense for each word. In order to ensure reliable tags are assigned to each word, the annotation process was performed over a number of iterations. In the first run, a highly trained human annotator assigned senses to each polysemous word of the corpus. The annotator could also leave notes when an adequate sense did not exist in WordNet. A second annotator verified the senses assigned, and made any necessary changes. The notes about missing senses were later examined by lexicographers who made changes, when necessary, to WordNet's information. ConText was used once more by an annotator to assign senses to the leftover untagged words, thus completing the iterative process. To ensure consistency, each Brown file was completely tagged by one tagger.

In order to ensure the quality of the annotations in Semcor, after the corpus was annotated, every 11^{th} semantically tagged word was examined. If mistakes were found they were corrected. A list of particularly difficult words was created during the quality control phase, and each instance of the difficult words was then re-checked to ensure correctness. Finally, every 12^{th} tagged word was re-examined to give a new error rate, again correcting each mistake found.

The summary for the version of Semcor used with the WSD system described in chapter 6 is given in the Table 2.3. A particularly useful piece of information that

Category	Group Name			Total
	brown1	brown2	brownv	
unique noun senses	11399	9546	0	20945
unique verb senses	5334	4790	6520	16644
unique adjective senses	1754	1463	0	3217
unique adverb senses	1455	1377	0	2832
unique adjective satellite senses	3451	3051	0	6502
Total unique senses	23393	20227	6520	50140
Total tagged words	107118	86255	41607	234980

Table 2.3: Semcor Summary

is unfortunately not available for Semcor is the inter-annotator agreement. Such a

statistic would allow for an upper-bound to be set for computer-based WSD systems. A possible source for such information can be calculated from considering matching documents in the DSO corpus (Ng and Lee, 1996) which also uses a subset of the Brown corpus. Because information about the individual annotators, such as age and social background, is not provided, it is not possible to deduce what the reason for any disagreement could be. The inter-annotator agreement between the two corpora is 57% (Kilgarriff, 1998a).

2.3 CMU Link Grammar Parser

The Carnegie Mellon University (CMU) link grammar parser (Temperley, 1999) is used to annotate and determine the context for the WSD system described in chapter 6. The parser is used due to its flexibility, as words and rules can be added and changed as required. WordNet could be used to provide the basis of a dictionary for the parser, although this is not done for the work presented here as it would be an ambitious task by itself. The parser is also robust, returning partial structures when not all necessary constraints are satisfied for the words in the sentence and handling unknown words to some extent.

The CMU link grammar parser produces grammatical structures between words in a way related to dependency grammars. Each word is associated with directional connectors of different types to its left and right. A link between two words is formed if a left connector of one word can connect with the right connector of another word.

Figure 2.2: CMU Link Example

In total, there are 107 different link-types, with a number of further subscripts. For example, the previous example shows a determiner link “D” between the words “the” and “man”. The link’s subscript “s” means that the link between the words is for a singular relation. If the word had been “men”, the subscript of the link would be “m*” for plural (where “*” can be replaced with “c” or “u” to distinguish between

countable or mass nouns). A sentence is valid if all its words are connected in some way satisfying the required word rules and certain global rules. Word rules, specified in the parser's dictionary, describe the combinations of connectors possible for words, while global rules control the way words and links are limited. Two examples of global rules are the "crossing-link" rule and the "connectivity" rule. The "crossing-link" rule does not allow for links to cross each other, therefore the links in Figure 2.3 would be invalid.

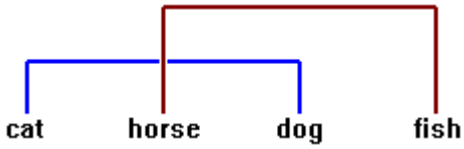


Figure 2.3: Example of Invalid Links Given the 'Crossing-Link' Rule

The "connectivity" rule ensures that a valid sentence must have all of its words connected, therefore the links in Figure 2.4 would be invalid.

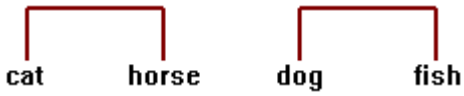


Figure 2.4: Example of Invalid Links Given the 'Connectivity' Rule

The global rules are defined in the parser's knowledge file. The total structure produced for an entire sentence is referred to as a linkage. More complex sentences are likely to produce a large number of alternative linkages.

Whilst the linkage structures do not obviously look like the typically known Chomskian parse trees, they can be used to produce traditional sentence structures. Version 4.0 of the CMU link grammar parser now has a feature to generate such structures automatically.

2.4 NLP application

During the course of this work, an application has been developed to assist in the development of the similarity measures and WSD systems produced. The application was

created using the Microsoft Foundation Classes (MFC) and handles NLP documents containing information about sentences, grammatical structures and semantic information in the form of sense labels from WordNet. The application consists of a number of ActiveX components wrapping the various tools described above. A further set of experimental tools have been incorporated within the application, written in either Prolog or C++. Figure 2.5 shows all the tools used in the NLP application. The additional

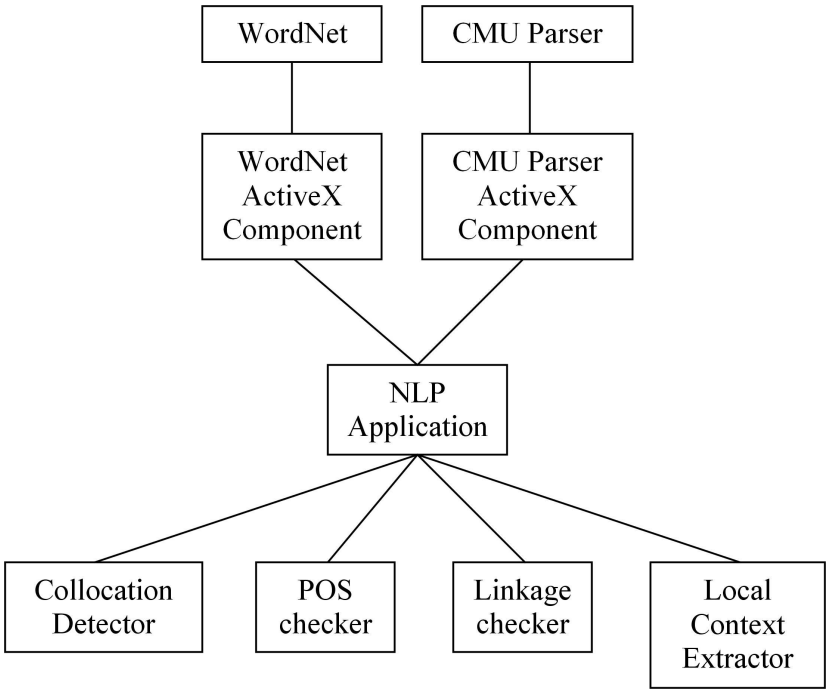


Figure 2.5: NLP Application Tools

tools are provided to assist users in reducing syntactical ambiguity before producing information for the WSD system:

- POS checker

  This tool reports to the user ambiguities, if they exists, for the POS of words in the sentence contained in the set of adequate linkages. For each possible ambiguous POS, the checker also returns the frequency of the POS, where frequency is defined as the number of linkages for which the word is tagged with the POS. The user can then systematically select the correct POS for each word and thus reduce the number of linkages considered for a sentence.

- Linkage checker

  This tool is similar to the POS checker, however it returns information about the ambiguous links available in the linkages, along with their frequencies according to the linkages. Again, this information can be used to assist in reducing the ambiguity of the linkages produced by the CMU parser.

- Local Context Extractor

  This tool extracts the local contexts for all words in a text according to the definition of context given in chapter 6. The local contexts are extracted from the first valid linkage for each sentence the CMU parser returns, where the first linkage is most likely to be correct. This information forms the basis of the corpus for the WSD system also described in chapter 6.

- Collocation Detector

  The collocation detector examines multi-word terms in a sentence to determine if they are treated as collocations according to WordNet. Any combinations of words detected to be potential collocations are reported to the user, therefore the phrase "breach of trust with fraudulent intent" can be interpreted as consisting of the terms "breach trust fraudulent intent", "breach_of_trust fraudulent intent" or "breach_of_trust_with_fraudulent_intent" according to WordNet.

Each of the "checkers" above are used to maximum efficiency if the user validates or invalidates the most frequent POS or Linkages first. This way, the largest number of linkages can be potentially reduced. The following list summarises the features of the application:

- Sentence linkages can be viewed graphically, rather than the ASCII based diagrams or relational structures produced by the CMU parser, and navigated via toolbar buttons. Linkages can also be manually invalidated using this graphical interface.

- Linkages can be reduced rapidly using a number of available tools. These changes are shown in the resulting linkage diagrams for the sentences, as invalid links are identified.

- The application handles multiple sentences and documents.

- All linguistic information about the text handled within a document is saved and loaded together with the text.
- The application can perform WSD for noun groups (See 4.5.2).

At this current stage, the WSD system developed for open-texts is not yet used directly by the NLP application, although the framework used is such that the system can be attached with ease and minimal additional work. Figure 2.6 shows a screenshot of the NLP application.

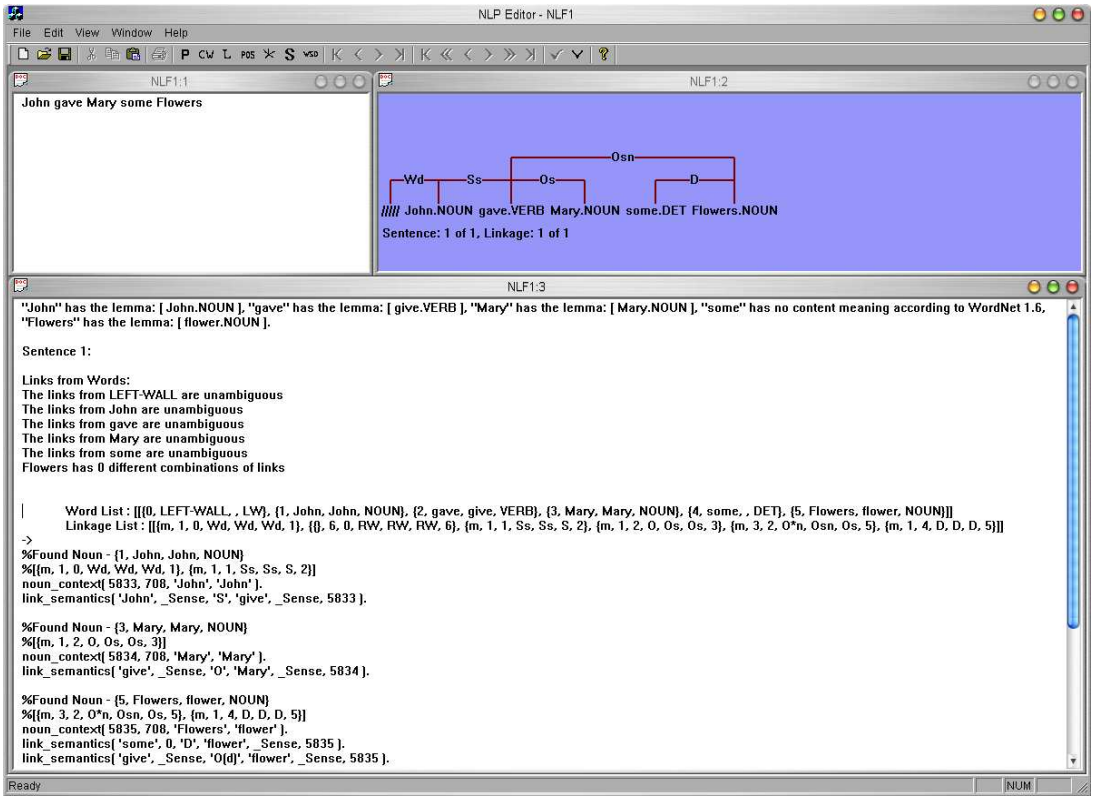


Figure 2.6: Natural Language Processing Application

Chapter 3

Introduction to Semantic Similarity

Semantic similarity has a long history in the field of artificial intelligence and natural language processing. Many of the ideas currently exploited are originally derived from the field of psychology, where similarity is believed to lie close to the core of cognition. As William James states "This sense of sameness is the very keel and backbone of our thinking" (James, 1950). In general, the aim of the work is to create a measure accepting two or more terms as input (where a term is a word, concept or word sense) and to return as output some classification of their similarity. Measures of general similarity (i.e. visual or semantic similarity) can be split into four psychological models:

- Geometric – Stimuli are represented in terms of their values on different dimensions.
- Feature-based – Stimuli are represented in terms of the presence or absence of weighted features.
- Alignment-based – Stimuli are represented in terms of alignment processes over structural representations.
- Transformational – Stimuli are represented in terms of transformation processes on sensory input to match with predetermined subconscious rules. Of the best-known transformation models is Chomsky's Transformational Grammar for modelling the human process of understanding syntax (Chomsky, 1957, 1965) (although Chomsky himself never claimed that his theories presented a psychological model).

Prior to creating such similarity measures, the type of similarity being measured must be defined.

This chapter introduces the field of semantic similarity, although discussion is restricted to techniques determining similarity between nouns. Section 3.1 introduces some basic terminology, explaining the details of different types of measure between words. Section 3.2 gives details of different representations used when distinguishing similarity between terms. Section 3.3 discusses the different resources typically used for measuring semantic similarity. Section 3.4 introduces a number of the current well-known, state-of-the-art techniques used for calculating semantic similarity. Finally, section 3.5 briefly discusses verb similarity.

3.1 Terminology

The term ‘similarity’ when applied to lexical information must be clearly understood. Three main distinctions of how word and word sense similarity can be defined are typically found in literature (Budanitsky, 1999):

- Semantic Relatedness
- Semantic Similarity
- Semantic Distance

This section defines the above distinctions.

3.1.1 Semantic Relatedness

Semantic relatedness between words makes use of information other than pure lexical semantics of concepts or words, and therefore measures of semantic relatedness require additional information than that found in WordNet’s lexical taxonomy. This extra information may be of a particularly subjective form unique to individuals, such as information about their personal view of the world. Consider the two words “strawberry” and “tennis”. Some people would associate some relation between the two words because strawberries are typically found at some tennis games. However it is difficult to see how any information in their semantic definitions could link the two. Other examples of such associative relations would include:

- “tennis” and “scone”
- “ice cream” and “summer”
- “car” and “journey”
- “car” and “petrol”
- “train” and “passenger”

Of the three common definitions of lexical similarity, these kinds of associations are the least explored area, mainly due to difficulties in producing adequate knowledge resources from which to define measures. It is also difficult to give a clear definition of semantic relatedness due to its inherently subjective nature. However, statistical techniques based on Latent Semantic Analysis (LSA) (Landauer and Dumais, 1997) or Context Vectors (Chen and You, 2002) are able to implicitly capture some of this type of relatedness if given enough training examples.

3.1.2 Semantic Similarity

Semantic similarity is a more restricted notion than semantic relatedness. It characterises similarity only in terms of the lexical semantics of words or word senses. Such a definition of similarity would assign low similarity for the previous word pair associations, as each word-pair shares little semantic information.

The increase in publicly available machine-readable dictionaries and semantic networks has stimulated the development of a large number of techniques to calculate semantic similarity. These techniques typically assign a scalar value denoting the similarity of two words or word senses according to a semantic taxonomy. Prior to these more recent techniques, other techniques were developed making use of thesauri (Morris and Hirst, 1991; Okumura and Honda, 1994), relying on the semantic relations implied by the thesaurus entries to give a more coarse grained similarity distinction.

3.1.3 Semantic Distance

Semantic distance describes how different two words are by showing how far apart they are semantically. Since only semantic information is taken into account, a measure of this type can be considered as the inverse of semantic similarity. The more “distant”

two words or senses are, the less similar they are. Indeed, most recent semantic similarity approaches use semantic distance in order to determine semantic similarity.

3.2 Representations of Similarity

The choice of representation for similarity varies with the task for which the similarity measures are intended. Techniques generally use one of the following representations for similarity:

- Boolean judgements

  Given the limitations of earlier knowledge sources, early systems typically gave results merely in terms of "similar" or "not similar" (Budanitsky, 1999). Whilst the techniques developed in the next chapter are designed to produce numerical values, later work in chapter 6 only considers boolean results from the techniques developed.

- Enumerated judgements

  Later techniques, such as thesaurus based measures (Morris and Hirst, 1991; Jarmasz and Szpakowicz, 2001a,b, 2003), use improved knowledge sources allowing for a coarse but more refined representation of similarity, compared with simple boolean judgements. Such a representation generally produced an answer from a predefined set of possibilities.

- Scalar judgements

  The goal of most modern techniques (Rada et al., 1989; Kozima and Furugori, 1993; Sussna, 1993, 1997; Wu and Palmer, 1994; St-Onge, 1995; Hirst and St-Onge, 1998; Richardson and Smeaton, 1995; Agirre and Rigau, 1995, 1996; Resnik, 1995a,b, 1999; Jiang and Conrath, 1997; Lin, 1997, 1998a,b,c; Leacock and Chodorow, 1998) is to assign numerical values of similarity to words. Whilst, in practice, the notion of assigning a numerical value can be deemed as an abstract task compared with human cognition, it allows for a finer distinction of similarity between different pairs of words. This representation is typically of greater use for a number of applications as it may be used to produce finer distinctions of similarity according to some target application's requirements. It is

also often useful to normalise the final results so that all word pairs are measured within a pre-set range of values.

3.3 Resources for Calculating Semantic Similarity

The resources used for calculation of semantic similarity play a fundamental role in a similarity measures. Such a resource must contain the information necessary to calculate similarity. Apart from a small number of exceptions using hand-tailored or specialised lexicons, the majority of well-known similarity measures make use of, or have been adapted to make use of, one of the following machine readable dictionaries:

- Roget's Thesaurus

  Prior to the public availability of structured machine-readable dictionaries, a large body of similarity measure work made use of various versions of Roget's Thesaurus. Although information is generally insufficient to calculate scalar values of similarity, the structure is sufficient to produce a number of different classes of similarity (Jarmasz and Szpakowicz, 2003). Researchers make use of the format of information within entries, such as the way information within entries is separated by punctuation marks, to produce sub-entries or groups of words. Some versions of Roget's thesaurus group entries into more general classes, allowing the exploitation of a simple hierarchy between words as a further source of information.

- Longman's Dictionary of Concise English

  Probably the most widely used resource in earlier work with machine readable dictionaries (Guthrie et al., 1996), and also the first dictionary to be publicly available to researchers (Budanitsky, 1999), is the Longman's Dictionary of Concise English (LDOCE) (Procter, 1978). The most significant influence that LDOCE has had on semantic similarity measures was the provision of some structure between words, although this structure is not given as explicitly or to the same level of detail as in WordNet. Information is organised into domains using subject fields, and box codes are used to hierarchically organise words. Further to the organisation of words within the LDOCE, the Longman Defining Vocabulary (LDV) was developed to give LDOCE a controlled vocabulary for

its headword definition. The collection of words was selected based upon results from West's (West, 1953) work about restricted vocabulary, producing a collection of 2,851 words in the LDV. All headword definitions in LDOCE are defined in terms of the LDV collection of words.

- WordNet

  WordNet (Miller et al., 1990; Fellbaum, 1998) was the first psycholinguistic dictionary available in a machine-readable form and has proven extremely influential in the field of similarity measurement. Details for this machine-readable dictionary are given in chapter 2. Currently, the majority of work concentrates on measuring similarity between nouns, almost exclusively using hypernym relations. The most likely reason for this may lie in the shape of the noun taxonomy. Compared to other parts of speech, the noun hypernym taxonomy of WordNet is deep, rather than wide, resulting in longer path lengths between nodes. Whilst the structure of the verb taxonomy is still fairly large and detailed, it seems that little positive work has emerged to this point making use of WordNet alone. Given the way people distinguish the similarity between verbs, verb argument structure is essential for measuring such similarity, and unfortunately WordNet is weak in this area. The taxonomies for adjectives and adverbs are far less developed than those for nouns and verbs, making them less suitable for calculating similarity without additional information.

3.4 Existing Techniques

The most common techniques for measuring semantic similarity can be split into the following general classes:

- Thesaurus techniques
- Taxonomic techniques
- Statistical techniques
- Hybrid techniques

Most recent techniques follow the geometric tradition of similarity; where word similarity is a metric of distance measured according to lexical relations (Tversky, 1977).

As standard, such geometric techniques assume the following conditions (Tversky, 1977):

- Minimality: $dist(A, B) \geq dist(A, A) = 0$
- Symmetry: $dist(A, B) = dist(B, A)$
- Triangle Inequality: $dist(A, B) + dist(B, C) \geq dist(A, C)$

Even though Tversky criticises the properties above, most modern techniques still abide by these properties. The next section introduces to the most recent well-known techniques according to the classification above.

3.4.1 Similarity Calculated from Thesaurus information

Some of the earliest techniques for the calculation of similarity between words made use of thesauri as the main knowledge source. Morris and Hirst (1991) used Roget's Thesaurus (Chapman, 1977) and Okumura and Honda (1994) used an equivalent Japanese thesaurus called Bunrui Goi Hyo (Shuppan, 1964). Given the limited information available in thesauri for calculating semantic similarity, the results of such techniques are typically presented using boolean or enumerated values, for instance the similarity between two words might be classified as either "close" or "not close" (Budanitsky, 1999). Another aspect of these techniques is that given the wide variety of related words within a single thesaurus entry, such techniques tend to detect semantic relatedness, rather than semantic similarity. As such, these techniques do not produce scalar values for similarity and are of limited use for many applications.

More recently, techniques have appeared using thesaurus information to produce similarity measures with more detailed distinctions of similarity. Whilst in many cases results from such techniques are given as numbers, these numbers still represent a ranked set of enumerations. Jarmasz and Szpakowicz (2001a,b, 2003) present a technique using Roget's Thesaurus of English Words and Phrases (Kirkpatrick, 1998) to measure semantic distance. Distances between two words are selected according to the organisation of information in the thesaurus. Distances are selected according to the following criteria relating two words:

- Length 0 – The same semicolon group of the thesaurus entry.

- Length 2 – The same paragraph of the thesaurus entry.
- Length 4 – The same part of speech entry in the thesaurus entry.
- Length 6 – The same head.
- Length 8 – The same head group.
- Length 10 – The same thesaurus sub-section.
- Length 12 – The same thesaurus section.
- Length 14 – The same thesaurus class.
- Length 16 – Both words are in the thesaurus.

Similarity is calculated by subtracting the path length from the maximum path length, therefore:

$$SemanticDistance = L \tag{3.1}$$

$$SemanticSimilarity = 16 - L \tag{3.2}$$

3.4.2 Similarity Measures Based on Taxonomies from a Machine-readable Dictionary

A much more widely researched approach is to use the relationships between words contained in modern machine-readable dictionaries (MRDs). The premise of such techniques is that considering lexical relations as edges in trees gives a way to measure the geometric distance between two words or word senses. Such a conceptual distance measured from path lengths can be used to calculate how similar the two words are. Rada et al. (1989) present one of the earliest path length techniques. Given the inherent simplicity of such approaches, a number of different techniques have been developed over a relatively short period of time that can all be applied using WordNet's lexical taxonomy.

Rada et al. (1989) implemented a technique for measuring the similarity of two concepts solely considering the path-lengths between them according to a semantic network. Using the Medical Subject Headings (MeSH) knowledge source (MeSH,

1995), a semantic network consisting of medical terms, Rada et al. show that semantic distance can be calculated simply from the shortest edge distance between two nodes in the semantic network. Rada et al. also show that such a path length alone is enough to satisfy Tversky's properties of a distance metric (Tversky, 1977). The edges used in MeSH conform closely to WordNet's hypernym taxonomy, however they occasionally reflect holonym (part-of) information. Semantic distance is given as:

$$dist_{Rada}(c_1, c_2) = \text{minimum number of edges from } c_1 \text{ to } c_2 \tag{3.3}$$

where c_1 and c_2 are terms or concepts in a semantic network or taxonomy. Rada's algorithm can be adapted to measure similarity with the following changes:

$$sim_{Rada}(c_1, c_2) = 1/dist_{Rada}(c_1, c_2) \tag{3.4}$$

Resnik (1995a,b, 1999) gives a more refined approach to measuring similarity using such a distance metric, along with an algorithm specifically tailored to work with polysemous words:

$$sim_{Resnik}(w_1, w_2) = 2d_{max} - \left(\min_{\substack{c_1 \in senses(w_1), \\ c_2 \in senses(w_2)}} len(c_1, c_2) \right) \tag{3.5}$$

where w_1 and w_2 are words, c_1 and c_2 are senses of w_1 and w_2 respectively and d_{max} is the maximum depth of the taxonomy.

Whilst Rada's simple approach takes into account differences in the semantic information of two words, it makes no attempt to use information common to both words or concepts, and the technique also assumes that all edges in the semantic network have equal distances. Whilst this technique works well with the MeSH knowledge source, Richardson and Smeaton (1995) found that the inherent irregularities of a taxonomy such as WordNet have a negative impact on Rada's approach.

Sussna (1993, 1997) addresses the issue of non-uniform distances between nodes in WordNet using a depth-relative scaling technique. This takes into account different weights for WordNet's various semantic relationships, and considers depth in order to assign shorter distances to relationships of nodes found deeper in taxonomies. Firstly, each relation, r, is assigned a minimum and maximum weight as shown in Table 3.1.

The actual weight for the relationship between two directly connected nodes is given

Relation (r)	$\min_r$	$\max_r$
Synonymy	0	0
Hypernymy	1	2
Hyponymy	1	2
Holonymy	1	2
Meronymy	1	2
Antonymy	2.5	2.5

Table 3.1: Sussna's Lexical Relation Weights

in equation 3.6.

$$w(c_1 \underset{r}{\rightarrow} c_2) = max_r - \frac{max_r - min_r}{n_r(c_1)} \tag{3.6}$$

where $n_r(c_1)$ is the number of arcs of relation r connected to c_1. Note that given the above definition, the weight of a relationship between two nodes is dependant on the direction in which the relationship is used. Taking "hammer#2" as an example, the weight assigned from "hammer#2" to "hand tool#1" via a hypernymy relation is calculated using c_1 = "hammer#2", c_2 = "hand tool#1", r = "hypernymy", min_r = 1, max_r = 2 and n_r = 1 according to WordNet 1.6, resulting in a weight of 1.

The actual distance between two directly connected nodes is calculated as the average weights of the relationship in both directions. Sussna also adjusts the distance of two nodes using the depth where the relationship occurs in the taxonomy. The resulting distance measure is shown in 3.7.

$$dist_{Sussna}(c_1, c_2) = \frac{w(c_1 \underset{r}{\rightarrow} c_2) + w(c_2 \underset{r'}{\rightarrow} c_1)}{2d} \tag{3.7}$$

where r' is the inverse of relation r, a relation connecting c_1 and c_2, and d is the depth of the relationship, that is if r is a hypernym relation, r' is the hyponym relation between c_2 and c_1. Using the depth of the relationship in such a way ensures that smaller distances are assigned between nodes found deeper in the taxonomy. For the previous example with "hammer#2" and "hand tool#1", r' = "hyponymy" giving $w(c_2 \underset{r'}{\rightarrow} c_1)$ = 1.97, and this in turn gives $dist_{Sussna}(c_1, c_2)$ = 0.19 where d = 8, according to WordNet 1.6. Using this definition of distance between two adjacent nodes, the total distance between any two arbitrary nodes is calculated as the sum of the distances of the nodes

along the path connecting both nodes, as with Rada et al.'s technique. Similarity is calculated from the distance using a similar approach to Rada et al. (1989).

Wu and Palmer (1994) introduce a metric for measuring similarity between two concepts that, whilst assuming equal distance for all relations in a taxonomy, makes use of information common to these concepts. The general form of the metric, shown in equation 3.8, measures the ratio of semantic information common to both concepts to the amount of total semantic information.

$$sim(c_1, c_2) = \frac{\text{common information}(c_1, c_2)}{\text{total information}(c_1, c_2)} \tag{3.8}$$

Similarity is measured using the path length between nodes in a conceptual hierarchy, for instance consider Figure 3.1. In this figure, c_1 and c_2 represent two arbitrary con-

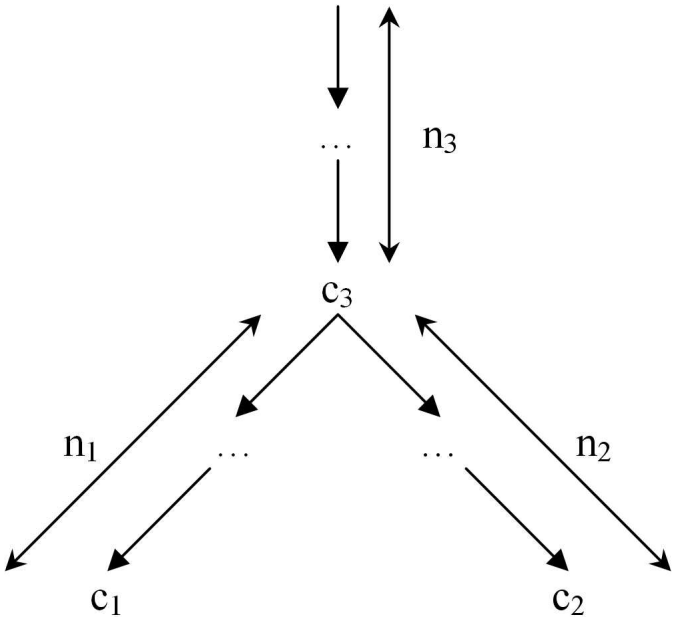


Figure 3.1: An Example of a Conceptual Hierarchy

cepts, c_3 represents the deepest concept common to both c_1 and c_2, and n_1, n_2 and n_3 are path lengths within the taxonomy. For hypernym taxonomies, c_3 is referred to as the most informative subsumer (MIS) of both c_1 and c_2 in later discussions. The path from the root of the taxonomy to the MIS denotes the semantic information common to the two concepts being compared, and the nodes below the MIS represent semantic

information distinct to c_1 and c_2. The path lengths from those parts of the taxonomy are used to measure similarity as in equation 3.9.

$$sim_{Wu\&Palmer}(c_1, c_2) = \frac{2d_3}{d_1 + d_2} \tag{3.9}$$

where d_1, d_2 and d_3 are the depths of c_1, c_2 and c_3 respectively. Considering the two word senses "hammer#2" and "drill#1", the MIS is "tool#1", d_1 = 8, d_2 = 9 and d_3 = 6, giving a total similarity of 0.71 according to WordNet 1.6 and Wu and Palmer (1994).

St-Onge (1995) and Hirst and St-Onge (1998) introduce a measure carefully designed to make use of further relations in WordNet, other than and including hypernym relations. The main intention behind this work was to use the ideas developed by Morris and Hirst (1991), which used Roget's thesaurus, with WordNet. St-Onge defines 3 types of relations according to WordNet:

1. Extra-Strong – This only occurs if both words are identical.

2. Strong – This occurs if one of the following conditions is satisfied:

   - The two words can be synonyms of each other.
   - The two words or concepts are related by a horizontal link. St. Onge defines a horizontal link as one of antonymy, similarity and "see also" relations from WordNet.
   - One word is a compound word or phrase that contains the other word, and the two words are connected via a link in WordNet.

3. Medium-Strong – A number of allowable relational patterns are defined. Any configuration of relationships up to a path length of 5 that fit with the allowable patterns are said to constitute a medium-strong relation. Such patterns were carefully selected whilst considering the psycholinguistic relationships they represent in order to ensure their validity. In general, all patterns containing no more that one change in direction are allowed. Full details of allowable and disallowed patterns are given in (St-Onge, 1995; Hirst and St-Onge, 1998).

Given this framework, Hirst and St. Onge calculate similarity as follows:

- If w_1 and w_2 are related via an extra-strong relation, $sim_{Hirst\&StOnge}(w_1, w_2) = 3C$

- If w_1 and w_2 are related via a strong relation, $sim_{Hirst\&StOnge}(w_1, w_2) = 2C$
- If w_1 and w_2 are related via a medium-strong relation,
  $sim_{Hirst\&StOnge}(w_1, w_2) = C - dist_{Rada}(w_1, w_2) - (k \times \delta)$,
  where C and k are constants, δ is the number of direction changes in the path from w_1 to w_2, and $dist_{Rada}(w_1, w_2)$ is the path length from w_1 to w_2.

A further similarity measure using path length in WordNet's hypernym structure is given by Leacock and Chodorow (1998). Again, this measure makes use of a semantic distance between two concepts similar to Rada's. However, the final value is normalised against the maximum depth of the taxonomy. The measure presented takes words as input, rather than word senses or concepts, as shown in equation 3.10.

$$sim_{Leacock\&Chodorow}(w_1, w_2) = -\log \frac{\min\limits_{\substack{c_1 \in senses(w_1),\\ c_2 \in senses(w_2)}} len(c_1, c_2)}{2d_{max}} \tag{3.10}$$

where d_{max} is the maximum depth of the taxonomy and $len(c_1, c_2)$ is the number of nodes connecting c_1 and c_2, rather than the number of edges between the nodes, therefore synonyms are assigned a length of 1 apart. According the WordNet 1.6 with a maximum hypernym taxonomy depth of 17, Leacock and Chodorow's similarity measure will range from 0 to almost 5.1 (using $\log_2$). Considering the two words "hammer" and "drill", $\min len(c_1, c_2)$ = 4 over all senses of "hammer" and "drill", giving a similarity of 3.09 according to WordNet 1.6 and Leacock and Chodorow (1998).

A number of other measures appear in recent publications that do not explicitly publish sufficient details in order to be reproduced, such as Richardson and Smeaton (1995) and Agirre and Rigau (1995, 1996). The latter technique employs a different approach to the use of a semantic network for the calculation of semantic similarity. Their approach was to develop a measure sensitive to the following conditions:

- The shortest length of any path connecting two concepts.
- The depth of concepts in a taxonomy in order to assign higher similarity to deeper concepts.
- The conceptual density of the taxonomy, such that senses in denser taxonomies are deemed closer than those in sparser taxonomies.

A metric for the conceptual density of a hierarchy for use in measuring semantic similarity is given. However, no explicit formula for the calculation of similarity is given. The definition of conceptual density is given in equation 3.11 and 3.12.

$$CD(c,m) = \frac{\sum_{i=0}^{m-1} nhyp_i^{0.2}}{descendants(c)} \quad (3.11)$$

$$descendants(c) = \sum_{i=0}^{h-1} nhyp_i \quad (3.12)$$

where c is the top most node of a sub-hierarchy containing the concepts under consideration, $nhyp$ is the average number of hyponyms contained in c's sub-hierarchy, m is the number of concepts under consideration within c's sub-hierarchy and h is the height of c's sub-hierarchy. The value 0.2 used in the formula was selected experimentally in order to fine-tune the algorithm. The techniques introduced in chapter 4 show some similarity with Agirre and Rigau's approach, although the lack of published results, a complete similarity measure and details of how c is selected means that no direct comparison is possible.

3.4.3 Similarity Calculated using Statistical Information

Kozima and Furugori (1993) automatically generate a semantic network called Paradigme using entries from LDOCE whose headwords belong to the LDV. The extracted sub-dictionary, referred to as Glosseme, contains 2,851 entries from LDOCE containing 101,861 words. The network, referred to as Paradigme, is generated from Glosseme by creating a node for each headword, and creating links between each headword node and all other nodes for headwords contained in the dictionary entry's definition. Given this technique for generating the semantic network, the links of the network are defined as one of two types:

- Référant links - Where a node, x, is linked to another node, y, because y contains a word contained in the definition of x.
- Référé links - Where a node, x, is linked to another node, y, because x contains a word contained in the definition of y.

Each link in turn is also assigned a 'thickness' calculated from the frequency of its

headword in Glosseme and other sources. The result is a 2,851 node semantic network related via 295,914 unnamed weighted links.

Using the semantic network produced, Kozima and Furugori (1993) calculate similarity by analysing the spreading activation of the network. An activation value, denoted by av_n, is associated with each node of the network, and equation 3.13 calculates for each iteration T, $av_n(T+1)$.

$$av_n(T+1) = \varphi\left(\frac{R_n(T) + R'_n(T)}{2} + e_n(T)\right) \tag{3.13}$$

where T is the current iteration of activity, $R_n(T)$ and $R'_n(T)$ are the composite activities of the référants and référés of n at time T. φ is a function normalising the values of av_n to lie within the range $[0, 1]$ (see Kozima and Furugori (1993)). The similarity of words w_k and w_l is calculated as follows:

1. The activity for all nodes in Paradigme is reset.

2. Node k, associated with word w_k, is activated with strength $e_k = s(w_k)$. The term $s(w_k)$ is the significance of w_k, calculated using the normalised information content value according to the 5,487,056-word West corpus (West, 1953). The normalised information content value is calculated using equation 3.14.

$$s(w) = \frac{\log(freq(w))}{\log(1/C)} \tag{3.14}$$

   where C is the word count of the entire corpus.

3. The activation pattern for the network is calculated over 10 iterations.

4. Similarity is calculated using equation 3.15.

$$sim_{Kozima\&Furugori}(w_k, w_l) = s(w_1) \times \alpha(P(w_k), w_l) \tag{3.15}$$

   where $\alpha(P(w_k), w_l)$ is the activation value for w_l in the pattern produced by w_k in the activation pattern $P(w_k)$ produced by Paradigme.

This gives a way of measuring the similarity between any two words from the LDV collection. However, the LDV words only account for 5% of the total words contained in LDOCE.

In order to extend the measure to any word of LDOCE, Kozima and Furugori adapt their algorithm to use of the definitions of words from LDOCE as input for the similarity measure. Using the words in the definitions of one of the two words, any words also contained in the LDV set are activated with strength using equation 3.16.

$$sim_{Kozima\&Furugori}(W, W') = \psi\left(\sum_{w' \in W'} s(w') \times \alpha(P(W), w')\right) \tag{3.16}$$

where $P(W)$ is the pattern produced using all the words in the set W.

3.4.4 Hybrid Approaches

Some of the most accurate techniques developed recently augment lexical taxonomies with statistical information. The augmented information helps to reduce problems produced by irregularities found in practical lexical taxonomies (Resnik, 1995a,b, 1999). The earliest technique to do this using WordNet's lexical taxonomy is Resnik's Information Content similarity measure (Resnik, 1995a,b, 1999). Resnik's approach is to add the information $p(c)$ to each synset in WordNet, where $p(c)$ is defined in terms of concept frequencies, as given in equation 3.17.

$$freq(c) = \sum_{n \in words(c)} count(n) \tag{3.17}$$

where $words(c)$ is the set of words subsumed by the synset c, and $count(c)$ is calculated from a corpus. In Resnik's work with WordNet, the set of subsumers of a concept is given as the hypernyms of the concept.

$$p(c) = \frac{freq(c)}{N} \tag{3.18}$$

where N is the total number of nouns contained in the given corpus. Resnik uses the Brown Corpus of American English (Kucera and Francis, 1967; Francis, 1980; Francis and Kucera, 1982) containing 1,014,232 words of text from a range of genres. Similarity is calculated from the information content (Ross, 1976) of the most informative subsumer (MIS) of the two words, quantified using the negative log likelihood of the

synset, as shown in equation 3.19.

$$sim_{Resnik}(c_1, c_2) = \max_{c \in S(c_1,c_2)} (-\log p(c)) \tag{3.19}$$

where $S(c_1, c_2)$ is the set of concepts common to both concepts c_1 and c_2. Only the deepest node for the concept in the set $S(c_1, c_2)$, the MIS, is used as this node will have the largest negative log-likelihood. Notice that no further information about the taxonomy is used in the calculation of similarity. Again, Resnik's work only uses hypernym relations from WordNet's lexical taxonomy.

The similarity between two words is deemed to be the maximum similarity between any two senses of the words, and is calculated in equation 3.20

$$sim_{Resnik}(w_1, w_2) = \max_{\substack{c_1 \in senses(w_1), \\ c_2 \in senses(w_2)}} (sim_{Resnik}(c_1, c_2)) \tag{3.20}$$

A number of criticisms have been made about Resnik's approach. Firstly, similarity is not assigned in a standard normalised scale across words. This is most noticeable in the similarity of synonyms, and even the similarity of a word with itself, as similarity in these situations varies across words where one might assume it should not. This leads to "exaggerations" in the content values, depending on the shape of the taxonomies used in the calculation of similarity (Richardson et al., 1994; Richardson and Smeaton, 1995). Further criticism is made that such a similarity measure makes no further use of WordNet's lexical taxonomy's structure and relationships, and therefore any concepts sharing the same most informative subsumer will be assigned equal similarity.

Jiang and Conrath (1997) refine the notion of similarity measures using information content, making use of further information from the structure of the lexical taxonomy. The technique measures semantic distance between two words considering the information content of both the most informative subsumer of two concepts and the information content of the concepts of the words themselves. This way the measure considers both common information and disjoint information between two words.

$$dist_{Jiang\&Conrath}(c_1, c_2) = 2 \log p(mis(c_1, c_2)) - (\log p(c_1) + \log p(c_2)) \tag{3.21}$$

where $mis(c_1, c_2)$ is the MIS of c_1 and c_2.

Lin (1997, 1998a,b,c) gives a further information content approach, this time also

addressing the problem of normalised similarity values across words. The technique is related to Wu and Palmer's approach (Wu and Palmer, 1994) in that it measures the ratio of information shared by two concepts against their disjoint information. This is shown by equation 3.22.

$$dist_{Lin}(c_1, c_2) = \frac{2 \log p(mis(c_1, c_2))}{\log p(c_1) + \log p(c_2)} \tag{3.22}$$

All these hybrid techniques solely make use of WordNet's hypernym relation. Such techniques could be improved further by considering further relations to calculate semantic similarity. Some work, such as (Richardson et al., 1994; Richardson and Smeaton, 1995), examine the possibility of using further relations, although work is still ongoing.

3.5 Discussion of Verb Similarity

Whilst most of the current state-of-the-art similarity measures making use of WordNet's taxonomy are restricted to nouns, finding equivalent measures for measuring similarity between verbs is not easy. This may be because WordNet's taxonomy lends itself well for similarity measures using path length as the noun taxonomy is deep meaning that a reasonable variation in distance exists between noun senses. The same cannot be said in WordNet for words belonging to other gramatical classes. Resnik and Diab (2000) is the most notable work currently available using WordNet to evaluate verb similarity. They adapt Resnik's earlier information content approach from (Resnik, 1995a,b, 1999) to evaluate the similarity of verbs. Experimental results showed that results are poorer for verbs compared with nouns, and that the average inter-agreement rate for human evaluation of the similarity of verb pairs is also lower, suggesting that "word similarity is harder for subjects to quantify for verbs than for nouns". All other path length based similarity measures can also be used with WordNet's verb taxonomy, however current research chooses not to present results for verbs. This suggests, along with the fact that WordNet's verb taxonomy is far shallower than its noun taxonomy, that the type of information present in WordNet is less suited for directly measuring similarity amongst verbs.

Chapter 4

Using Lexical Taxonomies for Measuring Semantic Similarity

The previous chapter defined semantic similarity, and surveyed the various techniques that have been created to automatically calculate the similarity or distance between two word senses. This chapter introduces a number of new techniques for calculating semantic similarity between nouns. In order to improve on the current body of work, a number of difficulties are considered and strategies are proposed to tackle these difficulties. The work presented here was first presented by Dionisio et al. (2001). A copy of the seven page version of the paper is included in appendix A.

The first section discusses the difficulties arising for techniques making use of WordNet's lexical taxonomy for calculating the similarity of two concepts. Section 4.2 re-visits the question of what constitutes similarity and introduces a number of axioms which characterise desirable qualities for the results of techniques making use of WordNet's lexical taxonomy. These axioms are introduced with a description followed by a logical representation of the axiom. Section 4.3 introduces a number of hypotheses about WordNet's taxonomy that form the basis of the similarity measures introduced in section 4.4. Section 4.4 presents several new similarity measures based upon variations of the ideas introduced in section 4.2 and 4.3. Section 4.5 evaluates the quality of the results from the measures introduced in section 4.4. Finally, section 4.6 describes further work arising from the ideas presented here, and section 4.7 summarises the chapter.

4.1 Problems with Current Techniques

The most widely recognised issue in assessing similarity between words or word senses using WordNet's taxonomy arises from irregularities within its taxonomy (Richardson et al., 1994; Resnik, 1995a,b, 1999; Leacock and Chodorow, 1998). Inspection of different parts of the taxonomy reveals aspects that are unhelpful in trying to replicate human judgement about similarity, for instance:

- There is no uniform way in which senses are split into subsequent hypernyms, making some sub-hierarchies more developed than others.
- There are missing word senses.
- The taxonomy includes terms that are not in most people's regular vocabulary, such as technical terminology.
- Some relations that seem natural between words do not exist.
- Some words have more than one definition, where the extra definitions seem superfluous. This is partly due to the fine-grained nature of WordNet.

4.1.1 Different Levels of Sub-hierarchy Development

Figure 4.1 shows how sub-hierarchies of WordNet's noun taxonomy can show large differences in how detailed and developed they are. Sub-classes of "animal#1" illustrate a highly developed taxonomy, including detailed sub-classifications of different types of animals. This can be seen in the detail of the taxonomy between "cat" and "animal" in Figure 4.1a. Typically, a path length of over 4 hypernyms is used to sub-classify different biological classes of animals, thus making these structures reasonably deep. In contrast, the sub-hierarchy for "person#1" tends to be very shallow, and does not contain the detailed sub-classifications found in the animal sub-hierarchy.

Techniques that only make use of hypernym relations to calculate similarity and assume that hypernym relations always express the same level of generalisation fall foul of such irregularities. Many animal nouns, such as cat, dog, rat, etc... have alternative meanings relating to different types of people. The following list shows the senses and glosses of the words "cat" and "dog" that refer to a type of person. Note that in some cases a synonym of "cat" or "dog" is used in the glosses:

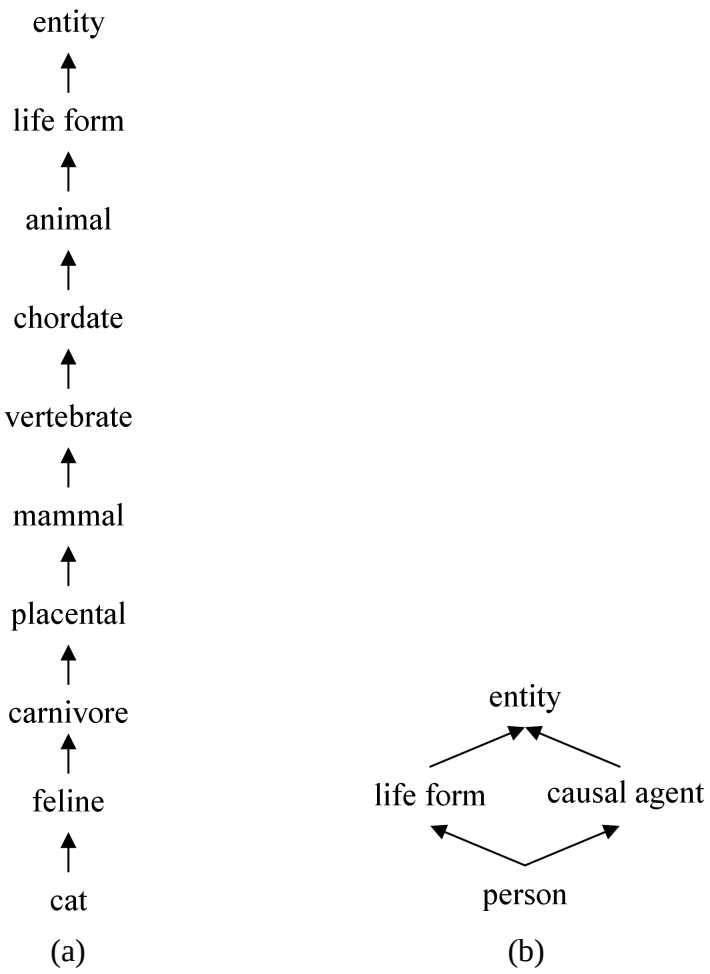


Figure 4.1: Hypernym Taxonomy for “cat#1” (a) and for “person#1” (b)

cat#2 – (an informal term for a youth or man; “a nice guy”; “the guy’s only doing it for some doll”)

cat#3 – (a spiteful woman gossip; “what a cat she is!”)

dog#2 – (a dull unattractive unpleasant girl or woman; “she got a reputation as a frump”; “she’s a real dog”)

dog#3 – (informal term for a man: “you lucky dog”)

dog#4 – (someone who is morally reprehensible; “you dirty dog”)

As a result, when considering two polysemous nouns that normally refer to animals, but contain senses referring to types of people, their "people" definitions will be assigned much higher similarity values when path length alone is used as a measure. This is clearly an undesirable situation for all path length based similarity measures, but one that occurs often in WordNet.

4.1.2 Missing Word Senses

The previous examples show that WordNet is a huge lexical resource giving fine-grained distinctions between definitions. However, some everyday word meanings are still missing, for instance:

- The word "chip" has no reference to its equivalent British meaning, as in "fish and chips".
- The word "Greece" has no reference to ancient Greece, nor does WordNet contain an entry for "ancient Greece".
- The word "fiducial" is missing a sense for when it is used as a reference or comparison "a fiducial mark".
- There is no entry in WordNet for "viva", not for its examination meaning or any of its other alternatives.

4.1.3 Terminology in Hypernym Structures

A number of the highly developed substructures within WordNet relate to a large number of scientific, or domain specific terms. Such terms shall be referred to as non-layman terms. Comparing two different hypernym structures, such as Figure 4.1a and 4.1b, it can be seen that an algorithm based on path lengths would generally assign a greater similarity to pairs of senses with less technical hypernym structures. When people make decisions about similarity in such situations, the scientific terms included in Figure 4.1a would not normally be taken into account. Most people would not even consider such terms, even if they are known, as these terms are normally only used to group things into abstract families.

4.1.4 Missing Relations

The relation that an “animal cub” is a “young mammal” is made explicitly in WordNet. However, there is no relation to the fact that a “young mammal” is also a “mammal”. This makes “animal” the most informative subsumer (MIS) between a kind of “young mammal” and elder equivalent. It would seem natural that by virtue of an animal x being a younger version of an animal y, that x also be a y such as is the case between “young mammal” and “mammal”. This is also the case for all hyponyms of “cub”, including “bear cub”, “lion cub” and “tiger cub” where no explicit relation is made between the cub and the class of animal that the cub belongs to. Such situations extend to other WordNet relations such as meronymy.

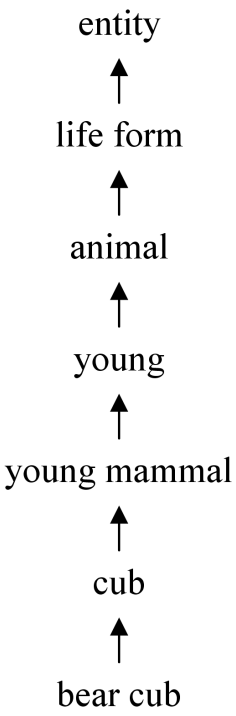


Figure 4.2: Hypernym Taxonomy for “bear cub#1”

4.1.5 Unnecessary Additional Word Senses

Examples can be found within WordNet of words that contain sense distinctions that may be considered overly fine-grained for calculating semantic similarity, such as the

word carnivore:

carnivore#1 – (terrestrial or aquatic flesh-eating mammal; terrestrial carnivores have four or five clawed digits on each limb)

carnivore#2 – (any animal that feeds on flesh: “Tyrannosaurus Rex was a large carnivore”; “insectivorous plants are considered carnivores”)

This brings about undesirable situations in the assessment of word similarity, as it can produce situations where senses of other words are related to only one of the very closely related senses of a word. For instance, a “canine#2” (any member of the canine family, such as “dog#1”) is related via hypernymy to “carnivore#1” above but not related to “carnivore#2”. However, a “canine#2” is an animal that feeds on flesh.

4.1.6 Problems for Hybrid Similarity Measure Techniques

Hybrid methods try to avoid some of the problems above by making use of statistical information. However, they fall foul of other problems. The most common problem for such hybrid statistical approaches is due to the lexical acquisition bottleneck problem (Gale et al., 1993), where insufficient examples of words or word senses are available to train classifiers that generalise well to new examples. This causes these statistical techniques to overly prefer senses within particular domains of meanings due to bias introduced by their training data.

Some of the hybrid methods also do not make use of the information contained within the intermediate structures between the two meanings being tested and the MIS. Techniques, such as Resnik’s information content approach, assign equal values to a large number of meanings when there is an obvious, even if only slight, difference in their similarities. The consequence of this is that similarity distinctions are coarser and bias may be present within sub-hierarchies for which more data is available.

4.2 What Constitutes Similarity in a Lexical Taxonomy?

Before creating a similarity measure, it helps to revisit the problem of word similarity and define what aspects of a lexical taxonomy help in assessing similarity between terms. The approach taken here is to define semantic similarity between two terms in a similar approach to that taken by Lin (1997). Lin makes a number of assumptions

in order to define similarity according to WordNet. Lin then produces a measure that is "proven" by ensuring it satisfies the initial assumptions made about similarity. The following axioms introduce a new general definition of similarity calculated using a lexical taxonomy. To assist in understanding the logical representation given for each of the axioms, each axiom is introduced with an informal description.

4.2.1 Axiom 1: Synonymy

It is clear that synonyms represent terms with the exact same meaning, therefore synonymy represents the closest form of similarity between terms (see 2.1.1). Thus it seems natural that a similarity measure should assign synonyms an upper bound value, as no two words can be any more similar than two synonyms. While this is an advantage, development of the measures is not restrained to guarantee this condition as long as synonymy is still treated as the most similar state between any two distinct terms.

$$
\begin{aligned}
&\forall x, y : sense \cdot \\
&\quad x \in synonyms(y) \land \\
&\quad s = sim(x, y) \Rightarrow \\
&\qquad \forall z : sense \cdot \\
&\qquad\quad sim(x, z) \leq s \land \\
&\qquad\quad sim(y, z) \leq s
\end{aligned}
$$

where $synonyms(w)$ is the set of synonyms for a word sense w.

4.2.2 Axiom 2: Hypernymy

The hypernyms of any sense are closely related to the original sense. The similarity represented by this relation is related to the distance from a word sense to one of its inherited hypernyms along a hypernym tree. The closer a word sense is to one of its inherited hypernyms, the higher the similarity shared by the two senses.

$$
\begin{aligned}
&\forall x, y, z : sense \cdot \\
&\quad y \in hypernyms(x) \land \\
&\quad z \in hypernyms(x) \land \\
&\quad distance(x, y) < distance(x, z) \Rightarrow \\
&\qquad sim(x, y) > 0 \land \\
&\qquad sim(x, z) > 0 \land \\
&\qquad sim(x, y) \geq sim(x, z)
\end{aligned}
$$

where $hypernyms(x)$ is the set of all hypernyms of a word sense x and $distance(x, y)$ is the path distance between x and y.

4.2.3 Axiom 3: Depth of the Most Informative Subsumer in the Taxonomy

Senses that are common to the hypernym structures of two other word senses can be used to determine the information common to the two word senses. Assuming two separate pairs of word senses, both with equal path distance between each other, the pair that shares the deepest MIS in the taxonomy should be deemed as more similar, as this pair shares more common information.

$$\begin{aligned}
&\forall a, b, c, x, y, z : sense \cdot \\
&\quad z = MIS(x, y) \wedge \\
&\quad c = MIS(a, b) \wedge \\
&\quad distance(x, y) = distance(a, b) \wedge \\
&\quad depth(z) > depth(c) \Rightarrow \\
&\qquad sim(x, y) \geq sim(a, b)
\end{aligned}$$

where $depth(a)$ is the depth of a sense a in a given hypernym structure and $MIS(a, b)$ is the MIS for two senses, a and b.

4.2.4 Axiom 4: Meronymy/Holonymy

The use of meronymy/holonymy relations to calculate the similarity between two senses needs to be handled with care. In general, it is agreed that such relations contribute toward similarity (Budanitsky, 1999; Budanitsky and Hirst, 2001). However, it has been difficult to define effective similarity measures that take advantage of these relations. Considered here is a fairly restricted use of meronymy. Two senses, x and y, share some similarity if x is an inherited meronym of y or vice versa, even if x and y share no common subsumer in their hypernym structures.

$$\begin{aligned}
&\forall x, y : sense \cdot \\
&\quad (x \in inherited_meronyms(y) \vee y \in inherited_meronyms(x)) \Rightarrow \\
&\qquad sim(x, y) > 0
\end{aligned}$$

where $inherited\_meronyms(x)$ is the set of all meronyms of x, including the meronyms of meronyms.

As holonymy is the inverse of meronymy, it is implicitly handled in the above

definition. For the purposes of this work and due to the difficulties of using meronym relations in similarity measures, not all the new measures produced are guaranteed to follow this axiom. This will allow the usefulness of this axiom to be tested.

4.2.5 Axiom 5: Co-ordinate terms

Co-ordinate terms are already deemed to be similar according to the previous axioms. However, it helps to give this relation a special status. For the purposes of this work, the definition of co-ordinate terms will be extended to include senses that share a common hypernym that are "generalised" by the hypernym by a similar amount (see 4.3.2). Terms sharing a similar level of "generalisation" from their MIS are deemed to be more similar than words at different levels of "generalisation".

$$\begin{aligned}&\forall x, y, z, m : sense \cdot \\ &\quad m = MIS(x, y) = MIS(x, z) \wedge \\ &\quad (|Gen(x, m) - Gen(y, m)| < |Gen(x, m) - Gen(z, m)|) \Rightarrow \\ &\qquad sim(x, y) > sim(x, z)\end{aligned}$$

where $Gen(a, b)$ is the generalisation of b to its hyponym a.

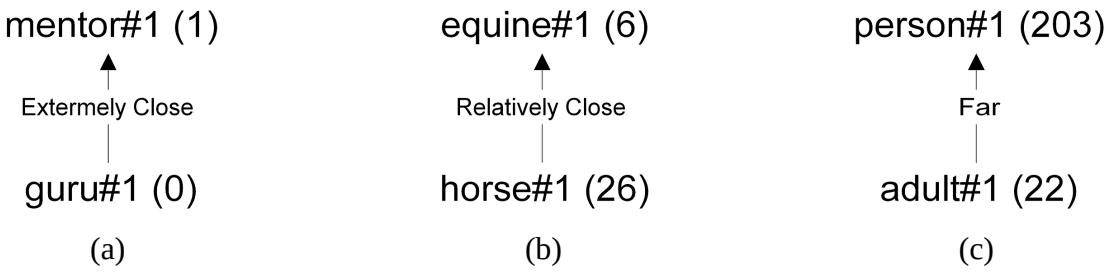
4.3 Towards a Better Similarity Measure

In addition to the new axioms introduced earlier in the chapter, the following new hypotheses are introduced regarding the use of taxonomies to assess similarity between words and word senses.

4.3.1 Hypothesis 1: Hyponym Branching Information Adjusts Hypernym Path Lengths

Within a hypernym hierarchy, sub-hierarchies differ in the granularity of development. The number of hyponyms a word sense has can be seen to influence the perceived distance between the word sense and its hyponyms. The hypothesis is that word senses with fewer hyponyms have a closer relation to their hyponyms, and as such, the hypernym distance between a word sense and its hypernym is related to the number of hyponyms it has.

Figure 4.3, shows three examples of word senses and their associated hypernyms.



(a) (b) (c)

Figure 4.3: Hyponym Branching Adjusted Hypernym Distance Examples

To the right of each word sense is a label showing the number of hyponyms it has. This number can be used to calculate how close hyponym and hypernym relations are as it gives an indication of the level of abstraction or generalisation between a word sense and its hyponyms. Figure 4.3a shows a close relation as the only hyponym of "mentor#1" is "guru#1" and differences between "mentor#1" and "guru#1" are minimal; Figure 4.3b shows a relatively close hypernym relation between "equine#1" and "horse#1". However, it is not as close as the relation between "mentor#1" and "guru#1" as "horse#1" is only one instance of a hyponym of "equine#1" out of a possible 6 different hyponyms; Figure 4.3c shows a distant hypernym relation between "person#1" and "adult#1" due to the large number of hyponyms beneath "person#1". This large distance does not necessarily mean that there is a substantial difference between "person#1" and "adult#1". It does, however, recognise that there are a substantial number of semantic features that differentiate the 203 different hyponyms of "person#1", therefore the relation is deemed to be more general than for the previous examples. This brings about rules 4.1 and 4.2 about hyponym distance.

$$\begin{aligned} &\forall x, y : sense \cdot \\ &\quad y \in direct\_hyponyms(x) \wedge \\ &\quad hyponym\_distance(x, y) = f(\#(direct\_hyponyms(x))) \end{aligned} \tag{4.1}$$

$$\begin{aligned} &\forall x, y, a, b : sense \cdot \\ &\quad y \in direct\_hyponyms(x) \wedge \\ &\quad b \in direct\_hyponyms(a) \wedge \\ &\quad \#(direct\_hyponyms(x)) < \#(direct\_hyponyms(a)) \Leftrightarrow \\ &\quad hyponym\_distance(x, y) < hyponym\_distance(a, b) \end{aligned} \tag{4.2}$$

where $direct\_hyponyms(x)$ is the set of word senses that have a hyponym path length of 1 to term x, and $f(x)$ is a function of the value x.

Normally, the hypernym distance is the relation of interest when assessing word similarity. As hyponym relations are the inverse of hypernym relations, equation 4.3 shows the definition of hypernym distance used.

$$hypernym\_distance(y, x) = hyponym\_distance(x, y) = \#(direct\_hyponyms(x)) \tag{4.3}$$

This approach adjusts the hypernym path distances to compensate for differences in the degrees of refinement of sub-hierarchies of a lexical taxonomy. Wider structures, such as for the term "adult#1", are thus penalised by being assigned longer path lengths to the hypernym relation between word senses. Longer, thinner sub-structures, such as for "horse#1", are conversely assigned short distances. Measures using this approach are less sensitive to differences in degrees of sub-hierarchy development.

4.3.2 Hypothesis 2: A Different Word Similarity Approach other than Using Edge Distances or Statistical Augmentation

All similarity measure techniques described in the previous chapter, with the exception of (Agirre and Rigau, 1995, 1996; Rigau et al., 1997), are based solely on the hypernym relations of WordNet's taxonomy. Hypothesis 2 extends the idea of co-ordinate terms to start considering the use of further relations in WordNet's taxonomy of use for improving similarity measures. Whilst the resulting similarity measures are still based in essence on WordNet's hypernym taxonomy, they differ significantly in the approach of existing techniques.

Axiom 5 states that co-ordinate terms have a special relation to each other in addition to being members of the hyponym set of a word sense. Examination of different senses in a lexical taxonomy reveals that the hypernym edge distances between some word senses are distant, but the senses may still be considered to be semantically close. It also follows that such word senses can be more similar to one another than their hypernyms would be, thus showing that path-distance may sometimes produce inaccurate

evaluations of similarity. For example, consider when the situation in 4.4 is true:

$$\begin{aligned} &\exists x, y, z : sense \cdot \\ &\quad hypernym\_path\_distance(x, y) < hypernym\_path\_distance(x, z) \wedge \\ &\quad sim(x, z) > sim(x, y) \end{aligned} \tag{4.4}$$

where $hypernym\_path\_distance(a, b)$ is the number of arcs in a hypernym tree between two word senses, a and b.

An alternative approach to using edge distance to calculate the similarity of two word senses is to assess the difference in abstraction found in their hypernym structures to a hypernym common to both word senses. This can be achieved when considering the number of hyponyms directly below each word sense. To clearly show this alternative method, consider a taxonomy where all senses other than terminal senses have exactly two hyponyms, illustrated by the binary tree structure in Figure 4.4. In

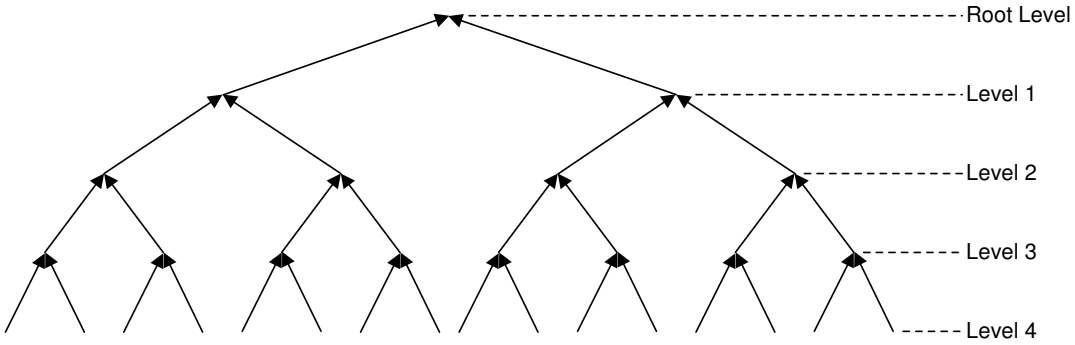


Figure 4.4: Binary Tree Example

the binary tree, each node represents a word sense, and the arcs represent hypernym relations between the senses. By definition, all word senses with a common direct hypernym are co-ordinate terms. Also of interest are all the senses that share a common hypernym at the same amount of generalisation. For example, it is easy to see that words denoting feline animals are semantically similar due to co-ordinate term relationship. However, the words “cat” and “dog” are normally considered similar to some lesser extent. Furthermore, people would associate “cat” and “canine” less strongly than they would associate “cat” and “dog”, even though the edge distance between “cat” and “canine” is smaller. This situation is analogous to “cat” and “dog” being at

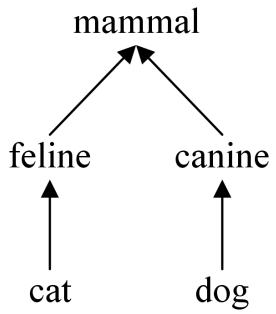


Figure 4.5: Binary Taxonomy Example for “Cat” and “Dog”

approximately the same level of generalisation from some common node (in this example this node is “mammal”) in a binary tree, as they are at the similar depths in the taxonomy, as shown in Figure 4.5. In a real world lexical taxonomy, such as WordNet, depth alone is not a reliable source of information for measuring the generalisation between two word senses on a particular hypernym path. An alternative way to calculate this difference in the amount of generalisation that avoids relying on such word senses being at the same depth is to use information about the branching of the hypernym subtrees of word senses. This branching information will be referred to as the “shape” of the hypernym structure for a word sense, where the structure of interest goes from the word sense to a hypernym that is common to some other given word sense. Given the shape information for two word senses, the ratio of generalisation between two word senses is given in equation 4.5.

$$Generalisation\ Ratio = \frac{shape(x)}{shape(y)} \tag{4.5}$$

Given that “shape” is a function of the hyponym branching along a hypernym path, two methods of calculating the “shape” of a hypernym structure are given by 4.6, or by 4.7 if a sense m is not known.

$$shape(w,m) = \begin{cases} 1 & : \quad if\ w = m \\ \#(\psi(\lambda(w))) \text{ <OP> } shape(\omega(w),m) & : \quad \text{otherwise} \end{cases} \tag{4.6}$$

$$shape(w) = shape(w, root(w)) \tag{4.7}$$

where w and m are word senses, m is a hypernym of w (normally the MIS between

w and another word sense), $\psi(x)$ is the set of hyponyms for a word sense x, $\lambda(x)$ is a hypernym of a word sense x, $<\text{OP}>$ can be either $+$ or $\times$, $\#s$ is the number of element in a set s and $root(w)$ is the root of the hypernym structure for w. The term $\#(\psi(\lambda(w)))$ can be thought of as the number of hyponyms for the hypernym of word sense w.

The two different arithmetic operators give different interpretations of the notion of "shape". Using the $+$ operator, shape will measure the pure hyponym branching along the path, that is to say the number of nodes that branch off the hypernyms of a particular structure. This is referred to as shape$_+$. The $\times$ operator gives an estimate of the total number of senses in a substructure of the hypernym structure. It is only an estimate because nodes outside the hypernym structure are unlikely to branch to a comparable degree. This is referred to as shape$_\times$. Consider the hypernym structure in figure 4.6. In

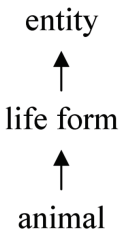


Figure 4.6: Hypernym Taxonomy for "animal#1"

order for either shape$_+$ or shape$_\times$ to be calculated, the number of direct hyponyms for each inherited hypernym of "animal#1" must be known, in this case 36 hyponyms for "life form#1" and 14 hyponyms for "entity#1". Whilst assuming a virtual root above "entity#1", $shape_+(\text{"animal\#1"}) = 36 + 14 + 1 = 51$ and $shape_\times(\text{"animal\#1"}) = 36 * 14 * 1 = 504$, where in both cases $root(\text{"animal\#1"})$ is the virtual root.

Such a definition for the ratio of generalisation becomes necessary with WordNet's taxonomy in order to create similarity measures that are less sensitive to differences in path length between a word sense and its hypernym. For instance, consider the following two noun senses:

- "mammal#1" sense 1 has 5 direct hyponyms;
- "man#1" sense 1 has 45 direct hyponyms.

As “mammal#1” has fewer immediate hyponyms than “man#1”, these hyponyms are deemed to be more closely related to “mammal#1” than the hyponyms of “man#1” are to “man#1”. In other words, the hyponyms of “mammal#1” show a smaller level of specialisation and are therefore only slightly more specific classes of “mammal#1”, whereas the hyponyms of “man#1” show a larger level of specialisation and are therefore more specific sub-classes of “man#1”. Path distances should reflect this level of specialisation, or generalisation if considering hypernym path distance, so the path distance from “mammal#1” to one of its hyponyms is closer than that between “man#1” to one of its hyponyms.

4.3.3 Hypothesis 3: Collapsing WordNets Taxonomy to Include Only Layman Terms

WordNet’s taxonomy has a large number of domain specific terms that are not used in day to day conversation, or even known by many people. Examples of such words include scientific terms used to sub-classify animal nouns. Terms such as “placental mammal” are not often considered by people when they assess the similarity of terms like “dog” and “cow”. Such terms artificially increase hypernym path distances between senses thus making them seem less similar.

Work has been previously performed to reduce the hypernym structures to include only layman terms. Tengi (1998) makes use of WordNet 1.5’s familiarity index to detect non-layman terms in hypernym structures. The work was used to reduce terms in WordNet’s taxonomy to closer match what Tengi refers to as the “mental lexicon” using psycholinguistic principles. For WordNet 1.5, the familiarity index is not based on occurrence frequencies taken from corpora, as such frequencies would be inadequate for a lexicon as large as WordNet due to the lexical bottleneck problem. Instead, an alternative method is used, based upon the correlation between occurrence frequency and polysemy (Zipf, 1945; Jastrezembski and Stanners, 1975; Jastrezembski, 1981). Every word in WordNet 1.5 has a familiarity index calculated from the polysemy of the word according the Collins online dictionary. Given this familiarity index, Tengi’s approach is then to remove all words with an index less than or equal to 1.

Figure 4.7, taken from (Tengi, 1998), shows the words of hypernym structure for “bronco#1” and their associated familiarity index according to WordNet 1.5. The effectiveness of using the familiarity index to reduce hypernym structures to layman

hypernym structures is seen clearly in this simple example.

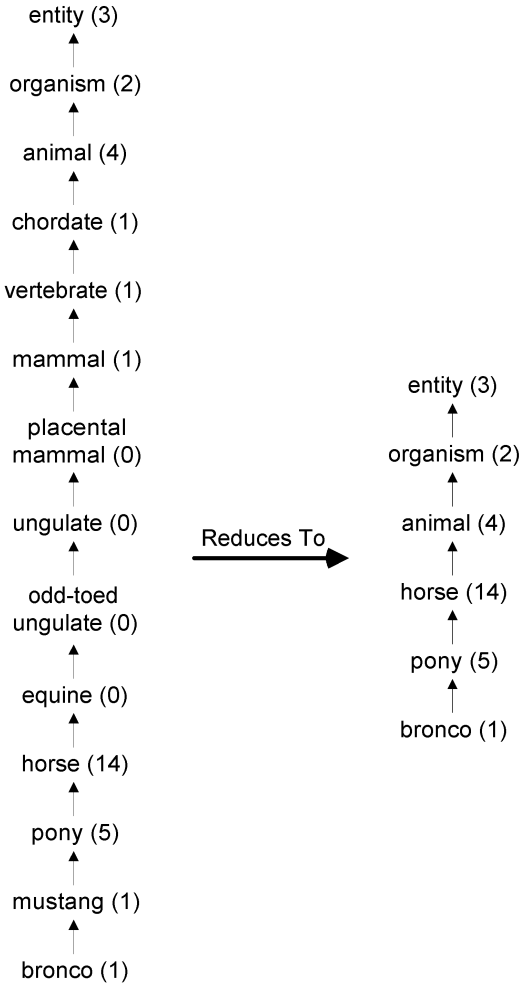


Figure 4.7: “Bronco#1” Hypernym Structure Reduction from (Tengi, 1998)

Version 1.6 of WordNet no longer calculates familiarity indexes from an alternative dictionary, but from the polysemy counts within WordNet itself. As a result, the frequency indexes between WordNet 1.5 and 1.6 differ significantly and can no longer be applied to Tengi’s technique for reducing hypernym structures. Taking the initial hypernym structure in Figure 4.7, the frequency indexes for the words bronco to entity in WordNet 1.6 using Tengi’s approach are:

$$1, 1, 5, 6, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1$$

The main difference with Tengi's example is that "animal" and "entity" are lost in the layman structure. Another issue is that the most common word in WordNet 1.6 for each sense in the hypernym structure of "bronco#1" is not the same as those given in the example. The most common word for the synset of "organism" in the example is "life form" which has a polysemy count of 1. Therefore another term would be lost if an automatic system is based upon using the most common word of a WordNet synset. This produces a problem regarding which word for a WordNet synset should be selected to calculate the polysemy count automatically.

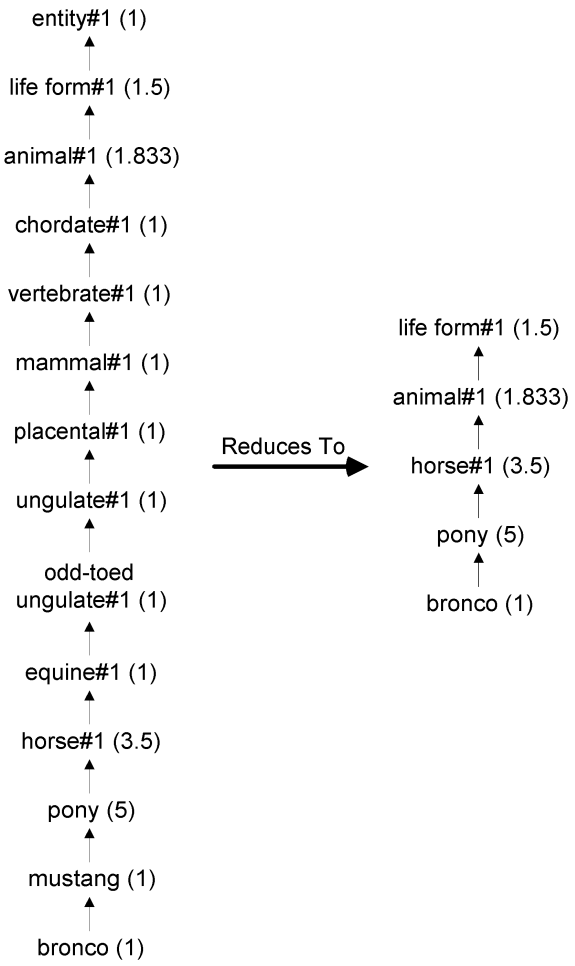


Figure 4.8: "Bronco#1" Hypernym Structure Reduction Using New Approach

A new approach to reducing WordNet 1.6 hypernym structures is presented here,

which takes into account the polysemy of all words in a given synset. Once their polysemy is established, the average polysemy for all the words in the synset can be used as an alternative familiarity index for a synset. Similar results to those produced by Tengi's proposal are possible with this alternative approach. For instance, applying this technique to the "bronco" hypernym structure gives the results shown in Figure 4.8. Now the only difference with the example using WordNet 1.5 is that "entity" is lost.

Using Layman Hypernym Structures with Similarity Measures

Reduced hypernym structures can be readily used with similarity measures that only consider path distance between two senses. As such measures are only adding a value to an ongoing distance, if a term is a layman term the distance is increased, otherwise it remains unchanged. For the similarity measures using $\text{shape}_\times$, more thought is necessary.

When multiplying the hyponym branching of senses, the shape measure estimates the number of nodes beneath a sense in some given hypernym sub-structure, therefore ignoring technical terms may lose vital information. As such, there are two possibilities to be considered:

- Ignore the hypernym branching for non-layman word senses (Layman Hypernym Structure).
- Retain the branching information of non-layman word senses whilst disregarding the nodes for the non-layman word senses. This is achieved by adding the sum of the hypernym branching of non-layman word senses to the branching of the next layman word sense. This corresponds to flattening the non-layman terms to the same level as layman terms in a hypernym structure so that their information is not lost. This is only applicable for $\text{shape}_\times$ (Flattened Hypernym Layman Hypernym Structure). If this were applied to shape_+, this would produce approximately the same results as when considering the entire hypernym structure.

