From E-Language to I-Language:

Foundations of a Pre-Processor for the Construction Integration Model.

Christopher Mark Powell

Submitted in partial fulfilment of the requirements of Oxford Brookes University for the degree of Doctor of Philosophy

February 2005

Abstract

This thesis is concerned with the 'missing process' of the Construction Integration Model (CIM - a model of Discourse Comprehension), namely the process that converts text into the logical representation required by that model and which was described only as a requirement by its authors, who expected that, in the fullness of time, suitable grammar parsers would become available to meet this requirement. The implication of this is that the conversion process is distinct from the comprehension process. This thesis does not agree with this position, proposing instead that the processes of the CIM have an active role in the conversion of text to a logical representation.

In order to investigate this hypothesis, a pre-processor for the CIM is required, and much of this thesis is concerned with selection and evaluation of its constituent elements. The elements are: a Chunker that outputs all possible single words and compound words expressed in a text; a Categorial Grammar (CG) parser modified to allow compounds and their constituent words to coexist in the chart; classes from abridged WordNet noun and verb taxonomies comprising only the most informative classes; revised handling of CG syntactic categories to take account of structural inheritance, thereby permitting incremental interpretation, and finally extended CG semantic categories that allow sense lists to be attached to each instantiated semantic variable.

In order to test the hypothesis, the elements are used to process a Garden Path sentence for which human parsing behaviour is known. The parse is shown to build interpretation incrementally, to appropriately sense-tag the words, derive the correct logical

representation and behave in a manner consistent with expectations. Importantly, the determination of coherence between proposed sense assignments of words and a knowledge base, a function of the CIM, is shown to play a part in the parse of the sentence. This provides evidence to support the hypothesis that the CIM and the pre-processor are not distinct processes.

The title of this thesis, 'From E-Language to I-Language: Foundations of a Pre-Processor for the Construction Integration Model', is intended to circumscribe the work contained herein. Firstly, the reference to Chomsky's notions of E-Language (External(ised) Language) and I-language (Internal(ised) Language) make clear that we acknowledge these two aspects of language. Chomsky maintains that E-Language, such as English, German, and Korean, are mere 'epiphenomena', a body of knowledge or behavioural habits shared by a community, and as such are not suitable subjects for scientific study. I-Language, argues Chomsky, is a 'mental object', is biologically/genetically specified, equates to language itself and so is a suitable object of study. We shall not pursue the philosophical arguments and counter-arguments concerning E-Language and I-Language (but see for example [DUMM86], [CHOM96]), but shall use the notions of E-Language and I-Language to differentiate between the natural language text to be processed, which can be unique to a community, geographical and/or temporal location, or to some extent to an individual, and the internal, structured, world-consistent representation of that text, and the cognitive processes involved in the representation creation, which being 'genetically specified' can be assumed common to all humans. This thesis is therefore concerned with the interface between these two aspects of language, and specifically in how the internal

cognitive processes of I-Language, outlined in theories such as the Construction-Integration Model, interact with external representations of language in order to construct internal representative models of that E-Language.

Secondly, ‘Foundations’ indicates that this work does not deliver a fully functioning natural language processing system, but draws together ‘distinct’ linguistic research threads (e.g. Chunking, Word-Sense Disambiguation, Grammar Parsing, and theories of grammar acquisition), to describe the process of converting a natural language text into a logically structured and plausibly sense-tagged representation of that text. As such, this thesis is a ‘proof of concept’, and must be followed by future evaluative work.

Acknowledgements

Firstly, I would like to thank my first supervisor, Mary Zajicek, and second supervisor, David Duce, for keeping me on the straight and narrow, for the encouragement they gave, and for making me believe that I would actually cross the finish line. I am most grateful for their efforts in proofreading the thesis and the helpful feedback they provided - my submission deadline was approaching fast and they pulled out all the stops to make it happen. I am also indebted to Mary for the many opportunities my association with her have presented, for the interesting projects and foreign travel I have enjoyed, and for her continued support and promotion.

I must also thank my examiners, Mary McGee Wood and Faye Mitchell, for an enjoyable viva and for their constructive comments and enthusiasm both during and after.

I owe thanks to Marilyn Deegan for inviting me to 'The Use of Computational Linguistics in the Extraction of Keyword Information from Digital Library Content' workshop, Kings College London, Feb. 2004. Preparation for the workshop gave me a vital push at just the right moment and led to a consolidation of my work on Specialisation Classes. I would also like to thank Dawn Archer and Tony McEnery of Lancaster University for their useful and encouraging comments during the workshop.

Acknowledgements

My fellow research students, Alvin Chua, Jianrong “ten pints” Chen, Samia Kamal, Sue Davies, Tjeerd olde-Scheper and Nick Hollinworth contributed hugely to an enjoyable and rewarding time in the Intelligent Systems Research Group. They provided useful insights from the perspectives of their own research fields, and shoulders to cry on when the going got tough. A big thanks to my good friend Tjeerd who is always happy to play Scully to my Mulder, and whose knowledge of Chaotic Computation is second only to his knowledge of the finest single malts. Our anticipated research trip to Islay will be most interesting.

Thanks are due to Ken Brownsey, chair of the East Oxford Logic Group, who once taught me inspirational and useful things like LISP and Functional Programming. His jokes baffle some and delight others.

Writing up was a very solitary and sedentary experience, as was the design and implementation of the software developed during the course of this work. However, I was helped during these times by two special chums - a big thanks to Daisy for taking me on daily walks to ensure I got fresh air in my lungs and the sun on my face, and to Splodge who slept on my lap and kept it warm whilst I worked at the computer.

Finally I thank Lindsay for putting up with me through my times of elation, depression, absence, and presence. Without her love and support I would never have been able to complete this work, and I shall be eternally grateful to her. She’s embarking on her own research degree next year, so it is my turn to be tested in the supporting role.

Table of Contents

1 Introduction

As every user of web search-engines knows, plenty of chaff is returned with the wheat. A sighted user can quickly form value judgements as to the relevancy of each returned page by opening and visually scanning them – essentially having a quick look at the headings, images, text, and matching them against their own line of enquiry. A visually impaired user does not have this luxury of random access via the visual mode, instead relying instead on modal transducers such as refreshable Braille displays and synthesised speech to render the selected document's text, both of which present the document text serially from beginning to end.

BrookesTalk, a web browser for the blind and visually impaired developed at Oxford Brookes University, addresses these orientation issues by including a term-frequency based keyword and extract generator which provides the user with approximately ten keywords from the current page, allowing them to use their own cognitive abilities to rapidly identify a context into which the presented keywords fit, hopefully suggesting the general topic of the document without the need to listen to the speech-rendered document in its entirety. The extract served the same purpose, providing more sentential context for the keywords.

Although the term-frequency based summariser produced agreeable results for single domain documents, such as journal articles, problems arise when attempting to summarise mixed-topic documents such as online newspapers, a confusing mix of keywords and sentences extracted from each topic being presented. This and the plethora of different document genres available on the web led to the decision to look for an alternative summarising technology to term-frequency for use in BrookesTalk,

resulting in this research work. However, because of the complexity of natural language processing systems, we took a step back from summarisation, concentrating on an underdeveloped element of the Construction Integration Model (CIM), a model of discourse comprehension that we have selected as the basis of a future summarisation system because of its robust and human-like processes.

The underdeveloped process that became the focus of this thesis is that which converts text into the logical representation required by the CIM and which was described only as a requirement by its authors, who expected that, in the fullness of time, suitable grammar parsers would become available to meet this requirement. This implies that the conversion process is distinct from the comprehension process. This thesis does not agree with that position, proposing instead that the processes of the CIM have an active role in the conversion of text to a logical representation on the grounds that sense-based coherence is common to both, as is shown in Section 4.1.1.

This question is important as it has implications for grammar parsing and word sense disambiguation in general; if the hypothesis is true, then grammar and sense are linked, and a successful grammar parser will have to take account of word sense. Similarly, a word sense disambiguation algorithm will have to take into consideration the plausible grammatical contexts of the words it is attempting to sense-tag.

1.1 Structure of thesis

The thesis consists of two parts consisting of chapters 2 to 4 and 5 to 9. The first part looks at automatic text summarisation and selects psychological and cognitive over statistical methods as they are involved in the only working language comprehension system available for study, i.e. the Human language facility, and therefore can

reasonably be expected to contribute to artificial language comprehension systems exhibiting qualities comparable to our own. Due to the complexity of performing full discourse comprehension, the thesis focuses on the early stages which are often glossed-over by cognitive models of discourse comprehension such as the Construction Integration Model.

The second part recognises the huge grammatical, linguistic and world knowledge requirements of the initial stages of a discourse comprehension system, along with the processing effort needed to use utilise them effectively. It addresses this by identifying areas in which efficiencies can be made, and in doing so shows that further consistency with the human language processor, in the form of incremental processing and handling of a class of sentence known as Garden Path sentences, is possible.

Chapter 2 reviews summarisation techniques, grouping them into General, Psychological, AI and Current approaches. It also reviews summaries as cohesive text, and looks at summarisers designed specifically for use with the web. In selecting an approach for use in a future BrookesTalk, easily implemented surface-statistical and positional systems are compared to human-like but complex and resource-hungry psychological and cognitive techniques. As the statistical/positional methods are not compatible with all document types (e.g. stories, short texts) and cannot account for many linguistic phenomena or incorporate grammatical relations or word sense into their calculations, the more complex, human-like methods based on psychological and cognitive models of human language comprehension are selected as the basis for further study.

Chapter 3 takes the view that effective summarisation is only possible through comprehension of the original text and consequently *discourse comprehension* should

be the initial step in summary production. The Construction Integration Model (CIM) is selected as a suitable model as it is consistent with psychological observations, and recognises that summarisation is a necessary step in discourse comprehension. Supporting evidence is presented, along with an overview of the CIM itself in which it is noted that the model requires a pre-processing step in which text is converted into a logical, sense-tagged representation.

Chapter 4 looks at the requirements of the pre-processor in more detail, and focuses on the three main elements: logical form transformation, sense, and coherence. A cognitively viable grammatical parser, identified as one that is consistent with current theories of grammar acquisition (The Coalition Model) is proposed to complement the psychologically oriented CIM; Categorial Grammar (CG) is chosen for this component for these reasons, and for its impressive abilities in handling a wide variety of grammatical phenomena, as well as its consistency with current models of grammar acquisition.

Chapter 5 recognises that the nature of the chart-parsing algorithm leads to high processing loads when processing longer sentences. An inability of the chart parser to build compounds from individual terms is also recognised, and both of these factors are used to justify the use of a Chunker to handle this inability. Chunking itself is justified in terms of the human visual system, theories of word recognition, and the processing of Garden Path sentences. As the proposed Chunker outputs individual words as well as compounds and their constituent words, allowing the grammar parser/coherence mechanism to select the grouping (if any) that is most plausible in terms of the current parse, the chart parser is extended to allow 'Parallel Shifts' (introduced in Section 5.2.3)

of both compounds and their constituent terms. The parallel-shifting chart parser is evaluated on a Garden Path sentence, the results being consistent with expectations.

Chapter 6 focuses on the sense indicators that are needed to enable plausibility testing of propositions generated by the grammar parser. Recognising that a fine-grained sense representation will result in a huge number of permutations in a knowledge base built around them, a novel method of producing tree-cuts is presented, which is based on selection of WordNet classes (i.e. Specialisation Classes) that exhibit the maximum change of information along the noun and verb hypernym taxonomies. The method is demonstrated to reduce the number of senses significantly, and Specialisation Classes are shown in a recall exercise to retain the sense-distinctions necessary to discriminate between polysemous senses to a high degree.

Chapter 7 qualifies Specialisation Classes by evaluating them in a Word Sense Disambiguation task, using Selectional Association as the sense selection mechanism. It is shown in a comparative evaluation that Specialisation Classes perform better than the full range of WordNet classes in this task.

Chapter 8 returns to the CCG parser, and notes that the structure of syntactic categories prevents incremental interpretation and the benefits to parsing it brings. Comparison of an unconfigured and configured grammar reveals that only a very small proportion of the possible syntactic categories supported by the innate human language facility are actually used once configured to a particular language. Further study reveals that syntactic categories are related structurally through inheritance. Inheritance is demonstrated to promote incremental interpretation by giving early access to left-embedded, right-looking categories.

Chapter 9 presents a walkthrough of the proposed pre-processor as a proof of concept. The walkthrough evaluates the elements of the pre-processor against expected human behaviour when processing a Garden Path sentence. Not only does the system meet expectations, and produce a correctly grammatically structured and sense-tagged parse of the sentence, but it demonstrates that coherence determination, an element of the Construction Integration Model, is instrumental in producing that parse, thereby supporting the hypothesis that the pre-processor and CIM are not separate processes.

2 Review of Summarisation Techniques

A variety of automatic summarisation techniques have been developed since the 1950s when computer technology reached a sufficient level of ability and availability to make such processes possible, and when an increasing quantity of electronic texts and data made automatic summarisation desirable.

This chapter presents an overview of those summarisation techniques, evaluating each in terms of their potential for summarising the mixed topic and multi-genre documents that typify web pages. In doing so it contrasts systems that are easily realised and rapidly executed but rely on surface statistics with those that operate along the lines of human language processing and require extensive lexical, grammatical, knowledge and processing resources.

2.1 Early Summarisation Methods

Early summarisation approaches were influenced by the contemporary computer technology; limited storage capacity and processing power, together with a dearth of linguistic resources (corpora, electronic dictionaries/thesauri, parsers etc.) dictated that implemented procedures were computationally inexpensive and required minimal linguistic resources.

2.1.1 Statistical

Luhn [LUHN58] produced the first automatic document abstract generator. It was based on the premise that the most important elements in a text will be presented more frequently than the less important ones. However, *closed class* words, comprising of a small set of frequently used terms (e.g. prepositions, articles, conjunctions) tend to

dominate [KUCE67], and must first be eliminated by means of a *stopword* list. Morphological variation thwarts conflation of terms, and a normalisation procedure is required; Luhn used substring matching, but *stemming* – the generation of artificial word roots by (repeated) removal of suffices – as delivered by the Porter Stemmer for example [PORT80], has replaced this. With the most frequent document terms identified, the top *n* words can be extracted as *keywords*. Luhn used the set of keywords to assign scores to each sentence, and presented the highest scoring sentences as an *extract*.

A number of points may be raised concerning Luhn's *Term Frequency* (TF) approach: Although morphological variation has, in part, been accounted for, other *linguistic phenomena* have not: It does not conflate *synonyms* or differentiate between *homonyms*; words of differing *syntactic class* are readily combined; no attempt is made to resolve *anaphors,* which are generally filtered out by the stopword list, whilst their *antecedent* enter into the term frequency calculation; nor is the *sense* of any word accounted for. Finally, arguments developed over a number of sentences may be inadequately represented in the resulting summary if some of those sentences' scores fall below the selection threshold, and anaphors may be left dangling should the sentence containing the antecedent be similarly eliminated.

Regarding homonymy, it has been argued that in any discourse, all occurrences of a word will have the same meaning 96% of the time [GALE92, YARO92, YARO95]. Krovetz [KROV98] argues convincingly against this, demonstrating that on average this occurs only 67% of the time - it would seem that the effects of homonymy are greater than first thought, and hence only the synonyms of comparable senses should contribute to a 'term' score in a TF based summariser [CHOD88].

Krovetz also investigated the effects of tagging words with part of speech (POS) on an Information Retrieval (IR) task [KROV97], testing the intuition that conflating like terms across POS is beneficial. Results showed that POS-tagging harmed IR performance. As the source of the degradation could be attributed either to errors in POS designation or to the separation of related terms, Gonzalo et al [GONZ99] attempted to make this distinction. However, the results were inconclusive, finding no significant difference in IR performance using untagged, automatically POS-tagged and manually POS-tagged texts. They theorise that terms matching on both stem and POS are ranked more highly and improve IR performance, but this gain is counterbalanced by a reduction in performance due to fewer matches.

2.1.2 Formal Patterns

Formally written texts implement a number of conventions in order to better present information to the reader: Titles and subheadings orient the reader to text that follows, an abstract may be explicitly presented, and paragraphs and documents tend to follow the exposition (say what you're going to say), development (say it), recapitulation (say what you've said) model. The Formal Pattern (FP) approach [EDMU61, EDMU63, EDMU69, WYLL68] attempts to utilise this knowledge of human writing conventions in the generation of summaries. Thus, selecting the first/last *n* sentences of a paragraph tend to introduce/recapitulate the information developed in the middle of the paragraph; in-text summaries can be identified by the headings '*Abstract*' '*Introduction*', '*Conclusion*', '*Problem Statement*' and the like; Lexical or phrasal cues such as '*in conclusion*', '*our results show that*', *'in a nutshell*', can indicate parts of text that are likely to contain information that should be contained in the extract. The document title and subtitles are also taken as sources of significant words, sentences containing these

words being weighted more highly. Each method is weighted (weights derived manually) and contributes to the score of a sentence. Again, sentences with the highest combined scores are then extracted into a summary.

A related method was employed in the ADAM summariser [POLL75]. Here, each item in the list of lexical phrases and cues (the word control list, or WCL) includes a code indicating whether the lexical item denotes information to be extracted (bonus items), such as those mentioned above, or ignored (stigma items) such as '*we believe'* and '*obviously'*, which should not. To improve readability, dangling anaphors are eliminated through *shallow cohesion streamlining* as described by Mathis [MATH72, MATH73].

2.1.3 Discussion

The TF method defines the most significant terms in a document as those that, after stop-words have been removed, occur most frequently. This makes the method domain-agnostic as the intuition behind the method holds for any non-narrative text. FP extends the TF method by providing cue phrases indicative of high-content text. These can be weighted positively and negatively.

As an all-round web-page summarising technique, TF is attractive because of its simple knowledge requirements – a stop-word list and a stemmer/morphological normaliser – and because its algorithmic simplicity allows real-time summarising on even the most modest of computers.

However, a basic assumption of both the TF and FP methods is that a single topic is being presented. If this is not the case the keywords, and hence the key sentences, selected will be drawn from all topics presented. I also speculate that the 'significant'

terms selected would contain general words, present in all topics, in preference to the topic-specific words, which will differ from topic to topic, the topical words effectively diluting each other. However, by taking the paragraph, rather than the document, as the unit processed, topic boundaries are detectable through change in the extracted terms, as in CHOI00. This would certainly be necessary when summarising web pages, which can be very magazine-like.

Extracting sentences on the basis of how many high-frequency terms they contain is questionable; an argument may be built over more than one sentence, and an extract may misrepresent that argument if part of it is missing. A good summary would address this problem through discourse analysis. Similarly, an extracted sentence containing an anaphoric reference can lead to a misleading summary if the sentence containing its antecedent is not extracted.

The FP method is less useful as a web page summariser as the inclusion of the WCL makes it topic and genre specific; a WCL suitable for summarising formally-written journal articles will not perform as well on newspaper articles for example.

Finally, it should be noted that both the TF and FP methods are concerned only with the surface form of a document; no account is taken of part-of-speech, grammatical role, or sense, and no attempt is made to deal with other linguistic phenomena such as *synonymy, homonymy,* and *compounds*.

2.2 Linguistic Approaches

Linguistic approaches to summarisation follow the general pattern of transforming input sentences into some internal representation, followed by a compaction phase where

repeated and redundant information is removed. The *condensate* so formed then undergoes reverse transformation to yield the natural language summary.

2.2.1 Linguistic String Transformation

Chomsky [CHOM57] and Harris [HARR51] introduced the term *kernel sentences* to describe a set of simple irreducible sentences that are related to non-kernel sentences by a series of transformations. Conversion of a document to kernel sentences gives the opportunity to select only the most important kernels for inclusion in the summary, which is produced by reverse transformations upon the set of important kernels. This method has been examined in CLIM61 and NIST71.

2.2.2 Micro to Macro Proposition Transformation

Micro to Macro Proposition Transformation (MMPT), similar in principle to the Linguistic String Transformation (LST) method, involves the parsing of natural language input into predicate-argument-structured *micro-propositions*, rather than kernel sentences. Where necessary, inferences are made to coerce coherence of any surface-incoherent micro-propositions through consultation with knowledge in the form of similarly encoded propositions stored in long-term memory. This normalises the content of a text at a logical representation level (the *micro-structural* representation of the text). Again, a compaction/elimination phase is employed - macro rules, which embody domain knowledge, are applied to the micro propositions in order to generate a set of *macro propositions.* The macro rules also ensure that the resultant macro propositions are entailed by their corresponding micro propositions. The collection of macro propositions reflects the *macro-structural* representation of the text, constituting the condensate of the original text. [VAND77];[KINT73,74,78,88,90,92,94].

2.2.3 Discussion

These techniques attempt to use linguistic theory as a method of summarisation. Both require conversion of input text into some internal representation, a simple statement that belies the complex grammatical processing required to accomplish it. The generation of kernel sentences is also complex, requiring syntactic and lexical information [BERZ79], as is the conversion of text into a propositional form. Grammatical parsers of suitable robustness are only now becoming available (e.g. [STEE00]). The transformations of LST are essentially syntax-based, leaving the system open to the usual set of problems caused by insufficient linguistic processing (e.g. attachment, homonymy, synonymy etc). MMPT on the other hand employs domain knowledge in its macro rules, which permits semantic (as opposed to syntactic) interpretation.

Using a greater depth of knowledge than LST, MMPT is more able to accurately process general texts and so seems to be the better choice of Linguistic Approach to web page summarisation. Additionally, MMPT is based on supporting experimental psychological evidence [GOMU56] [KINT73] [KINT74] [RATC78] [MCKO80], and as such might be said to model the processes by which humans read. However the sum of evidence is not yet conclusive.

These techniques are largely theoretical, and the complex procedure and enormous linguistic and domain knowledge requirements of the Linguistic Approach have so far resulted in the absence of any implemented systems.

2.3 Psychological Approaches.

2.3.1 Text-Structural Abstracting

Text Structural Abstracting (TSA), developed in RUME75 and RUME77, involves the mapping of surface expressions onto a schematic text structure typical of a document genre. Typically this may involve identification of the introduction, hypothesis, experimentation and conclusion sections (and their subsidiaries) of a report. Supplementary nodes of the structure are then pruned, leaving the core of the document, say, the hypothesis and conclusions, by way of some a priori interest specification. The core of the document may consist of text chunks or knowledge representations, resulting in a summary or a condensate respectively.

2.3.2 Discussion

TSA requires the document to be parsed syntactically and semantically in order that elements of the text are assigned to appropriate portions of a rhetorical structure tree through application of schemata applicable to that document genre [HAHN98]. Hence TSA requires excellent linguistic capabilities, a collection of suitable schemata, and appropriate domain knowledge in order to produce a rhetorical structure tree. Additionally, text-structurally licensed pruning operations are also required to eliminate all but the essential elements of the tree. The pruning operation thus requires appropriate domain knowledge in the way of ontological data and inference rules. This, together with the genre-specific schemata make TSA unsuitable for automatic processing of web pages, where different document genres and domains will be encountered.

2.4 AI Approaches.

AI approaches generally involve the mapping of document words onto representative knowledge structures. These are then combined through reasoning processes to form a representation of the document or its sentences, which is then presented to the user.

2.4.1 FRUMP

Working in the domain of newswire stories, FRUMP [DEJO82] interprets input text in terms of *scripts* that organise knowledge about common events. The occurrence of a particular word in a document will activate a (number of) script(s). The script states what is expected in that event, and instances of those expectations are sought in the document. Constraints on script variables attempt to eliminate erroneous agreement between expectations and elements of the document. This is necessary as frame activation is imperfect, being based on recognition of a cue word (or words), or implicit reference to the script by elements normally related to the event it covers.

2.4.2 SUZY

SUZY [FUM82] attempts to utilise the human approach to summarisation by employing a propositional text representation as outlined in KINT74,78. The concept of Word Expert Parsing [SMAL82] is extended to cover syntax and semantics and is then utilised in the grammatical parsing of the input text as a precursor to proposition generation. Logical text structure is determined through the location of conceptual relations between sentences and through rhetorical structure analysis via a supplied schema. Elements of the logical text structure are weighted according to importance, and the summary is formed by discarding the least important elements [FUM84]. The

discarding procedure makes use of structural, semantic and encyclopaedic rules [FUM85a, 85b].

2.4.3 TOPIC

The TOPIC summariser [HAHN90, KUHL89] operates in the domain of Information and Communication Technology, and proceeds by identifying nouns and noun phrases in the input text. These activate word experts [SMAL82] that conspire, through patterns of textual cohesion, to identify superordinating concepts contained in TOPIC's thesaurus-like ontological knowledge base. Reformulating natural language from text related to those superordinating concepts which posses frequently activated subordinate concepts generates a summary.

2.4.4 SCISOR

SCISOR [RAU89] uses conceptual knowledge about possible events to summarise news stories. It may be viewed as a retrieval system where the input document becomes a query used to retrieve conceptual structures from its knowledge base [JACO90]. SCISOR is thus applicable to multi-document summarisation. SCISOR employs three levels of abstraction in its memory organisation, inspired by current theories of human memory, these being semantic knowledge (concept meaning encoding), abstract knowledge (generalisations about events), and event knowledge (information relating to actual episodes, plus links to related abstract knowledge). In action, SCISOR responds to a user query by retrieving the most relevant elements of its knowledge base, using specific (event) and general (abstract) information as necessary, but currently can only respond to simple queries about well-defined events, such as corporate take-overs.