4.3.4 Hypothesis 4: Handling Hypernym Trees with Multiple Paths from Sense to Root Sense

A complete definition of a similarity measure based upon information contained in a hypernym taxonomy requires a decision regarding how alternative paths are handled when calculating the final similarity. For instance, whilst considering hypernym relations, there are a number of concepts in WordNet, such as “brew#1”, that have multiple alternative hypernym paths.

The approach taken here is similar to the approaches in previous work (Rada and Bicknell, 1989; Rada et al., 1989; Lee et al., 1993; Resnik, 1995a,b, 1999). Where multiple paths are available from a synset to the root of a hypernym tree, the shortest path including the MIS between two senses in the hypernym structure is used. For instance, Figure 4.9 shows the complete hypernym structure for “brew#1”. Each synset in the structure has been additionally labelled with its depth relative to a virtual root node above “entity#1”. In practice, the depth of “brew#1” will be dependant on the word sense it will be compared to. Normally a depth of 7 will be assigned to “brew#1”. However, should the MIS to “brew#1” and another synset by either “fluid#1” or “liquid#1”, its depth becomes 8 as the hypernym path being considered contains an extra edge.

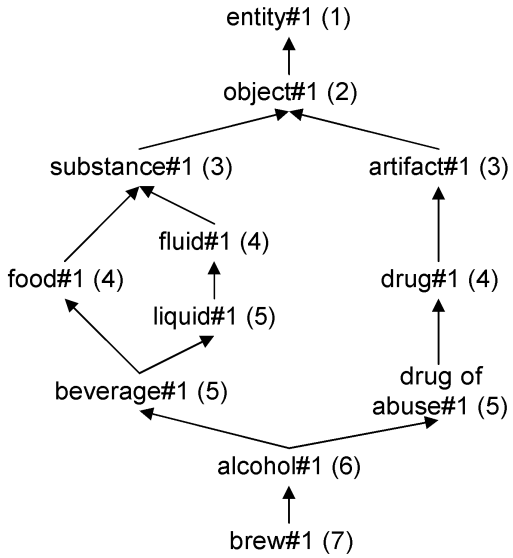


Figure 4.9: “Brew#1” Hypernyms

Such a criterion has been followed in previous work in order to maximise the similarities assigned to words when using path distances to calculate similarity. When considering layman hypernym structures, following the same criteria will produce equivalent results, however the resulting MIS will be likely to change. Such a technique for selecting a unique path may not generate the best possible similarity when using generalisation ratios to calculate similarity.

4.4 Shape-Based Similarity Measures (SBSMs)

Following the "shape" definition given in hypothesis 2, a number of new similarity measures, referred to as SBSMs, have been implemented using varying approaches to assign similarity measures to word sense pairs. Work and evaluation of SBSMs was first published by Dionisio et al. (2001).

4.4.1 Similarity Measures based on Hypernym Structure Shape

The first of the new SBSMs, shown in equation 4.8, is a simple test of the ratio of generalisation between two senses and forms the basis of further SBSMs.

$$Sim_{SBSM1}(c_1, c_2) = \begin{cases} \frac{shape(c_1)}{shape(c_2)} & : \quad if\ shape(c_1) < shape(c_2) \wedge \\ & \quad c_1 \neq MIS \wedge c_2 \neq MIS \\ \frac{shape(c_2)}{shape(c_1)} & : \quad if\ shape(c_1) > shape(c_2) \wedge \\ & \quad c_1 \neq MIS \wedge c_2 \neq MIS \\ 1 & : \quad \text{otherwise} \end{cases} \tag{4.8}$$

where c_1 and c_2 share at least one common subsumer in their hypernym structures.

Such a measure is intended to demonstrate behaviour described in axioms 1, 2 (although different depths for the MIS are not considered), 5 and hypotheses 1 and 2. In order to tackle structures with multiple paths, the shortest structure containing the MIS is selected in accordance with hypothesis 4.

This measure makes no provisions for handling information common to two senses, as stated in axiom 3. The following SBSMs adjust values from Sim_{SBSM1}, referred to simply as $SBSM_1$, with a multiplier calculated from information contained in the hypernym taxonomy above the MIS, where the multiplier determines a value for the amount of common information between two word senses.

4.4.2 Similarity Measures based on Hypernym Structure Shape Adjusted by a Common Information Multiplier

There are a number of ways in which information in a hypernym structure above the MIS of two senses can be used to calculate similarity:

1. Path distance from the MIS to the root of the taxonomy. This gives a measure of how deep in the taxonomy the MIS occurs.

2. Shape from the MIS to the root of the taxonomy. This gives an estimate of the amount of common information that is expressed above the MIS.

3. Average hyponym branching of nodes from the MIS to the root of the taxonomy. This gives an estimate of the overall abstraction the root has relative to the MIS.

These notions give a means of measuring information that is common to two word senses which can be used to adjust measures by considering common information. This produces three further SBSMs, shown in equations 4.9, 4.10 and 4.11.

$$Sim_{SBSM2}(c_1, c_2) = Sim_{SBSM1}(c_1, c_2) \times d(c_3) \quad (4.9)$$

$$Sim_{SBSM3}(c_1, c_2) = Sim_{SBSM1}(c_1, c_2) \times shape(c_3) \quad (4.10)$$

$$Sim_{SBSM4}(c_1, c_2) = Sim_{SBSM1}(c_1, c_2) \times \beta(c_3) \quad (4.11)$$

where in each case c_1 and c_2 share a common subsumer, c_3 is the MIS of c_1 and c_2, $d(c)$ is the depth of sense c's hypernym structure and $\beta(c)$ is the average hyponym branching of a word sense c.

As $SBSM_1$ will always produce values within the range $0 < Sim_{SBSM1}(c_1, c_2) \leq 1$, the common information multipliers (CIMs) in $SBSM_2$, $SBSM_3$ and $SBSM_4$ can overly influence the final result. In order to reduce the influences the measures of common information have on the final SBSM, they can be normalised to be within the range $0 \leq CIM \leq 1$. This guarantees that the CIMs will not be the overall-determining factor of similarity. In order to restrict the range of values, the CIMs are normalised as shown in equation 4.12.

$$normalise_{CIM}(CIM) = \begin{cases} 1 - \frac{1}{CIM} & : \quad if\ CIM > 1 \\ k & : \quad \text{otherwise} \end{cases} \quad (4.12)$$

where k is a constant for each SBSM. This constant is used so that a CIM of 1 or less will not produce undesirable results. For instance, if the CIM is 1, the normalised multiplier would be 0 therefore producing a similarity measure of 0 which is clearly inappropriate as some similarity has been found in the taxonomy. A small enough value for k was selected experimentally for each measure:

- $k = 0.1$ for $SBSM_2$
- $k = 0.00001$ for $SBSM_3$
- $k = 0.05$ for $SBSM_4$

The last value of k is contentious, as when the values of the average hyponym branching of a structure lie between 1 and 1/0.95, the value produced by the normalisation function will produce values smaller than 0.05. This is overlooked due to the unlikely event of such values being encountered.

Given these new common information multipliers, another three SBSMs are produced as shown by 4.13, 4.14, 4.15.

$$Sim_{SBSM5}(c_1, c_2) = Sim_{SBSM1}(c_1, c_2) \times normalise_{CIM}(d(c_3)) \quad (4.13)$$

$$Sim_{SBSM6}(c_1, c_2) = Sim_{SBSM1}(c_1, c_2) \times normalise_{CIM}(shape(c_3)) \quad (4.14)$$

$$Sim_{SBSM7}(c_1, c_2) = Sim_{SBSM1}(c_1, c_2) \times normalise_{CIM}(\beta(c_3)) \quad (4.15)$$

where in each case c_1 and c_2 share a common subsumer, c_3 is the MIS of c_1 and c_2, $d(c)$ is the depth of sense c's hypernym structure and $\beta(c)$ is the average hyponym branching of a word sense c.

4.4.3 Similarity Measures based on Hybrid Versions of Hypernym Structure Shape

The new SBSMs described thus far consider similarity as a function of the ratio of generalisation between two hypernym structures, with some added adjustment given by a function of the information common to the two structures. Previous work has mostly been based upon path distances, so it would be useful to determine if the shape function could improve results from such measures. The Wu and Palmer (1994) similarity measure can be readily adaptable to make use of these ideas, therefore two further hybrid SBSMs are also considered.

The simplest hybrid form for a SBSM would be to use the product of two measures, as shown by equation 4.16, therefore making use of the discriminating aspects of both measures. The final value from such an approach will be a compromise from the values of the individual measures used.

$$Sim_{SBSM8}(c_1, c_2) = Sim_{Wu\&Palmer}(c_1, c_2) \times Sim_{SBSM?}(c_1, c_2) \tag{4.16}$$

Only SBSMs assigning values between the range of 0 and 1 will be used so that overall neither term can overly bias results of the similarity measure.

Wu and Palmer (1994) uses an approach that measures the differences and similarity between two measures in order to calculate the final similarity value. Such a similarity measure can be easily adapted to make use of shape in order to assign different weights to different hypernym edges, as shown in equation 4.17.

$$Sim_{SBSM9}(c_1, c_2) = \frac{2shape(c_3)}{shape(c_1) + shape(c_2)} \tag{4.17}$$

where c_3 is the MIS of c_1 and c_2.

Using shape$_\times$, it is unlikely that the above function will produce well-distributed similarity values immediately. This is mainly due to the difference in magnitude for values of shape. Results can be improved by post-processing the results from the shape function, for instance considering logarithms of the shape. This is possible as each of the different SBSMs are designed to make use of WordNet's lexical taxonomy in different ways. However, fine-tuning of the distribution of similarity values across differing levels of similarity is left open to further investigation. Given the design of the SBSMs, it is natural that some measures may assign high values for word-pairs with low similarity. However, it is expected that the relative ordering of different word-pairs according to the similarity values assigned by the SBSMs will be reasonable. Post-processing values from the SBSMs presented here is considered only to a limited extent during the evaluation.

4.4.4 Calculating the Average Hyponym Branching of the Hypernym Structure

Some of the SBSMs use the average hyponym branching of a structure. This value is calculated given the shape and depth of the hypernym structure below some word sense, x. For shape$_+$, an accurate measure of the average branching is given by 4.18.

$$\beta(x) = \frac{shape_+(x)}{d(x)} \tag{4.18}$$

For shape$_\times$, an approximation of the average branching is given by 4.19.

$$\beta(x) = shape_\times(x)^{1/d(x)} \tag{4.19}$$

4.4.5 SBSM Parameters

In their current state, the SBSMs implement ideas from axioms 1, 2, 3 and 5, and from hypothesis 1, 2 and 4. In order to test the remaining axioms and hypothesis, parameters in the form of flags will be used that change some of the characteristics of the measures. Each different combination of parameter values can be thought of as producing a different similarity measure, although given that the parameters only slightly modify the behaviour of the SBSMs they need not be considered as such. Indeed, one of the main aims of this work is to determine the best combination of parameters for measuring similarity. The parameters for the SBSMs are:

- Use of layman structures (from hypothesis 3)
- Use of flattened layman structures (from hypothesis 3)
- Consider the meronym/holonym terms of the senses being tested (from axiom 4)
- Normalisation of results so that values fit into a standard scale (axiom 1)

The latter parameter follows from axiom 1 so that an upper bound value is assigned when synonyms are tested. For $SBSM_1$ and $SBSM_9$ this is not an issue, as all synonym pairs will be assigned values of 1, and $SBSM_8$ is dependant on the SBSM chosen to work with the Wu and Palmer measure. The other SBSMs, however, assign different similarity values to different synonym pairs. This is a situation that seems

undesirable because if two terms represent the same idea, surely there is no way they can be anymore similar to each other. Further, it seems unnatural to say that two pairs of identical terms differ in their magnitude of similarity. SBSMs 2 to 7 assign different values to synonym pairs depending on their depth in the taxonomy, therefore it is these measures that must be normalised so that final values are equal for all synonym pairs. Currently, this normalisation has been implemented for $SBSM_5$ to $SBSM_7$ using equation 4.21 and 4.22.

$$\nu(c_1, c_2) = \max(CIM(c_1), CIM(c_2)) \tag{4.20}$$

$$\varphi(c_1, c_2) = \begin{cases} 1 - 1/\nu(c_1, c_2) & : \quad if\ \nu(c_1, c_2) > 1 \\ c & : \quad \text{otherwise} \end{cases} \tag{4.21}$$

$$Sim_{normalisedSBSM}(c_1, c_2) = \frac{Sim_{SBSM}(c_1, c_2)}{\varphi(c_1, c_2)} \tag{4.22}$$

where c_1 and c_2 are word senses, $\varphi(c_1, c_2)$ is the normalisation factor, and $CIM(c)$ is the CIM calculation for a given SBSM applied using a word sense c. Such a normalisation technique is used with all results so that they fit within the range of 0 to 1, where 1 signifies perfect synonymy.

4.5 Evaluating Similarity Measures

The question of what constitutes an adequate evaluation method for similarity measures remains open. Previous work on objectively evaluating similarity measures has proven difficult as similarity measures differ in the task for which they are used, indeed quite often no formal evaluation is performed on the similarity measures in isolation of the task for which they are created. Where similarity measures are evaluated, the most common approach is to compare these with results from measures of human judgements on a set of word-pairs, such as the Rubenstein and Goodenough (1965) or Miller and Charles (1991) word-pairs. More recently, Finkelstein et al. (2002) made available a larger set of human judgements consisting of similarity judgements for 353 word-pairs, although this is not used to evaluate the measures introduced in this chapter as it became available too late. Some work, for instance (Lin, 1997), prove that their measures possess certain desirable qualities, such as the properties specified by the axioms and hypotheses introduced earlier in this chapter. Other work, for example (Resnik,

1995a,b, 1999; Budanitsky, 1999; Budanitsky and Hirst, 2001), develop more rigorous application-orientated tests rather than collecting the data required for comparison against human judgements. Resnik (1995a,b, 1999) tests a similarity measure against the senses of semantically related word groups collected from thesauri entries and from known collections of noun groupings in order to have an application-oriented evaluation. However, human judgements are still required for the final evaluation. The results from the similarity measures are used to disambiguate the words against each other, given the senses available in WordNet. Budanitsky (1999) and Budanitsky and Hirst (2001) set about the problem of evaluation with an alternative application-orientated approach, this time automatic detection of malapropisms in texts. Malapropisms are spelling mistakes that occur due to confusions made between words that sound similar, such as "diary" and "dairy". Such errors prove difficult to detect, as the spelling of the mistake is itself correct for another word and is also often syntactically appropriate. The evaluation involved adding artificial malapropisms into a corpus, by replacing words with variations that appear in WordNet. Budanitsky (1999) and Budanitsky and Hirst (2001) used 500 documents from the Wall Street Journal corpus, with 1408 artificially created malapropisms as in Hirst and St-Onge's original experiment (Hirst and St-Onge, 1998). The results of the similarity measures were reduced to boolean values, related or unrelated, by analysing the scatter graphs produced by the measures given the Rubenstein-Goodenough word-pairs. For example, whilst examining Figure 4.10, the chart for the Rubenstein-Goodenough human judgements, a gap is seen between the similarity values assigned to "magician-oracle" and "crane-implement". Given this information, similarity values from human judgements above 2 were deemed as being meaning that two words are similar. The similarity measures were also used to detect malapropisms by testing nouns with other nouns within a particular context window. A noun with no senses related to the senses of words in its surrounding context window became a suspected malapropism. If a spelling variation of the suspect word was found to be related to any of the words in the context window, it was diagnosed as a malapropism. The results of the evaluation were given in terms of precision, recall and f-measure.

Two tests are performed to evaluate the performance of the new SBSMs with two different tasks:

- Comparing similarity values against human judgements.

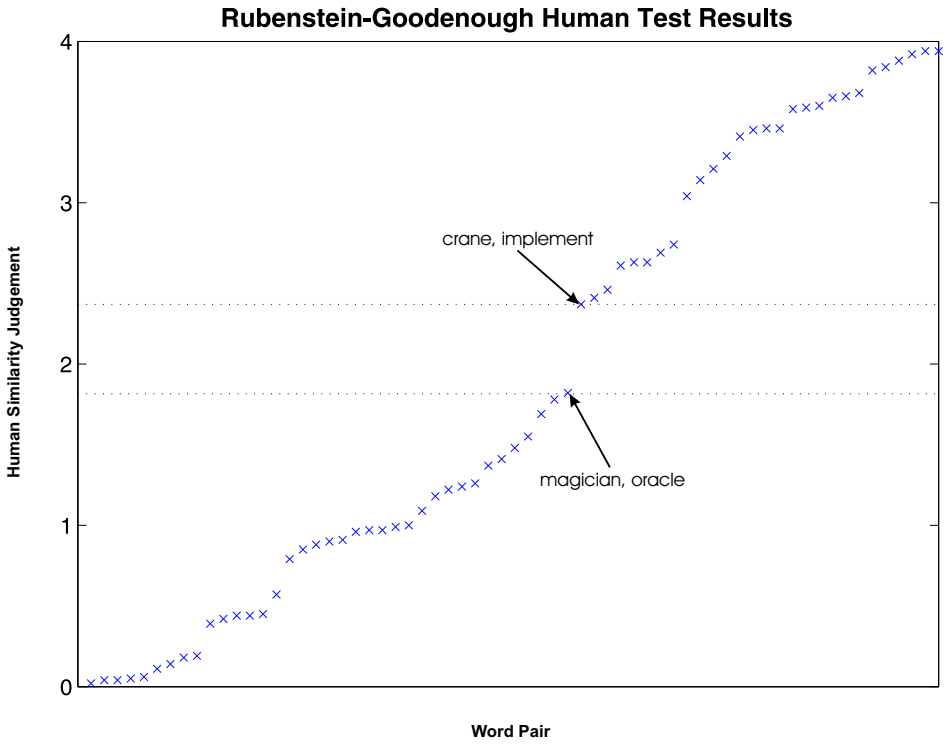


Figure 4.10: Scatter Chart of Rubenstein and Goodenough Human Similarity Judgements

The similarity measures are used to produce similarity values for the word-pairs from Rubenstein and Goodenough (1965), Miller and Charles (1991) and Resnik (1999). The results are compared against human judgements using Pearson's and Spearman's correlation techniques.

- Disambiguation words against thesaurus entries.

  Simple Word Sense Disambiguation (WSD) techniques using similarity measures are used to disambiguate thesaurus entries in order to perform a more application-orientated evaluation. Disambiguating thesaurus entries seems a natural use for such similarity measures as words in thesaurus entries are already semantically related, therefore considering only semantic information should yield highly precise results. The simple WSD techniques use semantic similarity between words to assign a sense to each word in a thesaurus entry. The sense assigned is deemed to be an adequate sense for the word as it relates to the whole

thesaurus entry. The results of the WSD systems are then compared to a gold-set of human classifications for each word.

4.5.1 Human Judgement Comparison

Commonly, researchers have assessed the accuracy of similarity measures by comparison with results from human judgements. In such evaluations, human judgements are taken as a gold standard established by human intuition about semantic similarity, although arguments can be made against such an assumption. Unfortunately there are no large data sets of human judgements publicly available for evaluating semantic similarity measures. The two most commonly used sets are the Rubenstein and Goodenough (1965) set, containing 65 word-pairs, and the Miller and Charles (1991) set, a subset of 30 word-pairs from the Rubenstein and Goodenough word-pair list. A further set of human judgements for the Miller and Charles word-pairs can be found in (Resnik, 1999), although word-pairs with "woodland" were removed due to a lack of training data for the Resnik similarity measure. As these data sets were collected using people with different social backgrounds and at different periods of time, all three sets of human judgement data are used in the evaluation presented in this chapter in order to reduce any unwanted bias that may be present. The word-pairs are listed in Appendix B together with their respective human judgements.

It is interesting to note how the method of collecting data for human judgements differs between the different approaches. Rubenstein and Goodenough used 51 undergraduates split into two groups. Each individual was given a shuffled deck of 65 slips of paper with a pair of words on each slip. They were then asked to order the pieces of paper from least similar pair to most similar pair. Once this was completed, the individuals assigned similarity values from 0 (no similarity) to 4 (perfect synonymy) to each of the word-pairs on the slips of paper. The average of the similarity value given for each word by the human test subjects was then taken to represent the human judgement of similarity for the word-pair. Such an approach forces the individuals to make definitive choices even when they may be uncertain about differences, and by initially ordering the word-pairs, some bias may have been introduced to the similarity values assigned to the word-pairs. The Miller and Charles data set was produced by carefully selecting 30 of the word-pairs from the Rubenstein-Goodenough collection that represented word-pairs with high, medium and low levels of similarity. 38 undergraduates

were then given the word-pairs and asked to give each pair a value representing the similarity of meaning, again within the range of 0 to 4. The average of the given similarities assigned was once again taken as the human judgement. Resnik approaches the experiment in a similar way to Miller and Charles using the same word-pair set with entries containing "woodland" removed. A group of 10 computer science students, both undergraduate and postgraduate, gave judgements about the similarity of the word-pairs. Half of the candidates were given the word-pairs in a random order, and the other half were given the word-pairs in descending similarity order, according to the similarity judgements giving by Miller and Charles. The two tests did not force people to order the word-pairs before assigning a similarity measure to them, and this may have influenced the results.

The evaluation of the SBSMs introduced in this chapter involves calculating word-pair similarity using each of the SBSMs with each of the various parameters where applicable. The final word-pair similarity is defined similarly to the Resnik (1995a, 1999) approach, as given by equation 4.23.

$$WordPairSim(A, B) = \max_{\substack{x \in senses(A), \\ y \in senses(B)}} Sim(x, y) \tag{4.23}$$

Using the results produced by each similarity measure, a comparison is made against each of the human judgement sets to see how well the results correlate. This comparison will be made using two different correlation coefficients, both giving measures between -1 (perfect negative correlation), 0 (no correlation) and 1 (perfect positive correlation):

- Pearson's Coefficient

  Also known as the product-moment correlation coefficient, Pearson's correlation coefficient is the most commonly used correlation coefficient. The coefficient measures the strength of the linear association between two sets of data, and not the relative ranking of the values within the datasets. As a result, some correlation tests may look misleadingly low between two related data sets if their relationship is not linear, for instance when the distribution between the values is different. Given two data sets, X and Y, with elements $x_i \in X$ and $y_i \in Y$ where $i = 1, \ldots, N$, Pearson's coefficient is estimated using 4.24.

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2}\sqrt{\sum_i (y_i - \bar{y})^2}} \tag{4.24}$$

- Spearman's Rank Coefficient

  This is an example of a non-parametric correlation coefficient measure. Such a coefficient makes no assumption about the relationship between values, such as distribution, other than the rank of the values within a data set. This is often a better coefficient to consider as it gives a clearer impression of the possible relationship between two data sets, without allowing the distribution of the values to introduce noise into the coefficient. The final function is similar to Pearson's coefficient, but uses the relative ranking between values and not the values themselves. Each $x_i \in X$ is used to calculate R_i, the rank of x_i within the data set. For situations where more than one element is allocated the same rank, a mid-rank[1] value is assigned to each instance. The same is done for each $y_i \in Y$ to calculate S_i. The resulting function is the linear correlation between the ranks, calculated using 4.25.

$$r = \frac{\sum_i (R_i - \bar{R})(S_i - \bar{S})}{\sqrt{\sum_i (R_i - \bar{R})^2}\sqrt{\sum_i (S_i - \bar{S})^2}} \tag{4.25}$$

  Some sources use an alternative form of Spearman's correlation, as shown in 4.26

$$r = 1 - \left(\frac{6\sum_i (R_i - S_i)^2}{N(N^2 - 1)} \right) \tag{4.26}$$

Given the difficulty of collecting accurate metric values in human tests, it may be more sensible to consider relative ranking of word-pairs, as the main interest is the accuracy to which similarity measures order the similarity between word-pairs. This latter task seems natural for a human to perform. However, for a human to assign a similarity value seems unnatural and forced. Pearson's coefficient however gives a reasonable estimate of how good the values of the similarity measures are, and this may be of use

[1] Mid-rank = the average of the ranks that would be assigned to a range of values that are equal in some data set.

in evaluating their usefulness in further applications.

Previous work has used correlation values between human judgements to define an upper target for the expected performance of a computerised technique. Resnik (1999) uses the average correlation over his 10 subjects against Miller and Charles results, ending with an upper target of $r = 0.88$ (only for Pearson's coefficient). For the evaluation presented here individual human judgements were not available, therefore the lowest coefficient between the different human judgement tests will be used as an upper target. Using table 4.1, a Pearson's coefficient and Spearman's coefficient of 0.9

	Goodenough' vs. Miller'	Goodenough' vs. Resnik	Miller' vs. Resnik
Pearson's Product-Moment	0.968	0.896	0.955
Spearman's Rank	0.891	0.944	0.937

Table 4.1: Inter Human Judgement Data Set Correlation

is taken as an upper target for the machine based similarity measures in this evaluation, by rounding the lowest results in table 4.1 up to one significant figure.

Pearson and Spearman correlation coefficients are calculated for each similarity measure and parameter combination tested. These results are summarised in Figures 4.11, 4.12 and 4.13 and more detailed results are presented in Appendix B. The charts show the results for each similarity measure separated by the vertical lines. Note that for $SBSM_8$, $SBSM_1$ is used with the Wu and Palmer measure for the evaluation. For each measure, six results are given:

1. Basic use of the similarity measure, denoted using "$*$".
2. Basic use of the similarity measure, but with normalised results, denoted using "●".
3. Layman structures used, denoted using "■".
4. Layman structures used with normalised results, denoted using "★".
5. Flattened layman structures used, denoted using "⧫".
6. Flattened layman structures used with normalised results, denoted using "✸".

In order to save space, results considering meronyms terms have not been given, as they do not affect the result for any of the word-pairs, as no word-pair from the human judgement data sets is associated via inherited meronymy within WordNet. Each correlation coefficient is statistically significant with $p < 0.01$. These charts show the effect of using shape$_\times$ (the first 9 SBSM results), and of using shape$_+$ (the last 9 SBSM results). In order to evaluate the effect of normalising common information multipliers ($SBSM_5$ to $SBSM_7$), the normalised results have been arranged next to their non-normalised counterparts. The charts also show two results for each similarity measure and parameter combination; Pearson's coefficients are shown in blue and Spearman's coefficients are shown in red.

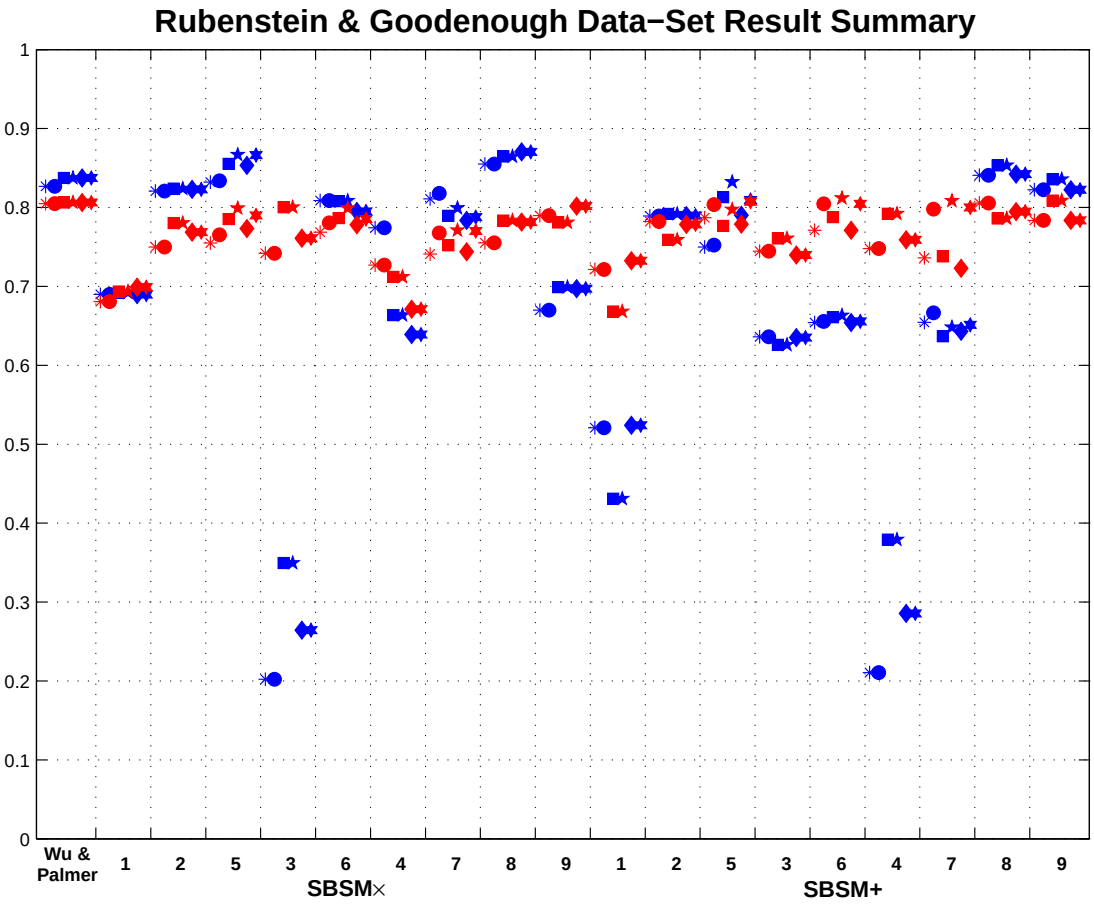


Figure 4.11: Pearson's and Spearman's Coefficients for Rubenstein and Goodenough (1965) Word-Pairs

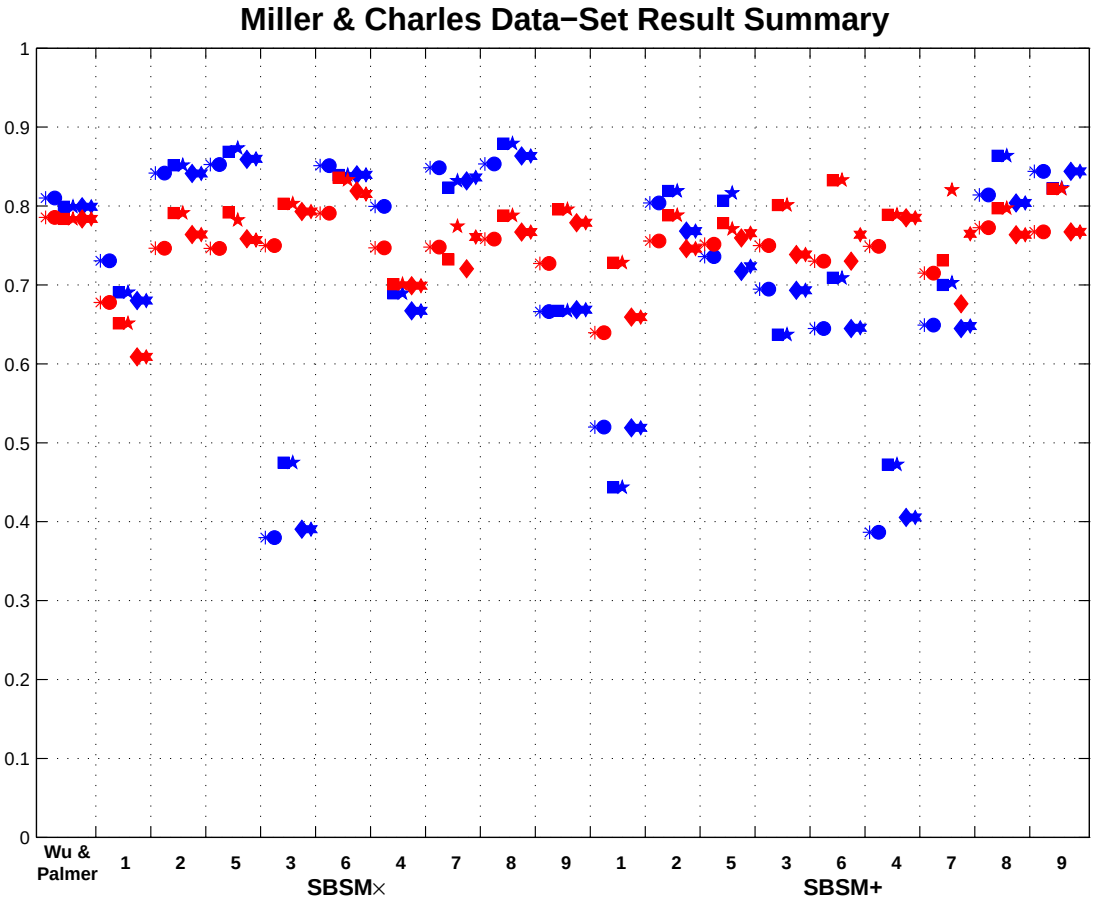


Figure 4.12: Pearson's and Spearman's Coefficients for Miller and Charles (1991) Word-Pairs

Overall Measure Performance

Tables 4.2, 4.3 and 4.4 give the ranks in ascending order, from 1 to 16, of the best result for each measure, without considering any parameters at this stage. From these tables it is possible to see which measures give the best results when compared to each other. The tables show that linear correlation techniques, such as Pearson's coefficient, may not present the most reliable evaluation for objectively comparing the performance of similarity measures. Pearson's correlation coefficient leads to discounting measures that produce improved ordering of similarity between word-pairs. The results show that shape$_\times$ produces better values for SBSMs based on the ratio of generalisation be-

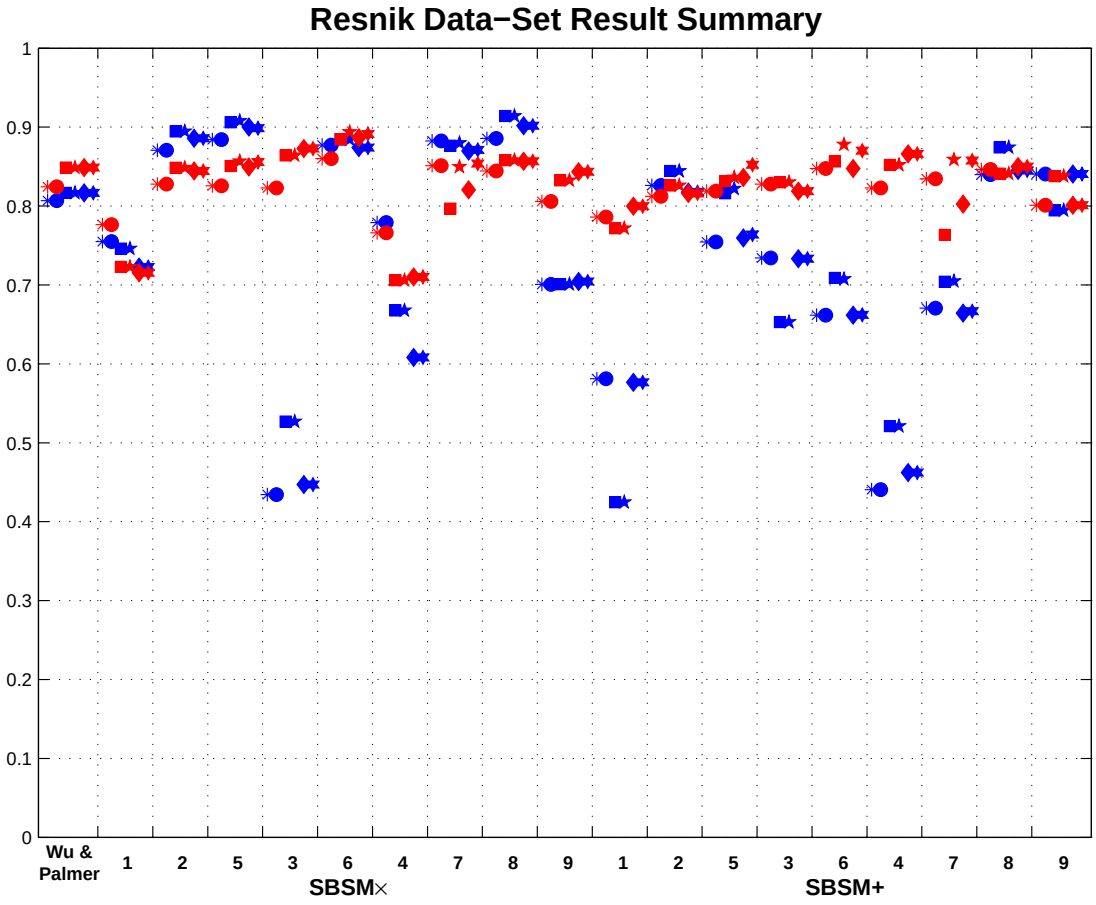


Figure 4.13: Pearson's and Spearman's Coefficients for Resnik (1999) Word-Pairs

tween two senses than using shape$_+$ to a statistical significance of $p < 0.01$, measured using Wilcoxon's matched-pairs signed-ranks test. However, the SBSM+s produce interesting results when only considering the relative ordering of word-pairs. For instance, inherent in the design of $SBSM_3$ is the fact that the shape measure of information common to two word senses has a large influence in the final measure, and that the distribution of the similarity values is likely to be quite dramatic. It is interesting to see that this type of similarity measure consistently produces a good ordering of word-pairs.

The effect of normalising the multipliers of $SBSM_2$, $SBSM_3$ and $SBSM_4$ sees an improvement in most cases. This implies that measures $SBSM_2$, $SBSM_3$ and

Rubenstein & Goodenough	Pearson's Correlation		Spearman's Correlation	
	Rank	Coefficient	Rank	Coefficient
$SBSM_{\times 1}$	12	0.69	18	0.70
$SBSM_{\times 2}$	6	0.82	13	0.78
$SBSM_{\times 3}$	17	0.35	7	0.80
$SBSM_{\times 4}$	10	0.77	17	0.73
$SBSM_{\times 5}$	2	0.87	8	0.80
$SBSM_{\times 6}$	8	0.81	9	0.80
$SBSM_{\times 7}$	7	0.82	14	0.77
$SBSM_{\times 8}$	1	0.87	11	0.78
$SBSM_{\times 9}$	11	0.70	6	0.80
$SBSM_{+1}$	16	0.52	16	0.73
$SBSM_{+2}$	9	0.80	12	0.78
$SBSM_{+3}$	15	0.64	15	0.76
$SBSM_{+4}$	18	0.38	10	0.79
$SBSM_{+5}$	5	0.83	4	0.81
$SBSM_{+6}$	14	0.66	1	0.81
$SBSM_{+7}$	13	0.67	3	0.81
$SBSM_{+8}$	3	0.85	5	0.81
$SBSM_{+9}$	4	0.84	2	0.81

Table 4.2: SBSM Summary of Evaluation using Rubenstein and Goodenough (1965) Data Set

Miller & Charles	Pearson's Correlation		Spearman's Correlation	
	Rank	Coefficient	Rank	Coefficient
$SBSM_{\times 1}$	11	0.73	18	0.68
$SBSM_{\times 2}$	5	0.85	10	0.79
$SBSM_{\times 3}$	17	0.47	5	0.80
$SBSM_{\times 4}$	10	0.80	16	0.75
$SBSM_{\times 5}$	2	0.87	9	0.79
$SBSM_{\times 6}$	6	0.85	1	0.84
$SBSM_{\times 7}$	4	0.85	15	0.77
$SBSM_{\times 8}$	1	0.88	13	0.79
$SBSM_{\times 9}$	15	0.67	8	0.80
$SBSM_{+1}$	16	0.52	17	0.73
$SBSM_{+2}$	8	0.82	12	0.79
$SBSM_{+3}$	14	0.70	6	0.80
$SBSM_{+4}$	18	0.47	11	0.79
$SBSM_{+5}$	9	0.82	14	0.78
$SBSM_{+6}$	12	0.71	2	0.83
$SBSM_{+7}$	13	0.70	4	0.82
$SBSM_{+8}$	3	0.86	7	0.80
$SBSM_{+9}$	7	0.84	3	0.82

Table 4.3: SBSM Summary of Evaluation using Miller and Charles (1991) Data Set

Resnik	Pearson's Correlation		Spearman's Correlation	
	Rank	Coefficient	Rank	Coefficient
$SBSM_{\times 1}$	11	0.75	17	0.78
$SBSM_{\times 2}$	3	0.90	11	0.85
$SBSM_{\times 3}$	17	0.53	3	0.87
$SBSM_{\times 4}$	10	0.78	18	0.77
$SBSM_{\times 5}$	2	0.91	8	0.86
$SBSM_{\times 6}$	4	0.88	1	0.89
$SBSM_{\times 7}$	5	0.88	5	0.86
$SBSM_{\times 8}$	1	0.91	7	0.86
$SBSM_{\times 9}$	15	0.70	12	0.84
$SBSM_{+1}$	16	0.58	16	0.80
$SBSM_{+2}$	7	0.84	15	0.83
$SBSM_{+3}$	12	0.73	14	0.83
$SBSM_{+4}$	18	0.52	4	0.87
$SBSM_{+5}$	9	0.82	9	0.85
$SBSM_{+6}$	13	0.71	2	0.88
$SBSM_{+7}$	14	0.70	6	0.86
$SBSM_{+8}$	6	0.87	10	0.85
$SBSM_{+9}$	8	0.84	13	0.84

Table 4.4: SBSM Summary of Evaluation using Resnik (1999) Data Set

$SBSM_4$ need no longer be considered as they are consistently improved by $SBSM_5$, $SBSM_6$ and $SBSM_7$.

Overall SBSM Performance Summary

In summary, SBSMs based on shape$\times$, SBSM$\times$s, generally produce better values than SBSMs based on shape$+$, SBSM$+$s. Also, all SBSMs that adjust $SBSM_1$ with information about the semantics common between two senses improve the ranking order of the word-pairs.

It seems that measures $SBSM_1$, $SBSM_{\times 7}$, and $SBSM_9$ can be disregarded due to their poor performance. One may also choose to ignore measures $SBSM_2$, $SBSM_3$ and $SBSM_4$ as their results are regularly improved by normalisation of their multipliers. This leaves measures $SBSM_5$, $SBSM_6$, $SBSM_{+7}$ and $SBSM_8$ for further consideration.

Similarity Measure Parameter Evaluation

The effects that the parameters have on each measure are now evaluated. Only measures that thus far are deemed to produce reasonable results are considered here. Table 4.5 shows the relative effect that each parameter has on similarity measure[2]. The results were created by counting the number of instances where a particular parameter improved results from the SBSMs. If only 50% of the instances were improved, this means overall the parameter did not improve results. For this reason, two results are shown in the table; Firstly, the number of improved instances is shown, followed by the overall improvement shown by using the parameter, calculated using equation 4.27.

$$\text{improvement} = \frac{n - (m/2)}{(m/2)} \times 100 \tag{4.27}$$

where n is the number of improved instances, and m is the number of results considered.

	Pearson's Correlation	Spearman's Correlation
Results improved by normalisation of the final values	35 of 45 56% Improvement	40 of 45 78% Improvement
Results improved by considering layman structures instead of the full WordNet 1.6 structure	20 of 24 67% Improvement	19 of 24 58% Improvement
Results improved by considering flattened layman structures instead of the full WordNet 1.6 structure	12 of 24 No Significant Improvement	14 of 24 17% Improvement
Results improved by considering flattened layman structures instead of non-flattened layman structures	3 of 21 71% Decline	6 of 21 43% Decline

Table 4.5: SBSM Parameter Evaluation Summary

The results in the top three rows of table 4.5 show that all the parameters tested

[2]Only where parameters are applicable.

have positive effects over the basic similarity measures. Normalisation of the final values has a consistent positive effect with most of the measures and in combination with other parameters. Layman structures also improve results, especially when using non-flattened layman structures. The last row of table 4.5 suggests that flattened layman structures have no significant advantage over non-flattened layman structures, however in some cases the correlation difference between the two different techniques for the same similarity measure is very close, suggesting that there is little significant advantage for either technique in such cases. Further analysis is needed to determine if flattened layman structures have any advantages.

Analysis of Scatter Graphs

Appendix B presents the scatter graphs produced by each of the selected SBSMs. These scatter graphs indicate how well these measures perform compared to each other, and whether they produce the desired characteristics. They also give information about which word-pairs are constantly assigned poor similarity so they may be investigated further.

The spread of values for $SBSM_5$, $SBSM_{\times 6}$ and $SBSM_8$ are quite tight and show a reasonably linear association to the human judgements. However, the spread of values for $SBSM_{+6}$ and $SBSM_{+7}$ is more sparse. The result of this sparseness in the scatter graphs can, in part, account for the lower Pearson's correlation. By using a function of the values produced from the similarity measures, the distribution of the values in the scatter graphs may be improved. Figure 4.14 shows the results of raising the results of $SBSM_{+7}$, using layman structures and normalised results, to the power of 15. The values of $SBSM_{+7}$ are raised by a power as the original distribution shows a logarithmic association to the human judgement values. The power of 15 was chosen by considering the resulting line of best fit, produced using linear regression for the values, to more closely match the line of best fit for the scatter graph produced by the human values against word-pair. A line of best fit has been added to the figure to show the resulting trend for the data.

The result of raising the results from $SBSM_{+7}$ by a power of 15 improves the Pearson correlation of the results to 0.85. However, the rank correlation is not changed as the relative rank order of the similarity values remains constant, as intended. This raises the question of the suitability of using Pearson's correlation results to evaluate

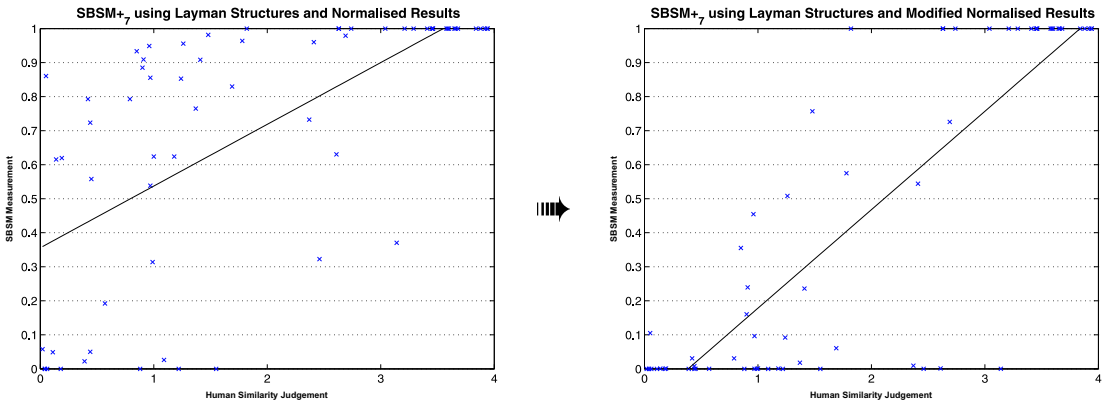


Figure 4.14: Result of raising similarity values from $SBSM_{+7}$

the overall performance of similarity measures, especially given the abstract nature of the task for a human to assign a similarity value to a word-pair. For a human to order word-pairs according to similarity seems natural. However, the subjective nature of assigning similarity values may make these values less suitable for objective evaluation of similarity measures. For instance, what does a similarity value of 1.5 in a scale of 0 to 4 mean?

The SBSMs presented in this chapter were created by considering different ways of using information within a lexical taxonomy to evaluate similarity between nouns. However, more consideration can be made about the values produced by the SBSMs. Such a technique of adjusting values to improve Pearson's correlation with human judgements can be used to fine tune the values produced by the SBSMs; however this is unnecessary as this does not necessarily change the performance on the specific tasks for which these measures may be used. Indeed, the need to adjust the similarity values is dependant on the task for which the measures will be used. For instance, the WSD system presented in chapter 6 uses a similarity measure to detect if two nouns are similar to each other, and therefore only requires a boolean result calculated by detecting if the similarity between two nouns is above a predefined threshold. Whilst the threshold is dependant on the values produced by the measure, changing the initial distribution of the values will not change the performance of the measures in this case, as long as the threshold used is also changed accordingly.

In general, where the SBSMs consistently give poor similarity values for a word-pair, it can be seen that the values assigned are generally pessimistic (i.e. low). Specific

word-pairs that consistently show poor results can be assessed to explain the poor values. The information in WordNet's lexical taxonomy used by the SBSMs to assess the similarity for each of the word-pairs has been analysed to determine if sufficient information is available to adequately calculate similarity, or if poor results can be explained by considering potential missing information. The following list presents the word-pairs that consistently show poor values for all selected SBSMs, and the most likely reasons that these poor value assignments arose according to WordNet's taxonomy. If reasons are left empty, no obvious answer was found solely considering WordNet's taxonomy:

Forest, Graveyard There is no information that "forest" is a place or location in a similar way to "graveyard", therefore the MIS between the two words is "object".

Food, Rooster There is no information in the hypernym taxonomy of rooster that rooster is a kind of food. However, the meronym structure for rooster makes reference to rooster being part "chicken meat", which in turn has food in its hypernym structure. Therefore to make use of this relation the measures would also need to make use of the hypernym structures of all meronyms of a word sense.

Cemetery, Woodland This situation is identical to forest and woodland, where cemetery is a synonym of graveyard, and woodland is a synonym with one of the two senses of forest according to WordNet 1.6.

Shore, Voyage An association between shore and voyage would require semantic relations other than hypernymy and meronymy.

Furnace, Implement

Car, Journey The similarity between "car" and "journey" comes for information about how both concepts relate to each other in the real word, i.e. "journey#1" requires "transport#1" and "car#1" is a "transport#1" via its hypernym structure therefore the two are related. However, this information is not considered for semantic similarity.

Cemetery, Mound Again, the relations that make an association between the word-pair possible are not available, therefore the similarity assigned is low.

Sage, Wizard

Oracle, Sage

Furnace, Stove The description in WordNet for furnace states that a furnace is a heating device; however this information is not reflected in furnace's hypernym structure.

Comparison with other Similarity Measures

A number of existing similarity measures have been tested with the word-pairs used for this evaluation. Budanitsky (1999) gives results for 5 similarity measures using the Rubenstein and Goodenough (1965) and the Miller and Charles (1991) word-pairs:

- St-Onge (1995); Hirst and St-Onge (1998)
- Jiang and Conrath (1997)
- Leacock and Chodorow (1998)
- Lin (1998a, 1997, 1998b,c)
- Resnik (1995a,b, 1999)

Using the results presented by Budanitsky, and results calculated for the Wu and Palmer (1994) similarity measure, Pearson's and Spearman's correlation coefficients are calculated for each of the algorithms and compared to the results of $SBSM_{\times 5}$, $SBSM_{\times 6}$ and $SBSM_{\times 8}$ with layman structures and normalised results. The Jiang-Conrath measure produces negative correlation as it measures semantic distance as oppose to similarity. The results are shown in Table 4.6. From the current results, the chosen SBSMs correlate more closely with human results using both Pearson and Spearman correlation techniques.

Human Judgement Comparison Conclusions

Whist the product-moment coefficient for $SBSM_{\times 5}$ and $SBSM_{\times 8}$ comes close to the upper target of 0.9, there is still room for improvement for the similarity values produced by $SBSM_{\times 6}$. In general, values from the SBSMs are reasonably good. However, it seems that the order in which the SBSMs rank the word-pairs contains some

Similarity Measure	Rubenstein & Goodenough		Miller & Charles		Resnik	
	Pearson	Spearman	Pearson	Spearman	Pearson	Spearman
$Wu\ \&\ Palmer$	0.827	0.805	0.810	0.786	0.806	0.824
$Hirst\ \&\ St{-}Onge$	0.786	0.767	0.744	0.735	0.775	0.793
$Jiang\ \&\ Conrath$	-0.781	-0.712	-0.850	-0.813	-0.861	-0.840
$Leacock\ \&\ Chodorow$	0.834	0.783	0.816	0.766	0.829	0.818
Lin	0.819	0.777	0.829	0.782	0.860	0.840
$Resnik$	0.778	0.753	0.774	0.749	0.803	0.806
$SBSM_{\times 5}$	0.867	0.799	0.873	0.782	0.908	0.857
$SBSM_{\times 6}$	0.808	0.799	0.839	0.833	0.884	0.894
$SBSM_{\times 8}$	0.864	0.783	0.879	0.788	0.914	0.858

Table 4.6: Comparison Between Existing Similarity Measures and the Best SBSMs

errors. The SBSMs show a significant improvement over other existing similarity measures. Results for all of the human judgement comparison, including all scatter graphs, are contained in Appendix B.

4.5.2 Disambiguation Words Against Thesaurus Entries

The second evaluation of the SBSMs is performed using a system to disambiguate the nouns contained in thesaurus entries. This provides a more application-oriented approach to evaluating similarity measures. The application of thesaurus entry labelling was chosen as words in thesaurus entries are already semantically grouped by idea (Rubenstein and Goodenough, 1965) and therefore provided a natural platform to test similarity measures. A number of simple WSD algorithms are tested using the SBSMs to disambiguate words contained in the entries of the Wordsmyth thesaurus. The best combinations of WSD algorithm and SBSM are then compared against results from Resnik's WSD approach (Resnik, 1995a,b, 1999) on the Wordsmyth thesaurus. The evaluation is split into the following sections:

- Developing adequate WSD algorithms.

- Producing test data.
- Evaluation of the algorithms together with existing similarity measures and the SBSMs.
- Comparing best results against the results provided by Wordsmyth.

WSD Algorithms for Labelling Thesaurus Entries

For the task of disambiguating the senses of nouns in thesaurus entries, an algorithm is required which accepts a bag of nouns as input, and returns a list of senses per word and the likelihood that each sense is correct for the given group of nouns. Each of the WSD algorithms presented below only makes use of semantic information to disambiguate noun groupings. Their basis is that senses of a word similar to senses of other words in the word group are likely to be the best candidates for most closely relating it to the word group as a whole. Therefore the algorithms use the results of similarity measures to increase support for senses of the nouns in the noun groups. The last two algorithms attempt to improve results further by selectively increasing support only for certain senses.

Two baseline algorithms are considered for the thesaurus labelling tests, selecting the first sense of words and the Resnik algorithm for disambiguating noun groupings (Resnik, 1995a,b, 1999). Three new WSD algorithms are also considered:

- A Greedy WSD algorithm is produced by calculating the sum of the similarity for each word sense in the noun-group against all other word senses in the noun group. This sum is then normalised using the sum of the similarity of each word sense of a word, against all other word senses in the noun group. The algorithm selects the sense for each word with the highest resulting value as the correct sense for the word according to the noun group.
- The Exclusive Greedy algorithm is similar to the basic Greedy algorithm. However, for all word senses only similarity values greater than a predetermined proportion of the highest similarity value assigned per word sense are considered. The changes are made to avoid increasing support for the sense of a word when the similarity detected between pairs is low in comparison to the highest similarity detected for another of the word's senses.

- Again, the 'Related Senses Only' algorithm is similar to the Greedy algorithm. However, for all word senses only similarity values greater than a specified threshold are considered. The threshold is selected such that any word-pair with a similarity above the threshold will be classed as related, and anything below the threshold is considered sufficiently different not to be semantically related. Therefore the algorithm only increases support when two word sense pairs are significantly similar.

  The thresholds for each similarity measure are calculated using a genetic algorithm, trained with manually tagged Wordsmyth entries for the randomly selected, reasonably polysemous words *Car*, *Cat*, *Drink*, *Key*, *Line*, *Man* and *Report*, that maximises accuracy, where accuracy is calculated using 4.28.

$$Accuracy = (C + D)/N \tag{4.28}$$

  where C is the number of correct word sense classifications, D is the number of words correctly left unclassified and N is the number of words evaluated.

Appendix C gives the pseudo code for each of these new WSD algorithms. The following section describes how the noun groupings were collected from the Wordsmyth entries.

Test Data

Test data is collected from the Wordsmyth thesaurus. Noun entries were selected from a randomly generated set of 214 nouns. From these 214 noun entries, all isolated nouns are extracted to form the noun group for the thesaurus entry. Only entries containing more than one noun are used for the test, reducing the number of main words to 62, producing a test set of 1365 nouns in 186 thesaurus entries. The extracted nouns in the noun groups were manually sense labelled with all applicable WordNet senses for the thesaurus entries to which they belong. When no applicable sense exists in WordNet, no senses were given but the word is still considered during the evaluation. A copy of the human classification is presented in Appendix D.

Finally, the results from the algorithms presented here are compared to results of Resnik's algorithm (Resnik, 1999) for disambiguating noun groupings. This WSD algorithm makes use of Resnik's information content based similarity measure (Resnik,

1995a,b, 1999), created by bootstrapping statistical information to WordNet's lexical taxonomy. This evaluation uses the Wordsmyth thesaurus containing experimental links to WordNet calculated using the Resnik WSD algorithm (data supplied by Dr. Robert Parks via personal communication). The data used was automatically generated and has thus far not been formally evaluated against human judgements. Also, not all nouns for each thesaurus entry used are labelled with WordNet senses, although given the description of Resnik's technique it is assumed that all nouns in the thesaurus entry were used as input for the WSD algorithm. This is likely given the number of examples where only one noun is labelled, due to the fact that the WSD algorithm he presents requires at least one pair of nouns.

Evaluation of Noun Group WSD Techniques

The evaluation of the SBSMs is performed by testing the best SBSMs with each of the WSD algorithms discussed earlier. Each test produces a number of statistics evaluating different aspects of the system:

- Accuracy

  The overall accuracy of a system evaluates the percentage of correct sense classifications and correctly unlabelled words over all words in the test. The equation is given in 4.29.

  $$Accuracy = (C + D)/N \tag{4.29}$$

  where C is the number of correct word sense classifications, D is the number of words correctly left unclassified and N is the number of words evaluated.

- Precision

  Precision evaluates the percentage of correct decisions made by a classifier over all classifications made by the classifier. The equation is given in 4.30.

  $$Precision = C/Z \tag{4.30}$$

  where C is the number of correct word sense classifications and Z is the number of words classified.

- Recall

Recall evaluates the ratio of test words with adequate senses in WordNet correctly disambiguated by a classifier, as given in 4.31.

$$Recall = C/M \tag{4.31}$$

where C is the number of correct word sense classifications and M is the number of words with at least one adequate sense according to WordNet.

- No Sense Accuracy

  The number of words with no sense correctly left unlabelled by the classifier is evaluated to see how well systems can detect when no adequate sense exists in WordNet. This is of interest because using information about all word senses for a word group may introduce relationships between words not normally considered by humans for particular thesaurus entries. The equation is given in 4.32.

$$NoSenseAccuracy = D/(N - M) \tag{4.32}$$

  where D is the number of words correctly left unclassified, N is the number of words evaluated and M is the number of words with at least one adequate sense according to WordNet.

- Average number of senses considered

  Lastly the average number of senses considered by a system is of interest. Senses considered by a system are defined as the senses given support greater than zero. Note that only the best scoring sense is selected. The equation is given in 4.33.

$$Ave.SensesConsidered = S/W \tag{4.33}$$

  where S is the total number of word senses considered by the classifier and W is the number of words classified by the classifier.

The tests are performed using SBSMs and the Wu and Palmer similarity measure with the following parameters:

- shape$_\times$
- Non-flattened Layman Taxonomies

- Normalised Measures
- Meronyms are not considered

Tables 4.7, 4.8, 4.9, 4.10 and 4.11 present the results of the evaluation statistics grouped by WSD algorithm. Table 4.7 and results using the Wu and Palmer similarity measure are used as baselines.

Accuracy	Precision	Recall	No Sense	Ave N^o Senses
57.87%	57.87%	64.91%	0%	1

Table 4.7: Results for Wordsmyth Thesaurus Labelling Evaluation for Selecting the First Sense for each Word

Measure	Accuracy	Precision	Recall	No Sense	Ave N^o Senses
Wu $\& Palmer$	83.41%	82.88%	90.11%	28.31%	2.32
$SBSM_{\times 1}$	83.21%	82.68%	89.89%	28.31%	3.37
$SBSM_{\times 2}$	83.47%	82.95%	90.18%	28.31%	2.55
$SBSM_{\times 3}$	81.97%	81.40%	88.50%	28.31%	2.38
$SBSM_{\times 4}$	82.95%	82.41%	89.6%	28.31%	2.52
$SBSM_{\times 5}$	84.32%	83.83%	91.14%	28.31%	2.52
$SBSM_{\times 6}$	83.41%	82.88%	90.11%	28.31%	2.47
$SBSM_{\times 7}$	83.47%	82.95%	90.18%	28.31%	2.50
$SBSM_{\times 8}$	84.85%	84.37%	91.72%	28.31%	2.55
$SBSM_{\times 9}$	80.60%	79.99%	86.96%	28.31%	2.93

Table 4.8: Results for Wordsmyth Thesaurus Labelling Evaluation using the Resnik WSD Algorithm

Measure	Accuracy	Precision	Recall	No Sense	Ave N^o Senses
Wu $\& Palmer$	81.06%	80.76%	87.03%	31.93%	3.36
$SBSM_{\times 1}$	75.18%	74.39%	80.88%	28.31%	3.70
$SBSM_{\times 2}$	84.00%	83.82%	90.33%	31.93%	3.36
$SBSM_{\times 3}$	82.30%	82.05%	88.42%	31.93%	3.36
$SBSM_{\times 4}$	81.97%	81.71%	88.06%	31.93%	3.36
$SBSM_{\times 5}$	84.39%	84.23%	90.77%	31.93%	3.36
$SBSM_{\times 6}$	82.50%	82.26%	88.64%	31.93%	3.36
$SBSM_{\times 7}$	82.04%	81.78%	88.13%	31.93%	3.36
$SBSM_{\times 8}$	84.65%	84.5%	91.06%	31.93%	3.36
$SBSM_{\times 9}$	80.60%	79.99%	86.96%	28.31%	3.70

Table 4.9: Results for Wordsmyth Thesaurus Labelling Evaluation using the Greedy WSD Algorithm

Measure	Accuracy	Precision	Recall	No Sense	Ave N^o Senses
Wu $\& Palmer$	84.91%	84.93%	90.40%	39.76%	1.76
$SBSM_{\times 1}$	78.58%	77.84%	84.40%	30.72%	2.87
$SBSM_{\times 2}$	80.08%	80.11%	84.98%	39.76%	1.35
$SBSM_{\times 3}$	82.69%	82.56%	87.77%	40.96%	1.43
$SBSM_{\times 4}$	81.91%	81.98%	86.96%	40.36%	1.55
$SBSM_{\times 5}$	85.76%	85.86%	91.21%	40.96%	1.62
$SBSM_{\times 6}$	84.39%	84.60%	89.74%	40.36%	1.92
$SBSM_{\times 7}$	83.41%	83.49%	88.57%	40.96%	1.87
$SBSM_{\times 8}$	85.11%	85.17%	90.48%	40.96%	1.57
$SBSM_{\times 9}$	78.58%	77.95%	84.69%	28.31%	1.30

Table 4.10: Results for Wordsmyth Thesaurus Labelling Evaluation using the Exclusive Greedy WSD Algorithm

Measure	Accuracy	Precision	Recall	No Sense	Ave N^o Senses
Wu & $Palmer$	83.08%	86.76%	85.42%	63.86%	1.70
$SBSM_{\times 1}$	82.36%	87.18%	83.66%	71.69%	1.58
$SBSM_{\times 2}$	84.65%	87.31%	87.69%	59.64%	1.90
$SBSM_{\times 3}$	11.69%	72.22%	0.95%	100.00%	0.01
$SBSM_{\times 4}$	79.75%	85.00%	81.39%	66.27%	1.56
$SBSM_{\times 5}$	85.76%	87.63%	89.30%	56.63%	2.01
$SBSM_{\times 6}$	82.95%	86.39%	86.01%	57.83%	1.81
$SBSM_{\times 7}$	81.65%	85.90%	83.88%	63.25%	1.70
$SBSM_{\times 8}$	84.78%	85.30%	90.11%	40.96%	2.71
$SBSM_{\times 9}$	80.86%	81.68%	84.91%	47.59%	2.16

Table 4.11: Results for Wordsmyth Thesaurus Labelling Evaluation using the Related Senses Only WSD Algorithm

Interpretation of Results

The results for the tests show much less variation in quality than the human judgement tests presented previously, indicating that in general results are not greatly affected by the similarity measure used. The only exception to this rule is for $SBSM_{\times 3}$ with the WSD algorithm that only considers related senses. $SBSM_{\times 8}$, a hybrid measure making use of $SBSM_{\times 1}$, marginally produces the best results using the Resnik and Greedy WSD algorithms. Overall, $SBSM_{\times 5}$ consistently produces the best results with the two selective WSD algorithms.

Selecting the best WSD algorithm from the tests is not as straightforward as selecting the best similarity measure as a tie exists between the Exclusive Greedy algorithm and the Related Senses algorithm when using $SBSM_{\times 5}$. Sorting the results of all algorithms and measures by precision shows that the Related Senses algorithm regularly produces more precise results, and as such more confidence can be placed on the results of this algorithm. The Related Senses WSD algorithm also considers less senses per word on average, therefore the Related Senses WSD algorithm will be used to compare results against other systems.

Comparison with Wordsmyth Test WordNet Links

The results from the previous evaluation are compared to the accuracy of Wordsmyth's experimental links to WordNet (provided by Dr. Robert Parks via personal communication). These experimental links were created using results from Resnik's Information Content based similarity measure and WSD algorithm for noun groups (Resnik, 1995a,b, 1999). In order to compare the results, the experimental links in the same Wordsmyth thesaurus entries used in the previous evaluation were extracted and compared against the human classifications. Typically, only the links calculated for nouns in the 'SYN' section of a thesaurus entry are given in the data provided, although it is assumed that the inputs to the WSD algorithm follow a similar approach to that used during the evaluation presented in the previous section. As only a small number of links are given per thesaurus entry, precision and recall results are recalculated for the results obtained from the related senses algorithm with $SBSM_{\times 5}$ considering the same nouns. The results of both WSD algorithms are given in table 4.12. The poor

	Precision	Recall
Wordsmyth Test Links	80.44%	71.26%
Related Senses with $SBSM_{\times 5}$	88.28%	90.94%

Table 4.12: WSD Comparison with Wordsmyth Experimental Links to WordNet

recall values for the Wordsmyth experimental links can be possibly explained by a lack of training data for the information based similarity measure, although no evidence is available for this. Comparing the two approaches shows that using WordNet's taxonomy with $SBSM_{\times 5}$ and the Related Sense WSD algorithm significantly improves over the current Wordsmyth test links.

4.6 Further Work

A number of areas are considered for extending the work presented in this chapter. These can be grouped into four categories:

1. Improvement of evaluation techniques.

2. Work to improve the similarity values assigned by the SBSMs.

3. Evaluation of the effect of considering additional information from WordNet, for instance evaluating to what extend considering meronyms may assist in the calculation of semantic similarity.

4. Complete Wordsmyth links to WordNet using the techniques presented in this chapter.

4.6.1 Improving Evaluation Techniques

Currently the data available to compare algorithm results with human judgements only has a maximum of 65 human judgements. Whilst it can be shown that correlations using the existing data sets are statistically significant, the current number of examples may give biased results. Using a larger set of human judgements has a number of features that are beneficial to making a more objective test:

- An increased number of word-pairs will make distinctions between the similarity of word-pairs harder to judge by humans, especially if more word-pairs are deemed similar rather than dissimilar.

- More word-pairs will reduce bias potentially introduced in current tests.

- A larger number of human judgements to compare with will produce a better correlation estimate giving a more objective result.

4.6.2 Improving the Similarity Values Assigned by SBSMs

The current SBSMs have been created by considering how to make use of a lexical taxonomy for evaluating semantic similarity between nouns. However, further consideration can be given to the way in which similarity values will be distributed across noun pairs of varying similarity (for instance, low, medium and high similarity according to the human judgement data sets). Section 4.5.1 gives a rather crude example of how to improve the distribution of values produced by $SBSM_{+7}$. However, in order to fine tune the values produced by the SBSMs further investigation is required. The example in section 4.5.1 improves results by adjusting values from $SBSM_{+7}$ such that the lines of best fit produced by linear regression techniques on the values produced

closer match the line of best fit for the human value judgements against word-pair rank. This assumes that the line of best fit is linear between similarity value and similarity rank.

For further fine-tuning of the SBSMs, a non-linear line of best fit should be calculated for the human similarity values with the Resnik (1999) data set against word-pair rank. The Resnik data set is chosen for this task as it is the smallest data set and therefore results with the Rubenstein and Goodenough (1965) and the Miller and Charles (1991) data sets should still be objective. This ensures that the technique does not fit results to a particular test, thus artificially improving results. The values for all SBSMs should then be adjusted such that the similarity values calculated for the Resnik data set produce a line of best fit closely matching the human line of best fit. This will mean that the values produced by the SBMSs across different ranking word-pairs will more closely match the distribution of the values assigned by humans. The effect of this adjustment should then be measured using the Rubenstein and Goodenough (1965) and Miller and Charles (1991) data sets, and also using application-oriented evaluation techniques. It is assumed that the resulting Pearson's coefficients will be more comparable across the different SBSMs after such a change. This would make the Spearman's rank coefficient a more objective test as it is not affected by the adjustment of the values. If this holds true, and if the adjustments made whilst considering human judgements improve results for other applications, such a technique will be applicable for similarity measures as a way of fine-tuning their results.

4.6.3 Considering and Evaluating Further WordNet Relations for Semantic Similarity Measures

Currently, the SBSMs can use the WordNet meronymy relationships to assist in the calculation of similarity between two noun senses. However, due to the execution times for WordNet to collect meronymy information for a noun and the experiments used, this feature of the SBSMs remains to be evaluated. Suitable techniques are firstly required to increase the speed of searching for meronyms.

Other WordNet relationships should also be considered further to evaluate their usefulness in calculating semantic similarity. However, care must be taken in how such relationships are applied as work has shown that using all relationships in an unguided way can produce worse results (St-Onge, 1995; Hirst and St-Onge, 1998). An example

of a further relationship being considered is to use hypernyms of a noun's meronyms as a source of information from which to calculate semantic similarity.

4.6.4 Complete Wordsmyth Evaluation

Automatically disambiguating all noun entries in Wordsmyth does not pose a large amount of extra work. However, the human disambiguation of the words necessary to produce precision, recall and accuracy results for the verification of the automatic process requires additional time.

4.7 Summary

This chapter has introduced a number of ideas fundamental to similarity measures based on the general shape of lexical taxonomies (SBSMs), using WordNet as the source for the required taxonomies.

Initial tests comparing the new SBSMs to human judgements give results which compare well with other existing similarity measures. However, larger number of human judgements should be obtained in order to further substantiate this evaluation. It is also shown that using Pearson's Product Moment Coefficient may not be the best correlation coefficient to compare different similarity measures as the technique is too sensitive to the values of similarity assigned. It is argued that the relative ranking of word-pairs according to similarity is of more interest as values from similarity measures can be adjusted after they are calculated, and because the human assignment of values is subjective in its nature. Therefore correlation using Spearman's Rank Coefficient may be more suitable for comparing different measures. This is especially true when values from measures are not normalised between a range of values. The evaluations show that SBSMs come close to matching human performance and as such they show an improvement over current state-of-the-art measures at simulating human decisions about similarity between words.

A final more application-orientated approach to evaluating these SBSMs is used to evaluate the similarity measures with a number of simple WSD algorithms for use with noun groups. The evaluation uses nouns contained within Wordsmyth thesaurus entries to test the disambiguation performance of the different WSD algorithms and SBSMs. The best results are also compared with a collection of experimental links created using

Resnik's Information Based similarity measure and WSD algorithm (Resnik, 1995a,b, 1999), summarised in table 4.13. The results of the final comparison show a marked

	Precision	Recall
Wordsmyth Test Links	80.44%	71.26%
Related Senses with $SBSM_{\times 5}$	88.28%	90.94%

Table 4.13: WSD Comparison with Wordsmyth Experimental Links to WordNet

improvement over the information based approach by using a SBSM with a WSD algorithm considering only related senses of words for classifying the senses of nouns in Wordsmyth thesaurus entries.

Chapter 5

Introduction to Word Sense Disambiguation

The field of Word Sense Disambiguation (WSD) has been of considerable interest since the early stages of natural language processing (NLP) (Ide and Véronis, 1998). WSD aims to provide a sub-component, in general for other NLP applications, to automatically relate words in text with definitions according to one or more lexical resources. Once a WSD system determines a single word sense for a word, that word is said to be sense tagged, or sense labelled. Thus, most research treats WSD as an "intermediate task" (Wilks and Stevenson, 1996; Gonzalo et al., 2003) in some larger NLP process.

Research on WSD, given the length of time it has been undertaken, has had limited success. WSD has been considered an AI-complete problem (Gale et al., 1993), meaning that it presupposes a solution to the "strong AI problem", i.e. the simulation of human intelligence, and therefore can only be solved once all other difficult problems in AI have been tackled. Improvement in the representation of knowledge, especially with the emergence of recent semantic networks and corpora of sense labelled text, such as WordNet and its associated corpora, resulted in WSD becoming a more tractable problem. This is illustrated by the increase in the number of techniques since the 1990s when public resources such as WordNet became more available. Indeed, the field of WSD has also grown in prominence, and "is frequently cited as one of the most important problems in NLP research today" (Ide and Véronis, 1998).

This chapter initially describes how WSD helps other NLP tasks. Section 5.2 presents a brief history of work particularly important to WSD, giving particular prominence to some of the most influential techniques. A number of techniques of interest in

relation to the work described in chapter 6 are introduced in section 5.3. Finally, given the increased activity during the 1990s in the field of WSD, a gold standard evaluation framework called SENSEVAL was introduced in order to help researchers objectively compare results from different systems in a standard accepted way. Section 5.4 gives details of the various SENSEVAL conferences, showing how evaluation of WSD is performed.

5.1 How WSD can Help other WSD Problems?

The disambiguation of word meanings in texts is believed to be fundamental to improve results within the following applications of NLP:

- Machine Translation (MT)
- Information Retrieval (IR)
- Content and Thematic analysis
- Parsing
- Speech Processing

Early research within these fields, particularly with MT, resulted in the emergence of WSD, although for a long period the majority of the WSD research was performed as part of larger projects. Within each field, the polysemy of words is seen as one of the major factors influencing the results from the techniques implemented. This early work was able to place restrictions on domains and granularity of the resources used, and in some cases quite accurate results were produced.

5.1.1 Machine Translation (MT)

A central issue in translation is selecting the correct word in a target language to reflect the intended meaning in the source language. This is a consequence of different sense distributions of words in the source language to those of the target language, and gives rise to various definitions of a word being realised by different words in the target language, for instance the Portuguese word “sentido” can be realised by any of “meaning”,

"sense", "side", "direction" or "feeling" in English. The accuracy from current state-of-the-art WSD systems means that they are not widely used by current MT systems given alternative approaches. Whilst most modern translation systems make use of statistical information from bilingual resources in order to side-step the need to explicitly use WSD, such resources are limited and unavailable for a number of languages, such as sign languages which currently have no widespread written form. It is "abundantly clear to all in MT that word sense ambiguity is a huge problem" (Kilgarriff, 1997).

5.1.2 Information Retrieval (IR)

Most classic IR techniques find information by matching words in documents, however this produces two significant problems:

1. Synonymy has the consequence that more than one word may reflect a particular concept of interest. Without considering synonyms of a word, appropriate documents may be missed during a search. These situations mean that recall drops for word-form based techniques.

2. Given the polysemy of words, if a match is based on word-form there is no guarantee that all matches found reflect the intended meaning of the word. Such situations reduce the overall precision of these techniques.

A number of researchers have evaluated the impact the use of WSD has on IR tasks (Weiss, 1973; Salton and McGill, 1983; Salton and Buckley, 1989; Voorhees, 1993; Schütze and Pedersen, 1995; Towell and Voorhees, 1998). Results have been mixed, showing that WSD could improve results for IR by at least 1%, and in some cases by up to 14%. Given the current performance of state-of-the-art WSD techniques, actual findings so far have been fairly discouraging (Kilgarriff, 1997), and in many cases results actually declined. Thus, some have concluded that whilst WSD has the potential to improve accuracy, "the performance of IR systems is insensitive to ambiguity but very sensitive to erroneous disambiguation" (Sanderson, 1994).

5.1.3 Content and Thematic analysis

A number of content and theme tagging approaches make use of a set of words whose distribution is analysed in order to classify them against pre-defined categories. It has

long been believed that WSD can improve results (Quillian, 1967; Litkowski, 1997) so that words are only considered when used in a pre-determined sense. The problems faced here are related to problems faced by the IR community.

5.1.4 Parsing

Of interest for parsing techniques is the use of WSD to tackle a number of problems, such as determining the gender of a noun in Latin-based languages where the word can be either male of female depending on its sense. WSD is particularly important for agreement phenomena and prepositional phrase attachment for verbs (Jensen and Binot, 1987; Whittemore et al., 1990; Hindle and Rooth, 1993; Alshawi and Carter, 1994).

5.1.5 Speech Processing

A characteristic of words that creates a large problem for speech recognition systems is homonymy, when words are pronounced in the same way but are spelt differently. A classic example of this is seen in the sentence:

> "Write to Mr. Wright right away."

WSD assists speech recognition systems by only presenting for consideration the definitions of the different words and selecting the word with the most likely sense within the context it is found.

5.2 Historically Important Events in WSD

WSD research emerged from various fields of NLP, and for a long period of time the majority of WSD research was performed as part of larger projects, often placing restrictions on domains and granularity of the resources used, but able in some cases to give very accurate results. In the 1960s, work appeared where WSD was studied in isolation, although due to a lack of resources, many examples of such early work produced very limited hand-tailored systems (Weiss, 1973). Hirst (1987) gives a comprehensive review of these early systems. Once more suitable resources became available for large-scale WSD to be possible, many of these early systems did not scale-up well to

larger systems and there was a significant shift from producing hand-tailored systems to systems making use of automatically collected information. Ide and Véronis (1998) give a comprehensive review of the history of WSD and the open problems that face the field of WSD. The most recognised problems facing WSD are involved with the collection and representation of information. These range from the "lexical-bottleneck" problem, where researches are unable to collect large enough quantities of hand labelled examples, to issues about handling different domains or limiting the domains considered and the granularity to which sense distinctions are made in lexicons.

The remainder of this section introduces some historic events and techniques in the field of NLP, from early work in MT, to the development of recent systems using information collected from corpora to assist WSD.

5.2.1 Early Machine Translation (1950s)

The earliest examples of word polysemy becoming a real issue in NLP began in early work of MT field. Hutchins (1997a,b) discusses the pioneering work in MT, much of which was limited to technical texts from restricted domains. During this period, a number of key ideas were established which persist today. Probably one of the most influential ideas was that of context windows, first discussed in a memorandum by Weaver (1949). In this memorandum, Weaver made the following statement relating context to meaning, and giving a suggestion for a definition of context:

> "If one examines the words in a book, one at a time as through an opaque mask with a hole in it one word wide, then it is obviously impossible to determine, one at a time, the meaning of the words. ... But if one lengthens the slit in the opaque mask, until one can see not only the central word in question but also say N words on either side, then if N is large enough one can unambiguously decide the meaning of the central word. ... The practical question is: 'What minimum value of N will, at least in a tolerable fraction of cases, lead to the correct choice of meaning for the central word?'"(Weaver, 1949)

This idea was exploited by various researchers over the subsequent years (Kaplan, 1955; Koutsoudas and Korfhage, 1955; Choueka and Lusignan, 1985; Preiss, 2001). Tests were performed with human subjects to find the smallest reasonable size for

a context window. The results found that humans interpreted meaning with context windows of size N = 2 to an equal accuracy to when they had the entire sentence, therefore concluding that N = 2 is an adequate size for a context window.

In addition to the idea of context window, Weaver's memorandum also calls for significant WSD work to be performed, making particular reference to his views of the statistical nature of the problem of WSD, an idea that is still prevalent today in much of WSD research.

Reifler (1955) is one of the first researchers to write about the link between syntax and semantics, and the ideas of "semantic coincidences" between a word and its context. He illustrates how the same words in German can express distinctly different meanings according to the syntactic configurations in which they are used, such as in gerund phrases, adjectival phrases or noun phrases. Later, Gougenheim and Michéa (1961) presented similar ideas for French, in which the sense of the verb "grossir" is determined whilst considering its syntactic complements.

Initial MT work recognised the difficulty in handling open-texts, therefore creating resources to simplify the overall translation problem. By splitting texts into fields of knowledge, or domains, such as physics, biology and economics for instance, the problems posed by synonym and polysemy could be constrained to some extent. Given the recognised importance of domain in WSD for MT, efforts were made to create a number of specialised lexicons for use in specific and limited domains (Oswald-Jr., 1952). The entries in these lexicons only contained definitions relevant to the domain of interest, and definitions for distinctions made during the translation between the two languages for which they were constructed to be used. The resulting lexicons contained no more than two-to-one correspondences between senses of the source and target languages. Further to these specialised lexicons, techniques were investigated to create richer knowledge representations for WSD. Most of the earliest work on knowledge representations used inter-lingua approaches (Richens, 1958); for instance the earliest implemented technique by Masterman (1957) for automatically creating a resource using a Latin-English dictionary together with Roget's Thesaurus that later developed into the idea of semantic networks.

Some of the earliest studies on the phenomenon of polysemy were performed by Harper (1957a,b), limited to texts within Physics and Science domains. He analysed the polysemy of words in a Russian dictionary, reporting 8.6 average polysemy and that English and Russian words are 5.6 quasi-synonymous.

Most early MT work stopped in the mid-1960s due to the withdrawal of funds in the United States following the conclusion in a report by the Automatic Language Processing Advisory Committee (ALPAC, 1966) that MT had not improved since the early 1950's (Hutchins, 1996). At this time there was a change of ideas in computational linguistics away from statistics, most notably with the shift towards rule-based systems as opposed to probabilistically based systems such as Chomsky's ideas on universal grammar and transformational rules of describing syntax promoted (Chomsky, 1957, 1965).

5.2.2 Artificial Intelligence Methods (1960-70s)

Some of the most important work to result from the considerable shift in paradigm which the ALPAC report produced was the investigation into semantic networks and symbolic approaches to organising lexical information. Masterman (1961) selected 100 primitive concepts from which to organise 15,000 entries in a dictionary. This approach was the first to represent words as nodes in a network, where the links between words represent semantic relationships. These ideas were the precursors to modern lexicons such as WordNet. Using such semantic networks, Quillian (1961, 1962a,b, 1967, 1968, 1969) introduced ideas for WSD which were later developed to connectionist models using spreading activation models (Meyer and Schvaneveldt, 1971; Collins and Loftus, 1975; McCelland and Rumelhart, 1981) where ideas can still be found in some more modern techniques such as Neural Network approaches (Cottrell, 1985; Hearst, 1991; Towell and Voorhees, 1998).

Wilks (1968, 1969, 1973, 1975a,b,c,d) introduced the highly influential "preference semantics" approach for WSD, described as "essentially a case-based approach" (Ide and Véronis, 1998). The preferences for combinations of lexical items are based on semantic features and were defined using 60-80 semantic primitives, or 'elements'. The semantic primitives selected were influenced by work earlier performed by Masterman. Although this technique was shown to successfully handle metaphor in language amongst other examples, Boguraev (1979) later demonstrates that such an approach is inadequate to handle highly polysemous verbs and attempts to improve Wilks' method using further linguistic information. Boguraev (1979) also links his approach to WSD with syntactic disambiguation. Wilks' approach still has a large influence in recent work on WSD, and has been recently revisited by Wilks and Fass (1990), McRoy

(1992) and Resnik (1993, 1996, 1997).

A number of template-based approaches arose during the early 1970s. Weiss (1973) approached the problem of WSD by testing a number of general context rules and template rules. After limited testing, Weiss concluded that template rules produce better results than general context rules for WSD. Template rules were created based on 20 instances of 5 words, and the accuracy of the WSD system was evaluated by disambiguating a further 30 examples for each word. Results given are approximately 90% precision and recall for the examples tested. In a larger experiment with 6,000 words, Kelly and Stone (1975) used a similar approach to Weiss, but also including rules checking certain grammatical aspects of context. They concluded after using various different approaches that "such a strategy cannot succeed on a broad scale".

Hayes (1976) and Hirst (1987, 1988) introduced techniques using case frames in combination with a semantic network to disambiguate senses of words. The disambiguation process itself is similar to Quillian's approach, where the context of a word activates nodes in the network to find semantic paths between them. This approach worked well to disambiguate words at the homograph level. However, it was less successful at finer grained levels of polysemy. Hirst's approach progressively removes inappropriate senses using "polaroid words". One of the main aspects of his technique was the inability to disambiguate any metaphorical interpretations of words as the polaroid words would eventually eliminate all available senses.

The main criticism of work from this era is that the systems worked on toy examples that were often unnatural (Sanderson, 1996; Ide and Véronis, 1998). In the main, this was due to the difficulty in finding varied and sufficiently numerous examples to work with, otherwise known as the lexical-bottleneck problem, or more generally the knowledge acquisition bottleneck problem (Gale et al., 1993). Given the level of effort required to build practical systems using the ideas introduced, these techniques remain theoretically interesting. However, they are of little practical use except in the most limited domains.

5.2.3 Knowledge-Based Methods (1980s)

The 1980s marked a re-birth in statistical techniques following the introduction of a number of significant machine-readable resources, such as Longman's Dictionary of Contemporary English (LDOCE) (Procter, 1978) and Collins English Dictionary

(CED), Roget's Thesaurus (Chapman, 1977) and WordNet (Fellbaum, 1998). This stimulated the production of disambiguated corpora from which to collect statistical information. Much debate exists about the exact number of senses necessary to describe all uses of a word, and the exact nature of how concepts can be organised hierarchically. Some problems are highlighted in the work presented in chapters 3 and 4, namely to do with the actual organisation of data within WordNet. Regardless of these debates, a large amount of work has been performed trying to harness the necessary semantic information from the available resources. This work shows a marked change from work in limited domains, or hand-crafted systems working with a small number of examples, to a more ambitious approach of creating techniques attempting to disambiguate words in open-texts. Krovetz and Croft (1989) gave an account of some of the most prominent machine-readable dictionaries (MRD) introduced during the 1980s.

Of the most influential methods introduced during this era for WSD is Lesk's dictionary overlap technique (Lesk, 1986). This technique calculates the overlap of words in the dictionary definitions (or glosses) of the target word's senses against the words contained in the definitions of the context words. A scoring function based on the co-occurrence of words in the definitions is used to determine the appropriate sense of the target word by selecting the top ranking definition of the word. Lesk showed results of 50%-70% accuracy using the Oxford Advanced Learner's Dictionary of Current English. The accuracy of his approach is highly sensitive to the exact wording of the definitions in the lexical resource used. Many examples can also be found where combinations of words cannot be classified, as they share no common words in their definitions. Wilks et al. (1990) relaxed these problems by creating a network using definitions from LDOCE and words commonly co-occurring with words found in the LDOCE definitions. This way, more semantically related words are available for an approach similar to Lesk's. The technique was evaluated using 197 sentences containing the word "bank". Results of 45% accuracy for disambiguation at LDOCE's fine-grained sense distinctions (13 senses) of "bank" and 79% for the more coarse-grained sense distinctions (5 senses) of LDOCE are reported.

Véronis and Ide (1990, 1991, 1995) used a large neural network to disambiguate text, creating the network using CED. The network links words with senses, and senses are in turn linked with the words in their definitions, and from those words to their senses, etc.... Ide gave results of 70%-85% accuracy on small experiments with varying parameters applied to the method. Sutcliffe and Slater (1995) tested the technique

on a full text, and gave results of 72% accuracy in contrast to 40% accuracy using Lesk's technique and CED with the same text.

LDOCE was widely used with later techniques (Guthrie et al., 1991; Cowie et al., 1992; Demetriou, 1993) applying its additional information such as box and subject codes, presented in the form of general primitives for each word. Later work has shown that matching LDOCE box codes alone is insufficient for WSD (Braden-Harder, 1993). In general, success for the LDOCE has been relatively modest compared to the work using CED. This may be due to differences in the average polysemy of words between the two resources.

Whilst MRDs provide a rich lexical source of information from which to perform WSD, it is recognised that further pragmatic information not included in MRD is required to further improve results. Thesauri organise information into groups of related words and therefore provide a source of more general relationships between words. Masterman (1957), as mentioned earlier, was the first to use Roget's thesaurus for WSD. Further examples of WSD with Roget's thesaurus can be found (Patrick, 1985; Yarowsky, 1992), the latter using 100 word contexts from a corpus of texts to create word classes for words with common categories using information about the collected contexts. Using Bayes' Rule on probabilities calculated from Grolier's Encyclopaedia (10,000,000 words), the classes of new examples of polysemous words are calculated, where the class is assumed to represent the sense of a word. An accuracy of 92% was given for 12 words with an average polysemy of 3 categories according to Roget's.

5.2.4 Corpus-Based Methods (1990-2000s)

The most recent work in WSD has involved empirically based techniques often attempting to reduce problems posed by the lexical-bottleneck problem. The most successful knowledge source to date, WordNet (Fellbaum, 1998), was created manually, although a number of attempts have been made to automatically generate such resources from available lexical resources (Michiels et al., 1980; Calzolari, 1984; Chodorow et al., 1985; Markowitz et al., 1986; Byrd et al., 1987; Nakamura and Nagao, 1988; Klavans et al., 1990; Wilks et al., 1990). Rather than creating large knowledge sources for WSD, work turned to create WSD systems using information automatically "learned" from corpora. A number of sense disambiguated corpora were created to aid this research. Some examples are given in table 5.1. It is important to note that these

resources are far smaller than corpora used for other statistical tasks due to the effort required in manually sense labelling them. The relatively small size of the avail-

Resource	What was tagged?
Semcor (Landes et al., 1998)	A varied subset of texts from the Brown corpus and the novel "The Red Badge of Courage" containing 234,113 instances of 23,346 lemmas in passages were 103 manually tagged with WordNet senses.
Semcor (Miller et al., 1993)	200,000 instances of 1,000 selected words were hand tagged from subset of Brown corpus.
HECTOR (Atkins, 1993)	The first example of creating a lexicon and sense tagged corpus in combination. 300 "word types" (dictionary headwords) with 300 to 1,000 instances in a pilot version of the British National Corpus (20,000,000 words) were tagged with senses from a lexicon created in tandem.
(Smeaton and Quigley, 1996)	8,816 instances of 2,304 lemmas from image captions were tagged with WordNet senses.
DSO Corpus (Ng and Lee, 1996)	192,800 sentences containing 120 selected nouns and 71 selected verbs from a subset of Brown and Wall Street Journal corpora were hand tagged with WordNet senses.
Cambridge University Press (Harley and Glennon, 1997)	4,000 words were hand tagged against the senses of the Cambridge International Dictionary of English (CIDE).
(Wiebe et al., 1997)	25 highly frequent verbs in 12,925 sentences from Wall Street Journal Treebank corpus were hand tagged (Marcus et al., 1993).
(Towell and Voorhees, 1998)	Over 12,000 instances of the noun "line", the verb "serve" and the adjective "hard" from the Wall Street Journal corpus were hand tagged with WordNet senses.
(1998)	Senseval 1 evaluation resources.
(2001)	Senseval 2 evaluation resources.
Open Mind Word Expert (Chklovski and Mihalcea, 2002)	An on-line resource provides an interface for users to add to a sense tagged corpus with WordNet senses

Table 5.1: Examples of Sense Tagged Corpora

able corpora undermines the use of established statistical approaches in NLP for WSD (Towell and Voorhees, 1998). Currently, the most accurate statistical systems in NLP have been developed for speech recognition and part-of-speech (POS) tagging. Table 5.2 summarises the size and complexity of the resources used for some state-of-the-art NLP systems. The task for WSD with WordNet would require statistical classifiers to disambiguate a total of 121,962 words and 173,941 senses, therefore the size of an adequate corpus would require a much greater number of examples than are currently available for established statistical techniques to be adequately applied to WSD. With the current level of storage capacity available, it is possible to collect such quantities

Problem	System	Accuracy	Ambiguity	No of Examples
Speech Recognition	(Rabiner and Juang, 1993)	95%	625 triphones	In the order of 1,000s of sentences
POS tagging	(Brill, 1991)	97%	64 POS tags	Corpora of 1,500,000 words

Table 5.2: Summary of Resources Used for Two State-of-the-Art NLP System

of information, however the effort required for the manual tagging of texts remains the main bottleneck and it is unlikely that a suitably large corpus will be available in the near future.

Systems developed during this period fall into one of the following categories of techniques:

- Knowledge-based Techniques – These techniques make use of information solely from lexical resources, such as the approach developed by Lesk and other techniques created during the 1980s. Further techniques use lexical information to measure similarity between words as a basis for WSD (Sussna, 1993; Agirre and Rigau, 1995, 1996; Li et al., 1995; Preiss, 2001). Levow (1997) gives further discussion about knowledge-based techniques.

- Supervised Training Techniques – These techniques require a tagged corpus of examples from which to train the system to disambiguate words, such as (Bruce and Wiebe, 1994; Ng and Lee, 1996; Lin, 1997; Wilks and Stevenson, 1997a,b,c, 1998b,c; Stevenson and Wilks, 1999, 2000; Ng, 1997; Stetina et al., 1998). While much work has been performed in producing such resources, it is believed that the number of examples available is still too few to produce high quality results using traditional statistical approaches for open-text WSD. However, the current state-of-the-art WSD use supervised techniques.

- Unsupervised Training Techniques – Rather than requiring large quantities of manually-tagged data, some research has attempted to train systems either totally without tagged examples, such as (Yarowsky, 1995; Pedersen and Bruce, 1997), or only using a small tagged sample from which to gather further non-tagged data for training (Hearst, 1991). These techniques have, in cases, made use of information directly from the World Wide Web given the large corpus

of information potentially available. However, techniques have so far produced fairly modest results.

- Hybrid Techniques – Some techniques have approached the WSD problem using a combination of knowledge-based and statistical techniques in an endeavour to improve performance by combining the strengths of these two approaches (Karov and Edelman, 1996; Mihalcea and Moldovan, 1998, 1999, 2000).

Rather than seeing research becoming more standardised or approaches becoming limited to a smaller set of techniques, the work undertaken during recent years appears to be more divergent, with researchers using an increasingly different number of knowledge sources and evaluation techniques.

Gale et al. (1992a) are some of the first authors to discuss the problem of evaluation for WSD in depth. Toward the end of the 1990s, a number of other researchers turned their attention to producing standard platforms for the evaluation and comparison of WSD systems (Resnik and Yarowsky, 1997; Kilgarriff, 1998a,b; Véronis et al., 1998), leading to the SENSEVAL conferences. These conferences produced a number of resources on which to train systems and a standard platform for WSD systems to be evaluated in various languages. These resources are known as the current gold standard for WSD evaluation. This has allowed for techniques to be compared in an objective way.

5.3 Recent WSD Techniques of Particular Interest

A number of WSD techniques are of particular interest to the research reported in chapter 6 and have had an influence in the design of the WSD approach described there. Wilks and Stevenson (1997a,b,c, 1998b,c) and Stevenson and Wilks (1999, 2000) approach the problem of WSD using a combination of results from partial-taggers. Lin (1997) uses a different definition of context to that typically found in the current literature, according to the thematic and syntactic information of a word, in order to improve WSD performance. Lastly, Suárez and Palomar (2002) evaluate a number of common statistical features using a Maximum Entropy (ME) model for WSD.

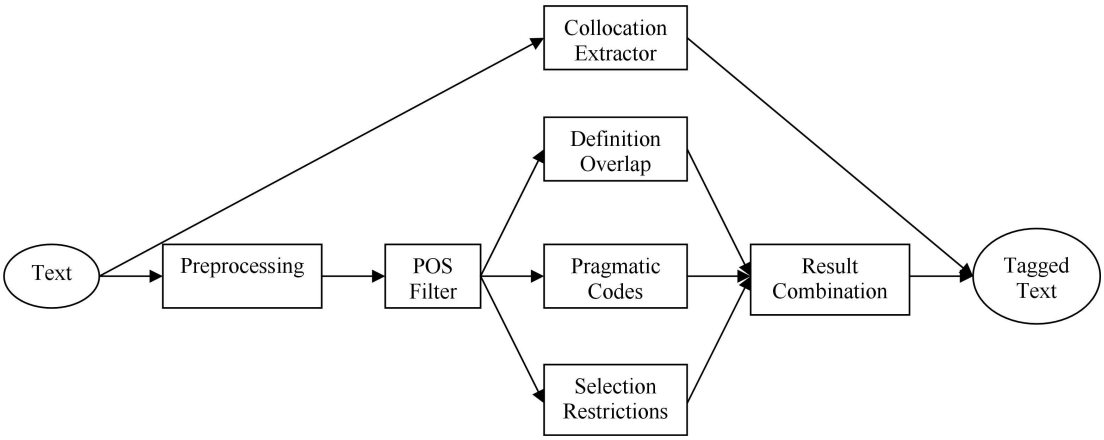


Figure 5.1: Wilks and Stevenson (1997a,b,c, 1998b,c); Stevenson and Wilks (1999, 2000) Partial Tagger WSD System

5.3.1 Partial WSD Tagger Approach

In an approach first proposed for WSD by McRoy (1992), and later adopted by Ng and Lee (1996) and Wilks and Stevenson (1997a,b,c, 1998b,c); Stevenson and Wilks (1999, 2000), a WSD system is developed using a combination of simple partial-taggers to tag all words in a text using LDOCE word senses. In Wilks' approach, a Brill POS tagger (Brill, 1991, 1992) is initially used to restrict the senses considered for each word to those comparable with the tagged word class, a technique now widely utilised in WSD (Wilks and Stevenson, 1998a). In experiments, this initial step reduced around 87% of possible word senses. The system then uses three "weak" tagging methods to either remove unlikely senses (filters) or to consider highly likely senses of a word (partial-taggers). The remainder of this thesis shall refer to both types of taggers as partial-taggers. The results of each partial-tagger are used as input to a learning algorithm to find an optimum combination of the results from the partial-taggers in order to produce a final solution. Figure 5.1 illustrates the complete system (Stevenson and Wilks, 1999).

Pre-processing & POS Filter

Before any of the tagging modules are able to process text, the text is pre-processed to mark 'content words', words with entries in the lexical resource being used. This

module is composed of four components from the Sheffield University Information Extraction system LaSIE (Gaizauskas et al., 1995); a POS tagger (Brill, 1991, 1992), a named entity recogniser, a shallow syntax parser and a lexical lookup component using the LDOCE.

The output from the pre-processing module is used by the POS filter to map each of the content words to senses from their equivalent entry in LDOCE. Only the senses of entries for the grammatical categories assigned by the POS tagger are retained.

Collocation Extractor

Wilks and Stevenson (1997a,c) give a limited discussion about using verb preposition information as found in sub-categorisation frames and in LODCE example sentences to disambiguate words. Using these frames and examples, a partial tagger is constructed to restrict interpretations using Yarowsky's one sense per collocation technique (Gale et al., 1993; Yarowsky, 1993, 1995).

Dictionary Overlap

Cowie et al. (1992) introduce a similar approach to Lesk's dictionary overlap technique by applying a simulated annealing algorithm to the problem in order to make the technique more practical for full sentences. Their approach was able to select senses from up to 10^{10} different combinations. The original algorithm was applied to WSD, giving results of 47% accuracy to the sense level with LDOCE, and 72% to the homograph level.

Wilks and Stevenson use an adapted version of the Cowie et al. algorithm, normalising the influence each common word makes to the overall score of senses using the length of descriptions. This avoids incorrectly preferring senses with longer definitions. They report an improvement in efficiency to 65% accuracy to LDOCE's sense level (Stevenson and Wilks, 1999), however these results are deemed not to be statistically significant (Wilks and Stevenson, 1997c). A further adaptation to the original algorithm was made in order to return a set of suggested senses.

Pragmatic Codes

The pragmatic codes from LDOCE can be used to restrict senses by only selecting senses associated with the most likely pragmatic code for the context in which it is

found. A technique similar to Yarowsky (1992) is used to calculate the most likely pragmatic code using a wide context window of 50 words to the left and right of the word being tested, and is trained using a portion of the British National Corpus (BNC) containing 14,000,000 words (Burnard, 1995). Using a voting system, results of 79% of the available senses can be disambiguated.

Selection Restrictions

In a similar approach to Wilks (1972), LDOCE's subject codes for nouns, related to each other via the hierarchical relations used by Bruce and Guthrie (1992), and grammatical links from the shallow parser are used with selection restrictions to constrain word senses. All senses that do not break any of the selection constraints are considered further. In tests, 44% of words were correctly disambiguated using this approach (Stevenson and Wilks, 1999, 2000).

Combining Results

The final module for WSD collects the results from the partial-taggers and selects a final sense for each word. The TiMBL memory based learning algorithm (Daelemans et al., 1998) is trained using a number of annotated examples and the results from the partial-taggers. The implemented module disambiguates new unclassified instances by determining which training example is most similar to it. In cases where more than one sense remains appropriate the first sense according to LDOCE is selected as the final choice.

Evaluation Results

The system was evaluated using Semcor as a tagged corpus. As Semcor tags refer to WordNet senses, SENSUS (Knight and Luk, 1994) was used to map the WordNet tags in Semcor to LDOCE tags. Given significant gaps in the mapping, the final LDOCE tagged corpus contained 36,869 words. The system was trained using 10-fold cross validation over the entire corpus, producing results of 90% accuracy at the LDOCE sense level and greater than 94% accuracy for its homograph level.

5.3.2 Syntactic Local Context Based Approaches

The most explored definition for context in NLP for statistical classification is the idea of context windows first discussed by Weaver (1949). He proposes that humans can accurately make judgements for the meaning of a word using a window of words to the left and right of the word of interest. A number of recent approaches make use of syntactic relations and thematic grids as a basis for a definition of context, such as (Lin, 1997; Resnik and Diab, 2000; Green et al., 2001a,b). The question about syntactic context can be seen in earlier work, such as in closing remarks of Lesk (1986). Lin (1997) defines context using dependency relationships gathered from a dependency grammar (Hudson, 1984; Mel'cuk, 1988). Such grammars relate words syntactically using asymmetric binary links where one word is the head in the relation, the other word is the modifier in the relation and the link represents a dependency relationship. The local context for a word can be represented by any of the relationships using the triple in 5.1.

$$\{\text{dependency relationship, word, position}\} \tag{5.1}$$

This definition of local context is used by Lin for WSD by collecting local context examples from a corpus to create a Local Context Database (LCD). For each local context, the triple in the form such as 5.2 is stored containing information about words related via a dependency relationship in a corpus of examples with the main word of the local context (Dunning, 1993).

$$\{\text{word, frequency, likelihood}\} \tag{5.2}$$

To disambiguate a new unclassified example, w, from a parsed sentence or text, all local contexts of w are collected and stored in LC_w. The set of most likely selectors is selected from the examples in the LCD using equation 5.3.

$$Selectors_w = \left(\bigcup_{lc \in LC_w} C(lc) \right) - \{w\} \tag{5.3}$$

The word w is tagged with the sense s that is most similar to $Selectors_w$. All other instances of w are also tagged with sense s implementing the "one sense per discourse" theory (Gale et al., 1992b, 1993). Lin uses his own similarity based disambiguation technique for measuring the similarity of the selectors and possible senses of words.

For evaluation, Lin constructed a system using the 25,000,000 word Wall Street Journal corpus. A LCD was constructed consisting of a total of 354,670 local contexts, all with likelihood ratio greater than a value selected arbitrarily, in this case 5. The local contexts consisted of 1,067,451 words. The system was evaluated using the "press reportage" section of Semcor consisting of around 2,000 words, and the algorithm was only applied to nouns. Furthermore, three test conditions where evaluated:

1. The selection is correct if $similarity(s_{answer}, s_{key}) = 1$ (Strictest criteria)
2. The selection is correct if $similarity(s_{answer}, s_{key}) \geq 0.27$ (Relaxed criteria)
3. The selection is correct if $similarity(s_{answer}, s_{key}) > 0$ (Weakest criteria)

The threshold for 2 was calculated empirically. Table 5.3 presents Lin's the results.

Criteria	System Accuracy
1	56.1%
2	68.5%
3	73.6%

Table 5.3: Accuracy of Lin (1997) WSD system

5.3.3 Maximum Entropy (ME) Approaches

There has recently been a marked increase in the use of ME statistical models for WSD systems. A description of the ME framework is given in chapter 6. However, the critical and most interesting aspects of ME for WSD involve the selection of features used by the ME model. A feature is implemented using a binary feature function returning 1 if the conditions specified by the function are true, 0 otherwise. Each feature is allocated a coefficient, or weight, which the ME framework trains in order to closely model a collection of prior probabilities collected from a corpus of examples. Suárez and Palomar (1993, 2002), Dang and Palmer (2002) and Klein et al. (2002) test a collection of different feature types designed to work using context defined as a context window around the ambiguous target word. The features selected for the analysis come from work produced by Ng and Lee (1996) and Escudero et al. (2000).

Traditionally, features model every combination of the information shown in triple 5.4.

$$\{\text{word sense, example feature of interest, position}\} \tag{5.4}$$

Suárez and Palomar propose an alternative approach designed to greatly reduce the number of features present in the final model. Rather than create a feature for each example of 5.4, all examples of interest for a word sense at particular locations are gathered to form a set of examples of interest. This enables the creation of features modelling combinations of tuple 5.5 as all information of interest is recorded in a single set.

$$\{\text{word sense, position}\} \tag{5.5}$$

The feature templates used to create features from corpus examples for the two types of features are referred to as template-word and template-set respectively. The templates are then used to extract features of the following types:

- Template-Word
  - 0-features
  - S-features
  - Q-features
  - Km-features
- Template-Set
  - L-features
  - W-features
  - B-features
  - C-features
  - P-features

0-features

0-features model information about the target word itself. For nouns and adjectives, aspects of word morphology are modelled, such as capitalisation and quantification. For verbs, additional aspects are modelled such as tense.

S-features

S-features model words appearing at specific positions relative to the target word, for instance if "red" appears to the left of "shirt" in an example and "shirt" is the target word, an S-feature models the fact that "red" appears in location -1 relative to "shirt".

Q-features

Q-features model the POS of words appearing in a 3-word window around the target word. These features look similar to S-features where co-occurring words are substituted by their grammatical category.

Km-features

Words found appearing for at least $1/m$ examples for a word sense are used to create Km-features. The feature simply models the fact that such words frequently co-occur with the target word.

L-features and W-features

L-features model the set of lemmas found at positions close to the target word. Sets for positions -3 words, -2 words, -1 word, 1 word, 2 words and 3 words around the target word are collected. W-features model the equivalent information for content words.

These features return 1 if a lemma or a content word in a particular location belongs to the set of lemmas or content words in the equivalent position.

B-features and C-features

B-features model the set of lemma collocations found at positions close to the target words. Only sets for collocations found at positions (-2, -1), (-1, 1) and (1, 2) words relative to the target word are collected. C-features are, again, the equivalent of B-features for content words.

These features return 1 in similar conditions to L-features and W-features, when a collocation at a particular location belongs to the set of collocations found at equivalent positions around the target word.

P-features

Lastly, P-features model the set of POS tags found near the target word, at positions -3 words, -2 words, -1 word, 1 word, 2 words and 3 words around the target word.

P-features return 1 when the POS tag found at a particular position around the target word belongs to the set of tags for the equivalent position.

Analysis of Feature Types

Evaluation of the feature types was performed with a selection of 10 nouns and 5 verbs using examples from the DSO corpus (Suárez and Palomar, 2002). Each classifier was trained using 10-fold cross validation and the results for the best combination of feature types was given for each word tested, shown in table 5.4. Work also showed

Word	Senses	Feature types	Accuracy
Age (Noun)	3	SQ	74.3%
Art (Noun)	4	OLWBCP	64.1%
Car (Noun)	2	WSB	96.9%
Child (Noun)	2	LWBCQ	94.5%
Church (Noun)	4	OLWSBCQ	65.4%
Cost (Noun)	3	SCQ	89.7%
Fall (Verb)	6	OLWBCK3	85.9%
Head (Noun)	7	SQ	81.4%
Interest (Noun)	6	OLWSBCQ	68.3%
Know (Verb)	6	OLWSBCQ	48.8%
Line (Noun)	22	OLWBCK3	56.9%
Set (Verb)	11	OLWBCPK3	58.0%
Speak (Verb)	5	SQ	76.2%
Take (Verb)	19	LWSBC	40.8%
Work (Noun)	6	LWBCPK5	51.8%

Table 5.4: Results from (Suárez and Palomar, 2002) for Best Combinations of ME Features

that using combinations of template-set feature functions only produces an average drop in accuracy of 1.75% (0.99% for all words apart from “child” for which the most drastic drop in accuracy occurred). The advantage of these template-set functions is the large reduction in model complexity as fewer feature functions are generated. This

results in a large reduction in computational time necessary for training the ME models. Although results are not given, L-features and W-features are reported to produce highly precise results with low recall, and O-features are deemed to be particularly useful for verbs. Q-features and P-features are reported to favour the most frequent sense of words, at the expense of less frequent senses.

One conclusion made by Suárez and Palomar (2002) is that more examples and deeper syntactic data about sentences are required in order to improve current WSD techniques.

5.4 Gold Standards for WSD Evaluation

Towards the late 1990s efforts were made to create standard evaluation techniques for WSD. Some previous cases can be found where researchers shared corpora and resources, thus allowing results to be compared. However, most approaches to evaluation created ad hoc evaluation platforms using custom corpora with different sense distinctions and in some cases evaluating different aspects of a technique. This was the case even between closely related techniques, such as those techniques stemming from Lesk's "dictionary-overlap" approach (Lesk, 1986), where many researchers decided to use different test sets (Wilks et al., 1990; Véronis and Ide, 1990, 1991, 1995). By far the preferred evaluation technique is to measure the proportion of correct distinctions made, otherwise known as the accuracy of the technique as shown in equation 5.6.

$$Accuracy = 100 \times \frac{C}{N} \tag{5.6}$$

where C is the number of correctly disambiguated words, and N is the total number of words classified.

One problem with evaluating techniques with basic accuracy comes from situations where a system returns probabilities for senses, and the correct sense may be assigned a marginally lower probability to the sense selected via the algorithm. Accuracy does not give credit for near misses. Accuracy also does not account for situations where more than one sense of a word could apply in the same context, for instance consider "give" in "He gave his report to his superior". Given the rather strict interpretation of accuracy above, a number of alternative evaluation metrics have been proposed trying to relax this strict interpretation (Resnik and Yarowsky, 1997). Most contemporary

research still does not make use of these proposals, and the current gold standard for WSD evaluation comes from the SENSEVAL conferences.

Work for SENSEVAL started in 1997, following the workshop "Tagging with Lexical Semantics: Why, What and How?" held at the conference on Applied Natural Language Processing. The goal was to produce and run a test to analyse the strengths and weaknesses of WSD techniques across a number of varying texts in a number of different languages.

Two subsequent SENSEVAL exercises have been run, the first in 1998 and the second held at the Second International Workshop on Evaluating Word Sense Disambiguation Systems in 2001. A third exercise is planned for 2004.

5.4.1 SENSEVAL

The first pilot SENSEVAL experiment produced the essential elements necessary for a gold standard evaluation technique:

- A task definition.
- A 'Gold Standard' dataset. This is defined to be a reproducible corpus with manually labelled senses for each word. For such a corpus to be reproducible, agreement between human annotators must be suitably high, therefore it is necessary that all examples are tagged by at least 2 people. In practice, agreement above 90% between human taggers was deemed as acceptable. Kilgarriff (1998a) discusses this issue in greater detail.
- A framework for administering the evaluation to the highest level of objectiveness. It was proposed that a sample of roughly 200 ambiguous words with manually tagged examples in the corpus should be used as a test set for evaluation. This would be a manageable quantity for human taggers to produce tagged corpora each year (Kilgarriff, 1998b). The words are only released to test systems once they are "frozen" in order to avoid fine-tuning the systems to the test set. Furthermore, in order to compare systems tagging different types of words, for instance only nouns compared to all-words, or systems built using radically different approaches, such as supervised versus unsupervised techniques, considerations must be made to ensure a "level playing field".

The corpus and dictionary used for SENSEVAL were both from the HECTOR project (Atkins, 1993). The first Senseval chose only to test a selection of words, and not to perform an all word evaluation of techniques. In total, 35 words were selected for the evaluation, with 26 to 2,008 instances for 30 of the selected words available in the corpus. The test corpus included 8,448 examples from which 41 tasks (15 nouns, 13 verbs, 8 adjectives and 5 indeterminate examples) were created for the 35 words. A subset of 1,057 corpus entries for 4 words was re-tagged to ensure a gold standard for the corpus. Final agreement of 95% precision between annotators is reported (Kilgarriff and Rosenzweig, 2000a). Mappings from the HECTOR senses to WordNet senses were provided, although the mappings were typically many to many and gaps existed, therefore some information loss between the two resources is inevitable. Given this mapping, an upper bound of 79% was calculated for WordNet based systems by mapping evaluation answers from HECTOR senses to WordNet senses, and back to HECTOR senses and then calculating the agreement of the resulting senses. Kilgarriff and Rosenzweig (2000b) note that given the high frequency of one-to-many relationships between HECTOR and WordNet senses, WordNet based techniques "operate under a severe handicap", and thus comparison of their performance will yield little objective information.

In all, the first SENSEVAL test evaluated 16 English systems, 2 French systems and 1 Italian system. Systems were broadly split into two groups; supervised and unsupervised taggers, and each were tested at 3 different granularity levels:

- Fine-grained – Only tags identically to those assigned by human annotations are classed as correct.
- Mixed-grained – Mixed grained scoring gives full credit if a tagged sense is subsumed by the human judgement, and partial credit is given if it subsumes the human judgement.
- Coarse-grained – Sub-sense tags were ignored, therefore matches with human judgements are taken at a much coarser homograph level.

In the event of systems returning multiple answers the (normalised) probability of the correct answers returned are used as the score value added to the result of evaluation. Two main baseline techniques were also employed (although Kilgarriff and Rosenzweig (2000b) discuss a number of other baselines also considered); Lesk's dictionary

overlap (Lesk, 1986) to compare against unsupervised techniques and a Lesk-Plus-Corpus method employing training examples together with dictionary definitions to compare against supervised techniques. Results for the test are given in terms of precision (5.7) and recall (5.8).

$$Precision = (s/n) \tag{5.7}$$

$$Recall = (s/m) \tag{5.8}$$

where s is the score of the system, n is the number of items classified by the system and m is the number of items with classifications.

The results produced a number of conclusions for the systems tested:

- All systems tested gave improved results for the coarse-grained level compared to the fine-grained sense distinctions and the relative performance of systems tagging at fine-grained levels was equivalent for more coarse-grained sense distinctions.
- Supervised training techniques perform substantially better than unsupervised techniques.
- Few systems outperform their Lesk baseline equivalent.
- The state-of-the-art for automatically disambiguating fine-grained sense distinctions performs at around 77% precision and 82% precision at the coarse-grained level.

The best performing systems from the evaluation were the supervised Durham WSD system (Hawkins) and John Hopkins WSD system (Yarowsky) systems (Kilgarriff and Rosenzweig, 2000a,b).

5.4.2 SENSEVAL-2

Whilst the scoring guidelines remained the same, SENSEVAL-2 introduced some changes for the evaluation approaches of the first SENSEVAL test. Firstly, WordNet was selected as the lexicon to provide the inventory of senses for evaluation, and a corpus was created from a sample of the BNC, the Penn Treebank (Marcus et al., 1994) and from

live web-pages for a web subtask. A newer version of WordNet (1.7) became available, where some changes were made given information resulting from the annotation process of the SENSEVAL-2 resources. The use of WordNet was said to make the task more difficult, highlighted by a 10% lower manual inter-annotator agreement than for the original SENSEVAL.

The original SENSEVAL only evaluated systems using a 45 word lexical selection evaluation. However, SENSEVAL-2 also offered an all-word evaluation where systems had to tag all words in 3 texts providing a total of 5832 running words, and a Japanese to English translation task. A further difference was that SENSEVAL-2 did not provide any manually tagged training data, as systems were expected to use resources from the public domain.

A total of 94 systems ranging across 12 languages were evaluated during the 2001 SENSEVAL-2 workshop. The two best performing systems showed a significant drop to around 64% precision and recall for the fine-grained lexical sample test and around 64%-69% precision and recall for the all-word test, reflecting the difficulty that inter-annotators had in manually tagging the corpus with WordNet senses. The best performing systems for English were hybrid systems from Mihalcea and Moldovan (2000) and Yarowsky (2000), making use of multiple components and a variety of lexical information, such as syntax for the latter system. Baselines also showed a significant drop in accuracy, resulting in many of the systems now being able to surpass their equivalent baseline results. For Lesk-based baselines, this is probably an indication of the differences in the suitability of dictionary definitions between HECTOR and WordNet. As HECTOR typically contains longer definitions for senses, it may provide a better information source for Lesk's approach.

5.4.3 SENSEVAL-3

A further SENSEVAL evaluation is scheduled for 2004, with yet a greater number of groups showing an interest. Few detailed descriptions have been released so far. However, a number of additional tasks are being considered, such as sense labelling WordNet glosses.

5.5 Summary

A number of different phenomena of ambiguity in natural language have posed considerable problems to many NLP tasks, such as MT, IR, Content analysis, Parsing and Speech Processing. From the initial WSD work in the 1950s, the field has undergone a number of changes in approach due to changing trends in NLP research and given the provision of ever growing resources. Given the maturity of the field, successes have been modest to date, and the techniques applied by different researchers can be seen to be ever more divergent. Recent standardisation of evaluation approaches by the Senseval conferences has aided development of the WSD field. Central to Senseval is the provision of Gold Standard evaluation resources, including both material for the creation or training of WSD systems, and material for their evaluation. These resources are in the form of a sense labelled corpus of examples, where the manual inter-tagger agreement is ensured. The aim of the gold standard is to provide corpora with inter-tagger agreement above 90%, meaning that the resources are reproducible by different individuals thus making evaluation more meaningful and objective. Looking at the other available sense tagged corpora, when evidence is available about inter-tagger agreements, it is found that agreement is much lower than for the Senseval resources, for instance the Semcor and DSO corpora have an inter-tagger agreement of 57% (Kilgarriff, 1998a).

Three WSD techniques particularly influential to the research presented in chapter 6 were introduced in section 5.3. The first of the three techniques created a WSD system using a number of partial taggers (Wilks and Stevenson, 1997a,b,c, 1998c; Stevenson and Wilks, 1999, 2000). This approach has the advantage of combining results from several "weak" taggers to only consider the most confident decisions from each tagger. This means that different aspects of a word's context can be used in making a decision about its meaning. The second technique discussed applies a different syntactically-based definition of context to a WSD system (Lin, 1997). A related approach is introduced in chapter 6, however instead of solely considering syntactic relationships, the definition of context considers the semantic role of words (see section 6.2). Such a definition is useful as it targets the words in the surrounding context that are related, thereby avoiding noise introduced by other words in the surrounding context and considering fewer but more related words than a context window definition of context. A third ME-based approach (Suárez and Palomar, 1993, 2002) is introduced to illustrate

how the ME statistical paradigm is applied to the problem of WSD. Particular attention is paid to the design of features, as such features form the basis of the statistical model.

Chapter 6

Word Sense Disambiguation Using Lexical Taxonomies and Syntactic Context

A striking trend of current Word Sense Disambiguation (WSD) techniques discussed in chapter 5 is the variation in the type of information used, in many cases achieving little or no improvement. This variation is due to the use of multiple sources of information publicly available and the variety of linguistic theories which can be exploited in tackling at least part of the WSD problem. This chapter proposes a WSD system employing a number of partial-taggers designed to confidently reduce the number of senses being considered for words in an open-text, before finally making a decision about remaining ambiguities using a statistical WSD component. During this process, when only one sense remains under consideration for a word, the word is said to be sense tagged or sense labelled. The majority of the chapter is dedicated to describing the development of such a final WSD component within the Maximum Entropy (ME) framework. This component uses a new definition of context designed to target the information of interest for disambiguation of a word within its surrounding sentence.

Section 6.1 introduces a new multi-tagger approach to WSD, making use of existing theories. This new framework contextualises the research reported later in the chapter. Given the available time it is not feasible to construct all necessary partial taggers, therefore later sections are restricted to reporting the construction of a new statistical WSD component. Whilst the reported test results evaluate performance in isolation of

further techniques, it is intended that such a WSD system is used as the last stage in a multi-partial-tagger approach. Section 6.2 details the definition of context considered by the statistical WSD component, and section 6.3 shows how such a definition of context with semantic similarity can be used to build a statistical classifier for WSD using the ME framework. Test results are presented which illustrate how current results could be used to reduce the cost of manual WSD. Section 6.4 details possible future work and section 6.5 discusses the significance of this research.

6.1 Using Multiple Partial Taggers for WSD

Given the current accuracy of state-of-the-art WSD techniques, no single technique can deliver the level of performance necessary for high quality WSD of open-texts. High quality in this context is understood as being at least comparable to human inter-tagger agreement. Defining such a baseline has posed a significant problem for WSD researchers, with a low inter-tagger agreement of 57% (Kilgarriff, 1998a) between two of the most used resources for WSD, Semcor (Landes et al., 1998) and DSO (Ng and Lee, 1996). More recently, work for Senseval has produced gold-standard corpora for WSD evaluation, producing around 90% inter-tagger agreement for a sample of the Penn Treebank corpus (Marcus et al., 1993). In order to improve results, Wilks and Stevenson (1997c, 1998c, 1997a) and Stevenson and Wilks (1999, 2000) used multiple partial-taggers to reduce the number of senses under consideration for each word. The final sense is assigned by considering results from each partial-tagger. We propose a similar approach using ideas from Gale et al. (1993), Yarowsky (1993, 1995), and lexical theory to initially reduce the number of senses, coupled with a statistical component to make informed judgements about the remaining senses. The possibility is also available to use further partial-taggers, although consideration must be made about the order in which such taggers are applied. It is desirable to use more precise techniques with the lowest coverage early in the WSD process, with later lower precision techniques giving maximum coverage. Thus the system has maximum confidence about decisions it makes earlier, reducing potential errors by later techniques. By incorporating less confident techniques later in the WSD process, the system can evaluate residual ambiguity once the more confident techniques have been applied and also ensure maximum coverage. The general framework for such a system is illustrated in Figure 6.1. The pre-processing stage tags words with their part-of-speech (POS), and

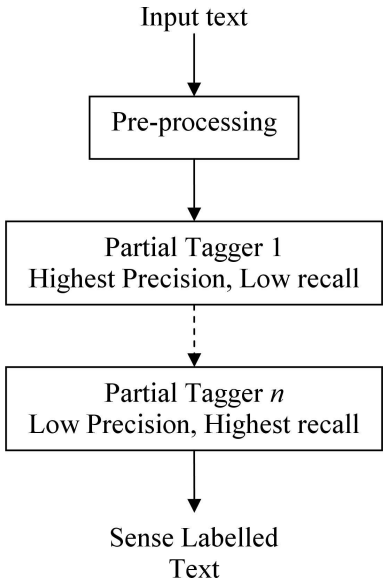


Figure 6.1: General Partial-Tagger WSD Framework

performs any further processing required for the partial-taggers. This initial stage is important not only for preparing documents for processing by subsequent components, but it is also the first stage in restricting senses for the system to consider by assigning POS tags to words (Wilks and Stevenson, 1998a; Towell and Voorhees, 1998).

The framework illustrated in Figure 6.2 shows the collection of techniques proposed as the minimum set of partial-taggers to constitute a complete WSD multi-tagger system. In contrast to the approach taken by Wilks and Stevenson (1997a,b,c, 1998b,c); Stevenson and Wilks (1999, 2000), where partial taggers are used in parallel to each other and the final sense selection is made considering results from each of the taggers, the approach here is more of a pipeline where each tagger incrementally reduces the number of senses being considered. The techniques within the framework are applied in the following manner:

1. One Sense per Collocation

   Gale et al. (1993) and Yarowsky (1993, 1995) create a decision tree based WSD system to tag common word collocations with the same senses, based on the hypothesis that the senses of words in a collocation do not change across different instances. The decision trees are structures used as classifiers for WSD. Each arc

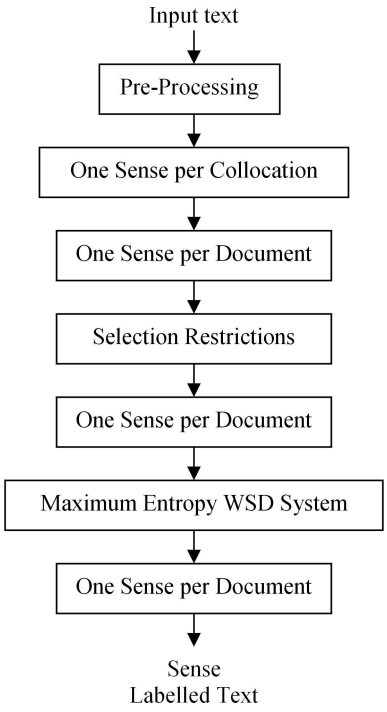


Figure 6.2: Proposed Minimal Set of Partial-Taggers for WSD

in the decision tree represents a decision that can be made given some input stimulus, nodes within the tree represent different stages in the decision making process, and the leaf nodes of the tree represent the final decisions made by the classifier. The work demonstrates 99% precision when tested with words having two senses. However it has been questioned whether this hypothesis holds for fine-grained sense distinctions in dictionaries such as WordNet. Martinez and Agirre (2000) show that the hypothesis does not hold as well across genre and topic variations, presenting results of 70% precision and low coverage with similar corpora tests. They propose using topic and genre information as an input parameter to the decision list in order to tune the results.

This technique (Gale et al., 1993; Yarowsky, 1993, 1995) could be tuned to only consider decision lists producing highly confident results when tested across corpora with topic and genre variations in order to produce an adequate first partial-tagger for the multi-tagger WSD technique.

2. One Sense per Discourse

The one-sense-per-discourse hypothesis (Gale et al., 1992b) assumes that the sense of a word remains highly consistent within a given document. Such a hypothesis is useful for assigning senses to untagged instances of words that have been tagged in other parts of a text, and in correcting errors made by WSD systems within a document. Yarowsky (1995) gives examples of 10 words for which this approach was tested with 37,232 instances. The results of the test gave a total accuracy of 99.8% and showed that the technique was applicable for 50.1% of each word's occurrences (The technique is applicable to words occurring more than once within a document).

Given the accuracy of this approach and the precision of the previous partial tagger, an approach using this hypothesis can tag unlabelled instances of words tagged elsewhere during stage 1. However, some care must be taken. A number of the most frequent and ambiguous words, such as the verb "to be", regularly violate this rule. Prior to such a rule being suitable for use in open-texts, the set of words that consistently violate the one-sense-per-discourse rule must be determined. Given that such words are likely to occur frequently, this should be possible using existing sense tagged corpora. Once this set of exception words has been found, the one-sense-per-discourse rule can then be confidently applied to any word outside the exception set.

3. Using Selection Restrictions to Reduce Senses

   Section 6.2 illustrates a number of ways in which the configuration of verbs and their complements can be described, and how the noun complements of a verb have an important role in the selection of senses for both the verb and nouns. Such information can form the basis of a selection restriction tool. Data to create such a system could come from information in a variety of sources, for instance:

   - Information contained in dictionaries, or from dictionary glosses, although in some cases, such as WordNet, this information can be particularly weak.
   - From resources such as the Levin Verb Classes (LVC) (Levin, 1993). There is currently no link between the LVCs and WordNet synsets. However, a technique has been proposed (Green et al., 2001a,b) that essentially links the two resources in a task to tag verbs in a verb database.

This technique is applied after the one-sense-per-collocation and one-sense-per-discourse techniques in order to make any possible word sense reductions.

4. Repeat One Sense per Discourse.

   Partial-tagger 2 is applied again to make further possible reductions to the document's ambiguity.

5. Final Statistical Sense Discrimination Tool.

   At this stage only one further tagger is considered. The purpose of such a tagger is to make the final decision about the correct sense of a word, given any residual ambiguity. Given the body of work available, a large number of techniques can be used as shown in the previous chapter and by the Senseval work. Currently the best results are attained by supervised learning techniques. The main topic for the remainder of this chapter is a new statistical technique to be used for the final selection of the word sense.

6. Repeat one sense per discourse.

   Finally partial-tagger 2 is applied once more to make any last possible reductions to the document's ambiguity, if any words are left ambiguous. At this stage, the one-sense-per-discourse theory can also be applied to correct erroneously assigned sense tags. By examining each word within the text that are not part of the one-sense-per-discourse exception list, if senses are inconsistent, the most frequently assigned sense tag can be assigned for each instance of a word.

For partial-taggers 3 and 5, an additional pre-processing stage is required. These partial-taggers require syntactic information; therefore a parser is needed to determine the syntactic structure of each sentence in the input text. For the purposes of the research reported in this chapter, the CMU Link Grammar parser is used. This collection of partial-taggers constitutes a minimum set of components for the system proposed, as all but the statistical discrimination tool use existing WSD theory, are simple to implement and have relatively high degrees of confidence in the classifications made. The first 4 taggers do, however, suffer from low recall, and therefore the penultimate tagger is required to make judgements about regarding residual ambiguities in order to maximise recall for the processed documents. The problem currently with the kind of component to be considered for the statistical discrimination tool is its relatively low

precision; therefore the initial taggers are used to remove as many senses as possible in order to reduce errors by the later components. As such, this framework allows for further modules to be added to the system.

A new technique is proposed in section 6.3 for statistical WSD to use the lexical taxonomy of WordNet to evaluate word similarity as presented in chapter 4. The technique also uses a new definition of a word's context utilising semantic relationships between words determined from their syntactic configuration. The remainder of this chapter concentrates on details of such a statistical WSD component, as the implementation of other partial-taggers is outside the scope of this thesis.

6.2 Using Syntactic Relationships for WSD

Fundamental to the statistical classifier for the new WSD technique presented here is the idea that words have semantic relations to other words within a tight context that is central to the human decision making process about the sense of a word (Weaver, 1949; Kaplan, 1955; Koutsoudas and Korfhage, 1955; Masterman, 1961; Choueka and Lusignan, 1985; Preiss, 2001). The most common definition for context used in the field of NLP uses the idea of context windows (Weaver, 1949). Using a context window of size n, the context of a word is represented as the n words to its left and right. Such a definition of context assumes that all words within the context window are important to evaluating the meaning of a word, and also that the significant information for establishing the word sense is contained within the window. The statistical classifier developed in this chapter uses an alternative definition. This alternative definition of context is similar in principle to that used by Lin (1997), Resnik and Diab (2000) and Green et al. (2001a,b). However, it differs in some important aspects. Such a context can be expressed in predicate form. The arguments for such a predicate form for a context can be used to restrict and rank various possible interpretations of a word.

This section presents this new definition of context, using the syntactic features of a sentence to detect relationships between the words within the sentence. These relations are assigned a semantic role according to the syntactic configuration within which they occur. For verbs, these semantic roles are represented as thematic roles. While preliminary examples will be restricted to verbs, section 6.3.2 discusses how relationships for words with other parts-of-speech (POS) can also be expressed in similar predicate forms. The CMU Link Grammar parser is used to determine the syntactic relationships

between words, although such an approach can be applied to the output from parsers generating more traditional Chomskian Sentence Structures. The section concludes by showing a refined version of the information that is used in a WSD classifier.

6.2.1 Sub-categorisation of verbs

Consider the examples of three related CMU linkages in Figures 6.3, 6.4 and 6.5. In

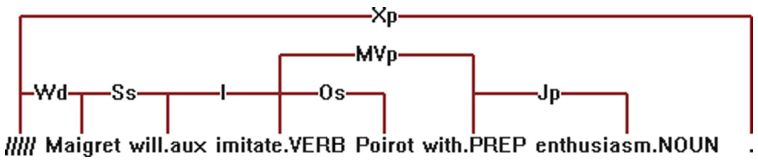


Figure 6.3: CMU Linkage for “Maigret will imitate Poirot with enthusiasm.”

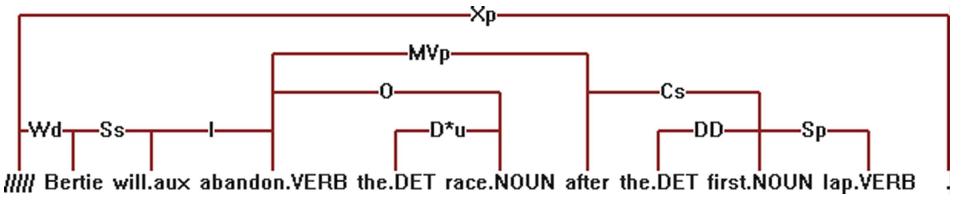


Figure 6.4: CMU Linkage for “Bertie will abandon the race after the first lap.”

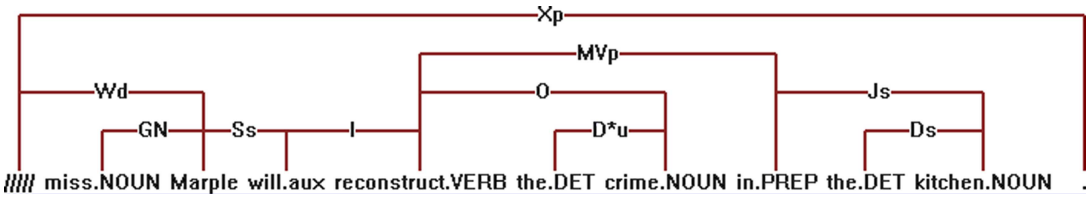


Figure 6.5: CMU Linkage for “Miss Marple will reconstruct the crime in the kitchen.”

each case, the verb shares a common structure. Each example shows a transitive verb with an optional verb modifier. Each of the modifiers provide information about the manner, time or location of the action.

In traditional grammar, verbs are placed into three categories, relating to the number of objects appearing to the right of the verb in an active sentence:

- Intransitive - No object.
- Transitive - One obligatory object.
- Ditransitive - Two obligatory objects expressed either as a pair of nouns, or a noun and the noun of a verb modifier.

Figures 6.6, 6.7 and 6.8 give an example from each sub category. Haegeman (1994)

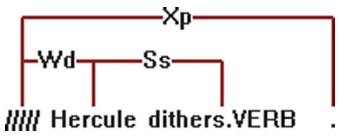


Figure 6.6: Example of an Intransitive Sentence

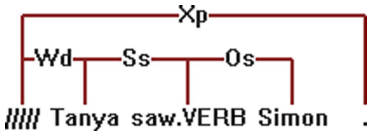


Figure 6.7: Example of a Transitive Sentence

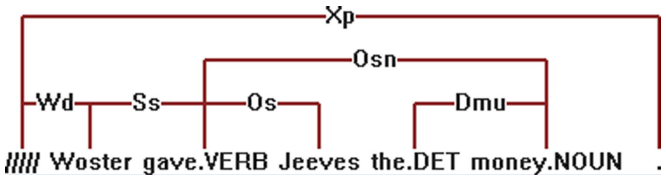


Figure 6.8: Example of a Di-transitive Sentence

gives an in-depth introduction to sub-categorisation with justifications from Chomskian Government and Binding theory. Aside from glosses, this is the only type of information given about the number of arguments for a verb in most common dictionaries, such as the Oxford Dictionary of Concise English.

Sub-categorisation gives some information with respect to the number of required object arguments for a verb. However, nothing is said about a verb's subject or the semantic relationship between the verb and its arguments.

6.2.2 Argument structure

Verbs can be considered as predicates, where nouns or specific prepositional phrases are syntactically related to the verb as arguments. These predicates take a verb's subject into account, along with its objects. Therefore, verbs can be represented by the following type of predicates according to their sub-classification:

- Intransitive verb – One-place predicate
- Transitive verb – Two-place predicate
- Ditransitive verb – Three-place predicate

The previous examples can be expressed in the following predicate forms:

1. Maigret imitates Poirot.

   Imitate(Maigret, Poirot)

2. Bertie abandoned the race.

   Abandon(Bertie, race)

3. Miss Marple reconstructed the crime.

   Reconstruct(Miss Marple, Crime)

4. Hercule is dithering.

   Dithering(Hercule)

5. Wooster gave Jeeves the money

   Give(Wooster, Jeeves, Money)

Argument structures show which arguments are obligatory for a given verb. Haegeman (1994) uses the following metaphor in her description of argument structures:

> "Predicates are like the script of a play. In a script a number of roles are defined and will have to be assigned to actors. The arguments of a predicate are like the roles defined by the script of a play. For an adequate performance of the play, each role must be assigned to an actor." (Haegeman, 1994)

For instance, consider the verb 'give'. In its most literal interpretation, it must take three arguments: The giver, the given and the receiver. Considering the argument structure of a verb derived from its conceptual meaning can enrich the representation. In traditional Chomskian grammar we can express the arguments by specifying the phrasal type to which the arguments belong. However, the following argument structure examples use CMU link labels to specify how the arguments are syntactically expressed. "S" links refer to verb subjects, "O" links refer to verb objects and MV links refer to verb modifiers, such as prepositional phrases ("x" denotes a wild card for sub link information):

1.	Imitate:	verb;	1	2	
			Sx	Ox	
2.	Abandon:	verb;	1	2	
			Sx	Ox	
3.	Reconstruct:	verb;	1	2	
			Sx	Ox	
4.	Dither:	verb;	1		
			Sx		
5.	Give:	verb;	1	2	3
			Sx	Ox	Oxn
			Sx	MVx	Ox

No further information, such as timing, manner or place, about verbs is expressed in their argument structures.

Situations where arguments are optional are expressed by using parentheses around the optional argument number:

Hercule bought Jane a detective story.

Hercule bought a detective story.

Buy: verb; 1 (2) 3

Sx Ox Oxn

WordNet gives some information similar to argument structure at a sense level in its sentence frame relation for verbs. However, rather than showing optional arguments explicitly, it enumerates various allowable structures, for instance the following example of give#5:

```
give, pay -- (convey, as of a compliment, regards,
attention, etc.; bestow; ‘‘Don’t pay him any mind’’;
‘‘give the orders’’; ‘‘Give him my best regards’’;
‘‘pay attention’’)
    *> Somebody ----s something
    *> Somebody ----s somebody something
    *> Somebody ----s something to somebody
```

Whilst this basic representation of arguments is available, the level of information provided by WordNet for verbal sentence frames is limited. Typically, the only argument types found are “somebody” and “something”, making their usefulness limited to establishing that a verb’s argument is either human or non-human.

6.2.3 Thematic structure

We have shown that verbs have an associated argument structure relating to syntactic relations of words in the surrounding context. However, none of the examples shown so far consider information about the semantic role these related words may have. Theta theory (Haegeman, 1994) describes such semantic information in terms of thematic roles or theta roles (θ-roles), refining the relationships between a verb and each of its arguments. These assignments can replace argument structures with thematic structures.

The importance of thematic structures is widely recognised, but as yet a standard theory has still not been agreed. Different linguists define different sets of θ-roles. At one level, a linguist may choose labels which give very specific relationships between two words, such as in “John gave Mary some flowers”, “John” can be labelled as the *giver*, “Mary” as the *receiver*, and “flowers” are the *given* item. For the purpose of the work presented here, a much more general definition with only a small number of

θ-roles is considered. Such a definition restricts the possible relationships to one of the following roles across all verbs:

- Initiator
- Goal
- Essence

For the previous example, "John" is the *initiator* of the action, "Mary" is the *goal* of the action, and "flowers" are the *essence* of the action.

As shown, the use of θ-roles considered for this work is rather restricted. The main reason for considering the θ-roles of verbal arguments will be to generalise information across different argument structures. Further details are given in section 6.3.2.

6.2.4 Context Features

The remainder of the chapter shows how a WSD system can use these ideas as the basis of a definition of context for a statistical model. The main purpose for the statistical classifier is to use a weight assigned to the importance of each component of the context during the disambiguation process for a given word. Such a component of context is referred to as a feature or statistical feature. Selecting these features is the most important part of the creation of a statistical classifier. Once features are available, their weights are trained according to a corpus of training examples.

6.3 A New Statistical Technique for WSD

Many tasks in Natural Language Processing (NLP) have been tackled using stochastic models which capture information about some phenomenon or behaviour of interest. Using such statistical models for WSD has provided promising results, but they still suffer from the lexical bottleneck problem (Ide and Véronis, 1998) as there are limited numbers of publicly available sense tagged corpora.

The WSD technique presented in this section is used in a Maximum Entropy (ME) framework that exploits the previously defined notion of context (section 6.2) and semantic similarity (chapters 3 and 4) to alleviate the lexical bottleneck problem.

6.3.1 Maximum Entropy

ME is a statistical technique that is becoming increasingly popular in NLP in tasks such as machine translation (MT) (Berger et al., 1996), part-of-speech (POS) tagging (Ratnaparkhi, 1996), text segmentation (Beeferman et al., 1999) and more recently WSD (Suárez and Palomar, 1993, 2002; Dang and Palmer, 2002; Klein et al., 2002). Such classifiers provide a way to make use of contextual information to estimate the probability of a classification, such as the linguistic class of interest for a word. The idea of ME can be dated back to pre-biblical times in the writings of Herodotus (425-485 BC) (Berger et al., 1996), but only recently has enough computational power been available for maximum entropy to be used effectively. The goal of ME is to create a classifier that will select the most likely possibility (given the context some problem lies in), *without assuming anything* about information that is not available at the time of training. Furthermore, ME techniques are used when the source of information for the model is known to be sparse, and thus where only estimates of the probabilities of certain classifications are available. Therefore the problem is to find a statistical distribution, $p(d, c)$ where d is a decision and c is a context, that can be used as a classifier which maximises entropy, or uncertainty, subject to constraints that represent evidence used in the decision making process.

This section illustrates the use of ME to translate the English word "in" to its French alternative. Section 6.3.2 introduces and justifies the set of features and classifier for use in a WSD framework. Finally, these ideas are preliminarily tested to assess their potential for WSD.

Basic Probability

The main interest in many NLP tasks is to characterise some linguistic phenomena, such as the determining meaning or word translation of a word, so that it closely matches human judgements. A common approach currently involves training a statistical system with judgements that have been manually collected from various sources, such as annotated text corpora or recordings. Such systems are then tested with new examples to check that they generalise sufficiently to handle future examples. This is commonly achieved by reducing the original problem to one of estimating the probabilities of a finite set of possible classifications in order to find the most likely classification.

The most basic probability to start from is that for a known situation, where the sum of the probability of all known examples must be 1.

$$\sum_{e \in E} p(e) = 1 \tag{6.1}$$

where E is the finite set of all examples. Equation 6.1 defines the first constraint for a classifier. Using the example from Berger et al. (1996), we can start to construct a classifier, p, which given the English word "in" will predict its French translation.

The English word "in" has five alternative translations in French; "dans", "pandant", "en", "à" and "au cours de". Given 6.1, 6.2 must hold.

$$p(\text{dans}) + p(\text{pandant}) + p(\text{en}) + p(\text{à}) + p(\text{au cours de}) = 1 \tag{6.2}$$

An infinite number of classifiers can be derived which meet the constraint imposed by 6.2. The Principle of ME recommends that probabilities be assigned in the *most non-committal fashion*. Where no empirical evidence is available, probabilities should be assigned without making any further assumptions regarding the distribution of the data, and therefore probability is assigned as uniformly as possible (Guiasu and Shenitzer, 1985). This reduces bias that could arise in the classifier. Given no further information about the translation of "in", and given that French has no further possibilities for the translation of "in", the most non-committal distribution is shown in table 6.1. As Berger et al. (1996) notes, this is not the most uniform model as that would grant equal probability to all French words.

Translation, f	$p(f)$
dans	$1/5$
en	$1/5$
à	$1/5$
au cours de	$1/5$
pendant	$1/5$
Total	1

Table 6.1: Most uniform distribution for the translation of "in".

The availability of empirical evidence about human decisions allows a more complex model to be created. For instance, equation 6.3 shows a constraint about the

frequency of "in" being translated to either "dans" or "en".

$$p(\text{dans}) + p(\text{en}) = 3/10 \tag{6.3}$$

Given 6.3, the most non-committal distribution is shown in table 6.2. Adding the

Translation, f	$p(f)$
dans	$3/20$
en	$3/20$
à	$7/30$
au cours de	$7/30$
pendant	$7/30$
Total	1

Table 6.2: Most uniform distribution for the translation of "in" given constraint 6.3.

further constraint 6.4 will, however, make the selection of a suitable classifier less obvious.

$$p(\text{dans}) + p(\text{à}) = 1/2 \tag{6.4}$$

Now the problem is to distribute probability evenly across the classifier, $p(f)$, but this is no longer trivial. In order to solve this, a way of measuring the uniformity of a classifier is required, so that a classifier can be found that maximises uniformity subject to any constraints that apply to the classifier.

ME calculates a statistical classifier that has maximum ignorance about anything outside the body of evidence with which it is supplied, i.e. the classifier assumes nothing about what is unknown at the time of training. For the simple examples, solutions are given in tables 6.1 and 6.2. However, it can be seen that increasing constraints rapidly increases complexity, and that selecting a suitable classifier soon becomes more than a trivial task.

Features and Context

Basic probability alone is too simple to produce a useful classifier for most tasks. It is impossible to predict the best classifications in most problems without considering information other than the statistical distribution of the classification's behaviour. A common practice in NLP is to consider the context surrounding the behaviour of the

terms being modelled (Weaver, 1949). This is normally limited to considering the characteristics of words directly to the left and right of the word of interest, referred to as a context window. ME models individual components of context that are of interest as *features*. Such features help predict the behaviour of interest. Therefore, we now consider the problem of calculating the probability of some decision d, given some context c by making use of information calculated for a set of features.

The first task in constructing such a classifier is to select a set of suitable features from the context surrounding a word. In the example by Berger et al. (1996), context is defined as the words directly surrounding "in". To select features, a large corpus of phrases containing the word "in", together with their French translation is used. Given this corpus, the empirical probability of contexts and classifications (or decisions), can be calculated using equation 6.5.

$$\tilde{p}(c,d) \equiv \frac{\sum_{(c,d)\in Sample} 1}{N} \tag{6.5}$$

where N is the number of examples in the sample, c is a context and d is a decision. Selecting a feature set for some classifier involves choosing a set of contexts and decisions that are significant in the decision making process. As an initial estimate, all contexts and decisions could be used to generate the feature set, although techniques do exist for automatically generating feature sets from corpus examples, as discussed in section 6.3.2. Features are considered as binary functions, the feature indicators, of the form shown in equation 6.6.

$$f(c,d) = \begin{cases} 1 & : \text{ if } d \text{ is the decision for context } c \text{ given some constraint} \\ 0 & : \text{ otherwise} \end{cases} \tag{6.6}$$

A feature can be interpreted as "d is a valid decision given some context c".

In the example of translating the word "in" into French, the training sample shows that if "April" follows "in", the translation of "in" is "en" 9/10 times. The feature indicator for such information is represented by feature 6.7.

$$f(c,d) = \begin{cases} 1 & : \text{ if } d = \text{"en" and "April" follows "in" in } c \\ 0 & : \text{ otherwise} \end{cases} \tag{6.7}$$

So, for each feature that is known to be significant, a binary feature indicator is intro-

duced to model it.

Given these features, the probability that some feature f was seen in the empirical data can be calculated with equation 6.8.

$$\tilde{p}(f) \equiv \sum_{c,d} \tilde{p}(c,d) f(c,d) \tag{6.8}$$

where $\tilde{p}(c,d)$ is the probability that the decision d and context c co-occur in the empirical data. Applying Bayes rule means that 6.8 can be re-written as 6.9.

$$\tilde{p}(f) \equiv \sum_{c,d} \tilde{p}(c)\tilde{p}(d|c) f(c,d) \tag{6.9}$$

Given that the final statistical classifier used must accurately reflect the known facts, the constraint shown by equation 6.10 must be true.

$$p(f) = \tilde{p}(f) \tag{6.10}$$

where $p(f)$ is the probability of a feature being active as calculated using a statistical classifier. With this constraint, the probability of a feature occurring is now calculated using equation 6.11.

$$p(f) \equiv \sum_{c,d} \tilde{p}(c) p(d|c) f(c,d) \tag{6.11}$$

This allows the possibility of automatically calculating the conditional probability, $p(d|c)$, to generalise the statistical model given by the empirical distribution, $\tilde{p}(d|c)$, but in such a way as to still conform to the distribution in the training sample. $p(d|c)$ forms the basis of the classifier in the final classification system.

The Maximum Entropy Framework

A training sample of data yields information about the decisions made within different contexts; however this only accounts for a small portion of all possible situations due to the sparse nature of the data for the task being modelled. The task of ME is to train a classifier, $p(d|c)$, that conforms to the empirical distributions of the training sample but in addition remains as uniform as possible for all other possibilities. Given information about how features affect decisions made in the test data, the task is to find a classifier that uses these features to calculate $p(d|c)$. That is to say, the principal of maximum

entropy is:

> "To select a model from a set C of allowed probability distributions, choose the model $p_* \in C$ with maximum entropy $H(p)$."

$$p_* = \arg\max_{p \in C} H(p) \tag{6.12}$$

where $H(p)$ is the measure of uniformity. Berger et al. (1996) give the mathematical measure of conditional entropy as a measure of the uniformity of $p(d|c)$, as shown in equation 6.13.

$$H(p) = -\sum_{c,d} \tilde{p}(c) p(d|c) \log p(d|c) \tag{6.13}$$

To ensure that the classifier will conform to the information about the features, the set C of allowable classifiers are defined by equation 6.14.

$$C \equiv \{p \in P | p(f_i) = \tilde{p}(f_i) \wedge i \in \{1, 2, \ldots, n\}\} \tag{6.14}$$

where P is the set of all possible models and n is the number of features used by the classifiers.

A classifier, $p(d|c)$, is constructed using the features collected from a training sample. Berger et al. (1996) and Berger (1997) give a method using Lagrange multipliers from the theory of constrained optimisation to train a ME classifier. For each feature, f_i, a Lagrange multiplier, λ_i, is introduced. The Improved Iterative Scaling (IIS) algorithm (Berger et al., 1996; Berger, 1997) trains the Lagrange multipliers for each feature until p_* is found. The resulting classifier can be used to disambiguate new examples using the formula in equation 6.15.

$$classification(c) = \max_{d \in decisions(c)} p(d|c) \tag{6.15}$$

where $decisions(c)$ is the set of possible decisions for the word being evaluated in context c. Appendix E gives further detail about the IIS algorithm and ME framework.

6.3.2 WSD with ME

ME has been applied, with some success, to the field of MT (Berger et al., 1996), amongst many other NLP fields (Ratnaparkhi, 1998). For such a task, ME classifiers

are trained using information from bilingual corpora. Part of the task of ME classifiers is to select the correct word in the target language with the same meaning as the word in the source text. This is similar, in principle, to selecting the correct sense for a word, given a finite number of senses from which to select. Some of the difficulty in directly applying such approaches to WSD is due to the lack of sense tagged examples from which to train, but also in part to the fine-grained nature of lexical resources used to select word senses. The following section describes how the ME framework can be used to produce a classifier for WSD, reflecting the distribution of the information crucial to the WSD task, using the approach described in the previous section. We also show how word similarity techniques using WordNet's taxonomy can generalise the model further. The WSD system produced differs to other ME based WSD systems in that a new definition of context is used, based on the syntactic configuration of a word within a sentence, and because semantic similarity is used to match words.

The main task is to define a ME classifier, $p(d|c)$, to model the human decision making process in selecting the correct meaning of a polysemous word within a context. The reason for choosing ME as the framework from which to produce a statistical classifier is that we can regard the Lagrange multipliers assigned to features as weights indicating their importance during the process of WSD. These features reflect individual aspects of the context in which a word appears. The resulting classifier estimates the probability, $p(w\#s|c)$, that the sense s of a word w was intended for the local context c. Before we can proceed we must clearly define what is meant by context and with this in mind define the set of feature templates that will be used to produce features for the classifier.

Context

The definition of context introduced here is based on the notion that syntax plays an important role in classifying the meanings of words (Reifler, 1955; Towell and Voorhees, 1998). Therefore, for any given word in a sentence, context is defined as those other words in the sentence which are deemed to be syntactically related, an approach similar to that taken by Gougenheim and Michéa (1961). This is contrary to a large body of previous work using ME WSD classifiers whose definitions of context use windows of words surrounding the word for which the context belongs. Tests have shown that the optimum context window size for computation systems is typically of size 3, therefore

the context of a word typically includes 3 content words to the left and right of the word in question. This definition assumes that related words directly surround each other. However, in practice it is found that some of the most importantly related words can be separated by large numbers of unrelated words, such as when related words are separated by a relative clause. Considering the syntactic structure of a phrase, context can be defined that is both compact, and contains only related words. Such a definition of context may also model better the human behaviour to make judgements about the meanings of words within a very limited amount of information (Reifler, 1955).

Earlier evidence showed that the subject and objects of a verb play an important role in understanding its meaning, but now the idea must be extended to consider other syntactic relationships. Such syntactic relationships are in turn labelled with the particular functional or semantic role they represent within their context. For instance, consider the following example:

"John gave Mary flowers."

The example sentence produces the CMU linkage in Figure 6.9. Using θ-theory to

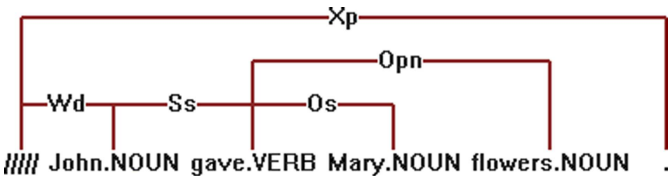


Figure 6.9: CMU Linkage for "John gave Mary flowers"

label the syntactic links to the verb "give", we can produce the local context in 6.16.

$$[\text{ambiguous\_word}(\text{give}), \text{initiator}(\text{man}, 1), \text{goal}(\text{woman}, 1), \text{essence}(\text{flower}, 1)] \tag{6.16}$$

In the example local context, the numbers represent the corresponding WordNet word senses, and the labelled relationships are represented by predicates containing word senses as arguments. For verbal arguments, this labelling is reasonably straightforward as the θ-roles of the noun complements can be automatically detected from the verb's syntactic configuration. Notice also that the local context includes the main word of interest, labelled as the ambiguous word. To produce the full local context of a word we

collect information about all the words of interest that are syntactically linked to that word. Other information may also be of use, such as information about parts-of-speech of surrounding words, and word co-occurrence as tested by other ME WSD techniques (Suárez and Palomar, 2002). However, for the purposes of this work, only information collected from the syntactic relation of words is considered.