2.4.5 Discussion

AI approaches to summarisation require language-processing elements (syntax, semantics, and grammar) plus expert knowledge. Often, full parsing of input text does not occur, the systems being satisfied with finding coherence between nouns, although this is due to the difficulty of producing a full parse rather than a necessity of the method. All AI approaches attempt to express relationships between semantic elements by grouping elements into knowledge structures. This is advantageous in that it allows for expectation-based processing and for inferencing on unspecified information. However, the activation of stored knowledge representations is not always accurate; these representations tend to look for positive evidence for activation and ignore evidence to the contrary, unless it is explicitly encoded as constraints within the representation. Given appropriate linguistic capabilities and word-scale knowledge, the AI approach promises to be a generally applicable summarisation procedure capable of processing documents and web pages alike. However, like the Linguistic Approaches, the huge linguistic and knowledge requirements limit its operation to a few domains in which appropriate knowledge structures have been constructed.

2.5 Renaissance Approaches

Renaissance approaches generally revisit existing techniques, enhancing them with modern lexical resources (e.g. corpora: BNC [BNC], Brown [KUCE67], SemCor [FELL90] and dictionaries: LDOCE [PROC78], WordNet [MILL90]), computational power and storage capacity. These permit a degree of semantic and statistical analysis not previously possible.

2.5.1 Paragraph extraction

In order to improve coherence of generated summaries, MITR97 and SALT97 propose paragraph extraction as an alternative to sentence extraction. They represent each paragraph as a vector of weighted terms. Pairs of paragraph vectors are then compared for similarity through vocabulary overlap, a high similarity indicating semantic relationship between paragraphs. Links between semantically related paragraphs are forged, and those paragraphs with the greatest number of links, indicating that they are overview paragraphs, are considered worthy of extraction.

2.5.2 Formal Patterns revisited

KUPI95 and TEUF97 revised the FP approach proposed in EDMU69 by replacing the manual method-weighting scheme by weights obtained by training the system on a corpus of documents and hand selected extracts or author-written abstracts. The probability of each sentence being extracted is calculated for the training set, adjusting weights to maximise the probability of extracting the given extract/abstract sentences. These weights are then used to form new extracts from previously unseen documents.

2.5.3 Lexical Cohesion

Lexical cohesion refers to the *syntactic* or *semantic* connectivity of linguistic forms at a *surface structural* level of analysis [CRYS85], and might be said to express the notion that *'birds of a feather flock together'*. Lexical cohesion exists where concepts refer (a) to previously mentioned concepts, and (b) to related concepts [HALL76]. Often, these aspects are used independently by researchers, and have been usefully employed in *text segmentation* (i.e. the identification of homogenous segments of text) and in *word sense disambiguation*.

BARZ97 uses *lexical chains*, based on the ideas presented in HALL76, formed around nouns to identify major concepts in a text. A part of speech tagger identifies nominal groups, which are subsequently presented as candidates for chaining. Lexical relationships are obtained from WordNet [MILL90], and these are used as the basis for forming links between the presented nominals. The length, shape, and WordNet relation type of the chain between nominals, along with the size of window over which nominals are captured, provide a means to classify relations as *extra-strong, strong,* and *medium-strong*. The sentences that relate to the chains thus formed are then extracted to form a summary. MORR88 and MORR91 also demonstrated that the distribution of lexical chains in a text was indicative of its discourse structure.

WordNet is not the only source of relational knowledge. A document text may be used directly, a common vocabulary between parts of a text indicating that those parts belong to a coherent topic segment [HALL76], exploited in systems such as that described by Choi [CHOI00]. A common vocabulary thus consists of a set of terms that are cohesive within a topic.

A number of approaches have been employed in the acquisition of cohesive vocabularies: Term repetition has been found to be a reasonable indicator of coherence [SKOR72, HALL76, TANN89, WALK91, RAYN94]. Through *Corpus Statistical Analysis* of known coherent texts, sets of domain-related terms may be identified. As Firth [FIRT57] says:

> "*Collocations of a given word are statements of the habitual or customary places of that word.*"

Hearst's *Text Tiling* algorithm [HEAR94] uses a cosine similarity measure on a vector space to identify cohesive chunks of text and hence identify the boundaries between

those chunks. Kozima [KOZI93a, KOZI93b, KOZI94] uses reinforcement of activation of nodes within a semantic network over a given text window to indicate cohesion, the system being automatically trained on a subset of the LDOCE. Similarly, Morris and Hirst [MORR88, MORR91] identify cohesive chunks through *lexical chains*, chains of related terms discovered through traversal of the relations in Roget's Thesaurus. This approach has been adapted for use with WordNet [HIRS98]

Latent Semantic Analysis (LSA) [BERR95, LAND96, LAND97] has gained recent favour. As a corpus-based statistical method, it is similar to those previously mentioned. However, *Singular Value Decomposition* (SVD) [BERR92] is used to decompose the document-by-term vector space into three related matrices, the product of which reproduces the original document-by-term matrix. Calculating the product of the three matrices restricted to k columns (where n is the number of unique terms in the document collection, and $k << n$) then results in a best least square approximation of the original document-by-term matrix having rank k, that is, having a reduced dimensionality. It has been shown that LSA's performance is comparable to that of humans in a number of tasks: selection of appropriate word synonyms, rate of vocabulary growth [LAND96, LAND97], judgement of quality and quantity of knowledge contained in essays [FOLT99]. LSA also improves over the *bag of words* method of identifying related documents in IR [DEER90, DUMA91].

2.5.4 SUMMARIST

SUMMARIST attempts to provide domain-agnostic extracts and abstracts. It employs a three subtask processing strategy [HOVY97]:

Summarisation = topic identification + interpretation + generation

An updated version the FP location method, *Optimal Position Policy* (OPP) is used for topic identification. OPP involves a list of the title and sentence numbers, obtained thorough corpus analysis, of the likely locations of topic-related sentences for a particular domain. Interpretation involves selecting concepts from WordNet, which, through exploitation of the WordNet relationships, subsume concepts in sentences selected by OPP, thereby presenting the semantically related concepts hierarchically. The dominant concept in any hierarchy can then be said to summarise that hierarchy. Interpretation also involves assigning concepts from the OPP selected sentences to a set of *concept signatures*, broad topic classifications such as *finance, environment,* etc. Generation involves the output of topic lists (i.e. keywords), phrases formed by integrating noun phrases and clauses, and natural language sentences resulting from sentence planning.

2.5.5 Discussion

In general, current approaches apply modern resources to previously explored techniques, improving elements of those techniques. The advent of electronic dictionary resources such as WordNet and LODCE has moved term-matching toward an ontology-based concept matching, where semantic distance or similarity in information content replace simple string matching [RESN95a]. Such improvements in knowledge sources and similarity metrics are instrumental in the production of lexical chains and the identification of subsuming concepts, and ultimately give rise to the possibility of concept (as opposed to term) counting. Concept counting has distinct advantages over term counting as it, by definition, accounts for linguistic phenomena such as synonymy and homonymy. Also, as MORR88 and MORR91 suggest, topical boundaries may be more accurately located through concept matching.

Coherence is subject to the 'chicken or egg' problem; is a word's sense defined by its inclusion in a group of cohesive words, or is the cohesive group created by collecting word senses that are related in some way? Lexical chaining involves the former, seeking some relation through traversal of the WordNet relations for example, thereby acting as a Word Sense Disambiguation (WSD) procedure. However, as the grammatical relations between the words are not factors, inappropriate sense disambiguations, and hence inaccurate coherence, are made. For example, the WordNet definition for *alarm-clock* (below), when processed by the lexical chainer described in HEAR98, incorrectly identifies coherence:

Alarm-clock: wakes sleeper at preset time.

Seeking coherence between the first two nouns, *alarm-clock* and *sleeper*, a superordinate class DEVICE is found between the (given) sense of *alarm-clock* and the *railway-sleeper* sense of *sleeper*; the verb *wakes* is more closely associated with sleep, as is *sleeper*, and would have been the better choice to seek a relation. Although capable of discovering relations implicit in knowledge structures such as WordNet, lexical chaining is slow in operation due to the large number of relation-traversals and node comparisons necessary.

Like lexical chainers, approaches based on LSA do not use grammatical information, treating documents as bags of words from which co-occurrence information is derived. LSA is the opposite of lexical chaining in that the sense of coherent words is defined by the context that the coherent group provides. However, the actual nature of the relations between the identified coherent terms is unknown.

2.6 Web Page Summarisation

The approaches discussed so far have been concerned with plain text documents. Today, a huge number of documents are available over the Internet as Web Pages, and it is the production of summaries of these pages as an assistive technology for blind and visually impaired users that is the driving force behind this project. A number of summarisation techniques relate directly to the web itself.

2.6.1 Page Layout Analysis

Our early work attempted to use the HTML markup tags to identify features of the document, such as headings, which might be highly semantically loaded. However the lack of consistency between visual features and the tags used to generate them in different documents/document authors proved problematic. For example, headings may be defined by the tags <*h1*>..<*h6*>, or may be constructed by use of the *size* attribute of the <*font*> tag. To overcome this problem, information regarding the visual effect (e.g. size of text, text font, spacing around text) of the tag rather than the tag itself was used to provide a *Page Layout Analysis* similar to that employed when applying Document Image Analysis and Optical Character Recognition to a printed document [PAVL92], thereby allowing the identification of headings, text blocks, footnotes, figure and table labels. However this work has been temporarily abandoned in order to address the underlying problem of extracting meaning from text regardless of its source.

2.6.2 BrookesTalk

BrookesTalk [ZAJI97a, ZAJI97b, ZAJI99] implements a basic *Term Frequency* (TF) summariser, comprising a stopword list, porter stemmer [PORT80] and stem frequency analysis augmented by trigram analysis [ROSE97]. It also incorporates positive

weighting of words from headings and links in the assumption that these elements are indicative of important document topics; in general, headings summarise the information to follow and links provide access to related information. Although popular, TF suffers from a number of drawbacks, briefly: it cannot account for linguistic phenomena such as synonymy, lexical ambiguity, or multi-topic documents.

2.6.3 Discourse segmentation

Choi addresses the multi-topic problem by identifying discourse segments through *linear text segmentation* [CHOI00]. Discourse segments are then grouped into topics either through application of a clustering algorithm [RAYN94], or, as some alignment has been observed between topical shift and presentational features, through observation of those presentational features. With the topic boundaries defined, a combined word-frequency, word-position and word length summarisation procedure [CHOI99] produces a keyword list for each topical segment within the document. In theory, short documents may suffer as a result of this procedure, as the further reduction in word count due to topicalisation may affect the word frequency statistics. Then again, the topicalisation procedure will have concentrated related words, possibly assisting the frequency statistics.

2.6.4 Gists

Other methods use external data to draw-in additional related words, which in part addresses the linguistic phenomena issues: Berg [BERG00], noting that web pages contain “a chaotic jumble of phrases, links, graphics and formatting commands which are not suited to standard statistical extraction methods”, utilises word frequency, trigram and word relatedness models derived from a collection of human generated web

page summaries [OPENDP] to select and arrange words from a web page to form a *gist*. A related method uses the hyper structure of the web to provide alternative defining information for a web page - as a hyperlink generally points to a page related to the paragraph containing that link, Amitay et al [AMIT00] collect paragraphs around links pointing to a target page. A filter, constructed through analysis of human-selected 'best paragraphs', is then applied to those paragraphs to identify the best description of the target page.

2.6.5 The Semantic Web

As recognised in Section 2.6.1, the HTML markup used by today's web pages is concerned with formatting for human readability. This is because the means of accessing information in a web page is expected to be natural language. The Semantic Web [BERN00] is a vision of machine-readable documents and data, canonically annotated to allow software agents to determine the document topic(s), and the people, places and other entities mentioned within. Documents marked up in this way would be, given appropriate software agents, be amenable to such applications as knowledge discovery and summary production.

The Semantic Web requires two classes of metadata to facilitate this: Ontological support services to maintain and to provide on demand the entity-related metadata, and large-scale document annotation using semantic markup formulations such as XML, RDF [W3C99] and OWL [OWL].

Retro-fitting semantic metadata to the billions of existing web pages would be impossible to achieve by hand, leaving automatic annotation as the only viable route. Attempts have been made to use Machine Learning (ML) (e.g. Naïve Bayes [MITC97],

K-Nearest Neighbour [DUDA75]) to extract structured information from web pages [KOSH97; LERM01; COHE01], but these require significant training before they can be productive [DILL03]. Edwards [EDWA02] proposes that:

> "Once content has been extracted from documents, the next step is to apply information retrieval techniques, such as stopword removal, stemming, term weighting and so on. A bag-of words representation is then used to form the training instances required by the learning algorithm."

As has been stated previously, these approaches do not attempt any linguistic analysis and are subject to the linguistic phenomena outlined in Section 2.1.1. Their utility as a means of producing training data for ML algorithms is therefore questionable. However, Natural Language Processing (NLP) techniques, such as the system proposed in this thesis, applied to the plain text of web pages would permit 'cleaner' training datasets to be obtained.

The Semantic Web and NLP of the kind proposed here share a common interest in the ontological support services, which is a fundamental element of both. As shall be seen in Chapter 6, the WordNet 1.6 lexical database is used in this work as a ready-made sense ontology. Although WordNet is an excellent lexical research tool, it has many lexical and relational omissions that prevent it from being a world-scale ontology. The ontological support services developed as part of the Semantic Web project will be beneficial to the NLP community in this respect.

2.6.6 Discussion

The above methods involve extensions to the TF method, either by attempting to identify topic boundaries, or by drawing in additional information from related web pages in order to bolster the statistics. Ultimately, these approaches rely on analysis of surface features and do not attempt any form of linguistic processing such as

grammatical parsing, WSD, etc, and so are subject to the same problems as the TF method.

2.7 Conclusions

The summarisation techniques presented above present a spectrum of capabilities and requirements: TF based summarisation requires little knowledge, executes rapidly, and is domain-agnostic. However, it is not amenable to certain document types, such as narratives or short texts, where there is no overall theme to detect or insufficient information to reliably detect significant words. Also it is subject to error induced by the lack of linguistic processing: anaphors are left dangling, synonyms are not identified, etc. Although ingenious, these approaches ultimately rest on statistical features derived from the surface analysis and clustering of surface elements from web pages and/or training text. Our impression of surface feature based techniques is that by not utilising the full spectrum of linguistic information available (for example, they don't make use of sense or grammar), they are imposing an upper limit on their performance.

At the other extreme are those techniques that heavily involve linguistic processing, requiring wide coverage grammars, WSD, kernel transformations and the like. These promise excellent summarisation capabilities, and have support from psychological studies, but require a huge amount of knowledge which in turn requires a significant amount of processing, resulting in current systems that operate on limited domains.

The choice then is between systems that are easily realised and rapidly executed but rely on surface statistics, and those that operate along the lines of human language

processing (which is the only working language processing system we know of) but which require huge amounts of knowledge and processing effort.

As it would appear that there is little progress to be made by pursuing the statistical, non-linguistic approaches, the only option is to look at the psychologically oriented methods of summarisation. However, although not necessarily computationally intractable, the complex nature of psychological approaches and their huge linguistic, grammatical and knowledge requirements make the production of a wide coverage summariser built along such lines unfeasible within the context of a thesis. As a precursor to such a system however, an attempt to quantify and reduce the workload of a psychologically based summariser would seem prudent.

3 A Model for Discourse Comprehension

This chapter proposes discourse comprehension as an initial step in summary production, and presents the Construction Integration Model as a suitable model to perform that task. Evidence for the model is presented, and in doing so, we note that the model proper accepts input in a logical form; the details of how natural language is converted into logical form is not a defined part of the model.

When a human summarises a text, the basic operations involved are the reading and comprehension of that text, followed by the reproduction of that which was comprehended, but in a more concise and/or tailored form. From this we suggest that the task of producing indicative summaries includes that of discourse comprehension. It would therefore seem appropriate to approach the task of automatic summarisation from the direction of discourse comprehension, that is, via the Linguistic, Psychological, and/or AI approaches introduced in Chapter 2. Of these, the Linguistic approaches of LST and MMPT offer the better starting point as they:

1. do not require knowledge of rhetorical structures as would Psychological approaches;
2. do not require the scripts employed by AI approaches.

The scripts and rhetorical structures above place constraints on the capabilities of a discourse comprehension system as these items firstly must be prepared beforehand, secondly a missing script or structure description will impair accurate comprehension and/or induce domain specificity, and thirdly, a mechanism must be employed to select

the appropriate scripts and structure definitions in any situation. Of course Linguistic approaches have their own requirements such as kernel sentence extraction and macro rules, but these are of a very general nature, trading efficiency for coverage [KINT90]. As a development of MMPT, the Construction Integration Model (CIM), has been selected as the framework for study in this work as it proposes a model of discourse comprehension based upon and supported by psychological evidence, although the evidence is in no way conclusive.

The CIM makes an interesting setting for a summarisation system as it also acknowledges summarisation as a necessary step in human text comprehension. The reading of text involves the recognition of written symbols and the subsequent construction of some meaning representation based upon them This in turn requires the combination of current symbols with symbols read previously and now residing in memory. However, Miller [MILL56] demonstrated that the human short-term memory (STM), used for active processing, had a capacity of 7±2 chunks of integrated pieces of information. As limited STM is available for active processing, it follows that the representations (i.e. integrated chunks) are necessarily summarised in the natural course of the reading process [ENDR98].

3.1 Background to the CIM

Chomsky suggested that External Languages (E-Languages), such as English and German, were mere ‘epiphenomena’ that required further definition with ‘socio-political and normative factors’, presenting as evidence the observation that there exist similarities in dialects across the borders of countries such as Germany and the Netherlands (where the languages are different but similar) and local dialects and shifts in grammar within a country (where the language is the same but different) [CHOM88].

Essentially Chomsky said that the development of an E-Language is subject to the behavioural response to external factors, and that these are difficult or impossible to predict or incorporate into definite rules. This view was also put forward by the descriptive linguist Twaddle when he said:

> '*We know that the habit is the reality and the rule is a mere summary of the habit.*' [TWAD48].

Attempts are being made to describe E-Language grammars as complex adaptive systems [DILL97], but if E-Language is, as Chomsky and Twaddle suggest, a tradition perpetuated for convenience, then the real work of comprehension occurs at the I-Language level (i.e. at the *internal* rather than *external* representational level). Indeed, the CIM is mostly concerned with I-Language; only the first step of the Construction process involves E-Language (see Section 3.3).

3.2 Experimental Evidence Supporting the CIM

Gomulicki showed that it was extremely difficult for people to distinguish between a précis of a story and the recalled memory of that story, the conclusion being that a précis resembles a story memory, as both contain the main points and ignore unnecessary detail. This is understandable when one again considers the storage limitations of working memory [GOMU56].

Kintsch and Keenan demonstrated that time taken to read sentences with a fixed number of words increased according to the number of propositions they contained, concluding that extra cognitive effort is required to integrate the additional propositions into the internal representation. [KINT73].

3.2.1 Evidence for Propositions

Although Gomulicki's experiment indicates a reduced representation, it does not suggest what that representation might be, or how it is structured. Kintsch and van Dijk proposed a model of story comprehension which consisted of two basic elements: the *argument*, equating to the representation of the meaning of a word, and the *proposition* equating to the smallest unit of meaning to which a truth value can be assigned (a phrase or clause in general). In addition, it was proposed that the story is processed to form two main structures: the *micro-structure*, being the connected structure formed by the propositions extracted from the text, and the *macro-structure*, being the reduced version of the micro-structure (i.e. the main points of the story) [KINT78]. This was partly based upon the experiment performed by Kintsch and Keenan where subjects were presented with sentences of a fixed number of words but varying numbers of propositions. In recording the subjects' reading times of the sentences it was found that reading time increased by around one second for each additional preposition in a sentence [KINT73]. Their conclusion was that the increase in reading time could be accounted for by the additional cognitive effort required to incorporate the increased number of propositions into the internal representation.

An experiment by Ratcliff and McKoon demonstrated that propositions might be the basis of internal representations. They presented subjects with a number of sentences, and followed this with a word recognition test. The subjects had to decide as quickly as possible whether words presented by the experimenters were from the previously presented sentences. It was found that recognition times decreased for any word where

the preceding word was from the same proposition. This priming effect suggests that the first word causes the recall into working memory of the proposition relating to that word. If the subsequent word is from the same proposition, then the representation is already in working memory and the recall overhead is avoided [RATC78].

Further evidence for the importance of propositions was provided by Johnson-Laird and Stevenson: They presented experimental subjects with sentences, and later performed recognition memory tests on those subjects. In the recognition test some of the original sentences were replaced by others with the same meaning but different wording and syntax. In these cases, the test subjects mistakenly reported that they recognised the sentences and that they had been previously presented; the meanings of the sentences were remembered rather than the words of those sentences. Thus, as propositions express meanings rather than words, it can be concluded that the propositions are an integral part of the internal representation.

3.2.2 Evidence for Micro and Macro Structures.

There is also evidence for the existence of the micro and macro-structures. An experiment by McKoon & Ratcliff [MCKO80] involving subjects reading a paragraph, and then being tested for recognition memory through presentation of additional sentences, and finally having to determine whether the concepts expressed by the sentences were contained in the original paragraph. The response times to the recognition test were recorded and used to demonstrate that the fastest recognition times were obtained when two sentences from the paragraph that formed part of the same micro-structure were presented consecutively. This priming effect was not shown in

cases where the two consecutively presented sentences were not part of the same micro-structure.

Kintsch also demonstrated the difference between micro and macro-structures [KINT74]. Subjects were given a text to read, and immediately after reading were asked whether certain explicit or implicit inferences were contained within the text, and their recognition times recorded. A second group of subjects were given the same task, but the recognition test was performed 15 minutes after the text was read. Again recognition times were recorded. Analysis of the results showed that, for the immediately tested subjects, the explicitly presented propositions were recognised faster than implicitly presented propositions. However, for the subjects tested 15 minutes after reading the text, there was no discernible difference in recognition times. The conclusion drawn was that the micro-structure representation of explicit propositions is better than that for implicit propositions, but the representation of both is comparable in the macro-structure. The information contained within the micro-structure is available immediately but appears to degrade quickly, whereas the macro-structure appears to be a more permanent entity.

3.3 The Construction-Integration Model

Kintsch further developed the model described above into the Construction-Integration model [KINT88, 92, 94]. The stages of the process are as follows:

Construction:

1. Propositions are formed from the input sentences.

2. The propositions form a *propositional net* in a short-term buffer.
3. Propositions related to those in the propositional net are recalled from long-term memory and combined with the propositional net to form the *elaborated propositional net*.

Integration:

4. The most highly interconnected propositions within the elaborated propositional net are selected through a spreading activation process.
5. The selected propositions are organised into a *text representation*, which is stored in *episodic memory*.
6. Through a learning process, episodic memory is transferred to *long term memory*, where it becomes available for subsequent constructions of elaborated propositional nets.

The model improves upon the previous model in a number of ways: It shows how stored knowledge can interact with textual information, and provides a framework for inference-making. The model also makes the assumption that during the construction of the elaborated propositional net, many possible propositions are included. Most of these will be irrelevant, making the process inefficient but generally applicable. The alternative is to use 'intelligent' rules to select the correct propositions each time, an approach that possibly eliminates the flexibility and robustness of the construction-integration model [KINT90].

What evidence is there for the CIM itself? It predicts three levels of representation, the surface level (the text itself), the propositional level, and a situational level (or episodic memory, equating to a representation similar to that obtained through direct experience of the situation). As we have seen, it also predicts that memory for the original text form degrades more rapidly than that for the more summarised propositional memory. It follows from the model that propositional memory should degrade more rapidly than episodic memory, episodic memory being a generalised version of propositional memory. These predictions were compared against experimental observations of subjects given recognition tests immediately, or at varying time intervals ranging up to four days, after being presented with a situation (going to a restaurant, going to see a film) [KINT90]. Analysis of results showed that the surface representation was quickly forgotten, the propositional representation was partially forgotten, but there was no forgetting of the situational representation. Thus the predictions of the model were consistent with observation.

3.4 Conclusion.

The construction-integration model proposed by Kintsch & van Dijk is reasonably well supported by experimental evidence; observation of memory recognition and retention at the surface, propositional and situational representational levels are consistent with the theory, thus supporting the idea of micro and macro-structural levels in discourse processing. It is possible that propositions frequently recalled into working memory, or remaining in working memory for extended periods, are those that relate to the main themes of a text. However there is evidence to show that factors other than re-presentation of arguments or propositions are also involved in the production of an

internal representation of a text. For example, sentences that are linked causally are read more quickly than those that are not [TRAB85]. This suggests that structures other than propositions linked by similarity are needed in order to fully describe a discourse comprehension system. However, this underlines a general problem of the CIM as it currently stands; although defined in general terms, which has previously been recognised as promoting coverage, no specific details of the operation and underlying structures of the model are given – the question 'how are propositions formed, inferences made, and how do stored propositional and situational representations interact with textual representations?' remains unanswered. To be fair, Kintsch recognises this when he says:

> '*comprehension always involves knowledge use and inference processes. The model does not specify the details of these processes*'. [KINT78].

It seems that the issue of transforming a text into propositional form has also been sidestepped:

> *'the model takes as its input a list of propositions that represent the meaning of a text.'* [KINT78].

This statement places the CIM entirely in the I-Language domain. Recognition of the E-Language to I-Language transformation is given however:

> '*a full grammar, including parser, which is necessary for the interpretation of input sentences and for the production of output sentences, will not be included*'. [KINT78].

It would seem from the above statements that the conversion of text into propositional form (step 1 of the CIM) is being relegated to a pre-processing step.

We do not believe this position to be accurate, and a study of a grammar compatible with the CIM demonstrates why.