For further types of syntactic relationships we must define what the relationship is. In many cases the relationship we define merely substitutes for the syntactic relation. However, in other cases different syntactical relationships are treated as equivalent and therefore are assigned the same type of relationship. In order to define the relationships, all the possible links available that are deemed to be informative for WSD must be considered. When looking at such links in detail, it is helpful to concentrate on information particular to individual parts-of-speech. The following two sections detail the relationships currently defined for verbs and nouns. As an initial step to defining the contexts of adverbs and adjectives, the inverse of the relationships where they appear below could be used. This is currently untested and would typically only yield one relationship in the context for either adjectives or adverbs, most frequently the noun or verb they are syntactically associated with. Local contexts only consider the canonical form of words according to WordNet.

Verb Context Constituents: We have already shown an example of creating a verb context using subject and object links. Given the syntactic configuration of the verb's arguments, the theta role of the verb are detected automatically. Table 6.3 shows for some CMU link configurations the interpretation which determines the constituent of

Syntactic Link	Context Relationship
Noun -S*- Verb	<theta_role>(Noun, Noun_Sense)
Verb -O*- Noun	<theta_role>(Noun, Noun_Sense)
Adverb -E- Verb	verb_attribute(Adverb, Adverb_Sense)
Verb -MV- Preposition -J*- Noun Verb -P- Preposition -J*- Noun	action_is_<Preposition>(Noun, Noun_Sense)
$Noun_1$ -S*- aux -P*- Verb -MVp- "by" -J*- $Noun_2$	essence($Noun_1$, $Noun_1$_Sense) and initiator($Noun_2$, $Noun_2$_Sense)

Table 6.3: Verb Context Constituents

a local context. A complete local context is created by considering all possible constituents in a sentence syntactically related to a word.

Also, CMU uses "B*" links to link back to nouns used outside the scope of the argument structure for the verb under consideration. These links are evaluated to produce constituents of the type "<theta_role>(Noun, Noun_Sense)" depending on the configuration of the noun and verb relative to each other.

Noun Context Constituents: Similarly to the verb context constituents, table 6.4 shows for some CMU link configurations the equivalent interpretation to make a constituent of a local context.

Syntactic Link	Context Relationship
Determiner -D*- Noun	determiner(Determiner)
Adjective -A- Noun	attribute_of(Adjective, Adjective_Sense)
Noun -S*- Verb	has_<theta_role>(Verb, Verb_Sense)
Verb -O*- Noun	has_<theta_role>(Verb, Verb_Sense)
$Noun_1$ -AN- $Noun_2$ or $Noun_2$ -Mp- "of" -J*- $Noun_1$	modifier($Noun_1$, $Noun_1$_Sense) and modified($Noun_2$, $Noun_2$_Sense)
$Noun_1$ -Mp- "in" -J*- $Noun_2$	in($Noun_1$, $Noun_1$_Sense) and contain($Noun_2$, $Noun_2$_Sense)
$Noun_1$ -Mp- "on" -J*- $Noun_2$	on($Noun_1$, $Noun_1$_Sense) and hold($Noun_2$, $Noun_2$_Sense)
$Noun_1$ -Mp- "to" -J*- $Noun_2$	to($Noun_1$, $Noun_1$_Sense) and to_rev($Noun_2$, $Noun_2$_Sense)
$Noun_1$ -Mp- "under" -J*- $Noun_2$ or $Noun_2$ -Mp- "over" -J*- $Noun_1$	under($Noun_1$, $Noun_1$_Sense) and over($Noun_2$, $Noun_2$_Sense)
Verb -MV- Preposition -J*- Noun or Verb -P- Preposition -J*- Noun	done_<Preposition>(Verb, Verb_Sense)

Table 6.4: Noun Context Constituents

Feature Templates

ME classifiers make use of a set of statistical features, collected prior to training the classifier, in order to predict the statistical distribution of a given data set. The features typically reflect individual, or combinations of, constituents of local contexts, therefore the features for the ME classifiers use information contained in the components

of context described in the previous section. The simplest form of feature for the context definition would match words given word-form; however the features used here go further by matching words using semantic similarity. This is achieved by considering WordNet's lexical taxonomy using the techniques described in chapter 4. This is discussed later in this section.

In order to generate features from some source, feature templates are used to collect specific information of interest. Using the basic layout of a feature, open slots defined in the feature templates are filled using relevant information extracted from examples. For instance, the following list of equations gives the features templates for handling verbs and their nominal arguments.

$$f(c,d) = \begin{cases} 1 : & \text{if the ambiguous word in } c \text{ is a verb } v \wedge \\ & \beta(v, d, <\text{Feature\_Verb}>, <\text{Feature\_Verb\_Sense}>) \wedge \\ & c \text{ contains an initiator } i \text{ with sense } i_s \wedge \\ & \nu(i, i_s, <\text{Feature\_Initiator}>, <\text{Feature\_Initiator\_Sense}>) \\ 0 : & \text{otherwise} \end{cases} \quad (6.17)$$

$$f(c,d) = \begin{cases} 1 : & \text{if the ambiguous word in } c \text{ is a verb } v \wedge \\ & \beta(v, d, <\text{Feature\_Verb}>, <\text{Feature\_Verb\_Sense}>) \wedge \\ & c \text{ contains a goal } g \text{ with sense } g_s \wedge \\ & \nu(g, g_s, <\text{Feature\_Goal}>, <\text{Feature\_Goal\_Sense}>) \\ 0 : & \text{otherwise} \end{cases} \quad (6.18)$$

$$f(c,d) = \begin{cases} 1 : & \text{if the ambiguous word in } c \text{ is a verb } v \wedge \\ & \beta(v, d, <\text{Feature\_Verb}>, <\text{Feature\_Verb\_Sense}>) \wedge \\ & c \text{ contains an essence } e \text{ with sense } e_s \wedge \\ & \nu(e, e_s, <\text{Feature\_Essence}>, <\text{Feature\_Essence\_Sense}>) \\ 0 : & \text{otherwise} \end{cases} \quad (6.19)$$

where $\nu(n_1, n_{s1}, n_2, n_{s2})$ is a boolean function which is true when noun sense $n_1\#n_{s1}$ is similar to $n_2\#n_{s2}$ and $\beta(v_1, v_{s1}, v_2, v_{s2})$ is a boolean function which is true when verb sense $v_1\#v_{s1}$ is similar to $v_2\#v_{s2}$. Further templates are used to create the features pertaining to other components of context as defined in the previous section. Additional to such templates which gather information from local contexts, a further set of features models the distribution of senses for a word. Such features are introduced to reduce potential problems when using semantic similarity to match words, because multiple similar words may distribute senses differently. Feature templates for such features are

of the form illustrated by 6.20.

$$f(c,d) = \begin{cases} 1 & : \begin{array}{l}\text{if the ambiguous word in } c \text{ is } <\text{Feature\_Word}>\wedge \\ d =< \text{Feature\_Word\_Sense} >\end{array} \\ 0 & : \text{otherwise} \end{cases} \tag{6.20}$$

These features are referred to as distribution features.

The example from the previous section is used to illustrate how a set of features is created for it. It was shown that the sentence "John gave Mary flowers." produces the local context 6.21 for "give".

$$[\text{ambiguous\_word}(\text{give}), \text{initiator}(\text{man}, 1), \text{goal}(\text{woman}, 1), \text{essence}(\text{flower}, 1)] \tag{6.21}$$

Now, further local contexts are considered for all other content words in the sentences, as shown in equations 6.22, 6.23 and 6.24. These additional local contexts model "give" as having sense 8 (give#8) according to WordNet 1.6.

$$[\text{ambiguous\_word}(\text{man}), \text{has\_initiator}(\text{give}, 8)] \tag{6.22}$$

$$[\text{ambiguous\_word}(\text{woman}), \text{has\_goal}(\text{give}, 8)] \tag{6.23}$$

$$[\text{ambiguous\_word}(\text{flower}), \text{has\_essence}(\text{give}, 8)] \tag{6.24}$$

Using feature templates, such as those described above, features 6.25 to 6.34 are extracted from the local contexts generated from the example.

$$f(c,d) = \begin{cases} 1 & : \begin{array}{l}\text{if the ambiguous word in } c \text{ is a verb } v \wedge \beta(v, d, \text{give}, 8)\wedge \\ c \text{ contains an initiator } i \text{ with sense } i_s \wedge \nu(i, i_s, \text{man}, 1)\end{array} \\ 0 & : \text{otherwise} \end{cases} \tag{6.25}$$

$$f(c,d) = \begin{cases} 1 & : \begin{array}{l}\text{if the ambiguous word in } c \text{ is a verb } v \wedge \beta(v, d, \text{give}, 8)\wedge \\ c \text{ contains a goal } g \text{ with sense } g_s \wedge \nu(g, g_s, \text{woman}, 1)\end{array} \\ 0 & : \text{otherwise} \end{cases} \tag{6.26}$$

$$f(c,d) = \begin{cases} 1 & : \begin{array}{l}\text{if the ambiguous word in } c \text{ is a verb } v \wedge \beta(v, d, \text{give}, 8)\wedge \\ c \text{ contains an essence } e \text{ with sense } e_s \wedge \nu(e, e_s, \text{flower}, 1)\end{array} \\ 0 & : \text{otherwise} \end{cases} \tag{6.27}$$

$$f(c,d) = \begin{cases} 1 : & \text{if the ambiguous word in } c \text{ is a noun } n \wedge \nu(n,d,\text{man},1) \wedge \\ & c \text{ contains a verb } v \text{ with sense } v_s \text{ with } n \text{ as its initiator} \wedge \\ & \beta(e,e_s,\text{give},8) \\ 0 : & \text{otherwise} \end{cases} \quad (6.28)$$

$$f(c,d) = \begin{cases} 1 : & \text{if the ambiguous word in } c \text{ is a noun } n \wedge \nu(n,d,\text{woman},1) \wedge \\ & c \text{ contains a verb } v \text{ with sense } v_s \text{ with } n \text{ as its goal} \wedge \\ & \beta(e,e_s,\text{give},8) \\ 0 : & \text{otherwise} \end{cases} \quad (6.29)$$

$$f(c,d) = \begin{cases} 1 : & \text{if the ambiguous word in } c \text{ is a noun } n \wedge \nu(n,d,\text{flower},1) \wedge \\ & c \text{ contains a verb } v \text{ with sense } v_s \text{ with } n \text{ as its essence} \wedge \\ & \beta(e,e_s,\text{give},8) \\ 0 : & \text{otherwise} \end{cases} \quad (6.30)$$

$$f(c,d) = \begin{cases} 1 : & \text{if the ambiguous word in } c \text{ is the verb ``give'' } \wedge d = 8 \\ 0 : & \text{otherwise} \end{cases} \quad (6.31)$$

$$f(c,d) = \begin{cases} 1 : & \text{if the ambiguous word in } c \text{ is the noun ``man'' } \wedge d = 1 \\ 0 : & \text{otherwise} \end{cases} \quad (6.32)$$

$$f(c,d) = \begin{cases} 1 : & \text{if the ambiguous word in } c \text{ is the noun ``woman'' } \wedge d = 1 \\ 0 : & \text{otherwise} \end{cases} \quad (6.33)$$

$$f(c,d) = \begin{cases} 1 : & \text{if the ambiguous word in } c \text{ is the noun ``flower'' } \wedge d = 1 \\ 0 : & \text{otherwise} \end{cases} \quad (6.34)$$

Sources for Generating Features: A number of possible sources of information to create features are available. Ideally, the source of such information will give direct access to a wide variety of examples without generating excessive quantities of features. The first two of these sources have already been mentioned in the context of providing potential information for creating selection restrictions, although their limitations become more problematic for use in generating statistical features:

- WordNet

  WordNet offers both sentence frames for verbs and gloss examples for its synsets which potentially provide sources for generating features. However, the information currently available in WordNet is not adequate to be used for this task. Firstly, the sentence frames are too general to produce useful information, and

the glosses are fairly ad-hoc and do not give a rich enough source of information to create an adequate set of features. On a positive note though, if adequate information were available, then this information would already be related to word senses and thus would provide a direct source of information for generating features. Work is in progress to provide improved sentence frame information (Baker et al., 1998).

Other lexical resources can be considered. However, a link is required to WordNet's senses in order for the resource to be useful.

- Levin Verb Classes (LVC)

  The LVCs offer a source of information for different classes of verbs and their possible complements. It can provide a wide coverage set of features. However, there are doubts about whether the information is varied enough to be useful for a ME WSD system. A further problem is that the concepts of LVCs verbs and WordNet are not yet related, although work by Green et al. (2001a,b) may provide the necessary information to link the two resources.

- Corpus examples

  Using examples from a pre-tagged corpus can retrieve a varied set of examples to use as a feature set for a ME WSD classifier. In practice, using all available examples to produce a feature set is not desirable for two reasons; Firstly, too many examples introduce additional problems as some examples may contradict each other, or the sheer number of examples could deteriorate precision. Secondly, the high number of features introduces unnecessary complexity into a statistical model, thus increasing the time required to train and to calculate results from the classifier for new examples.

Similarity Relations: In order to match words and word senses using semantic similarity, the features use four similarity relations, one for each part-of-speech, that match concepts via information about WordNet's lexical taxonomy. The similarity relation for nouns uses techniques described in chapter 4. However, no relations have been defined previously for verbs, adjectives and adverbs. In order to handle the remaining parts-of-speech, a set of simple heuristics is used. When selecting adequate heuristics,

care was taken to use only the strictest relationships in order to ensure reliable results. The following section details the heuristics used for each similarity relation.

Noun Relations: Chapter 4 details a number of different parameterised methods for calculating semantic similarity between two noun senses based upon the hypernym structure of both senses. For the purposes of this work, and given the results from chapter 4 for the cross lexicon labelling, $SBSM_{\times 5}$ is used with non-flattened layman structures and normalised results. In order to use this measure to produce a boolean result for whether or not two senses are similar, a threshold value of 0.19296, taken from results for labelling Wordsmyth thesaurus entries, is used. Therefore, the similarity relation for two nouns is given in 6.35.

$$SimNouns(n_1, n_{s1}, n_2, n_{s2}) \iff SBSM_{\times 5}(n_1, n_{s1}, n_2, n_{s2}) \geq 0.19296 \quad (6.35)$$

where n_1 and n_2 are nouns, and n_{s1} and n_{s2} are sense labels for n_1 and n_2 respectively.

Verb Relations: For verbs, the heuristic in 6.36 is used to check WordNet's lexical taxonomy for the existence of some specific relationships.

$$SimVerbs(v_1, v_{s1}, v_2, v_{s2}) \iff \begin{array}{l} synonym(v_1, v_{s1}, v_2, v_{s2}) \vee \\ hypernym(v_2, v_{s2}, v_1, v_{s1}) \vee \\ antonym(v_1, v_{s1}, v_2, v_{s2}) \end{array} \quad (6.36)$$

where v_1 and v_2 are verbs, and v_{s1} and v_{s2} are sense labels for v_1 and v_2 respectively, $synonym(v_1, v_{s1}, v_2, v_{s2})$ is true only if $v_1\#v_{s1}$ and $v_2\#v_{s2}$ are synonyms, the directional relationship $hypernym(v_1, v_{s1}, v_2, v_{s2})$ is true only if $v_1\#v_{s1}$ is a hypernym of $v_2\#v_{s2}$, and $antonym(v_1, v_{s1}, v_2, v_{s2})$ is true only if $v_1\#v_{s1}$ and $v_2\#v_{s2}$ are antonyms.

Adjective Relations: As with verbs, similar adjectives are detected using a heuristic, illustrated in 6.37.

$$SimAdjectives(a_1, a_{s1}, a_2, a_{s2}) \iff \begin{array}{l} synonym(a_1, a_{s1}, a_2, a_{s2}) \vee \\ antonym(a_1, a_{s1}, a_2, a_{s2}) \vee \\ pertainym(a_1, a_{s1}, a_2, a_{s2}) \end{array} \quad (6.37)$$

where a_1 and a_2 are adjectives, and a_{s1} and a_{s2} are sense labels for a_1 and a_2 respectively, and $pertainym(v_1, v_{s1}, v_2, v_{s2})$ is true only if $v_1\#v_{s1}$ and $v_2\#v_{s2}$ are pertainyms.

Adverb Relations: Finally, similar adverbs are detected by heuristic 6.38.

$$SimAdverbs(ad_1, ad_{s1}, ad_2, ad_{s2}) \iff \begin{array}{l} synonym(ad_1, ad_{s1}, ad_2, ad_{s2}) \vee \\ antonym(ad_1, ad_{s1}, ad_2, ad_{s2}) \vee \\ pertainym(ad_1, ad_{s1}, ad_2, ad_{s2}) \end{array} \quad (6.38)$$

where ad_1 and ad_2 are adverbs, and ad_{s1} and ad_{s2} are sense labels for ad_1 and ad_2 respectively.

Feature Reduction

Once an initial set of potential features is available, this set should be reduced to consist only of those features useful for WSD. Using all these initial features is likely to produce a model over-trained to the training examples from which the features were extracted. Four different approaches are proposed for reducing features:

- A manual feature reduction process.
- A mathematical feature induction approach.
- Two linguistically based feature reduction approaches.

Of the least practical solutions proposed is the manual feature selection process. The manual process involves testing a trained ME classifier and analysing erroneous results. Analysis of the features used in the calculation of erroneous results may show that some features are unhelpful to the WSD process, and may also show some contradicting features creating difficulties for the WSD process. For instance, in the tests presented in section 6.3.3, it can be seen that for the verb "give", the initiator from many senses is "person#1". As such, this evidence is unhelpful to the WSD process as it does not discriminate between senses of the verb "give". In fact, as later senses of "give" only have examples for which "person#1" is the initiator, these senses are generally preferred when little additional evidence is available in a word's context. This is clearly an undesirable situation, therefore such features should be manually removed.

An established mathematical technique for inducing features from a set of potential features is given by Berger et al. (1996) and Pietra et al. (1995, 1997). The method they

propose captures the salient properties of the empirical distribution of the training data by incorporating an increasingly detailed collection of features. This allows better generalisation to new examples. New features are greedily added, incrementally, to a ME model by calculating the improvement each candidate feature adds to the model. The candidate features initially comprise of all the potential features extracted from the training data and any other feature source. In order to calculate improvement, or gain, the IIS is required to estimate the Lagrange multipliers of the candidate features. Once the gain given by the remaining candidate features becomes sufficiently small, the iterative method stops adding new features.

An initial linguistically-based approach is to group information common to multiple word senses of a word into a single feature. Suárez and Palomar (1993, 2002) propose a set of template features that group information from several examples into one feature, referred to as set-features. Their approach results in a reduced number of features at a marginal degradation of accuracy, around 1.75% in the tests presented. This approach can be applied in a selective manner to the new context features. If more than one word sense of a word shares the same component of their contexts, for instance the initiator of a verb, above a high threshold probability, a single feature can model the component of the context shared by multiple word senses rather than using multiple features.

Finally, a new linguistically based approach is proposed that may also be used to further reduce the number of features used in a WSD ME model as presented in this chapter. As the features proposed use semantic similarity to match words, it may be found that a number of features will be similar to each other thus meaning they may apply to the same examples. For instance, a particular verb sense may have either "man#1" or "woman#1" as an initiator, therefore producing two different but similar features. Two such features may be reduced to one feature by generalising their initiators using their most informative subsumer (as defined in chapter 4), therefore producing a single feature where "person#1" is the initiator. Such a technique can be applied to any group of features where the arguments of the features are sufficiently similar. However, some care must be taken. If this process is performed in an unsupervised fashion, some important information and distinctions may be lost.

Detailed discussion of the above techniques is outside the scope of this thesis as each technique relies on there being a working implementation of a ME training algorithm. As such, the creation of such reduction techniques is left open to investigation.

Other Considerations

Due to the use of similarity to match concepts instead of word-form, the standard framework for ME becomes unsuitable for use with the features described here. The main problem is that for any given ambiguous word, w, in a local context, its possible decisions are the senses for the word w according to WordNet. If two or more such senses are similar, independence between the different classifications is no longer possible, and as such features that match with the similar meanings will add to the normalisation value. In the situation where only examples for one of the similar senses are available, the empirical distribution of the senses will never be calculated properly due to the increasing size of the normalisation value. This in turn will very quickly produce a computational overflow issue during the calculation of the normalisation value. In order to avoid this, the standard normalisation function $Z_\lambda(c)$ is adapted to handle dependant decisions. If two different senses match with the exact same examples they could be treated as applicable as each other in those situations. There are naturally cases where they do not both match with the same set of examples due to the non-transitive nature of the similarity measures used here (see chapter 4). In such a case, the more likely of the two senses should take precedence. However, the shared examples should also be reflected in the calculation of the likelihood.

Consider the standard $Z_\lambda(c)$ function, illustrated in 6.39.

$$Z_\lambda(c) = \sum_{d \in D} \exp\left(\sum_{i=1}^{n} \lambda_i f_i(c,d)\right) \tag{6.39}$$

It is currently assumed that each decision is independent, but this is not always the case. If decisions are now treated as sets of unrelated decisions, $Z_\lambda(c)$ can be represented as:

$$Z_\lambda(c) = \sum_{D \in S(c)} \exp Z'_\lambda(c,D) \tag{6.40}$$

$$Z'_\lambda(c,D) = \max_{d \in D}\left(\sum_{i=1}^{n} \lambda_i f_i(c,d)\right) \tag{6.41}$$

where $S(c)$ is a set of similar sets of word senses for the ambiguous word in c. Therefore, all decisions or word senses in D will be similar to each other. Using this normal-

isation may result in the situation illustrated by 6.42.

$$p(d|c) \geq 1 \tag{6.42}$$

The situation in 6.42 will occur if two or more senses of the ambiguous word in the local context are similar, as they may both be applicable as part of the surrounding context in which they appear.

6.3.3 Experiments

Prior to the more formal experiments documented in this section, tests were performed to see if such a ME classifier would produce promising results. To do this, the verb "give" was selected as a test candidate. "Give" was chosen as it is highly polysemous (45 senses), and thus provides a challenging problem to investigate. It has been shown that the more ambiguous a word is the more frequently it is used in every day language (Zipf, 1945; Jastrezembski and Stanners, 1975; Jastrezembski, 1981). By making a larger number of examples available, a variety of difficulties for WSD and a large amount of variations in the contexts in which such words are found can be examined. If a word with few senses is chosen, for which few examples are available, little variation in the word's uses would be found. Additionally, the accuracy on such a simpler problem will also do little to help in an open-text situation, as again in practice the more ambiguous words in a language tend to be found more frequent in day to day examples, therefore highly ambiguous words cannot be ignored. These initial tests helped in assessing the required feature templates and the influence that using similarity measures has on training a ME classifier. Ideally, more than one word is desirable for such an analysis. However, given the time needed for producing the required tagged corpus of examples, analysis was initially restricted to "give". Preliminary results from testing the initial classifier created for "give" with some hand-tailored examples yielded promising results.

After the preliminary tests, attention was turned to seeing how well such a ME classifier would perform with WSD. It is important to note that as no form of feature reduction or feature induction has been used, results presented here give an indication of the lower bound to the potential accuracy that would be expected for the classifiers.

The goal of the experiments presented in this section is to test if the ME classifiers produce interesting and expected results in their current form, not to provide a conclu-

sive evaluation comparing the WSD classifiers to other techniques. This is because the classifiers created do not currently represent a complete WSD system, and a number of elements remain unfinished. The first experiment evaluates a single ME model trained to perform WSD for all test words. Experiment 2 assesses whether creating separate classifiers for each word improves results and the effect of removing distribution features. Experiment 3 determines if the current ME classifiers can be used to reduce the cost of manually labelling word senses. Finally, experiment 4 assesses the performance of the best classifiers produced on two tasks:

1. Disambiguating words senses for contexts where more than one word in the context is a test word, and where the context words have unknown senses.

2. Disambiguating examples of the test words where the correct sense was not seen in any of the training examples.

Experiment 1 Creating a WSD ME Classifier to Disambiguate Different Words

As an initial experiment, it was decided to see if a unified ME classifier could be created for WSD. A unified classifier uses a single statistical model to handle all input words. To create such a classifier, a corpus of test and training data was created since there were no publicly available corpora containing both the sense information required about words and the syntactical structure required to generate local contexts for sentences. A number of considerations influenced experimental design:

- The time required to create such a corpus;

- The number of words required to show that these ME classifiers produce reasonable results;

- If any tools are available to aid in the creation of such a corpus.

Within the framework set out so far, it was decided to only use resources required by the WSD system and to include the verb “give”, for which data already existed, plus ten reasonably ambiguous nouns found within the same local context as “give”, presented in table 6.5. Sentences containing any of the above words were extracted from Semcor to create the sample database. Thus far, sentences containing synonyms or similar words to those tested were not intentionally extracted due to the amount of time required to manually process the data subsequently. If any information is available for

Word	Senses
Dog	6
Eye	5
Family	7
give	45
Information	5
Instruction	4
Party	5
Report	7
Suggestion	5
Vote	5
Work	7

Table 6.5: Number of Senses per Test Word According to WordNet 1.6

similar words, then this is available as these similar words coincidentally appeared in the sentences extracted. Table 6.6 summarises the information extracted from Semcor. The table includes all sentences containing the words of interest, regardless of part-of-

Word	Number of Sentences
Dog	37
Eye	177
Family	124
Give	677
Information	132
Instruction	16
Party	53
Report	195
Suggestion	20
Vote	55
Work	429

Table 6.6: Summary of Example Sentences

speech and whether the CMU parser generates a usable linkage. Also, only Semcor sentences where the nouns are sense labelled were used for the nouns in the test. These sentences were then parsed and checked manually using the NLP application's interface to the CMU link parser 2.3. In order to produce adequate and meaningful linkages

from the CMU parser, it was necessary to split or rearrange some sentences. In total, 1,971 linkages were created from the input sentences. Any possible word collocations were automatically detected and checked using the information from the original Semcor tags. From the resulting data, the local contexts for all content words were extracted automatically, giving a total of 10,966 local contexts. As some Semcor sentences were changed to produce adequate parses from the CMU parser, the sense tags for all the words had to be labelled manually. This gives the basis of the data to be used for both training and testing the ME classifier. Table 6.7 shows the average polysemy of the final dataset.

	Average Polysemy
All Test Word Examples	22.1 Senses
Test Noun Examples	6 Senses

Table 6.7: Average Polysemy of Examples in Final Dataset

The local context data was divided so that the contexts for 70% of all sentences was reserved as training data, and 30% for test data.

Training the ME WSD Classifier: Using only the training data, a total of 14,635 possible features were generated from the local context examples. From these features, a complete set of empirical probabilities for the training examples and features were calculated and cached for use with the training algorithm. By caching this information, the performance of the training algorithm is improved by reducing redundant calculations. It is worth noting that the strictest interpretation of probability was applied to the training data, i.e. that the data constituted a complete and closed set of examples. An alternative to such an approach could be to assume at least one unknown example, therefore removing cases where some examples are assigned probability 1 by introducing a margin for error. Where many examples are available for a word sense, this error will be small. However, if only one example is available for a word sense the error introduces a larger influence. It was decided not to apply the later approach during these experiments as it assumes information that is not included in the training data. However, this is deemed to be satisfactory in the case of this WSD problem as it is known that the dataset is not closed, and that important examples may be missing.

For a ME classifier to be completely trained it must accurately compute probabilities that match with the empirical distribution of the training sample, and it must also be as uniform as possible. This would not necessarily be desirable as the ME WSD classifier would become over-trained to the training examples, and would not necessarily generalise well to new examples, a widely recognised problem often referred to as overtraining. As the classifier is intended for use in WSD, equation 6.43 is used to measure the accuracy of the classifier.

$$Accuracy = c/n \tag{6.43}$$

where c is the number of correct classification according to Semcor made by a classifier and n is the number of examples used to test the classifier. In situations where two senses have the same conditional probability, the sense with the lowest sense ID is selected. This equation provides the information used to measure the success of the classifier as in practice the primary and dual problems of ME are purely measures about the statistical model itself. Figure 6.10 shows the classifier's accuracy over the training data. Unfortunately, the classifier could not be trained past iteration 31 as the Lagrange multipliers for features become too large to compute the conditional probabilities using standard floating point precision numbers. This is most likely due to contradictions in the training data requiring large Lagrange multipliers in order to more closely match the empirical distribution of the examples. It may also be due to some unhandled factor due to using similarity relations in the features, though this is harder to determine as it would most likely appear as contradictions in the data.

Analysing the results, however, shows some interesting aspects. It seems likely that a classifier with such complexity would require many more iterations before all words would reach their maximum accuracy, but for "dog", "give" and "report" we see a negative trend emerging up to iteration 32. For "dog", senses 2, 3 and 4 relate to human type definitions, and given the high percentage of examples of words similar to "person#1", we can assume that at iteration 13 the classifier starts to assign senses 2, 3 or 4 higher probabilities for certain examples. "Report" shows a more worrying trend. This is easily understood due to the nature of WordNet's definitions for "report". We see that out of the 7 senses available for "report", 3 senses are very closely related via "information", and one further sense related also to these 3 senses via "communication". Given the fine-grained distinction between over half

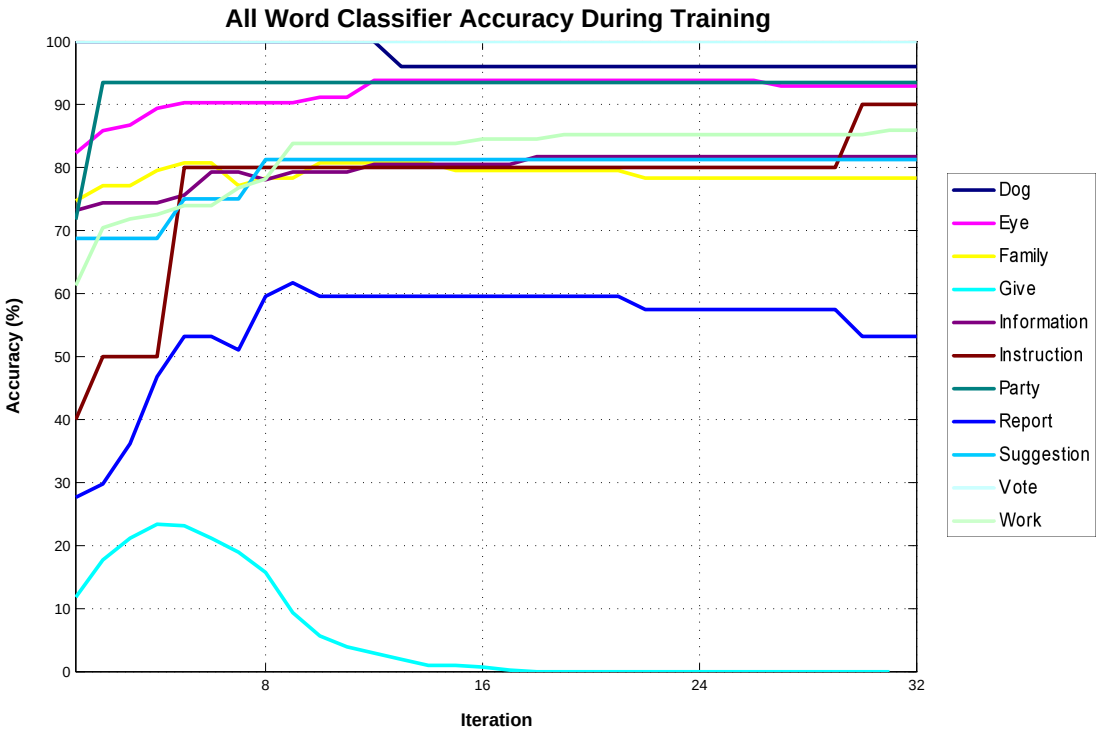


Figure 6.10: All Word Classifier Accuracy During Training

of the senses of "report" it makes it likely that most of the misclassified examples are genuinely ambiguous given the contexts being considered, for instance:

> "She wanted his report first thing in the morning"

Here, "report" may refer to a verbal report, a written account, a study or a paper. The correct interpretation here is clearly influenced by information not contained in the local context. Results for "give" are discussed later in this section.

Overall, using the input sentences as a source of features is adequate for this type of ME classifier. However, the problems encountered and some of the misclassifications in the training data can be attributed to not reducing the feature set so that only relevant features are considered.

Testing the ME WSD Classifier: Attention is now turned to the accuracy of the new classifier in sense tagging the test data. This test is performed to see if the classifier generalises well to new examples. Figure 6.11 shows the accuracy of the classifier at

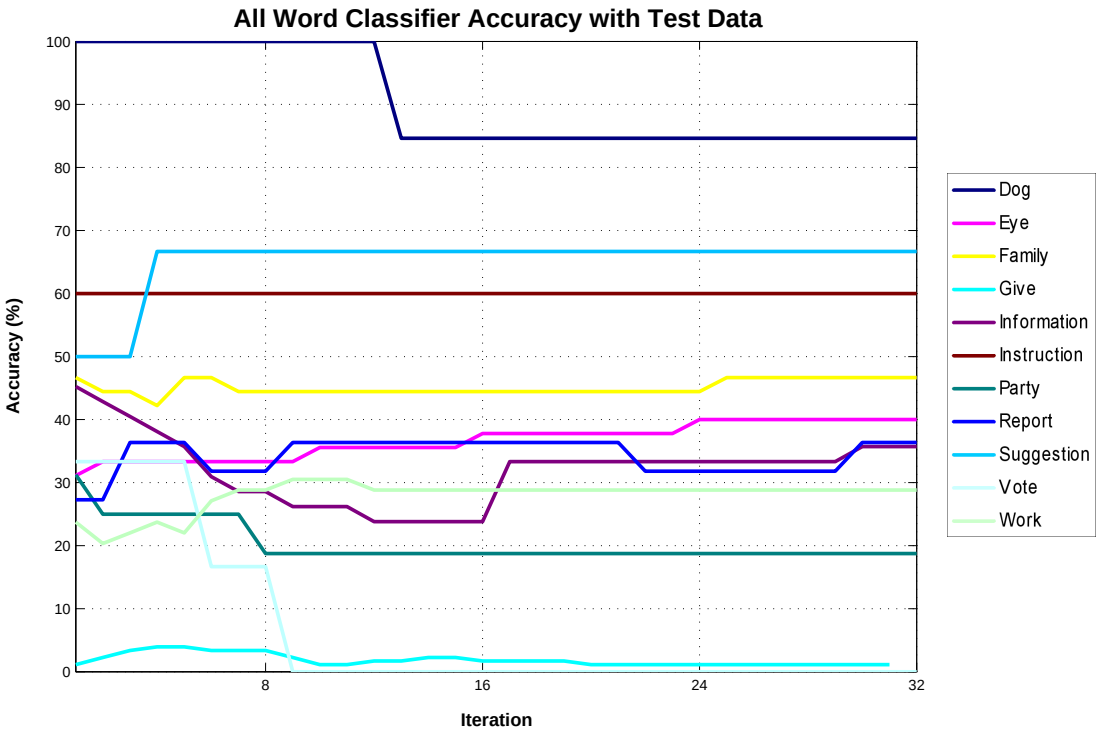


Figure 6.11: All Word Classifier Accuracy with Test Data

disambiguating the test data at various iterations of training. The test accuracy results give a mixed picture. Overall the results are fairly modest, but the range of accuracy of the words is large. We could assume that dog would get good results as only examples for sense 1 are found in Semcor. Again, at iteration 13 we see a dip in accuracy due to a human related sense of "dog" being selected. The poor results for "give" are also unsurprising given the training results. Results for "vote" are surprising though, even given the low number of examples available. Looking at the summary of the results we can see that the classifier starts to assign a higher probability to "vote#5" than to any other sense of "vote" for the test examples. Given that there are no examples for "vote#5" in Semcor, features taken from other words that are similar to "vote#5" bias the results. For the other test words there is evidence of a trend to improve, but we do not yet see the typical signs of overtraining. This suggests that further training is possible, but does not give further information about the potential of such a classifier.

The performance across all evaluation words is shown in figures 6.12 and 6.13,

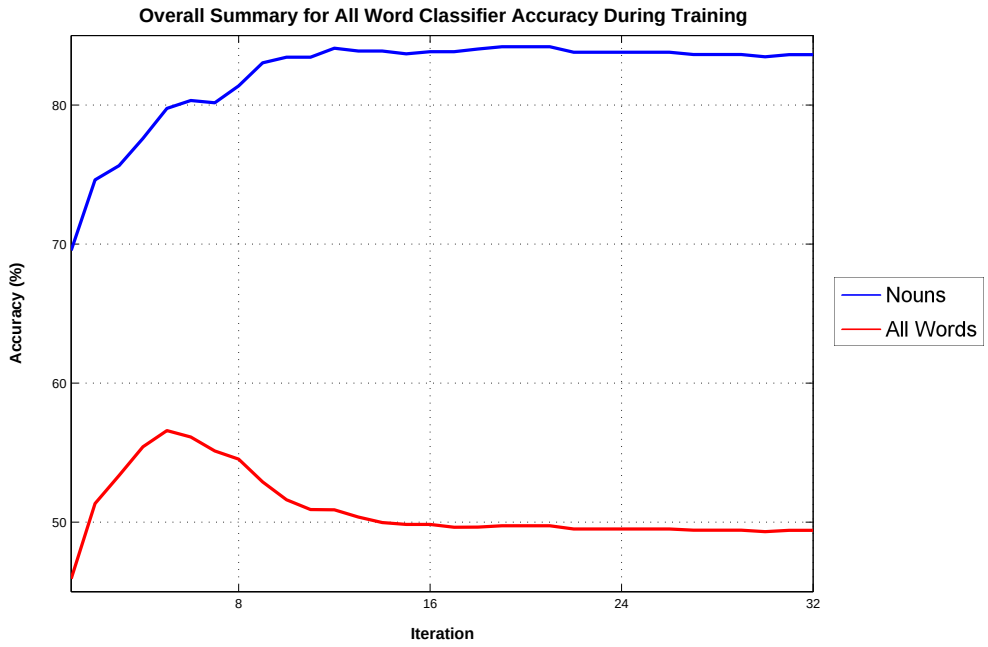


Figure 6.12: Overall Summary for All Word Classifier Accuracy During Training

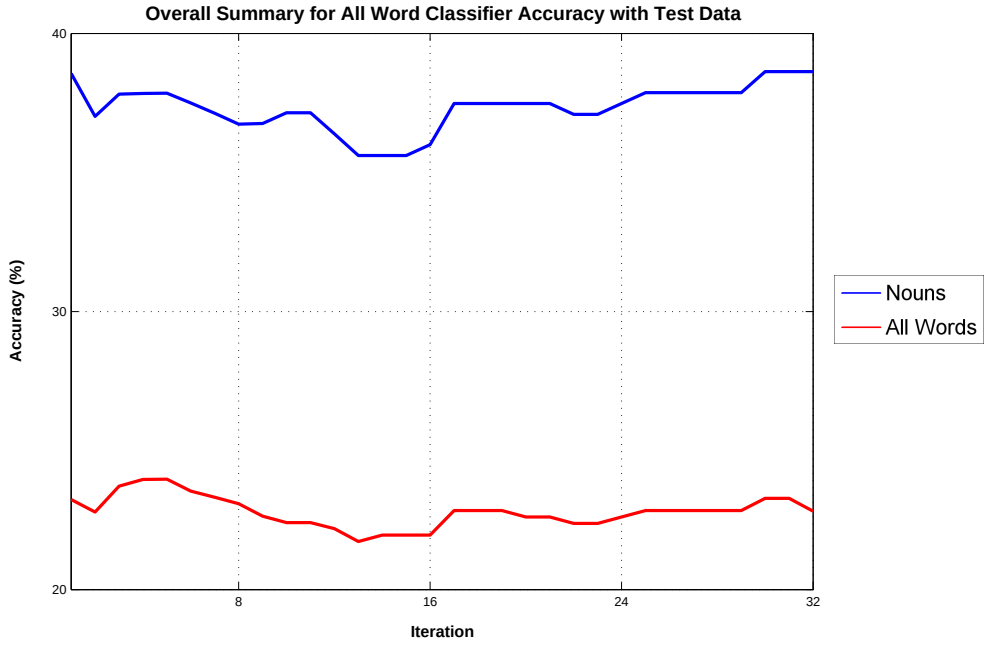


Figure 6.13: Overall Summary for All Word Classifier Accuracy with Test Data

firstly using the training examples, and then using the test examples. The red line on the figures represents results when classifying all words in the training and test set, and the blue represent results when only classifying nouns. The overall performance with the training examples is greatly impaired by the results for "give". However, when considering only the test nouns the classifier gives promising results. The results for the test examples imply that the classifier has not yet stabilised as there are no clear indications of overtraining. There is an initial dip in accuracy explained by the fact that prior to training, at iteration 0 the classifier always picks the first, and most frequent, sense of words. For Semcor this gives high accuracy as the distribution of senses in WordNet was calculated using frequency counts from Semcor. However, the accuracy of such an approach for other corpora tends to be lower as demonstrated in the Senseval 2 English lexical sample test where the overall accuracy for selecting the most frequent sense is 47.6%. As the classifier starts to make more informed predictions, the number of errors made initially increases compared to selecting the first sense. By the 16^{th} iteration the accuracy starts to increase. Currently, the classifier does not generalise well given the substantial gap between training example and test example accuracy.

Considering only the performance with the test examples, the classifier currently reaches its best performance for all evaluation words at iteration 5 with all test words, and at iteration 31 when only considering nouns. Table 6.8 shows the results at this iteration. In order to provide a baseline accuracy, the accuracy whilst only considering the first sense for the test words is included in table 6.8. According to WordNet, the first sense of a word is the most frequent sense within the Semcor corpus. Whilst this baseline provides the percentage of instances the first sense of a word is taken as correct within the Semcor corpus, it tells us little about the comparative performance of the classifier against other WSD systems. Also, as the frequency of word senses in Semcor was used to order senses in WordNet, it is expected that such a baseline will be biased for Semcor. This is reflected in similar baselines for alternative corpora, for instance for the Senseval lexical sample test data, only 48% of words are assigned the first sense in WordNet (Note that the lower baseline of 45% is heavily influenced by the verb "give". The most polysemous word in the Senseval lexical sample test has only 16 senses). It is further reflected by the fact that the noun baseline here is higher than the human inter-tagger agreement rate of 57% between Semcor and the DSO corpus (Kilgarriff, 1998a). Other work follows the gold standard of Senseval to implement an alternate WSD system such as an adapted version of Lesk (1986). However, it is not

Word	All Word Best Accuracy	First Sense Accuracy	Noun Best Accuracy	Noun First Sense Accuracy
Dog	100%	100%	84.62%	100%
Eye	33.33%	95.56%	40%	95.56%
Family	46.67%	42.22%	46.67%	42.22%
Give	3.93%	21.35%		
Information	35.71%	62.5%	35.71%	62.5%
Instruction	60%	60%	60%	60%
Party	25%	56.25%	18.75%	56.25%
Report	36.36%	72.73%	36.36%	72.73%
Suggestion	66.67%	66.67%	66.67%	66.67%
Vote	33.33%	50%	0%	50%
Work	22.03%	38.98%	28.81%	38.98%
All Words	23.97%	45.06%	38.63%	61.48%

Table 6.8: Best Test Data Results For All Word ME WSD Classifier

possible within the time available to do the same here. Also, the current system only represents a prototype of a ME WSD system. As such, a number of elements which may improve the classifier, such as feature reduction, are unavailable as a working ME training system was required prior to the development of such elements. The experiments presented here do not represent a full scale evaluation of such a ME WSD system, only preliminary tests.

Problems with give: The results for verb "give" show some discouraging results at this stage. It is expected that the results would be at best modest given the ambiguity of the verb. However, the results for the test examples are at best just under twice better than random (1/45). We can also see from the results that it is due to "give" that the classifier cannot be trained further, as some of the statistics of the training examples for "give" become incomputable at iteration 32. There are encouraging results up to iteration 4. However, the classifier quickly degenerates from that stage on. To understand why this is occurring we must look more carefully at the results.

Accuracy only gives us information about the most likely sense for each example. However, to have a more complete impression of what is occurring with the classifier, it is beneficial to consider where the correct sense occurs in a list of senses ordered by the probability assigned by the classifier. The average rank of the correct sense

in such a list can be used as a measure to see if the WSD ME classifier is improving or deteriorating in accuracy. Figure 6.14 shows the average rank of the correct sense of the verb "give" when testing with training data (red line) and test data (blue line). The graph in figure 6.14 gives a clearer idea of how the classifier is performing. Over

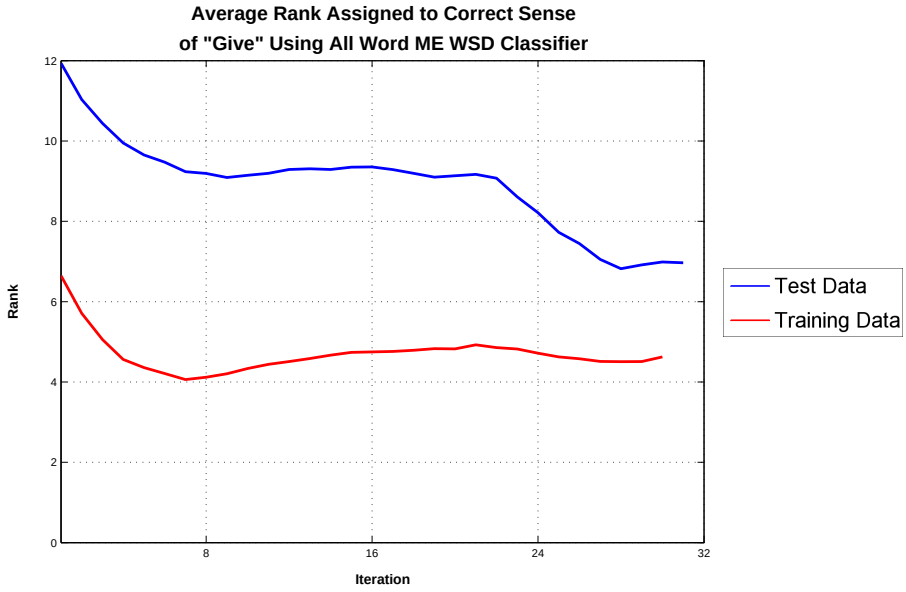


Figure 6.14: Average Rank Assigned to Correct Sense of "Give" Using All Word ME WSD Classifier

the course of training, the correct sense of "give" according to the Semcor examples becomes more likely compared to other senses, showing that some progress is made in improving the ME classifier. By the 5^{th} iteration, on average the correct sense occurs in the top 9% of senses as ranked by the classifier. In contrast, classifications for the test set produce the best ordering of senses at iteration 28 where the correct sense occurs on average in the top 14% of senses ranked by the classifier, with later iterations showing a slight deterioration in results.

Throughout these experiments, all the potential features found in the training examples were used for the ME classifier. The likelihood is that many of the features over-complicate the statistical model and consider relationships that are potential contradictions, or do not assist in the task of WSD. Such examples include features that are highly probable for local contexts that differ in the word sense assigned. For "give", we see that in many cases the initiators and goals of the verb for many different senses are

people. This leads us to assume that in most cases we are not interested in modelling such facts. Until suitable feature induction techniques are developed for this classifier, it is impossible to tell how significant any improvement would be with a reduced feature set.

Experiment 2 Creating Individual Classifiers for Specific Words

The first experiment generated a single WSD classifier for any word using a ME WSD classifier. Given the inherent difficulty in the field of WSD to produce single statistical models capable of modelling all words in a language to a high level of accuracy given limited training data, it is common to find that individual models are created per word in a language for WSD. Words attain their best accuracy at different stages of the training, and due to the examples and features for "give" the classifier is only trained up to iteration 31. By splitting the single classifier into different classifiers, one for each test word, it is also possible to train the classifiers further. It is assumed that results will improve as only relevant examples from the corpus are considered by each classifier, and there is also the benefit of using classifiers at varying iterations of training for each of the words.

Experimental parameters of the first experiment were retained where possible, specifically:

- The same data is used
- The feature set from the first test remain the same

The training and test examples are split into 11 sets such that in each set only examples of words similar to the word of interest are kept. This allows for 11 different classifiers to be trained, one for each word of interest. Table 6.9 shows the data available for evaluating each of the classifiers independently. The "Number of Similar Training Examples" column shows the number of examples for words used during training that are semantically similar but have a different word form to the test word. For further detail about the distribution of the examples for each sense, see Appendix F. Two tests are performed with the different example sets. Firstly, we shall use the same features from experiment 1. A further experiment is performed to see how the distribution features affect these classifiers by removing such distribution features. The results of both tests are then combined to give a final combined result.

Word	Number of Training Examples	Number of Similar Training Examples	Number of Test Examples	Split Ratio
Dog	25	364	13	65.79%
Eye	113	72	45	71.52%
Family	83	407	45	64.84%
Give	406	32	178	69.52%
Information	81	225	40	66.94%
Instruction	10	270	5	66.67%
Party	46	299	16	74.19%
Report	47	292	22	68.12%
Suggestion	16	279	6	72.73%
Vote	12	110	6	66.67%
Work	142	252	59	70.65%

Table 6.9: Data Available for Each Word of Interest

Results Using Distribution features: Figure 6.15 shows the disambiguation accuracy of each new WSD ME classifier with the training examples. The effects of the additional training can be seen immediately. The chart also shows that the individual word classifiers attain success earlier in training process, and classifiers can be trained much further than before. The improvement in the initial stages of training for individual words is due to the comparative simplicity of the classifiers in comparison with the large complex classifier used for experiment 1. Figure 6.15 shows training up to iteration 304. However, most classifiers were trained much further to see if further improvements were possible.

Results for "report" show surprising characteristics. In the first experiment it could be seen that toward the later stages of training, the accuracy for disambiguating "report" showed a slight negative trend. This trend occurs much later in the new classifier, but is more dramatic. Again, this is probably due to not reducing the feature set, and because four different senses of "report" are very similar. This means that some context examples are insufficient for resolving the ambiguity given the fine-grained sense distinctions contained within WordNet.

"Give" shows similar difficulties with the new classifier, but again will not train past iteration 31 without feature reduction. Given that data for other similar verbs was not specifically collected, this is expected to some extent.

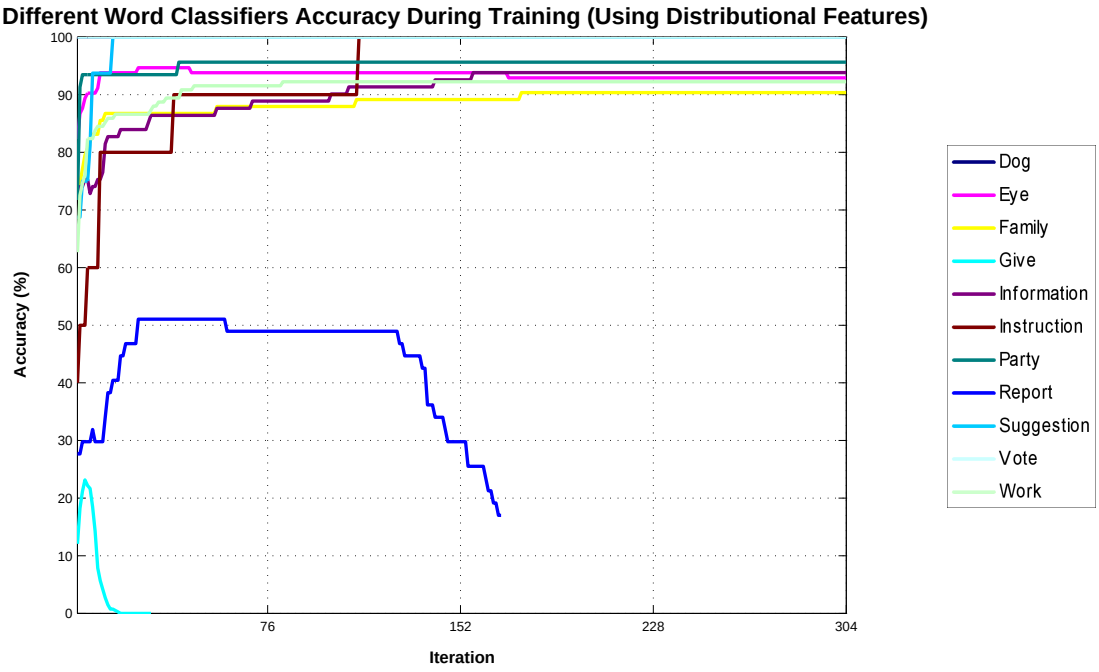


Figure 6.15: Individual Word ME WSD Classifier Accuracy During Training

Figure 6.16 illustrates the accuracy of disambiguating the test examples with the new classifiers. The accuracy of the classifiers with the test examples shows an overall trend to improve over more training iterations, except for "party" where the best results are attained during early iterations. This suggests some overtraining for "party" almost as soon as the classifier starts training. This may be due to tagging errors within Semcor, for instance consider the sense of "party" in:

> "She wrote it down right between the weekly PTA meetings and the Thursday night neighborhood card parties." (Semcor Source: br-f08 paragraph 16 sentence 1)

Here, a natural interpretation of "party" would be WordNet sense 2, meaning a social event where people are gathered for entertainment. However, Semcor has "party" tagged as sense 4, meaning the actual group of people that are gathered for pleasure. Further similar examples exist within Semcor. Such inaccuracies not only skew results from the classifier, but may also cause contradictions within the features extracted that create difficulties for the ME WSD classifier.

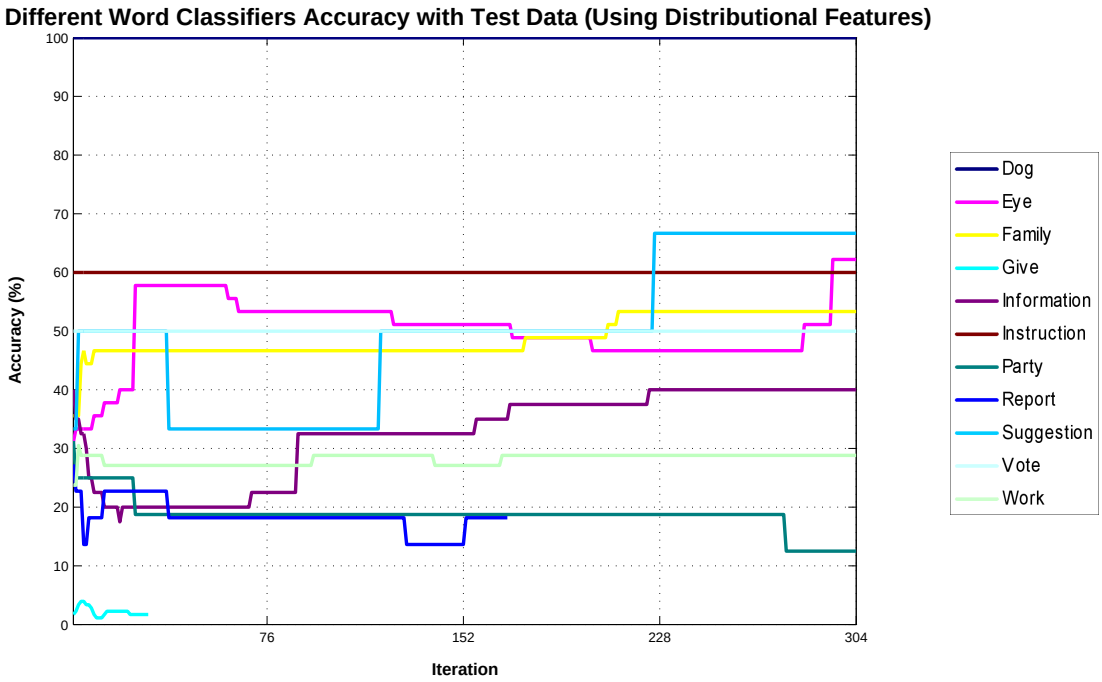


Figure 6.16: Individual Word ME WSD Classifier Accuracy with Test Data

Table 6.10 summarises the best test results for each word, giving the best training accuracy during those iterations, the first sense accuracy as a baseline, and quoting the best iterations. The best iterations are found by determining the set of iterations for which the best test results are attained, then by considering the best training results for those iterations and the lowest average rank for the correct sense. Overall results show improvement over those of experiment 1. However, results for "give" are not significantly improved and results for "vote" are worse (partly due to a lack of examples).

Results Without Distribution features: A similar set of classifiers without distribution features was also tested. This may improve the results of some classifiers as each classifier is trained only using relevant examples for the word the classifier represent, and the distribution of senses may be less affected by the distribution of similar senses for different words. Disambiguating the training examples over the various iterations produces the graph in Figure 6.17. Without using the distribution features, more

Word	Best Test Accuracy	Test First Sense Accuracy	Training Accuracy	Training First Sense Accuracy	Iterations
Dog	100%	100%	100%	100%	All Iterations
Eye	62.22%	95.56%	92.92%	92.04%	295→
Family	53.33%	45.22%	90.36%	48.19%	212→
Give	3.93%	21.35%	23.15%	21.67%	4
Information	40%	62.5%	72.84%	61.73%	1
Instruction	60%	60%	100%	50%	112→
Party	31.25%	56.25%	71.74%	54.35%	1
Report	27.27%	72.73%	27.66%	72.34%	1
Suggestion	66.67%	66.67%	100%	56.25%	226-367
Vote	50%	50%	100%	91.67%	All Iterations
Work	30.51%	38.98%	74.65%	39.44%	3
All Words	29.19%	45.06%	55.86%	45.57%	
Nouns	46.69%	61.48%	78.96%	62.43%	

Table 6.10: Best Results for Individual Word ME WSD Classifiers (Using Distributional Features)

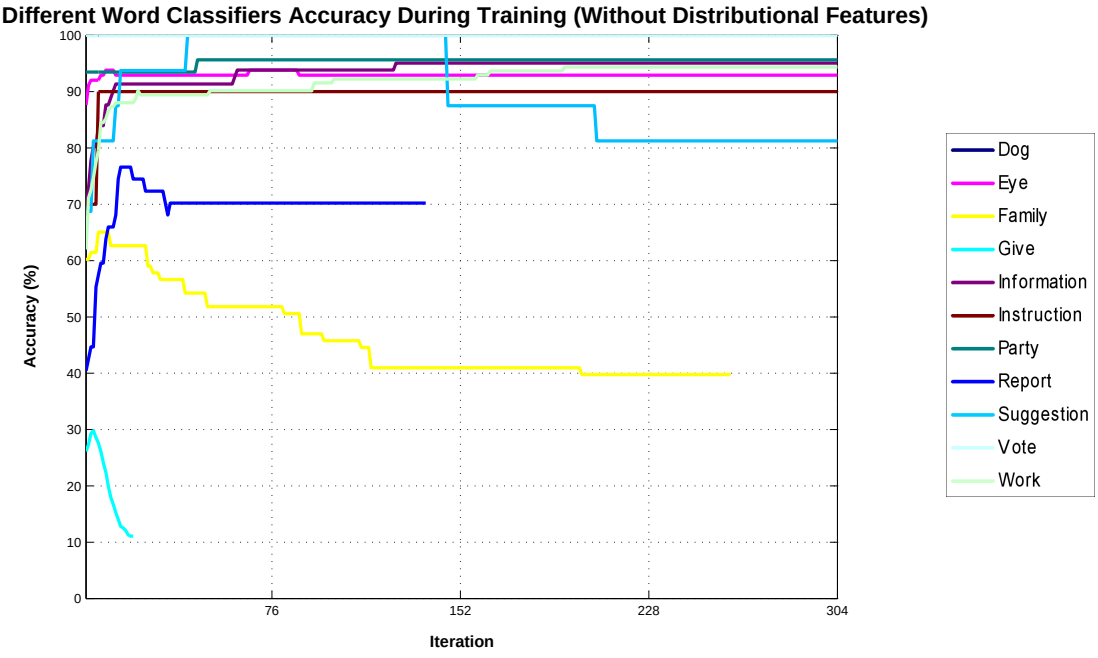


Figure 6.17: Individual Word ME WSD Classifier Accuracy During Training

classifiers have difficulties during training, namely those for “family” and “suggestion”. This suggests that for those words, distribution features aid production of a better classifier. For “suggestion” this is possibly due to the lack of examples and thus the distribution features present a more important role in the WSD process. However, the same cannot be said for “family”. Looking more closely at the senses of “family” suggests that the classifier could have problems with the similarity of the senses relating to “group”, and therefore using distribution features helps in reducing errors by favouring the selection of frequently occurring senses for a given example. The classifiers with training problems when using sense distribution features still show the same behaviour; however the overall accuracy attained while training shows significant improvement.

Figure 6.18 shows the graph of results for disambiguating test examples using the new classifiers without distributional features. “Dog” performs marginally worse here

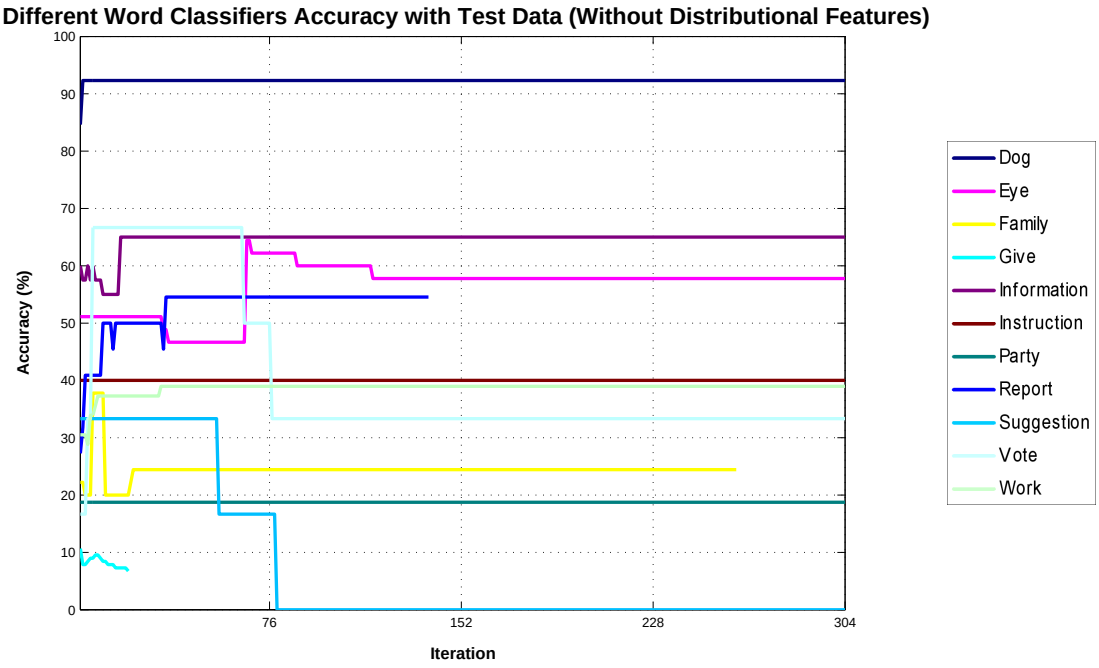


Figure 6.18: Individual Word ME WSD Classifier Accuracy During Training

as the classifier selects a “person” definition in a limited number of cases. “Party” also shows slightly worse performance, although this would be expected if the corpus contained incoherent sense tags. “Family” portrays some of the problems present for “report”, due to similarity within a number of the senses of the word, and now it can

only be trained up to iteration 261. “Instruction” and “suggestion” also perform worse, although this is likely to be due to the lack of examples to determining suitable features. Overall, however, the results achieved without using sense distribution features give significant improvements over the results with the distribution features as they tend to perform better with words that have larger numbers of examples. Whilst still not giving adequate accuracy for “give”, there is a significant improvement. Table 6.11 summarises the best results for the classifiers not using distribution features.

Word	Best Test Accuracy	Test First Sense Accuracy	Training Accuracy	Training Sense First Accuracy	Iterations
Dog	92.31%	100%	100%	100%	2→
Eye	64.44%	95.56%	93.81%	92.04%	67,68
Family	37.78%	42.22%	65.06%	48.19%	10
Give	10.67%	21.35%	26.11%	21.67%	1
Information	70%	62.5%	96.3%	61.73%	354-696
Instruction	40%	60%	90%	50%	6→
Party	18.75%	56.25%	95.65%	54.35%	46→
Report	54.55%	72.73%	70.21%	72.34%	45-48
Suggestion	33.33%	66.67%	100%	56.25%	42-55
Vote	66.67%	50%	100%	91.67%	30-65
Work	38.98%	38.98%	92.25%	39.44%	104-139, 141-150
All Words	34.71%	45.06%	62.59%	45.57%	
Nouns	51.36%	61.48%	88.35%	62.43%	

Table 6.11: Best Results for Individual Word ME WSD Classifiers (Without Distributional Features)

From the current experiment, 22 different classifiers have been produced to perform WSD on 11 words. If the best classifiers for each word are selected, the results in table 6.12 are achieved. Whilst the overall accuracy of the system across all words is fairly low, the tests performed here show results for extreme situations, given the average polysemy of the words used in the tests. The words tested have a higher classifier polysemy than would be expected for typical words in WordNet or even in standard texts. Statistics about word polysemy in the tests performed, within WordNet and in open-texts are summarised in table 6.13. Given the additional senses to consider, and the lack of feature reduction, these results can be seen as baseline values for the approach

Word	Best Test Accuracy	Test First Sense Accuracy	Training Accuracy	Training First Sense Accuracy
Dog	100%	100%	100%	100%
Eye	64.44%	95.56%	93.81%	92.04%
Family	53.33%	42.22%	90.36%	48.19%
Give	10.67%	21.35%	26.11%	21.67%
Information	70%	62.5%	96.3%	61.73%
Instruction	60%	60%	100%	50%
Party	31.25%	56.25%	95.65%	54.35%
Report	54.55%	72.73%	70.21%	72.34%
Suggestion	66.67%	66.67%	100%	56.25%
Vote	66.67%	50%	100%	91.67%
Work	38.98%	38.98%	92.25%	39.44%
All Words	37.7%	45.06%	64.83%	45.57%
Noun	56.42%	61.48%	92.17%	62.43%

Table 6.12: Best Performance for Individual Word ME WSD Classifiers

POS	Test Average Polysemy	WordNet Average Polysemy	Natural Text Average Polysemy
Noun	6	1.23	≈4.7
Verb	45	2.17	≈8.3

Table 6.13: Statistics about Polysemy

described. The results when only considering nouns are, however, very promising. Improvements in these results are expected to follow from use of feature reduction techniques.

Experiment 3 Improving the Cost of Manual WSD

The previous experiments show that the current accuracy of the classifiers is inadequate to perform automatic WSD to the quality required for applications. In order to use output from the WSD classifiers, each classification must be checked manually. One way in which the classifiers can aid manual WSD is to reduce the number of senses to

be considered, therefore simplifying the task for a human tagger. This is a natural way forward, for when erroneous classifications are made the correct sense classification is typically ranked highly according to the classifiers.

A cost function is firstly required in order to test the improvement possible by reducing the number of senses being considered during WSD. Assume that the "cost" of tagging a word is proportional to the average number of senses for all words in a text being annotated, hence the cost function for tagging a word when an incorrect classification is made is given in equation 6.44.

$$c = Tn \tag{6.44}$$

where T is the average time to disambiguate a word, and n is the average number of senses for a word minus one. Now suppose that only a reduced number of senses, αn on average, are supplied to the human tagger, then the cost becomes:

$$c = \alpha Tn \tag{6.45}$$

where α is a parameter that reduces the senses being considered. In its simplest form, α is the proportion of senses to be considered, and is bound by equation 6.46.

$$0 \leq \alpha \leq 1 \tag{6.46}$$

However this does not take into account situations where the correct sense is not listed in the reduced set of senses being considered. On average, the classifier will have a probability, p_e, of the correct sense not being a member of the set of reduced senses. The total cost function is therefore represented as:

$$c = \alpha Tn + (1 - \alpha)\, p_e Tn \tag{6.47}$$

The cost function can be further simplified, as Tn is a constant, k, for a given set of words, so we can write:

$$c = k\left(\alpha + p_e - \alpha p_e\right) \tag{6.48}$$

$$r = \frac{c}{k} = \alpha + p_e - \alpha p_e \tag{6.49}$$

where r is the relative improvement in the cost of tagging whilst using a WSD classifier to reduce the senses being considered. With the relative improvement function, if $\alpha = 0$, the classifier only picks the most probable sense meaning. The additional relative cost for checking other senses will be the error for the classifier. The worst case scenario is when $\alpha = 1$, as the human tagger must consider all word senses (i.e. the number of senses is not reduced).

Testing the best classifiers produced in experiment 2 with the cost function 6.49 for considering senses not initially selected by the classifiers produces the graph in Figure 6.19. The red line represents the results when disambiguating the test examples, whilst the blue line shows the results for disambiguating nouns. The green and magenta lines represent the manual tagging cost of all words and nouns respectively at sense reduction α. For both nouns and all word tests, the classifiers minimise the relative cost

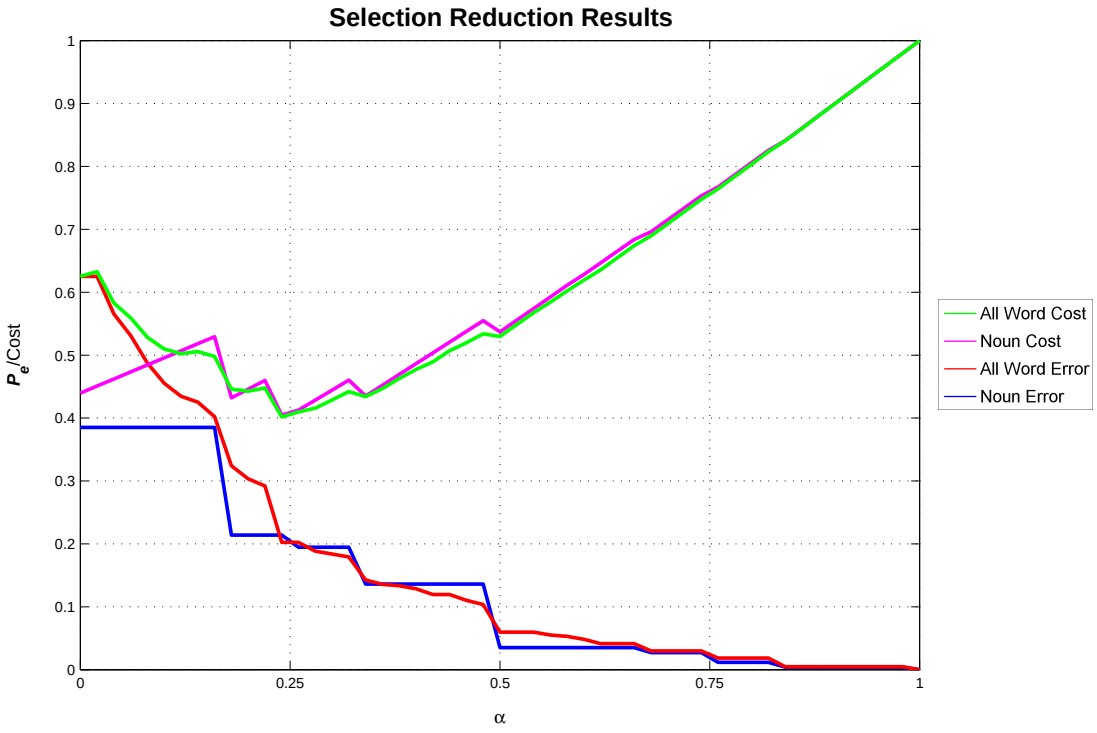


Figure 6.19: Selection Reduction Cost Reductions

of manual tagging by selecting a sense and then leaving a further 25% of the leftover senses as potentially correct. The additional cost of manually checking senses assigned

to words at this stage is 0.4 for both all words and for nouns. This is equivalent overall to a potential 60% reduction in the cost required to manually disambiguate words in open-texts.

A further way of reducing the senses for a word would be to select senses which meet the condition expressed in 6.50 given the results from the classifier.

$$p(c|s) \geq p(c|s_1) \times t \tag{6.50}$$

where s is a sense for the ambiguous word given in context c, s_1 is the most likely sense selected by the ME classifier and t is a threshold. In order to evaluate the efficiency of reducing the number of senses to consider, the results show the effect of the threshold over the proportion of senses being considered and the resulting cost. These results are illustrated in Figure 6.20 for all words, and Figure 6.21 for nouns. The red line shows the percentage of senses being considered at threshold t, the blue line represents the error of the classifier at threshold t, and the green line represents the manual cost tagging cost at reduction threshold t.

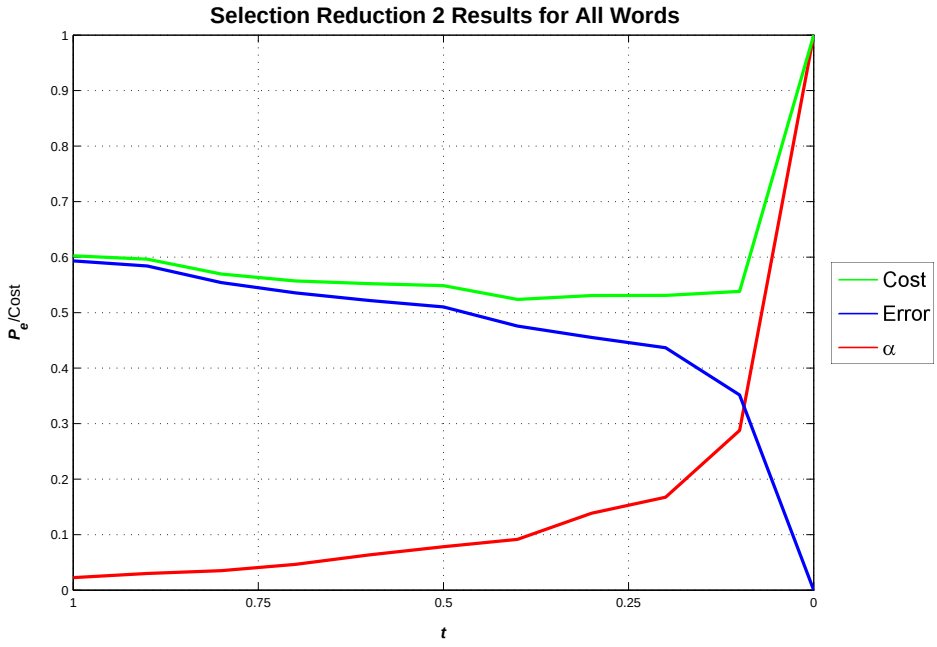


Figure 6.20: Selection Reduction 2 Cost Reductions (All Words)

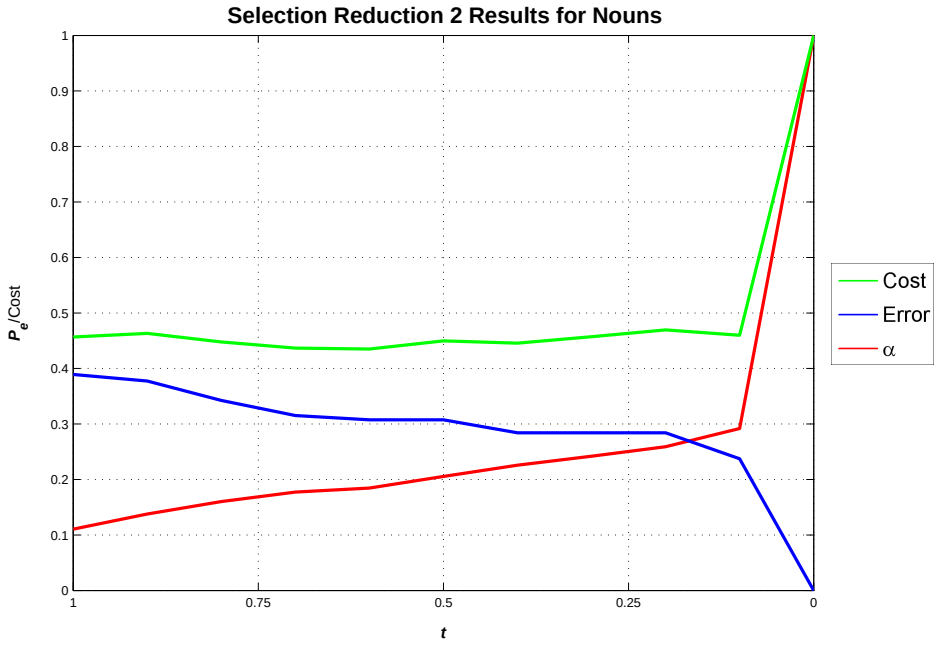


Figure 6.21: Selection Reduction 2 Cost Reductions (Nouns)

Both graphs show that by using thresholds few extra senses are considered before $t = 0.1$, although this is not totally unexpected. Given the relatively small number of examples, there may be aspects of the test contexts not modelled by the features. This, together with the number of training iterations used for the classifiers, means that some more generic aspects of the test contexts mislead the classifiers into being fairly confident about incorrect classifications. A confident classifier will assign a relatively high probability to a small group of senses compared to other senses of a word. Unfortunately the gain made in precision when considering more senses is not significant enough to warrant using such a technique for reducing the number of senses being considered. It is unnecessary to test for when the threshold is between 0 and 0.1, as this represents the weakest results from the classifiers and therefore will select more incorrect senses resulting in an increase in the cost of manually disambiguating the words. The best results are shown in table 6.14.