4 A Psychologically Plausible Grammar

Chapter 3 showed that the CI model is attractive as a model of *discourse comprehension* as:

- it has psychological validity;
- it addresses both *local* and *global* coherence;
- it results in a *coherent text-base* from which *kernel sentences* may be extracted and used as the basis of a summary.

This chapter examines the fundamental elements of a pre-processor for the CIM, which are identified as being the conversion of text to logical form, sense assignment, and coherence determination. It shows that sense is instrumental in the conversion to logical form *and* to coherence determination, and consequently proposes that these elements are interrelated and should not be treated as autonomous processes. It further proposes that to achieve this integration, the processes of the CIM should extend into the pre-processor itself. After proposing that the grammar parser is the process that unites the three elements, theories of grammar acquisition are examined, and Categorial Grammar is selected for this work as it is shown to be consistent with those theories of grammar and capable of producing output in logical form.

4.1 Elements of a CIM Pre-Processor

An implemented summariser based on full language comprehension is of course beyond the scope of this thesis. However, in accepting the CIM as the basis for such a system, the major elements of a summariser become those of the CIM itself, which in this work are taken to be:

1. conversion of *surface form* to *logical form*;
2. assigning *senses* to the *predicates* and *arguments* of the logical form;
3. seeking *coherence* between *logical units*.

At first inspection, it would appear that each of the above elements of the CI model might be investigated separately: Point 1 involves the conversion of text from surface to logical form, essentially a grammatical analysis of an input sentence - the discovery of the underlying relationships between the input units. Many *grammar parsing methodologies* are in existence which could potentially perform this function, for instance Phrase Structure Grammar [CHOM57, GAZD85, POLL94]; Tree Adjoining Grammar [JOSH75, XTAG]; Categorial Grammar [BAR53, LAMB58]; Dependency Grammar [MELC88]; Government and Binding/Principles and Parameters [CHOM82, HAEG94].

Point 3 is concerned with the seeking of coherence between the propositional units discovered by grammatical analysis, which involves the matching of arguments of one proposition against those of another [VAND77];[KINT78].

However, the second point, that of assigning senses to predicates and arguments, is central to both points 1 and 3, as shown below:

4.1.1 Sense is central to grammatical form

It is straightforward to demonstrate through prepositional attachment ambiguity that sense information (point 2) is needed to successfully determine the grammatical structure of a sentence (point 1). Consider the following sentence:

`John made the cake in the` **`X`**`.` (1)

The grammatical structure of sentence (1), specifically the site of prepositional phrase (PP) attachment, is dependent upon how the *meaning* of the word **x** interacts with the meaning of the other words, as illustrated below:

John made the cake (in the **box**). (2)

John made the cake (in the **kitchen**). (3)

John made the cake (in the **nude**). (4)

In sentences (2)-(4) the site of PP attachment is underlined, and is determined *pragmatically* – cakes are stored, not made, in boxes; cakes are typically made in kitchens; only people may be nude. If no account of word sense is taken during the parse, then the parser must generate all parses licensed by a PoS grammar or lexicalised grammar.

4.1.2 Sense is central to coherence discovery

Regarding coherence (point three), Kintsch and van Dijk present a worked example of coherence checking, using the document "Bumper Stickers and the Cops" [KINT78]. In their example, coherence is found (for example) between the argument terms '*law enforcement officer*' and '*police*'. The problem here is that, working at the surface level of representation, these two terms cannot possibly be identified as similar as they have different surface forms. This exemplifies the problems encountered when seeking coherence between words, for example, between synonyms such as *dog* and *hound*, *car* and *jalopy*, and between polysemes such as *bicycle* and *unicycle*, *car* and *lorry*, *breakfast* and *dinner*. The simple conclusion is that coherence may not be sought between the surface forms of arguments and predicates, only between their *meaning*

representations, implying that some notion of *sense* be employed during the coherence seeking exercise.

4.1.3 A mutually constraining approach

The above has shown that sense is central to both grammar parsing and coherence discovery. However, determining the sense of words (Word Sense Disambiguation – WSD) is an extremely difficult problem that has received much attention (e.g. [LESK86];[RESN95b];[YARO95];[BRUC94]). We propose that to avoid this problem, the processes of grammatical parsing, WSD and coherence discovery should not be viewed as separate sequential processes. Instead we propose that parallel, mutually constraining processes better describe the situation presented in Sections 4.1.1 and 4.1.2, which show that grammar, coherence and sense are related. So, given a system that produces all possible grammatical parses with all possible senses assigned to the grammatical constituents, selection of the correctly sensed grammatical parse is achieved as a consequence of identifying the most pragmatic and world-consistent coherent grammatical structures. This approach is consistent with the CIM in that context is provided by the entire body of text processed (the local and global contexts) together with world-knowledge from long-term memory, not by the current sentence alone (the local context) as is the norm for PoS taggers, WSD algorithms and grammar parsers. This is of course an extension of the CIM into the E-Language domain; rather than seeking coherence between correctly parsed and sensed logical elements, the CIM would select from the possible logical units and senses by seeking those most coherent, that is, those most consistent with local, global and world knowledge. In this way the discovery of coherence, grammatical structure and sense are integrated into the CIM

itself. However, a mutually constraining system of this type will still need a lexicon, grammar parser, and some sense representation, together with appropriate knowledge to allow pragmatic coherence discovery to take place.

4.2 Selection of the grammar parser

A grammar parser is needed to identify structural relations between the terms of a sentence, a precursor to the conversion of a natural language text into a logical representation. We propose that the grammar parser is the site at which the sense and cohesion elements intersect because the parser is the process that generates logical forms, and these have been shown in Section 4.1.1 to be affected by sense, and in [KINT78] to be subject to coherence testing.

In order to complement the psychologically valid CIM, a similarly valid system of grammar acquisition/processing was sought. This was because, having only one example of an implemented language system available for study, i.e. the human language system, it seemed prudent to follow this as closely as possible.

There are two major schools of thought regarding the acquisition of language, differentiated by the terms *Inside-Out* and *Outside-In*. Essentially, Inside-Out theories assume that *language-specific* cognitive processes are involved in language acquisition and comprehension, that is, language arises from specialised processes *inside* the brain. Outside-In theories assume that language is just another type of input to the brain, and its acquisition and comprehension requires only *general* cognitive processes which are applicable to any form of input. In this case, language is an external phenomena occurring *outside* the brain.

4.3 Inside-Out Theories

The essence of Inside-Out theories was presented by Karmiloff-Smith [KARM89] who noted that all species seem to be endowed "with a considerable amount of biologically specified knowledge". Chomsky [CHOM88] justified this position through his P*overty of the Stimulus* argument, noting that linguistic input is "far too impoverished and indeterminate" for a distributional analysis to be the basis of the acquisition mechanism.

4.3.1 Evidence for the Poverty of the Stimulus Argument.

Observation of linguistic phenomena provides evidence for the Poverty of the Stimulus Argument. Here we shall examine three phenomena: Empty Categories, Hierarchical Organisation, and Surface Cues.

Empty Categories [CHOM75, CHOM81] are categories not present in the linguistic input, but which nevertheless must exist at some abstract level for full comprehension to occur by the recipient. For example, the imperative sentence "Stop that!" implies the empty category "you", as in "You stop that!". The existence of the abstract representation of empty categories is supported by results of online sentence processing [BEVE88], showing that learning from imperative sentences such as "Get dressed!", leads to the generation of parallel sentences such as "Get dressed yourself!" by the recipient, whereas incorrect sentences like "Get dressed himself!" are not.

Hierarchical organisation of sentence structure cannot be derived directly from sentence surface form, again suggesting that some innate knowledge is employed in revealing the structure. The following example (from [HIRS99]) demonstrates that identification of the main clause of a statement is required in order to convert that statement into a question via verb 'fronting':

```
The man who will come is John.
```

The verb of the main clause of the sentence (`The man` **`is`** `John`) may be fronted to make the question:

```
Is the man who will come John?
```

However, the verb from the subordinate noun phrase (`who will come`) may not be fronted, as to do so would result in the ungrammatical question:

```
Will the man who come is John?
```

Identification of main and subordinate clauses via parsing of surface form is not guaranteed to provide just one derivation, hence hierarchical organisation is ambiguous, making generalisations (such as 'question formation by verb fronting') difficult, if not impossible, to acquire.

Surface cues can be used to make generalisations about thematic structure, as described by the *Competition Model* [BATE87]. However, when no surface cues are present, humans do not make the errors in interpretation due to overgeneralisation that the Competition Model predicts. For example, in the following three (impoverished) sentences (from [GIBS92]), the preverbal noun is predicted to be associated with agency:

```
 i. The chicken cooked the dinner.
ii. The chicken was cooked.
iii. The chicken cooked.
```

As Gibson [GIBS92] points out, only in the first sentence is the preverbal noun (chicken) the agent, becoming the *theme* in the latter two. However, despite the lack of

surface cues in the above sentences, children seem to correctly identify agent and theme, again suggesting a mechanism not reliant upon analysis of an impoverished surface form.

The evidence presented above supports the Poverty of the Stimulus argument as a justification for Inside-Out theories of language acquisition mainly because the Inside-Out viewpoint denies the possibility of a language system built on general cognitive principles from bootstrapping from input alone – the input is too impoverished for such a feat, and therefore the suggestion is that language-specific cognitive processes must exist which are sensitive to the above phenomena.

4.3.2 Principles and Parameters

Chomsky [CHOM72] proposed the *Language Acquisition Device*, consisting of linguistic rules and transformations, as the mechanism for dealing with all linguistic phenomena that could not otherwise be accounted for by analysis of linguistic input. However, the ever increasing number of rules and transformations resulted in Chomsky's development of his *Principles and Parameters Theory* [CHOM81], consisting of a set of invariant parameters, such as the projection of nouns and verbs into noun phrases and verb phrases, which are configured by parameters set by the local linguistic environment, an example being verb position (SVO, SOV etc.). Thus Principles provide a mechanism for expectation of behaviour, and Parameters configure the Principles for local use. The Principles and Parameters Theory is equated with the *Language Facility* [CHOM88], that is, the innate linguistic system before exposure to a linguistic environment. Chomsky also assumes that the Language Facility is replete

with knowledge of concepts such as physical objects, causality, intention and goal, and proposed that exposure to a few examples would be sufficient to set the parameters to the local environment. However, experimental evidence [WEXL85] involving the late appearance of Principle B of the binding theory caused parameter setting to be relegated to 'biological maturation'.

In addition to the inherent knowledge listed above, Pinker [PINK84] includes knowledge of syntactic class, word class, grammar structure and primitives, and assumes that the language learner is sensitive to them in the input. Pinker therefore suggests that language learning becomes *Semantic Bootstrapping*, where identified elements in the input are mapped onto innate elements in the Language Facility. However, this presupposes an ability on the part of the language learner to select/make certain mapping hypotheses over others regarding words and grammar, in order to obtain a correct mapping [GLEI90].

4.3.3 Against the Inside-Out Theories

As an Inside-Out theory, Chomsky's Principles and Parameters addresses the Poverty of the Stimulus argument by providing a mechanism for innate knowledge to interact with impoverished input, leading to the bootstrapping, via configuration and sensitivity, of a local language processing system. However, arguments have been levelled against the theory: firstly, that the Language Facility is innate is unfalsifiable [FODO87], secondly, evidence that the Language Facility develops rather than is merely configured has been presented [KUCZ77], and finally, stating that biological maturation is responsible for parameter setting does not improve the understanding of how and when principles come into play [WEIN87].

4.4 Outside-In Theories

Outside-In theories were developed as a response to the Chomskyan view, and propose that general cognitive processes perform language acquisition and understanding, removing the need for a Language Facility by relegating language to 'just another input' which may be processed through application of the same domain-general learning techniques as any other learning task.

4.4.1 Evidence for domain-general language acquisition

Social Interactions have provided evidence that general learning techniques are employed in language acquisition. Brunner [BRUN75] proposed that children at play take on, for example, the roles of 'giver' and 'receiver' of actions, and that these map directly onto the linguistic roles of 'agent' and 'recipient of action', hence social interaction promotes language learning. It has also been proposed that Social interactions provide source material for the construction of *scripts*, which themselves form the substrate upon which language is constructed [NELS85].

In this social-interaction view of language acquisition, the language learner has no need to bring specific linguistic knowledge to the task because language develops as a means of interpreting the social interactions the learner observes or is involved in.

Cognitive Semantic Categories and Relations such as agency, animacy, causality, location, it has been proposed, provide a child with the means to interpret their environment, and acquiring a language involves learning how to express these categories and relations, evidenced by child event perception and memory studies [SCHL88]. Evidence has been presented for categories such as agency [GOLI81], animacy [GOLI84], causality [COHE93] and location [MAND88, MAND92]

Bates and MacWhinney [BATE87, BATE89] similarly propose a domain-general model of language acquisition which possesses little initial linguistic structure and in which the grammatical properties are founded on non-linguistic categories and processes. Within the model, a grammar is learned by distributional analysis of the input through use of general-purpose cognitive mechanisms such as induction and hypothesis testing. This approach to language learning has been modelled by Plunkett [PLUN95].

4.4.2 Against the Outside-In Theories

Outside-In theories of language acquisition demonstrate that it is possible to learn language using domain-general cognitive processes. However, the theories make a number of assumptions: Firstly they presuppose the existence of 'categories' upon which the processes act, without describing how they were gained. Secondly, although general purpose reasoning processes can be used to analyse input in terms of those categories, how this ultimately becomes an adult language processing system is not clear [BLOO75]. Thirdly, Chomsky's Poverty of the Stimulus argument indicates that the input is not ideal for distributional analysis because some information is omitted from the surface representation.

4.5 The Coalition Model

Initially, Inside-Out and Outside-In theories appear to be in opposition, making initial language either 'structure linguistic and innate' or 'cognitive/social and constructed', and language learning either domain-specific or domain-general (respectively).

Hirsh-Patek and Golinkoff [HIRS99] note that the above theories each contain aspects of the other, and attempt to show, by way of their 'Coalition Model', that the theories "differ more in degree than in kind". For example, Chomsky [CHOM88] assumes that

both social understanding (e.g. a knowledge of theta-roles) and general cognitive processes, normally associated with Outside-In theories, contribute to language learning within the Inside-Out theory. Conversely, Schlesinger [SCHL71, SCHL88] presupposes sensitivity to inflectional markers, a position which would not be out of place in an Inside-Out theory. It appears then that both schools of thought allow for initial (although undeveloped) linguistic sensitivities, and are capable of making language-relevant generalisations about their environment.

In order to make linguistic generalisations via domain-general learning processes (i.e. Outside-In) the language learner must possess knowledge that enables linguistic elements to be identified. For example, research has shown that language learners are sensitive to word order, inflectional marking, tense [SCHL79, BATE87], and even verbs [GIBS92]. Thus the Outside-In theories employ knowledge of linguistic units that belong in the Inside-Out theories, and the learning processes become more domain-specific. The effect of employing language-specific knowledge is to reduce the hypothesis space of the domain-general learning processes, thereby promoting the formation of appropriate linguistic generalisations, and ultimately acknowledges a degree of innateness in the language processing system.

Hirsh-Pasek and Golinkoff [HIRS99] formulate the Coalition model by taking all elements from Outside-In and Inside-Out theories, as opposed to their intersection, and placing them in a framework of three questions:

What does the language-learner initially bring to the task? Both theories assume that the language-learner is sensitive to linguistic elements (e.g. nouns, verbs, phrases and clauses, inflectional markings) and their possible arrangements, Inside-out theories accounting for this in an innate knowledge of structure, whereas Outside-In theories look to the general analytical processes employed. In either case, linguistic elements are made available to the language learner.

What mechanisms are employed in language acquisition? Both theories employ processes, be they language specific or general, to process the language units. These processes detect hierarchical structure by noting the order/relation of language elements with respect to the larger language elements that contain them. The processes must also deal with impoverished input, such as Empty Categories, in order to accomplish this task.

What input types drive language-acquisition? The variety of input types needing analysis, such as social interaction, prosody, and syntactic patterns are each taken as central by different theorists. The Coalition Model accepts that all are important, and that "reliable covariance" between the input types provides a strong force driving language development forward.

4.6 Categorial Grammar

The Coalition Model recognises the importance of elements of grammar acquisition regardless of the founding theories of those elements, and a grammar system for use with the CIM should also acknowledge those elements in order to achieve cognitive validity. We believe Categorial Grammar (CG) [WOOD93];[STEE00] to be such a

system of grammar. Before showing consistency with the Coalition Model, a brief introduction to CG is presented.

CG is an *explanatory* theory of grammar, much research showing that it can account for many linguistic phenomena such as *empty categories*, *relativisation*, *parasitic gaps*, *subject/object asymmetries*, *coordination*, *extraction of subject/object from embedded clauses*, and *gapping*; this wide coverage of linguistic phenomena therefore make it suitable as a general grammar processor. CG employs syntactic and semantic categories and a system of combinators to syntactically parse a given sentence and reveal its logical structure in propositional form.

The basic claim of CG is that:

> ' *…syntactic structure is the characterisation of the process of constructing a logical form, rather than a representational level of structure that actually needs to be built*.' [STEE00]

CG thus attempts to account for the surface-level syntactic structure of a text in terms of the logical form a representation of that text must take. This is accomplished primarily through recognition of the rule-to-rule relation between syntax and semantics, which gives rise to three consequences:

1. Every syntactic rule has a semantic interpretation. This in itself implies that syntactic rules can only combine or yield rules. This consequence is known as *The Constituent Condition on Rules*.
2. Only grammatical entities that have interpretations are constituents.
3. Syntax and semantics should have the property of monotonicity – no rule should transform an ill-formed structure into a well-formed one. This should be true whether the rule is being applied to constituents derived from the surface

structure, constituents which express a fragment of a sentence, or combination of either.

From these consequences a general implementational framework may be devised which consists of syntax, semantics and compositional rules. The following subsections present a brief CG primer; for a full introduction see [WOOD93, STEE96, STEE00].

4.6.1 Syntax.

CG utilises a simple alphabet, comprising N (noun), NP (noun phrase), PP (prepositional phrase) and S (sentence), together with the forward and backward slashes '/' and '\'. These may be combined to form constituents which are assigned to input words through lexicon lookup, or derived through application of combinatory rules.

A categorial grammar is in general equivalent to phrase structure (PS) grammars in that it expresses the same ordering of syntactic components. Consider the following simple PS grammar rule:

$$S \rightarrow NP\ VP$$

The PS rule for the sentence (S) states that a sentence is formed by a noun phrase (NP) followed by a verb phrase (VP). This could be rewritten to state that a VP is a sentence with an NP missing:

$$VP := S\text{-}NP$$

From the order of nodes on the rhs of the PS rule, it can be seen that the NP occurs to the left of the VP, and consequently the NP is also missing from the left of the sentence. CG uses the *slash* operator to indicate the location of the missing node: forward slash '/' indicates that the node is missing to the right, backslash '\' to the left. The VP rule may

be now rewritten using the slash operator, presenting the CG syntactic category for intransitive verbs:

VP := S\NP

Using CG syntactic categories, the sentence ‘John likes cake’ is parsed, using forward and backward functional application, as follows:

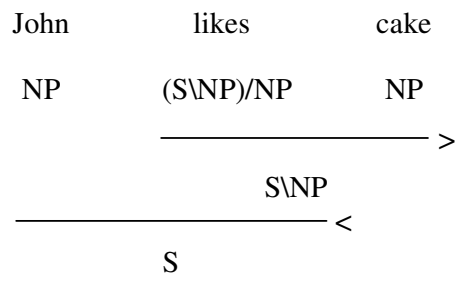


Fig. 4.1 CG Syntactic parse.

Here, the transitive verb ‘likes’ is given the syntactic category (S\NP)/NP, which states that the verb is a sentence with an NP missing to the right (the object) and an NP missing to the left (the subject).

4.6.2 Semantics

A logical representation of a syntactic category is provided by an associated lambda expression that defines the ways arguments are bound into logical forms. Thus the transitive syntactic category (S\NP)/NP may be associated with the semantic category:

λo.λs.v’(o s), where the lambda-variable o represents the verbal objects, s the subject, and v’ a literal verb.

Using a colon separator, the syntactic category and semantic interpretation of a transitive verb may be written as:

(S\NP)/NP:λo.λs.v'(o s)

The sentence above may now be parsed both syntactically and semantically, producing an output in logical form:

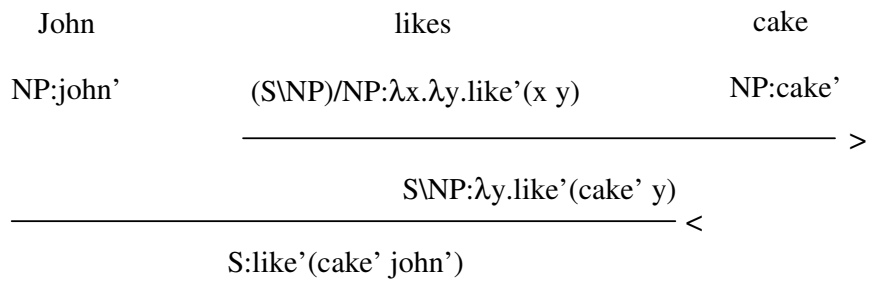


Fig. 4.2 CG Syntactic and semantic parse.

The semantic derivation obtained reveals the logical relationships between predicate and arguments; john' is the verbal subject, cake' is the verbal object.

4.6.3 Combinatory Rules

CG proves grammatical correctness of a given text by unifying the constituents obtained at the surface-level through application of combinatory rules, such as the typical set shown in Table 4.1:

Rule name	**Argument 1**	**Argument 2**	**Result category**
Functional application (>)	X/Y:f	Y:a	X:fa
Functional application (<)	Y:a	X\Y:f	X:fa
Coordination <Φ>	X:g CONJ:b X':f		X'':b(f)(g)
Forward composition (>B)	X/Y:f	Y/Z:g	X/Z:λx.f(gx)
Backward composition (<B)	Y\Z	X\Y	X\Z
Subject type-raising (>T)	NP:a		T/(T\NP):λf.fa
Subject type-raising (<T)			T\(T/NP) :λf.fa

Table 4.1 Typical rules of syntactic and semantic category combination.

The combinatory operators control constituent building through pattern-matching of categories. Application allows constituents to be derived from combination of functions and arguments, Composition and Substitution allows constituents to be derived from non-traditional constituents, such as functions, and Type-raising transforms an argument into a function, thereby permitting composition and substitution.

4.6.4 The parsing process

A CG parse generally proceeds by shifting words, one at a time, into a CKY chart parser [KASA65], [YOUN67], after having first assigned appropriate syntactic and semantic categories to the current word through lexicon-lookup. As each word - and its categories - is shifted-in, attempts are made to build new constituents from the current and previously shifted categories through application of the combinatory rules above.

4.7 CG Compatibility with the Coalition Model

To show that CG is compatible with the coalition model, and hence is consistent with a cognitively viable system of grammar it is necessary to compare it with the main elements of the coalition model as outlined in Section 4.5 above, and to the Principles and Parameters theory:

4.7.1 Sensitivity to input elements and their arrangement

CG does indeed express sensitivity to input type in that it uses the simple alphabet N, NP, PP and S, termed *atomic categories*. *Complex categories*, i.e. those constructed from atomic categories, limit the acceptable sequences of categorised input elements, for example, intransitive, transitive and ditransitive verbs have the forms S\NP, (S\NP)/NP and ((S\NP)/NP)/NP respectively. It is not necessarily the direct ordering of

the constituents within these categories, but the fact that the transitive category contains an embedded intransitive category, and that the ditransitive category contains an embedded transitive category, that we attribute to sensitivity to arrangement. Without sensitivity to arrangement of this kind, a grammar would consist of random arrangements such as S\NP, (S/NP)\NP, and ((S\NP)/(NP\NP) for the three verbal categories in question, which we propose would make the grammar cumbersome and difficult to learn and understand as it would be drawn from non-contiguous areas of the problem-space and would have, we believe, implications for incremental parsing, which will be discussed further in Chapters 8 and 9. Indeed, the Universal Alignment Hypothesis (UAH) of Relational Grammar [PERL84], which states that for any given language, initial semantic relations can be allocated on the basis of semantic roles, represents this kind of hierarchy. Although the UAH has been weakened in the light of work such as that of Rosen [ROSE84] who demonstrates the difference between *objective* and *culturally perceived* (i.e. subjective) assignment of semantic relations, it generally involves the hierarchy:

subject	direct object	indirect object	obliques
1	2	3	

4.7.2 Capable processes act on language units

CG describes the universal combinatory operators (Table 4.1) which may act on atomic and complex categories, which are assumed to be innate and common to all humans. The descriptions of the combinatory operators themselves deal with complex categories, for example, categories of forms X/Y, X\Y, and (X\Y)/Z, suggesting that complex categories are expected by the innate processing system. Through use of operators and complex categories, CG is capable of dealing with linguistic phenomena such as *anaphoric binding, auxiliary verbs, causatives, clitics, co-ordination, grammatical*

relations, inflexional morphology, intonation, long-distance dependency, modifiers and specifiers, nominal compounding, parasitic gaps, passives, raising, reflexives, relative clauses, switch reference, synthetic compounding, verb gapping and word order [WOOD93].