Given these results using 6.50, the initial sense reduction technique of adjusting α in 6.49 is preferred as a method for reducing the cost of manual WSD. The sense reduction techniques show that even error-prone techniques can lead to significant cost

	Cost	Threshold	% of Extra Senses Considered
All Words	0.524	0.4	9.2%
Nouns	0.435	0.6	17.7%

Table 6.14: Best Results for Threshold-Based Sense Reduction

benefits over manual sense tagging.

Experiment 4 Tests Using Ambiguous Contexts and for Handling Untrained Word Senses

Handling Ambiguous Contexts These experiments have determined how well the classifiers can classify words given unambiguous local contexts. However, the situation where context words are ambiguous has not been tested. In a complete system, as described in section 6.1, the senses to be considered for most words have already been reduced, and some words will have even been disambiguated with Yarowsky's one-sense-per-collocation and one-sense-per-discourse hypotheses (Gale et al., 1993; Yarowsky, 1993, 1995). At this point in time not enough data was available to be able to create other partial taggers and to extend the current statistical component to work with more words. With more data, the effect of calculating the product of the probabilities of any combination of senses could be used to select senses from multiple ambiguous words. Applying such an approach means that consideration must be made about optimisation due to the combinatorial explosion arising from the number of possible sense assignments (Wilks et al., 1990). This experiment demonstrates the effects of simultaneously disambiguating two words, and demonstrates that the classifiers can generalise to the point of disambiguating word senses for which no training data was available.

Examples are selected from the Semcor data where "give" contexts contain a noun for which a classifier has been created, and the equivalent noun contexts were also used. Rather than exhaustively testing every combination of possible senses, the senses of the words in the context will be set to 0, as sense 0 considers all senses of a word at once. Disambiguating the example phrases produces the results in table 6.15, where the rank of the correct sense is presented when it is not selected as most likely.

Sentence ID	Sentence Frame	Training/ Test Data	Noun Result	"give" Result
br-k18 para-26 sent 2	"…she'd give#0 me a look out of narrowed eyes#0…"	Test	Correct	Incorrect 4^{th}
br-p12 para-41 sent 5	"…softening eyes#0 gave#0 her a look…"	Training	Correct	Incorrect 12^{th}
br-g11 para-1 sent 6	"…the family#0… gives#0 us both some immunity… and a way…" (note the 2 local contexts for give)	Training	Correct	Incorrect 2^{nd}
br-a02 para-36 sent 1	"…the ballot couldn't give#0 enough information#0… for the voters…"	Training and Test	Correct	Correct
br-f03 para-24 sent 1	"…priest… gave#0 him… information#0 …"	Training	Correct	Incorrect 2^{nd}
br-j11 para-14 sent 1	"Detailed information#0 on record lengths… is given#0 in the section …"	Test	Correct	Incorrect 2^{nd}
br-j01 para-2 sent 2	"…give#0 otherwise unobtainable information#0…"	Test	Correct	Correct
br-j03 para-6 sent 3	"They also give#0 information#0 which will aid…"	Training and Test	Correct	Correct
br-l14 para-29 sent 2	"…who gave#0 the information#0…" ("person" substituted for "who")	Training	Incorrect 3^{rd}	Incorrect 12^{th}
br-j37 para-7 sent 5	"Both parties#0… were busily at_work… trying… give#0 the elections a… degree"	Training	Correct	Correct
br-c04 para-39 sent 1	"The party#0… gave#0 the "chorines" a chance…"	Test	Incorrect 5^{th}	Incorrect 3^{rd}
br-k29 para-6 sent 2	"…permission to give#0 a camp reunion Halloween party#0…"	Training	Correct	Correct
br-a02 para-22 sent 2	"…give#0… a favorable report#0…"	Test	Correct	Correct
br-j34 para-3 sent 6	"They will give#0 suggestions#0…"	Test	Correct	Incorrect 2^{nd}
br-j34 para-12 sent 9	"…his suggestions#0 are given#0 the consideration they deserve…"	Training	Correct	Incorrect 2^{nd}

br-h11 para-1 sent 2	"This work#0 gave#0 a heat_of_formation…"	Training and Test	Correct	Correct

Table 6.15: Ambiguous Context WSD Test Results

For the Semcor examples tested here, very promising results are seen. In some cases, the selected senses considered incorrectly classified would also make for valid interpretations. However, the test performed here is to see how well the classifier predicts Semcor's sense classification. Table 6.16 summarises the precision of the classifiers selecting exactly one sense, and selecting a reduced set of senses for further disambiguation. The most likely reason for the relatively good results in this smaller test

	Precision
Classifier Selecting 1 sense	65.6%
Classifier also considering 25% of senses other than the most likely	87.5%

Table 6.16: Precision Summary for Ambiguous Context Test

compared to experiments 1 and 2 is that all contexts used here contain a number of relationships. In the previous tests a number of contexts are tested containing smaller numbers of relationships, for instance only a determiner linked to the noun. More words and relationships make for a richer context, allowing more information to be available to the ME WSD classifier. Primarily, results for "give" are dramatically better for this smaller test.

Handling Unseen Senses A further aspect of the classifiers to be considered is the ability of the classifiers to generalise sufficiently to disambiguate senses for which no examples were available in the corpus. This is possible as the features use semantic similarity to match words instead of word-form. Given that no word sense disambiguated examples for such senses are currently available, the problem in question is how to evaluate the classifiers' performance at generalising to such new senses. In order to show that this generalisation is at least possible, a small number of examples containing word senses similar to the word senses of interest are selected, as long as

the context still makes sense when the words are replaced with the word of interest. Table 6.17 shows the selection, giving with each example its source and the result of disambiguation after the word of interest has been substituted into the example. Examples of classifying contexts with the missing senses of dog, family, instruction and suggestion are not given here as the best classifiers created so far make use of sense distribution features, making them much less likely to generalise to new senses. The classifiers not making use of sense distribution features do, however, show evidence of

br-p01 paragraph 12 sentence 1 "...glued to Drexel_Street..." → "...glued to eye..." or "...glued to the eye..." "Eye" successfully labelled with sense 4, meaning a center of a location.
br-k17 paragraph 34 sentence 3 "...get inside Majdanek..." → "...get inside the eye..." or "...get inside the eye of the city..." "Eye" successfully labelled with sense 4, meaning a center of a location.
br-j55 paragraph 16 sentence 1 "...defending...through loopholes..." → "defending...through eyes..." "Eye" successfully labelled with sense 5, meaning a hole.
br-g31 paragraph 11 sentence 9 "...atmosphere...content..." → "...atmosphere...information..." "Information" successfully labelled with sense 2, meaning data.
br-j32 paragraph 1 sentence 1 "...organize the...contents..." → "...organize the...information..." "Information" successfully labelled with sense 2, meaning data.
br-e23 paragraph 18 sentence 2 "...illusion of depth..." → "...illusion of information..." "Information" successfully labelled with sense 4, meaning selective information/entropy.
br-e30 paragraph 67 sentence 1 "...eye to minimum inconvenience to the operation..." → "...eye to minimum information to the operation..." "Information" successfully labelled with sense 4, meaning selective information/entropy.
br-e25 paragraph 23 sentence 1 "...a description of the...parts..." → "...a report of the...parts..." "Report" sense 5 is as likely as sense 1, where sense 5 is a written evaluation.
br-j31 paragraph 3 sentence 2 "...saying in a...condemnatory tone..." → "...saying in a...condemnatory report..." "Report" sense 5 is as likely as sense 1, where sense 5 is a written evaluation.
br-j12 paragraph 7 sentence 4 "...criticism of...views..." → "...reports of...views..." "Report" successfully labelled with sense 6, meaning a composition/paper.
br-f03 paragraph 19 sentence 3 "...impulses in...associated word symbols..." → "...impulses in...associated word reports..." "Report" successfully labelled with sense 6, meaning a composition/paper.
br-j37 paragraph 8 sentence 1 "...deterioration of local party organization." → "...deterioration of local party vote." "Vote" successfully labelled with sense 4, meaning a body of voters.
br-h18 paragraph 8 sentence 1 "...agenda of...dozens of international bodies..." → "...agenda of...dozens of international votes..." "Vote" successfully labelled with sense 4, meaning a body of voters.
br-j06 paragraph 2 sentence 2 "...determine values...of...reactions..." → "...determine votes...of...reactions..." or "...determine reaction votes..." "Vote" successfully labelled with sense 5, meaning a voter turnout.
br-g15 paragraph 1 sentence 1 "...number of characteristic elements..." → "...number of characteristic votes..." "Vote" successfully labelled with sense 5, meaning a voter turnout.

br-g14 paragraph 6 sentence 1 "...record of a_few pictures..." → "...record of a_few works..." "Work" sense 5 is as likely as sense 2, where sense 5 is a body of work from a writer.
br-j19 paragraph 19 sentence 7 "...votes for...pair of pictures..." → "...votes for...pair of works..." "Report" sense 5 is as likely as sense 2, where sense 5 is a body of work from a writer.

Table 6.17: Examples of Disambiguating Word Senses Where No Training Data Was Available

this kind of generalisation.

Currently, the results in table 6.17 are possible without collecting further examples. However, results could be improved by including sentences containing words similar to the test words in the training data, thus producing a broader spread of examples. We must also assume that results are biased towards senses for which examples have been collected; this is clearly seen with contexts containing determiners and for classifiers created with limited numbers of examples. This is because any examples for words similar to those particular words' senses without examples in Semcor have been collected by chance.

6.3.4 Limitations

The current limitations of the technique introduced in this chapter can be categorised as; test limitations, data limitations, or feature reduction limitations.

Test Limitations

There are two subcategories of limitations in the tests presented here; the type of tests performed and the objectivity of the tests performed. The classifiers produced thus far only comprise part of a total WSD system, as described at the start of the chapter. The intention of the statistical component tested in this chapter is to make a decision, given the senses remaining from previous components of the WSD system, regarding which sense is most likely for the ambiguous words. As such, the classifiers make use of very tight contexts that do not use any cross sentential information, nor information about other words outside the local contexts. If the tests were performed by human participants, it would be likely that different conclusions would be made by the participants to the classification of the sense found in Semcor. Therefore it may be more sensible

to compare results from the classifiers with results from humans when given the same information. This would reflect a more suitable test for the way in which the classifiers work.

The objectiveness of the tests is also a consideration, as they do not give results to directly compare with results from other existing techniques. The situation is compounded by using examples for words with an average polysemy greater than the average polysemy for full texts. A word like "give" cannot give results indicative of the performance of classifiers to disambiguate most other verbs. This can be addressed either by implementing other techniques and testing them with the same data, although the data sample is small, or by collecting enough data to be able to repeat the Senseval tests for which test results are available for a large number of different techniques. The latter approach is preferable as Senseval currently represents the gold standard for evaluating WSD techniques.

Data Limitations

Given the small number of both syntactically and sense labelled examples available it is not possible to perform more large scale evaluations across entire documents, nor is it currently possible to disambiguate adjectives or adverbs. This has been due to the cost associated with manually checking the syntactic structures produced by the CMU link grammar parser.

A further problem with the data is due to the way in which the sentences were selected. All sentences containing at least one word of interest were extracted from Semcor. Local contexts for all words in the selected sentences were kept as the classifiers could be trained with, and make use of words similar to the test words. The classifiers would benefit from being trained with as many examples of words similar to the words of interest. However, not all sentences containing at least one word similar to a test word were extracted. This may have produced biased results by omitting examples from Semcor that could have been used to train the WSD ME classifiers. By using the data from similar words, the classifier could be trained with a richer source of more varied examples.

Feature Reduction

Currently all features have been collected from training examples. However, their validity and usefulness remains to be evaluated. This unnecessarily increases the complexity of the classifiers, and causes incorrect conclusions to be made during WSD. For instance, considering the examples for the verb "give", it is clearly seen that most of the time the initiator and goal of the verb is "person#1" for most senses of "give". Indeed, for some of the senses for which there are few examples in Semcor, all examples use "person#1" for the initiator and goal if they are available. This would indicate a conditional probability of 1 for later senses of "give" when the context only comprises of an initiator similar to "person#1". It would be more beneficial to consider the earlier senses of "give" given such a weak context. Clearly this shows that such features actually hinder the performance of WSD.

6.4 Future Work

The work presented leaves many aspects for further investigation before it can be used for large scale WSD. Given the automatic method used for collecting features from example local contexts, the most immediate requirement is the implementation of a suitable feature reduction technique. A number of techniques are briefly discussed in section 6.3.2. However, the implementation of such techniques is beyond the scope of this thesis. Once an adequate reduced set of features is available, work should concentrate on creating further example data from Semcor, Senseval, and further available sources, in order to test the technique using the Senseval experiments. With this data available, classifiers could be created and evaluated comparatively against other WSD techniques.

Evaluating the impact of using semantic similarity to match words in features, compared with using only word-form, is currently untested. The current hypothesis is that by using semantic similarity the coverage of the features will be expanded, and thus require less training data to give comparable results. There are two tests that should be performed to measure the effectiveness of using semantic similarity in the statistical features:

1. Firstly, the "similar word" relations for each part-of-speech should be defined to be true only if both the word-form and sense are the same for the pair of words

being tested. With this change to the relations, new classifiers would need to be trained using the same data as the classifiers presented in this chapter, and then the two different types of classifiers could be compared to see if matching using similarity improves results.

2. Classifiers using the current feature set could be trained only using data for similar word senses and not with specific examples for the words of interest themselves. The classifiers should then be tested using all local contexts for the words of interest to evaluate how well they perform WSD on examples for words not used in the training data. This would give an indication of how successful the classifiers could be at disambiguating word senses for which no examples were available.

Another aspect of the current work which should be tested comparatively is the definition of local context introduced in this chapter. The difference in the accuracy attained from using syntactic relations to create local contexts, rather than using a more traditional context window should be tested, where a context window represents context as the n content words directly surrounding the word being evaluated.

In order to further improve WSD accuracy using only the ME classifiers, a number of options should be considered:

- Using Discourse Representation Structures (Kamp, 1981) as input to the system instead of only the syntactical structure, in order to solve issues with anaphora and to include cross sentential relationships to provide richer local contexts

- Implement features from other WSD ME classifiers, for instance the features tested by Suárez and Palomar (2002); Dang and Palmer (2002); Klein et al. (2002), and test features reflecting the syntactic structure of local contexts and morphological information

There is also no reason to restrict the statistical component to only the ME paradigm. During the course of the work it was found that the constant information calculated for training lends itself almost directly to creating Support Vector Machines (SVM) (Boser et al., 1992; Cortes and Vapnik, 1995), a technique currently regarded as state-of-the-art for statistical classification. A simplified description of SVMs is that they can be used to create binary decision trees (Platt et al., 2000). Such a decision tree contains at each node a SVM capable of making a binary choice. Training of SVMs is typically

much faster than training ME classifiers, as SVMs are purely trained for classification and not to predict some conditional probability of the decision for some context. This gives the possibility of improved results using the current training method of splitting the data into training and test data. However, it can also give the possibility of training using cross validation techniques (Stone, 1974). This makes it possible to use a larger percentage of the data for training data, whilst still ensuring the classifier can generalise well to new examples.

Finally, it is not the intention to use the statistical technique alone to perform WSD across full texts, but to use such a technique as a partial-tagger in a larger WSD system as described at the start of the chapter. In order to complete the WSD system, other partial taggers must be created, potentially using the same data as used by the statistical technique. With all four partial-taggers implemented, the effectiveness of such a multi-tagger technique can be evaluated.

6.5 Summary

In this chapter, a definition for context based on syntactic and semantic features of language was introduced. This new definition was used in the creation of a statistical classifier for the purpose of WSD. The Maximum Entropy framework, as used by other WSD and classification systems, was followed to create a collection of classifiers. Finally a number of tests were performed to evaluate the usefulness of such a classifier in isolation of further processing. In order to perform the tests, a corpus of local contexts was generated from selected sentences in Semcor for 11 words with an average polysemy of 22.1 senses. From this corpus, 70% of the selected sentences were reserved for training the ME WSD classifiers and the remainder for testing.

Table 6.18 summarises the results for the ME WSD classifiers at disambiguating the training examples taken from Semcor. Iterations producing the best results while disambiguating the test examples are used to create these results.

Table 6.19 summarises the best results for the ME WSD classifiers at disambiguating the test examples, again taken from Semcor.

It is difficult to objectively compare these results with those of other techniques, as the test presented here is too small and no other techniques have been evaluated using the same data. Also, techniques using similar syntactic approaches for context, such as the approach taken by Lin (1997), group senses to form coarser sense distinctions

Classifier	All Words	Nouns
All word classifier, with distribution features	56.50%	80.04%
Individual word classifiers, with distribution features	55.86%	78.96%
Individual word classifiers, without distribution features	62.59%	88.35%
Best individual word classifiers	64.83%	92.17%

Table 6.18: ME WSD Classifier Training Summary

Classifier	All Words	Nouns
All word classifier, with distribution features	23.97%	38.63%
Individual word classifiers, with distribution features	29.19%	46.69%
Individual word classifiers, without distribution features	34.71%	51.36%
Best individual word classifiers	37.70%	56.42%

Table 6.19: ME WSD Classifier Test Summary

in order to avoid difficulties with handling similar meanings. The average polysemy of the test words is also higher than would be expected in a normal text, and significantly higher than the average polysemy of the words in WordNet. Additionally, currently the features which form the basis of the ME classifiers are automatically extracted from the training examples and do not undergo feature reduction. This has the effect of the classifier using an unnecessary number of features, increasing computation time. Examining the majority of the training results, it can be seen that the features selected seem reasonable, as most of the training examples are classified correctly; however the results with the test data are much lower indicating that some of the features selected may produce erroneous classifications and do not generalise well. As such, it is argued that the results presented here estimate the baseline precisions and recalls for the type of classifier developed in this chapter.

Given the high accuracy required for the purposes of using a WSD to improve the results of translation systems on open-texts with domain and topic variations, there are currently no systems capable of working completely independently from human involvement. Given this situation, ways of reducing the number of senses to be considered during WSD were developed, along with a way to measure the relative cost of manually disambiguating a text. The cost ratio measure used is illustrated by equation 6.51.

$$r = \alpha + p_e - ap_e \tag{6.51}$$

where r is the cost ratio, α is the proportion of leftover senses to be considered once the most likely sense according to a classifier has been removed, and p_e is probability that the correct sense is neither the most likely sense according to the classifier nor part of the set of extra senses being considered. When considering all word senses, $\alpha = 1$. Two different ways of restricting senses were considered and evaluated to see if they significantly reduce the cost of manual WSD

- Selecting the first x% of the most likely senses, other than the most likely sense.
- Selecting senses where the probability of the sense is greater or equal to the product of the probability of the most likely senses and some threshold, t.

The best improvement followed from selecting the 25% most likely senses, after considering the most likely sense, with a cost in the order of 0.4. This suggests an reduction of 60% in cost over considering all senses of words being disambiguated whilst performing manual WSD.

Testing was concluded by showing examples from the training and test data where both the verb “give” and one of the test nouns are found in the same local context. This small test yielded some higher than average results using the ME WSD classifier. The results are summarised in table 6.20. These above average results were most likely obtained due to the nature of the data collected, as the data was specifically collected to handle most of the context words in the examples. For the larger tests, most contexts tested only consist of one of the test words.

Finally it was shown that the classifiers generalise sufficiently to handle examples for words senses for which there was no specific training data. This is possible due to the use of semantic similarity in the statistical features to match words with similar meanings (rather than word-form).

	Precision and Recall
Classifier Selecting 1 sense	65.6%
Classifier also considering 25% of senses other than the most likely	87.5%

Table 6.20: Precision Summary for Ambiguous Context Test

Overall, the current state of the work shows promising results. Senseval 2 scores range from of 28.7% precision and 3.3% recall for the lowest ranked system to 69% precision and recall for the best system for the "all word" evaluation on 3 texts totalling 5832 running words. The best results for the Senseval 2 lexical selection evaluation, a similar evaluation to tests presented here using 45 different words with an average polysemy of 5.2 senses, perform at a slightly lower precision and recall of 64%. Even though direct comparison with these results is not possible, the results presented in this chapter compare favourably with the state-of-the-art systems evaluated in Senseval 2. As feature reduction techniques are not yet available, the current results represent a lower bound to the potential accuracy of such a disambiguation technique. Further work to establish larger training data sets would permit objective comparison with other research results.

Chapter 7

Conclusions

This chapter summarises the work and results presented in the thesis. Section 7.1 recaps the techniques presented in chapters 4 and 6, discussing the quality of the results obtained by the test systems developed. Section 7.2 proposes future work to extend the work presented. Finally, section 7.3 summarises the contribution made to the fields of study.

7.1 Summary of Work Presented

The research presented in this thesis build upon two distinct fields of research:

1. Semantic Similarity
2. Word Sense Disambiguation (WSD)

The best resulting system developed from the work with semantic similarity is used in the work performed for WSD.

7.1.1 Semantic Similarity

Semantic similarity between words is measured using WordNet's lexical taxonomy for nouns to produce a number of similarity measures for use as a sub-task of larger natural language processing (NLP) systems. The work develops an original approach using WordNet's hypernym and meronym relations. By considering the shape of hypernym structures, and reducing such structures to only consider nodes for non-technical layman concepts, the similarity measures produced outperform existing measures. The

shape of hypernym structures is calculated using either the product or sum of the hyponym branching of nodes within the structures and the total structure can be reduced to a layman structure by removing concepts where the average polysemy of the words for the concept is not greater than one. Using these ideas, a number of different measures are produced, each capable of considering three different types of structures for measuring shape:

1. Full hypernym structures

   Shape is calculated considering the hyponym branching of every node in a hypernym structure.

2. Layman hypernym structures

   Only the branching of the nodes for layman terms is considered for calculating the shape the structure.

3. Flattened layman hypernym structures

   In order to consider the hyponym branching of non-layman terms, their branching is added to the hyponym branching of the next layman term higher in the hypernym structure. This is equivalent to flattening hypernym structures to only consider layman terms, with the branches of non-layman terms associated with the most relevant layman hypernym.

The similarity measures produced, called SBSMs, exploit the shape of the resulting taxonomy in a variety of ways. In general the SBSMs use the ratio of generalisation between two nouns if they have a common subsumer, and incorporate information common to both nouns given by the structure above the most informative subsumer of the nouns. In addition, hybrid techniques are also considered using shape together with ideas from existing path based techniques for measuring similarity. A number of parameters may also varied to influence the results from the SBSMs, such as considering meronyms when calculating similarity, the normalisation of values to a standard scale for all word pairs and to select whether to use product or sum shape measures.

Two evaluations were performed; the first compared SBSM results with human judgements, and the second used the SBSMs to disambiguate the sense of semantically related words. The evaluation comparing SBSM results to human judgements was performed using three publicly available data sets with human judgements for 65, 30 and

28 word pairs. For each set of human judgements, results are available for a number of existing similarity measure techniques, commonly evaluated using Pearson's product-moment correlation. As it is more natural for humans to order or rank word-pairs according to their similarities, and as values of similarity measures can be adjusted post calculation without changing their relative ordering, the use of Pearson's correlation is believed not to produce the most objective comparison of similarity measures. The evaluation follows the common approach using Pearson's correlation. However, Spearsman's rank correlation is also used to compare the relative ordering of word-pairs according to similarity. Table 7.1 summarises the results from a selection of the best existing measures and SBSMs tested. For each set of human judgements an upper

Correlation Coefficient	Existing Techniques	SBSM Techniques
Pearson	0.75-0.86	0.86-0.91
Spearman	0.71-0.84	0.78-0.86

Table 7.1: Summary of the Best Similarity Human Judgement Correlation Results for Existing Measure and SBSMs

target is available for Pearson's correlation given the average agreement between the human candidates for the test. For this work, we consider this upper target for Pearson's and Spearman's correlation to be 0.9, given the worst correlation between the 3 different sets of human judgements. The results show that new SBSMs improve results when comparing against human judgements, and that their best accuracies nearly match human performance.

WSD of semantically related words is performed using the Wordsmyth thesaurus for which experimental links to WordNet are available. The links are calculated using the Resnik information-content similarity measure. The evaluation is performed using the SBSMs together with a number of simple WSD algorithms. Selecting the first sense and using the Wu and Palmer measure are used as baselines in the evaluation. Given a selection of randomly selected hand-tagged noun entries in Wordsmyth, accuracy, precision and recall are calculated from the results. The precision and recall results from the best performing SBSM and WSD algorithm are compared to the results for the experimental links to WordNet contained in Wordsmyth. The summary of these results is given in table 7.2. As multiple tags are assigned to words where necessary,

	Wordsmyth Links	SBSM WSD Algorithm
Precision	80%	88%
Recall	71%	91%

Table 7.2: Summary of the Best Similarity Human Judgement Correlation Results for Existing Measure and SBSMs

it is assumed that given enough human candidates for the annotation stage, the upper target for such an evaluation should be close to 100% accuracy, although only one candidate annotated the entries used in the evaluation performed. The results show a significant improvement in both recall and precision over the links calculated using Resnik's similarity based WSD technique.

Results were also calculated to show how frequently the WSD systems correctly detect words with no adequate WordNet senses for an entry from Wordsmyth. The best WSD algorithm and SBSM combination accurately detects 57% instances of words with no adequate senses.

From the work performed with SBSMs, the measure 7.1 with shape$_\times$, referred to as $SBSM_{\times 5}$, performs most robustly across both evaluations.

$$Sim_{SBSM5}(c_1, c_2) = Sim_{SBSM1}(c_1, c_2) \times normalise_{CIM}(d(c_3)) \tag{7.1}$$

$$Sim_{SBSM1}(c_1, c_2) = \begin{cases} \frac{shape(c_1)}{shape(c_2)} & : \quad if\ shape(c_1) < shape(c_2) \wedge \\ & \quad c_1 \neq c_3 \wedge c_2 \neq c_3 \\ \frac{shape(c_2)}{shape(c_1)} & : \quad if\ shape(c_1) > shape(c_2) \wedge \\ & \quad c_1 \neq c_3 \wedge c_2 \neq c_3 \\ 1 & : \quad otherwise \end{cases} \tag{7.2}$$

$$shape(c) = \begin{cases} 1 & : \quad if\ c = root(c) \\ \#(\psi(\lambda(c))) \times shape(\omega(c)) & : \quad otherwise \end{cases} \tag{7.3}$$

where c, c_1 and c_2 are word senses, c_3 is the most informative subsumer for both c_1 and c_2, $d(c)$ is the depth of concept c according to WordNet's hypernym taxonomy, $\psi(x)$ is the set of hyponyms for a word sense x, $\lambda(x)$ is a hypernym of a word sense x, and $root(w)$ is the root of the hypernym structure for w.

7.1.2 Word Sense Disambiguation

The work on WSD introduces ideas for a WSD system using a combination of partial-taggers, as illustrated in Figure 7.1. Each of the initial partial-taggers use existing ideas, and are intended only to restrict senses before a penultimate statistical component is used to disambiguate as many of the remaining ambiguous words as possible. Work was restricted to developing a statistical WSD system suitable for such a task. An initial maximum entropy (ME) based system was developed using a number of new ideas:

- A new definition of local context for words based on linguistic principles rather than the classical context window based approach to context. Using grammatical structures according to the CMU link grammar parser, local context is defined considering information collected from the resulting links to a word.

- A new set of features is considered. These features reflect the information in the new definition of local context, and use word similarity according to WordNet's lexical taxonomy for word matching. By using this alternative approach to match words, it is possible to gather information about similar words as input to train a statistical model thus alleviating the lexical bottleneck problem.

A sample corpus was created to develop statistical classifiers for 10 nouns and 1 verb. The corpus consisted of a subset of Semcor sentences containing a word of interest. These sentences were parsed with the CMU parser and manually checked to select adequate linkages. The final corpus therefore consisted of both sense tagged words and linkages from which local contexts were extracted. From this corpus, 70% of examples were used to generate features and train the statistical classifiers, whilst the remainder was reserved for testing purposes.

Although a complete system was not created due to time constraints, with the lack of feature reduction techniques being most notable in the results, the current performance of the maximum entropy approach with the new set of features was evaluated in five different tests:

- Evaluation of the performance of a generic classifier built to disambiguate instances of the 11 test words.

- Evaluation of the performance of specialised classifiers built to handle each of

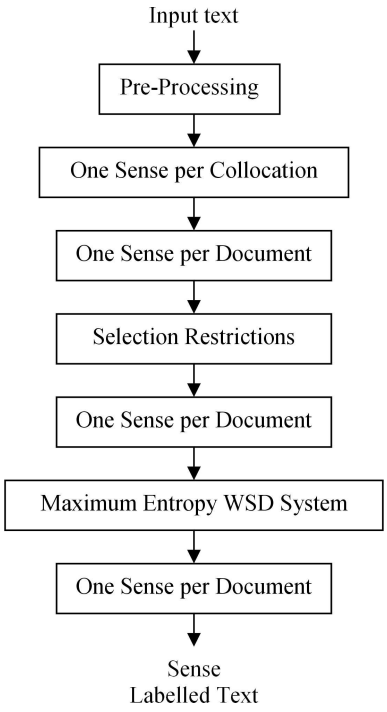


Figure 7.1: Proposed Minimal Set of Partial-Taggers for WSD

the 11 test words. Two classifiers were trained for each word, one including the use of sense distribution features, and one without.

- Evaluation of the ability of each of the best performing classifiers for each word to simplify the task of manual annotation of senses.
- Evaluation of the best performing classifiers to disambiguate example contexts containing the test verb and at least one test noun. These contexts are completely ambiguous, i.e. context words were treated as ambiguous and not sense labelled.
- Demonstration of the ability of the classifiers to successfully assign senses for words not available in the training data.

The results show that use of specialised classifiers for each word, instead of a generic classifier for all words, produces improved results at 56% precision and recall for the nouns tested. Sense distribution features are most useful for words with few available examples. However, they can reduce accuracy for more ambiguous words with an adequate number of examples. Using the classifiers to reduce the number of senses to be

considered by a human annotator can reduce the cost of annotation by approximately 60%. Finally, contexts containing only words considered show improved results, giving an average precision and recall of 66% across all 11 test words.

Overall, results for the WSD ideas developed so far have been fairly modest, especially when also considering results for the test verb used having 45 senses according to WordNet. The main intention of the tests performed was to give evidence of the utility of the new ideas and demonstrate encouraging results early in their development into a full WSD system. For this reason, and given the difficulty of preparing a large enough corpus with the CMU parser to provide data for the ME approach, the tests performed cannot be directly compared to existing systems. Relative performance to other systems can however be inferred. This suggests that these ideas produce results at the upper end of the Senseval 2 results scale. Difficulties for direct comparison are further compounded as Semcor does not present a gold-standard data set for the evaluation of WSD techniques, as average agreement between Semcor and the DCO corpus is only approximately 57% (Kilgarriff, 1998a), therefore some of the sense tags may be incorrect. To directly compare the current implementation would firstly require a gold-standard sense and syntax tagged corpus. However, it is not sensible to perform this until further work is performed on the implemented solutions, such as the inclusion of feature reduction techniques to simplify the ME models. Once this ME implementation is complete, together with implementations of the other partial-taggers described for a complete system, then it would be sensible to use the Senseval 2 data in order to compare results objectively with other systems.

7.2 Future Work

Overall, the work performed for calculating semantic similarity between nouns was highly successful. Possible extensions of this work involve further efforts to use other WordNet relations to calculate semantic similarity, although it is also believed that it would be timely to start developing more rigorous evaluation techniques for similarity measures. A number of approaches should be considered. Firstly, the question of how results from systems are compared to human judgements should be revisited. For instance, is Pearson's product-moment correlation the best measure to objectively compare the standard of the results of the different systems, or is the relative ranking of word pairs more important than the final values assigned? As it is more natural for

humans to rank words relative to each other according to similarity than it is for them to assign similarity values to word pairs, and as similarity values from measures can be fine-tuned to particular tasks, Pearson's correlation of linear dependence between values may not provide an objective comparison between similarity measures. In the light of this, Spearman's Rank correlation coefficient may provide a more objective evaluation measure to compare similarity measures against each other. A technique is proposed for fine-tuning similarity measure results so they fit to a more suitable distribution curve on a graph for a number of words. Also, the current sets of human judgements available are fairly small, with the 30 word pairs given by Miller and Charles (1991) being the most commonly used set, therefore work should be performed to develop a gold-standard data set for evaluating similarity measures.

As similarity measures are generally used as sub-tasks within a larger NLP system, they should also be evaluated using an accepted set of application specific problems such as the WSD of related nouns problem visited in chapter 4. Again, for such evaluations to be truly objective, the labelled data must be considered to be sufficiently replicable and large enough to represent a gold-standard data set for evaluation. Finally, the measures implemented can be used to completely link words in Wordsmyth entries to WordNet with higher accuracy than Resnik's approach.

The work on WSD leaves a number of tasks open to further investigation. The primary task requirement is to find suitable feature reduction techniques for the new features defined in chapter 6.

Once suitable feature reduction techniques have been implemented, attention should be turned to produce adequate quantities of training data in order to be able to repeat the Senseval experiments, enabling the possibility of objectively comparing results from the approach taken against existing WSD systems. This will also provide a more suitable test platform to isolate the effects of original aspects of the system, where specific aspects can be evaluated in isolation, such as:

- The effect of using semantic similarity to match words in the ME features.
- The effect of using the new definition of context based on semantic relationships determined using syntactical information in contrast to using a classical context window approach.

Further tests should be performed to determine how a combination of the features introduced in this thesis together with features from other work for WSD with ME can

further improve WSD, in a similar way to the tests performed by Suárez and Palomar (2002).

Some further alternative approaches have also been considered that are of interest. Firstly, using Discourse Representation Structures (Kamp, 1981) as input to the system may provide a richer source of context than simply using the syntactic relationships between words in a sentence. Also, given the similarity between features in ME and those used in support vector machines (SVMs) (Boser et al., 1992; Cortes and Vapnik, 1995), it is believed that SVMs can be readily created using the information calculated for ME classifiers. As SVM train much faster than ME models, they may be a more practical approach to creating future classifiers for WSD.

Finally, further partial-taggers should be implemented once a suitable final statistical tagger is available in order to evaluate a full, robust WSD system.

7.3 Contributions of the Research

This section lists the original contributions the work in this thesis provides to the fields of semantic similarity and word senses disambiguation:

7.3.1 New Ideas

- A new way to use WordNet's lexical taxonomy for the calculation of semantic similarity between nouns that outperforms existing techniques (Section 4.3 and 4.4).
- A technique to reduce the hypernym taxonomy of WordNet 1.6 to only contain layman terms (Section 4.3.3).
- A new set of WSD algorithms for disambiguating semantically related words by calculating the similarity between senses of the related words (Section 4.5.2 and appendix C).
- A proposal for a multiple partial tagger approach for WSD of all words in texts (Section 6.1).
- A new definition of local context for use in a WSD system (Section 6.2 and 6.3.2).

- A new set of features based on this definition of context, and using semantic similarity to match words instead of word-form (Section 6.3.2).
- An evaluation approach to evaluate the cost of manually disambiguating words while assisted by a WSD algorithm that reduces the senses to be considered (Section 6.3.3).

7.3.2 Tools and Systems Produced

- A number of new shape-based similarity measures (Section 4.4).
- A system for disambiguating groups of nouns, assuming the groups contain semantically related nouns (Section 4.5.2 and appendix C).
- A number of experimental ME WSD classifiers created specifically to disambiguate 11 selected words (Section 6.3.3).
- An application to assist users in parsing sentences with the CMU parser by providing information for more rapid disambiguation of ambiguous linkages (Section 2.4).
- A tool for extracting local contexts from CMU linkages (Section 2.4).
- A tool for extracting ME features from local contexts (Section 2.4).

7.3.3 Data and Resources

- A subset of the Wordsmyth thesaurus with all nouns manually labelled with WordNet senses. This small corpus of thesaurus entries is intended for evaluating WSD systems for related nouns (Section 4.5.2).
- A subset of Semcor has been parsed using the CMU link grammar parser for the development of WSD systems requiring syntactic information (Section 6.3.3).
- A set of local contexts, and ME features are available for the subset of data extracted from Semcor (Section 6.3.3).

7.4 Final Thoughts

Overall, the work presented in this thesis has been successful, though some results have been modest. WSD still has far to go before truly automated systems become robust to the point of being practical without human intervention. However, the current trend of research is producing ever more promising results with increasingly informed lexical resources. An aspect of this work that is for the most part missing in existing research, is use of semantic similarity to match words. The main justification given here for this approach is to address the lexical bottleneck problem, and to be able to disambiguate words or word senses for which no examples were available. The motivation for such an approach can, however, be more ambitious. Once high quality WSD approaches are developed, once multilingual lexicons such as EuroWordNet (Vossen, 1997) reach further maturity, and when a large enough corpus of examples is available for one language, using semantic similarity to match words opens the possibility to create WSD classifiers for different languages using resources from only one language. Whilst the resulting WSD systems may have difficulty capturing some of the cultural differences particular to different languages, this will greatly improve possibilities for various tasks within the NLP field.

Appendix A

Using Hyponym Branching Similarity Measures Comparable to Statistical Alternatives for Word Sense Disambiguation

(Originally published as (Dionisio et al., 2001))

A.1 Abstract

This paper presents 8 similarity measures for use with a word sense disambiguation system for tagging words from open texts with senses according to WordNet. These similarity measures employ hypernym and hyponym information contained within the WordNet taxonomy to assign a value representing the similarity between two word senses. Comparative results show that the measures perform well against the Wu & Palmer similarity measure, and thus is comparable to the original statistically based measure of the word sense disambiguation algorithm used.

A.2 Introduction

Word Sense Disambiguation (WSD) is a major sub task of many Natural Language Processing (NLP) tasks (Kilgarriff, 1997), ranging from machine translation of docu-

ments to information extraction. The main aim in WSD is to use an algorithm, or suite of algorithms, to sense tag words in a document according to some lexical resource.

Two of the most widely used lexical resources in recent years has been the Longman Dictionary of Contemporary English (LDOCE) (Procter, 1978) and, more increasingly, WordNet (Miller et al., 1990; Fellbaum, 1998). Recent techniques can be classified into one of three types (Mihalcea and Moldovan, 1998):

1. WSD techniques solely making use of information from lexical resources (Agirre and Rigau, 1995), (Wilks and Stevenson, 1998b), (Lesk, 1986)

2. Statistical WSD techniques trained from sense tagged training corpora, referred to as supervised training methods (Stetina et al., 1998), (Gale et al., 1992b).

3. WSD using statistical techniques trained with untagged training corpora, referred to as unsupervised training methods (Yarowsky, 1995), (Resnik, 1995a, 1999), (Rigau et al., 1997).

Supervised training methods suffer from the "lexical bottleneck" problem due to the lack of training examples. Attempts to alleviate this problem have used unsupervised training techniques, making use of open texts without sense tagged information. Finding sufficient training data to enable these techniques to work well for open texts still remains a problem. This paper investigates methods belonging to the first class of algorithms and shows results comparable to the statistical techniques for tagging nouns according to senses in WordNet, without requiring statistical training.

The organisation of lexical information within WordNet can be problematic for techniques relying on its taxonomy in order to disambiguate lemmas. Resnik (Resnik, 1995a) shows that hypernym relations vary in the amount of generalisation they represent, therefore they are deceptive for measures relying on edge counting techniques. Resnik tries to tackle this problem by weighting these relations according to statistically collected information.

Section 2 presents measures that take into account the branching of hyponyms for each sense within a hypernym path when calculating the similarity between two word senses. These investigate the sensitivity of such measures to highly developed subhierarchies of WordNet's taxonomy. Section 3 gives a WSD algorithm, as presented in (Resnik, 1995a), which is used as a vehicle for the comparison of the measures

$$sim_{Wu\&Palmer}(c_1, c_2) = \frac{2 * d(c_3)}{d(c_1) + d(c_2)} \quad \text{(a)}$$

$$sim_{shape}(c_1, c_2) = \frac{shape(c_1)}{|hyponym(hyp(c_2))|^i} \text{, if } shape(c_1) < |hyponym(hyp(x))|^i \quad \text{(b)}$$
$$= \frac{|hyponym(hyp(c_2))|^i}{shape(c_1)} \text{, otherwise}$$

where c_1 & c_2 have a common subsumer

$$sim_{shape}(c_1, c_2) = \frac{shape(c_1)}{|hyponym(hyp(c_2))|^i} * d(c_3) \text{, if } shape(c_1) < |hyponym(hyp(c_2))|^i \quad \text{(c)}$$
$$= \frac{|hyponym(hyp(c_2))|^i}{shape(c_1)} * d(c_3) \text{, otherwise}$$

where a c_1 & c_2 have a common subsumer, and c_3 is the most informative subsumer.

$$sim_{shape}(c_1, c_2) = \frac{shape(c_1)}{|hyponym(hyp(c_2))|^i} * \left(1 - \frac{1}{d(c_3)}\right) \text{, if } shape(c_1) < |hyponym(hyp(c_2))|^i \quad \text{(d)}$$
$$= \frac{|hyponym(hyp(c_2))|^i}{shape(c_1)} * \left(1 - \frac{1}{d(c_3)}\right) \text{, otherwise}$$

where c_1 & c_2 have a common subsumer, and c_3 is the most informative subsumer.

$$sim_{shape}(c_1, c_2) = \frac{shape(c_1)}{|hyponym(hyp(c_2))|^i} * shape(c_3) \text{, if } shape(c_1) < |hyponym(hyp(c_2))|^i \quad \text{(e)}$$
$$= \frac{|hyponym(hyp(c_2))|^i}{shape(c_1)} * shape(c_3) \text{, otherwise}$$

where c_1 & c_2 have a common subsumer, and c_3 is the most informative subsumer.

$$sim_{shape}(c_1, c_2) = \frac{shape(c_1)}{|hyponym(hyp(c_2))|^i} * \left(1 - \frac{1}{shape(c_3)}\right) \text{, if } shape(c_1) < |hyponym(hyp(c_2))|^i \quad \text{(f)}$$
$$= \frac{|hyponym(hyp(c_2))|^i}{shape(c_1)} * \left(1 - \frac{1}{shape(c_3)}\right) \text{, otherwise}$$

where c_1 & c_2 have a common subsumer, and c_3 is the most informative subsumer.

$$sim_{shape}(c_1, c_2) = \frac{shape(c_1)}{|hyponym(hyp(c_2))|^i} * \left(1 - \frac{1}{ave\_hyponym\_branch(c_3)}\right), \text{ if } shape(c_1) < |hyponym(hyp(c_2))|^i \quad \text{(g)}$$
$$= \frac{|hyponym(hyp(c_2))|^i}{shape(c_1)} * \left(1 - \frac{1}{ave\_hyponym\_branch(c_3)}\right), \text{otherwise}$$

where c_1 & c_2 have a common subsumer, and c_3 is the most informative subsumer.

$$sim_{hybrid}(c_1, c_2) = sim_{Wu\&Plamer}(c_1, c_2) * sim_{shape}(c_1, c_2) \quad \text{(h)}$$

where $0 \leq sim_{shape}(c_1, c_2) \leq 1$

$$sim_{hybrid}(c_1, c_2) = \left(1 - \frac{1}{d(c_3)}\right) * \left(1 - \frac{1}{shape(c_1) + shape(c_2)}\right) \quad \text{(i)}$$

where c_3 is the most informative subsumer.

Where for all the above algorithms:

$i = d(c_1) - d(c_3)$

and hyp = hypernym

and |hyponym(y)| is the number of hyponyms for sense y

and $shape(x) = 1$, if x is a root node

$shape(x) = shape(hyp(x)) * |hyponym(hyp(x))|$, otherwise

Figure A.1: Similarity Measures

presented in section 2. Section 4 compares the measures using input words taken from categories of Roget's Thesaurus (Procter, 1978) and shows how these relate to results in (Resnik, 1995a). Section 5 describes some of the ongoing and future direction of the work presented in this paper.

A.3 Similarity Measures

Resnik (Resnik, 1999) presents a slightly revised version of the original Wu & Palmer similarity measure (Wu and Palmer, 1994), as shown in Figure A.1 in (1a). Results from (1a) are comparable to results from the original statistically based similarity measure of the WSD algorithm (Resnik, 1999). This measure is used as a baseline against which the other similarity measures presented in Figure A.1 can be compared. The measure calculates values based on hypernym tree depths. d(x) is the depth of a particular hypernym subtree. Values for d(c_1) and d(c_2) are calculated from routes in the hypernym subtrees of c_1 and c_2 respectively that contain a sense, c_3, that is a common hypernym to both of c_1 and c_2. This sense, c_3, is referred to as the most informative subsumer (MIS).

In order to handle pairs of noun senses with no common sense in their hypernym structures, a "virtual" node is used. A match at this node states that the only similarity between two senses is that they are nouns. The depth at the virtual node, d(virtual), is 0. Other WordNet root nodes (e.g. entity, abstraction, etc...) have a depth of 1. For the hypernym structure in Figure A.2, if when compared to another sense's structure the MIS is liquid or fluid, d(brew) = 8. Otherwise in other cases d(brew) = 7.

The measures (1b) to (1i) are all based around the idea that senses with a larger number of daughter nodes (hyponyms) have a more general/abstract relation to their hyponyms. To reflect this idea in a similarity measure, the hypernym distances should be related to the number of hyponyms of a sense's hypernym. This notion is the basis of a measure that uses information about the taxonomy of WordNet to add biases to hypernym distances along a hypernym subtree. shape(x) is a measure of the hyponym branching along a path from parent(x) up to the virtual node, although in practice it is only necessary to calculate shape up to the MIS for the input senses. In order to give preference to senses where either of the two senses is an ancestor in the hypernym subtree of the other sense, shape(x)/hyponyms(parent(y)) is replaced with 1 in the sim_{shape} measures.

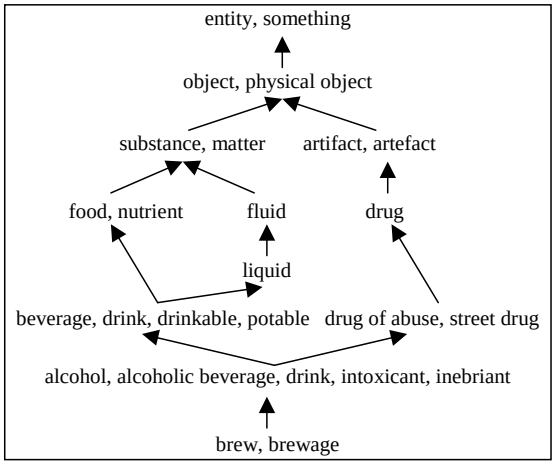


Figure A.2: Hypernym structure for the noun “brew” (Sense 1)

The motivation behind (1b) is to have a similarity measure that is less sensitive to some of the irregularities of WordNet’s taxonomy. The measure prefers senses where the average number of hyponym branches along one sense’s path is similar to the number of branches of the other sense’s hypernym.

Measure (1b) does not take into account information shared by two different senses, only the differences. A measure of this common information can be calculated from the WordNet taxonomy using the subtree of the MIS. Measures (1c) to (1g) extend (1b) using different multipliers, based on information contained above the MIS, to prefer pairs of senses that share a MIS deeper in the hypernym subtree.

Measures (1c) and (1e) use multipliers calculated according to the depth and shape (respectively) above the MIS. The potential problem with these measures comes from the difference in magnitude between (1b) and the multipliers in (1c) and (1e). As (1b) produces values within the range of 0 to 1 and depth and shape measures produce values above 1, the final measure may become overly influenced by the multiplier. Measures (1d) and (1f) overcome this by normalising the multiplier to within the range of 0 to 1. These measure produce values closer to (1b) the deeper the MIS appears in the subtree, but reduce the value of (1b) if the MIS is close to the root.

In (1g), ave_hyponym_branch(c_3) replaces shape(c_3) to determine whether the average hyponym branching of the hypernyms of the MIS produces improved results. This paper will only investigate a measure using the average hyponym branching value

normalised to within 0 and 1.

Measures (1h) and (1i) combine ideas from both the Wu & Palmer measure (1a), and from the hypernym branching methods. (1h) calculates the product of the results from (1a) and one of the algorithms sim_{shape}, for measures where $0 \leq sim_{shape} \leq 1$ is guaranteed. (1i) is an adaptation of the (1a), using the hypernym branching measure shape(x) instead of hypernym depths.

A.4 WSD Algorithm

The WSD algorithm presented in (Resnik, 1995a) produces high quality results when disambiguating noun groupings (Resnik, 1995a, 1999), and can be parameterised with different similarity measures of the form $sim(sense_x, sense_y)$. Experiments have shown that Resnik's original statistical similarity measure and the Wu & Palmer measure produce comparable results when used with this WSD algorithm (Resnik, 1999).

```
Given W = {w[1],…,w[n]}, a set of nouns

for i = 1 to n, for j = i to n {
   v[i, j] = sim(w[i], w[j])
   c[i, j] = MIS of w[i] and w[j]

   for k = 1 to num_senses(w[i])
      if c[i, j] is an ancestor of sense[i, k]
         increment support[i, k] by v[i, j]

   for k' = 1 to num_senses(w[j])
      if c[i, j] is an ancestor of sense[j, k']
         increment support[j, k'] by v[i, j]

   increment normalisation[i] by v[i, k]
   increment normalisation[j] by v[i, k]
}

for i = 1 to n, for k = 1 to num_senses(w[i]) {
   if !(normalization[i] == 0.0)
      phi[i, k] = support[i, k] / normalisation[i]
   else
      phi[i, k] = 1 / num_senses(w[i])
}
```

Figure A.3: Resnik's Word Sense Disambiguation algorithm

Figure A.3 shows the Resnik WSD algorithm. The measure for word i sense k is contained in the variable "phi". The sense with the highest measure is selected as

the most suitable sense. In the case of a tie, the sense with the smallest sense number according WordNet's sense ordering is selected.

A.5 Comparison

The measures are compared using a thesaurus classes example taken from (Resnik, 1995a). This example was intended to show how Resnik's WSD algorithm, along with Resnik's statistically based similarity measure, performed when disambiguating the highly ambiguous noun "line". It is difficult to make a direct comparison with the original example as the results from (Resnik, 1995a) use a different version of WordNet, and Resnik does not state which version was used. Additionally, reproduction of this work requires collection of the supporting data to derive statistical information. Hence the use of the comparable performing non statistical Wu & Palmer measure as the baseline.

The comparison uses words and phrases from 13 different Roget's thesaurus categories containing the noun "line", found online. Replacing <category number> with the Roget's category number of interest produces the entry for the relevant category. These words and phrases are then reduced to nouns with WordNet entries, omitting obsolete and foreign words.

The noun "line" has 29 different senses according to WordNet 1.6. A full description of these definitions can be obtained using the WordNet web interface. Only relevant senses used in the results in Figure A.4 are listed here:

> 2. line – (a mark that is long relative to its width; "He drew a line on the chart"; "The substance produced characteristic lines on the spectroscope")
>
> 5. line – (a linear string of words expressing some idea; "the letter consisted of three short lines")
>
> 7. line – (a fortified position (especially one marking the most forward position of troops); "they attacked the enemy's line")
>
> 9. cable, electrical cable, line, transmission line – (an electrical conductor connecting telephones or television or power stations)
>
> 10. course, line – (a connected series of events or actions or developments; "the government took a firm course" or "historians can only point out those lines for which evidence

is available")

11. line – (a spatial location defined by a real or imaginary unidimensional extent)

13. pipeline, line – (a pipe used to transport liquids or gases; "a pipeline runs from the wells to the seaport")

14. line, railway line, rail line – (railroad track and roadbed)

15. telephone line, phone line, line – (a telephone connection)

17. lineage, line, line of descent, descent, bloodline, blood line, blood, pedigree, ancestry, origin, parentage, stock – (the descendants of one individual; "his entire lineage has been warriors")

18. line – (something long and thin and flexible)

19. occupation, business, line of work, line – (the principal activity in your life; "he's not in my line of business")

20. line – (in games or sports; a mark indicating positions or bounds of the playing area)

24. agate line, line – (space for one line of print (one column wide and 1/14 inch deep) used to measure advertising)

26. tune, melody, air, strain, melodic line, line, melodic phrase – (a succession of notes forming a distinctive sequence; "she was humming an air from Beethoven")

27. note, short letter, line – (a short personal letter; "drop me a line when you get there")

29. production line, assembly line, line – (a factory system in which an article is conveyed through sites at which successive operations are performed on it)

Figure A.4 shows the results of the WSD algorithm using different similarity measures. For each similarity measure, the top three sense IDs selected are displayed in the lefthand column, along with their respective measures in the righthand column. For measure (1h), (1b) is used as a suitable sim_{shape}.

As (Resnik, 1995a) mentions, it is difficult to select acceptable senses of "line" according to WordNet 1.6 for some of Roget's categories used in the comparison performed (e.g. #200, #203 and #466). This explains some of the less satisfactory results.

Figure A.5 shows how the selected senses in the above results compare to the senses selected in (Resnik, 1995a). Results from #200, #203 and #466 have not been

used for this comparison. Also, there is no assumption that the results presented in (Resnik, 1995a) are correct, only that they are good. As a result, the percentages given in Figure A.5 do not show accuracy, and are only intended as a comparison with results of Resnik's example. The reader could decide whether the results in (Resnik, 1995a) are actually correct (especially for Roget categories #413 & #597), or if other senses according WordNet 1.6 better describe the meaning of "line" against the thesaurus classes.

To get a clearer view of the comparison between the measures, 5 people were asked to select suitable senses for each of the Categories used in the creation of Figure A.5. The people were to select all appropriate senses from WordNet's definitions of "line". These intuitions were then compared with the results from all measures, including those given in (Resnik, 1995a), and are summarised in Figure A.6.

Similarity measure (1e) is shown to consistently give poor results for this example. This shows that, due to the size of the values the shape(x) measure generates, using shape(x) as a multiplier in (1e) overly influences the measure's results. It is interesting that (1d) gives comparable results to (1f), which suggests that the normalised multipliers used by the two measures improve results. Further investigation is required to see if the additional processing in (1f) has any advantage over (1d).

A.6 Conclusions & Future Work

Results in Figure A.5, and especially those in Figure A.6 are very positive for all measures presented in this paper apart from measure (1e). This suggests that (1e) performs badly against highly ambiguous words. The results of the other measures can be seen to be above the baseline provided by (1a), and that they are comparable to results from Resnik's statistically based measure.

Work is currently being undertaken to determine how well the similarity measures compare in other situations, including distributionally derived noun groupings, to further assess how well the alternative shape similarity measures perform against the Wu & Palmer (1a) similarity measure.

Work to determine how well these techniques perform with the task of open text disambiguation is also being undertaken using a collection of Semantic Concordance files, SemCor (Miller et al., 1994; Fellbaum, 1998), which are semantically tagged against WordNet 1.6. As the Resnik WSD algorithm has been developed to disam-

biguate small groups of nouns with similar semantic meaning, it is likely that it will not be adequate for the task of full open text WSD. Thus, other approaches and algorithms will be developed to try and maximise the efficiency of using semantic similarity measures for the task of WSD.

The techniques described here will be used along with different techniques to restrict the senses of words to consider within a context, and to assess the best possible sense tags to assign to words according to WordNet.

Roget's Category	(1a)		(1b)		(1c)		(1d)		(1e)		(1f)		(1g)		(1h)		(1i)	
#45 Connection	9	0.650	18	0.749	9	0.622	9	0.619	9	0.993	9	0.671	9	0.672	9	0.529	9	0.771
	15	0.361	9	0.708	15	0.493	15	0.532	15	0.006	15	0.614	15	0.510	15	0.468	15	0.329
	18	0.347	15	0.688	29	0.397	29	0.448	29	0.000	29	0.550	29	0.521	18	0.429	29	0.274
#69 Continuity	17	0.260	17	0.487	17	0.262	17	0.236	26	0.954	17	0.222	18	0.308	17	0.271	17	0.222
	10	0.207	10	0.462	10	0.226	10	0.225	13	0.040	26	0.198	26	0.274	10	0.242	9	0.198
	26	0.121	18	0.401	26	0.119	26	0.181	9	0.002	18	0.198	9	0.272	26	0.158	13	0.119
#166 Paternity	17	0.530	17	0.753	17	0.650	17	0.700	13	0.988	17	0.654	17	0.602	17	0.703	17	0.472
	13	0.236	7	0.586	13	0.205	13	0.117	9	0.009	13	0.176	9	0.162	13	0.159	27	0.213
	9	0.190	9	0.586	9	0.176	2	0.103	17	0.003	14	0.176	15	0.162	9	0.144	2	0.211
#167 Posterity	9	0.553	9	0.893	9	0.530	9	0.494	9	0.846	9	0.594	9	0.620	9	0.548	9	0.456
	15	0.553	15	0.893	15	0.530	15	0.494	15	0.846	15	0.594	15	0.620	15	0.548	15	0.401
	29	0.553	29	0.893	29	0.530	29	0.494	29	0.846	29	0.594	29	0.620	29	0.548	29	0.401
#200 Length	9	0.553	9	0.797	9	0.400	9	0.403	9	0.912	9	0.377	9	0.468	9	0.326	9	0.375
	15	0.553	24	0.778	24	0.285	15	0.332	15	0.085	15	0.347	15	0.434	15	0.292	24	0.316
	29	0.553	15	0.778	15	0.260	29	0.332	29	0.031	29	0.347	29	0.434	29	0.292	27	0.128
#203 Narrowness, Thinness	9	0.462	18	0.861	9	0.444	9	0.451	9	0.428	9	0.487	9	0.550	9	0.459	9	0.429
	18	0.366	9	0.826	15	0.444	15	0.451	27	0.273	15	0.487	15	0.550	15	0.459	15	0.243
	13	0.324	13	0.826	29	0.444	29	0.451	2	0.216	29	0.487	29	0.550	29	0.459	29	0.243
#205 Filament	18	0.639	18	0.810	18	0.503	18	0.628	9	0.713	18	0.747	18	0.742	18	0.663	18	0.485
	9	0.428	9	0.548	9	0.371	9	0.459	20	0.279	9	0.541	9	0.554	9	0.342	9	0.441
	13	0.376	13	0.527	15	0.263	15	0.367	2	0.047	13	0.447	15	0.452	15	0.295	13	0.327
#278 Direction	11	0.326	11	0.701	11	0.352	11	0.441	13	0.727	11	0.534	11	0.505	11	0.405	11	0.257
	7	0.178	7	0.516	14	0.138	7	0.214	14	0.090	7	0.285	7	0.245	7	0.208	9	0.191
	9	0.122	14	0.516	7	0.129	14	0.210	15	0.072	14	0.246	14	0.237	9	0.179	7	0.176
#413 Melody, Concord	27	0.294	26	0.736	27	0.311	26	0.428	26	0.988	26	0.420	26	0.450	26	0.328	27	0.641
	9	0.243	9	0.691	9	0.272	9	0.389	27	0.009	9	0.397	9	0.422	9	0.286	2	0.112
	26	0.189	13	0.686	26	0.233	15	0.378	9	0.002	15	0.388	15	0.412	15	0.277	20	0.112
#466 Measurement	9	0.577	9	0.760	9	0.538	9	0.588	9	0.657	9	0.555	9	0.612	9	0.496	9	0.564
	15	0.102	13	0.756	15	0.527	15	0.580	2	0.310	15	0.549	15	0.605	15	0.488	27	0.135
	29	0.102	14	0.756	29	0.527	29	0.580	20	0.310	29	0.549	29	0.605	29	0.488	15	0.079
#590 Writing	9	0.370	26	0.797	9	0.376	9	0.433	27	0.525	9	0.521	9	0.410	9	0.414	27	0.508
	27	0.358	9	0.796	27	0.338	15	0.433	26	0.458	15	0.521	15	0.410	15	0.414	9	0.301
	15	0.268	15	0.796	15	0.333	29	0.433	2	0.014	29	0.521	29	0.410	29	0.414	15	0.214
#597 Poetry	27	0.332	26	0.806	26	0.418	26	0.454	26	0.860	26	0.491	26	0.524	26	0.448	27	0.528
	26	0.273	5	0.794	9	0.310	9	0.365	27	0.134	9	0.404	9	0.427	9	0.334	26	0.163
	9	0.150	11	0.767	5	0.285	15	0.349	5	0.004	15	0.404	15	0.427	15	0.318	5	0.116
#625 Business	19	0.516	19	0.734	19	0.332	19	0.457	26	0.969	19	0.479	19	0.514	19	0.571	27	0.229
	26	0.136	26	0.488	9	0.212	26	0.226	27	0.023	26	0.244	26	0.226	26	0.170	19	0.166
	9	0.130	9	0.472	26	0.205	9	0.189	9	0.003	9	0.215	9	0.195	9	0.141	2	0.162

Figure A.4: Comparison Results

	(1a)	(1b)	(1c)	(1d)	(1e)	(1f)	(1g)	(1h)	(1i)
% Matching Results	80%	50%	70%	60%	20%	60%	50%	60%	80%

Figure A.5: Percentages of the number of selections which match the first selections (the sense with the highest measure) from (Resnik, 1995a)

	(1a)	(1b)	(1c)	(1d)	(1e)	(1f)	(1g)	(1h)	(1i)	Resnik
% Correct	60%	70%	70%	80%	50%	80%	70%	80%	60%	80%

Figure A.6: Percentages of the number of selected senses that match with manually selected tags

Appendix B

Data and Scatter Graphs for Human Similarity Judgement Correlation

B.1 Human Judgement Data

B.1.1 Rubenstein and Goodenough (1965) Human Judgements

Word 1	Word 2	Similarity
cord	smile	0.02
rooster	voyage	0.04
noon	string	0.04
fruit	furnace	0.05
autograph	shore	0.06
automobile	wizard	0.11
mound	stove	0.14
grin	implement	0.18
asylum	fruit	0.19
asylum	monk	0.39
graveyard	madhouse	0.42
glass	magician	0.44
boy	rooster	0.44
cushion	jewel	0.45
monk	slave	0.57
asylum	cemetery	0.79
coast	forest	0.85
grin	lad	0.88

shore	woodland	0.9
monk	oracle	0.91
boy	sage	0.96
automobile	cushion	0.97
mound	shore	0.97
lad	wizard	0.99
forest	graveyard	1
food	rooster	1.09
cemetery	woodland	1.18
shore	voyage	1.22
bird	woodland	1.24
coast	hill	1.26
furnace	implement	1.37
crane	rooster	1.41
hill	woodland	1.48
car	journey	1.55
cemetery	mound	1.69
glass	jewel	1.78
magician	oracle	1.82
crane	implement	2.37
brother	lad	2.41
sage	wizard	2.46
oracle	sage	2.61
bird	crane	2.63
bird	cock	2.63
food	fruit	2.69
brother	monk	2.74
asylum	madhouse	3.04
furnace	stove	3.14
magician	wizard	3.21
hill	mound	3.29
cord	string	3.41
glass	tumbler	3.45
grin	smile	3.46
serf	slave	3.46
journey	voyage	3.58
autograph	signature	3.59
coast	shore	3.6
forest	woodland	3.65
implement	tool	3.66
cock	rooster	3.68
boy	lad	3.82
cushion	pillow	3.84

cemetery	graveyard	3.88
automobile	car	3.92
midday	noon	3.94
gem	jewel	3.94

Table B.1: Rubenstein and Goodenough (1965) Human Judgements

B.1.2 Miller and Charles (1991) Human Judgements

Word 1	Word 2	Similarity
rooster	voyage	0.08
noon	string	0.08
glass	magician	0.11
chord	smile	0.13
lad	wizard	0.42
coast	forest	0.42
monk	slave	0.55
shore	woodland	0.63
forest	graveyard	0.84
coast	hill	0.87
food	rooster	0.89
cemetery	woodland	0.95
monk	oracle	1.1
journey	car	1.16
lad	brother	1.66
crane	implement	1.68
brother	monk	2.82
tool	implement	2.95
bird	crane	2.97
bird	cock	3.05
food	fruit	3.08
furnace	stove	3.11

midday	noon	3.42
magician	wizard	3.5
asylum	madhouse	3.61
coast	shore	3.7
boy	lad	3.76
journey	voyage	3.84
gem	jewel	3.84
automobile	car	3.92

Table B.2: Miller and Charles (1991) Human Judgements

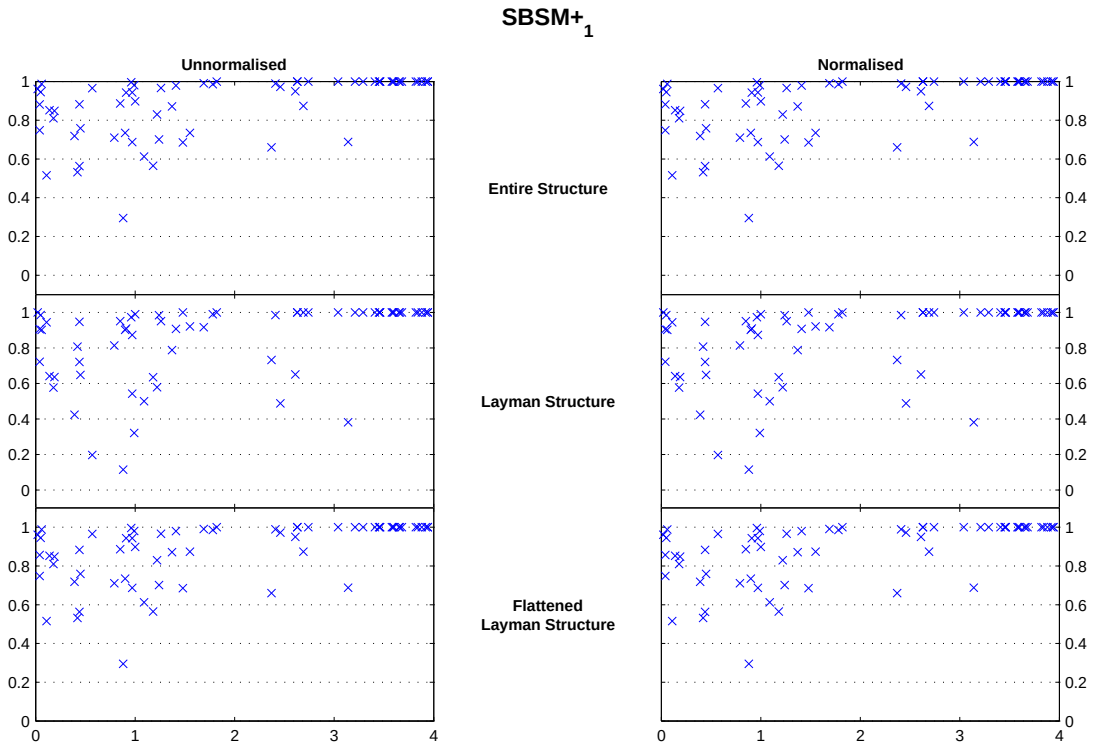
B.1.3 Resnik (1999) Human Judgements

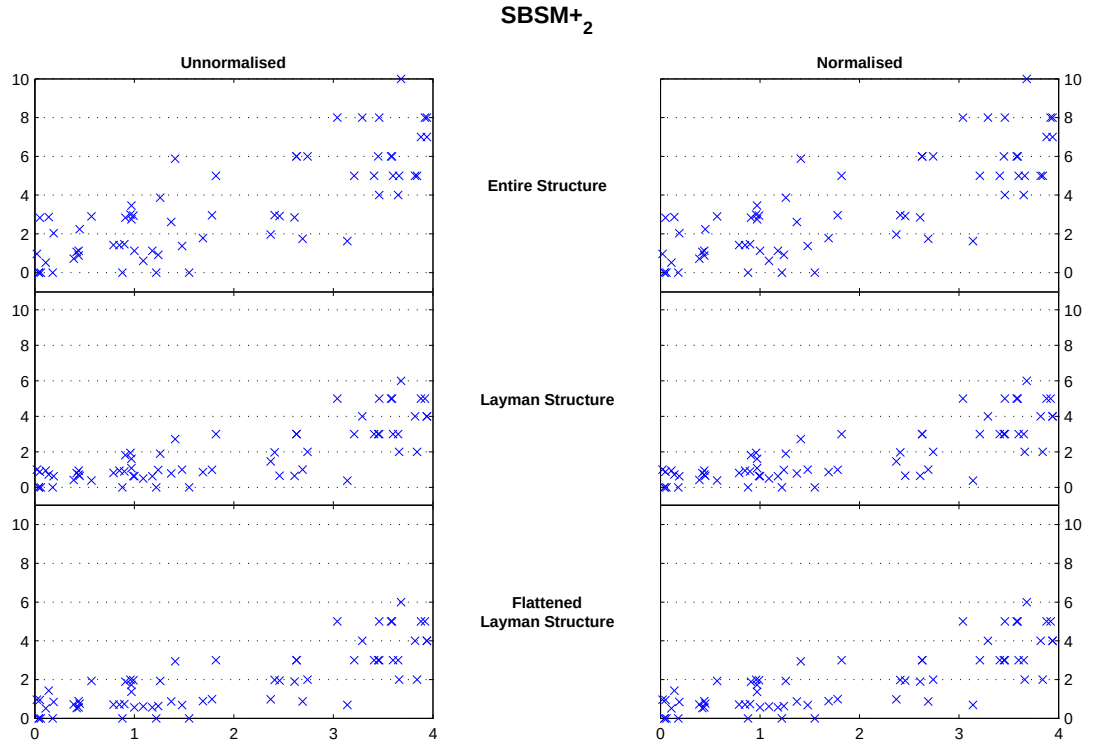
Word 1	Word 2	Similarity
rooster	voyage	0
noon	string	0
glass	magician	0.1
chord	smile	0.1
crane	implement	0.3
coast	forest	0.6
forest	graveyard	0.6
lad	wizard	0.7
monk	slave	0.7
coast	hill	0.7
journey	car	0.7
monk	oracle	0.8
food	rooster	1.1
lad	brother	1.2
bird	cock	2.1
furnace	stove	2.1
food	fruit	2.2
brother	monk	2.4

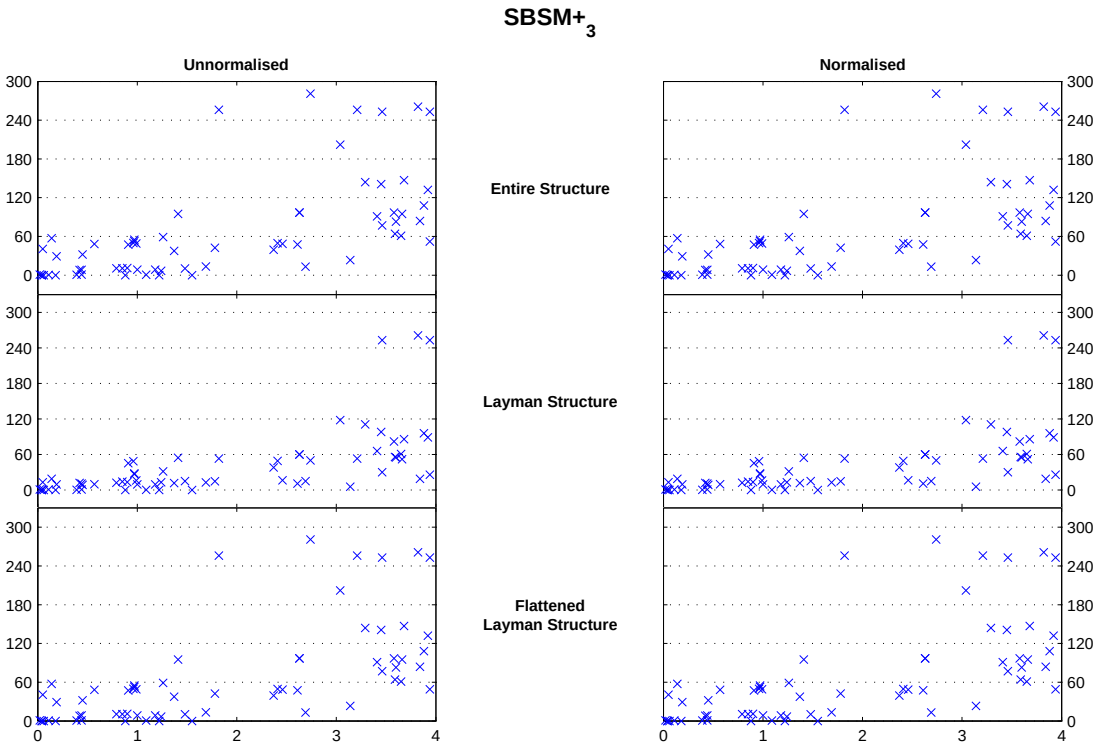
tool	implement	2.4
midday	noon	2.6
bird	crane	3.4
magician	wizard	3.6
asylum	madhouse	3.5
coast	shore	3.5
boy	lad	3.5
journey	voyage	3.5
gem	jewel	3.5
automobile	car	3.9

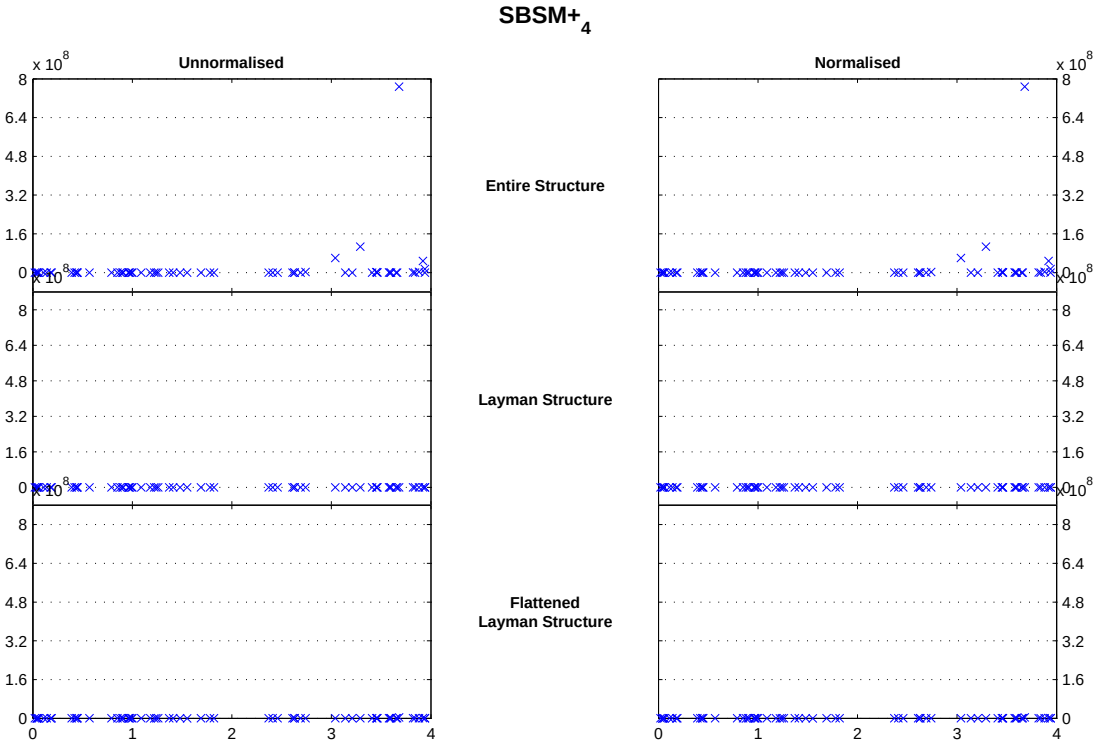
Table B.3: Resnik (1999) Human Judgements

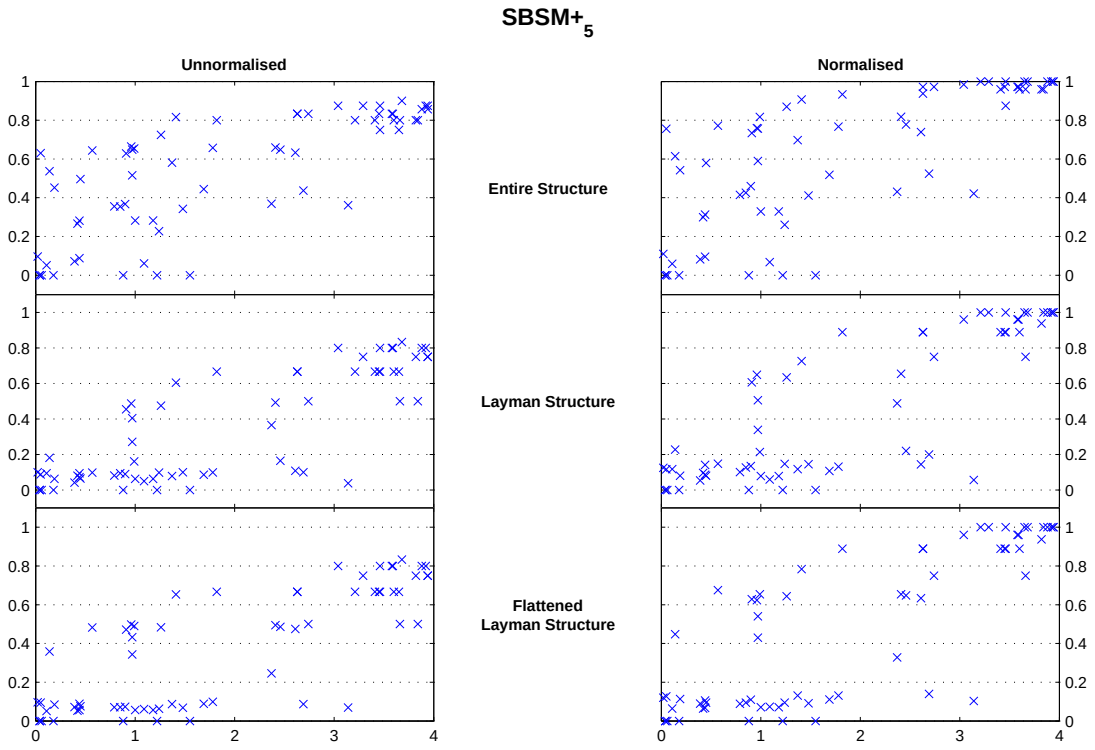
B.2 Rubenstein & Goodenough Human Judgement Correlations