4.7.3 Principles and Parameters

CG is compatible with Chomsky's Principles and Parameters Theory in that the general elements it consists of, that is, the syntactic and semantic categories and combinatory rules, are applicable to all human languages, insofar as it has been shown to cope with Dutch, Dyrbal, English, Finnish, French, German, Hopi, Icelandic, Italian, Japanese, Korean, Luiseño, Malagasy, Maori, Russian, Spanish, Tairora, Turkish and Warlpiri texts [WOOD93]. As human infants are not genetically predisposed to any particular human language, the variety of complex categories permissible by the innate system must therefore include all human languages whatever their syntax scheme, and so the processor can not be limited initially in its categoric expectations to exclude any of these. Applicability to all human languages is a property of the Principles and Parameters Theory described by Chomsky [CHOM81], which describes *configuration* of an *innate* linguistic processor to *local linguistic environments*.

4.7.4 CG demonstrates configuration of innate language processor

Configuration can be demonstrated by parsing two parallel sentences, one English, the other German:

The sentence 'I like dogs' is parsed as shown in Figure 4.3 below. Using forward and backward functional application, the sentence is parsed, revealing 'I' as the subject of the sentence, and 'dogs' the object.

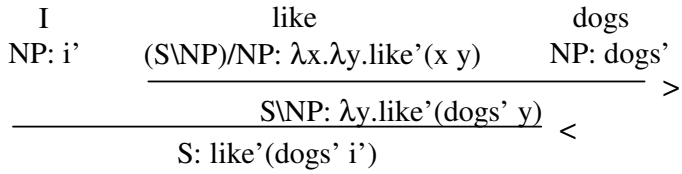


Fig. 4.3 CG parse of sentence 'I like dogs'.

The parallel German sentence 'Ich hunds möge' is parsed as follows:

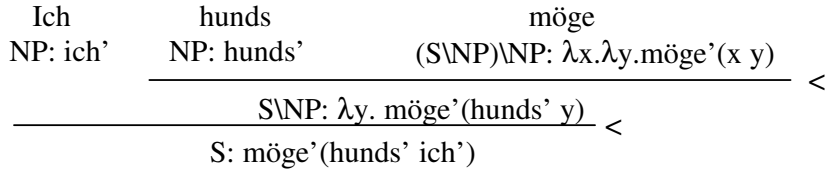


Fig. 4.4 CG parse of sentence 'Ich hunds möge'.

The logical forms are shown to be identical in structure, Verb(DObj Subj); the subjects (I, Ich) and direct objects (dogs, hunds) occupying the same argument positions with respect to their verbal predicates (love, möge). This is desirable as, according to Chomsky's theory, the logical form exists within the I-Language domain, is representational in nature, and is E-Language independent.

The above parallel sentences show that local configurations with respect to English and German occur in three places:

1. The syntactic form of the verb – $(S\backslash NP)/NP_{English}$ and $(S\backslash NP)\backslash NP_{German}$
2. The combinatory operator required to combine the verb and direct object - $>_{English}$ (forward application) and $<_{German}$ (backward application) - being necessary to accommodate the SVO and SOV structures of English and German respectively.

3. The surface form of the words used to occupy the predicate and argument positions – like/möge.

The above demonstrates that the same grammar-parsing scheme has been successfully applied to parallel English and German sentences, resulting in the same logical structure for each. The only real differences between the two are lexical – i.e. English and German words - and syntactic – different syntactic categories are required to accommodate the difference in syntactic expectation. Note however, no changes are necessary to the semantic categories or the combinatory operators; both English and German make use of forward and backward application.

If CG is taken as the E-Language syntactic/semantic processor responsible for generating logical-form I-Language representations of the E-Language, then some conclusions regarding the nature of the Language Facility may be drawn:

1. In its innate form, the language facility presents a small and finite set of combinatory operators, such as those presented in Table 4.1, and appears to utilise a relatively small set of semantic categories

2. In order to accommodate all human languages, the innate language facility must be capable of dealing with a potentially infinite set of (complex) categories built upon a very small set of primitive categories, such as N, NP, PP and S.

3. Configuration of the innate facility to some local environment consists of the selection of some (complex) syntactic categories over others from those available (e.g. selecting (S\NP)/NP or (S\NP)\NP as the syntax for transitive verbs), equating to the local syntax scheme.

4. Configuration to a local environment includes the mapping of local word-forms onto the semantic and selected syntactic categories.

4.8 Conclusions

In this chapter it has been proposed that the major initial components of an automatic text comprehension system consist of conversion of text from surface to logical form, sense assignment to the predicates and arguments of the logical form, and the seeking of coherence between elements of the logical form. From this, it has been demonstrated that, as *sense* is an important factor in the logical-form transformation and of coherence determination, the transformation, sense-assignment and coherence cannot be treated as separate processes or attributes. It s proposed that these elements can become mutually constraining if the processes of the CIM are extended from the I-Language into E-Language domain, and augmented by a grammar parser to reveal possible relations. As only the human language system is available for study, a cognitively valid grammar system in the form of CG was selected for this component, which was then shown to be consistent with the Coalition Model, and configurable as expected by the Principles and Parameters theory of grammar acquisition.

5 The Chunking Element

This chapter proposes that the decision of whether a sequence of words represents individual words or a compound word requires grammatical analysis. It also argues that the chart parser should not have the responsibility of constructing compounds itself on grounds of the non-compositionality of many compounds, and consequently compounds are recognised prior to shifting into the chart. Compounds and phrases can be detected by chunking, and justifications for the inclusion of a Chunker in a psychological model of text comprehension are presented. Chunking is shown to have positive advantages in that it effectively reduces the number of terms input into the chart, and allows the N and NP atomic categories to be merged. However, it is also shown that compound words and their constituent words must be evaluated in parallel, which is addressed by the introduction of a novel parallel-shift enabled chart structure.

Parsing methodologies employing bottom-up strategies, such as CG, are commonly constructed around *Chart Parsers*, which typically use the CKY [KASA65];[YOUN67] or Earley [EARL70] algorithms. The impetus for using such schemes is that they eliminate the *backtracking* responsible for the exponential complexity of such parsing methods; a chart parser records all parse-tree edges as they are discovered, and does so only once, whereas standard parsing techniques, when encountering an insoluble subgoal, will 'unwind' all work done since the point of the most recently solved goal. Subtrees may be traversed many times before a soluble rule is found. Note however that the CKY algorithm comprises three nested loops giving an algorithmic complexity of

N^3, where N is the number of words to be parsed, leading to extensive processing effort for longer sentences.

To summarise the chart parsing process, a chart parser mechanically discovers all edges licensed by the grammar over the words of the sentence by application of a two-step procedure. Firstly all appropriate categories are assigned to the sentence words by lexicon lookup [HOCK03]. Secondly, the categories are combined using the CG combinatory rules. Complete parses are identified by edges that span the entire sentence.

Using this approach, the parser calculates all possible combinatory fragments and complete parses of the given sentence, and so expends effort unnecessarily building not only correct parses, but also complete but incorrect parses as well as constituent fragments which ultimately do not feature in any complete parse, a situation first noted by [KAY96].

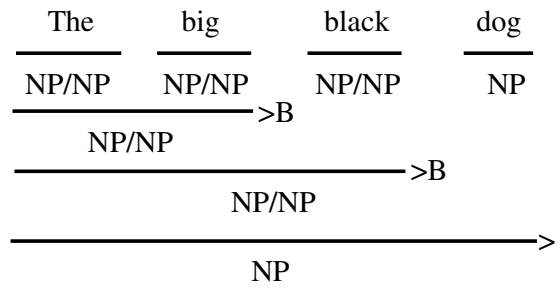
The situation is complicated by *Spurious Ambiguity*, which manifests itself as a set of alternative but 'equivalent' constituent derivations. Take for example the noun phrase 'The big black dog', which requires the following lexicon expressed using CG notation (Sections 4.6.1 to 4.6.3).

{The : NP/NP}
{Big : NP/NP}
{Black : NP/NP}
{Dog : NP}

To parse this sentence a further combinator, *functional composition*, (B), is needed, which is specified as

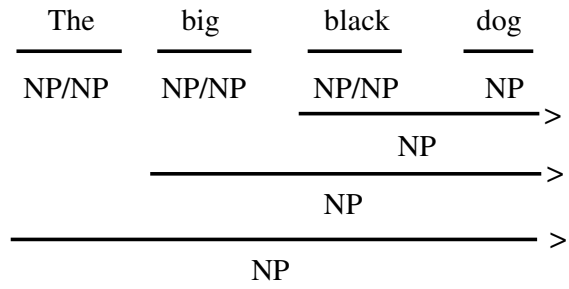
i.	X/Y	Y/Z	⇒	X/Z	(>B)
ii.	Y\Z	X\Y	⇒	X\Z	(<B)

Using the lexical categories with just forward composition and forward application, the noun phrase can be parsed in five ways. The first is obtained by the following derivation:



resulting in: i. ((The big) black) dog $_{NP}$.

The second, The (big (black dog)) $_{NP,}$ is obtained as follows:



Similarly:

iii. The ((big black) dog) $_{NP}$
iv. (The (big black)) dog $_{NP}$
v. ((The big)(black dog))$_{NP}$

Technically, the right branching derivation: 'The (big (black dog))' is the only correctly structured parse, but all resultant derivations have the same syntactic category of NP; the parser has done a lot of unnecessary work to determine that 'The big black dog' is a noun-phrase. That one derivation may be arrived at via alternative parses is an expression of *Spurious Ambiguity*.

Spurious ambiguity also features when processing *compounds*, these being "a habitual co-occurrence of individual elements" [CRYS85], which can be subdivided into *lexical phrases* (Computer programmer, Washing machine), *idiomatic phrases* (an eye for an eye, Beat about the bush), and *proper nouns* (New York, George W. Bush). Verbal compounds are also possible (send for, pick up). A compound, using more than one term to represent one object or action, poses a problem for the parser in that, using the standard method described above, each of its component terms is assigned a category from the lexicon as it is shifted in to the chart; no single category is assigned to the compound as a whole, leaving the parser to identify the compound through mechanical application of rules as shown above. For example, the individual terms of a proper noun will each receive categories, assuming the lexicon contains plenty of names:

{George : NP, NP/NP }
{W. : NP, NP/NP }
{Bush : NP, NP/NP }

Again an NP will result, expressed by the two bracketings ((George W) Bush)$_{NP}$ and (George (W Bush))$_{NP}$, but redundant processing has occurred.

The lexical phrase 'washing machine' is more complex in that 'washing' may be a verb or a gerund, resulting in the assignment of both verbal and gerundive categories from the lexicon and an increase in the processing load and time.

Idiomatic phrases may or may not be parsed correctly by assignment of categories to individual terms, depending on the representation of the constituent terms in the lexicon. For example, the first two sentences below have the same meaning, and the underlined constituent is parsed to give the category S\NP in each case.

The suspect <u>gave information about the crime.</u>$_{S\backslash NP}$

The suspect <u>spilled the beans.</u>$_{S\backslash NP}$

The same is not true of the following pair of sentences, and the idiom 'a piece of cake' cannot be assigned an adjectival category to match that of 'easy', although this is most likely due to the behaviour of the verb 'to be', which takes the syntactic categories VP/AP and VP/NP[1] [KEEN88]:

The task will be easy.$_{N/N}$

The task will be a piece of cake.$_{NP}$

Also, being non-compositional, no semantic information about the idiomatic phrase 'a piece of cake' may be gained from any of the constituent terms. Idiomatic phrases may therefore lead to a parse failure, or to incorrectly structured parses. In any case, processing effort is again wasted.

Attempts have been made to reduce the amount of work the parser has to do: Eisner proposes a constraint whereby a constituent resulting from forward or backward composition is disallowed from being the primary functor in another forward or backward composition or application [EISN96]. The number of categories assigned by the lexicon to a word may be limited both by assigning only those categories that occur more than a given frequency (Clark uses a cutoff of 10) in the lexicon training data [CLAR02]. Supertagging, using PoS features over a five-word window, has also been used to reduce the number of assigned categories with little loss of coverage and an improvement in performance [CLAR02]. Additionally, allowing categories to combine by a combinatory rule only if that particular combination of categories and rule has been seen in the training data (sections 2-21 of CCGBank) reduces the number of combinations performed [HOCK03].

[1] Where VP is the verb phrase S\NP, AP is the adjective phrase N/N, and NP is a noun phrase.

The above approaches operate by either eliminating unnecessary category/combinator combinations, or by assigning the most probable categories from the lexicon to a term. None addresses the fundamental problem posed by multi-term words and phrases. Possibly this is because it is difficult to reconcile a system that appears to work incrementally with a need to process grouped terms as found in compounds.

5.1 Chunking

Sentence 1 below consists of 8 terms, each of which would normally be shifted into a chart parser individually:

The big black dog chased the ginger cat (1)

However, by identifying phrasal chunks, the sentence is reduced to just 3 terms (2).

$(\text{The big black dog})_{NP}\ (\text{chased})_{(S\backslash NP)/NP}\ (\text{the ginger cat})_{NP}$ (2)

Considering the CKY algorithm's complexity of N^3, the computational saving is readily apparent.

The chunking of sentences in this way was initially proposed as a shallow parsing strategy by Abney [ABNE91], is often used to provide robust parsing in Information Retrieval and Terminology Extraction applications [GREF92], [APPE93], and is commonly accomplished by recognition of syntactic patterns in PoS-tagged text by application of finite state recognisers [ABNE91].

Parse trees may be constructed from identified chunks, for example, through application of separate processes to the chunks [EJER83];[ABNE91], or by application of a Maximum Entropy tagger [RATN96] trained on a subset of the Penn-Treebank [MARC93] to successively identify base chunks in the current parse state. Memory

Based Learning techniques [TJON02], Hidden Markov Models [MOLI02], and PoS taggers [MEGY02] have all been used to induce chunking rules, complementing the regular expressions over PoS tagged texts method used by Abney [ABNE90], [ABNE91].

Chunking has recently been applied to chart parsing, along with Index Filtering and n-gram based Edge Pruning, together demonstrating that sentences can be ***syntactically*** parsed in approximately one second whilst exhibiting less overgeneration.

Abney [ABNE91] suggested that Chunking is computationally less expensive than full parsing. This can be demonstrated intuitively by consideration of the complexity of phrase recognition. As has been stated above, the complexity of the CKY chart-parsing algorithm is N^3. The complexity of a Chunker may be stated as N * P, where N is once again the number of input terms, and P the number of patterns that may be recognised. Trivially, both methods require the same number of computations to recognise a string of length N when $P = N^2$. So, to recognise a noun-phrase of length 2, eight chart computations can be performed. To recognise the same phrase by chunking, four patterns are permissible before chunking becomes more expensive. However, only three patterns are necessary to recognise a two-term noun-phrase, these being (Noun, Noun), (Determiner, Noun), and (Adjective, Noun). The situation is further improved by both the exponential growth of the number of permissible patterns necessary to recognise phrases of increasing length, and the fact that *regular expressions* can economically represent a number of patterns as a single expression. For example, the regular expression

[Determiner]?[Adjective]*[Noun]+

is a single pattern as far as complexity is concerned, but recognises typical noun phrases of length 1 to n, including:

dog, dogs, the dog, old dog, the old dog, the big black dog, the dog food, …

If P is taken not as first stated the number of patterns to be recognised, but instead the number of *pattern recognisers*, as evidenced by the regular expression above, then it appears likely that P will always be less than N^2, and that the intuition regarding the economy of chunking over full parsing is correct.

Although chunking provides a useful reduction in the number of terms shifted into a chart parser, does it have any place in a psychologically-oriented language processor? Specifically, is there any mechanism by which the human equivalent of the chart parser receives chunks rather than individual, space-delimited terms? In order to answer this question, it is necessary to examine the human visual system, which lies between the printed page and the parsing system discussed so far.

5.2 Justification of chunking in a psychological model

5.2.1 Visual Acquisition

The eye, when reading, does not scan smoothly from left to right across a line of printed text, but proceeds by an alternating sequence of *fixations* and *saccades*. A fixation

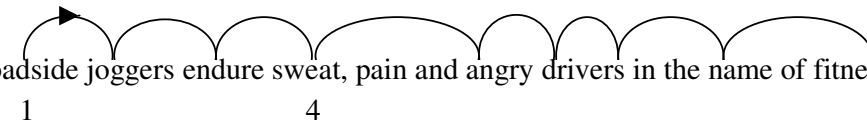


Figure 5.1 Saccadic eye movements. From [LARS04]

occurs when the eye stops moving and fixes focus upon a point on the page; a saccade is a rapid movement of the eye from one fixation to another point of fixation.

Not every word in a text receives a fixation – the eye appears to use its acuity to acquire a number of words in one 'gulp'. The *Moving Window Paradigm* is a technique that allows the number of characters taken in during a fixation to be measured through use of an eye-tracker. The eye tracker measures precisely the subject's point of fixation on text displayed on a computer screen, and software corrupts all text beyond a given distance from that point of fixation [MCCO75]. When the subject's eye fixates after a saccade, the display is updated to corrupt all text except that around the new fixation point. Figure 5.2 illustrates the fixation point, moving window, and corrupted display of the technique. Selecting too small a window, that is, corrupting text too close to the fixation, degrades reading speed and comprehension, whereas selecting an overly large window has no such effect. The boundary condition was found when the *foveal + parafoveal information* (i.e. the window) comprised 4 characters prior to and 15 characters following the fixation point (the fixation point providing the *foveal information*) left uncorrupted; reduction in window size beyond this, on either side of the fixation, resulted in reduced reading speed and comprehension whereas an increase beyond these sizes had no effect. (Interestingly, for right-to-left readers, the 15-character window extends to the left and the 4-character to the right of the fixation point.) This result suggests that when reading, visual information is acquired as a series of images, each image containing up to 20 characters.

Additional work has identified three zones on the visual field: the zone closest to the fixation is usually large enough to encompass the entire fixated word and smaller closed-class words to the right of the fixation, and is where word recognition occurs.

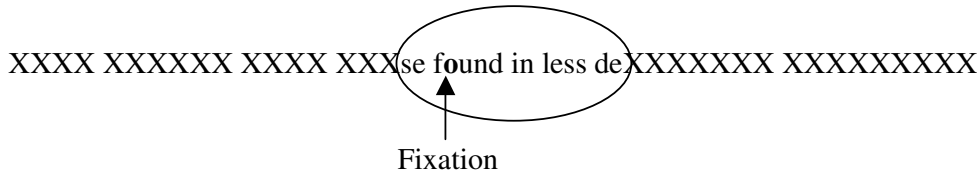


Fig. 5.2 Simulation of the Moving Window Paradigm visual field.

The second zone covers a few letters immediately following the recognition zone, and provides a look-ahead, gathering information about the beginning of the next word. The third zone includes the remaining characters to the right of the fixation, and is thought to give information about the lengths of the next words, which is then used for planning the location of the next fixation. Figure 5.1 shows that, for example fixation 1 recognises the word 'Roadside', and acquires the first two letters and word length of the word 'joggers', whilst fixation 4 recognises the short words 'sweat' and 'pain', including the intervening comma, and acquires the initial letters of the word 'and', which being a high frequency closed-class word, is enough to allow 'and' to be skipped entirely.

Words can then be extracted from the visual image in different ways: The *Parallel Allocation Hypothesis* [ERIK86, HEND91, ENGB02, REIL04] proposes that attention is allocated to the foveal and parafoveal word(s) in parallel during a fixation, whereas the *Sequential Allocation Hypothesis* [HEND88, FERR90, POLL90, MORR84, REIC98] allocates attention sequentially to parafoveal words only when foveal processing is complete. Later work [HEND95] supports the Sequential Allocation Hypothesis, but recognises the benefits afforded by preview of parafoveal words, that is, by Parallel Allocation. Engbert [ENG04] unifies these two hypotheses by proposing that words are processed in parallel, but due to decreasing visual accuity at greater distances

from the fovea, the fixated word is processed fastest. It follows that the most extreme parafoveal word will be completed last, and the overall impression is of sequential processing.

5.2.2 Word Recognition

Looking more closely at word recognition, three explanatory models have been proposed: The Word Shape model assumes word recognition occurs on the basis of overall word shape, or *Bouma*, and was first proposed by Cattell [CATT86]. The Serial Letter Recognition model assumes that reading occurs letter-by-letter from left to right, each letter subdividing the lexical search space further until the word is recognised. The Parallel Recognition Model assumes that letters from a word are recognised in parallel, and is now the favoured model as it explains phenomena, such as the Word Superiority Effect, which states that letters in the context of a word are more quickly recognised than in isolation, which is impossible to explain in terms of the Serial Recognition Model, and although initially predictable by the Word Shape Model, is not sustainable in the light of evidence obtained by manipulation of the shape and letter combinations of words and pseudo words in test sentences [MCCL77].

As the Parallel Recognition Model best fits the facts, perhaps it can provide a justification for chunking text prior to shifting into a chart. Figure 5.3 (from [LARS04]) depicts the Parallel Recognition Model, showing how parallel recognition of letters leads to word recognition.

From the stimulus, feature detectors analyse the visual image, resulting in sequentially arranged recognised letters, and each word in the lexicon is activated for each

corresponding letter position. In Figure 5.3, FORK and WORD each receive three activations, but WORK receives four activations and so becomes the recognised word.

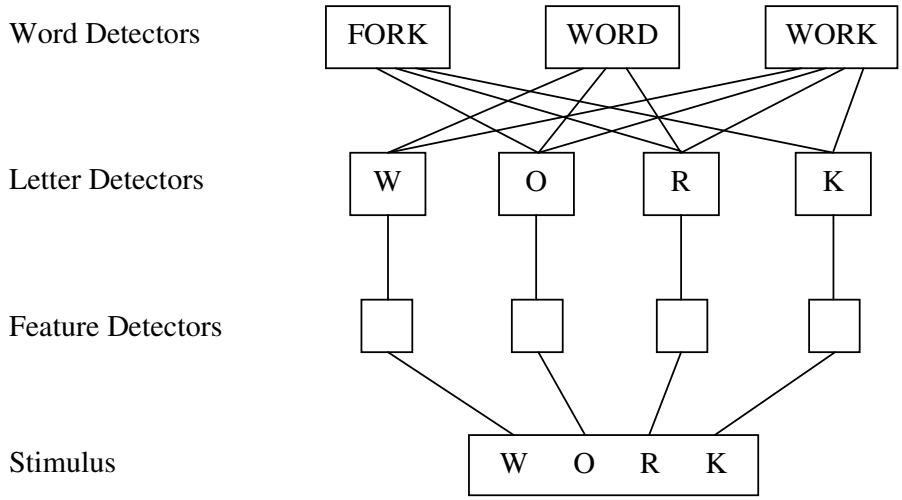


Fig. 5.3 The Parallel Recognition Model

As an interesting aside, the Parallel Recognition model is able to explain many of the impressive party tricks the visual/recognition system is capable of (although phonological and contextual information will also play a part), for instance recognising words using only the top half of the text. Sntncs wtht vwls cn b rd, and so can th.se wi.h eve.y fif.h let.er mi.sing. Similarly every fifth word **...** be omitted from a **...** without losing the meaning [RUSS79].

Using the Parallel Recognition Model, it is possible to show how compounds can be recognised. Consider the system depicted in Figure 5.4.

Presenting the visual system with the words ‘washing machine’, letters are detected and activate lexical entries with letters matching in corresponding positions. The lexical entry ‘washing’ is fully activated, and even ‘machine’ is partially activated, having four

positional character matches with ‘washing’, although one would expect the mismatching letters to inhibit the activation somewhat.

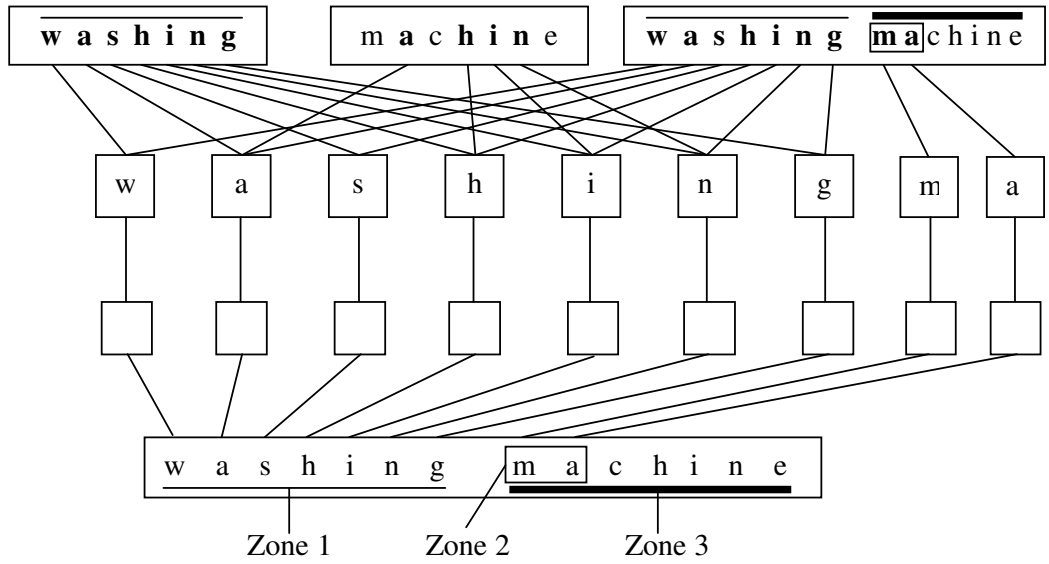


Fig. 5.4 Parallel Recognition of a compound

If it is accepted that entities in the real world have a conceptual representation in the mind, and that this representation includes the linguistic label associated with the entity, then it is reasonable to assume that, as a washing machine is a real-world entity, its linguistic label ‘washing machine’ will exist in the lexicon also. In this case the entry ‘washing machine’ will also become activated; although it receives the same number of activations as ‘washing’, it is not fully activated.