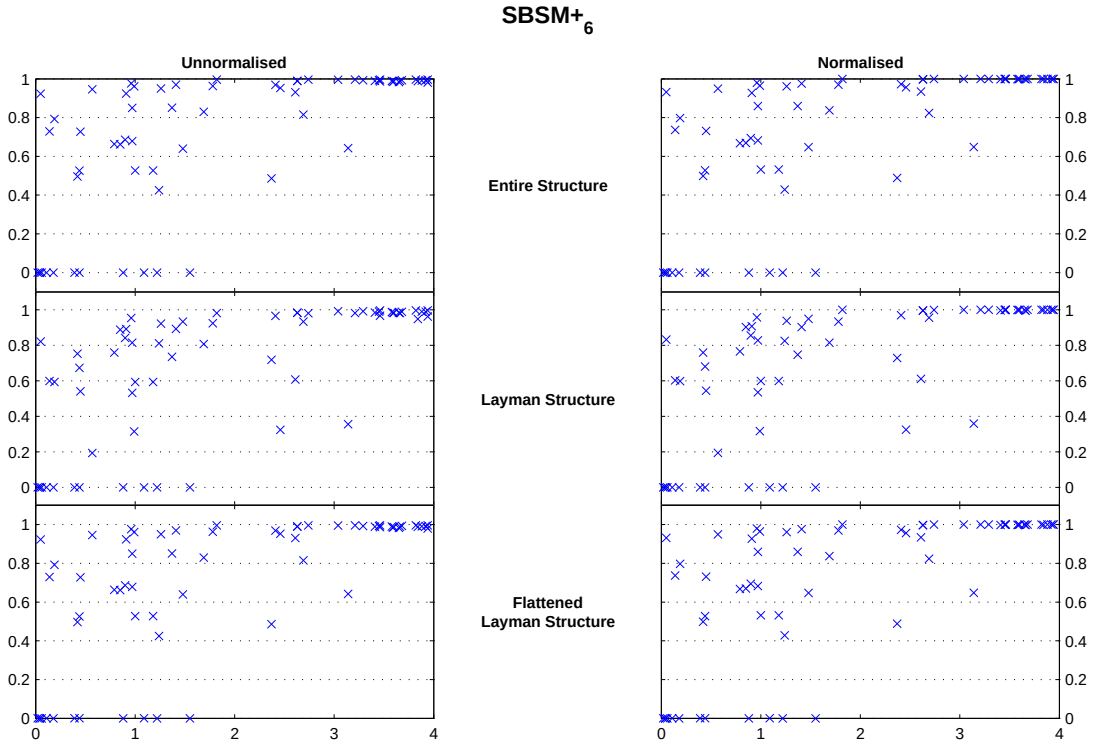

SBSM+1
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

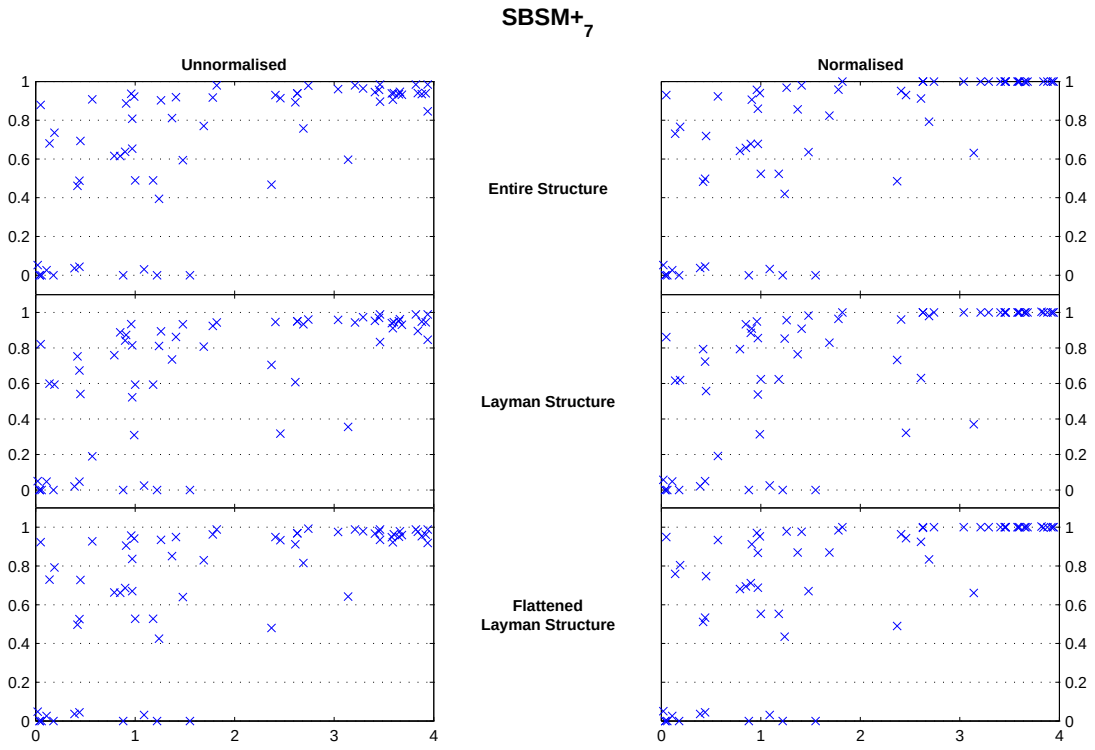

SBSM+2
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

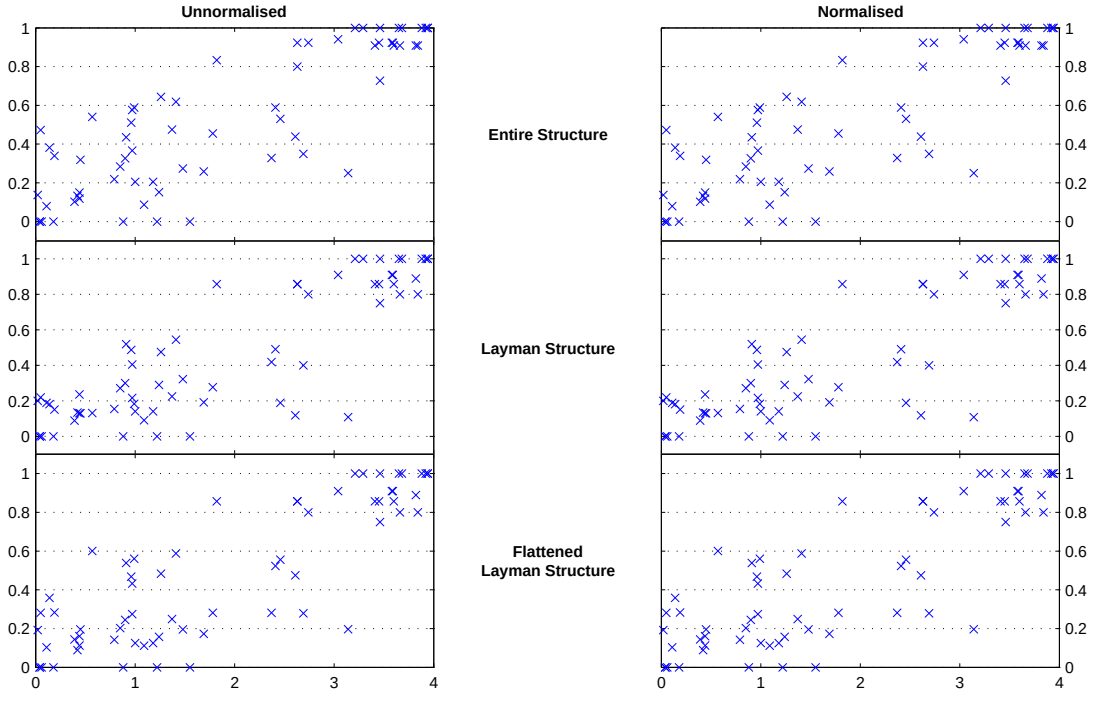

SBSM+3
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

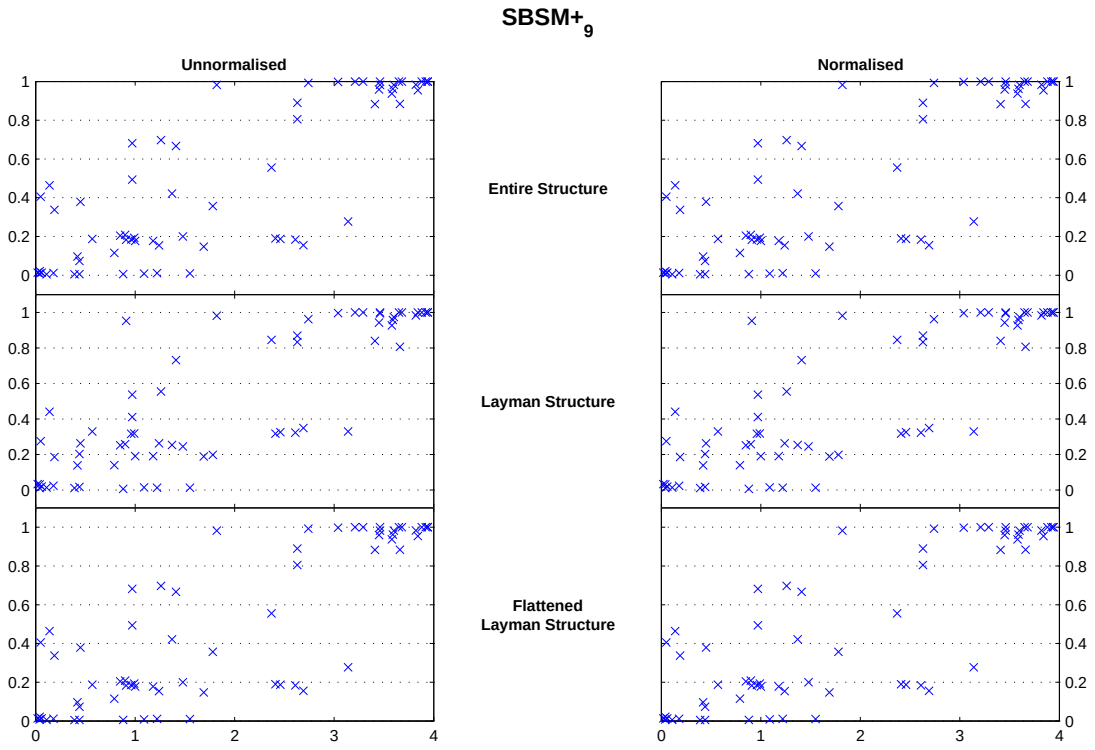

SBSM+4
Unnormalised
Normalised
x 10^8
Entire Structure
Layman Structure
Flattened
Layman Structure

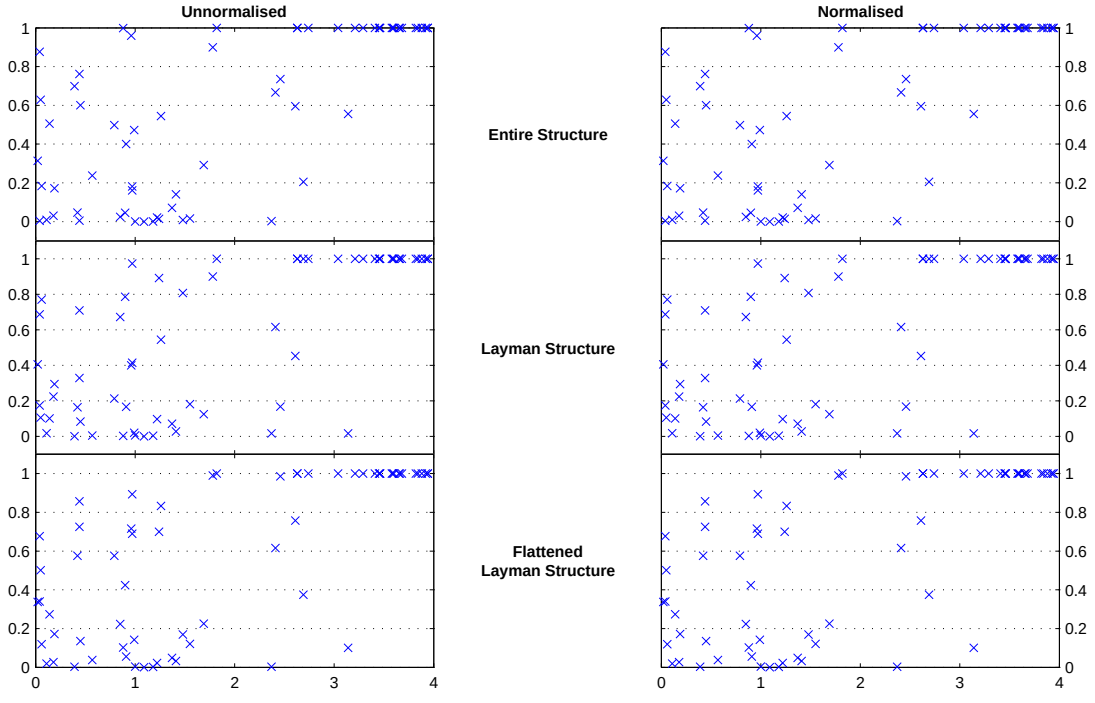

SBSM+5
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

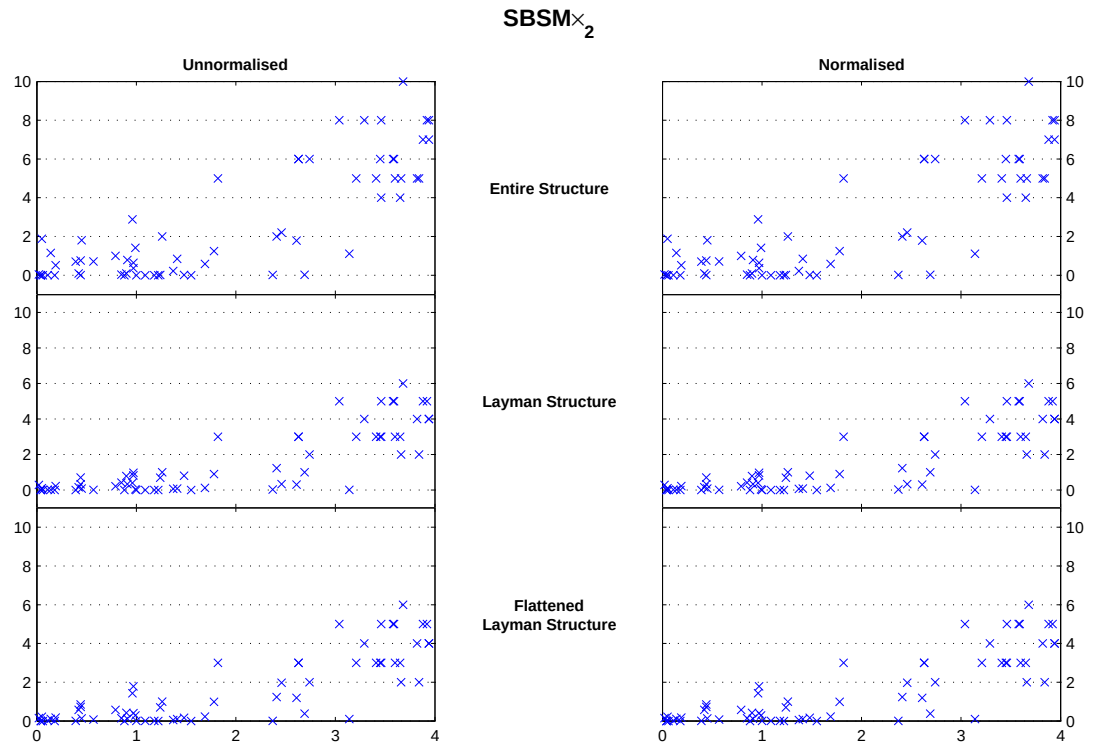

SBSM+6
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

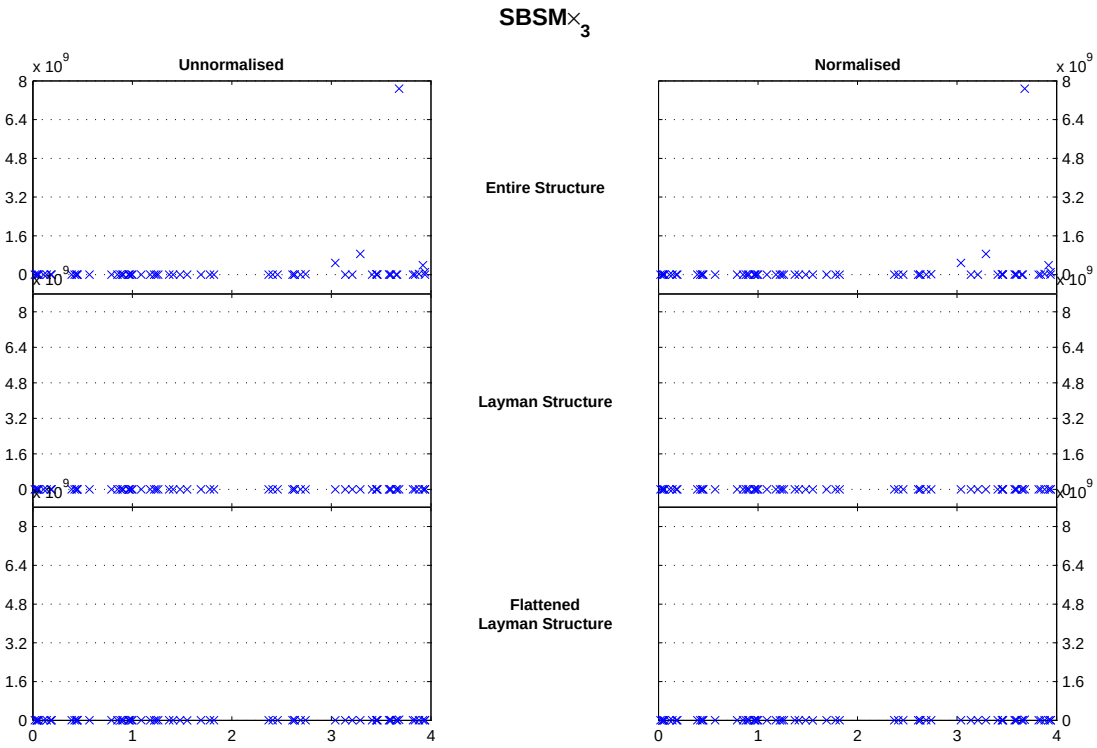

SBSM+7
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

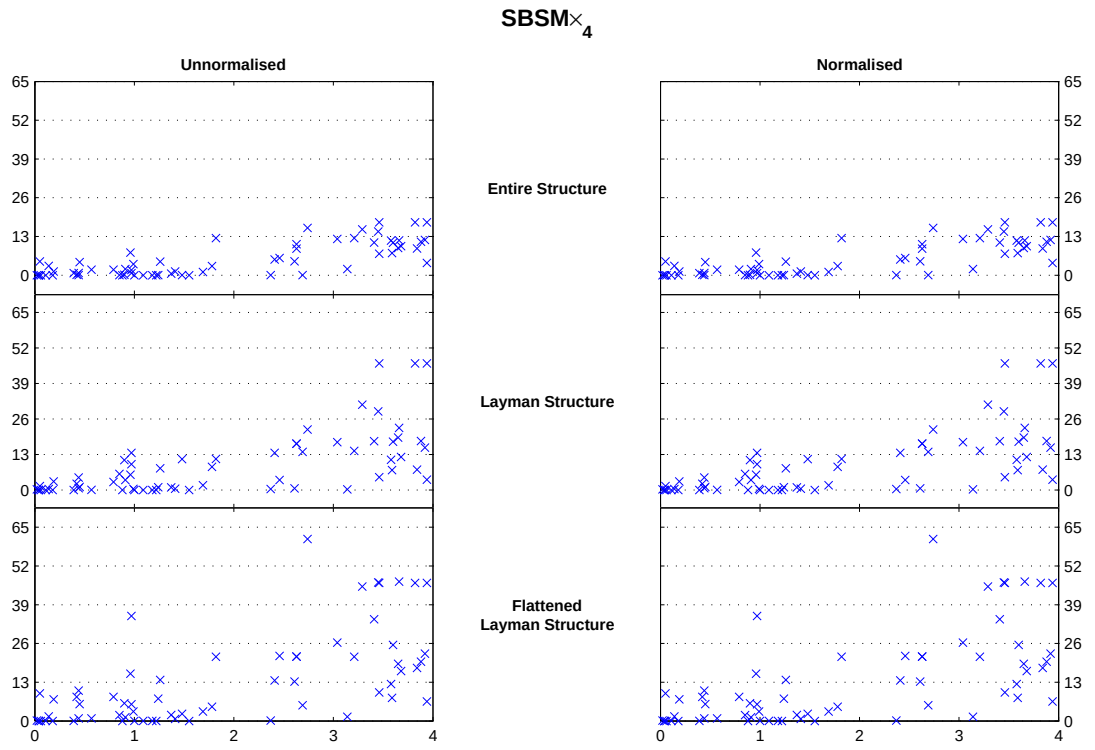

SBSM+8
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

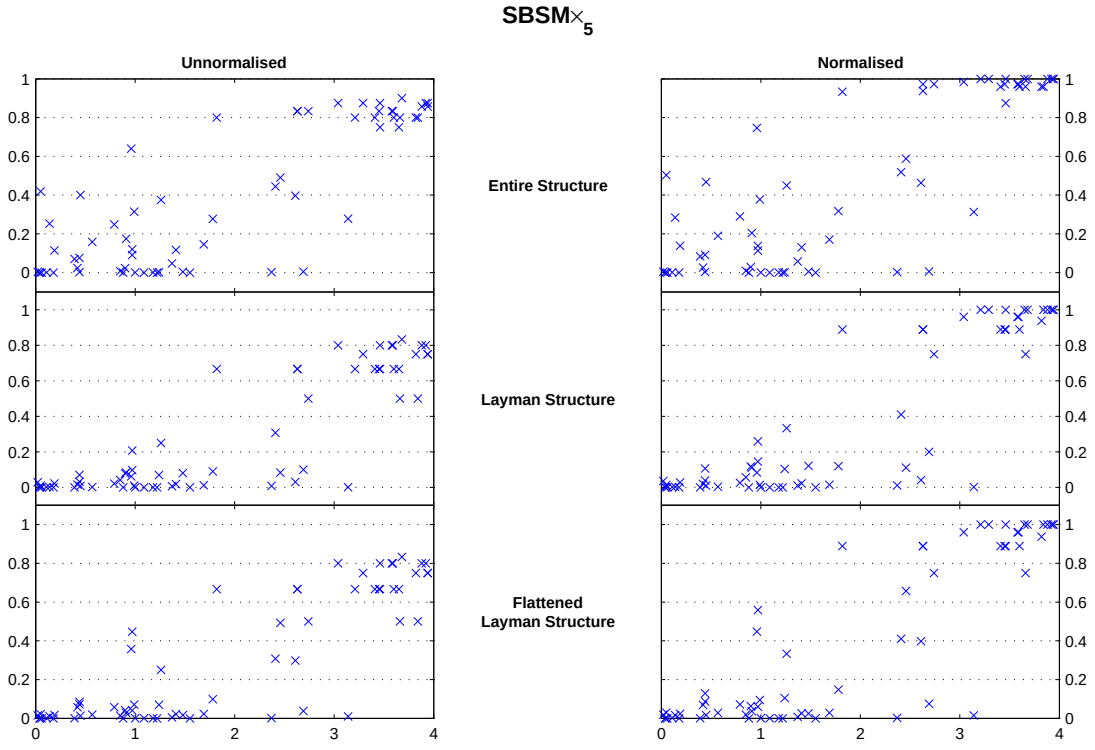

SBSM+9
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

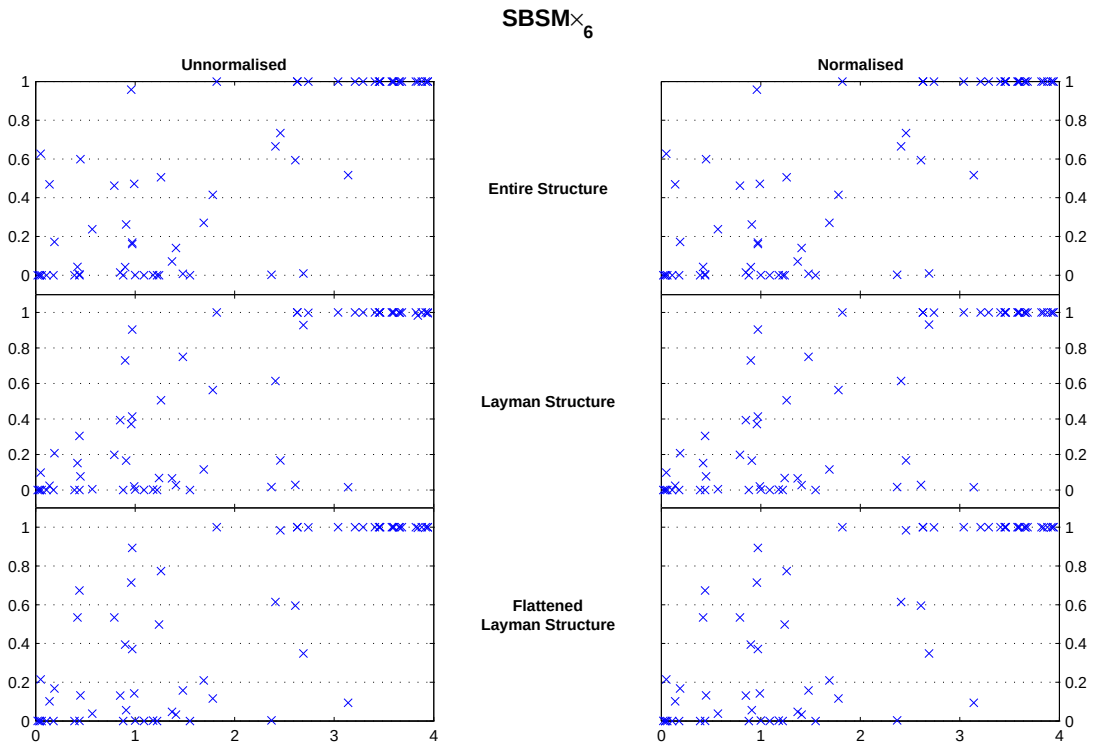

SBSM×1
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

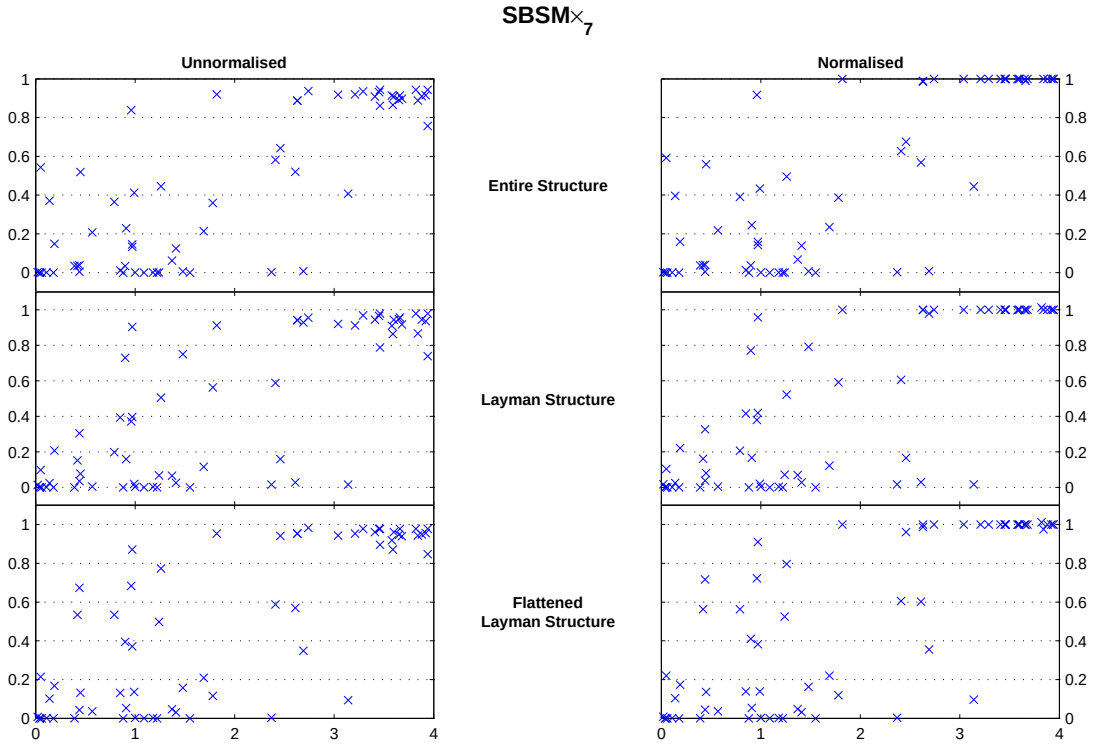

SBSM×2
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

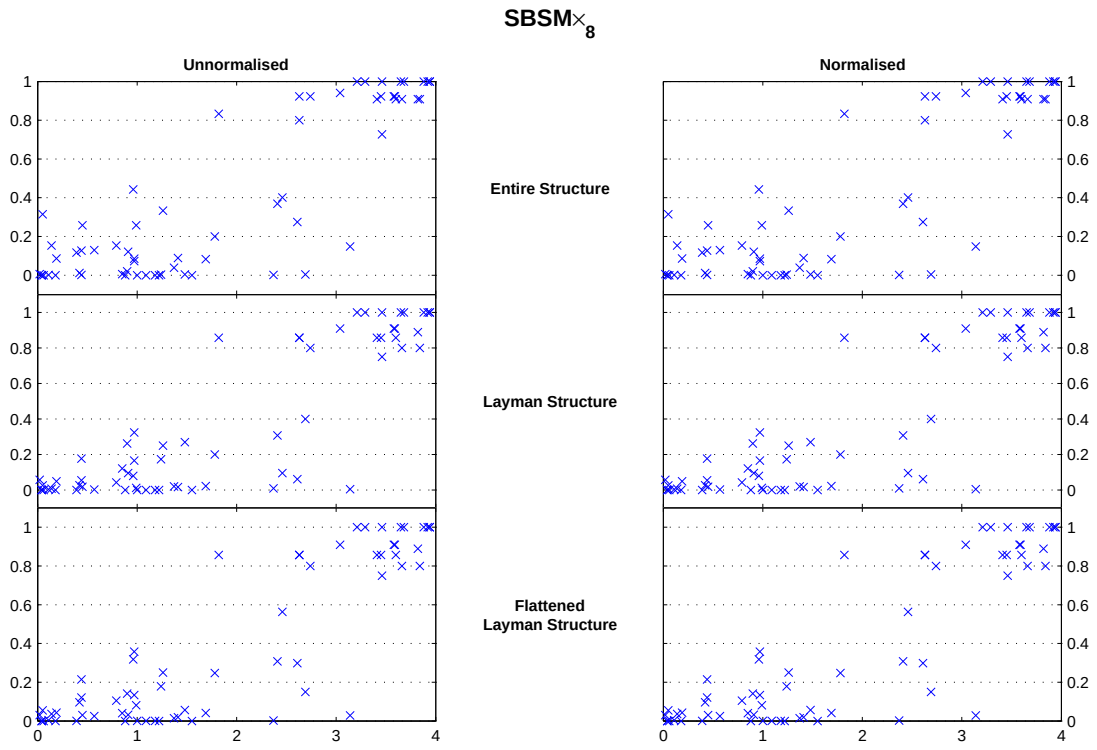

SBSM×3
Unnormalised
Normalised
x 10^9
Entire Structure
Layman Structure
Flattened
Layman Structure

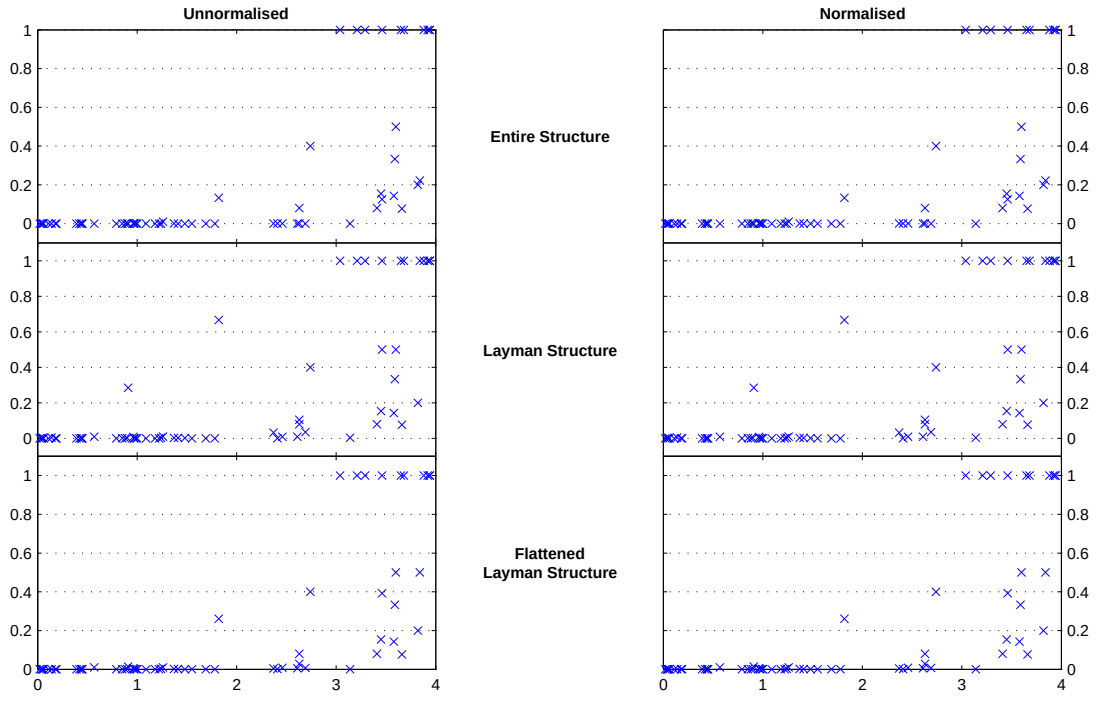

SBSM×4
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure


SBSM×5
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

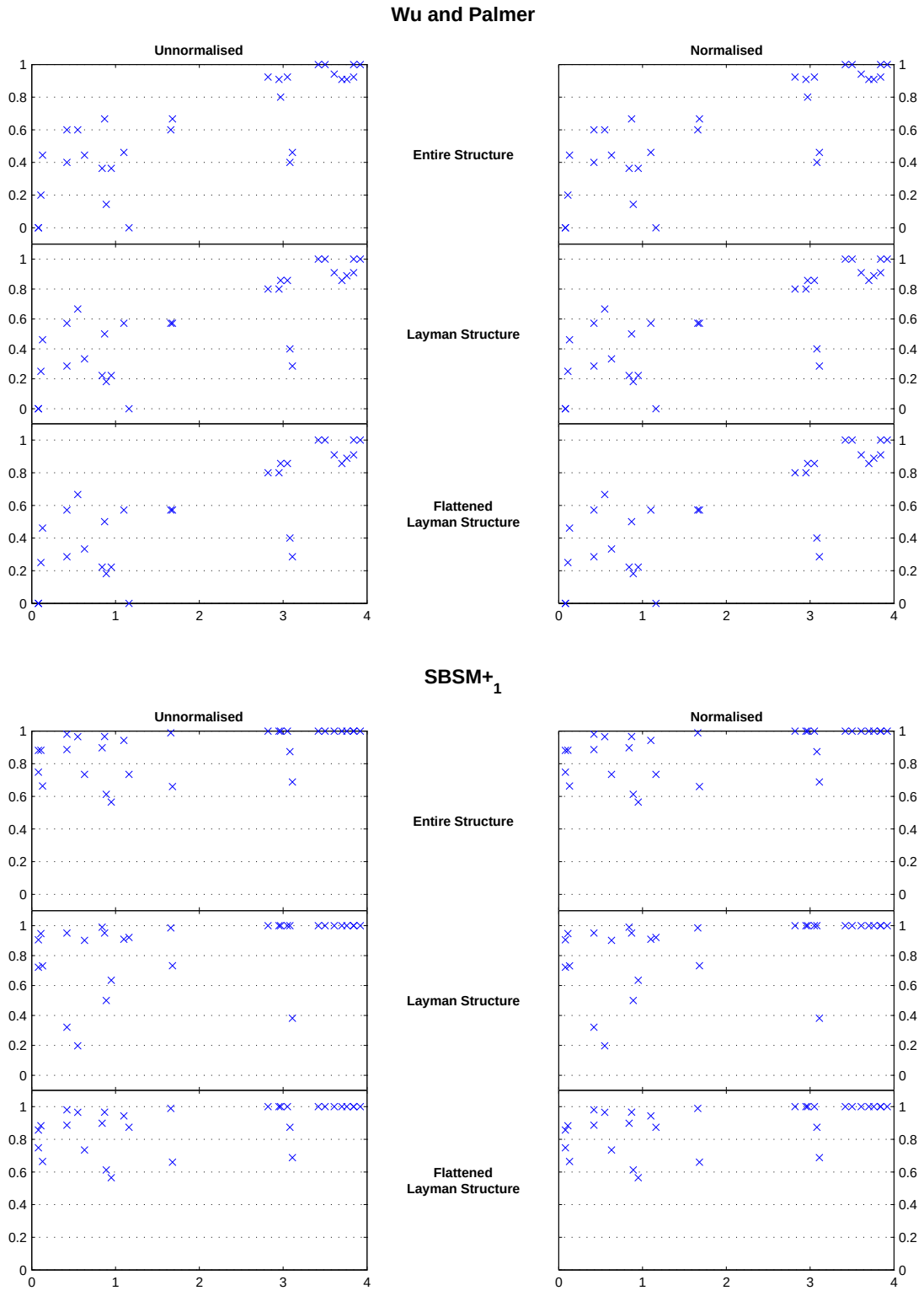

SBSM×6
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

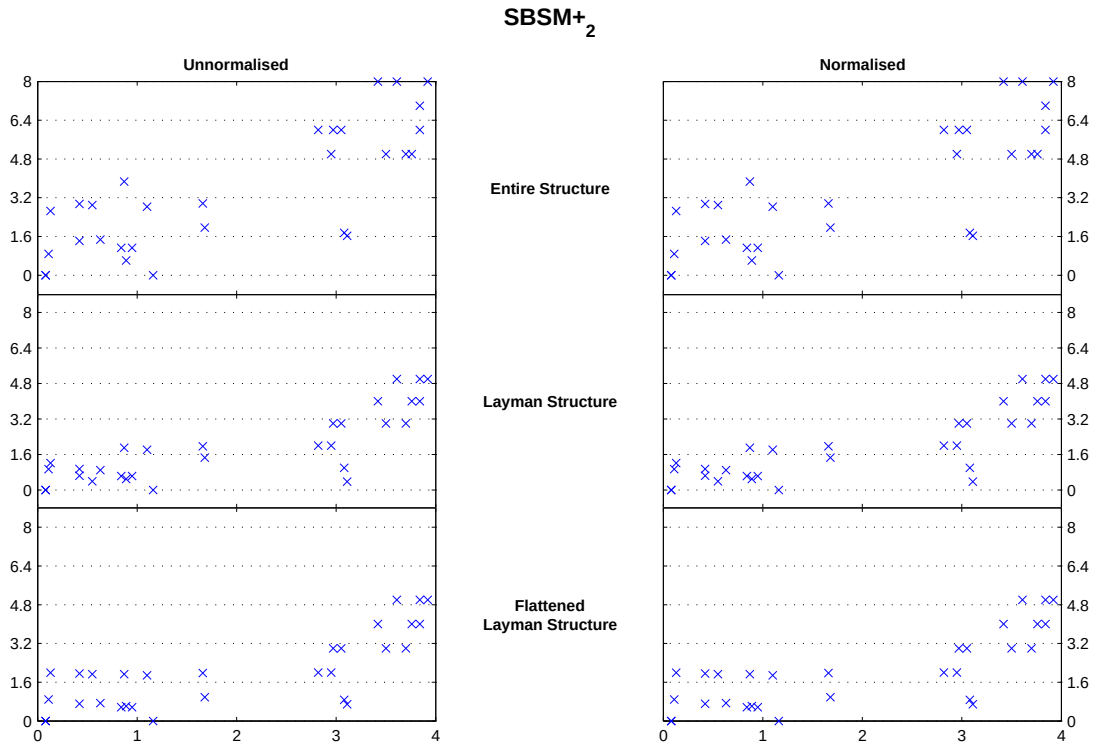

SBSM×7
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

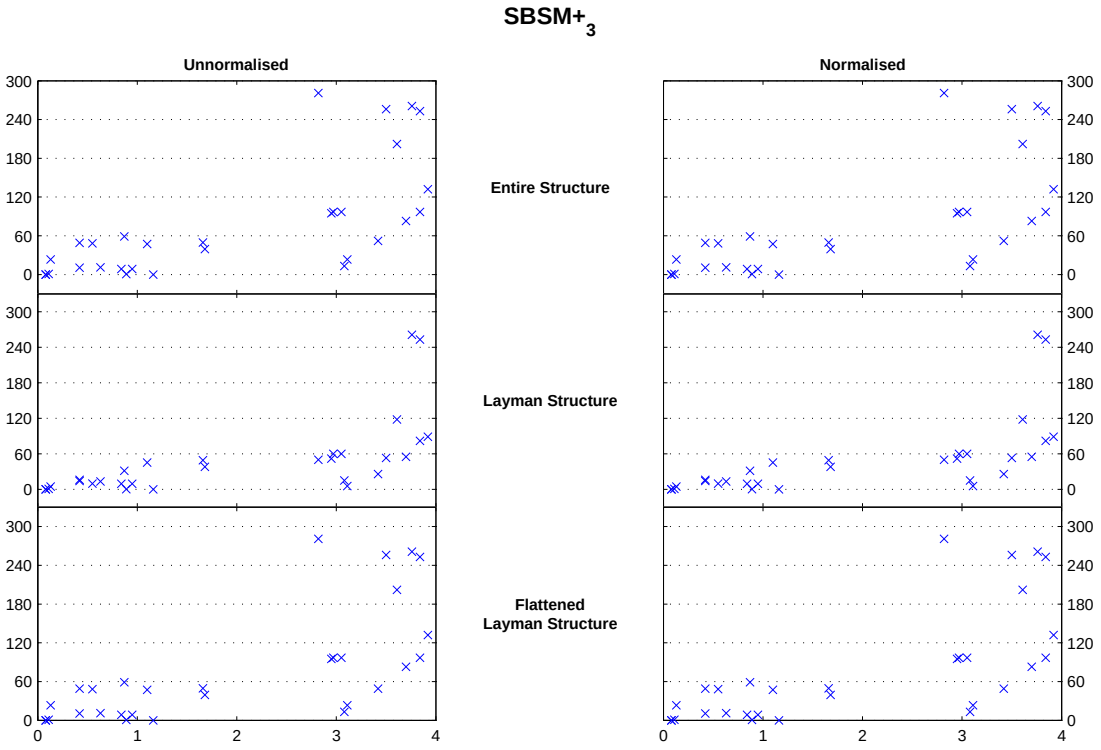

SBSM×8
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

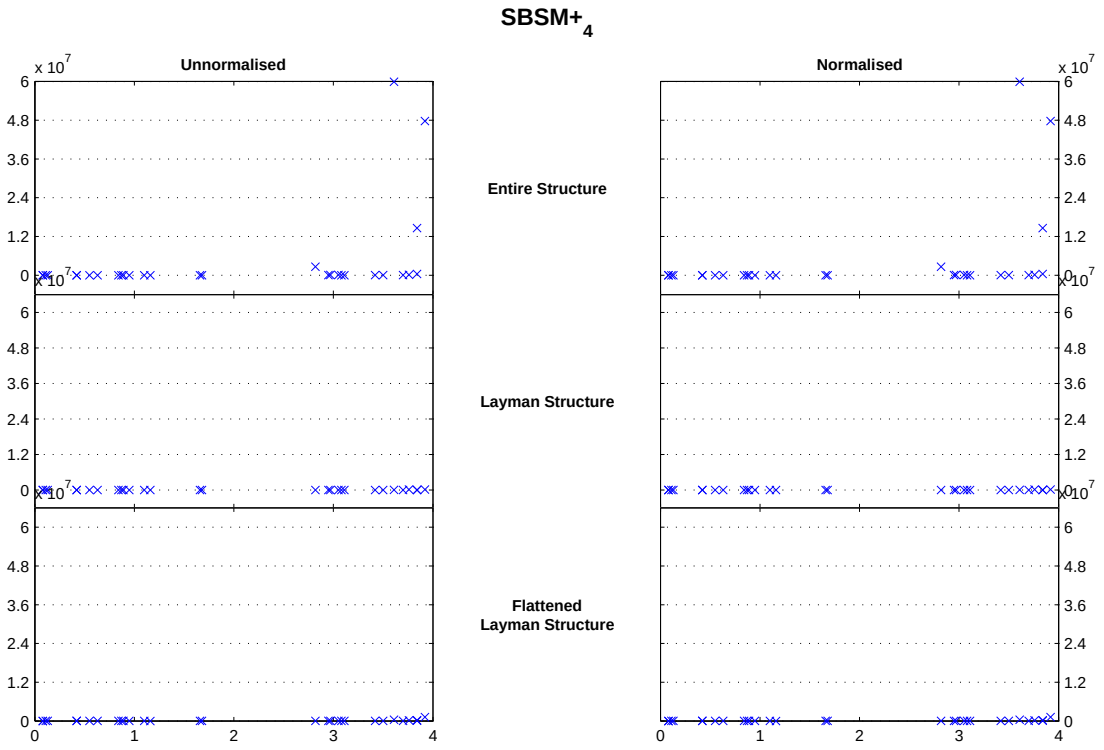

SBSM×9
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

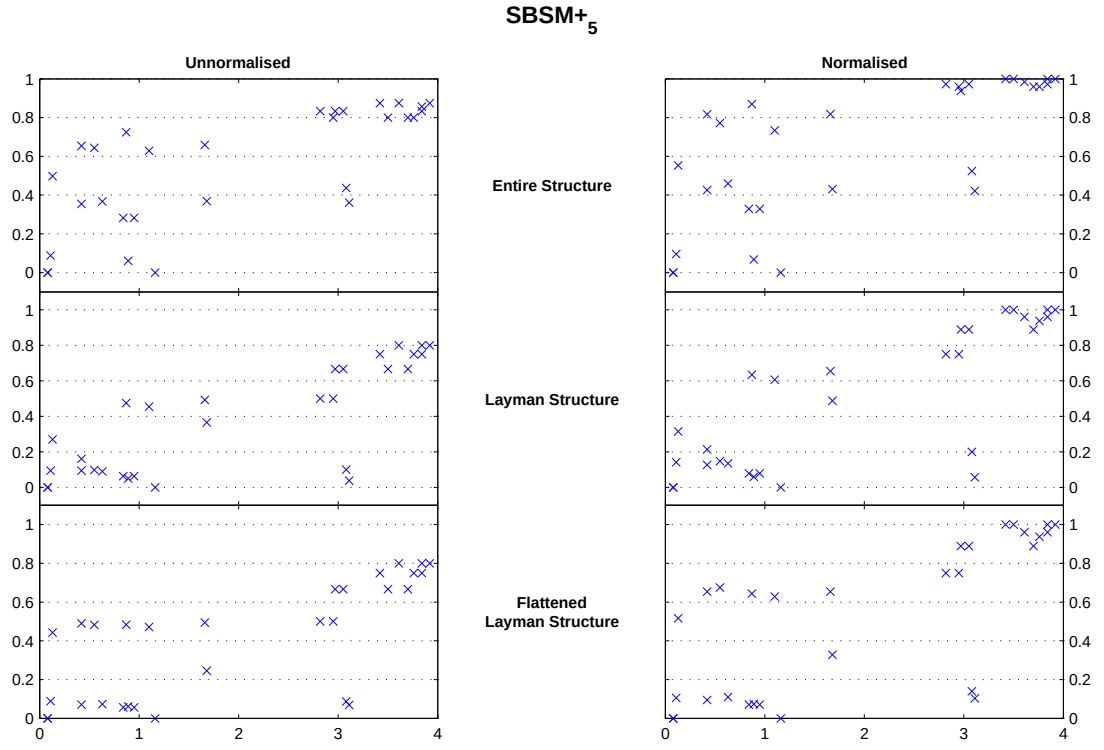
B.3 Miller & Charles Human Judgement Correlations