However, the letters ‘ma’, the initial letters of ‘machine’ have been acquired by zone 2 of the visual system and further activate the entry for ‘washing machine’. Upcoming word length gained from zone 3 may also provide some degree of activation by matching the length of the stimulus ‘ma-----’ with the expected continuation of the partially activated ‘washing machine’.

Of course, ‘washing machine’ is not necessarily a compound, as shown by the pair of sentences (3) and (4) below:

Washing machines are expensive. (3)

Washing machines is my job. (4)

The simple/compound decision appears to be made on the basis of verbal number agreement in this case; up to the point where the verb is integrated into the sentence representation, ‘washing’ + ‘machines’ and ‘washing machines’ are both potential interpretations. From this it can be expected that the parallel recognition model will output the following in parallel:

1. washing
2. machine
3. washing machine

What can be taken from this discussion is that within the most likely model of word recognition there is scope for actual recognition of compounds in their entirety, that is, before being shifted into a parser, and so the incorporation of a chunking component in a pre-processor to the chart parser would seem to be cognitively viable.

5.2.3 Evidence for Chunking from a garden path sentence

Garden Path sentences have been used to demonstrate early plausibility filtering during parsing [BEVE70], [WINO72], [HIRS87], [CRAI85], [ALTM88]. Given the pair of famous sentences below it is possible to explain why sentence (5) exhibits the garden path effect whilst sentence (6) does not.

The doctor sent for the patient arrived. (5)

The flowers sent for the patient arrived. (6)

The accepted explanation is that, as 'flowers' cannot 'send for' things, the interpretation in which the flowers are doing the sending is eliminated early, resulting in the correct interpretation of the sentence at the first attempt. 'Doctors', on the other hand can 'send for' things, and in particular they might be expected to 'send for' 'patients', and so this interpretation is initially preferred. It is only at the disambiguating term *arrived* that the error is noticed and backtracking occurs. Although we shall return to this point in Section 5.4 and in Chapter 8, there is another aspect to these sentences: in sentence (5) the verb is a compound (send_for) whereas in sentence (6) it is not. (send). Note also that in sentence (5), 'doctor' is the subject of the verb 'sent_for', whereas in sentence (6) 'flowers' is the direct object of the verb 'sent'. The truth table below (Table 5.1) shows the result of applying a pragmatic filter to the verbs and arguments of the two sentences:

X	X send	send X	X send_for	send_for X
doctor	true	true	true	true
flowers	false	true	false	true

Table 5.1 Truth table showing viability of doctor and flowers as verbal arguments.

The table shows that the only rejections on grounds of plausibility occur when 'flowers' is the subject of either verb; flowers cannot send things or send_for things. More interestingly, it also shows that 'doctor' is suitable both as subject or object for either verb. When applied to sentence (5) above, this information prompts consideration of why the verb 'sent_for', requiring the construction of a compound and ultimately ending in parse failure, was recognised in preference to the simple verb 'sent', as a parallel interpretation to sentence (6) which would have resulted in a successful parse when, as the truth table shows, both verbs are possible. The explanation we offer is that,

as doctors typically send_for patients, the noun ‘patients’ influences the recognition of the verb, selecting ‘send_for’ over ‘send’ as in this context it is the most plausible interpretation. In order for this test to occur, both ‘sent’ and ‘sent_for’ must be evaluated against each other, requiring that the compound ‘sent_for’ be constructed, recognised, and shifted into the parser in parallel with the simple verb ‘send’, resulting in a *beam search* of the possibilities at the next synchronisation point, that is, when both paths once again accept the same word, i.e. ‘patient’:

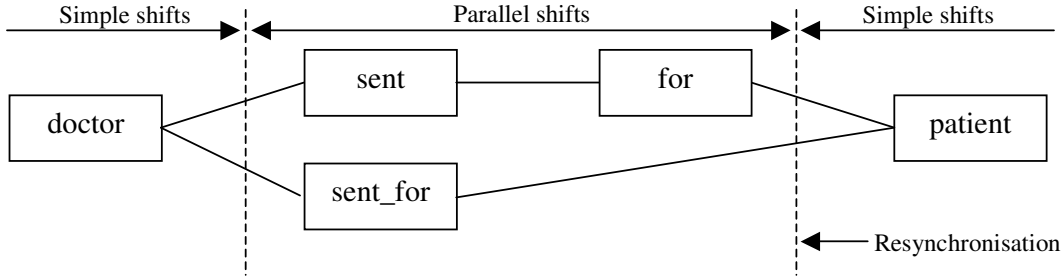


Fig. 5.5 Parallel paths required to parse ‘the doctor sent for the patient…’

The above shows that there is a requirement to evaluate the alternative groupings of words in order to select one for continued processing. It has also been shown that the grammar parser itself cannot be guaranteed to combine elements of a compound to give a category for that compound. From this we conclude that some other mechanism must be at work performing this task and providing the alternatives, and that it must do this before the shift into the chart. The non-compositionality of some compounds requires that they be represented in their entirety, either directly in the lexicon as would be the case for idioms and compounds like ‘kick the bucket’ and ‘washing machine’, or as patterns or templates that allow proper nouns like ‘The Institute for Interesting Science’, numerical quantities like ‘300000 kps’ etc. to be recognised.

The visual system, being able to acquire a number of word images at one fixation, is therefore compatible with a language that employs compound words and deals with them as distinct 'entities'. The images provided contain one or more words that can be recognised by the lexicon both individually and as a full or partial compound.

5.3 Quantification of work reduction through chunking.

In the introduction to this chapter, the theoretical benefit of chunking text before shifting into a chart parser was described; the cubic relation between the number of words input and the number of operations involved means that great savings in processing effort may be obtained by reducing the number of input terms. Here, an estimation of processing effort as number of operations performed is obtained for raw and chunked sentences.

To estimate the average reduction in sentence length caused by chunking a sentence, the Chunker described by Abney [ABNE96] was applied to SemCor [FELL98]; comparing the resulting number of chunks to the original number of words allows the number of operations required to parse the sentences to be estimated.

To obtain a true word count from SemCor, the pre-detected compounds it contains were first replaced by their constituent words. This resulted in a total of 378743 words in 186 documents, with sentence lengths ranging from 1 to 240 words.

The Abney Chunker produced 234339 chunks using the same documents, resulting in chunked sentences on average 61.4% of the size of the original corpus.

Corpus	**No. of Words**
SemCor	378743
Chunked SemCor	234339
Relative size	61.36%

Table 5.2 Word counts from raw and chunked SemCor.

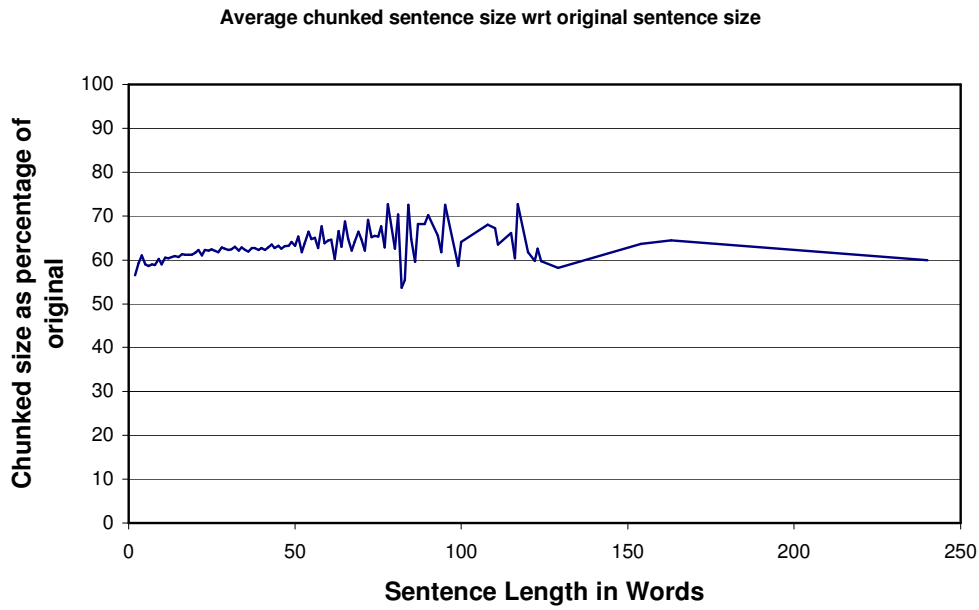


Figure 5.6. Reduction in size due to chunking for sentences of given length.

The graph of Figure 5.6 presents a relatively smooth line up to a sentence length of 64 words after which it becomes increasingly eccentric. This is to be expected, as 96.89% of the words (366965 words) are drawn from sentences of length 64 words or less - after this point the lack of data points leads to an increasingly noisy signal.

5.3.1 Results

Prior to chunking, the average sentence length calculated from the 378743 words in the 20138 SemCor sentences was 18.81 words per sentence. After chunking, the average falls to 11.64 chunks per sentence. The number of basic parser iterations for the average sentence length for SemCor and chunked SemCor can be calculated as N^3, corresponding to 6655.28 and 1577.10 operations respectively, reducing the number of operations by 76.3% and representing a very useful reduction in parsing effort required.

5.4 A proposal for a parallel-shift enabled chart parser

A basic implementation of such a system would involve cloning the chart to date, creating one instance for each of the alternative shifts. Later plausibility testing would then destroy the less viable chart(s), resulting in the single most favoured chart. However, as working memory in the human cognitive system is subject to limitations, this approach would appear unsustainable, and a more economical solution must be sought. A modification to the chart structure, involving a temporary increase in the number of dimensions of a chart column, provides just such a solution, and enables *reactivation*, a current research thread from the Neural Processing community [GURN04].

Consider the state of a chart having three terms previously shifted into it, with a fourth about to be shifted, as shown in Figure 5.7.

The standard chart arrangement shown above has no mechanism for accepting parallel shifts; each shift resulting in the creation of a new chart column that represents the next word from the sentence.

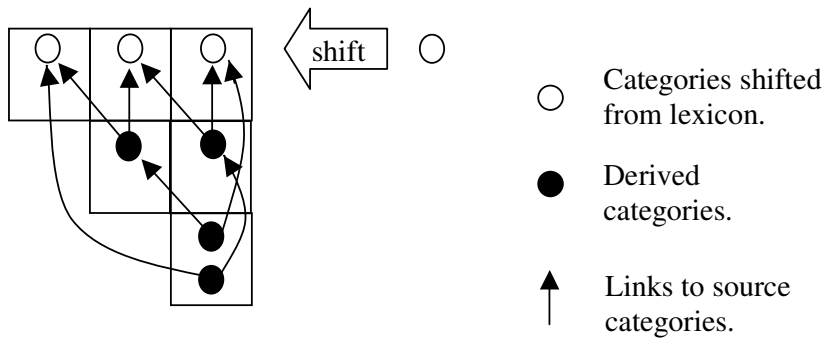


Fig. 5.7 Chart with three terms shifted.

If a chart is to accept parallel shifts, each item shifted must be tagged to indicate its start and end position in the sentence, where space-delimited words are numbered incrementally from one from left to right. The start and end positions, or *extent*, can be indicated by two digits, for example, a single space-delimited word would be assigned the extent x-x, whereas a compound would receive the extent x-y, where x and y are the start and end positions respectively. So, for a sentence such as 'Say goodbye to washing machine noise', the shifts and their extents would be:

(1-1 say) (2-2 goodbye) (3-3 to) (4-4 washing) (5-5 machine) (4-5 washing machine) (6-6 noise)

By associating an extent with each column it becomes obvious when a parallel shift occurs as the incoming extent overlaps with already shifted terms, and so gives the signal to increase the dimensionality of an existing column. Using the above sentence, the chart behaves as normal for the first five shifts as the extents are sequential. When the sixth shift (4-5 washing machine) occurs however, its start position is not greater by one than the end position of the last shift, and the start position of the sixth shift is used to identify the chart column to which a new dimension, and hence a parallel column, must be added to accommodate the new shift. The chart has effectively branched at position 4 to allow parallel paths to be followed.

The next shift (6-6 noise) is the next in sequence expected by both chart dimensions, as the end position indicated by both of them is 5. At this point the parallel chart paths converge again. This situation is illustrated by Figure 5.8, which for clarity shows incremental derivations only. Note that the (reversed) pointers to each derivation's source categories (shown as arrows) provide a means to identify the parent path of any

derivation, for example the derivation labelled ‘a’ is following the compound path, whereas ‘b’ is following the single word path.

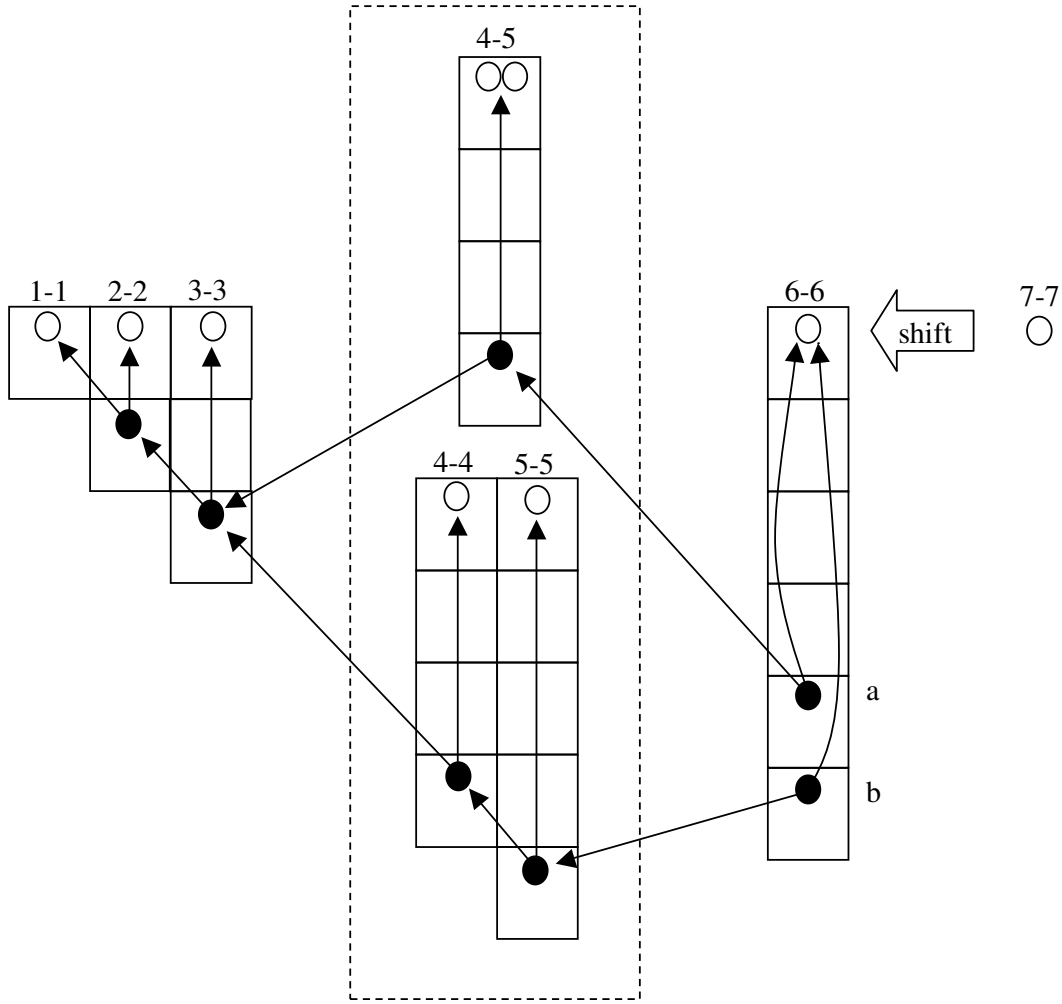


Figure 5.8 Parallel-shift enabled chart showing increased dimensionality of column 4 and convergence at column 6.

After convergence, memory constraints dictate that one of the paths must be selected, following the *Principle of Parsimony* [CRAI85], [ALTM88] which states that:

> The analysis whose interpretation carries fewest unsatisfied but accommodatable presuppositions or consistent entailments will be preferred. [STEE00]

Any unselected columns are deactivated, which may be achieved simply by setting a flag associated with each column; when set, the flag disallows that column from

participating in future derivations. Derivations obtained after convergence must also be deactivated by flagging, and may be located by following links back to the deactivated column(s), resulting in the chart shown in Figure 5.9.

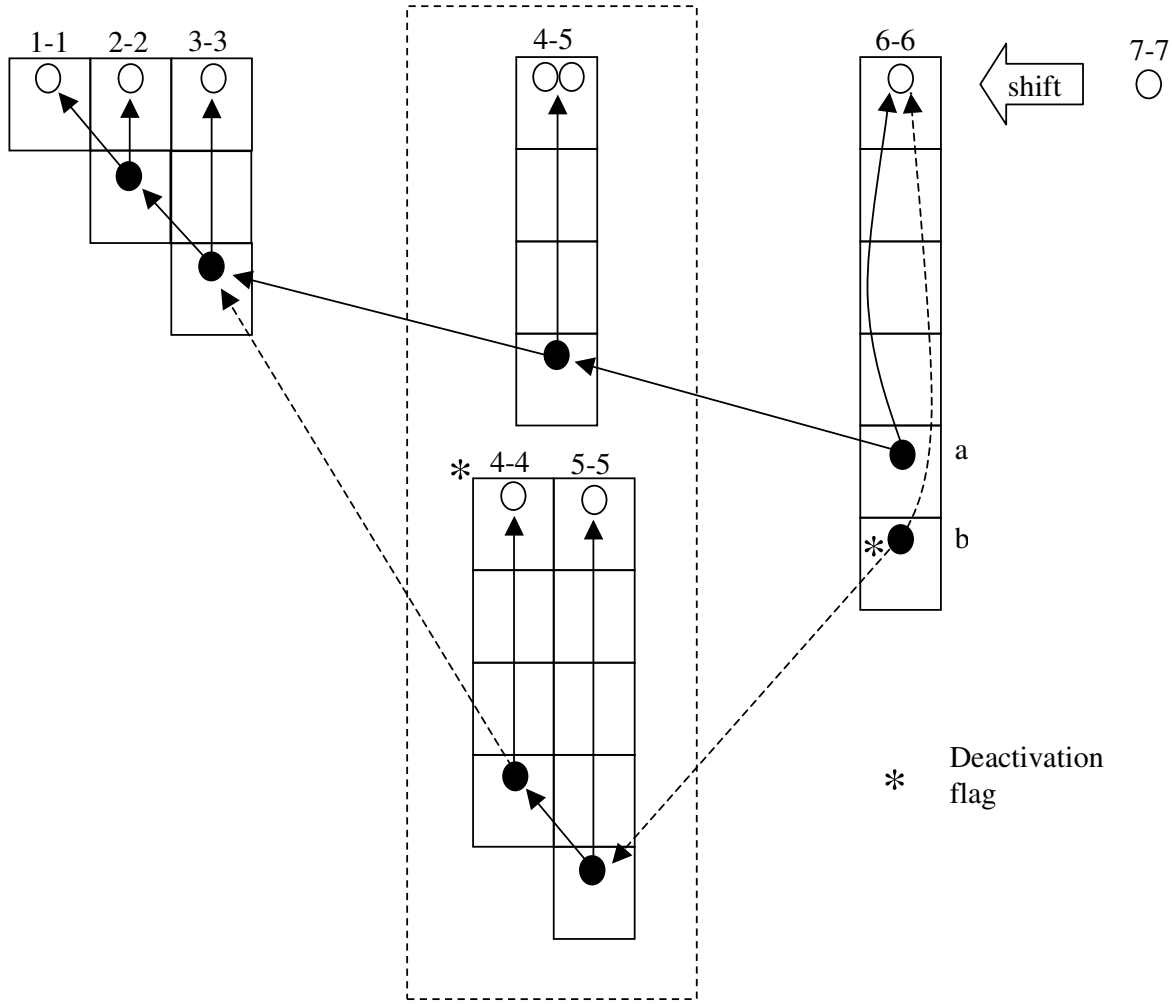


Fig. 5.9. Chart after column deactivation.

We see column and derivation deactivation as analogous to moving information out of working memory; although the deactivated chart elements have not actually moved anywhere, they are no longer in a position to use any cognitive resources.

Keeping deactivated chart elements in situ is also advantageous when the parser is required to backtrack to repair an error made in the selection of the most plausible path

(i.e. the selection of the wrong parallel column), as might be the case in garden path sentences: firstly, the columns in which the error was made are identifiable by their multi-dimensionality, the human equivalent being a regressive saccade in which the eye jumps backwards to some earlier position in the sentence, allowing all or part of the sentence to be re-read from that point. Secondly, the deactivated columns are present and able to participate immediately in a re-evaluation of the path selection. Should the re-evaluation result in a different parallel path being selected, the new most plausible column can be reactivated, along with any existing derivations it is involved in, and the previously favoured column and its derivations deactivated.

5.4.1 Impact of Parallel-Shifts on performance

The introduction of Parallel Shifts will of course increase the workload of the parser. Take for instance the sentence represented by Fig. 5.5. As Parallel Shifts allow potential compounds of different numbers of terms to coexist in the parser, an initial estimate of workload may be obtained from the parser N^3 complexity, and by examining each path in turn. The upper path has four terms and so requires 64 operations to fully complete the parse, whereas the lower path, having only three terms, requires 27 operations. This results in 91 operations to parse both paths. However, the Parallel Shift proposed here allows paths (via chart columns) to be deactivated on the basis of the Principle of Parsimony as shown in Section 5.4. In the best case (i.e. when no reactivation is necessary) this results in the abortion of the processing of one of the parallel paths and a consequent reduction to less than the full 91 parsing operations required. As the sentence and its set of individual and possible compound terms will affect the overall number of operations performed, we can only state that:

Parallel-Shift parser complexity >= N^3

This may at first appear to be a retrograde step. However, the pay-off is that the parallel-shifts enable the *senses* of both simple *and* compound terms to be represented in the parser. As will be demonstrated in Chapters 8 and 9, sense is a further constraint on the parsing process, the constraint resulting in additional eliminations of chart entries through plausibility testing and elimination of implausible derivations. This ultimately leads to sense and structural disambiguation of the input sentence. Further comparative work is required to determine whether parallel-shifts actually reduce the overall processing effort required however.

5.5 Merging N and NP categories, a justification

The incorporation of a Chunker into the parser has a useful consequence with respect to the *atomic* syntactic categories typically employed by CG. These *atomic* categories are N (noun), NP (noun phrase), PP (prepositional phrase), and S (sentence), and it is from these that all *complex* syntactic categories are built.

The consequence is that the N and NP atomic categories can now been combined, reusing the NP label for both. It is important to realise that we are not advocating the wholesale substitution of N categories by NP categories, as shown in Figures 5.10a and b below, as this dangerous action would allow grammatically incorrect sentences to be parsed, as illustrated by Figure 5.10c.

a) John is a big boy
NP (S\NP)/NP NP/N N/N N

b) John is a big boy
NP (S\NP)/NP NP/NP NP/NP NP

c) * John is big boy
NP (S\NP)/NP NP/NP NP

Fig. 5.10 Simple substitution of N by NP.

However, by employing a Chunker as a noun-phrase recogniser, the parser is relieved of the burden of building noun-phrases, and instead is only involved in integrating those noun-phrases into logical structures. This follows Abney's proposal for a *Chunking Parser* [ABNE91], comprising the two elements of a *Chunker* and an *Attacher* in which

> "*The chunker converts a stream of words into a stream of chunks, and the attacher converts the stream of chunks into a stream of sentences*."

The Attacher assembles chunks into full parse-trees through addition of missing arcs between chunks and parse-tree nodes, a role here filled by the CG parser. Due to the Chunker, the Attacher/CG parser never encounters bare nouns, only noun phrases, hence the global use of NP. Quantifiers are traditionally type-raised in the lexicon [STEE00] and are easily accommodated by a Chunker. The lexicon delivers syntactic categories such as:

Every := (T/(T\NP))/N	Every := (T\(T/NP))/N
Some := (T/(T\NP))/N	Some := (T\(T/NP))/N
Some := (T/(T\NP))/NP	Some := (T\(T/NP))/NP

It is a simple matter to arrange for the syntactic category of a chunked/recognised quantified noun-phrase to reflect the type-raising within the syntactic category of a chunked quantifier, as shown in Fig. 5.11 below. More complex quantified

$[\text{Some}_{(T/(T\backslash NP))/N}\ \text{boy}_{N}]_{T/(T\backslash NP)}$ $[\text{Some}_{(T/(T\backslash NP))/N}\ \text{boys}_{NP}]_{T/(T\backslash NP)}$

Fig. 5.11 Chunker-assigned type-raised quantified noun phrases.

constructions may be handled similarly, but may result in ambiguity that must be resolved further into the analysis. For example, the phrase 'some dogs and cats' most likely means 'some dogs and SOME cats', but it is possible that it could also mean 'some dogs and ALL cats'. Only by seeking the most plausible interpretation of 'some

dogs and cats' within the context of the sentence in which the phrase is presented can this ambiguity be resolved. The Parallel-Shift introduced above provides a mechanism for the evaluation of such interpretations, allowing both forms to co-exist within the parser.

However, by recognizing noun-phrasal chunks and assigning the NP category as described above, it once again becomes possible to parse sentence 5.10c above:

$$[\text{John}]_{NP} \quad [\text{is}]_{(S\backslash NP)/NP} \quad [\text{big boy}]_{NP}$$

Now, we believe it is important to acknowledge a distinction between a socially correct grammar and a grammar for comprehension: A 'correct' grammar might for example be used by a proof-reader as a 'gold standard' by which texts are judged and corrected, and will include prescriptive rules such as 'thou shalt not split an infinitive' and the like. A grammar for comprehension on the other hand is 'merely' concerned with the I-Language aspects of the grammar, that is, with the construction of plausible representational structures through selection from candidate predicates and arguments as presented by grammatical analysis. The sentence 5.10c contains sufficient grammatical conformity to allow comprehension in that the syntactically categorised chunks can be combined unambiguously into a logical form in which 'John' is the subject, 'is' a transitive verb, and 'boy' is the object, with the object modified by the adjective 'big'. This is particularly true if the chunker firstly converts the adjective + noun into a noun phrase.

Abney was aware of the difficulty in determining chunk end points, these being unmarked in the sentence, and addressed this problem by 'slipping-in' fake end-of-chunk markers into the look-ahead buffer of his Chunker. The fake markers did not

impact performance to any extent, but did allow for error-recovery in unanticipated grammatical structures [ABNE91]:

> "Since sentences often contain structures that were not anticipated in the grammar, and since we want to get as much information as possible even out of sentences we cannot completely parse, error recovery of this sort is very important."

The above is perhaps slightly out of context, but in reading for comprehension, rather than for some E-Language notion of grammatical correctness, we do indeed want to extract as much information as possible and so a degree of error-recovery in chunk detection, and hence in situations such as that of sentence 5.10c, is desirable. The effect of using NP categories in place of N and NP categories promotes this kind of error-correction, but it may also introduce further parsing ambiguities, which shall be investigated in future work.

5.6 Conclusion.

The traditional chart parser accepts one space-delimited term at a time, and because of the N^3 complexity of the chart-parsing algorithm, longer sentences impact heavily on processing resources available. Chunking of text has been shown to be beneficial in that it reduces the number of terms shifted into a chart parser, thereby significantly reducing the number of operations necessary to parse the source sentence. Justification of the inclusion of a Chunker in a psychological model of text comprehension is presented in terms of the Parallel Recognition Model, the human visual system being shown to acquire a number of words at one fixation, leading to activation of compounds in the lexicon prior to shifting. It is also reasoned that non-compositionality of the syntactic categories of a compound's constituent words makes compound detection prior to

shifting a necessity. An explanation of errors made by humans when reading two parallel sentences, one of which is a garden path sentence, requires the selection of a compound over a single word on grounds of plausibility in one instance, whilst the reverse is true in the other. This leads to the conclusion that both alternatives (the single and compound words) must coexist for a selection to occur, and hence any chart parser must be capable of supporting single and compound words in parallel. A novel chart structure is presented to address this conclusion, extending the standard chart model by offering support for parallel shifts, doing so in a resource-efficient manner, and by permitting deactivation of parallel chart columns and their reactivation when an error is detected and re-evaluation is required. The new parallel shift chart parser is demonstrated in Chapter 9.

It is also proposed that, on the grounds that nouns and noun phrases take the same role in the overall grammatical structure of a sentence, and that the Chunker has responsibility for building noun-phrases, the atomic CG categories N and NP may be unified into a single category NP, resulting in fewer complex category variants.

The implication to be drawn from the chunking described above, particularly from the examination of the parallel parses, is that the chunking process is closely coupled with the chart itself; the purpose of the Chunker is to present the parser with all potential single and compound words present in any given sentence, thereby making them available for plausibility testing when building derivations. This is not to suggest that stand alone Chunkers such as Abney's have no place; the system described above utilises the lexicon as its recogniser. The lexicon contains words classified as *common*, and perhaps idioms, but will not necessarily contain any from the class *proper*, which,

like phrases, are more readily detected by a pattern-matching Chunker, and intuitively are not likely to generate output that requires parallel shifts.

The discussion presented here therefore supports the notion of chunking as not only cognitively viable and algorithmically possible, but also a necessity.

6 The Sense Element

This chapter argues that fine-grained sense representations will have a detrimental impact on processing resources because the large number of possible permutations of those senses, when combined to form knowledge representations, and will have to be evaluated against each other in the decision making processes (such as coherence determination in the Construction Integration Model - Chapter 3). A novel method of abridging the WordNet noun and verb hypernym taxonomies (effectively a new tree-cut model) is presented to address this problem; the Specialisation Classes at the cuts are shown to be few in number but to retain the sense-distinctions of the original taxonomies to a high degree.

As the basic representation of linguistic knowledge used by the Construction Integration Model is propositional, it follows that the knowledge integrated into working memory from long-term memory is also propositional. Although the nature of the propositional arguments is not known [KINT78], it can be deduced that as, according to [KINT78], the argument is the basic unit of meaning, it should at the very least provide some index into a larger meaning representation which would provide explicit information regarding the entity represented by the argument. For example, the argument DOG provides an index to a knowledge structure associated with dogs, which would contain information such as:

- Has four legs
- Covered in fur
- Wags tail when happy
- Barks when excited
- Likes to eat sausages
- Likes chasing cats
- Kept as pets by humans

This use of a 'lexical' key into a knowledge structure has been explored primarily as a means of constructing knowledge bases, generally using WordNet [MILL95] as the raw lexical/knowledge resource, for example [HARA98], [HIRS98], [POWE00].

In terms of efficient storage and processing, a propositional approach to representation may cause problems. Take for example knowledge about things that are suitable for drinking, that is, which entities can realistically take the direct-object position of the verb *drink*: tea, coffee, beer, wine, whisky, water, … are all candidates, and to store these individually, along with all other knowledge, would result in a large search space. The consequently lengthy search times would make this fine-grained knowledge representation unsuitable for a real-time task such as summarising web pages. This chapter addresses this problem by proposing a novel method of abstracting the WordNet noun and verb hypernym/hyponym taxonomies that significantly reduces the number of entries with little loss of sense distinction. This is achieved through observance of *sense similarity*.

6.1 Similarity

Various methods of assessing *word similarity* have been proposed: Using a taxonomy such as WordNet, two nodes are similar if they share a hypernymically related node. The degree of similarity may be determined by counting edges [RADA89]; [LEAC98]. *Semantic Similarity* [RESN95a] measures similarity as *information content* of the common subsumer, obtained from taxonomy node probabilities assigned through corpus frequency analysis. This approach has been augmented by factoring-in path length [JIAN97], itself similar to the *Similarity Theorem* based *Lin Measure* [LIN97]. Relations other than hypernym/hyponym have been used, employing defined sequences

of directed relational types [HIRS98]. Tree-Cut Models (TCM) employing Minimum Description Length [QUIN89] have been used to partition noun taxonomies on similarity of case-frame slot fillers [LI95a]; [LI96]. As an alternative to these approaches, Lesk proposes *dictionary definition overlap* [LESK86], where increasing definition-word overlap indicates greater similarity.

The similarity metrics above, with the exception of the Tree-Cut Model, all produce a measure of how similar two senses are (or will state that they are not similar). So, given CAR and LORRY, these metrics will report that they are very similar, and share the hypernym MOTOR VEHICLE. CAR and SKATEBOARD are less similar, but similar nonetheless, and share the hypernym ARTEFACT. However, by the same token, CAR and PENCIL are also similar, again sharing the hypernym ARTEFACT. To avoid this unacceptable result, a similarity threshold would be required; those cases where the similarity value was found to be above the threshold accepted as similar, and those below rejected. This presents yet another problem in that a suitable threshold must be selected. The Tree-Cut Model on the other hand partitions the hypernym/hyponym taxonomy, thereby collecting similar senses under each cut. Using this scheme it is possible to give a yes/no answer to the question ‘are these senses similar?’. However, the proposed TCM is designed to identify senses that are similar with respect to their roles in case frames, requiring consideration of their coocurrence probabilities with some predicate. Nevertheless, having a preselected set of cuts, and hence groups of similar senses, is attractive considering the real-time application we have in mind.

6.2 A Method for Predefining Groups of Similar Senses

A method of defining sets of similar senses presents itself if one considers Resnik's procedure for calculating the *information content* (IC) of nodes in the WordNet noun hypernym taxonomy [RESN95a]; [RESN98]. Recall that in the construction of the probabilistic model of WordNet, the frequency of any class *c* is calculated recursively as the number of occurrences of that class plus the sum of the frequencies of its hyponyms, shown in equation 1 from [RESN98] below.

$$\text{freq(c)} = \sum_{w \in words(c)} \frac{1}{|\, classes(w) \,|} \text{freq(w)} \qquad (1)$$

where *words*(c) is the set of words in any synset subsumed by *c*, and where *classes(w)* is the set $\{c \mid w \in words(c)\}$.

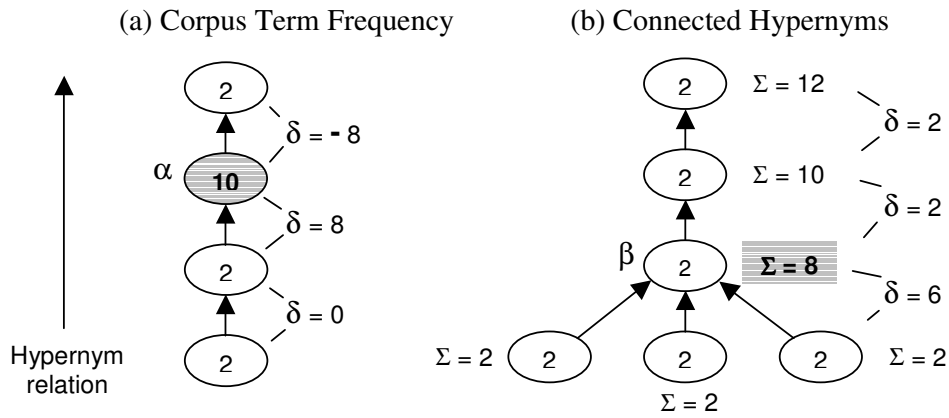


Fig. 6.1. The derived frequency of a class depends upon both term frequency (a), and on number of hyponyms (b). Individual term frequencies are shown within nodes, sum indicates cumulative class frequency. δ signifies the change in (cumulative) frequency between a class and its subsumer.

Two factors are involved in the calculation of the IC value of a WordNet class: Firstly, the raw frequency of occurrence of terms, as derived from corpus analysis, is assigned

to appropriate classes. This results in the more frequently occurring classes having a higher frequency score than less occurring classes, as illustrated by node α in Fig. 6.1a. In some way, this echoes Luhn's observation that term frequency and term significance are related [LUHN58]. Secondly, the frequency scores are cumulatively propagated along the hypernym relation, resulting in the summed class frequency being additionally influenced by its hyponyms, as shown by node β in Fig. 6.1b, which is reminiscent of a *spreading-activation network*.

In the two examples above, it can be said that the labelled nodes form *abstract classes*; node α is similar to a term frequency based *keyword* within its hypernym chain, and node β is highly activated by its subordinate nodes. Observe in each case, there is a large change in value (frequency and summed frequency respectively) between the labelled node and its immediate hyponym(s). This effect is shown clearly in Table 6.1, the cumulative frequency data for the hypernym chain of DOG (canine). Word frequency data is derived from the 100 million word British National Corpus (BNC) and applied to the noun taxonomy of WordNet 1.6.

Class	ΣFreq
ENTITY	963909
ORGANISM	385594
ANIMAL	38913
CHORDATE	21502
VERTEBRATE	21496
MAMMAL	13657
PLACENTAL	13391
CARNIVORE	2803
CANINE	1203
DOG	995

Table 6.1. Cumulative frequencies of hypernyms of DOG (canine)

Note that in Table 6.1 the summed frequencies do not change smoothly; there are particularly large changes when moving from ANIMAL to ORGANISM (Δ=346681), and from ORGANISM to ENTITY (Δ=578315). These are caused by the summation of frequencies of *all* subordinates of ORGANISM (including ANIMAL), and of *all* subordinates of ENTITY (including ORGANISM) respectively, of which there are many. From this we deduce that ORGANISM and ENTITY strongly abstract the hypernyms of dog. However, in an ideal situation we would prefer just the right level of abstraction, not strong abstraction - clearly ORGANISM does not discriminate between DOG and CAT, or even PLANT and ANIMAL. Worse still, ENTITY cannot discriminate between DOG and BICYCLE.

Following Resnik [RESN98], the information content value *I* for each class *c* was calculated using equation 3, after first deriving the class probabilities *p(c)* from the cumulative frequencies via equation 2.

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

$$\text{where } N = \Sigma_{c'} freq(c') \qquad \text{for c' ranging over all classes} \qquad (2)$$

$$I_c = -\log p(c) \qquad (3)$$

Table 6.2 shows that, as expected, the classes near the top of the taxonomy express relatively little information (column **IC**). Calculating the change (increase) in information (column **ΔIC**) reveals the greatest change takes place in the move from ORGANISM to ANIMAL.

Class	ΣFreq	Prob	IC	ΔIC
ENTITY	*963909*	*0.03962*	*1.40212*	
ORGANISM	*385594*	*0.01585*	*1.80003*	*0.39790*
ANIMAL	38913	0.00160	2.79606	**0.99603**
CHORDATE	21502	0.00088	3.05367	0.25761
VERTEBRATE	21496	0.00088	3.05380	0.00013
MAMMAL	13657	0.00056	3.25081	0.19701
PLACENTAL	13391	0.00055	3.25933	0.00852
CARNIVORE	2803	0.00012	3.93848	0.67915
CANINE	1203	0.00005	4.30596	0.36748
DOG	995	0.00004	4.38817	0.08221

Table 6.2. Class-based probability and information values for the hypernym chain of *dog*

If ENTITY and ORGANISM are strong abstractions of DOG, then it can be said that the classes ANIMAL..DOG are *specialisations* of the strong abstractions, Further, as the move from ORGANISM to ANIMAL presents the greatest change in IC, then the greatest specialisation happens at ANIMAL. We have chosen to designate the node that incurs the greatest positive change in IC a *Specialisation Class* (**SC**). Thus ANIMAL is the SC of those classes within the DOG hypernym chain. Intuitively, ANIMAL does seem to present a *plausible abstraction* of DOG, and it certainly discriminates between DOG and BICYCLE. By applying cuts to the WordNet noun hypernym taxonomy at the SCs we can construct an abridged WordNet noun hypernym taxonomy; the nodes of the taxonomy will be the SCs, and each SC will 'contain' all subordinate similar senses.

An SC can be formally defined using the 'Z' notation as follows:

Given: (4)

[CLASS]	the set of WordNet noun classes.
c: CLASS	c is of type CLASS
I: c ⇸ REAL	Function I returns the information content of class c.

The hypernym function H can be defined as: (5)

$H: CLASS \leftrightarrow CLASS$

$H(c) = c_h \mid c\ IS\_A\ c_h$

Note that: (6)

$H^n(c)$ represents n applications of H

Hence: $H^2(c) \equiv H(H(c))$

and

H^0 represents the identity, that is, $H^0(c) = c$

The Specialisation Class selection function SC can now be defined: (7)

$SC: CLASS \twoheadrightarrow CLASS$

$SC(c) = H^n(c)$ where
$\exists\ n : \mathbb{N} \mid n \geq 0 \bullet MAX(I(H^n(c)) - I(H^{n+1}(c)))$

6.3 Identifying the Specialisation Classes

Using the BNC as the reference source, the information content of each WordNet noun class was calculated as per equations 1 to 3 above. The specialisation class selection function *SC*, defined in equation 7, was then applied to identify the subset of WordNet noun classes that constitute the Specialisation Classes, as shown in equation 8. Initial examination of the results showed that for many nouns, the immediate hypernym of a root class was selected as the SC - an unsatisfactory result precipitated by the fact that these classes are the focus of many subordinate classes. To counteract this, the roots and their immediate hypernyms were disallowed as candidates for selection. SUBSTANCE, a third-level class, was also found to have a very high change in information content,

leading to its preferential selection, and so was similarly disallowed. This resulted in 145 noun *base classes* being disallowed. Although we have addressed the need to eliminate classes high in the taxonomies from being selected as Specialisation Classes simply by 'lopping-off' the top two or three levels of the taxonomies, other approaches are possible. For example, the number of subsumed classes and/or the depth of a subtree below a class under consideration could be factored in to the Specialisation Class selection algorithm. However, these more considered approaches shall be addressed by future research.

[CLASS] The set of WordNet noun classes.

SCLASS: ℙCLASS The set of noun Specialisation Classes.

$$SCLASS = \{\forall c: CLASS \bullet SC(c)\} \quad (8)$$

The verb taxonomy was similarly processed, with the exception that as no bias was found towards the top of the taxonomy, possibly due to the shallow, bushy nature of the verb taxonomies, there was no need to disallow any verb classes from the selection process. However, as the selection mechanism can never select the root of a taxonomy, 618 verb base classes were nevertheless ignored. We will return to this issue in Section 6.5**.**

6.3.1 Abridging Hypernym Chains

It is interesting to note that a class *c* selected as the SC of a noun sense *s* is not necessarily selected as the SC of all hyponyms of *s*. Take for example the classes DOG and HAMSTER. As has been seen above, ANIMAL is the SC of DOG, and is also a hypernym of HAMSTER. However, the SC of HAMSTER is RODENT. This observation

permits an abridged representation of HAMSTER to be constructed by selecting only the identified SCs, as shown in Fig. 6.2.

HAMSTER: **RODENT** → PLACENTAL → MAMMAL → VERTEBRATE →
CHORDATE → **ANIMAL** → LIFE_FORM → ENTITY

HAMSTER: **RODENT** → **ANIMAL**

Fig. 6.2 Abridged hypernym representation of HAMSTER using SCs

Complex taxonomic structures, such as that for BEER, see Fig. 6.3, are easily accommodated by traversing each hypernym path from leaf to root separately. Table 6.3 gives the change in information content values for the three paths associated with BEER, and shows that BEVERAGE, FLUID and DRUG are directly selected as SCs of BEER.

Path A			Path B			Path C		
Class	**Info**	**ΔInfo**	**Class**	**Info**	**ΔInfo**	**Class**	**Info**	**ΔInfo**
ENTITY	1.40212		ENTITY	1.40212		ENTITY	1.40212	
OBJECT	1.59641	0.19429	OBJECT	1.59641	0.19429	OBJECT	1.59641	0.19429
SUBSTANCE	2.30612	0.70971	SUBSTANCE	2.30612	0.70971	ARTEFACT	1.83769	0.24128
FOOD	2.76016	0.45404	**FLUID**	**3.45593**	**1.14981**	**DRUG**	**3.22856**	**1.39088**
			LIQUID	3.47550	0.01957	D ABUSE	3.59402	0.36545
BEVERAGE	**3.64836**	**0.88820**	BEVERAGE	3.64836	0.17286			
ALCOHOL	3.78927	0.14092	ALCOHOL	3.78927	0.14092	ALCOHOL	3.78927	0.19526
BREW	4.57581	0.78654	BREW	4.57581	0.78654	BREW	4.57581	0.78654
BEER	4.87778	0.30197	BEER	4.87778	0.30197	BEER	4.87778	0.30197

Table 6.3. Change in information content for hypernyms of BEER

Processing the entire noun taxonomy in this way selects 4373 of the available 66025 WordNet noun classes. Similarly, 931 of the 12127 verb classes were selected.

6.3.2 A Fully Abridged Taxonomy

Recall that the *base classes* disallowed by the selection process do not appear in the set of extracted SCs. Nevertheless, they may be encountered in texts, and are required in

order to reconstruct (in abridged form) the original noun and verb taxonomies. For these reasons the base classes were added to the set of SCs, resulting in a total of 4518 noun and 1625 verb classes in the abridged WordNet noun and verb taxonomies. This corresponds to an abridged noun taxonomy 6.8% of the size of the original, and an abridged verb taxonomy 13.4% the size of the original.

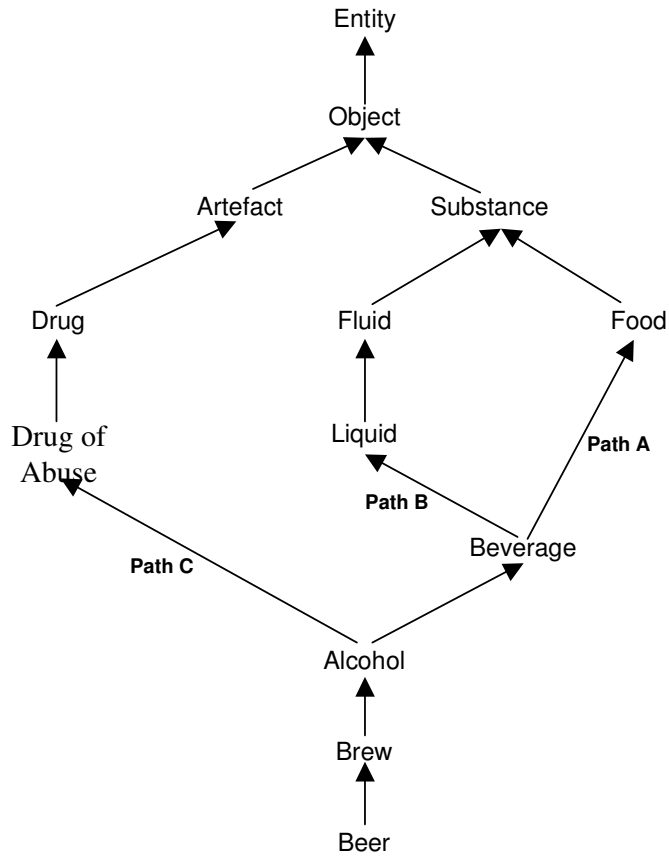


Fig. 6.3. WordNet Hypernym representation of BEER.

The abridged representation of BEER, constructed only of SCs, is shown without base classes in Fig. 6.4a, and with base classes in Fig. 6.4b. Note that BREW and FOOD are selected as SCs by processing other senses not shown here.

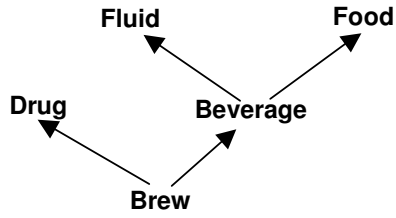


Fig. 6.4a. Abridged representation of Beer

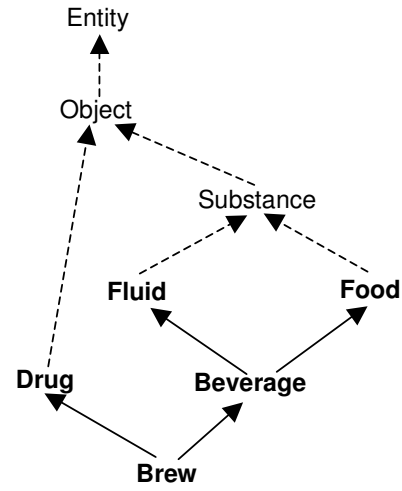


Fig 6.4b. Fully abridged representation of Beer

6.3.3 Discussion

Tables 6.4 and 6.5 show the distribution of noun and verb classes within their respective specialisation classes. **Size** indicates the number of classes subsumed by an SC, and **Freq** the number of occurrences of an SC containing **Size** classes.

Table 6.4 shows that 54 noun SCs do not subsume any other class, and consequently only synonymous lemmas can be grouped by these SCs. This is also true of the 271 instances of single-class verb SCs. The most frequent number of classes subsumed by a noun or verb SC is 2, corresponding to the SC class and one hyponym of that class. Although SCs containing few senses are frequent, the tables show that some SCs subsume a high number of senses – the highest containing 1920 nouns and 830 verbs. Examination of the data revealed that the PLANT (*flora*) SC held the most noun senses, closely followed by ANIMAL (*animate being*), FAMILY (*taxonomic*) and COMPOUND (*chemical*). For verbs, CHANGE (*transform*) was the most populous SC, followed by

CHANGE (*undergo change*), MOVE (*locomote*) and MOVE (*displace*). Considering the large numbers of animals, plants, and taxonomic classifications, together with ways to or be changed or moved, contained within a dictionary such as WordNet, it is entirely predictable that highly subsuming SCs will exist. However, as the senses within an SC are not distinguished from each other in any way (other than by lemma), no subdivisions within an SC exist. This results in no distinction being made between PARROT and DOG for example - both map on to the SC ANIMAL. This may be problematic if SCs are to be used as the basis for selectional association calculations; where it would only be possible to state FLY(ANIMAL), and not FLY(BIRD) for example.

A solution to the above problem, should one be necessary, would be to add all populous SCs to the base classes during SC extraction; as these are disallowed by the selection process, the class scoring the next highest change in information would be selected in its stead. So in the case of DOG (Table 6.2), ANIMAL would be disallowed, and CARNIVORE would be selected, and as a consequence the SC ANIMAL would no longer contain any carnivores directly. The process could be repeated until all SCs contained less than a predefined number of senses. On completion, base classes are combined with the selected classes to form the abridged taxonomy, and so ANIMAL, and all other senses shunted into the set of base classes, would again become available as an SC, albeit with fewer subsumed senses. However, this would lead to an increase in the number of SCs, and consequently increase the number of relations expressed in a World Model based upon them. Unless this recursive selection of SCs is performed in moderation, it is possible to reproduce the high-granularity of the original taxonomy.