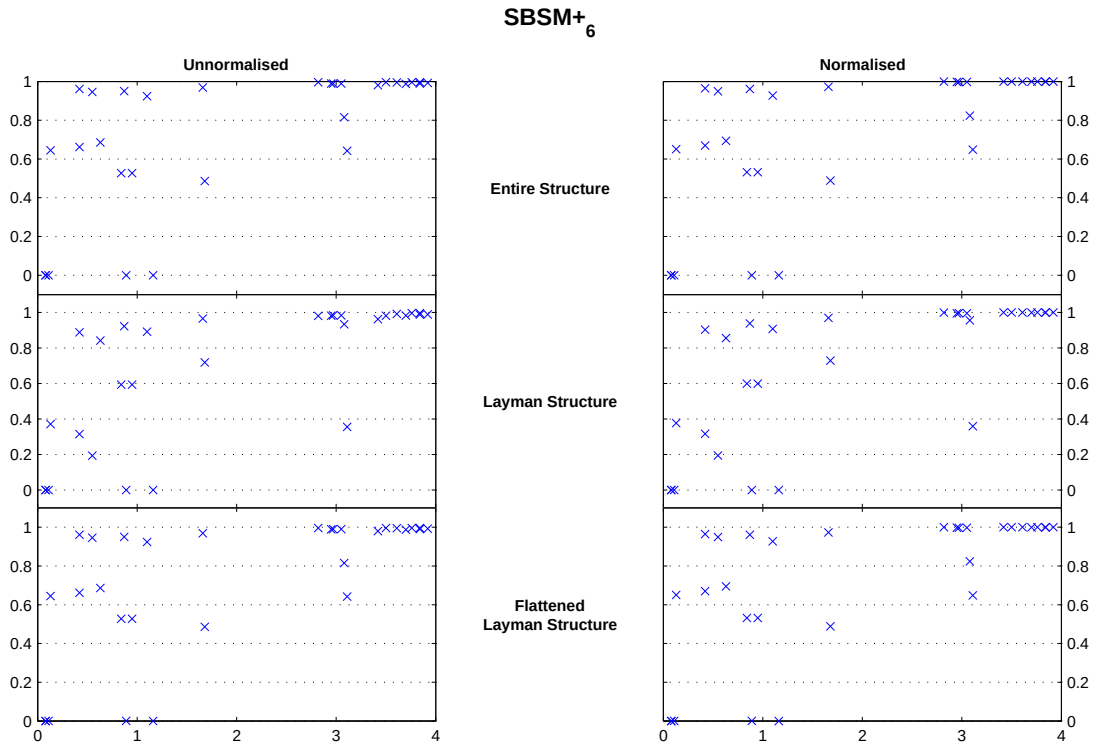


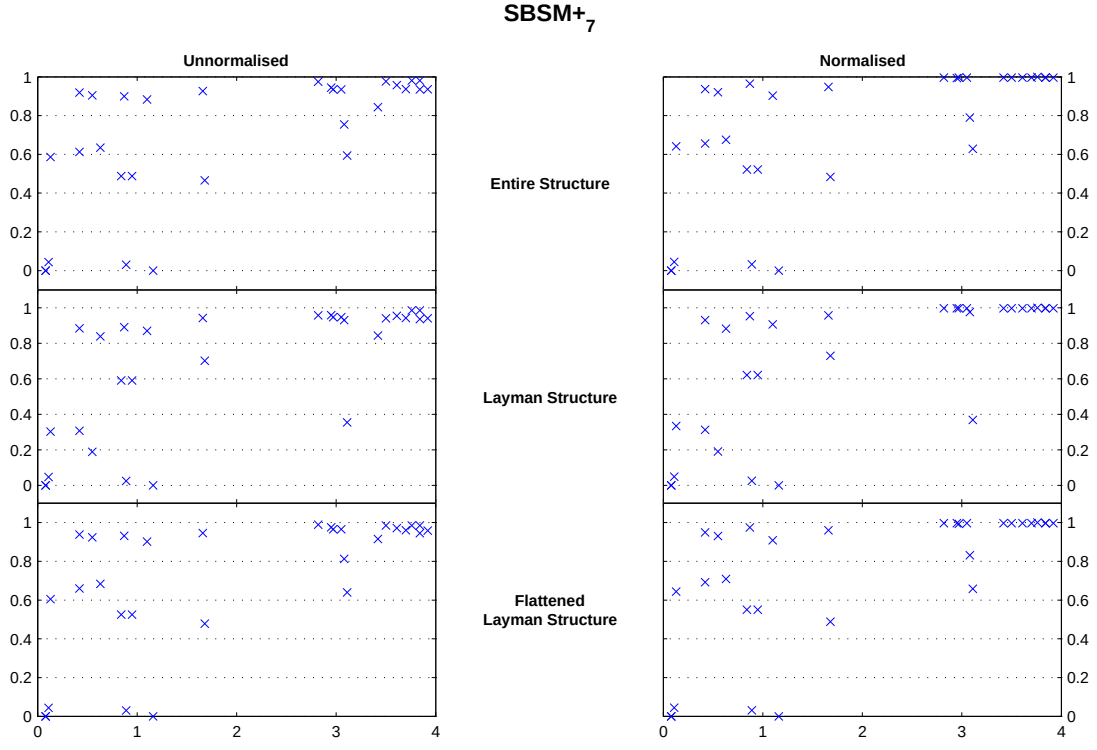

SBSM+2
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

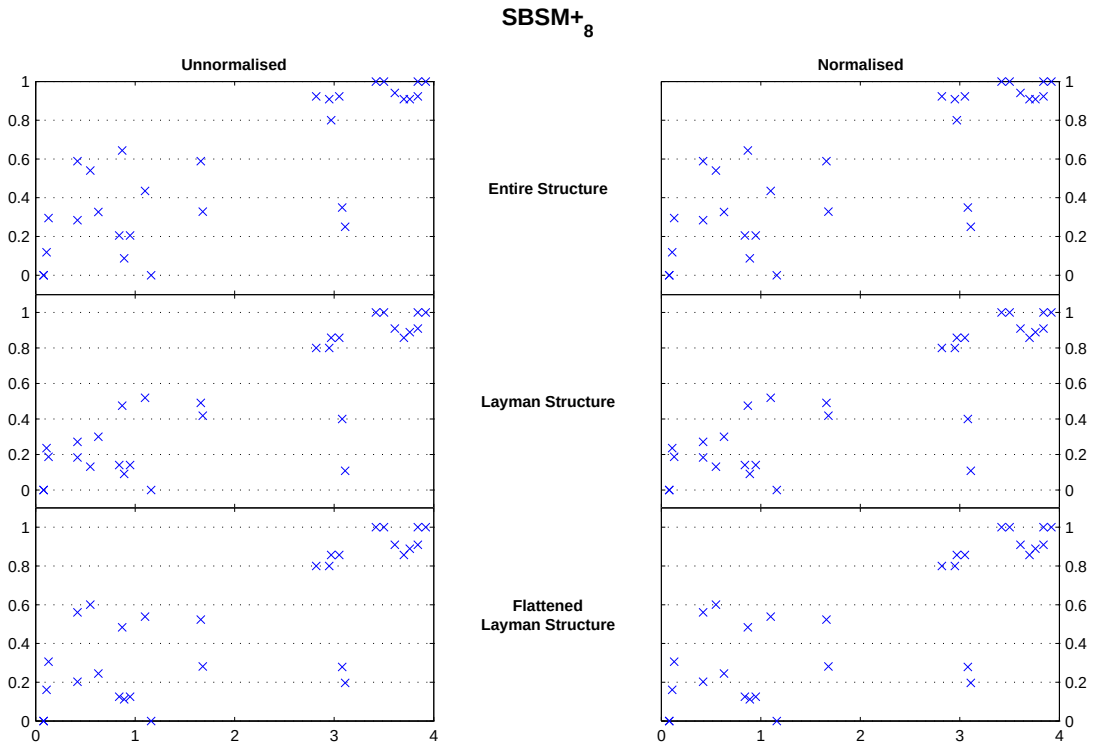

SBSM+3
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

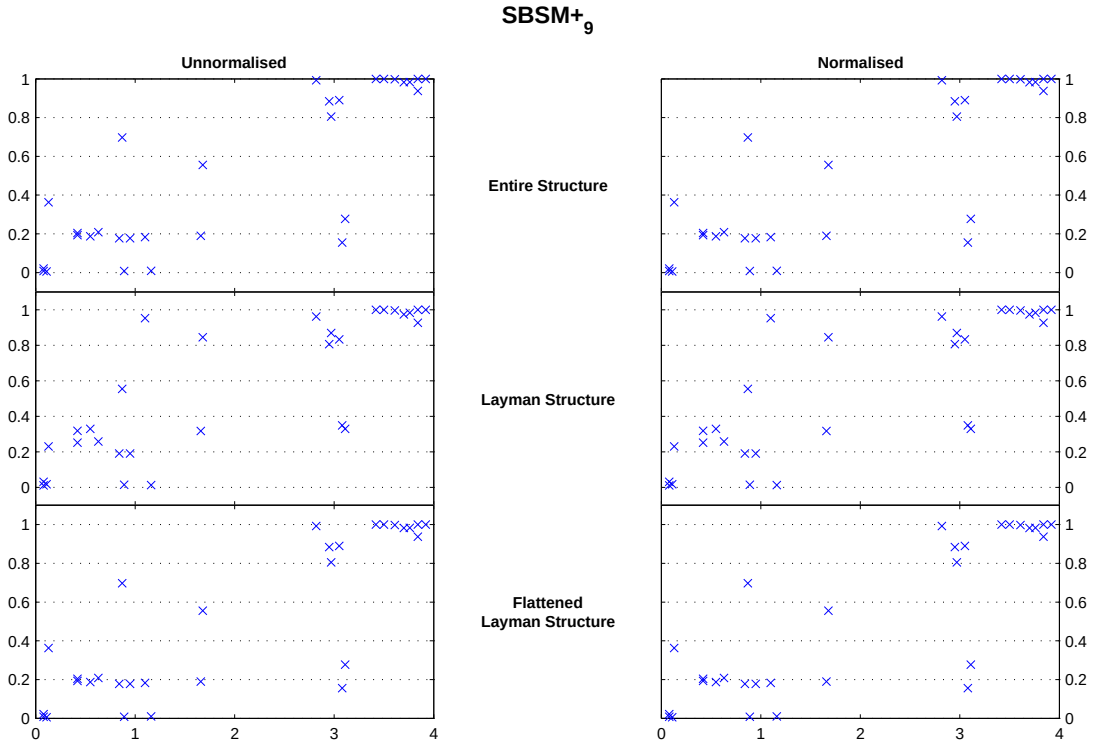

SBSM+4
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure
x 10^7

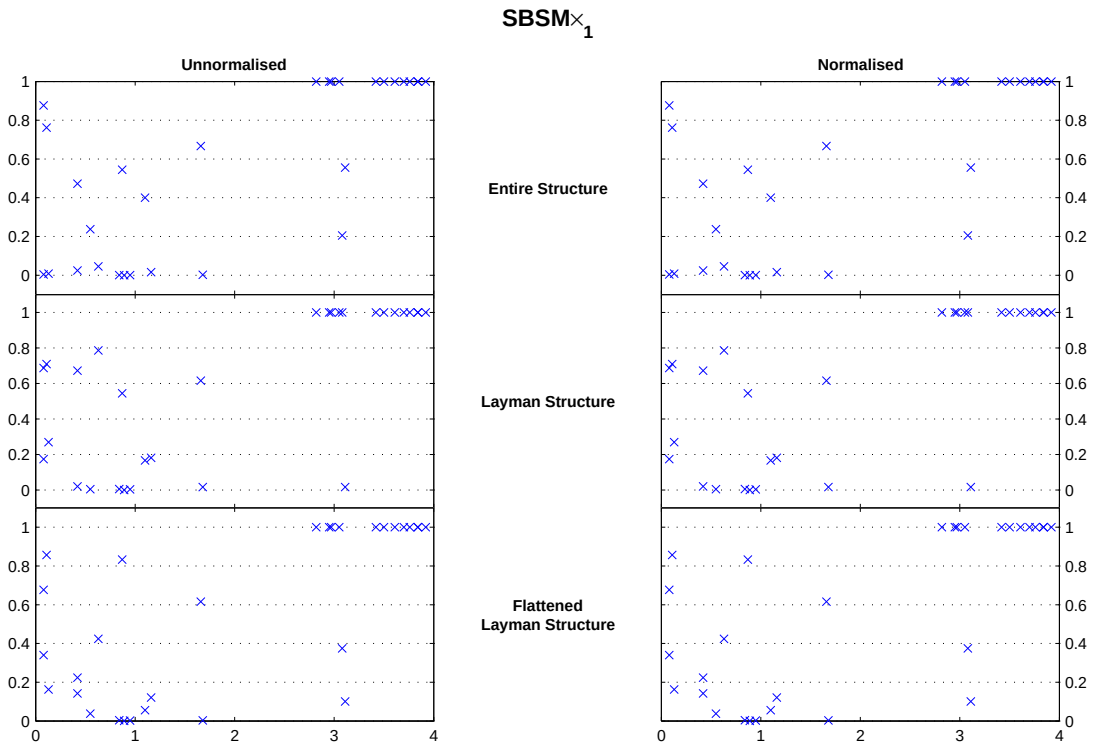

SBSM+5
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

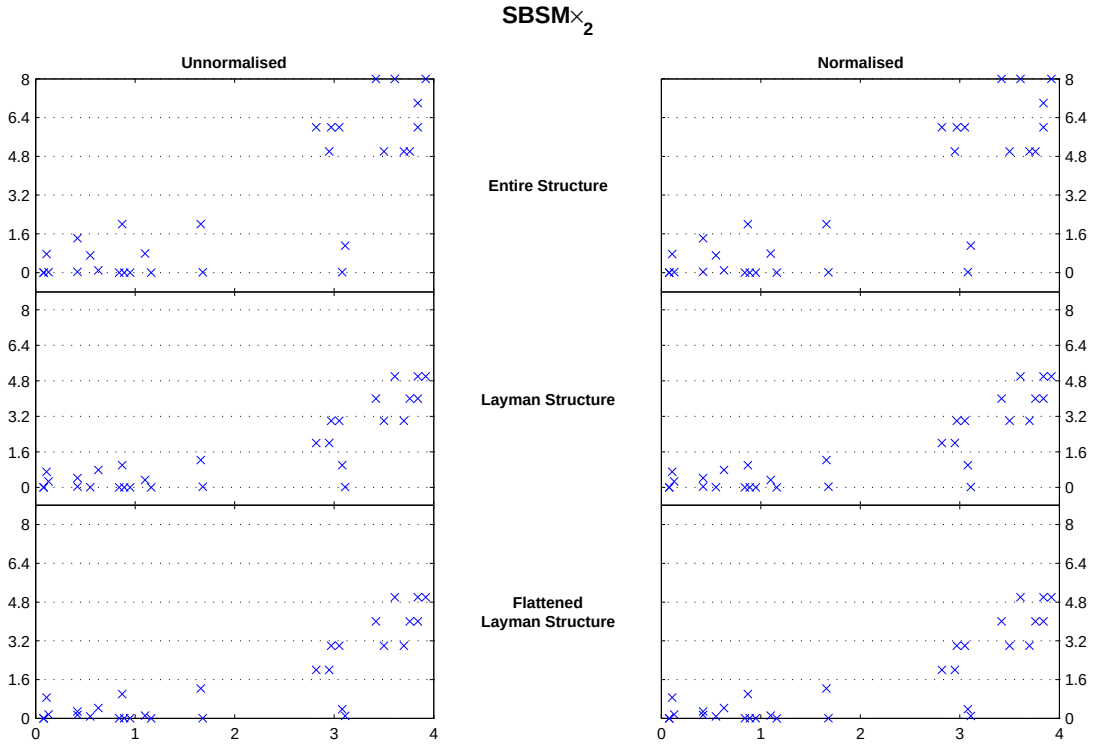

SBSM+6
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

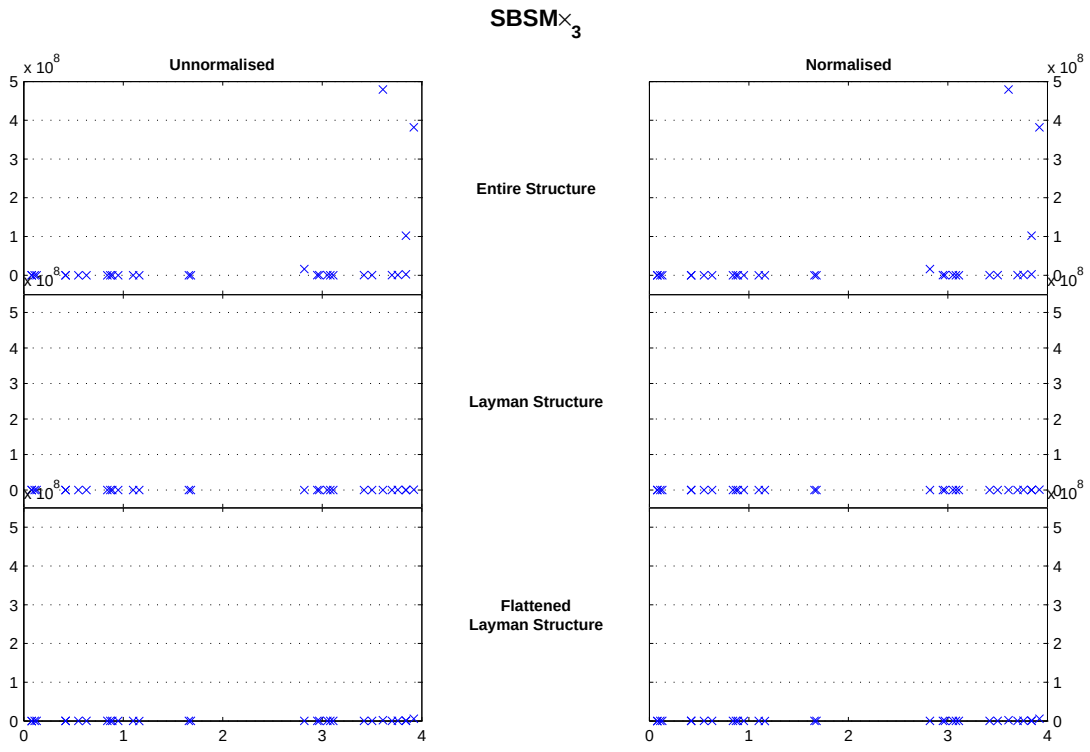

SBSM+7
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

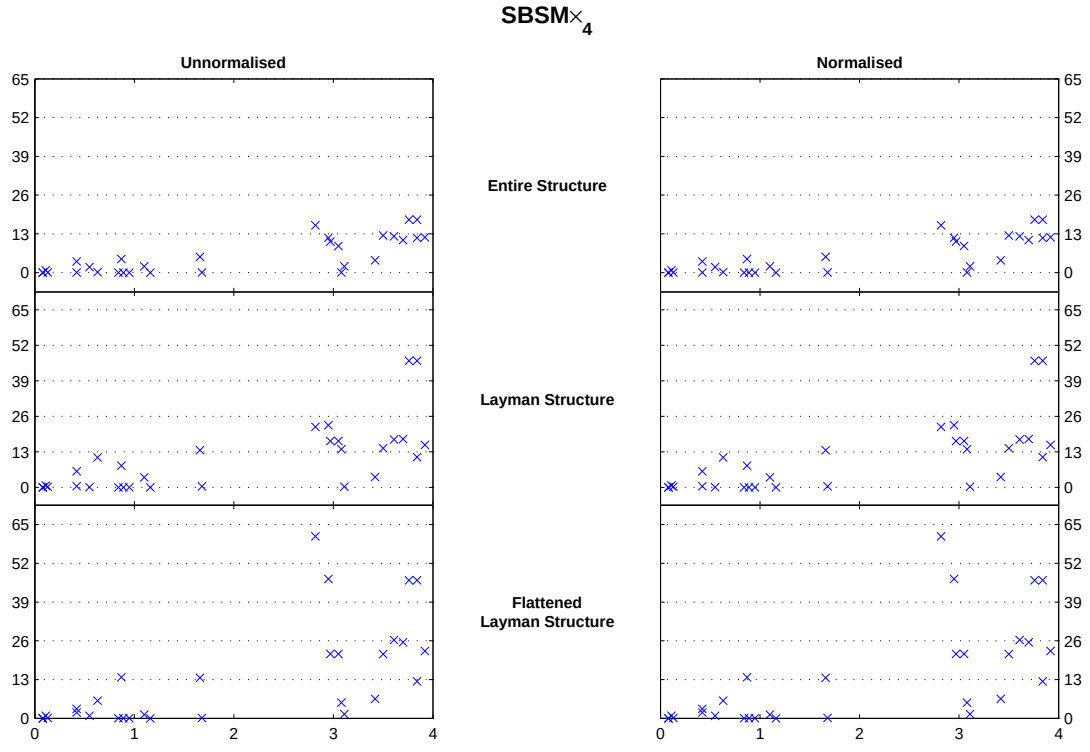

SBSM+8
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

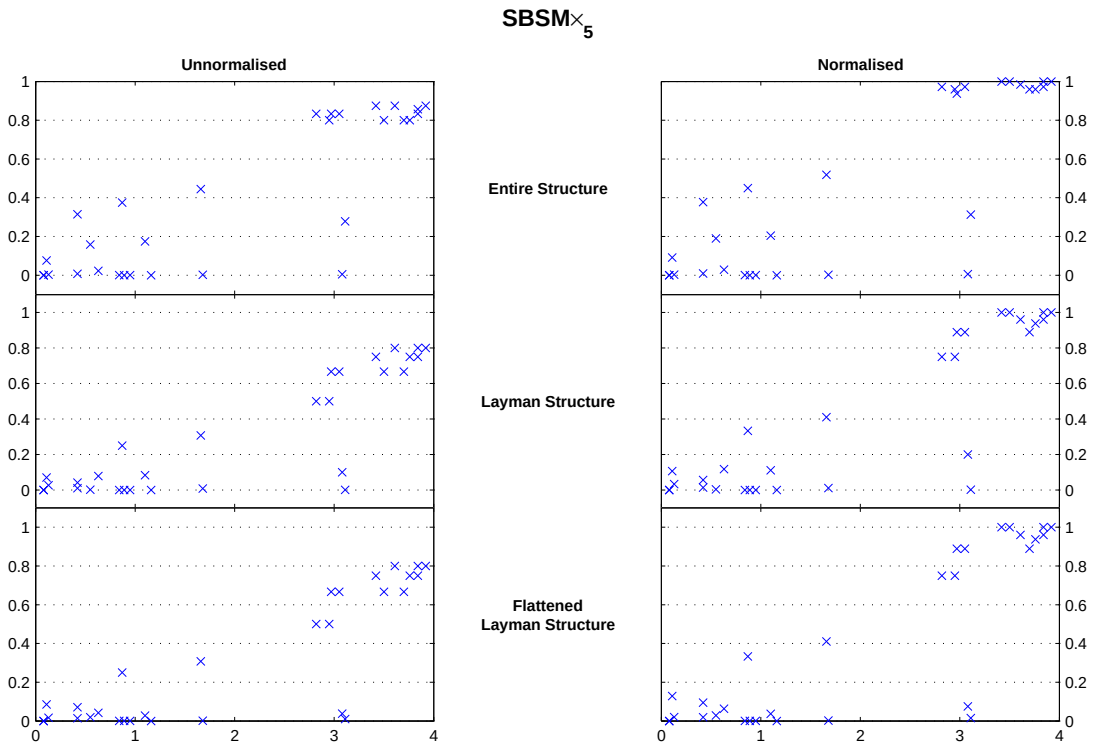

SBSM+9
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

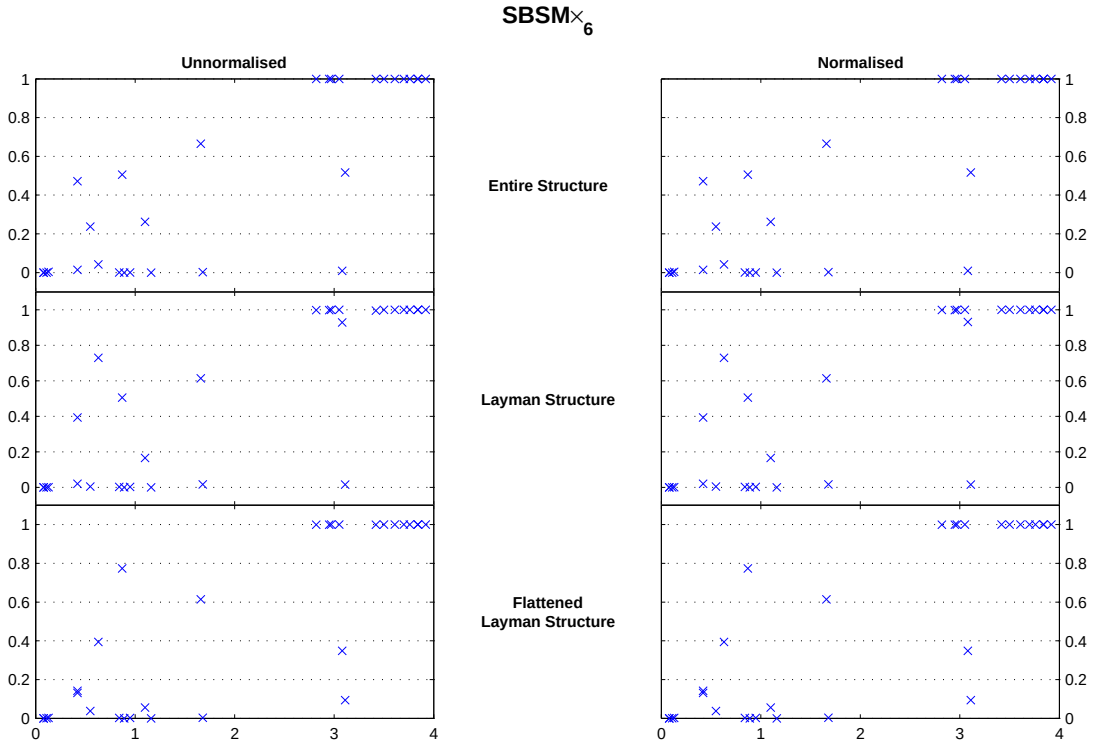

SBSM×1
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

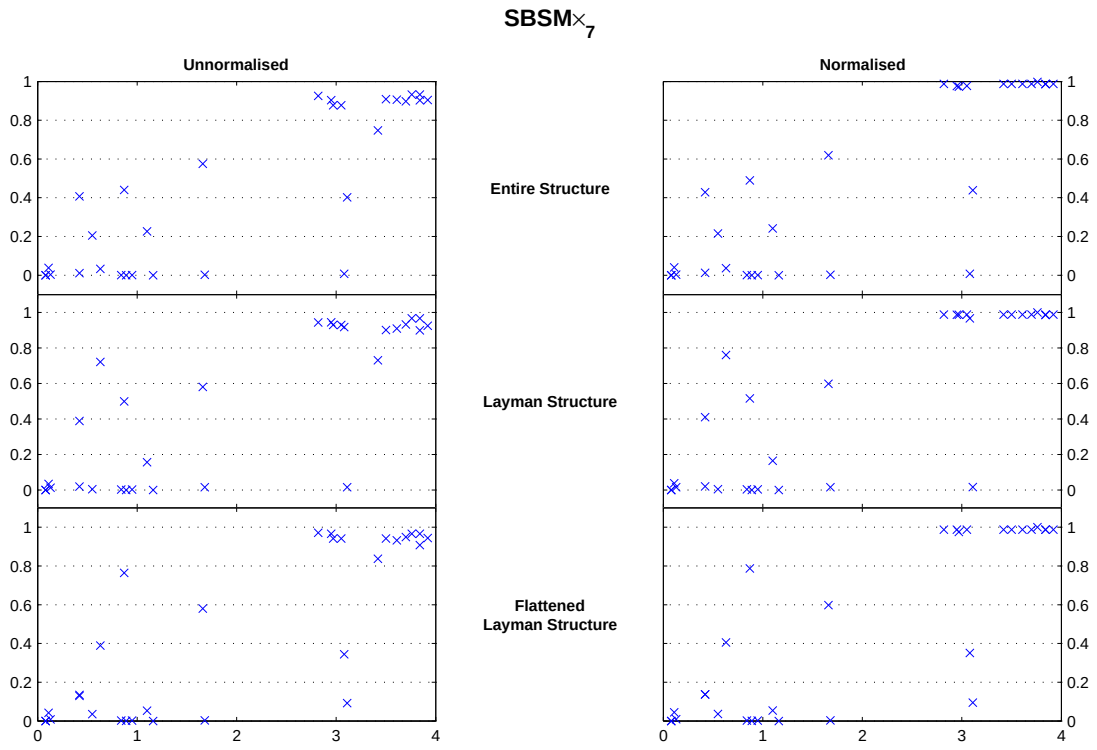

SBSM×2
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

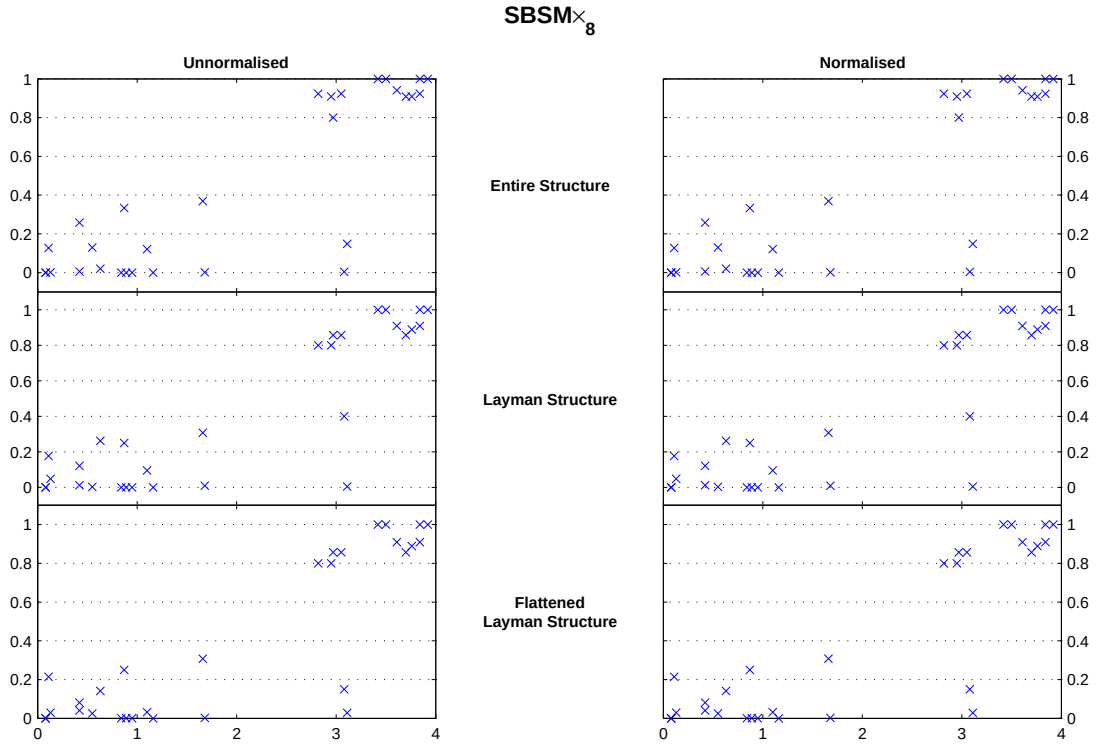

SBSM×3
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure
x 10^8

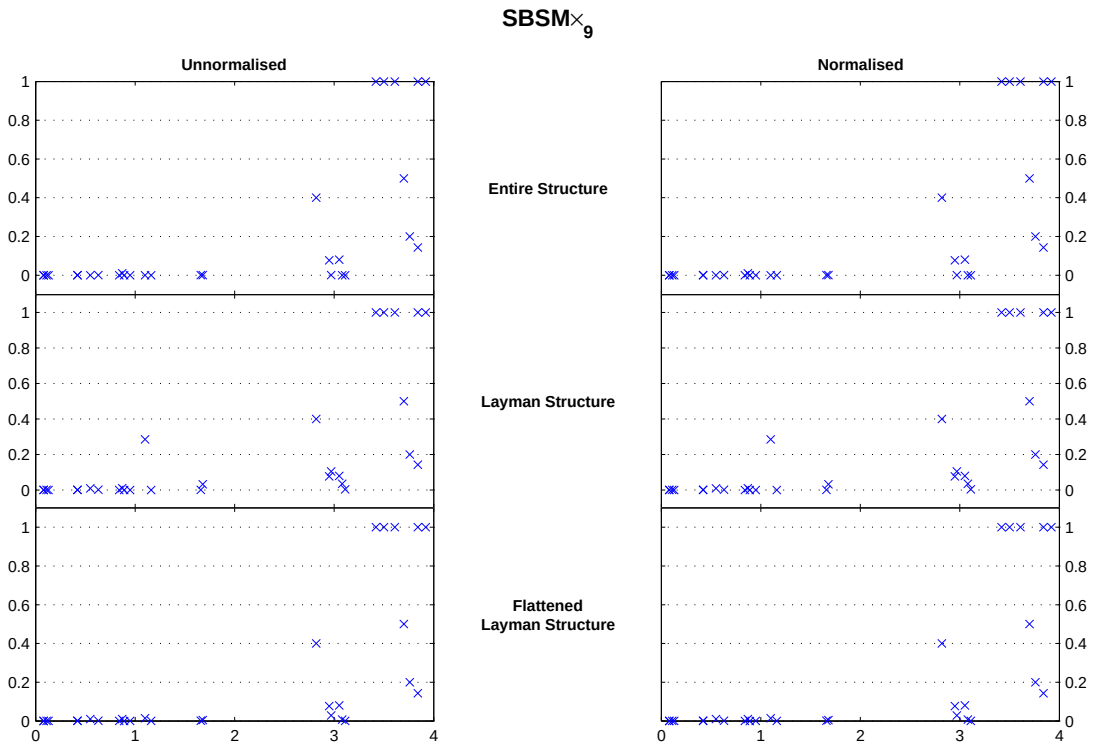

SBSM×4
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure


SBSM×5
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

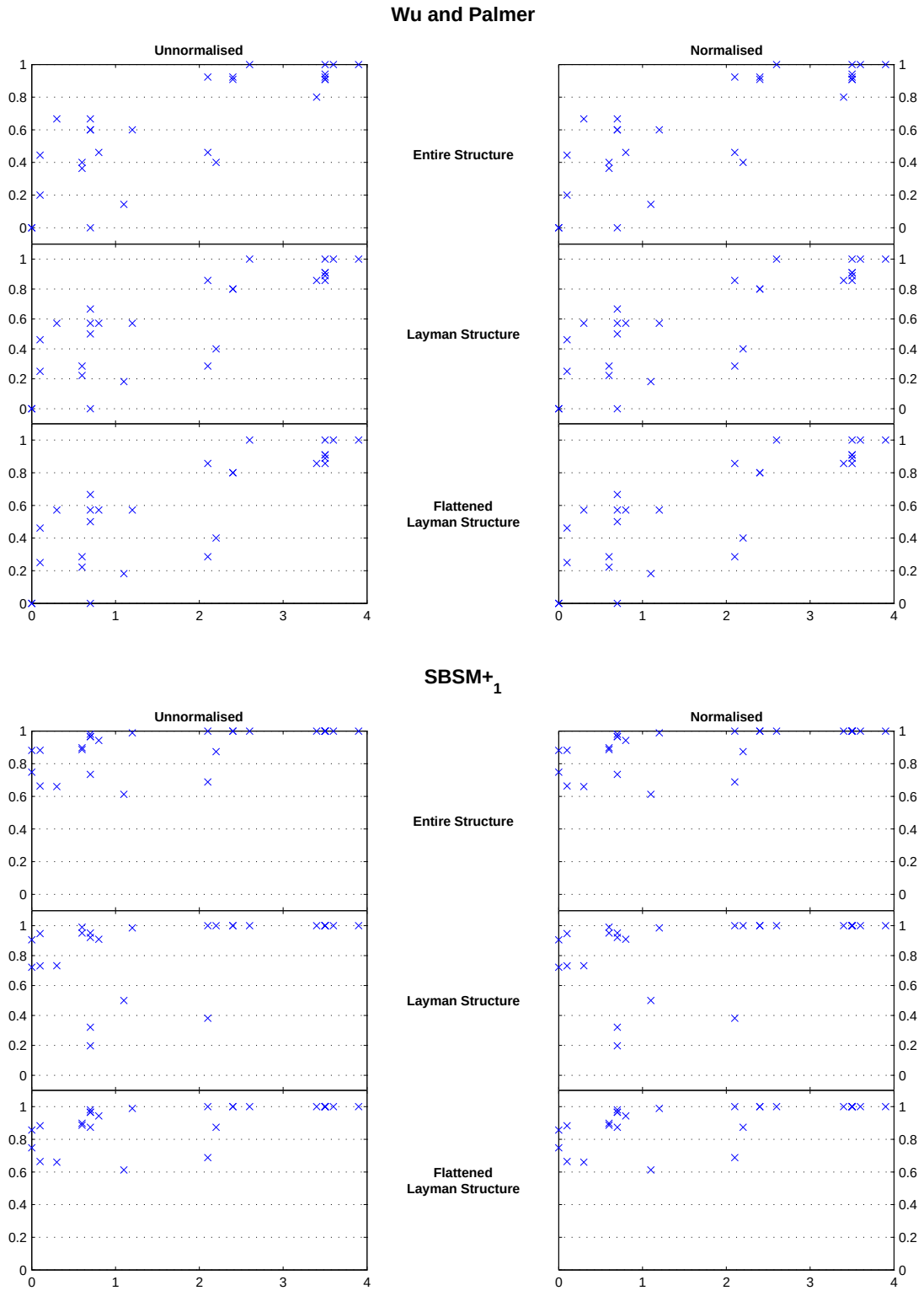

SBSM×6
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

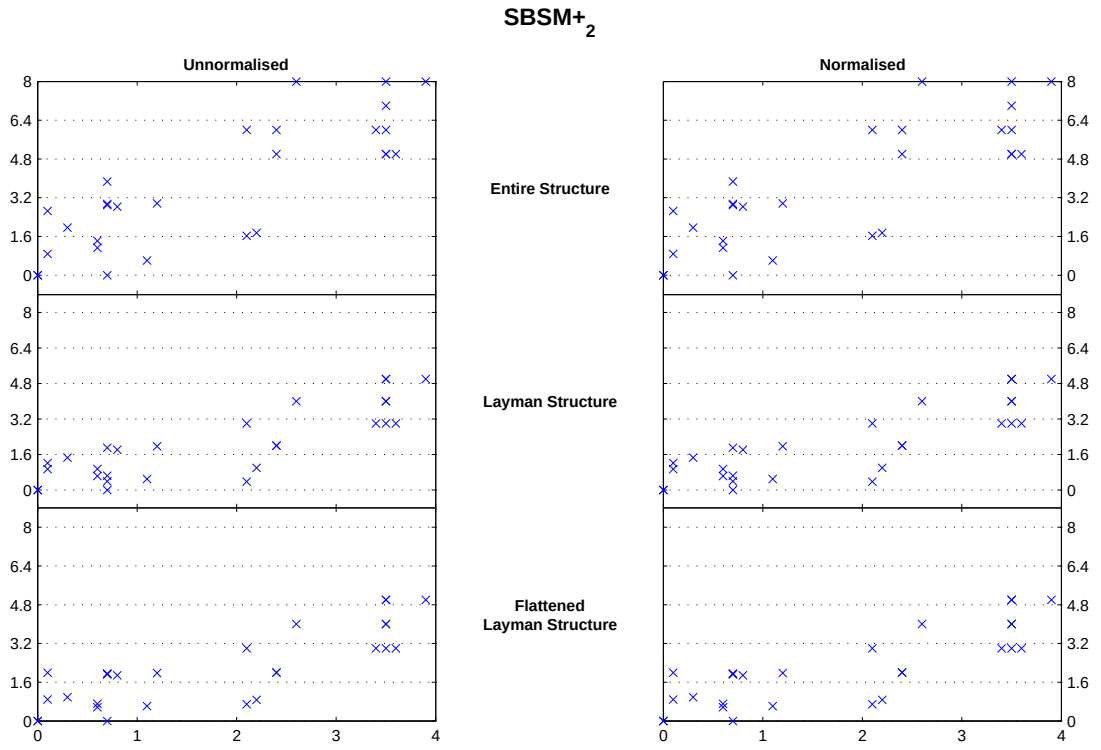

SBSM×7
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

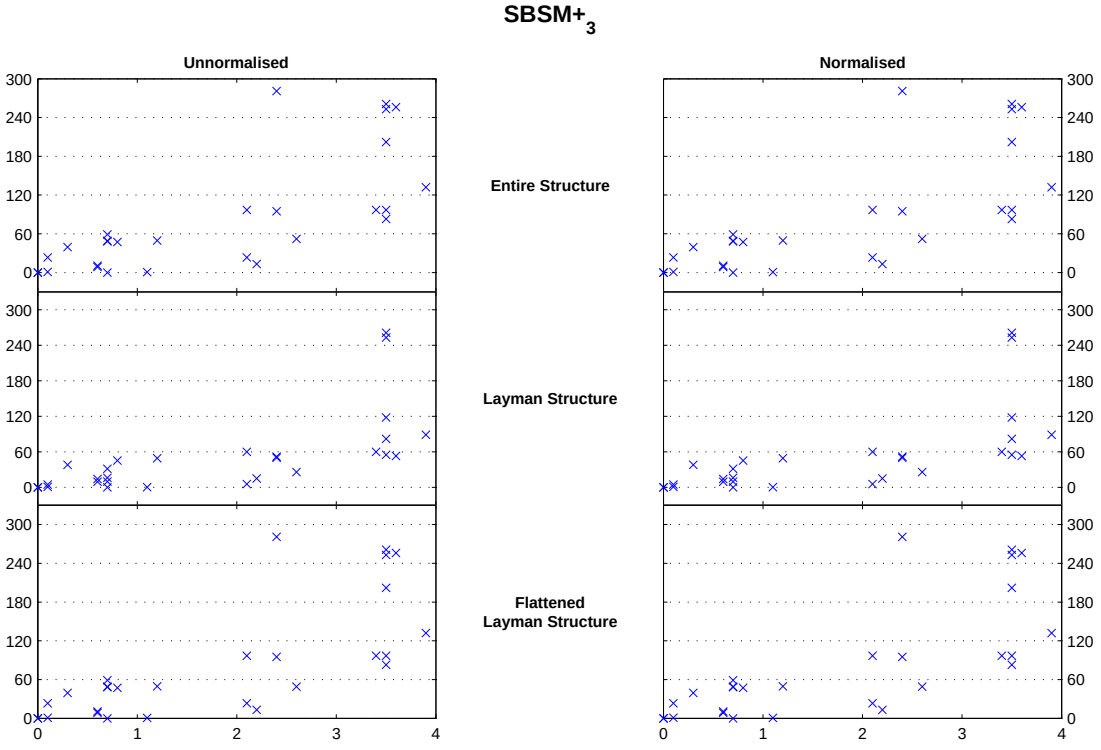

SBSM×8
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

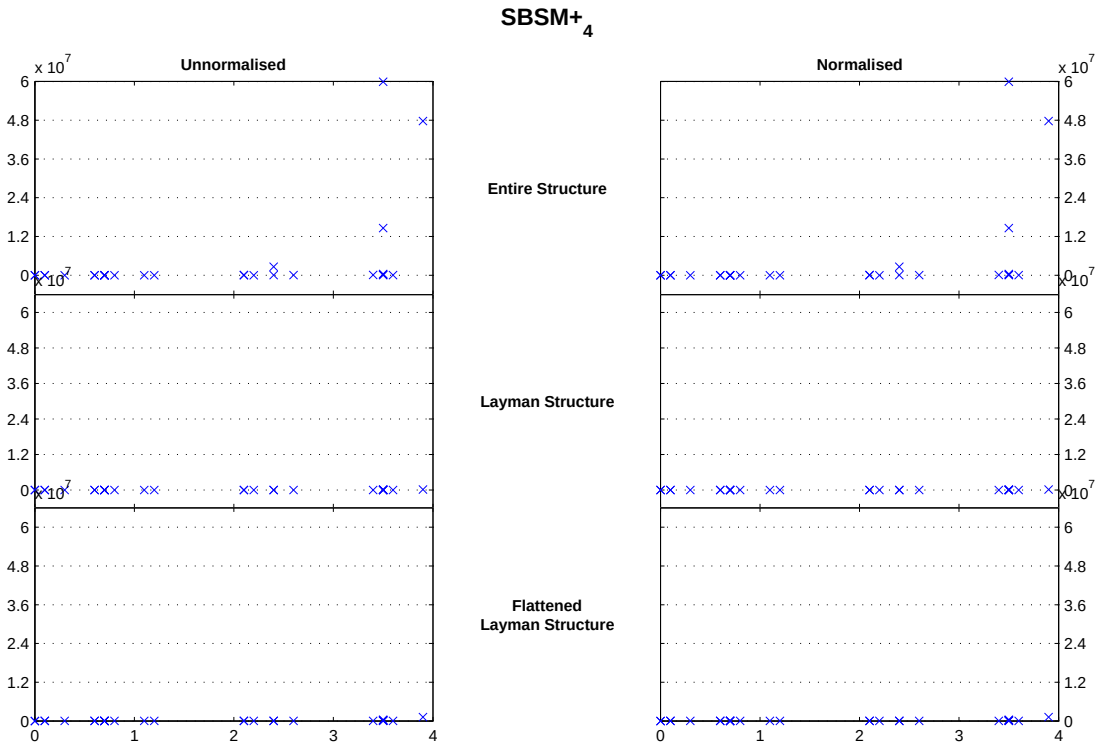

SBSM×9
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

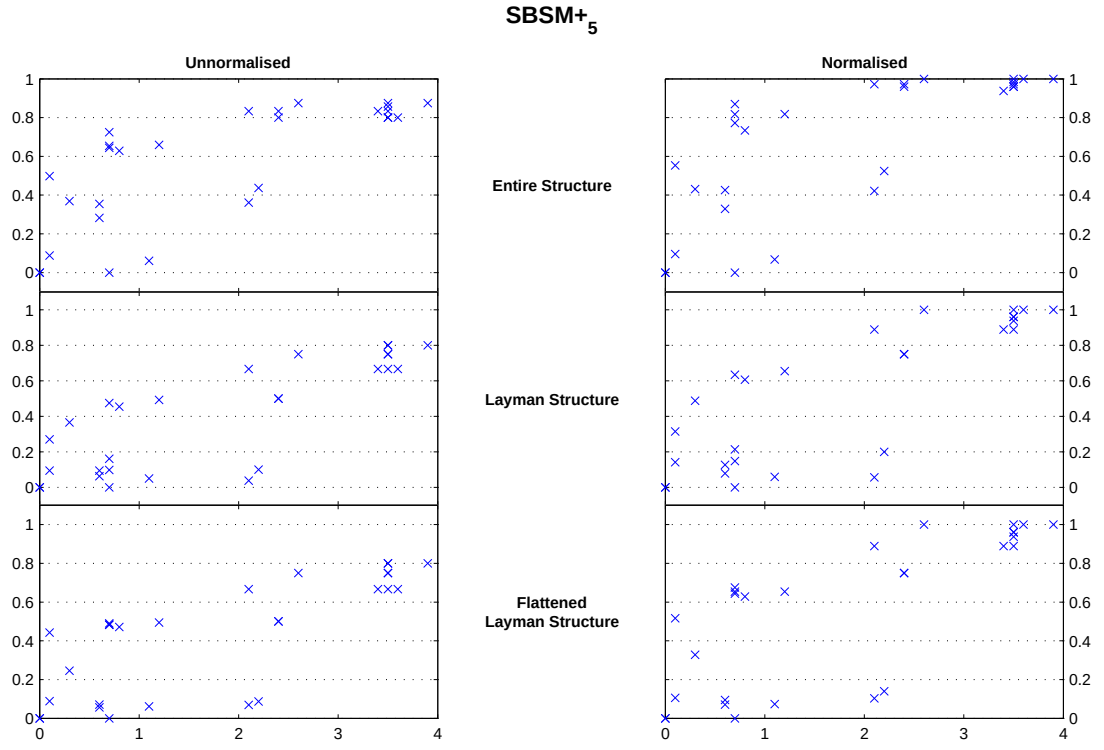
B.4 Resnik Human Judgement Correlations

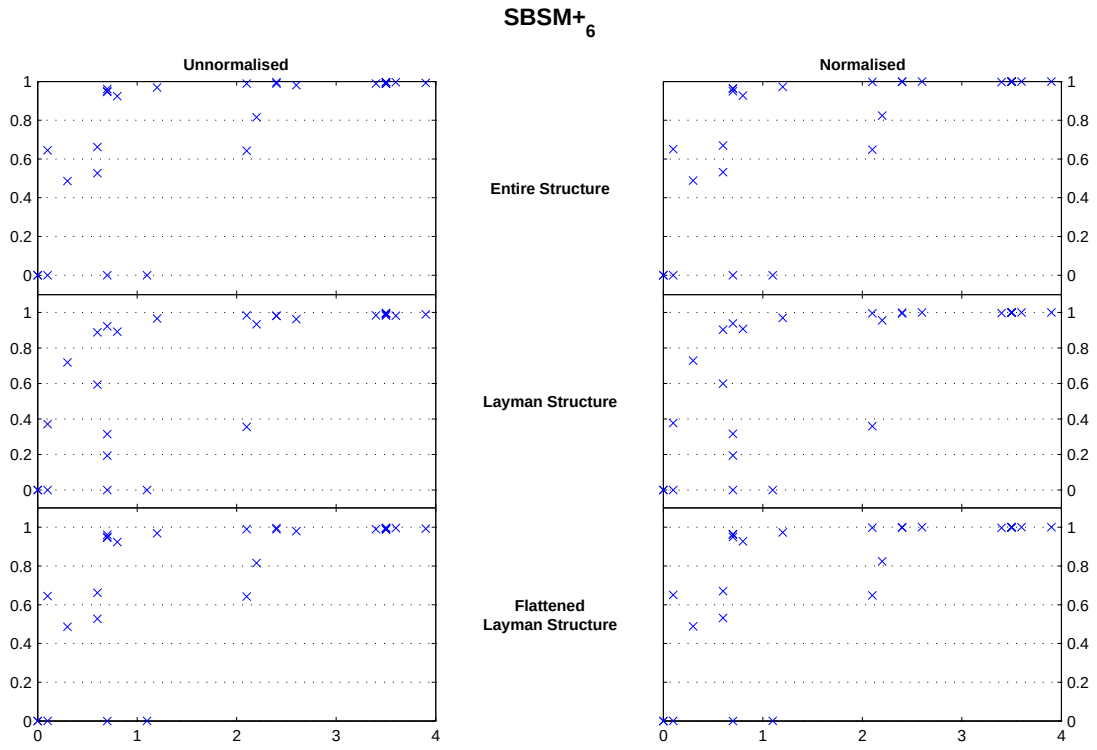


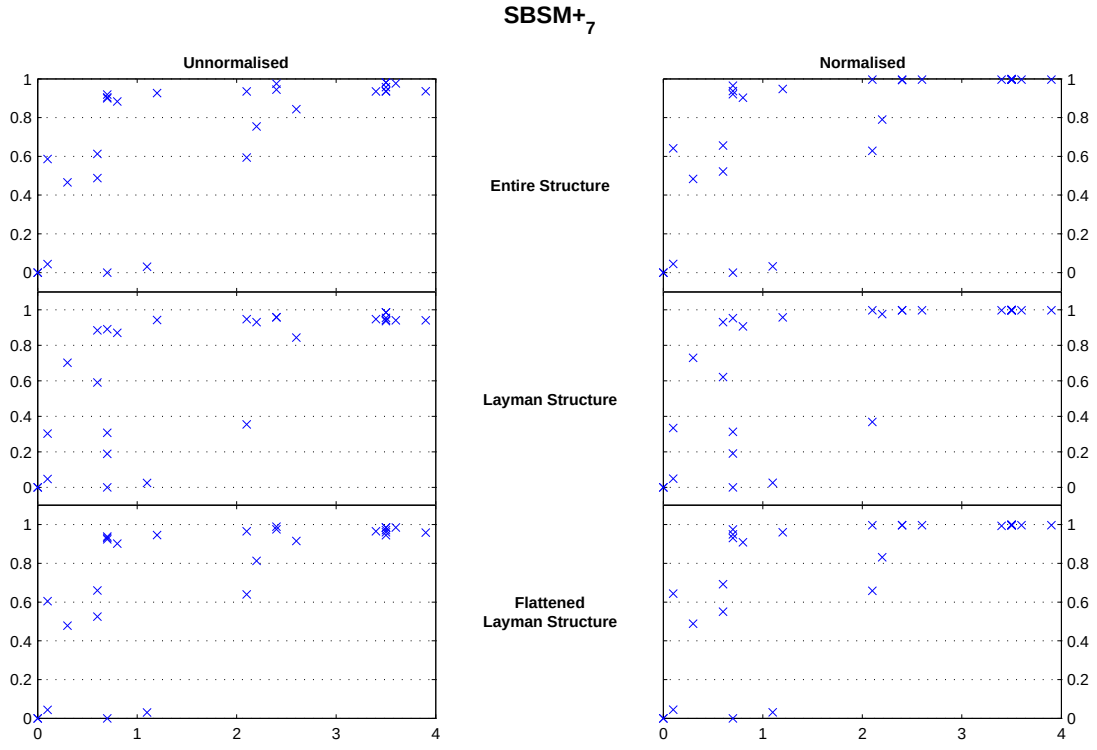

SBSM+2
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

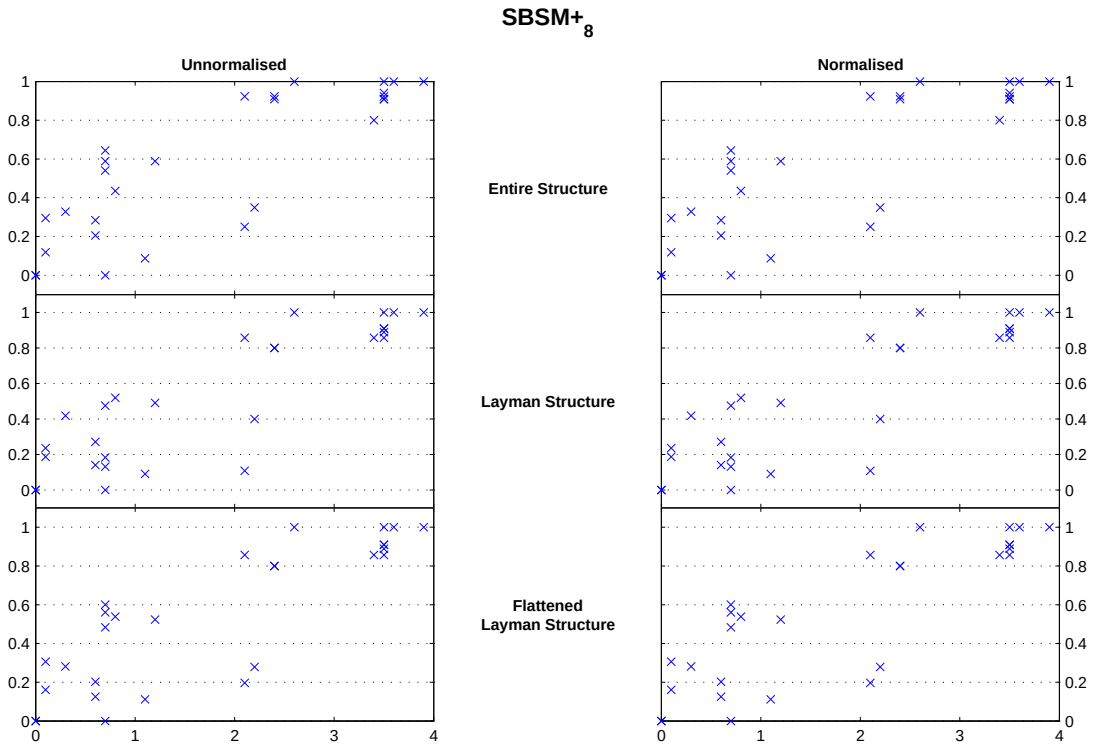

SBSM+3
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

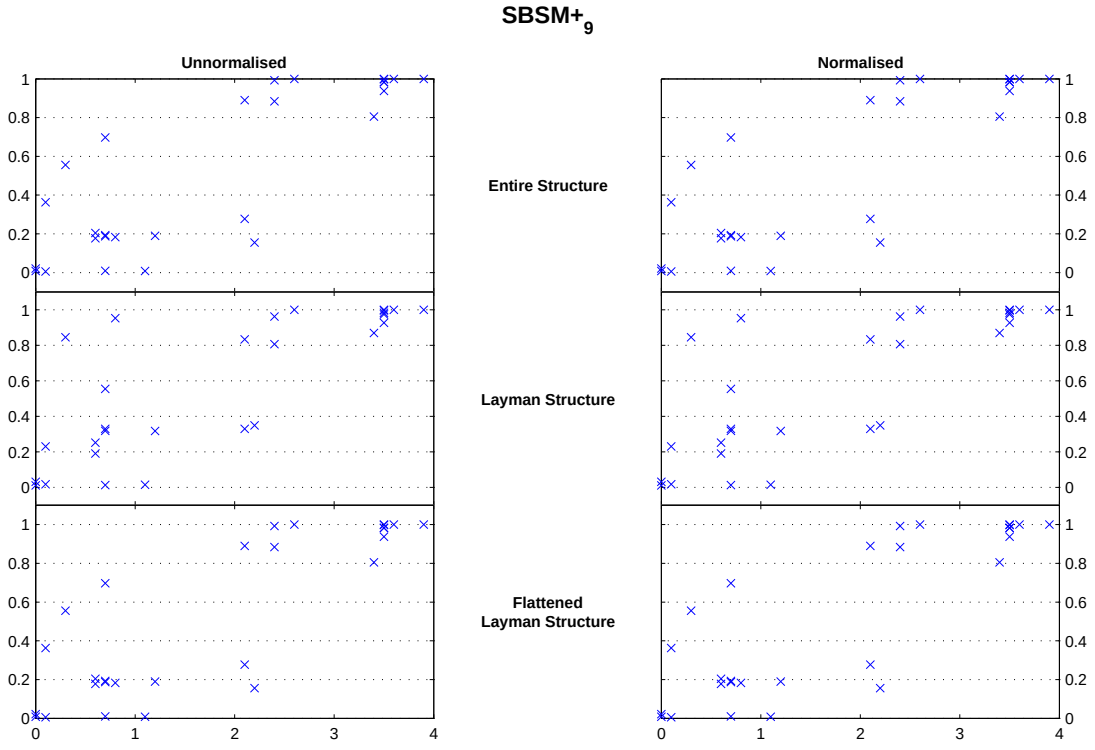

SBSM+4
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure
x 10^7

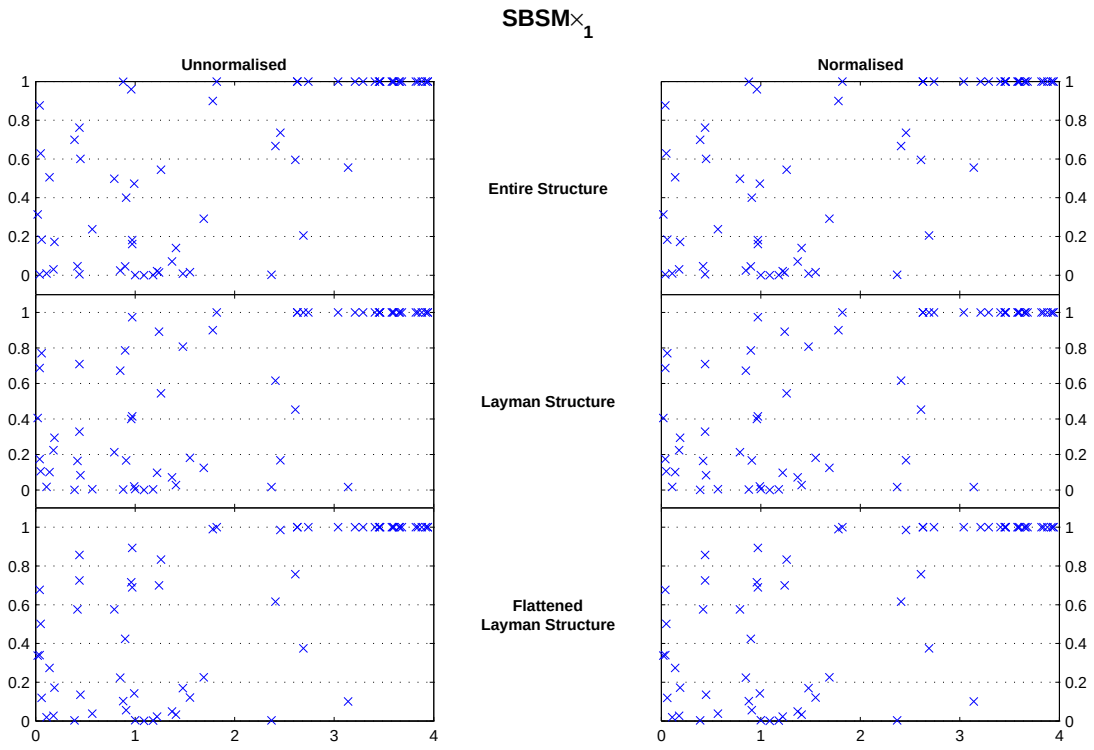

SBSM+5
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

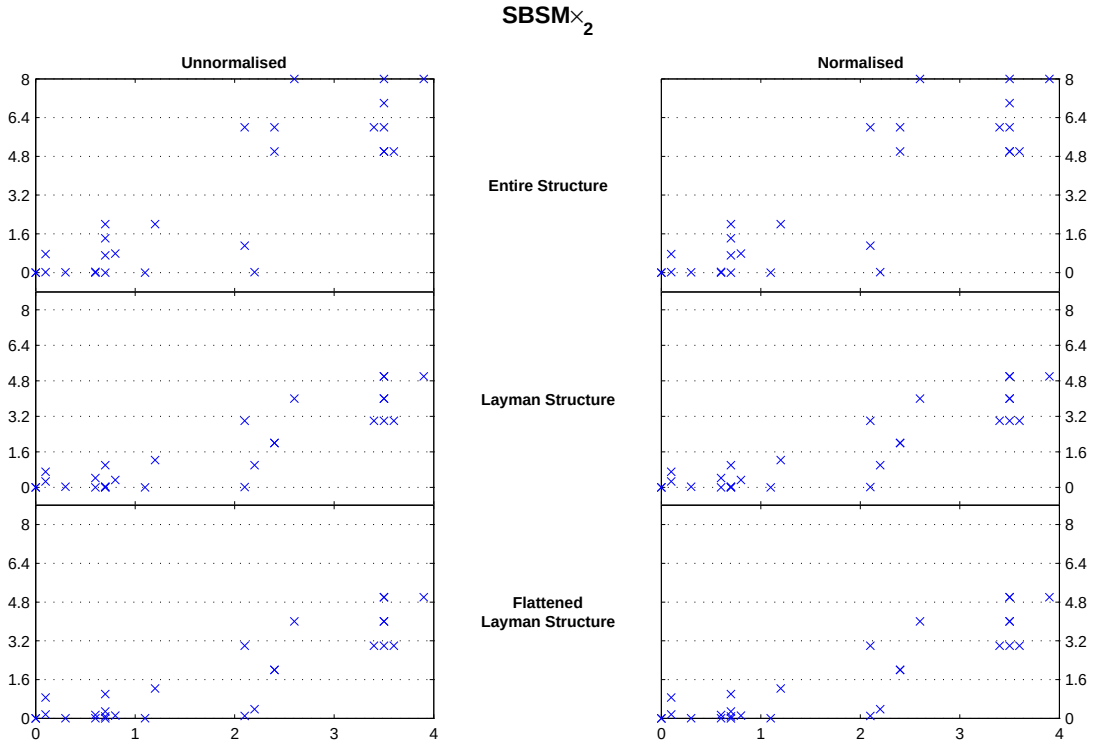

SBSM+6
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

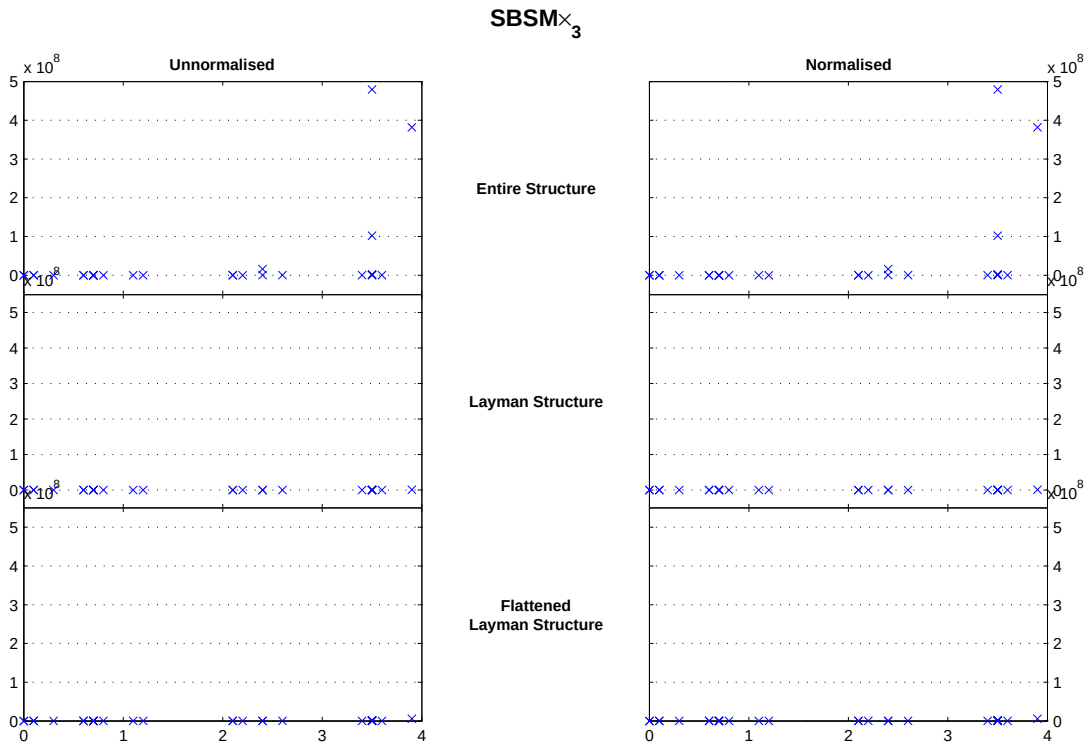

SBSM+7
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

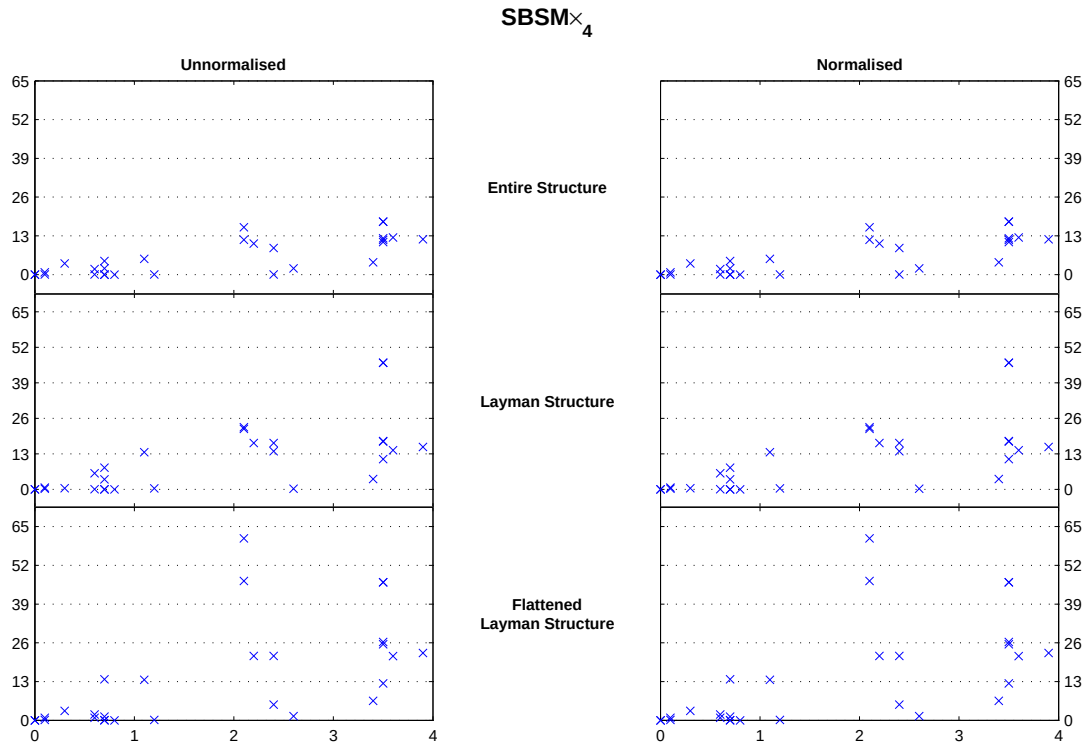

SBSM+8
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

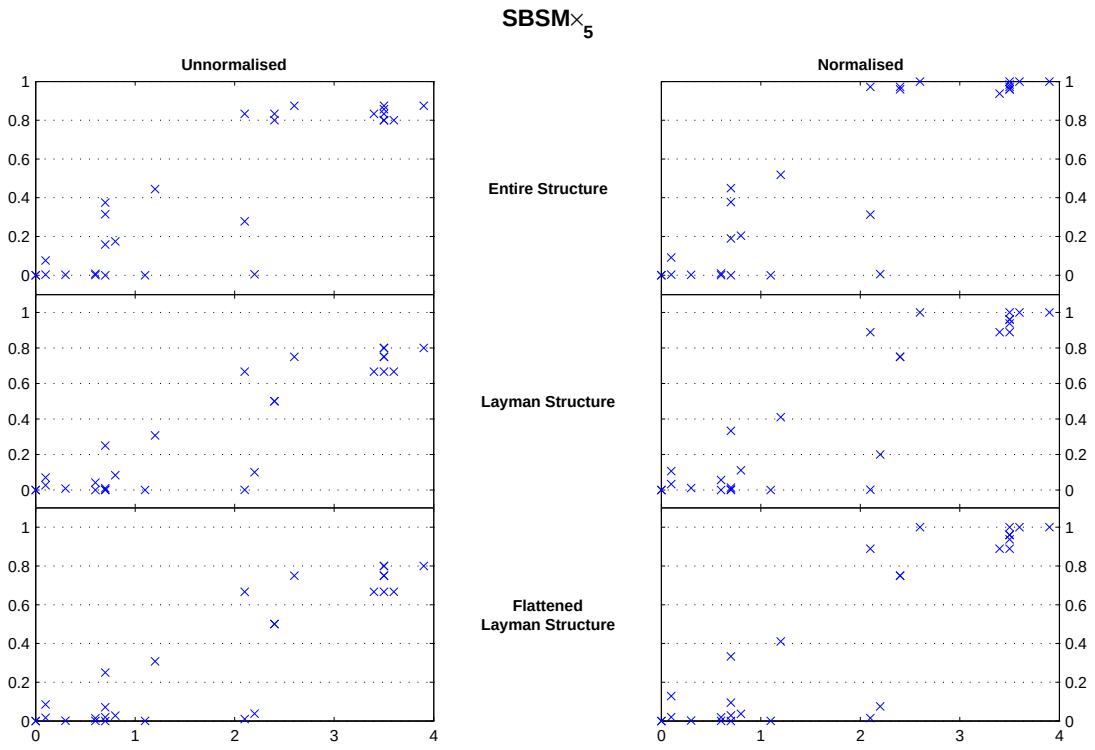

SBSM+9
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

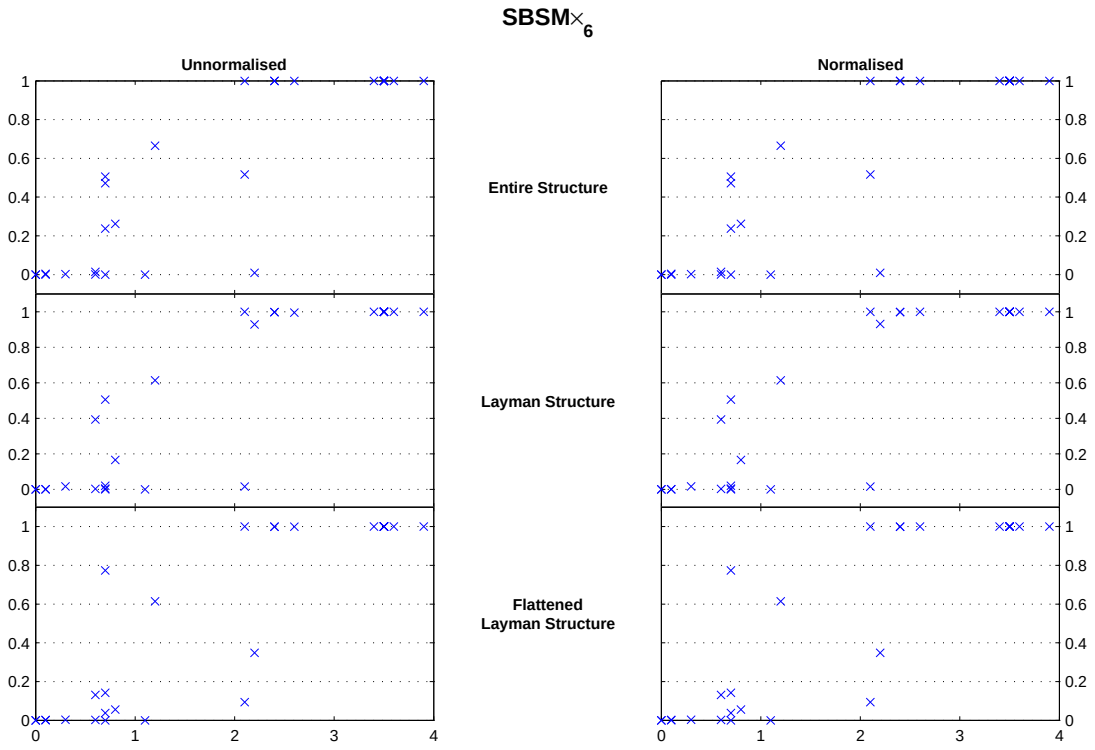

SBSM×1
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

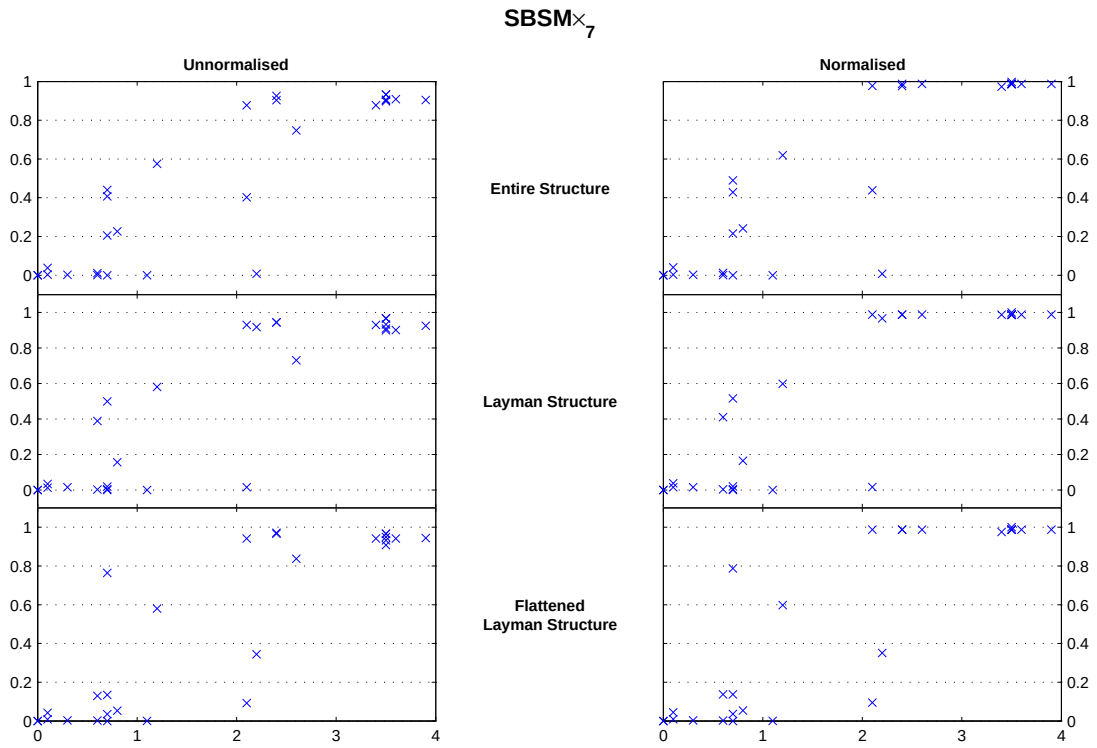

SBSM×2
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

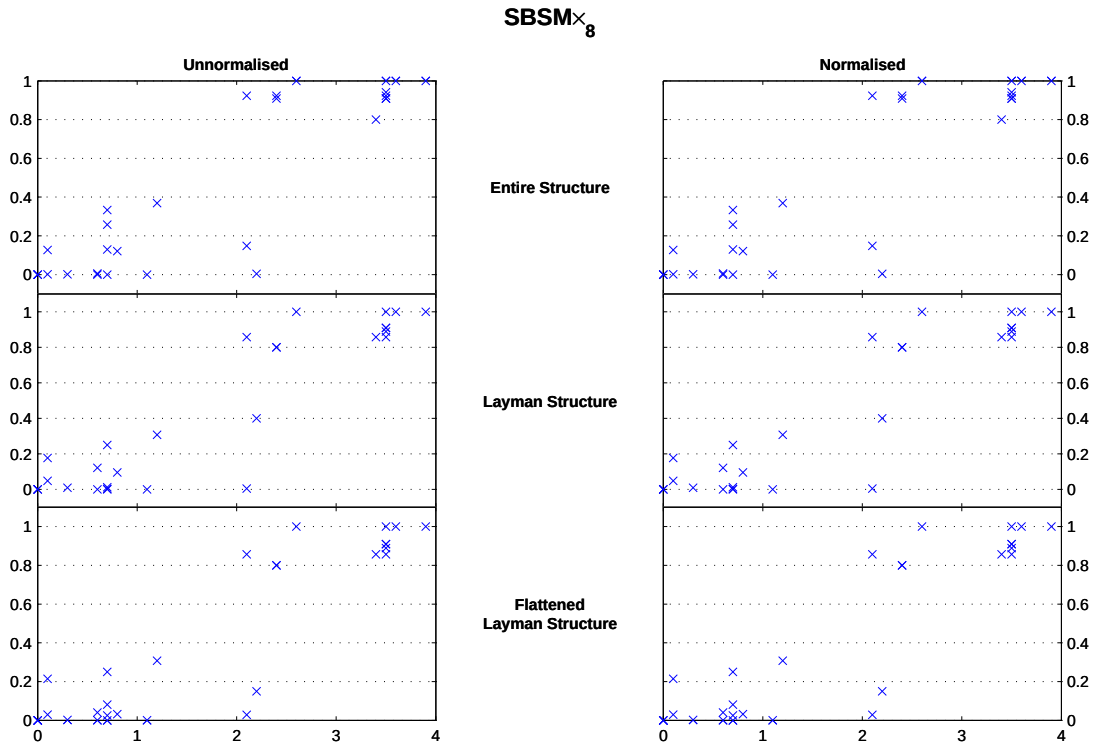

SBSM×3
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure
x 10^8

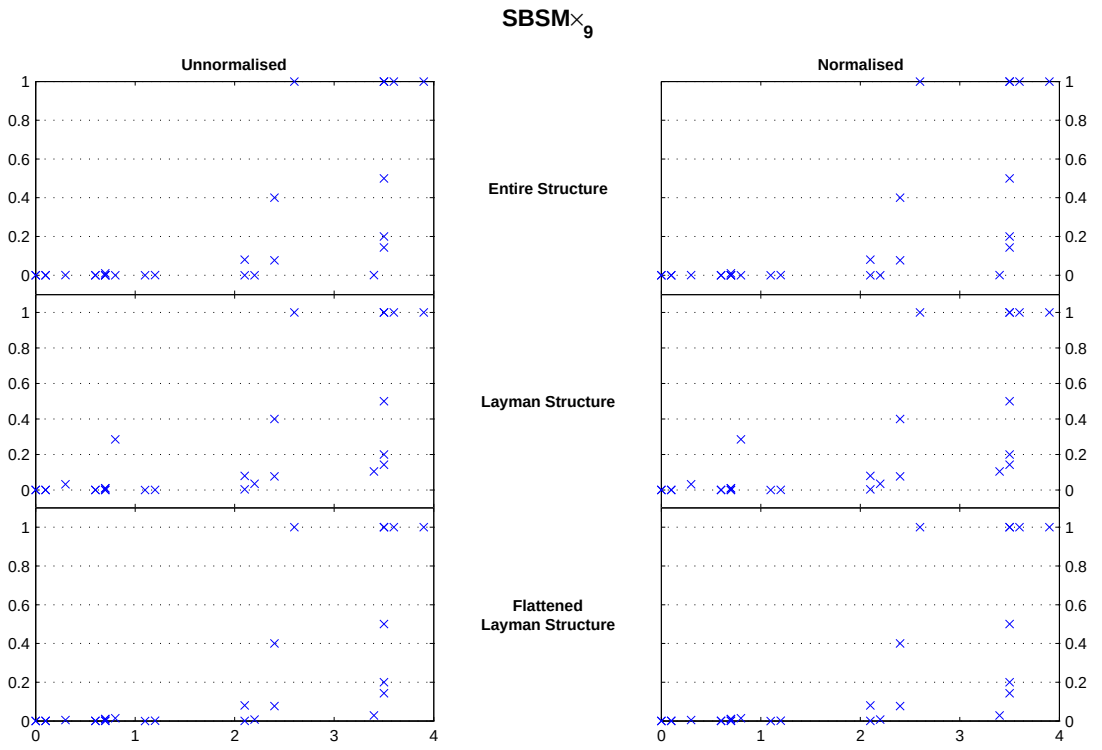

SBSM×4
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure


SBSM×5
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure


SBSM×6
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure


SBSM×7
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure


SBSM×8
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure


SBSM×9
Unnormalised
Normalised
Entire Structure
Layman Structure
Flattened
Layman Structure

Appendix C

Word Sense Disambiguation Algorithms for Noun Groups

C.1 Greedy WSD algorithm

A greedy WSD algorithm is produced by calculating the sum of the similarity for each word sense in the noun-group against all other word senses in the noun group. Similarity is calculated using a function “similarity” taking four arguments, two words and a word sense for each word. Therefore similarity is measures between two word senses, <word1>#<sense1> and <word2>#<sense2>. The sum is then normalised using the sum of the similarity of each word senses of a word, against all other word senses in the noun group. The algorithm selects the sense for each word with the highest resulting value as the correct sense for the word according to the noun group.

Listing C.1: Greedy WSD

```
Given the set of nouns W = {w1, ..., wn}

for word_index1 = 1 to n - 1
{
 word1 = W[word_index1]
 for word_index2 = word_index1 + 1 to n
 {
  word2 = W[word_index2]
  for sense1 = 1 to no_of_senses(word1)
  {
```

```
    for sense2 = 1 to no_of_senses(word2)
    {
     sim = similarity(word1, sense1, word2, sense2)

     normalization(word_index1) += sim
     normalization(word_index2) += sim
     support(word_index1, sense1) += sim
     support(word_index2, sense2) += sim
    }
   }
  }
}

for word_index = 1 to n
{
 word = W[word_index]
 for sense = 1 to no_of_senses(word)
 {
  if (normalization(word_index) != 0)
   support(word_index, sense) \= normalization(word_index)
 }
}
```

C.2 Exclusive Greedy WSD algorithm

This algorithm is similar to the Greedy algorithm, however for all word senses only similarity values greater than a specified percentage of the highest similarity value per word sense are considered. Such a percentage is specified as a threshold ranging from 0 to 1. The changes are made to avoid increasing support for the sense of a word when the similarity detected between pairs is low in comparison to the highest similarity detected for another of the word's senses.

Listing C.2: Exclusive Greedy WSD

```
Given the set of nouns W = {w1, ..., wn}

for word_index = 1 to n
{
 word = W[word_index]
```

```
 for sense = 1 to no_of_senses(word)
 {
  max_similarity( word_index , sense ) =
   find_highest_similarity( word , sense , words )
 }
}

for word_index1 = 1 to n - 1
{
 word1 = W[word_index1]
 for word_index2 = word_index1 + 1 to n
 {
  word2 = W[word_index2]
  for sense1 = 1 to no_of_senses(word1)
  {
   for sense2 = 1 to no_of_senses(word2)
   {
    sim = similarity(word1 , sense1 , word2 , sense2)

    if (sim >= max_similarity(word1 , sense1)*threshold)
    {
     normalization(word_index1) += sim
     support(word_index1 , sense1) += sim
    }

    if (sim >= max_similarity(word2 , sense2)*threshold)
    {
     normalization(word_index2) += sim
     support(word_index2 , sense2) += sim
    }
   }
  }
 }
}

for word_index = 1 to n
{
 word = W[word_index]
 for sense = 1 to no_of_senses(word)
 {
  if (normalization(word) != 0)
   support(word_index , sense) \= normalization(word_index)
```

```
  }
}
```

C.3 WSD Using Only Related Senses

Again, this algorithm is similar to the Greedy algorithm, however for all word senses only similarity values greater than a specified threshold are considered. Such a threshold is selected such that any word-pair with a similarity above the threshold will be classed as related, or similar, and anything below the threshold is considered sufficiently different to not be semantically related. Therefore the algorithm only increases support when two word sense pairs are significantly similar.

Listing C.3: Related Senses WSD

```
Given the set of nouns W = {w1, ..., wn}

for word_index1 = 1 to n - 1
{
 word1 = W[word_index1]
 for word_index2 = word_index1 + 1 to n
 {
  word2 = W[word_index2]
  for sense1 = 1 to no_of_senses(word1)
  {
   for sense2 = 1 to no_of_senses(word2)
   {
    sim = similarity(word1, sense1, word2, sense2)

    if (sim >= threshold)
    {
     normalization(word_index1) += sim
     support(word_index1, sense1) += sim
    }

    if (sim >= threshold)
    {
     normalization(word_index2) += sim
     support(word_index2, sense2) += sim
    }
```

```
    }
   }
  }
}

for word_index = 1 to n
{
 word = W[word_index]
 for sense = 1 to no_of_senses(word)
 {
  if (normalization(word) != 0)
   support(word_index, sense) \= normalization(word_index)
 }
}
```

Appendix D

Manually Tagged Selected Entries from Wordsmyth Thesaurus

This appendix includes all the entries selected from the Wordsmyth thesaurus for the evaluation in Section 4.5.2. All nouns within each thesaurus entry used have been manually tagged with their equivalent WordNet 1.6 sense/senses. If a noun has no sense tag, it is deemed not to have an adequate definition according to WordNet. Also, some entries within Wordsmyth omit information such as definitions.

Adult

#DEF: 1. a person who is fully grown, mature, and considered legally responsible.
adult, 1
grownup, 1
man, 1
woman, 1

#DEF: 2. a mature animal or plant.
adult, 2
adulthood

Air

#DEF: 2. an open place in the frozen surface of a lake, pond, or stream.
air_hole, 1
outlet, 3
blowhole, 2
flue, 3
duct, 1
exhaust
spiracle
vent, 1
opening, 9
orifice, 1
window, 2, 6
passage, 7
chimney, 1
smokestack, 1
spout, 1

#DEF: 3. see air pocket.
air_hole, 1
air_pocket, 1

#DEF: a route regularly used by aircraft; airway.
air_lane, 1
airway, 2
corridor
route, 1

#DEF: 1. the tasteless, odorless, and colorless mixture of nitrogen, oxygen, and ther gases that forms the earth’s atmosphere.
air, 1
atmosphere, 3, 5
ozone, 1
stratosphere, 1
oxygen, 1
gas, 2

#DEF: 2. all that is above the ground; sky.
air, 3
sky, 1
heaven
atmosphere, 3, 5
stratosphere, 1
welkin, 1
ether
airspace, 1

#DEF: 3. movement of the atmosphere; breeze or wind.
air, 6
wind, 1
airflow, 1
breeze, 1

draft, 2
current
zephyr, 1
waft
breath, 5

#DEF: 4. the peculiar character, manner, bearing, or aspect of a person or thing:
#EXA: He has a strange air.
air, 5, 7
character, 2, 3
atmosphere, 6
aura, 3
ambiance
manner, 2
bearing, 3
aspect, 1, 2
style, 2, 5
climate, 2
feel, 2, 3
impression, 2
appearance, 1
demeanor, 1
mien, 1
tone, 3, 4, 5, 10
spirit, 2, 3

#DEF: 5. (pl.) pretense or affectation:
#EXA: She is putting on airs.
airs, 1
affectedness, 1
affectation, 1
pretense, 4
pretension, 1
arrogance, 1
swank

#DEF: 6. travel or transportation by aircraft.
#EXA: He sent them by air.
air, 2
airplane, 1
plane, 1
aircraft, 1
jet, 1
jetliner

Airplane

#DEF: any of various aircraft that are heavier than air and are driven by propellers or jet engines.
airplane, 1
aircraft, 1
plane, 1
jet, 1
propjet, 1
turboprop, 1, 2
turbojet, 1
airship, 1
helicopter, 1
airliner, 1

Airport

#DEF: a large area of level land where airplanes can land and take off, usu. including a passenger terminal and cargo and repair facilities.
airport, 1
airfield, 1

```
airdrome, 1
flying_field, 1
airstrip, 1
landing_field, 1
air_base, 1
terminal, 1
```

Album

```
#DEF:       1. a book or
binder with blank pages
or empty pockets in
which a collection can
be inserted, as of
photographs, stamps, or
mementos.
album, 2
book, 2
scrapbook, 1
notebook, 1
portfolio, 1
folder
file
compilation, 1
catalogue, 2
binder, 3

#DEF:       2. a
phonograph record or set
of records, or the
jacket or binder thereof.
album, 1
record, 2
LP, 1
disk, 3
recording, 3
soundtrack, 1
CD, 4
compact_disk, 1
tape, 5

#DEF:       3. a printed
collection of pictures,
or musical or literary
selections.
album, 2
collection, 1, 2
anthology, 1
record, 5
documentation
chronicle, 1
```

Alphabet

```
#DEF:       2. the
fundamental principles
of a subject; rudiments.
alphabet, 1
script, 3
writing, 4
letters
```

Arm

```
#DEF:       1. either of
the two upper limbs of
the human body, between
the shoulders and the
wrists.
arm, 1
forelimb, 1
brachium
limb, 1
forearm, 1

#DEF:       2. any part
that extends from a main
body and resembles an
arm.
arm, 1, 2
appendage, 1
```

brachium, 1
limb, 1
branch, 6
ramification
bough, 1
offshoot
projection, 4
crosspiece

#DEF: 3. the part of an organization that specializes in operations or enforcement; authority.
arm, 5
authority, 5
power, 5
command, 2
division, 4
department, 1
force, 5, 7
detachment, 4

#DEF: 1. (usu. pl.) weapons, esp. those that shoot or explode.
arm, 3
firearm, 1
gun, 1, 2
rifle, 1
pistol, 1
revolver, 1
six-shooter, 1
shotgun, 1
machine_gun, 1
weapon, 1
armament, 1
munition, 1
ammunition, 1
cannon, 1, 3
artillery, 1
ordnance, 2

#DEF: 2. a part of a military force.
arm, 5
command, 2
ordnance
power
branch, 1
outfit, 1

#DEF: 3. (pl.) the insignia of a family or institution:
#EXA: a coat of arms.
arms, 2
coat_of_arms, 1
blazon, 1
heraldry, 2
crest, 4
insignia, 1
escutcheon, 3

Army

#DEF: 1. the military land force of a nation.
army, 1
soldier, 1
soldiery, 1
troops, 1
military, 1
armed_forces, 1
artillery, 2
cavalry, 1, 2
infantry, 1
militia, 1

#DEF: 2. a great number of people or things:

#EXA: The singer had an army of fans.
army, 2
host, 2
multitude, 2
legion, 4
crowd, 1
flock, 4
horde, 1, 3
throng, 1
bevy, 1
mass, 2
aggregation, 1

#DEF: 3. a large, organized group.
army, 2
legion, 4
battalion, 2
brigade, 1
throng, 1
crowd, 1
flock, 4
horde, 1, 3
force, 4, 5, 8
assemblage, 1
group, 1
troop, 1, 2, 3, 4

Baby

#DEF: 1. an extremely young girl or boy; infant.
baby, 1
infant, 1
babe, 1
bambino, 1
newborn, 1
neonate, 1
papoose, 1
child, 1, 2, 5, 6
suckling, 1
nursling, 1
weanling
toddler, 1
tot, 1
kid, 1, 3
youngster, 1

#DEF: 2. an young or newborn animal:
#EXA: This gorilla was tame when it was a baby.
baby, 3
newborn, 1
suckling, 2
weanling
young, 1
progeny

#DEF: 3. the youngest person in a family or group.
baby, 4
junior, 3
youngster

#DEF: 4. a person who behaves childishly or immaturely.
baby, 5
child, 3
naif

#DEF: 5. (informal) a young woman (usu. used in direct address).
baby, 2
babe
sweetheart, 3
honey, 2

girlfriend, 2
gal, 3
girl, 1, 4, 5
lass, 1

#DEF: 6. (informal) something of personal concern or pride:
#EXA: That project is his baby.
baby, 6
pride, 5

Backpack

#DEF: a pack used to carry objects, esp. camping gear, on one’s back; knapsack.
backpack, 1
knapsack, 1
rucksack, 1
packsack
pack, 9
sack, 1
bag, 1, 6
pouch, 1
tote, 1
kit, 1
luggage, 1
baggage, 1
package, 2
valise, 1

Balloon

#DEF: 1. a bag made of thin material that is filled with a gas that is lighter than air and causes it to rise.
balloon, 2
zeppelin, 1
dirigible, 1
blimp, 2
airship, 1
aerostat

#DEF: 2. such a bag used to transport passengers or scientific equipment.
balloon, 2
aerostat
zeppelin, 1
dirigible, 1
blimp, 2
airship, 1

Bank

#DEF: at a bank, the funds credited to a depositor and subject to withdrawal by him or her.
bank_account, 1
accumulation, 4
mass
sediment
funds, 1
deposit, 3
lees
dregs
settlings
precipitate
silt
alluvium

#DEF: a promissory note issued by an authorized bank.

```
bank_note, 1
bill, 3
treasury_note, 1
note, 6
paper_money, 1
legal_tender, 1
money, 1, 3
greenback, 1
currency, 1
certificate, 2
silver_certificate, 1
gold_certificate
promissory_note, 1
IOU, 1
green
tender, 1

#DEF:       1. a heap or
mass of something, such
as earth or clouds.
bank, 3, 7
heap, 1, 2
mass, 2, 3
pile, 1, 2
stack, 1, 2
drift, 4
accumulation, 2
bundle, 1
cock
shock, 7
rick, 2
mow
bale, 1

#DEF:       2. a slope,
usu. of earth.
bank, 2, 9
embankment, 1
mound, 4
slope, 1
acclivity, 1
incline, 1
dike
levee
parapet
drift
ridge, 1
rise, 3
hill, 2
knoll
dune, 1
hillock

#DEF:       3. the ground
at the edge of a river
or stream.
bank, 2
shore, 1
edge
beach, 1
foreshore, 1
littoral, 1

#DEF:       1. a business
concerned with the
safeguarding, exchanging,
and lending of money.
bank, 1
credit_union, 1
savings_bank, 1
Federal_Reserve_Bank, 1
thrift_institution, 1
depository, 1
trust_company, 1
S_and_L

#DEF:       2. the reserve
of money held by a
gambling establishment.
bank, 8
kitty, 1, 2
pot, 6

#DEF:       3. a supply
```

or reserve:
#EXA: a blood bank.
bank, 3, 4
storehouse, 1
warehouse, 1
repository, 1
store, 2, 4, 5
reservoir, 1
reserve, 2
depository, 1
stockpile, 1
stock, 4
supply, 1
fund, 2
piggy_bank, 1

Bath

#DEF: 1. a process of washing or soaking something in order to cleanse, refresh, or heal.
bath, 2
washing, 1
wash, 2
soak, 2
cleaning, 1
cleansing, 1
ablution, 1
immersion, 3
soaking, 3
rinse, 4
shower, 2
scrub, 2
scrubbing, 1
sponge
sauna

#DEF: 2. water or other liquid used for washing.
bath, 2
ablution, 1
water, 1
suds, 1
solution, 1
soak, 2

#DEF: 3. (often pl.) an establishment where people go to take a bath or to obtain therapy.
bath, 5
bathhouse, 2
sauna, 1
sudatorium
spa
sanitarium
Turkish_bath, 1, 2
health_club
natatorium, 1
sanatorium, 1

#DEF: 4. a bathroom.
#PHR: take a bath.
bath, 5
bathroom, 1, 2
washroom, 1
toilet, 1
lavatory, 1
water_closet, 1
W.C., 1
restroom, 1
can, 6
privy, 1
outhouse, 1
latrine, 1
powder_room, 1
lounge, 2
comfort_station, 1
commode, 1

Bathroom

#DEF: a room with a toilet and often containing a sink, bathtub, or other facility for washing.
bathroom, 1, 2
bath, 5
washroom, 1
toilet, 1
lavatory, 1
water_closet, 1
W.C., 1
restroom, 1
ladies'_room, 1
men's_room, 1
can, 6
privy, 1, 2
outhouse, 1
latrine, 1
powder_room, 1
lounge, 2
comfort_station, 1
commode, 1

Bed

#DEF: 1. a piece of furniture used for resting or sleeping.
bed, 1
bunk, 2, 3, 5
cot, 2, 3
four-poster, 1
truckle, 1
trundle_bed, 1
sack, 6
berth, 3

#DEF: 2. any place or thing used for resting or sleeping.
bed, 1
pallet, 2
sleeping_bag, 1
berth, 3
roost
chamber, 5
bedroom, 1

#DEF: 3. an area of ground used for planting, or the plants themselves:
#EXA: a bed of flowers.
bed, 2
garden, 1
plot, 2
patch, 2
plat
flat

#DEF: 4. the bottom of a body of water:
#EXA: a lake bed.
bed, 3
bottom, 5
base
floor, 5

#DEF: 5. a supporting base or layer:
#EXA: a bed of gravel under the bricks.
bed, 6
foundation, 3
substratum, 1
layer, 2
basis, 2

support, 7
substructure, 1
stratum, 1
deposit
seam, 3
lode, 1
base, 2, 8

Bible

#DEF: 1. the principal sacred writings of Judaism, comprising the Old Testament, and of Christianity, comprising both the Old and New Testaments.
Bible, 1
Holy_Scripture, 1
Scripture, 1
Holy_Writ, 1
Good_Book, 1
The_Book
Word, 7
Old_Testament, 1
New_Testament, 1
Gospel, 1

#DEF: 3. (l.c.) any book or text that is considered authoritative or official.
bible, 2
scripture, 1, 2
authority, 7
handbook, 1
guide, 3
manual, 1
vade_mecum, 1
guidebook, 1
reference, 4
primer, 1
textbook, 1
text, 3

Bomb

#DEF: 4. (informal) a failure:
#EXA: His concert was a bomb.
bomb, 3
flop, 3
failure, 2
failing, 2
dud, 1
lemon
bust, 1
defeat, 1
fiasco, 1
fizzle
washout, 1
debacle, 3
miscarriage, 1
muff, 2

Book

#DEF: 1. a collection of bound paper sheets, usu. containing written or printed words.
book, 1, 2, 8
volume, 3
edition, 1
folio
album, 2
booklet, 1
notebook, 1
handbook, 1
diary, 2

tome, 1
journal, 4

#DEF: 2. a literary work such as a novel or volume of poetry.
book, 1
edition, 1, 3
opus, 1
literature, 1
publication, 1
belles-lettres, 1
manuscript, 1

#DEF: 3. (pl.) financial or business records:
#EXA: He keeps the books.
book, 5
ledger, 1
daybook, 1
journal, 1
log, 4, 5
blotter, 2
record, 1, 7
transcript, 1

#DEF: 4. (cap.) the Bible (prec. by the).
book
Bible, 1
Word, 7
scripture, 1
Holy_Scripture, 1
Good_Book, 1
Holy_Writ, 1
Gospel, 1
Old_Testament, 1
New_Testament, 1

#DEF: 5. a set of similar things bound together into one unit, such as matches, stamps, or tickets.
book, 8
roll, 6
pad, 1
packet, 1

Boss

#DEF: 1. a person who employs others or supervises their work; manager.
boss, 1
manager, 1
executive, 1
CEO, 1
chief, 1, 2
leader, 1
foreman, 1
superintendent, 1
super
master, 2, 4
supervisor, 1
head, 4
taskmaster, 1
overseer, 1
administrator, 1
employer, 1

#DEF: 2. a politician who dominates a local party.
boss, 4
cacique
party, 5
man, 1
eminence, 1
kingmaker

```
war-horse, 2
whip, 2

#DEF:       1. a rounded
projection or swelling.
boss, 5
nub, 1
bubble, 1
knob, 1
node, 5
bulb, 5
stud, 2
knurl
nubble
blister, 1
bulge, 1
bump, 1
swell, 2
billow

#DEF:       2. an
ornamental projection,
such as a knob or stud.
boss, 5
stud, 2
nailhead, 1, 2
knob, 4
burl, 3
```

Bottle

```
#DEF:       1. a container,
usu. made of glass and
having a slender neck,
used mainly for storing
or serving liquids.
bottle, 1
carafe, 1
magnum, 1
vacuum_bottle, 1
demijohn, 1
decanter, 1
flagon, 1
flask, 1
cruet, 1
flacon
jug, 1
jeroboam, 1

#DEF:       2. the amount
such a container will
hold:
#EXA:           I used a
bottle of wine in this
stew.
bottle, 2
jar, 2
jug, 2
glassful, 1
quart, 1, 2
pint, 1, 3
cup, 2
gallon, 1, 2

#DEF:       3. formula or
cow’s milk fed to infants
in place of mother’s
milk, usu. contained in a
bottle fitted with a
nipple.
bottle
formula, 6
milk, 1, 4
```

Bowl

```
#DEF:       1. a deep,
rounded dish used mostly
for containing food,
liquids, or the like.
bowl, 1, 3
dish, 1
```

saucer, 2
porringer, 1
cup, 1
tureen, 1

#DEF: 2. the contents of such a dish:
#EXA: I ate a bowl of cereal.
bowl, 4
dish, 3
cup, 2

#DEF: 3. the rounded, dishlike part of something, as of a spoon, sink, or toilet.
bowl, 2
sink, 1
toilet, 2
washbasin, 2
basin, 1, 2

#DEF: 4. a rounded valley or other geographical depression or formation.
bowl, 2
valley, 1
hollow, 2
basin, 4
depression, 3
indentation, 1
dip, 1
crater, 3
hole, 5

#DEF: 5. a rounded stadium or outdoor theater.
bowl, 5
stadium, 1
amphitheater, 2
coliseum, 1
arena, 3

#DEF: 6. in the United States, a football game played at the end of the season by specially elected teams:
#EXA: the Super Bowl.
bowl
tournament, 1
playoff, 1
championship meet, 1

#DEF: 1. a large wooden ball shaped or weighted so as to roll in a curved path, used in lawn bowling.
bowl, 6
ball, 1

#DEF: 2. (pl., but used with a sing. verb) the game or sport of lawn bowling.
bowls, 1
lawn_bowling, 1
boules
boccie, 1

#DEF: 3. a roll or throw of the ball in bowling or bowls.
bowl
boules
roll, 15
throw, 1

Box

#DEF: 1. a container made of cardboard, wood, or other stiff material, usu. rectangular and having a lid for the top.
box, 1
container, 1
carton, 2
crate, 1
trunk, 2
package, 2
parcel, 1
chest, 2
case, 7
pack, 3
packet, 2

#DEF: 2. the amount contained in or the contents of a box; boxful.
box, 3
carton, 1
boxful, 1
case, 10

#DEF: 3. any of various enclosures that contain and protect:
#EXA: the gear box of an automobile.
box, 1
case, 7, 11, 13, 16
housing, 2
sheath, 1, 2
jacket, 2, 4
casement

#DEF: 5. an enclosed area in a theater where spectators sit.
box, 2
compartment, 2

#DEF: 6. a difficult situation; predicament; dilemma.
box, 7
dilemma, 1
predicament, 1
quandary, 1
plight, 1
conundrum, 1