Size	Freq	Size	Freq	Size	Freq	Size	Freq	Size	Freq	Size	Freq
1	54	31	9	61	3	96	3	146	1	255	1
2	1022	32	17	62	6	97	3	148	2	261	1
3	663	33	15	63	2	99	1	150	1	276	1
4	460	34	11	64	2	100	1	152	1	286	1
5	322	35	11	66	3	101	1	153	1	288	1
6	231	36	5	67	2	103	1	155	1	299	1
7	183	37	11	68	4	105	1	160	2	300	1
8	168	38	5	69	4	106	2	162	1	303	1
9	144	39	8	70	2	107	1	169	1	306	1
10	104	40	5	71	1	110	2	170	1	308	1
11	105	41	10	72	4	111	2	178	1	313	1
12	82	42	8	73	1	112	1	179	2	322	1
13	79	43	6	74	1	115	1	183	2	324	1
14	65	44	5	75	3	116	1	190	1	333	1
15	56	45	7	76	3	118	1	191	1	334	1
16	47	46	5	78	4	120	1	193	1	364	1
17	43	47	10	79	2	122	2	198	1	367	1
18	30	48	6	80	2	127	1	199	1	370	1
19	38	49	3	81	1	129	1	202	3	385	1
20	30	50	10	82	2	130	2	204	3	401	1
21	34	51	4	83	2	133	1	206	1	423	1
22	23	52	3	84	2	134	1	207	2	524	1
23	20	53	4	85	3	135	2	208	1	558	1
24	16	54	6	87	1	136	2	215	1	607	1
25	18	55	10	88	1	138	1	218	1	774	1
26	14	56	1	89	1	140	2	227	1	860	1
27	18	57	6	91	2	141	1	229	1	1070	1
28	23	58	4	92	1	143	2	239	1	1824	1
29	18	59	2	93	1	144	1	242	1	1920	1
30	19	60	3	94	3	129	1	245	1		

Table 6.4. Number of noun classes subsumed by noun SCs

Size	Freq	Size	Freq	Size	Freq	Size	Freq
1	271	17	15	33	3	71	1
2	425	18	10	35	4	74	1
3	234	19	12	36	2	75	1
4	130	20	4	38	1	80	2
5	107	21	4	40	1	87	1
6	93	22	12	41	2	91	1
7	51	23	7	43	1	143	1
8	48	24	1	45	1	150	1
9	30	25	2	49	1	152	1
10	19	26	7	50	1	154	1
11	22	27	2	53	1	187	1
12	21	28	4	55	1	236	1
13	12	29	5	58	1	295	2
14	14	30	3	61	1	376	1
15	7	31	1	64	1	830	1
16	9	32	3	65	1		

Table 6.5. Number of verb classes subsumed by verb SCs

6.4 Evaluation of SC Sense Distinctions

To determine the degree to which sense distinctions have been preserved in the abridged noun and verb hypernym taxonomies, a precision/recall experiment was devised to evaluate the ability of SCs to disjointly partition the senses of polysemic lemmas: by recognising that the function *SC* simply maps a given class on to itself or one of its hypernyms it can be seen that, ideally, the *n* senses of a polysemic lemma should map on to *n* SCs. The senses of that lemma may thus be considered *query terms*, and the mapped SCs the *target set*. Recall will always be 100% as the SCs will always be hypernyms of the query terms (or the query terms themselves), whereas Precision may be reduced if two or more query terms map on to the same SC. Precision is therefore calculated as follows:

> Let Λ be a lemma, $\sigma(\Lambda)$ a function returning the set of senses of Λ, and $\chi(\Lambda)$ a function returning the collection of SCs for all senses of Λ.
>
> $$\text{Precision} = \frac{\#\sigma(\Lambda)}{\#\chi(\Lambda)} \qquad (9)$$

6.4.1 Evaluation datasets

To evaluate the ability of SCs to discriminate between senses of a lemma, all 94474 noun (10319 verb) lemmas from the WordNet NOUN.IDX (VERB.IDX) tables were processed. Along with the set of 4518 noun (1625 verb) SCs extracted by the above method, for comparative purposes two additional sets of SCs were generated: (a) a baseline containing only the 145 noun (618 verb) base classes, and (b) a randomly selected set of 3907 noun (1605 verb) SCs (including the base classes).

Precision was calculated for each lemma obtained from the noun (verb) index according to equation 9 and recorded in an array indexed on $\#\sigma(\Lambda)$. For completeness,

monosemous lemma occurrences were recorded and, as only one sense is available to its SC, assigned a recall of 100%.

6.4.2 Results

The precision values for the three evaluations, for both nouns and verbs, are presented in Table 6.6. Column **#σ(Λ)** indicates the number of senses obtained for a lemma, **Count** the number of lemmas contained in each of the above groups, and **Bases**, **Rnd+Bases**, and **SC+Bases** the precision of the three abridgement sets.

In calculating the average precision, monosemous lemmas (#σ(Λ) = 1) were ignored, as were those values of #σ(Λ) for which no data was seen (Count = 0)**,** resulting in 21 noun and 39 verb precision values. Of the 4518 noun (1625 verb) SCs, 432 (28) corresponded to monosemous lemmas, the remaining 4086 (1597) to polysemous lemmas.

The relatively low number of noun bases presents a coarse-grained abridgement, which is reflected in its low recall (0.5381) in the polysemous lemma discrimination task. The random selection, covering more classes lower in the taxonomy, provides a better precision (0.7574), but the best precision is obtained using the extracted SCs (0.9464). The situation is similar for the verb discrimination task, the extracted SCs producing the highest precision (0.8328).

On three occasions the random verb precision equalled the SC verb precision (*#σ(Λ)* = 14, 17, 48) and on one occasion bettered it (*#σ(Λ)* = 30). No SC noun precision was equalled or beaten by a random noun precision.

	Noun Precision				Verb Precision			
Size		145	3907	4518		618	1605	1625
#σ(Λ)	**Count**	**Bases**	**Rnd + Bases**	**SC + Bases**	**Count**	**Bases**	**Rnd + Bases**	**SC + Bases**
1	81910	1.000	1.0000	1.0000	5752	1.0000	1.0000	1.0000
2	8345	0.7901	0.8991	0.9434	2199	0.9038	0.9293	0.9611
3	2225	0.7112	0.8661	0.9426	979	0.8488	0.8931	0.9302
4	873	0.6804	0.8411	0.9444	502	0.8237	0.8675	0.9268
5	451	0.6718	0.8483	0.9512	318	0.7679	0.8277	0.8931
6	259	0.6274	0.8346	0.9556	188	0.7660	0.8333	0.9069
7	140	0.5898	0.8102	0.9541	102	0.7507	0.8305	0.8978
8	82	0.5762	0.7835	0.9482	75	0.7333	0.7767	0.8867
9	68	0.5376	0.7598	0.9493	39	0.7009	0.7664	0.8803
10	42	0.5476	0.7429	0.9286	39	0.7359	0.7897	0.8769
11	23	0.5415	0.7312	0.9605	32	0.7358	0.8097	0.8835
12	18	0.5602	0.7500	0.9861	15	0.7444	0.8056	0.8833
13	9	0.5470	0.7436	0.9487	16	0.6827	0.7692	0.8606
14	8	0.4643	0.6696	0.9464	5	0.7571	0.8429	0.8429
15	7	0.4952	0.7333	0.9333	8	0.7667	0.7917	0.9167
16	3	0.4583	0.7083	0.9167	8	0.6641	0.7422	0.8359
17	6	0.4412	0.6275	0.9216	4	0.6324	0.7647	0.7647
18	1	0.5556	0.8333	1.0000	4	0.6528	0.7222	0.8333
19	1	0.4737	0.8421	0.8947	2	0.5789	0.7632	0.8158
20	0				2	0.5000	0.6250	0.7000
21	0				3	0.8095	0.8413	0.9048
22	0				3	0.5758	0.6212	0.8182
23	0				1	0.4783	0.5652	0.6087
24	1	0.2500	0.4583	0.9167	3	0.6528	0.7083	0.7222
25	0				2	0.6800	0.7800	0.9000
26	0				3	0.5769	0.7308	0.7821
27	0				1	0.5185	0.6296	0.6667
28	0				1	0.7500	0.7857	0.8571
29	1	0.4138	0.6897	0.9655	1	0.6552	0.6897	0.8966
30	1	0.3667	0.7333	0.9667	1	0.7333	0.8333	0.8000
32	0				1	0.5313	0.6875	0.7500
33	0				1	0.5455	0.7273	0.8182
36	0				1	0.6944	0.7778	0.8889
37	0				1	0.6757	0.7838	0.8378
38	0				1	0.6316	0.7105	0.7632
41	0				2	0.6341	0.7195	0.8293
42	0				1	0.5476	0.6667	0.8095
45	0				1	0.5556	0.6667	0.9333
48	0				1	0.5625	0.6667	0.6667
63	0				1	0.4921	0.6349	0.7302
Total	94474				10319			
	21 polysemous lemma groups				39 polysemous lemma groups			
Average		**0.5381**	**0.7574**	**0.9464**		**0.6671**	**0.7533**	**0.8328**

Table 6.6 Precision of sense distinctions of three abridgements for both nouns and verbs

6.5 Verbal Specialisation Classes and Polysemy

It is interesting to note that although the precision using SCs is not 100%, it does not necessarily follow that the SC selection procedure is somehow flawed. Take for example the lemma MUSIC, for which WordNet 1.6 lists 6 senses. Of these, senses MUSIC#2 and MUSIC#5 are described as '*any agreeable (pleasing and harmonious) sounds*' and '*the sounds produced by singers and musical instruments*' respectively. Both of these senses map on to the SC PERCEPTION#3. Further, these two senses of MUSIC share the immediate hypernym SOUND#2 (auditory sensation). It is therefore not surprising, and should be expected, that a certain number of one-to-many mappings between SCs and senses will be encountered.

The one-to-many mapping between SCs and lemma senses may be of use in identifying *polysemous* lemmas, that is, those lemmas that have different but related senses, such as MUSIC#2 and MUSIC#5 above. If the lemma senses are related through hypernymy, then polysemous lemmas will share a hypernym that is not too distant from the sense of the lemmas (distant common hypernyms, such as *entity*, would include many senses of seemingly unrelated lemmas). For example, WordNet 1.6 lists 5 senses for the verb DRINK, these being:

Sense	**Synonyms**	**Definition**
Drink#1	drink, imbibe	take in liquid
Drink#2	drink, booze, fuddle	consume alcohol
Drink#3	toast, drink, pledge, salute	propose a toast to
Drink#4	drink_in, drink	be fascinated or spellbound by
Drink#5	drink, tope	drink alcohol, be an alcoholic

Senses 1,2 and 5 all relate to consumption of liquid, senses 2 and 5 making that liquid alcohol. Sense 3 is an activity undertaken to honour someone or something, and sense 4

is a state of mind. The 5 senses of DRINK therefore divide into 3 homonymous groups relating to consumption, honouring, and cognition.

The SCs of the 5 senses of DRINK are shown in Table 6.7 together with their WordNet *hereiam* field values, which are commonly used to identify/locate senses within the WordNet databases. The table shows that senses 1,2 and 5 of DRINK map on to the same SC CONSUME and so have been identified as polysemous. Sense 3 maps onto the SC PRIZE, and sense 4 to STEEP, which with CONSUME gives 3 homonymous groups in total.

Sense	Specialisation Class	hereiam
1	**consume**	**786286**
2	**consume**	**786286**
3	prize → consider → judge	1544040 → 466852 → 452184
4	steep → cogitate	406072 → 426277
5	**consume**	**786286**

Table 6.7. Specialisation Classes of senses of the verb DRINK.

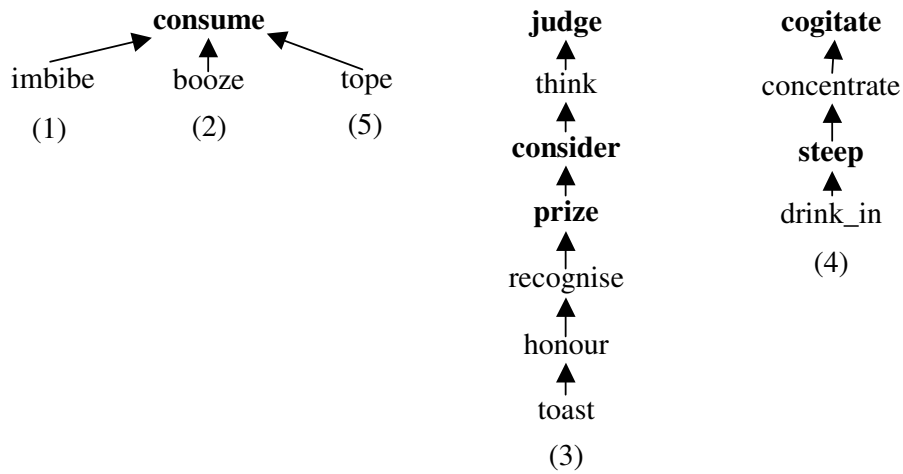


Figure 6.5. Hypernyms of senses of DRINK, SCs shown emboldened.

In this instance the SCs have successfully grouped polysemous senses, and have distinguished between homonyms. Without a definitive list of polysemous senses and homonymic partitions for senses expressed in WordNet, evaluation of the Specialisation Classes abilities in this respect is difficult to quantify. A feel for the homonymic partitioning may be obtained however by examination of a small sample of verbs. The six verbs selected, WRITE, READ, WARN, HEAR, REMEMBER, and EXPECT are taken from Resnik's table of selectional association for plausible objects [RESN98].

6.5.1 Write

Sense	**Definition (WRITE)**	**Specialisation Class**	**hereiam**	**Group**
1	indite	WRITE → *make*	1164896 → *1111638*	A'
2	communicate by writing	WRITE → *communicate* → *act*	671763 → *502333* → *1612822*	B'
3	publish	make	1111638	A
4	drop a line	correspond	680786	C
5	communicate by letter	*communicate* → *act*	*502333* → *1612822*	B
6	compose music	make	1111638	A
7	mark a surface	trace → change	1089750 → 83947	D
8	record data	WRITE → record → save → have	675295 → 675051 → 1522764 → 1508689	E
9	spell	spell	1166173	F

Table 6.8. Specialisation classes of the verb WRITE.

Table 6.8 shows that the nine senses of WRITE map directly on to eight Specialisation classes. Group A contains two polysemous senses, PUBLISH and COMPOSE_MUSIC, which map on to the SC MAKE (make or cause to be or to become). Sense 1 also maps on to MAKE, but not directly. Similarly, senses 2 and 5 both contain the SC INDITE, but

only as an immediate SC for sense 5. In these cases, the common SC is an indication of *similarity*, not polysemy.

Sense 1 could be coerced into the MAKE group (A) if reflexive SCs (which are shown capitalised) are ignored during polysemy determination; the SC of WRITE/671736 is WRITE/671736, which if ignored promotes the SC MAKE/1111638, resulting in a polysemous group containing INDITE, PUBLISH, and COMPOSE_MUSIC, which are all forms of bringing something written into being. Sense 2 also maps on to a reflexive SC, which if removed, enables a polysemous group concerned with COMMUNICATION to be formed with sense 5. None of the other senses of WRITE have any common hypernyms and so cannot form polysemous groupings.

6.5.2 Read

Sense	Definition (READ)	SC	hereiam	Group
1	interpret	READ → understand	423416 → 397666	A’
2	a certain wording	have	1794357	B
3	say out loud	talk → *communicate* → *act*	638109 → *502333* → *1612822*	C
4	scan	READ → understand	425290 → 397666	A’
5	interpret significance	guess → speculate → think	620765 → 627205 → 426277	D
6	interpret in a certain way	READ → understand	422928 → 397666	A’
7	study	READ	405251	E
8	register	indicate → inform → *communicate* → *act*	627736 → 564266 → *502333* → *1612822*	F
9	hear & understand	understand	397666	A
10	make sense of	understand	397666	A

Table 6.9. Specialisation classes of the verb READ.

Here, only one polysemous grouping can be directly constructed, consisting of senses 9 and 10, and is concerned with *understanding*. However, senses 1 4 and 6 possess reflexive SCs at the highest level. Again if these are eliminated, the five senses INTERPRET_IN_A_CERTAIN_WAY, INTERPRET, SCAN, HEAR_AND_UNDERSTAND, and MAKE_SENSE_OF can form a polysemous group. With the exception of sense 4 (SCAN), which has more to do with computers reading magnetic tape, the *understanding* sense of the group is acceptable. However, in the case of the computer, the 'understanding' is in the correct decoding of the data on the tape/media and its subsequent incorporation into some calculation etc. This is not the same as human interpretive understanding, but it can be viewed as analogous, and so we would argue that SCAN, in the context of computing devices, is a reasonable inclusion in the polysemous group.

Senses 3 and 8 cannot form a polysemous group as, although they both contain COMMUNICATE, making them similar, COMMUNICATE is buried within their SC taxonomies.

6.5.3 Warn

Sense	Definition (WARN)	Specialisation Class	hereiam	Group
1	notify	WARN → inform → communicate → act	589833 → 564266 → 502333 → 1612822	A
2	discourage	talk → communicate → act	638109 → 502333 → 1612822	B

Table 6.10. Specialisation classes of the verb WARN.

The verb WARN has only two senses, both of which include the SC COMMUNICATE. Again, the common SC is buried within the respective taxonomies, resulting in the two senses of WARN being similar but not polysemous.

6.5.4 Hear

Sense	Definition (HEAR)	Specialisation Class	hereiam	Group
1	perceive sound	perceive	1442173	A
2	learn, get wind of	HEAR	404522	B
3	examine evidence	probe → analyse	535682 → 435242	C
4	receive communication	perceive	1442173	A
5	Take heed	think	426277	D

Table 6.11. Specialisation classes of the verb HEAR.

Senses 1 and 4 of the verb HEAR both map on to the SC PERCEIVE, presenting a polysemous grouping. None of the other senses have any common SCs and so are all unrelated.

6.5.5 Remember

Sense	Definition (REMEMBER)	Specialisation Class	hereiam	Group
1	recall	REMEMBER	410666	A
2	think of	REMEMBER	412253	B
3	think back	REMEMBER	413589	C
4	reward	give	1506956	D
5	mention	think_of → *think*	494966 → *426277*	E
6	commend	think_of → *think*	494966 → *426277*	E
7	exercise memory	think	426277	F
8	commemorate	REMEMBER	413778	G

Table 6.12. Specialisation classes of the verb REMEMBER.

Senses 5 and 6 of REMEMBER map directly on to the SC THINK_OF. No other senses have any common hypernyms, although sense 7 maps on to THINK, which is common to senses 5 and 6, making *exercise_memory* similar to the polysemous group containing *mention* and *commend*.

6.5.6 Expect

Sense	Definition (EXPECT)	Specialisation Class	hereiam	Group
1	anticipate	judge	452184	A
2	require	EXPECT → demand → request → convey → move	513425 → 512630 → 510998 → 1527059 → 1263706	B
3	await	EXPECT	487408	C
4	consider reasonable	consider → judge	466852 → 452184	D
5	bear/carry	EXPECT	41553	E
6	expect child	await	487408	F

Table 6.13. Specialisation classes of the verb EXPECT.

No polysemous groups may be formed by the senses of EXPECT, although ANTICIPATE (regard something as possible or likely) and CONSIDER (consider reasonable or due) are similar, sharing the SC JUDGE.

Senses 5 (be pregnant with) and 6 (look forward to the birth of a child) might be expected to form a polysemous group, both being related to childbirth, however, the WordNet compilers have discriminated between carrying an unborn child and looking forward to its birth, which seems reasonable, and have placed the two senses in different subtrees of the WordNet verb taxonomy, thereby making them dissimilar.

6.6 Nominal Specialisation Classes and Polysemy

The nouns *letter, article, driver, story, reply* and *visit*, which are the objects in Resnik's table of selectional association for plausible objects [RESN98] will now be examined for polysemous grouping. Here, only the primary SC(s) will be shown, unless it is reflexive.

6.6.1 Letter, Article, Driver, Story, Reply, Visit

Sense	Definition (LETTER)	Specialisation Class	hereiam
1	missive	material	4799150
2	alphabetic character	signal	5085885
3	literal interpretation	interpretation	5361340
4	varsity letter	commendation signal	5013089 5085885

Table 6.14. Specialisation classes of the noun LETTER.

Sense	Definition (ARTICLE)	Specialisation Class	hereiam
1	prose	expressive_style piece	5293492 4740201
2	class of artefact	ARTICLE →object	12704 →9457
3	legal clause	ARTICLE →written_communication	4819775 →4786785
4	determiner	function_word	4767687

Table 6.15. Specialisation classes of the noun ARTICLE.

Sense	Definition (DRIVER)	Specialisation Class	hereiam
1	operates vehicle	operator	7444457
2	drives animals	DRIVER →worker	7228604 →6957738
3	golfer	contestant	6944043
4	software	writing	4794515
5	golf club	golf_equipment	2761965

Table 6.16. Specialisation classes of the noun DRIVER.

Sense	Definition (STORY)	Specialisation Class	hereiam
1	narrative	informing	5388175
2	fiction	literary_composition	4798536
3	building floor	STORY →structure	2700186 →3431817
4	chronicle	STORY →signal	4889518 →5085885
5	news report	STORY →information	5009122 →4977171
6	fib	lie	5063005

Table 6.17. Specialisation classes of the noun STORY.

Sense	Definition (REPLY)	Specialisation Class	hereiam
1	answer	REPLY →statement	5055491 →5040541
2	Speech exchange	REPLY →speech_act	5379686 5354574

Table 6.18. Specialisation classes of the noun REPLY.

Sense	Definition (VISIT)	Specialisation Class	hereiam
1	visit a person	assembly	798100
2	arrangement	social_gathering	6126145
3	residence as guest	sojourn	683061
4	call for inspection	investigation	416938

Table 6.19. Specialisation classes of the noun VISIT.

It is evident from the above tables that none of the six nouns form polysemous groups, although this should not be surprising; Table 6.6 shows that noun recall is 94.64%, suggesting that very few noun senses are grouped polysemically through their mapping on to SCs.

6.6.2 Kiss

A noun example showing polysemic grouping may be seen using the noun KISS (only primary SC is shown, unless it is reflexive):

Sense	Definition (KISS)	Specialisation Class	hereiam
1	caress with lips	KISS →touch	90133 →**79716**
2	a small candy	treat	5647296
3	glancing brush	touch	**79716**

Table 6.20. Specialisation classes of the verb KISS.

By eliminating the reflexive SC (KISS/90133) from sense 1 the secondary SC TOUCH is revealed, allowing senses 1 and 3 to form a polysemous group.

6.7 Reducing sense ambiguity through Specialisation Class mapping.

The above has demonstrated that the less-than-100% precision of lemma senses by use of their primary Specialisation Class as a search key is not necessarily a less than ideal result; grouping senses of a lemma on correspondence of their SCs results in polysemic groupings – lemma senses are grouped if they are similar. It can be argued that this is desirable, as by grouping similar senses the search space is being reduced without abstracting too far and the general sense is not lost. Additionally, ignoring primary Specialisation Classes that are reflexive with respect to an actual sense of the lemma in question facilitates greater polysemic grouping. This is a reasonable step to take as polysemy can only be detected by examination of the hypernyms of a lemma's senses and identifying those common to more than one sense; lemma senses are disjoint by definition and so cannot express polysemy through their primary WordNet sense.

When assigning a WordNet sense to a word, all senses must initially be considered. Contrary to the one word per sense assumption [GALE92, YARO92, YARO95] it cannot be assumed that all occurrences of a word within a text have the same sense [KROV98], and so every instance of every word must be similarly considered.

The reduction in the number of senses to be considered by using Specialisation Classes as *surrogate senses* can be determined for a sentence, document or corpus simply by comparing the number of possible WordNet senses with the number of possible Specialisation Classes for that collection.

Using SemCor once more as the test corpus, the number of WordNet senses for each noun and verb was found through WordNet lookup. The senses were then mapped on to their primary Specialisation Class, or the secondary where the primary was reflexive.

Those compounds mapped onto PERSON, LOCATION and GROUP were also ignored. The results are presented in Table 6.21.

	Noun Lemmas	Noun Senses	Noun SCs	Verb Lemmas	Verb Senses	Verb SCs
Count	78959	353453	323186	47698	494410	409866
Av. Senses		4.48	4.09		10.37	8.59
Reduction			**8.56%**			**17.10%**

Table 6.21. Reduction in lemma senses due to SC grouping.

For the noun and verb data extracted from SemCor, the average reduction in noun lemma senses is 8.56%, which is not particularly large. However, the verb sense reduction is 17.10% which, when one recalls that each verb sense will have up to four arguments to consider, is probably worth having as the argument evaluation calculations will also be reduced by 17.10%.

6.8 Conclusion

This chapter has presented and formally specified a novel method for abridging the WordNet noun and verb taxonomies, using change in class information content to identify those classes at which major specialisations occur. Although few in number when compared to the total number of classes in the taxonomy, these Specialisation Classes have been shown to retain the underlying sense distinctions of multi-sensed lemmas to a high degree. It has been argued that the reduction in precision of lemma senses, when keyed on specialisation classes, is a desirable property as the reduction is caused by conflation of similar lemma senses, that is, by polysemic grouping of senses. Further, as the abridgement algorithm effectively produces cuts in a taxonomy at specialisation classes, the classes under each cut may be viewed as similar, and therefore allows words of similar meaning to be grouped together.

It can be shown that the Specialisation Class selection mechanism bears some resemblance to the Category Utility (CU) function introduced by Gluck and Corter [GLUC85]. CU is used to rate the quality of partitioning functions employed as feature clustering algorithms for machine learning. Essentially, given a partition *C*, the CU of that partition is calculated as the average sum of the statistical contingency measure of decrease (i.e. delta value) in the proportion of incorrect predictions made by *C*. By varying the clustering algorithm, for example by varying the number or type of attributes used as the basis for the cluster, an optimal clustering algorithm can be selected for any given purpose. The Specialisation Class selection algorithm is not a cluster quality-ranking algorithm *per se* as it actually generates the clusters. However, in parallel with CU, the delta value is the change in information value of adjacent classes along a hypernym chain. The predictions as to the inclusion of a class in any cluster/partition are related to the class information value; classes, if taken as the point of partition, near the end of the hypernym chain have low information values and will make general predictions (i.e. a large cluster), whereas those near the beginning have higher information values and will make specific predictions (i.e. a small cluster). As large clusters and small clusters can potentially make erroneous predictions by inclusion and exclusion respectively, the Specialisation Class selection algorithm attempts to identify the optimal partition point at which the predictions are neither too general nor too specific. Therefore, if any class in a hypernym chain is a potential partition point between clusters, then the Specialisation Class selection algorithm is analogous to applying the CU function over a set of possible clusters - both range over a number of potential partitions in order to identify the best quality clusters.