#DEF: a hit or blow struck with the hand or fist.
box, 10
blow, 1
hit, 2
cuff
punch, 1
swat, 1
slap, 2
smack, 6
whack, 1
belt, 6
thwack, 1
buffet
knock, 3, 5
sock
jab, 1, 2

Boy

#DEF: 1. a male child or adolescent.
boy, 1
youth, 1
stripling, 1
child, 1, 2

son, 1
youngster, 1
adolescent, 1
teenager, 1
kid, 1, 3

#DEF: 2. (informal) a man.
boy, 2
dude, 1
chap, 1
fellow, 1
man, 1

#DEF: 3. a young immature man.
boy, 1
youth, 1
lad, 2

Car

#1. an automobile.
car, 1
automobile, 1
auto, 1
motorcar, 1
vehicle, 1
sedan, 1
coupe, 1
limousine, 1
limo, 1
convertible, 1
roadster, 1
runabout, 1
hot_rod, 1
rattletrap
jalopy, 1
crate
buggy, 1
heap, 3
cab, 3
taxi, 1
taxicab, 1
hackney, 1
hack, 4

#2. a vehicle that runs on rails, such as a streetcar or railroad car.
car, 2, 5
vehicle, 1
streetcar, 1
coach, 3
diner, 2
sleeper, 3
smoker, 3
caboose
Pullman, 1
tram, 1, 2
trolley, 1
cable_car, 1

#3. an enclosure for carrying people, as in an elevator or balloon.
car, 2, 3, 4, 5
cab, 1
elevator, 1
balloon, 1
trolley, 1
tram, 1, 2
cable_car, 1

Carpet

#DEF: 1. a heavy fabric covering for floors. (See rug.)
carpet, 1
rug, 1
mat, 1

scatter_rug, 1
area_rug
throw_rug, 1
runner

#DEF: 2. a covering similar to a carpet:
#EXA: a carpet of flowers.
carpet, 1
rug, 1
mat, 1, 3
runner
covering, 2

Cave

#DEF: 1. a natural hollow or series of hollows in the earth, esp. one with an opening in a hillside or cliff.
cave, 1
cavern, 2
grotto, 1
cove, 2
hollow, 1
cavity, 1
underground, 2
den, 2
mine, 1

#DEF: 2. an nderground storage chamber, esp. a wine cellar.
cave, 1
cellar, 1, 2, 3
wine_cellar, 1
grotto, 1
vault, 1, 2
basement, 1
chamber

Chair

#DEF: 3. the person occupying such a position; anyone who presides over a group or meeting.
chair, 3
chairperson, 1
chairman, 1
facilitator, 1
moderator, 2
head, 4

Chief

#DEF: the foremost or most important person in a group; leader.
chief, 1
leader, 1
head, 4
kingpin, 1
top
dozen
principal, 2
boss, 3
top_dog, 1
headman, 2
chieftain, 1, 2
master, 5
paramount

Child

#DEF: 1. a young

human; baby.
child, 1, 2
kid, 1, 3
youngster, 1
juvenile, 1
baby, 1
infant, 1
youth, 1
boy, 1, 3
girl, 2, 3
lad, 2
lass
stripling, 1
junior, 4
tot, 1
toddler, 1
tyke, 2
preteen

#DEF: 2. a son or daughter.
child, 2
offspring, 1
son, 1
daughter, 1
descendant, 1
progeny, 1
scion, 1
issue, 6

#DEF: 3. a descendant.
child, 2
offspring, 1
descendant, 1
progeny, 1
scion, 1
issue, 6
son, 1
daughter, 1

#DEF: 4. someone who acts in a childish or immature way.
child, 3
baby, 5
juvenile
adolescent
greenhorn

#DEF: 5. one who is considered to be the natural product of particular times or circumstances:
#EXA: a child of the revolution.
child, 2
product, 3
son, 1
daughter, 1
offshoot, 1

Church

#DEF: 1. a building for public Christian worship.
church, 2
meetinghouse, 1
tabernacle, 1
chapel, 1
cathedral, 1, 2
basilica, 1
temple, 1

#DEF: 2. such worship itself.
church, 3
worship, 1
devotion, 4
service, 3
mass, 4

communion, 1
office, 6
vespers, 2
novena
matins, 1
compline, 1

#DEF: 3. the congregation or membership of a religious denomination or sect.
church, 1
congregation, 1
fold, 2
communion
parish, 1
laity, 1
flock, 1

#DEF: 4. (often pl.) a particular Christian denomination or sect:
#EXA: the Baptist Church.
church, 1
denomination, 1
sect, 1
faith, 3
religion, 1, 2
cult, 1, 3
creed, 1, 2
persuasion, 2

#DEF: 5. the local or national organization and authority of a particular religious denomination.
church, 1
clergy, 1
ministry, 1
hierarchy, 2
episcopacy
papacy, 1
Vatican, 1
presbytery, 1
vestry, 1
Christendom, 1

#DEF: 6. organized religion in general:
#EXA: the role of the church in daily life.
church, 1
religion, 2
faith, 3
worship, 1
devotion, 3, 4

Clock

#DEF: a mechanical or electric device, other than a watch, for measuring or indicating time.
clock, 1
timekeeper, 3
timepiece, 1
chronometer, 1
time_clock, 1

Clown

#DEF: 1. a comic performer, as in a circus, who wears odd clothes and exaggerated makeup and entertains by jokes, tricks, juggling, and the like.

clown, 2
jester, 1
fool, 3
harlequin, 1
pantomime, 1
mime, 1
humorist, 1

#DEF: 2. a person who acts in a comical, prankish manner.
clown, 2
buffoon, 2
jester, 1
zany, 1
wag, 1
farceur
comedian, 1
joker, 1
cutup
merry_andrew, 1
harlequin, 1
comic, 1
prankster, 1

#DEF: 3. a crude, impolite, or oafish person.
clown, 1
boor, 1
churl, 1
lout, 1
joker, 2
brute, 1
yahoo, 1
oaf, 1

Compass

#DEF: 2. a boundary or limit, or the space or scope included within it:
#EXA: the compass of the town walls;
#EXA: the compass of the state’s authority.
compass, 2, 3
circumference, 1
limit, 1, 3, 4
circuit
perimeter, 1
periphery, 1
boundary, 1, 2
border, 1, 2
margin, 1
edge, 2
outline, 1

Cup

#DEF: someone or something that is liked or known well:
#EXA: Those people aren’t my cup of tea;
#EXA: His cup of tea is fixing computers.
cup_of_tea, 1
metier, 1
forte, 1
thing
bag, 9
partiality, 2
specialty, 1
predilection, 1
preference, 2
province

Cycle

#DEF: 1. an event or sequence of events repeated at regular or approximately regular time intervals:
#EXA: the cycle of seasons in a year;
cycle, 1
circle
round, 2
revolution, 3
series, 1
sequence, 1, 2
course, 2
rotation, 2

#DEF: 2. the time interval required for uch a sequence to occur; periodicity.
#EXA: a frequency of sixty cycles per second.
cycle, 1
period, 1
time, 2
generation, 3
session, 2
periodicity, 1

#DEF: 3. a long time; age; era.
cycle, 1
span
time, 2
eon, 2
years, 2
century, 1
decade, 1

#DEF: 4. a bicycle, unicycle, motorcycle, or the like.
cycle, 6
bike, 2
bicycle, 1

Diamond

#DEF: 2. a geometric shape with four equal straight sides, two equal, opposed acute angles, and two equal, opposed obtuse angles.
diamond, 1, 2
stone, 5
precious_stone, 1
gemstone, 1
rock, 2
gem, 2, 5
jewel, 1

Dress

#DEF: 1. a girl’s or woman’s one-piece garment consisting of a blouse connected to the waist of a skirt.
dress, 1
frock, 1
gown, 1
shift, 8
shirtwaist, 1

#DEF: 2. apparel; clothing.
dress, 2
apparel, 1
clothing, 1

raiment, 1
garb, 1
habit, 3
habiliments
duds, 1
threads, 1
toggery
wear, 2
costume, 1, 2, 3, 4
outfit, 2
getup, 1
togs, 1

#DEF: 3. formal clothing.
dress, 2
vestment, 1
attire, 1
evening_dress, 1
white_tie, 2
Sunday_best, 1
robe, 1
formal
dinner_jacket, 1
tuxedo, 1
apparel, 1
array, 3
caparison, 1
black_tie, 1

Drill

#DEF: 1. a tool consisting of a shaft that has sharp cutting edges and is used to make holes in wood, metal, or the like, usu. by means of rotation; drill bit.
drill, 1, 2
bit, 9
borer, 1
rotary

#DEF: 2. a device that holds and often powers a drill bit or drill shaft.
drill, 1, 2
borer, 1
rotary

#DEF: 3. a learning or training procedure consisting of frequent repetition of an action or item to be learned:
#EXA: a marching drill;
#EXA: a multiplication drill.
drill, 4, 5
exercise, 3
training, 1
practice, 2
regimen
routine, 1

#DEF: 4. any of various marine mollusks that kill oysters and the like by making holes in their shells.
drill
mollusk, 1
gastropod, 1

Drink

#1. a liquid for swallowing; a beverage or a certain quantity

of liquid.
drink, 3, 5
beverage, 1
quaff
liquid, 1, 3
refreshment, 1
soft_drink, 1
soda, 2
juice, 1

#2. an alcoholic beverage.
drink, 3, 5
potation, 1
intoxicant, 1
alcohol, 1
beverage, 1
liquor, 1
wine, 1
beer, 1
spirits, 1
booze, 1
sauce
moonshine, 2
firewater, 1
cocktail, 1
nightcap, 1
tipple, 1

#3. a certain quantity of alcohol.
drink, 1, 5
glass, 3
bottle, 2
can, 2
slug
brew, 1

#4. excessive use of alcohol:
#EXA: Drink caused him to lose his job.
drink, 2
insobriety
intemperance, 2
drunkenness, 1, 2
intoxication, 1
inebriety, 1
alcoholism, 1, 2
dipsomania, 1
crapulence, 1
tipple, 1
boozing, 1
bibulous

#5. (informal) a body of water:
#EXA: I fell off the boat and into the drink.
drink, 4
water, 2
ocean, 1
sea, 1
brine, 1
briny, 1
lake, 1
river, 1
pond, 1

Drum

#DEF: 2. a booming sound produced by or as if by a drum.
drum, 2
thunder, 1
rumble, 1
boom, 1
roll, 8, 9
growl, 1
roar, 1
resonance, 3

reverberation, 1

Earth

#DEF: 1. (often cap.) the fifth largest planet in the solar system, which is third in order from the sun and has a diameter of about 7, 930 miles.
earth, 1
globe, 1

#DEF: 2. all of the inhabitants that dwell upon Earth:
#EXA: Earth prays for peace.
earth
world, 1
mankind, 1
humanity, 1
race, 3
population, 1
people, 1

#DEF: 3. the outer layer of the planet; ground.
earth, 2, 3
land, 4
ground, 1, 3
soil, 2
terra_firma, 1
topsoil, 1
clay, 1

#DEF: 4. soil or dirt.
earth, 2
ground, 3, 7
soil, 2, 3
dirt, 1
terra_firma, 1
clay, 2
sod, 1
dust, 1

Electricity

#DEF: 2. the science concerned with such a phenomenon and its effects.
electricity
polarity
magnetism

#DEF: 4. a state of tension or excitement.
electricity, 3
current, 1
direct_current, 1
DC, 2
alternating_current, 1
AC, 2
power, 2
juice

Explosive

#DEF: a substance that is capable of causing an explosion, esp. an agent prepared for that purpose, such as dynamite.
#DER:
explosively, adv. ;
#DER:

explosiveness, n.
explosive, 1
fulminate
detonator, 1
charge, 15
dynamite, 1
TNT, 1
trinitrotoluene, 1
gunpowder, 1
cordite, 1
gelignite, 1
fuse, 2

Eye

#DEF: 1. the organ of sight and the area close around it, including the lids, lashes, and brow.
eye, 1
orb, 1
eyeball, 1
peeper, 2

#DEF: 2. skill in observing:
#EXA: an eye for color.
eye, 2
sense, 3
sensitivity, 1
awareness, 1
judgment, 7
discernment, 4
perception, 4

#DEF: 3. (usu. pl.) judgment or understanding:
#EXA: In society’s eyes, they are outlaws.
eye, 2
view, 5
opinion, 1
judgment, 1
understanding, 1
estimation, 4
ken, 1

#DEF: 4. close attention:
#EXA: Keep an eye on my things while I’m gone.
eye, 3
watch, 3
lookout
attention, 1

#DEF: 6. the center, as of a storm.
eye, 4
center, 1
middle, 1
hub, 2
heart, 4
midst, 1

#DEF: 7. any of various things that resemble an eye:
#EXA: the eye of the needle;
#EXA: the eye of the target.
eye, 5
eyelet, 1
grommet
hole
slit, 1

Family

#DEF: 2. such ancestors themselves.
family_tree, 1
ancestor, 1
predecessor, 1
forerunner, 1
forebear, 1
ancestress, 1
progenitor, 1
ancestry, 1
antecedent, 1
root
genealogy, 1
parentage, 2
background

#DEF: 1. a group consisting of parents and their children.
family, 1
folk, 3
household, 1
house, 4
menage, 1
kindred, 1
clan, 1
kin, 1, 2
extended_family, 1

#DEF: 2. all of one’s ancestors and descendants; those related by blood kinship.
family, 4, 5
relation, 3
people, 4
kin, 1, 2
kinfolk, 1
kindred, 1
ancestry, 1
relative, 1
folk, 3
descendants, 1
offspring, 1
progeny, 1
ancestor, 1

#DEF: 3. all those persons descended from a common ancestor.
family, 4
descendants, 1
offspring, 1
progeny, 1
posterity, 1
lineage, 1
relation, 3
people, 4
kinfolk, 1
kindred, 1
relative, 1
folk, 3
parentage, 2

#DEF: 4. any group living together, as if they were related by blood, in a single household.
family, 1
household, 1
menage, 1
house, 4
folk, 3
kindred, 1
clan, 1
people, 4

#DEF: 5. any group of things related in form, function, or period of manufacture or origin.
family, 3
class, 1

genus, 1
category, 1
group, 1
kind, 1
type, 1
order, 8, 10

Fan

#DEF: 1. a mechanical apparatus, usu. driven by electricity, that creates an air current by moving several vanes or blades in rotation.
fan, 1
blower, 1, 2
air_conditioning, 1
ventilator, 1

#DEF: 2. a hand-held device that opens out to form a triangular shape and that is used to cool the face or body by waving back and forth.
fan, 1
palm_leaf
punkah, 1

#DEF: an enthusiastic follower of an activity such as a sport or a performing art, for of a person or persons who engage in tha activity:
#EXA: a football fan;
#EXA: the fans of a movie star.
fan, 2, 3
enthusiast, 1
afficionado, 1
devotee, 1
buff, 1
fancier, 1
follower
hound
addict, 1
nut, 5
fiend, 3
junkie
groupie, 1
votary, 3
disciple, 1

Feather

#DEF: 4. condition or character:
#EXA: in fine feather;
feather
shape, 6
condition, 1
trim, 1
form, 7
order, 5
fettle, 1
kilter
health, 1

Festival

#DEF: 1. a day or more of celebration to commemorate a notable occasion, such as a religious holiday.

festival, 1
holiday, 2
feast, 3
fete, 1
celebration, 2
observance, 2
holy_day, 1
gala, 1
revel, 1
jubilee, 1
ceremony, 1
carnival, 1
Saturnalia, 1
fiesta, 1
name_day, 1
occasion, 2

#DEF: 2. a regularly occurring ceremony or celebration:
#EXA: the harvest festival.
festival, 1
celebration, 2
fete, 1
observance, 2
ceremony, 1
revel, 1
jubilee, 1
gala, 1
carnival, 1
Saturnalia, 1
fiesta, 1
occasion, 2

#DEF: 3. a series of presentations, or a gathering of exhibitors in one or more of the fine arts, theater arts, or crafts, or such a gathering based on a central theme, food, season, or the like:
#EXA: a music festival;
#EXA: the apple festival.
festival, 2
fair, 1
carnival, 3

Film

#DEF: 4. (often cap.) motion pictures generally, or the motion picture industry.
film, 1
movie, 1
motion_picture, 1
moving_picture, 1
picture, 6
feature, 3
showing, 1
show, 4
screening, 1
cinema

Fire

#DEF: a hydrant to which a firefighting hose can be attached; fireplug.
fire_hydrant, 1
hydrant, 2
fireplug, 1
plug, 6

#DEF: 1. the ffects, such as heat, light, and flames,

produced by burning.
fire, 3
flame, 1
combustion, 1
light, 1
spark, 1
glow, 3, 5
illumination, 3
incandescence, 2
sparkle, 2, 3
energy, 1
heat, 1, 2
radiance, 2

#DEF: 2. a particular burning, as in a stove or furnace.
fire, 1, 3
blaze, 1
bonfire, 1

#DEF: 3. an instance of destructive burning:
#EXA: There was a fire at the library last night.
fire, 1
blaze, 1
conflagration, 1
holocaust
wildfire, 1
inferno, 3
flare-up

#DEF: 4. passion or imaginative excitement:
#EXA: the fire of her poetry.
fire, 6
fervor, 1
ardor, 3
passion, 1
heat, 4
verve, 1
enthusiasm, 1
power, 1
vehemence, 1
intensity, 1, 2
imagination, 1

#DEF: 5. a severe trial.
fire, 7
trial, 6
ordeal, 1
trouble, 2, 3, 4
affliction, 1, 3
torture

#DEF: 6. the discharging of a weapon or weapons.
fire, 2
discharge, 9
shot, 3
flak, 4
fusillade, 1
volley, 1
barrage, 2
salvo, 2
cannonade, 1
enfilade
gunfire, 1

Flower

#DEF: 2. a plant capable of producing blossoms, grown primarily for visual enjoyment.
flower, 1
blossom, 1
inflorescence, 2

bloom, 2
bud, 1

#DEF: 3. the best or most flourishing example or state of something:
#EXA: He was the flower of his generation;
flower, 3
prime, 2
efflorescence, 1
heyday, 1
bloom, 5
summit, 1
peak, 2
zenith
climax, 4
flush, 1

Foot

#DEF: 4. the part of something that is lowest or opposite the head:
#EXA: the foot of the cliff;
#EXA: the foot of the bed.
foot, 3, 5
base, 2, 5, 8
rock_bottom, 1
bottom, 2
nadir
foundation, 3
belly
floor, 3, 5

Freeway

#DEF: a highway with limited access and no tolls; expressway.
freeway, 1
thruway, 1
turnpike, 2
pike, 1
interstate
expressway, 1
route, 2
parkway, 1
autobahn, 1
highway, 1
speedway, 1

Fruit

#DEF: 2. something that is a result or outcome:
#EXA: These are the fruit of my efforts.
fruit, 2
child, 2
product, 3
progeny, 1
issue, 6, 7
offspring, 1, 2
outcome, 1, 2
result, 1, 3
descendant, 1
heir, 2
offshoot, 1
spawn

Fungus

#DEF: any organism, including mushrooms, yeasts, molds, rusts, and others, characterized by

lack of chlorophyll and by subsistence on organic matter.
fungus, 1
mold, 5
mildew, 2
smut, 3
rust, 4
parasite, 1

Game

#DEF: 2. any planned strategy to reach an objective.
game_plan, 1
strategy, 1
plan, 1
game
scheme, 1
stratagem, 2

#DEF: 1. something done for amusement; diversion; pastime.
game, 3
diversion, 1
pastime, 1
distraction, 3
entertainment, 1
recreation, 1
amusement, 2
play, 14
fun, 1

#DEF: 2. a usu. competitive form of play or sport having certain rules and equipment for play:
#EXA: a game of chess;
#EXA: a football game.
game, 2, 3
sport, 1
play
competition, 2
match, 2
contest, 1

#DEF: 3. a strategy or plan.
game
strategy, 1
plan, 1
game_plan, 1
scheme, 1
stratagem, 2

#DEF: 4. wild animals hunted for sport or food.
game, 4
quarry, 3
wildlife, 1
big_game, 1
take

#DEF: 5. the flesh of such animals, used for food.
game, 8
fowl, 2
meat, 1
take

Garden

#DEF: 2. (often. pl.) a public park or recreation area, often

devoted to the housing and display of plants or animals.
garden, 1
plot, 1
patch, 2
bed, 2
plat
flat

Gas

#DEF: 8. (slang) something amusing or astonishing:
#EXA: Those crobats were a gas.
gas
chatter, 1
gossip, 1
small_talk, 1
chitchat, 1
prate, 1
palaver, 2
gab, 1
jabber, 1
babble, 1
patter, 1
prattle, 1
gibber, 1
twaddle, 1
blather, 1
blab
blast, 5
ball

Gate

#DEF: 2. a passage for entrance or exit.
gate, 1
portal, 1
entry, 5
entranceway, 1
entryway, 1
ingress, 1
door, 2
doorway, 1
hall, 1, 2
entrance, 1
inlet
approach, 3
driveway, 1
adit, 1
opening, 1
hatch, 3
postern, 1
hallway, 1
foyer, 1
access, 3

Gemstone

#DEF: a precious stone fine enough to cut and polish for jewelry.
gemstone, 1
gem, 2
bijou, 1
stone, 5
jewel, 1, 2
sparkler
rock

Girl

#DEF: 1. a female child or adolescent.
girl, 2
female, 2

maiden, 1
lass, 1
filly, 1
maid, 2
gal, 3
nymph, 3

#DEF: 2. an intimate female friend; sweetheart.
girl, 4
girlfriend, 2
sweetheart, 1
inamorata
lass, 1

#DEF: 3. (informal) a woman.
girl, 1, 5
woman, 1
lady, 1, 2
female, 2

God

#DEF: 2. (cap.) the omnipotent and omniscient being that is worshiped by Christians, Jews, and Muslims as the creator and ruler of the universe.
god, 1
angel, 1
seraph, 1
cherub, 2
archangel, 1
goddess, 1
celestial

#DEF: 3. a physical image or representation of a supernatural being; idol.
god, 4
Mammon, 1, 2
deity, 1
religion
golden_calf, 1
idol, 1
effigy, 1
statue, 1
relic, 2
fetish, 1
joss, 1

Appendix E

The Maximum Entropy Framework

A training sample of data yields some information about the decisions made within various contexts; however this only accounts for a small portion of all possible situations due to the sparse nature of the data for the task being modelled. The task of ME is to train a classifier, $p(d|c)$, that conforms to the empirical distributions of the training sample but in addition remains as uniform as possible for all other possibilities. Given information about how features affect decisions made in the test data, the task is to find a classifier that uses these features to calculate $p(d|c)$. That is to say, the principal of maximum entropy is:

> "To select a model from a set C of allowed probability distributions, choose the model $p_* \in C$ with maximum entropy H(p)"

$$p_* = \arg\max_{p \in C} H(p) \tag{E.1}$$

where $H(p)$ is the measure of uniformity. Berger et al. (1996) give the mathematical measure of conditional entropy as a measure of the uniformity of $p(d|c)$, as shown in equation E.2.

$$H(p) = -\sum_{c,d} \tilde{p}(c) p(d|c) \log p(d|c) \tag{E.2}$$

To ensure that the classifier will conform to the information about the features, the set C of allowable classifiers is defined by equation E.3.

$$C \equiv \{p \in P | p(f_i) = \tilde{p}(f_i) \wedge i \in \{1, 2, \ldots, n\}\} \tag{E.3}$$

where n is the number of features used by the classifiers.

Ratnaparkhi (1997) gives a simple example of the use of maximum entropy. Consider the task of estimating the distribution $p(c, d)$, where there are possible contexts $c \in \{x, y\}$ and possible decisions $d \in \{0, 1\}$. The only prior information available is that $p(x, 0) + p(y, 0) = 0.6$. Constructing a feature from the given information produces function E.4 and the probability in E.5.

$$f_1(c, d) = \begin{cases} 1 & : \text{ if } d = 0 \\ 0 & : \text{ otherwise} \end{cases} \tag{E.4}$$

$$\tilde{p}(f_1) = 0.6 \tag{E.5}$$

It is apparent that there are a large number of distributions that will satisfy the feature, such as table E.1. However, the maximum entropy approach selects the classifier deemed to be most uniform, or non-committal, given in table E.2. For small

p(c, d)	0	1	
x	0.1	0.3	
y	0.5	0.1	
Total	0.6	0.4	1

Table E.1: One Way To Satisfy The Constraints

p(c, d)	0	1	
x	0.3	0.2	
y	0.3	0.2	
Total	0.6	0.4	1

Table E.2: The Most Uniform Way To Satisfy The Constraints

examples, such as the one outlined above, the calculation for the distribution is trivial. However, for most problems of interest this is not the case. For such cases, an alternative approach is required. Berger et al. (1996) and Berger (1997) give a method using Lagrange Multipliers from the theory of constrained optimisation:

- The problem of finding $p_* \in C$ in the original optimisation problem is referred

as the primal problem, given in equation E.6.

$$\exists p_* = \arg\max_{p \in C} H(p) \tag{E.6}$$

- For each feature, f_i, a Lagrange multiplier, λ_i, is introduced. The Lagrangian $\Lambda(p, \lambda)$ is defined as E.7.

$$\Lambda(p, \lambda) = H(p) + \sum_{i=1}^{n} \lambda_i (p(f_i) - \tilde{p}(f_i)) \tag{E.7}$$

  where n is the number of features.

- The task is to find p_λ, the classifier where $\Lambda(p, \lambda)$ reaches its maximum. For this a new definition for $p(f_i)$ is required using the Lagrange multipliers. A dual problem, $\Psi(\lambda)$, is maximised to find the values of λ. When this maximum is reaches, $\Psi(\lambda)$ will be equal to $\Lambda(p, \lambda)$.

$$p_\lambda \equiv \arg\max_{p \in P} \Lambda(p, \lambda) \tag{E.8}$$

$$\Psi(\lambda) \equiv \Lambda(p, \lambda) \tag{E.9}$$

The dual problem expresses the conditional distribution $p(d|c)$ in terms of the features that are active, where a feature is active when its value is 1. The Lagrange multipliers, λ, weights the affect of each of the features in the final classifier. Equations E.10 to E.13 define the dual problem, and the new definition for calculating $p(d|c)$ using the Lagrange multiples.

$$p_\lambda(d|c) = \frac{1}{Z_\lambda(c)} \exp\left(\sum_{i=1}^{n} \lambda_i f_i(c, d)\right) \tag{E.10}$$

$$Z_\lambda(c) = \sum_{d \in D} \exp\left(\sum_{i=1}^{n} \lambda_i f_i(c, d)\right) \tag{E.11}$$

$$p_\lambda(f) = \sum_{\substack{c \in C, \\ d \in D}} \tilde{p}(c) p_\lambda(d|c) f(c, d) \tag{E.12}$$

$$\Psi(\lambda) = -\sum_{c \in C} \tilde{p}(c) \log Z_\lambda(c) + \sum_{i=1}^{n} \lambda_i \tilde{p}(f_i) \tag{E.13}$$

Maximising the unconstrained dual function (E.13) gives a set of Lagrange multipliers that also maximise $\Lambda(p, \lambda)$, therefore we solve the dual optimisation problem by satisfying E.14.

$$\exists \lambda_* = \arg\max_{\lambda} \Psi(\lambda) \tag{E.14}$$

It will sometimes be the case that λ_* cannot be calculated exactly. In the iterative training algorithm introduced later in this section, each iteration, i, produces a set of Lagrange multipliers, λ_i. $\Psi(\lambda)$ increases for each iteration of the algorithm, therefore λ_* can be estimated due to the fact that $\lambda_* \leftarrow \lambda_n$.

A fundamental principal of the theory of Maximum Entropy, the Kuhn-Tucker theorem, reinforces the relationship between the primal and dual problems given here. So it follows, as Berger et al. (1996) state:

> "The maximum entropy model subject to the constraints C has the parametric form p_λ, where the parameters values λ_* can be determined by maximising the dual function $\Psi(\lambda)$." (Berger et al., 1996)

The optimal set of Lagrange multiples, λ_*, can be calculated using a number of numerical methods given that the dual function $\Psi(\lambda)$ produces a smooth convex-$\cap$ graph against λ. Berger et al. (1996); Berger (1997) describe an improved iterative scaling (IIS) algorithm to calculate the Lagrange multipliers. The algorithm itself is a generalisation of the Darroch-Ratcliff procedure, and a proof for the convergence of the algorithm is given by Pietra et al. (1997, 1995). The algorithm can be applied to any model that meets the criteria in E.15.

$$\forall c \in C \cdot \forall d \in D \cdot \forall i \in \{1, 2, \ldots, n\} \cdot f_i(c, d) \geq 0 \tag{E.15}$$

where n is the number of features.

The main task of the algorithm in listing D.1 is to calculate the $\Delta\lambda_i$ that satisfies the equality in 2a for each iteration. If $\#f(c, d)$ is constant for f_i, $\Delta\lambda_i$ can be calculated directly. Rearranging the equality in 2a produces the equation in E.16.

$$\Delta\lambda_i = \frac{1}{\#f(c, d)} \ln \frac{\tilde{p}(f_i)}{p_\lambda(f_i)} \tag{E.16}$$

Algorithm D.1: Improved Iterative Scaling (IIS) Algorithm

1. $\forall i \in \{1, 2, \ldots, n\}, \lambda_i = 0$
2. $\forall i \in \{1, 2, \ldots, n\}$,
   (a) Let $\Delta\lambda_i$ be the solution to
   $\sum_{c,d} \tilde{p}(d|c) f_i(c,d) \exp\left(\Delta\lambda_i \# f(c,d)\right) = \tilde{p}(f)$
   where $= f(c,d) \equiv \sum_{i=1}^{n} f_i(c,d)$.
   (b) Update the value of λ_i according to: $\lambda_i \leftarrow \lambda_i + \Delta\lambda_i$.
3. Return to step 2 if not all λ_i have converged.

If $\# f(c,d)$ is not constant for f_i, Newton's method is applied to find $\Delta\lambda_i$. Newtons method is illustrated in equation E.17.

$$\alpha_{n+1} = \alpha_n - \frac{g(\alpha_n)}{g'(\alpha_n)} \tag{E.17}$$

For the dual problem (E.13), $g(\alpha_n)$ is calculated using E.18.

$$g(\alpha_n) = \left(\sum_{c,d} \tilde{p}(c) p_\lambda(d|c) f_i(c,d) \exp\left(\alpha_n \# f(c,d)\right) \right) - \tilde{p}(f_i) \tag{E.18}$$

The derivative of $g(\alpha_n)$ than be calculated trivially given rules E.19 and E.20.

$$f(x) = mc^{nx} - k \tag{E.19}$$

$$f'(x) = nmc^{nx} \tag{E.20}$$

where c, k, m and n are constants. Using these rules, $g'(\alpha_n)$ is given by differentiating $g(\alpha_n)$ is respect to α_n, producing E.21.

$$g(\alpha_n) = \left(\sum_{c,d} \# f(c,d) \tilde{p}(c) p_\lambda(d|c) f_i(c,d) \exp\left(\alpha_n \# f(c,d)\right) \right) - \tilde{p}(f_i) \tag{E.21}$$

The recursive algorithm runs until $g(\alpha_*) = 0$ is satisfied, and $\Delta\lambda_i = \alpha_*$. In the implementation used, it was found that Newton's method worked well using E.22 for α_0.

$$\alpha_0 = \frac{1}{average(\# f(c,d))} \ln \frac{\tilde{p}(f_i)}{p_\lambda(f_i)} \tag{E.22}$$

An alternative divide-and-conquer algorithm, shown in D.2, can also be used to find $g(\alpha_*) = 0$ when selecting α_0 becomes problematic. If a change of sign is detected between two values, α_u and α_l, α_* must lie between the two values. A divide-and-conquer algorithm is applied to find the value of α_* by testing the mean, α_m of α_u and α_l, and replacing the relevant α to reduce the range of values considered until α_u and α_l converge. At this point $g(\alpha_*) = 0$. This is shown diagrammatically in Figure E.1.

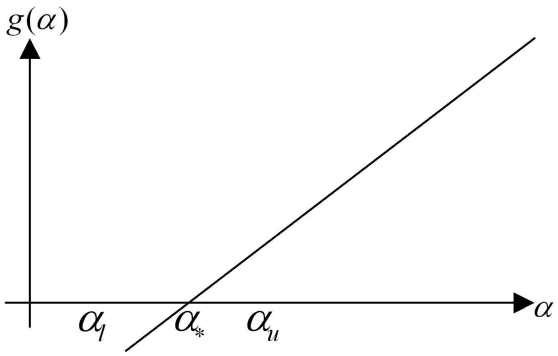


Figure E.1: Illustration of Divide-and-Conquer Algorithm

Algorithm D.2: Divide-and-Conquer Algorithm to Calculate α_*

1. Start with two values, upper bound α_u and lower bound α_l.
2. Calculate the average of the two values α_m.
3. If not$(g(\alpha_u) = 0)$ Then
   (a) Calculate $s = sign(g(\alpha_m))$
   (b) If (s = "+") Then α_u = α_m Else α_l = α_m

   Else
   (a) Return α_m
4. Goto 2

Given that α_* lies between α_u and α_l, the divide-and-conquer algorithm is guaranteed to find α_*.

Appendix F

Distribution of Examples for Word Sense Disambiguation Tests

Word	Number of Training Examples	Number of Similar Training Examples	Number of Test Examples	Split
Dog	25	364	13	65.79%
Eye	113	72	45	71.52%
Family	83	407	45	64.84%
Give	406	32	178	69.52%
Information	81	225	40	66.94%
Instruction	10	270	5	66.67%
Party	46	299	16	74.19%
Report	47	292	22	68.12%
Suggestion	16	279	6	72.73%
Vote	12	110	6	66.67%
Work	142	252	59	70.65%

Table F.1: Data Available for Each Word of Interest

Dog Sense	Number of Training Examples	Number of Similar Training Examples	Number of Test Examples	Split Ratio
1	25	4	13	65.79%
2	0	291	0	No Examples
3	0	291	0	No Examples
4	0	275	0	No Examples
5	0	13	0	No Examples
6	0	14	0	No Examples

Table F.2: Data Available for “Dog”

Eye Sense	Number of Training Examples	Number of Similar Training Examples	Number of Test Examples	Split Ratio
1	104	8	43	70.75%
2	6	0	0	100.00%
3	3	2	2	60.00%
4	0	61	0	No Examples
5	0	1	0	No Examples

Table F.3: Data Available for “Eye”

Family Sense	Number of Training Examples	Number of Similar Training Examples	Number of Test Examples	Split Ratio
1	40	34	19	67.80%
2	31	5	18	63.27%
3	6	16	4	60.00%
4	4	36	2	66.67%
5	1	317	1	50.00%
6	1	31	1	50.00%
7	0	71	0	No Examples

Table F.4: Data Available for “Famliy”

Give Sense	Number of Training Examples	Number of Similar Training Examples	Number of Test Examples	Split Ratio
1	88	0	38	69.84%
2	88	10	39	69.29%
3	53	151	23	69.74%
4	39	1	17	69.64%
5	13	0	6	68.42%
6	16	4	8	66.67%
7	14	0	7	66.67%
8	13	0	6	68.42%
9	13	0	5	72.22%
10	7	0	2	77.78%
11	8	0	3	72.73%
12	6	3	3	66.67%
13	7	0	2	77.78%
14	6	0	3	66.67%
15	3	0	1	75.00%
16	4	0	2	66.67%
17	7	11	3	70.00%
18	4	1	1	80.00%
19	2	0	1	66.67%
20	4	0	1	80.00%
21	3	0	1	75.00%
22	1	0	1	50.00%
23	1	0	1	50.00%
24	4	1	2	66.67%
25	1	2	1	50.00%
26	0	0	0	No Examples
27	1	0	1	50.00%
28-45	0	0	0	No Examples

Table F.5: Data Available for “Give”

Information Sense	Number of Training Examples	Number of Similar Training Examples	Number of Test Examples	Split Ratio
1	50	225	25	66.67%
2	0	140	0	No Examples
3	30	21	15	66.67%
4	0	22	0	No Examples
5	1	40	0	100.00%

Table F.6: Data Available for “Information”

Instruction Sense	Number of Training Examples	Number of Similar Training Examples	Number of Test Examples	Split Ratio
1	5	134	3	62.50%
2	4	108	1	80.00%
3	1	21	1	50.00%
4	0	114	0	No Examples

Table F.7: Data Available for “Instruction”

Party Sense	Number of Training Examples	Number of Similar Training Examples	Number of Test Examples	Split Ratio
1	25	29	9	73.53%
2	4	2	0	100.00%
3	7	2	3	70.00%
4	9	2	4	69.23%
5	1	264	0	100.00%

Table F.8: Data Available for “Party”

Report Sense	Number of Training Examples	Number of Similar Training Examples	Number of Test Examples	Split Ratio
1	34	136	16	68.00%
2	7	145	1	87.50%
3	5	4	4	55.56%
4	1	12	1	50.00%
5	0	170	0	No Examples
6	0	184	0	No Examples
7	0	0	0	No Examples

Table F.9: Data Available for “Report”

Suggestion Sense	Number of Training Examples	Number of Similar Training Examples	Number of Test Examples	Split Ratio
1	9	17	4	69.23%
2	6	177	2	75.00%
3	1	49	0	100.00%
4	0	3	0	No Examples
5	0	33	0	No Examples

Table F.10: Data Available for “Suggestion”

Vote Sense	Number of Training Examples	Number of Similar Training Examples	Number of Test Examples	Split Ratio
1	11	13	3	78.57%
2	0	20	3	0.00%
3	1	27	0	100.00%
4	0	23	0	No Examples
5	0	40	0	No Examples

Table F.11: Data Available for “Vote”

Work Sense	Number of Training Examples	Number of Similar Training Examples	Number of Test Examples	Split Ratio
1	56	98	23	70.89%
2	49	44	25	66.22%
3	22	55	6	78.57%
4	12	0	3	80.00%
5	0	73	0	No Examples
6	0	53	1	0.00%
7	3	9	1	75.00%

Table F.12: Data Available for “Work”

Bibliography

Web interface to roget's thesaurus. http://ecco.bsee.swin.edu.au/text/roget/entries/<category number>.html.

Web interface to WordNet 1.6. http://www.cogsci.princeton.edu/cgi-bin/webwn.

Agirre, E. and G. Rigau (1995, 14-16 September). A proposal for word sense disambiguation using conceptual distance. In *International Conference on Recent Advances in Natural Language Processing*, Tzigov Chark, Bulgaria.

Agirre, E. and G. Rigau (1996, August). Word sense disambiguation using conceptual density. In *Proceedings of the 16^{th} International Conference on Computational Linguistics (COLING-96)*, Copenhagen, Denmark, pp. 16–22.

ALPAC (1966). Languages and machines: Computers in translation and linguistics. a report by the Automatic Language Processing Advisory Committee, division of behavioral sciences, national academy of sciences, national research council. Technical report, Washington, D.C.: National Academy of Sciences, National Research Council. (Publication 1416.).

Alshawi, H. and D. Carter (1994). Training and scaling preference functions for disambiguation. *Computational Linguistics 20*(4), 635–648.

Atkins, S. (1993). Tools for computer-aided lexicography: The hector project. In *Papers in Computational Lexicography (COMPLEX '93)*, Volume 4(3), Budapest, Hungary, pp. 167–204.

Baker, C. F., C. J. Fillmore, and J. B. Lowe (1998). The Berkeley FrameNet project. In *COLING-ACL*, pp. 86–90.

Beeferman, D., A. Berger, and J. D. Lafferty (1999). Statistical models for text segmentation. *Machine Learning 34*(1-3), 177–210.

Berger, A. L. (1997). The improved iterative scaling algorithm: A gentle introduction, http://www.cs.cmu.edu/afs/cs/user/aberger/www/ps/scaling.ps.

Berger, A. L., S. A. D. Pietra, and V. J. D. Pietra (1996). A maximum entropy approach to natural language processing. *Computational Linguistics 22*(1), 39–71.

Boguraev, B. K. (1979). *Automatic Resolution of Linguistic Ambiguities*. Ph. D. thesis, Computer Laboratory, University of Cambridge.

Boser, B. E., I. M. Guyon, and V. N. Vapnik (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the 5^{th} ACM Workshop on Computational Learning Theory*, pp. 144–152.

Braden-Harder, L. (1993). Sense disambiguation using on-line dictionaries. In K. Jensen, G. E. Heidorn, and S. D. Richardson (Eds.), *Natural Language Processing: The PLNLP Approach*, pp. 247–261. Kluwer Academic, Dordrecht.

Brill, E. (1991). Discovering the lexical features of a language. In *Proceedings of the 29^{th} Annual Meeting of the Association for Computational Linguistics*, Berkeley, CA.

Brill, E. (1992). A simple rule-based part of speech tagger. In *Proceedings of ANLP-92, 3^{rd} Conference on Applied Natural Language Processing*, Trent, Italy, pp. 152–155.

Brown, P. F., J. Cocke, S. A. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer and P. S. Roossin (1990, June). A statistical approach to machine translation. *Computational Linguistics 16*(2), 79–85.

Bruce, R. and L. Guthrie (1992, August). Genus disambiguation: A study in weighted preference. In *Proceedings of the 14^{th} International Conference on Computational Linguistics (COLING'92)*, Nantes, France, pp. 1187–1191.

Bruce, R. and J. Wiebe (1994). Word sense disambiguation using decomposable models. In *Proceedings of the ACL-94, 32^{nd} Annual Meeting of the Association for Computational Linguistics*, Las Cruces, US, pp. 139–145.

Budanitsky, A. (1999). *Lexical Semantic Relatedness and Its Application in Natural Language Processing*. Ph. D. thesis, Department of Computer Science, University of Toronto.

Budanitsky, A. and G. Hirst (2001, June). Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and Other Lexical Resources, in the North American Chapter of the Association for Computational Linguistics (NAACL-2001)*, Pittsburgh, PA.

Burnard, L. (1995). *Users Reference Guide for the British National Corpus*. Oxford University Computing Services, Oxford.

Byrd, R. J., N. Calzolari, M. S. Chodorow, J. L. Klavans, M. S. Neff, and O. Rizk (1987). Tools and methods for computational linguistics. *Computational Linguistics 13(3/4)*, 219–240.

Calzolari, N. (1984, July). Detecting patterns in a lexical data base. In *Proceedings of the 10^{th} International Conference on Computational Linguistics (COLING-ACL'84)*, Stanford University, California, pp. 170–173.

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

Chen, K.-J. and J.-M. You (2002, August). A study on word similarity using context vector models. *Computational Linguistics and Chinese Language Processing 7*(2), 37–58.

Chklovski, T. and R. Mihalcea (2002). Building a sense tagged corpus with open mind word expert. In *Proceedings of the Workshop on Word Sense Disambiguation: Recent Successes and Future Directions (ACL'02)*, Philadelphia, PA, pp. 116–122.

Chodorow, M. S., R. J. Byrd, and G. E. Heidorn (1985, July). Extracting semantic hierarchies from a large on-line dictionary. In *Proceedings of the 23^{th} Annual Meeting of the Association for Computational Linguistics (ACL)*, University of Chicago, Chicago, pp. 299–304.

Chomsky, N. (1957). *Syntactic Structures*. Mouton and Co., The Hague.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. Cambridge, MA: MIT Press.

Choueka, Y. and S. Lusignan (1985). Disambiguation by short contexts. *Computers and the Humanities 19*, 147–157.

Collins, A. M. and E. F. Loftus (1975). A spreading activation theory of semantic processing. *Psychological Review 82*(6), 407–428.

Cortes, C. and V. N. Vapnik (1995). Support-vector networks. *Machine Learning 20*(3), 273–297.

Cottrell, G. W. (1985). *A Connectionist Approach to Word-Sense Disambiguation*. Ph. D. thesis, Department of Computer Science, University of Rochester.

Cowie, J., J. A. Guthrie, and L. Guthrie (1992, August). Lexical disambiguation using simulated annealing. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING'92)*, Nantes, France, pp. 359–365.

Daelemans, W., J. Zavrel, K. van der Sloot, and A. van den Bosch (1998). TiMBL: Tilburg memory based learner version 1.0, reference guide. Technical Report 98-03, ILK.

Dang, H. T. and M. Palmer (2002, July). Combining contextual features for word sense disambiguation. In *Proceedings of the Workshop on Word Sense Disambiguation: Recent Successes and Future Directions (ACL'02)*, Philadelphia, PA, pp. 88–94.

Demetriou, G. C. (1993). Lexical disambiguation using constraint handling in prolog (chip). In *Proceedings of the European Chapter of the ACL*, Volume 6, pp. 431–436.

Dionisio, N., I. Marshall, and Éva Sáfár (2001, September). Using hyponym branching similarity measures comparable to statistical alternatives for word sense. In *Proceedings of Recent Advances in Natural Language Processing 2001 (RANLP-2001)*, Tzigov Chark, Bulgaria, pp. 267–270. (A longer unpublished version is provided in appendix A.)

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

Escudero, G., L. Màrquez, and G. Rigau (2000). Boosting applied to word sense disambiguation. In R. L. de Mántaras and E. Plaza (Eds.), *Proceedings of the 11th*

European Conference on Machine Learning (ECML-00), Barcelona, Spain, pp. 129–141. Springer Verlag, Heidelberg, DE.

Fellbaum, C. D. (Ed.) (1998). *WordNet: An Electronic Lexical Database*. MIT Press.

Finkelstein, L., E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin (2002, January). Placing search in context: The concept revisited. In *Proceedings of the ARPA Human Language Technology Workshop*, pp. 116–131.

Francis, W. N. (1980). A tagged corpus: Problems and prospects. In *Studies in English Linguistic for Randolph Quirk*, pp. 192–209.

Francis, W. N. and H. Kucera (1982). *Frequency Analysis of English Usage*. Houghton Mifflin.

Gaizauskas, R., T. Wakao, K. Humphreys, H. Cunningham, and Y. A. Wilks (1995). University of sheffield: Description of the LaSIE system as used for MUC-6. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pp. 207–220. Morgan Kaufmann, San Francisco, CA.

Gale, W. A., K. W. Church, and D. Yarowsky (1992a). Estimating upper and lower bounds on the performance of word-sense disambiguation programs. In *Proceedings of the 30th Annual Meeting for Computational Linguistics ACL'92*, pp. 249–256.

Gale, W. A., K. W. Church, and D. Yarowsky (1992b). One sense per discourse. In *Proceedings of the DARPA Speech and Natural Language Workshop*, New York, pp. 233–237.

Gale, W. A., K. W. Church, and D. Yarowsky (1993). A method for disambiguating word senses in a large corpus. *Computers and the Humanities 26*, 415–439.

Gonzalo, J., I. Chugur, and F. Verdejo (2003). The web as a resource for WSD. In *1^{st} MEANING workshop: Word Sense Disambiguation and Lexical Acquisition*, Number CS-96-05, Miramar Jauregia, Donostia, Basque Country.

Gougenheim, G. and R. Michéa (1961). Sur la détermination du sens d'un mot au moyen du contexte. In *La Traduction Automatique*, Volume 2, pp. 16–17.

Green, R., L. Pearl, and B. J. Dorr (2001a). Mapping lexical entries in a verbs database to WordNet senses. Technical Report LAMP-TR-068, University of Maryland, College Park.

Green, R., L. Pearl, B. J. Dorr, and P. S. Resnik (2001b). Mapping lexical entries in a verbs database to WordNet senses. In *Meeting of the Association for Computational Linguistics*, pp. 244–251.

Guiasu, S. and A. Shenitzer (1985). The principle of maximum entropy. *The Mathematical Intelligencer 7*(1), 43–48.

Guthrie, J. A., L. Guthrie, Y. A. Wilks, and H. Aidinejad (1991). Subject-dependent co-occurrence and word sense disambiguation. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, Berkeley, CA, pp. 146–152.

Guthrie, L., J. Pustejovsky, Y. A. Wilks, and B. M. Slator (1996, January). The role of lexicons in natural language processing. *Communications of the ACM 39*, 63–72.

Haegeman, L. (1994). *Introduction to Government and Binding Theory*. Blackwell, Oxford.

Harley, A. and D. Glennon (1997, April). Sense tagging in action: Combining different tests with additive weights. In *Proceedings of the ACL-SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, Association for Computational Linguistics, Washington, D.C., pp. 74–78.

Harper, K. E. (1957a). Contextual analysis. *Mechanical Translation 4*(3), 70–75.

Harper, K. E. (1957b). Semantic ambiguity. *Mechanical Translation 4*(3), 68–69.

Hayes, P. J. (1976). On semantic nets, frames and associations. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, Cambridge, MA, pp. 99–107.

Hearst, M. A. (1991). Noun homograph disambiguation using local context in large corpora. In *Proceedings of the 7th Annual Conference of the University of Waterloo Centre for the New Oxford English Dictionary*, Oxford, UK, pp. 1–22.

Hindle, D. and M. Rooth (1993). Structural ambiguity and lexical relations. *Computational Linguistics 19*(1), 103–120.

Hirst, G. (1987). *Semantic Interpretation and the Resolution of Ambiguity*. Cambridge University Press.

Hirst, G. (1988). Resolving lexical ambiguity computationally with spreading activation and polaroid words. In G. C. S. Small and M. Tennenhaus (Eds.), *Lexical Ambiguity Resolution*, Chapter 3, pp. 73–107. Morgan Kaufmann Press, Palo Alto, California.

Hirst, G. and D. St-Onge (1998). Lexical chains as representation of context for the detection and correction of malapropisms. In Fellbaum (1998), pp. 305–332. MIT Press.

Hudson, R. A. (1984, November). *Word Grammar*. Oxford, UK: Blackwell Publishers Limited.

Hutchins, J. (1995, July). Reflections on the history and present state of machine translation. In *MT Summit V proceedings*, Luxembourg, pp. 89–96.

Hutchins, J. (1996, June). Alpac: The (in)famous report. In *MT News International 14*, pp. 9–12. Reprinted in: Readings in machine translation, ed. Sergei Nirenburg, Harold Somers, and Yorick Wilks (Cambridge, Mass.: The MIT Press, 2003), pp. 131-135.

Hutchins, J. (1997a). From first conception to first demonstration: The nascent years of machine translation, 1947-1954. a chronology. *Machine Translation 12*(3), 195–252.

Hutchins, J. (1997b, July 23-25). Looking back to 1952: The first MT conference. In *Proceedings of the 7th International Conference on Theoretical and Methodological Issues in Machine Translation*, pp. 19–30.

Ide, N. M. and J. Véronis (1998, March). Introduction to the special issue on word sense disambiguation: The state of the art. *Computational Linguistics 24*(1), 1–40.

James, W. (1890/1950). *The Principles of Psychology*. New York, Dover Publications, Inc.

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

Jarmasz, M. and S. Szpakowicz (2001b). Roget's thesaurus: A lexical resource to treasure. Technical Report TR-2001-01, University of Ottawa, Ottawa.

Jarmasz, M. and S. Szpakowicz (2003). Roget's thesaurus and semantic similarity. Technical Report TR-2003-01, University of Ottawa.

Jastrezembski, J. E. (1981). Multiple meanings, number of related meanings, frequency of occurrence, and the lexicon. In *Cognitive Psychology* (13 ed.)., pp. 278–305.

Jastrezembski, J. E. and R. F. Stanners (1975). Multiple word meanings and lexical search speed. *Journal of Verbal Learning and Verbal Behaviour 14*, 534–537.

Jensen, K. and J.-L. Binot (1987). Disambiguating prepositional phrase attachments by using on-line dictionary definitions. *Computational Linguistics 13*(3-4), 251–260.

Jiang, J. J. and D. W. Conrath (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceeding of International Conference on Research in Computational Linguistics (ROCLING X)*, Taiwan.

Kamp, H. (1981). A theory of truth and semantic representation. In J. Groenendijk, T. Janssen, and M. Stokhof (Eds.), *Formal Methods in the Study of Language*, Number 135, pp. 277–322. Amsterdam: Mathematical Centre.

Kaplan, A. (1955). An experimental study of ambiguity and context. *Mechanical Translation 2*, 39–46. Originally Mimeographed in 1950.

Karov, Y. and S. Edelman (1996, August). Learning similarity-based word sense disambiguation from sparse data. In *Fourth Workshop on Very Large Corpora*, pp. 42–55.

Kelly, E. F. and P. J. Stone (1975). *Computer Recognition of English Word Senses* (3 ed.). North Holland Linguistics series. North-Holland Publishing Company, Amsterdam.

Kilgarriff, A. (1997, December). What is word sense disambiguation good for? In *Proceedings of the Natural Language Processing Pacific Rim Symposium*, Phuket, Thailand, pp. 209–214.

Kilgarriff, A. (1998a). Gold standard datasets for evaluating word sense disambiguation programs. *Computer Speech and Language 12*(3), 453–472.

Kilgarriff, A. (1998b, May). Senseval: An exercise in evaluating word sense disambiguation programs. In *Proceedings of the First International Conference on Language Resources & Evaluation (LREC'98)*, Granada, Spain, pp. 581–588.

Kilgarriff, A. and J. Rosenzweig (2000a). English senseval: Report and results. In *Proceedings of the 2^{nd} International Conference on Language Resources & Evaluation (LREC'2000)*, Volume 3, Athens, Greece, pp. 1239–1244.

Kilgarriff, A. and J. Rosenzweig (2000b). Framework and results for English senseval. Technical Report ITRI-00-20, Information Technology Research Institute, University of Brighton. Also published in Computers and the Humanities 34 (1-2), Special Issue on SENSEVAL, pp 15-48.

Kirkpatrick, B. (1998, 6 August). *Roget's Thesaurus of English Words and Phrases*. Penguin Books.

Klavans, J. L., M. S. Chodorow, and N. Wacholder (1990). From dictionary to knowledge base via taxonomy. In *Proceedings of the 6^{th} Conference of the UW Centre for the New OED*, Waterloo, Canada, pp. 110–132.

Klein, D., K. Toutanova, H. T. Ilhan, S. D. Kamvar, and C. D. Manning (2002). Combining heterogeneous classifiers for word-sense disambiguation. In *Proceedings of the Workshop on Word Sense Disambiguation: Recent Successes and Future Directions (ACL'02)*, Philadelphia, PA, pp. 74–80.

Knight, K. and S. K. Luk (1994, July). Building a large knowledge base for machine translation. In *Proceedings of the 12^{th} National Conference of the American Association of Artificial Intelligence Conference (AAAI-94)*, Seattle, WA, pp. 773–778.

Koutsoudas, A. K. and R. Korfhage (1955). Machine translation and the problem of multiple meaning. *Mechanical Translation 2*, 46–51.

Kozima, H. and T. Furugori (1993). Similarity between words computed by spreading activation on an English dictionary. In *Proceedings of 6th Conference of the European Chapter of the Association for Computational Linguistics (EACL93)*, Utrecht, pp. 232–239.

Krovetz, R. and W. B. Croft (1989). Word sense disambiguation using machine-readable dictionaries. In *Proceedings of the 12th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR'89)*, Cambridge, MA, pp. 127–136.

Kucera, H. and W. N. Francis (1967). *Computational Analysis of Present-Day American English*. Brown University Press, Providence, Rhode Island.

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

Landes, S., C. Leacock, and R. I. Tengi (1998). Building semantic concordances. In Fellbaum (1998), pp. 199–216. MIT Press.

Leacock, C. and M. S. Chodorow (1998). Combining local context and WordNet similarity for word sense identification. In Fellbaum (1998), pp. 265–283. MIT Press.

Lee, J. H., M. H. Kim, and Y. J. Lee (1993, June). Information retrieval based on conceptual distance in isa hierarchies. *Journal of Documentation 49*(2), 188–207.

Lesk, M. (1986, June). Automatic sense disambiguation using machine-readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 1986 SIGDOC Conference*, Toronto, Canada, pp. 24–26.

Levin, B. (1993). *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press.

Levow, G.-A. (1997, 1,). Corpus-based techniques for word sense disambiguation. Technical Report AIM-1637, MIT AI Lab.

Li, X., S. Szpakowicz, and S. Matwin (1995). A WordNet-based algorithm for word sense disambiguation. In *Proceedings of IJCAI*, Montréal, Canada, pp. 1368–1374.

Lin, D. (1997, July). Using syntactic dependency as local context to resolve word sense ambiguity. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL'97)*, Madrid, Spain, pp. 64–71.

Lin, D. (1998a, August). Automatic retrieval and clustering of similar words. In *COLING-ACL98*, Montréal, Canada, pp. 768–774.

Lin, D. (1998b). Extracting collocations from text corpora. In *Workshop on Computational Terminology*, Montréal, Canada.

Lin, D. (1998c). An information-theoretic definition of similarity. In *Proceedings 15th International Conference on Machine Learning*, pp. 296–304. Morgan Kaufmann, San Francisco, CA.

Litkowski, K. C. (1997, April). Desiderata for tagging with WordNet synsets or mcca categories. In *Proceedings of the ACL-SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, Washington, D.C., pp. 12–17.

Marcus, M., G. Kim, M. A. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger (1994). The Penn Treebank: Annotating predicate argument structure. In *Proceedings of the ARPA Human Language Technology Workshop, (ARPA'94)*. also known as Treebank II.

Marcus, M. P., B. Santorini, and M. A. Marchinkiewicz (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics 19*(2), 313–330.

Markowitz, J., T. Ahlswede, and M. Evens (1986). Semantically significant patterns in dictionary definitions. In *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics (ACL)*, New York, pp. 112–119.

Martinez, D. and E. Agirre (2000). One sense per collocation and genre/topic variations. In *Proceedings of the Joint SIGDA T Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, Hong Kong.

Masterman, M. (1957). The thesaurus in syntax and semantics. *Mechanical Translation 4*(1-2), 35–43.

Masterman, M. (1961). Semantic message detection for machine translation, using an interlingua. In *International Conference on Machine Translation of Languages and Applied Language Analysis*, Her Majesty's Stationary Office, London, pp. 438–475.

McCelland, J. L. and D. E. Rumelhart (1981). An interactive activation of context effects in letter perception: Part 1. an account of basic findings. *Psychological Review 88*(6), 375–407.

McRoy, S. W. (1992). Using multiple knowledge sources for word sense discrimination. *Computational Linguistics 18*(1), 1–30.

Mel'cuk, I. A. (1988, February). *Dependency Syntax: Theory and Practice*. State University of New York Pr.

MeSH (1995). *MeSH - Tree Structures*. National Library of Medicine, http://www.nlm.nih.gov/mesh/meshhome.html.

Meyer, D. E. and R. W. Schvaneveldt (1971). Facilitation in recognizing pairs of words: Evidence of a dependence between retrieval operations. *Journal of Experimental Psychology 90*(2), 227–234.

Michiels, A., J. Mullenders, and J. Noël (1980). Exploiting a large database by longman. In *Proceedings of the 8^{th} International Conference on Computational Linguistics (COLING-ACL'80)*, Tokyo, Japan, pp. 374–382.

Mihalcea, R. and D. I. Moldovan (1998, August). Word sense disambiguation based on semantic density. In *Usage of WordNet in Natural Language Processing Systems (Coling-ACL'98)*, Montréal, Canada.

Mihalcea, R. and D. I. Moldovan (1999, June). A method for word sense disambiguation of unrestricted text. In *Proceedings of the 37^{th} Annual Meeting of the Association for Computational Linguistics(ACL'99)*, Maryland, NY, pp. 152–158.

Mihalcea, R. and D. I. Moldovan (2000, May). An iterative approach to word sense disambiguation. In *Proceedings of FLAIRS-2000*, Orlando, Florida, pp. 219–223.

Miller, G. A., R. Beckwith, C. D. Fellbaum, D. Gross, and K. J. Miller (1990). Five papers on WordNet. In *International Journal of Lexicography*, pp. 235–312.

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

Miller, G. A., M. S. Chodorow, S. Landes, C. Leacock, and R. G. Thomas (1994, March). Using a semantic concordance for sense identification. In *Proceedings of the ARPA Human Language Technology Workshop*, San Francisco, pp. 240–243.

Miller, G. A., C. Leacock, R. I. Tengi, and R. Bunker (1993, March). A semantic concordance. In *Proceedings of the ARPA Human Language Technology Workshop*, Princeton, New Jersey, pp. 303–308. Morgan Kaufmann.

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

Nakamura, J.-I. and M. Nagao (1988, August). Extraction of semantic information from an ordinary English dictionary and its evaluation. In *Proceedings of the 12^{th} International Conference on Computational Linguistics (COLING-ACL'88)*, Budapest, Hungary, pp. 459–464.

Ng, H. T. (1997, August). Exemplar-based word sense disambiguation: Some recent improvements. In *Proceedings of the Second Conference on Empirical Methods in NLP (EMNLP-2)*, Providence, RI., pp. 208–213

Ng, H. T. and H. B. Lee (1996, June). Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, Santa Cruz, California, USA, pp. 40–47. Morgan Kaufmann.

Okumura, M. and T. Honda (1994). Word sense disambiguation and text segmentation based on lexical cohesion. In *In Proceedings of the Fifteen Conference on Computational Linguistics (COLING-94)*, Volume 2, pp. 755–761.

Oswald-Jr., V. A. (1952, June 17-20). Microsemantics. In *The First M.I.T. Conference on Mechanical Translation*. Originally Mimeographed, and available on microfilm at M.I.T., Papers on Mechanical Translation, roll 799.

Patrick, A. B. (1985). An exploration of abstract thesaurus instantiation. Master's thesis, University of Kansas, Lawrence, KS.

Pedersen, T. and R. Bruce (1997, August). Distinguishing word senses in untagged text. In *Proceedings of the Second Conference on Empirical Methods in NLP (EMNLP-2)*, Providence, RI., pp. 197–207

Pietra, S. A. D., V. J. D. Pietra, and J. D. Lafferty (1995, May). Inducing features of random fields. Technical Report CMU-CS-95-144, Carnegie Mellon University.

Pietra, S. A. D., V. J. D. Pietra, and J. D. Lafferty (1997). Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence 19*(4), 380–393.

Platt, J. C., N. Cristianini, and J. Shawe-Taylor (2000). Large margin dags for multiclass classification. *Advances in Neural Information Processing Systems 12*, 547–553.

Preiss, J. (2001). Local versus global context for WSD of nouns. In *Proceedings of Computational Linguistics UK (CLUK-4)*, pp. 1–8.

Procter, P. (1978). *Longman Dictionary of Contemporary English (LDOCE)*. Longman Group Limited: Harlow, Essex, UK.

Quillian, M. R. (1961). A design for an understanding machine. King's College, Cambridge University, Cambridge, Presented at the Semantic Problems in Natural Language Colloquium.

Quillian, M. R. (1962a, July). A revised design for an understanding machine. *Mechanical Translation 7*(1), 17–29.

Quillian, M. R. (1962b). A semantic coding technique for mechanical English paraphrasing. Internal Memorandum of the Mechanical Translation Group, Research Laboratory of Electronics, MIT.

Quillian, M. R. (1967). Word concepts: A theory and simulation of some basic semantic capabilities. In *Behavioural Science*, Volume 12, pp. 410–430. Reprinted in R. J. Brachman & H. J. Levesque "Readings in Knowledge Representation" (1985), Kaufmann, pages 98-118.

Quillian, M. R. (1968). Semantic memory. In M. L. Minsky (Ed.), *Semantic Information Processing*, pp. 216–270. MIT Press, Cambridge, MA. Reprinted in Collins & Smith (eds.), Readings in Cognitive Science, section 2.1.

Quillian, M. R. (1969, August). The teachable language comprehender: A simulation program and theory of language. *Communication of the ACM 12*(8), 459–476.

Rada, R. and E. Bicknell (1989). Ranking documents with a thesaurus. *Journal of the American Society for Information Science (JASIS) 40*(5), 304–310.

Rada, R., H. Mili, E. Bicknell, and M. Blettner (1989). Development and application of a metric on semantic nets. In *IEEE Transactions on Systems, Man and Cybernetics*, Volume 19, pp. 17–30.

Ratnaparkhi, A. (1996, May). A maximum entropy part-of-speech tagger. In *Proceedings of the Empirical Methods in Natural Language Processing Conference (EMNLP-96)*, University of Pennsylvania, pp. 133–142.

Ratnaparkhi, A. (1997, May). A simple introduction to maximum entropy models for natural language processing. Technical Report 97-08, Institute for Research in Cognitive Science, University of Pennsylvania.

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

Reifler, E. (1955). The mechanical determination of meaning. In W. N. Locke and A. D. Booth (Eds.), *Machine Translation of Languages: Fourteen Essays*, pp. 136–164. The Technological Press of the Massachusetts Institute of Technology/Wiley & Sons, New York/Clapham & Hall, London.

Resnik, P. S. (1993, December). *Selection and Information: A Class-Based Approach to Lexical Relationships*. Ph. D. thesis, NSF Science and Technology Center for Research in Cognitive Science, University of Pennsylvania, Philadelphia.

Resnik, P. S. (1995a, June). Disambiguating noun groupings with respect to WordNet senses. In *Third Workshop on Very Large Corpora. Association for Computational Linguistics*, MIT, USA.

Resnik, P. S. (1995b). Using information content to evaluate semantic similarity in a taxonomy. In *IJCAI*, pp. 448–453.

Resnik, P. S. (1996, November). Selectional constraints: An information-theoretic model and its computational realization. *Cognition 61*(1-2), 127–159.

Resnik, P. S. (1997, April). Selectional preference and sense disambiguation. In *Proceedings of the ACL-SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, Washington, D.C., pp. 127–159.

Resnik, P. S. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research (JAIR) 11*, 95–130.

Resnik, P. S. and M. Diab (2000). Measuring verb similarity. Technical Report LAMP-TR-047, Language and Media Processing Laboratory, UMIACS, University of Maryland.

Resnik, P. S. and D. Yarowsky (1997, April). A perspective on word sense disambiguation methods and their evaluation. In *Proceedings of the ACL-SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, Washington, DC, pp. 79–86.

Richardson, R. and A. F. Smeaton (1995). Using WordNet in a knowledge-based approach to information retrieval. Technical Report CA-0395, School of Computing, Dublin City University, Dublin, Ireland.

Richardson, R., A. F. Smeaton, and J. Murphy (1994). Using WordNet as a knowledge base for measuring semantic similarity between words. Technical Report CA-1294, School of Computing, Dublin City University, Dublin, Ireland.

Richens, R. H. (1958). Interlingual machine translation. *Computer Journal 1*(3), 144–147.

Rigau, G., J. Atserias, and E. Agirre (1997). Combining unsupervised lexical knowledge methods for word sense disambiguation. In *Proceedings of the 35^{th} Annual Conference of the Association for Computational Linguistics and the 8^{th} Conference of the European Association for Computational Linguistics (ACL '97 and EACL '97)*, Madrid, Spain.

Ross, S. M. (1976). *A First Course in Probability*. Macmillan.

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

Salton, G. and C. Buckley (1989). A comparison between statistically and syntactically generated term phrases. Technical Report TR 89-1027, Department of Computing Science, Cornell University, Ithaea, NY 14853-7501.

Salton, G. and M. J. McGill (1983). *Introduction to Modern Information Retrieval: The SMART and SIRE experimental retrieval systems*. McGrawHill, New York.

Sanderson, M. (1994, July). Word sense disambiguation and information retrieval. In *Proceedings of SIGIR-94, 17^{th} ACM International Conference on Research and Development in Information Retrieval*, Dublin, Ireland, pp. 142–151.

Sanderson, M. (1996). Word sense disambiguation and information retrieval. Technical Report TR-1997-7, Department of Computing Science at the University of Glasgow, Glasgow G12 8QQ, UK.

Schütze, H. and J. O. Pedersen (1995). Information retrieval based on word senses. In *Proceedings of the Symposium on Document Analysis and Information Retrieval*, Volume 4, Las Vegas, NV, pp. 161-175.

Shuppan, S. (1964). *Bunrui Goi Hyo*. National Language Research Institute.

Smeaton, A. F. and I. Quigley (1996). Experiments on using semantic distances between words in image caption retrieval. In *Research and Development in Information Retrieval*, pp. 174–180.

Spark Jones, K. (1986). *Synonymy and Semantic Classification*. Edinburgh University Press, Distributed by the Columbia University Press, NY.

St-Onge, D. (1995). Detecting and correcting malapropisms with lexical chains. Master's thesis, Department of Computer Science, University of Toronto. Also published as Technical Report CSRI-319.

Stetina, J., S. Kurohashi, and M. Nagao (1998, August). General word sense disambiguation method based on a full sentential context. In *Usage of WordNet in Natural Language Processing Systems (Coling-ACL '98)*, Montréal, Canada.

Stevenson, M. and Y. A. Wilks (1999). Combining weak knowledge sources for sense disambiguation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'99)*, Stockholm, Sweden, pp. 884–889.

Stevenson, M. and Y. A. Wilks (2000). Large vocabulary word sense disambiguation. In Y. Ravin and C. Leacock (Eds.), *Polysemy: Computational and Theoretical Contributions*, Chapter 9. Oxford University Press.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society B 36*(1), 111–147.

Suárez, A. and M. Palomar (1993, March). A maximum entropy-based word sense disambiguation system. In *Proceedings of the 19^{th} International Conference on Computational Linguistics (COLING 2002)*, Volume 2, pp. 960–966.

Suárez, A. and M. Palomar (2002). Feature selection analysis for maximum entropy-based WSD. In *Computational Linguistics and Intelligent Text Processing, Third International Conference (CICLing 2002)*, Volume 2276 of *Lecture Notes in Computer Science*, pp. 146–155. Springer.

Sussna, M. J. (1993, November). Word sense disambiguation for free-text indexing using a massive semantic network. In *Proceedings of the Second International Conference on Information and Knowledge Management (CIKM 93)*, Washington, DC, USA, pp. 67–74.

Sussna, M. J. (1997). *Text Retrieval Using Inference in Semantic Metanetworks*. Ph. D. thesis. University of California, San Diego.

Sutcliffe, R. F. E. and B. E. A. Slater (1995). Disambiguation by association as a practical method: Experiments and findings. *Journal of Quantative Linguistics 2*(1), 43–52.

Temperley, D. (1999). An introduction to the link grammar parser. Carnegie Mellon University. http://hyper.link.cs.cmu.edu/link/dict/introduction.html

Tengi, R. I. (1998). Design and implementation of the WordNet lexical database and searching software. In Fellbaum (1998), pp. 105–128. MIT Press.

Towell, G. and E. M. Voorhees (1998, March). Disambiguating highly ambiguous words. *Computational Linguistics 24*(1), 125–145.

Turcato, D., P. McFetridge, F. Popowich, and J. Toole (1999). A unified example-based and lexicalist approach to machine translation. In *The 8th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-99)*, pp. 33–43.

Tversky, A. (1977). Features of similarity. *Psychological Review 84*, 327–352.

Veale, T., A. Conway, and B. Collins (1998). The challenges of cross-modal translation: English-to-Sign-Language translation in the zardoz system. In *Machine Translation 13*, pp. 81–106. Kluwer Academic.

Véronis, J., V. Houitte, and C. Jean (1998, April). Methodology for the construction of test material for the evaluation of word sense disambiguation systems. In *2^{nd} Workshop on Lexical Semantics Systems (WLSS98)*, Pisa, Italy.

Véronis, J. and N. M. Ide (1990, August). Word sense disambiguation with very large neural networks extracted from machine readable dictionaries. In *Proceedings of the 13^{th} International Conference on Computational Linguistics (Coling'90)*, Volume 2, Helsinki, Finland, pp. 389–394.

Véronis, J. and N. M. Ide (1991). An assessment of semantic information automatically extracted from machine readable dictionaries. In *Proceedings of the Fifth Conference of the European Chapter of the Association for Computational Linguistics*, Berlin, Germany, pp. 227–232.

Véronis, J. and N. M. Ide (1995). Large neural networks for the resolution of lexical ambiguity. In P. Saint-Dizier and E. Viegas (Eds.), *Computational Lexical Semantics*, pp. 251–269. Natural Language Processing Series, Cambridge University Press, Cambridge, UK.

Voorhees, E. M. (1993). Using WordNet to disambiguate word senses for text retrieval. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, 16, Pittsburgh, Pennsylvania, pp. 171-180.

Vossen, P. (1997, March). EuroWordNet: A multilingual database for information retrieval. In *Proceedings of the DELOS workshop on Cross-language Information Retrieval*, Zurich.

Weaver, W. (1949). Translation. Reprinted in *Machine Translation of Languages: Fourteen Essays*, Pages 15-23, Edited by William N. Locke and A. Donald Booth, Publisher The Technological Press of the Massachusetts Institute of Technology/Wiley & Sons, New York/Clapham & Hall, London, 1955.

Wiebe, J., J. Maples, L. Duan, and R. Bruce (1997, April). Experience in WordNet sense tagging in the wall street journal. In *Proceedings of the ACL-SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, Washington, D.C., pp. 8–11.

Weiss, S. F. (1973). Learning to disambiguate. *Information Storage and Retrieval 9*(1), 33–41.

West, M. P. (1953). A general service list of English words, with semantic frequencies and a supplementary word-list for the writing of popular science and technology.

Whittemore, G., K. Ferrara, and H. Brunner (1990, June). Empirical studies of predictive powers of simple attachment schemes for post-modifier prepositional phrases. In *Proceedings of the 28th Annual Meeting of Association for Computational Linguistics*, Pittsburgh, Pennsylvania, pp. 23–30.

Wilks, Y. A. (1968). On-line semantic analysis of English texts. *Mechanical Translation 11*(3-4), 59–72.

Wilks, Y. A. (1969). Getting meaning into the machine. In *New Society*, Volume 361, pp. 315–317.

Wilks, Y. A. (1972). *Grammar, Meaning and the Machine Analysis of Language*. Routledge, London.

Wilks, Y. A. (1973). An artificial intelligence approach to machine translation. In R. Schank and K. Colby (Eds.), *Computer Models of Thought and Language*, pp. 114–151. W. H. Freeman, San Francisco.

Wilks, Y. A. (1975a, May). An intelligent analyzer and understander of English. *Communications of the ACM 18*(5), 264–274.

Wilks, Y. A. (1975b). Preference semantics. In E. L. K. III (Ed.), *Formal Semantics of Natural Language*, pp. 329–348. Cambridge University Press.

Wilks, Y. A. (1975c, Spring). A preferential, pattern-seeking, semantics for natural language inference. *Artificial Intelligence 6*(1), 53–74.

Wilks, Y. A. (1975d). Primitives and words. In *Proceedings of the Interdisciplinary Workshop on Theoretical Issues in Natural Language Processing*, Cambridge, MA, pp. 38–41.

Wilks, Y. A. and D. Fass (1990). Preference semantics: A family history. Technical Report MCCS-90-194, Computing Research Laboratory, New Mexico State University, Las Cruces, NM.

Wilks, Y. A., D. C. Fass, C.-M. Guo, J. E. MacDonald, T. Plate, and B. M. Slator (1990). Providing machine tractable dictionary tools. *Machine Translation 5*(2), 99–155. Also in James Pustejovsky (Editor) Semantics and the Lexicon, MIT Press, Cambridge, MA, 1990.

Wilks, Y. A. and M. Stevenson (1996). The grammar of sense: Is word-sense tagging much more than part-of-speech tagging? Technical Report CS-96-05, Sheffield University, UK.

Wilks, Y. A. and M. Stevenson (1997a). Combining independent knowledge sources for word sense disambiguation. In *Proceedings of the 2^{nd} Conference on Recent Advances in Natural Language Processing (RANLP'97)*, Tzigov Chark, Bulgaria.

Wilks, Y. A. and M. Stevenson (1997b). Implementing a sense tagger within a general architecture for text engineering. In *Proceedings of New Methods in Language Processing (NeMLaP-3)*.

Wilks, Y. A. and M. Stevenson (1997c, April). Sense tagging: Semantic tagging with a lexicon. In *Proceedings of the ACL-SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, Washington, D.C., pp. 74–78.

Wilks, Y. A. and M. Stevenson (1998a, October). The grammar of sense: Using part-of-speech tags as a first step in semantic disambiguation. *Journal of Natural Language Engineering 4*(3), 135–144.

Wilks, Y. A. and M. Stevenson (1998b, August). Word sense disambiguation using optimised combinations of knowledge sources. In *Proceedings of 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL'98)*, Volume 2(36), Montréal, Canada, pp. 1398–1402.

Wilks, Y. A. and M. Stevenson (1998c, August). Word sense disambiguation using optimised combinations of knowledge sources. In *Proceedings of 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL'98)*, Volume 2(36), Montréal, Canada, pp. 1398–1402.

Wu, Z. and M. Palmer (1994). Verb semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics*, New Mexico State University, Las Cruces, New Mexico, pp. 133–138.

Yarowsky, D. (1992, August). Word sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING'92)*, Nantes, France, pp. 454–460.

Yarowsky, D. (1993). One sense per collocation. In *Proceedings of the ARPA Human Language Technology Workshop*, Princeton, NJ, pp. 266–271.

Yarowsky, D. (1995). Unsupervised word-sense disambiguation rivalling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL-95)*, Cambridge, MA, pp. 189–196.

Yarowsky, D. (2000). Hierarchical decision lists for word sense disambiguation. *Computers and the Humanities 34*(2), 179–186.

Zipf, G. K. (1945). The meaning-frequency relationship of words. *Journal of General Psychology 33*, 251–256.