The accuracy to which Specialisation Classes partition the senses of polysemous lemmas, along with the high degree of abridgement they afford, suggests that specialisation classes may be used as surrogates for noun and verb senses, effectively reducing the search space of any sense-discrimination procedure. It is this aspect of Specialisation Classes that shall be investigated in the following chapter.

7 Evaluation of Specialisation Classes in a Word Sense Disambiguation task

Chapter 6 detailed the creation of a compact sense taxonomy through detection of Specialisation Classes. Although SCs have been shown to significantly reduce the number of senses in the WordNet noun and verb hypernym/hyponym taxonomies whilst retaining lemma sense distinctiveness, their usefulness in linguistic applications has not yet been demonstrated. This chapter addresses this point by presenting a comparative study of the full and abridged WordNet taxonomies in the 'standard' linguistic task of Word Sense Disambiguation (WSD). This is achieved through use of Selectional Association, an approach to WSD that has been developed in conjunction with the WordNet hypernym taxonomy [RESN97], and employs a novel sense identification string that allows senses to be identified as the same, something not possible using WordNet Sense Keys.

Resnik proposes that the Selectional Association (SA) between a verb and an argument may be determined from the difference between the *prior distribution* of a noun in a given relation with *any* verb, and the *posterior distribution* of that noun with a particular verb within that grammatical relation. He goes on to demonstrate that, by calculating SA values from noun and verb-noun relation frequency data extracted from the structurally-annotated Brown Corpus [FRAN82], SA can assess the semantic fit of an argument in a given relation with a verb [RESN98]. The procedure for calculating SA values, as used by Resnik, is as follows:

- Calculate the probability of each noun class, *c,* expressed in the corpus. This gives the *prior* distribution of the noun classes $P_R(c)$.

- Calculate the probability of each Noun class and Verb pairing, $c|x$, expressed in the corpus. This gives the *posterior* distribution $P_R(c|x)$.

- Calculate the *Selectional Preference* value $S_R(x)$ of each verb, (x), in the verb taxonomy as:

$$S_R(x) = \sum \text{p(c|x)} \log \frac{\text{p(c|x)}}{\text{p(c)}} \quad (1)$$

- Calculate the *Selectional Association* value $A_R(x,c)$ of each verb/noun-class relation:

$$A_R(x,c) = \frac{1}{S_R(x)} \text{p(c|x)} \log \frac{\text{p(c|x)}}{\text{p(c)}} \quad (2)$$

The selectional preference (SP) value indicates how strongly the verb selects for its argument, for example the verb *eat* has a high SP value and selects strongly for its direct object (types of food), whereas *find* has a low SP as almost anything will serve as its direct object.

7.1 Resnik's Corpus Approach to Selectional Association

As the word frequency data is drawn from a text corpus in which neither noun nor verb has been sense-tagged, Resnik evenly distributes the credit for any observed noun among all the classes subsuming that noun. For example, given the verb-object pairs of (DRINK WINE) and (DRINK WATER), the credit for WINE is distributed amongst the senses WINE(beverage) and WINE(wine_coloured). Similarly, for WATER, the credit is distributed amongst the seven senses of water expressed in WordNet, these being

WATER(liquid), WATER(body_of_water), WATER(water_supply), WATER(archaic_element), WATER(perspiration), WATER(urine), and WATER(lachrymal_secretion). The above clearly shows that, as Resnik points out, related words are ambiguous in different ways [RESN98]. The contributions from the senses of WINE and WATER will therefore be cumulative only for shared hypernyms of those senses, for example LIQUID, which is common to both WINE(beverage) and WATER(liquid). Those hypernymic senses not shared will be distributed throughout the taxonomy and will not receive cumulative contributions. The result is a strengthening of the signal that (for example) LIQUID is associated with the verb DRINK in the object relation. Resnik shows that the SA model makes reasonable predictions of verb/noun-sense relations when compared to human judgements [RESN96, OLSE97], making the model a useable method of selecting the more appropriate senses from the full range of possibilities, if not necessarily providing a full WSD mechanism. Resnik's method of WSD using SA values performs rather poorly, Resnik reporting 44.3% accuracy of noun sense assignments, when compared to the 'first sense' algorithm, which selects the most frequently used sense of any word and provides 82.5% accuracy.

The attraction of the SA method here is twofold: firstly, as reported above, it allows human-like prediction of verb/noun-sense relations, and secondly it presents association values for information that may be described as propositional, that is between a verb and a noun, or predicate and argument.

7.1.1 Extending the SA model to verb classes

The model Resnik describes calculates SA values for relations holding between noun classes (senses) and verb lemmas (Fig. 7.1). The upshot of this is that the SA value tells us nothing about the semantic class of the verb.

We propose an extension to the model that replaces the verb lemma with all senses of that lemma (Fig. 7.2). This will of course introduce additional noise into an already

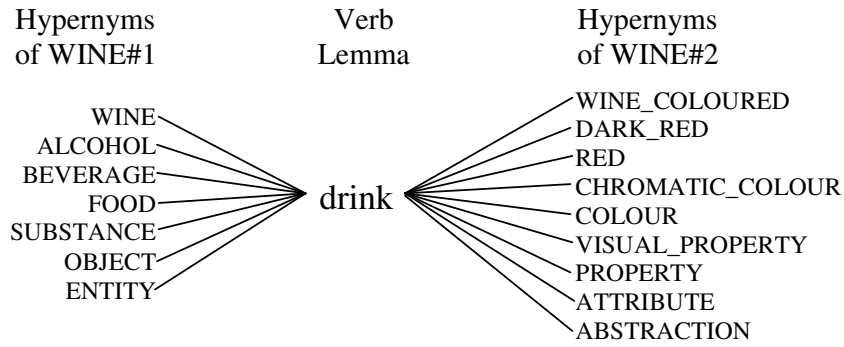


Fig. 7.1. SA is calculated between noun classes and verb lemmas.

noisy system. However, Resnik's argument above, that related words are ambiguous in different ways, applies to verbs also and so we would expect an accumulation of co-occurrences only in those cases where there is an actual association, assuming of course a sufficiently large training set is available to generate a detectable signal above the noise.

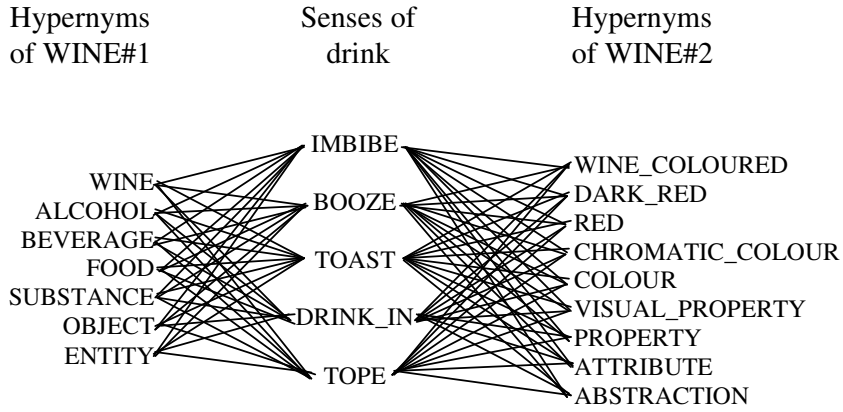


Fig. 7.2. SA is calculated between noun classes and verb classes.

Thus in Fig. 7.2, the *consumption* senses of drink (IMBIBE, BOOZE, TOPE) would associate with the WINE#1 senses of WINE, whereas the *cogitative* sense of drink (DRINK_IN) might associate more strongly with WINE#2.

The system depicted by Figure 7.2 is incomplete in that one symbol (drink) has been replaced with five synonyms (IMBIBE, BOOZE, TOAST, DRINK_IN, TOPE), to which each occurrence of drink contributes equally, resulting in five equal SA values; the problem of distinguishing between verb senses would remain. To overcome this, the subsuming senses of each sense of the verb must enter into the SA calculation (Fig. 7.3).

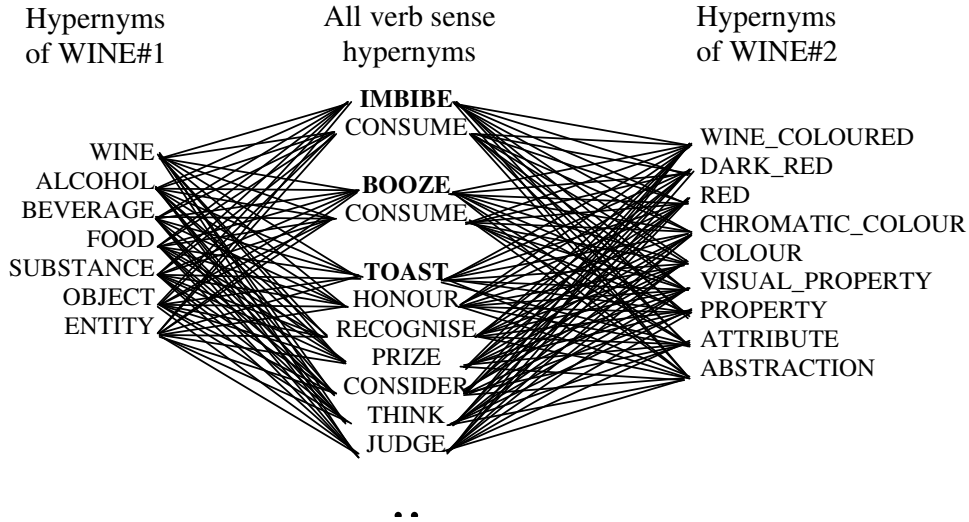


Fig. 7.3. SA is calculated between all subsuming classes of all noun and verb senses. Only first three senses of drink shown for clarity.

To facilitate this change in the model, the algorithm for calculating the conditional frequencies must be modified. The original conditional frequency (from which the conditional probability is derived) calculation is based upon observed predicate-argument co-ocurrences (x, w) as shown in equation (3) [RESN98]. The modification involves taking the predicate (x) and mapping it onto its set of WordNet classes, as shown in equation (4).

$$\text{freq}(x, c) = \sum_{w \in words(c)} \frac{\text{freq}(x, w)}{|\, classes(w) \,|} \tag{3}$$

$$\text{freq}(v, c) = \sum_{x \in words(v),\, w \in words(c)} \frac{\text{freq}(x, w)}{|\, classes(x) \,| \times |\, classes(w) \,|} \tag{4}$$

7.2 Generating the training data

The text corpus used to generate the Selectional Association (SA) values was the BNC Sampler [BNC], a 2% sample of the full BNC comprising one million written and one million transcribed spoken words. In order to calculate SA values, verb-subject, verb-object and verb-indirect object pairs must be extracted from the corpus[2]. As the BNC (and the sampler) are PoS tagged, but otherwise not marked-up for either grammatical structure or sense (Fig. 7.4), the grammatical structure necessary for verbal relation

```
<s n=0020><w PPY>You <w VM>will <w VVI>need <w AT1>a
<w JJ>Polish <w NN1>visa <w DDQ>which <w PPIS2>we <w VM>will
<w VVI>obtain <w IF>for <w PPY>you <c YSTP>.</s>
```

Fig. 7.4. Marked-up sentence from BNC Sampler (wrapped).

pairs to be extracted must first be imposed upon the corpus texts. To accomplish this, all mark-up was first removed from the sampler texts, resulting in reconstituted plain-text documents having one sentence per line (Fig. 7.5).

```
You will need a Polish visa which we will obtain for you.
```

Fig. 7.5. Reconstituted plain-text sentence from BNC Sampler.

[2] On reflection it would have been better to use just the written-word corpus as the paralinguistic utterances and colloquialisms of the spoken-word corpus appear to upset the grammar parser and hence affect the extracted predicate/argument pairs, as evidenced in Section 7.2.2

The cleaned-up documents were then parsed by MINIPAR [LIN93] in order to obtain the required grammatical structure (Fig. 7.6).

MINIPAR indicates relations in the central field of the parser output: subjects with the **:subj:** string, direct objects with the **:obj:** or **:obj1:** string, and indirect objects with the

```
fin       C:i:V          need
need      V:s:N          you
need      V:aux:Aux      will
need      V:subj:N       you
need      V:obj:N        visa
visa      N:det:Det      a
visa      N:mod:A        polish
visa      N:rel:C        fin
fin       C:whn:N        which
fin       C:i:V          obtain
obtain    V:s:N          we
obtain    V:aux:Aux      will
obtain    V:obj:N        which
obtain    V:subj:N       we
obtain    V:mod:Prep     for
for       Prep:pcomp-n:N you
```

Fig. 7.6. MINIPAR processed sentence from BNC Sampler

:obj2: string. The relation predicate (verb) is presented in the left hand field and the argument (noun) in the right. An AWK script was used to extract the predicate and argument from the parser output for the required relations, prefixing them with the characters ‘s’, ‘d’ and ‘i’ to indicate subject, direct-object and indirect-object respectively (Fig. 7.7). In total, 377651 relations were extracted.

```
d require visa
s need you
d need visa
d obtain which
s obtain we
```

Fig. 7.7. Extracted verbal relations

Possible WordNet senses of each verb and noun were determined through WordNet lookup, the senses being recorded as the sense location (SnsID) in the WordNet database (the 'hereiam' field of the synset structure). As each verb and noun may map onto more than one sense, the SnsIDs were concatenated by a semicolon (Fig 7.8), thereby retaining the three field {relation verb noun} structure.

```
d 1847498;1848500;588515 503611;494444
```

Fig. 7.8. Lemmas converted to WordNet SnsIDs

7.2.1 Assigning senses to pronouns

As WordNet does not contain senses for pronouns, these were assigned general senses wherever possible using the following algorithm, which also handles proper nouns and numerics in terms of WordNet classes:

```
if lemma in WordNet
    Assign the WordNet senses(s)
otherwise
    If lemma in {i, me, my, you, they, we, he, she, him, his, her, hers, their}
        Sense := SOMEBODY
    elseif lemma in {this, it}
        Sense := PHYSICAL_OBJECT;EVENT;LOCATION
    elseif lemma = 'that'
        Sense := PHYSICAL_OBJECT;EVENT
    elseif lemma = 'these'
        Sense = PHYSICAL_OBJECT
    elseif lemma = 'there'
        Sense := LOCATION
    elseif  lemma in {they, them, ours, our}
        Sense := GROUP;SOMEBODY
    elseif lemma is numeric
        Sense := NUMBER
    elseif PoS in {NNP, NNPS} (i.e. a proper noun)
        Sense := SOMEBODY;GROUP;LOCATION
    else FAIL
```

Fig. 7.9. Algorithm for assigning general senses to pronouns, numerics and proper nouns. For readability, SnsIDs have been replaced by text equivalents.

7.2.2 Failure Analysis

After WordNet sense assignment, 285746 useable sense relations were obtained from the 377651 lemma relations. Those 91905 (~24%) lemma relations failing the lemma to sense translation were written to a log for later examination. Due to the large number of failures, each of which would require individual scrutiny, only a cursory analysis was performed to reveal the general types of failure. These were:

1. Error in relation – MINIPAR imposes an incorrect grammatical structure. For example, in the extracted relation (s Graham phone), 'Graham' occupies the verbal position.

2. Unhandled exception – The algorithm for assigning senses to non verbs and nouns (Fig. 6) handles only relatively straightforward cases, and does not generalise the more difficult ones. For example, in the relation (d hate what), 'what' is not mapped to any WordNet sense.

3. PoS mismatch – There is disagreement between MINIPAR and WordNet with respect to the part of speech assigned to a word. For example, in the relation (s ugly cat), MINIPAR deems 'ugly' a verb, whereas in WordNet it is an adjective only.

4. Non-verbal utterances – As the spoken-word section of the BNC Sampler corpus was processed alongside the written-word section, it contains words like 'er', 'um', etc. MINIPAR attempts to parse sentences containing these, but any resultant extracted relation will contain the utterances, which are of course unknown to WordNet, for example: (s have erm).

5. Colloquialisms – WordNet does not account for dialectic or other cultural speech effects. As the transcription of the spoken-word is faithful to the speaker, it contains 'popular' words like 'wot', 'innit' etc that WordNet does not know, as in (s innit farmhouse).

7.2.3 Optimising the data for SA calculation

Recognising that processing 285746 data items, each consisting of multiple WordNet senses, would take some time to complete, a final optimisation of the data was made. As lines in the sense relation data prepared so far are repeated whenever the same verb and noun lemmas are extracted in a relation by MINIPAR, duplicates were removed and a count of the number of occurrences of that relation appended to the remaining instance (Fig. 7.10). This reduced the number of relations to process from 285746 to 112863, that is, by just over 60%.

```
d 101662;1791776;1789797;1863319 4123 2
d 101662;1791776;1789797;1863319 4123;14887 12
```

Fig. 7.10 Duplicate entries removed and count of instances appended.

7.2.4 Generation of Selectional Association values

SA data was generated using the modified Conditional Frequency value (eqn. 4 above) in the Selectional Preference and Selectional Association calculations, which were modified as follows:

- Calculate the *Selectional Preference* value $S_R(v)$ of each verb class, (v), in the verb taxonomy as:

$$S_R(v) = \sum \mathrm{p(c|v)} \log \frac{\mathrm{p(c|v)}}{\mathrm{p(c)}} \qquad (5)$$

- Calculate the *Selectional Association* value $A_R(v,c)$ of each verb-class/noun-class relation:

$$A_R(v,c) = \frac{1}{S_R(v)} \; p(c|v) \log \frac{p(c|v)}{p(c)} \qquad (6)$$

The algorithm for processing the data generated in Section 7.2 above is given in Figure 7.11, and is described as follows:

Step 2 of the algorithm extracts the data items from a line of the data set prepared as described above. Step 3 calculates the 'evenly distributed' contribution value, **contrib**, which is to be propagated along the hypernym taxonomy items.

Steps 4 and 5 select each verb-sense|noun-sense pair permutation (**vSns|nSns**) from those senses specified in the extracted data.

Given the structure of a training data record: {cnt, reln, vList, nList}

Where: cnt // number of occurrences
reln // the relation
vList // the list of verb senses
nList // the list of noun senses

```
1   foreach line of training data
2    extract reln, vList, nList, cnt from data
3    contrib := cnt / (|vList| * |nList|)
4    foreach vSns in vList
5     foreach nSns in nList
6      foreach vHyp Hypernym of vSns
7       foreach nHyp Hypernym of nSns
8        AddToFreq(reln, vHyp, nHyp, contrib)
9   CalcProbs
10  CalcPrefs
11  CalcAssocs
12  OutputAssocs
```

Fig. 7.11. Algorithm for calculating Selectional Associations from WordNet taxonomies

Similarly, steps 6 and 7 select each of the possible verb-hypernym|noun-hypernym sense permutations (**vHyp|nHyp**) of the verb|noun senses selected by steps 4 and 5, as depicted by Figure 7.3.

Step 8, the function **AddToFreq,** adds the calculated contribution **contrib** to a structure indexed on the given verb-sense|noun-sense pair for use in the posterior probability calculation, and also to another structure indexed on the noun-sense only, for the prior probability calculation.

Finally, steps 9 to 11 use the accumulated **contrib** values calculated above in both structures to calculate the Probability, Selectional Preference, and Selectional Association values according to equations 1..4 above.

The calculated Selectional Association values for each of the *subject*, *direct-object*, and *indirect-object* relations were written to separate files, using SGML markup for portability, and in the order of increasing verb *id* to facilitate searching on verb. A fragment of the direct-object relation file is presented in Fig. 7.12, and the markup tags used in Fig. 7.13.

```
<p id=795711  sp=1.014891 >
    <a id=1740 cnt=21.759257 Ppre=0.088637 Ppost=0.111248 sa=0.024906 /a>
    <a id=2086 cnt=9.250474 Ppre=0.034328 Ppost=0.047295 sa =0.014933 /a>
    <a id=3731 cnt=8.074447 Ppre=0.033161 Ppost=0.041282 sa =0.008911 /a>
    <a id=4123 cnt=7.867814 Ppre=0.032458 Ppost=0.040226 sa =0.008504 /a>
    <a id=8019 cnt=0.684891 Ppre=0.001100 Ppost=0.003502 sa =0.003995 /a>
    …
    <a id=10972097 cnt=0.040000 Ppre=0.000022 Ppost=0.000205 sa =0.000445 /a>
    <a id=10972592 cnt=0.040000 Ppre=0.000004 Ppost=0.000205 sa =0.000785 /a>
    <a id=10978183 cnt=0.014286 Ppre=0.000015 Ppost=0.000073 sa =0.000114 /a>
    <a id=10980504 cnt=0.014286 Ppre=0.000089 Ppost=0.000073, sa =-0.000015 /a>
</p>
```

Fig. 7.12. Sample of SA calculation output for direct-object of verb **drink** (imbibe)

The tags used are as follows:

p	: preference structure
id	: WordNet sense id (hereiam)
sp	: selectional preference of predicate
cnt	: the calculated frequency f(c|v)
Ppre	: prior distribution probability p(c|v)
Ppost	: posterior distribution probability p(c|v)
sa	: selectional association of predicate for argument)
a	: argument

Fig. 7.13. Tags used in markup of calculated SA output.

7.2.5 The two training datasets

The above describes the creation of a training dataset using the entire WordNet noun and verb hypernym taxonomies. The evaluation of Specialisation Classes in a WSD task will of course require an equivalent Selectional Association training dataset built around the abridged taxonomies, that is, taxonomies consisting only of SCs. This is accomplished by a small modification to the algorithm given in Fig. 7.11 in lines 6 and 7, instead of iterating each hypernym of the given verb and noun sense, the algorithm iterates each specialisation class of those senses. The modified algorithm is shown in Figure 7.14:

```
1   foreach line of training data
2    extract cnt, reln, vList, nList from data
3    contrib := cnt / (|vList| * |nList|)
4    foreach vSns in vList
5     foreach nSns in nList
6      foreach vSC SpecClass of vSns
7       foreach nSC SpecClass of nSns
8        AddToFreq(reln, vSC, nSC, contrib)
9   CalcProbs
10  CalcPrefs
11 CalcAssocs
12 OutputAssocs
```

Fig. 7.14. Algorithm for calculating Selectional Associations from abridged WordNet taxonomies

7.3 Generating the Evaluation data

The SemCor [FELL98] semantically-tagged corpus will provide the evaluation data; being already marked-up with WordNet senses SemCor provides a 'gold standard' for WSD task evaluations. Like the BNC, SemCor contains no structural information and so cannot be used to generate verb-sense|noun-sense pairs directly.

The SUSANNE Corpus [SAMS95], like SemCor, is a freely available subset of the Brown Corpus, and has been manually marked-up for grammatical structure. As there is a 32-document overlap between SemCor and SUSANNE, it is possible to create a sense-tagged and grammatically marked-up corpus from this intersection.

```
A01:0010.06     -       AT      The     the     [O[S[Nns:s.
A01:0010.09     -       NP1s    Fulton  Fulton  [Nns.
A01:0010.12     -       NNL1cb          County  county  .Nns]
A01:0010.15     -       JJ      Grand   grand   .
A01:0010.18     -       NN1c    Jury    jury    .Nns:s]
A01:0010.21     -       VVDv    said    say     [Vd.Vd]
A01:0010.24     -       NPD1    Friday  Friday  [Nns:t.Nns:t]
A01:0010.27     -       AT1     an      an      [Fn:o[Ns:s.
A01:0010.30     -       NN1n    investigation   investigation   .
A01:0020.03     -       IO      of      of      [Po.
A01:0020.06     -       NP1t    Atlanta Atlanta [Ns[G[Nns.Nns]
A01:0020.09     -       GG      +<apos>s        -       .G]
A01:0020.12     -       JJ      recent  recent  .
A01:0020.15     -       JJ      primary primary .
A01:0020.18     -       NN1n    election        election        .Ns]Po]Ns:s]
A01:0020.21     -       VVDv    produced        produce         [Vd.Vd]
A01:0020.24     -       YIL     <ldquo>         -       .
A01:0020.27     -       ATn     +no     no      [Ns:o.
A01:0020.30     -       NN1u    evidence        evidence        .
A01:0020.33     -       YIR     +<rdquo>        -       .
A01:0020.39     -       CST     that    that    [Fn.
A01:0030.03     -       DDy     any     any     [Np:s.
A01:0030.06     -       NN2     irregularities  irregularity    .Np:s]
A01:0030.09     -       VVDv    took    take    [Vd.Vd]
A01:0030.12     -       NNL1c   place   place   [Ns:o.Ns:o]Fn]Ns:o]Fn:o]S]
```

Fig 7.15. Fragment of the SUSANNE Corpus.

SUSANNE uses a flat, six field, one word per line format, an example being shown in Figure 7.15. Structural information is encoded in the sixth (parse) field, which includes logical and surface indicators for verb subject and direct and indirect objects.

SemCor on the other hand expresses one sense per line, and uses SGML tags to identify line components such as word, lemma, PoS, sense etc. Figure 7.16 presents the SemCor representation of the sentence shown in 7.15.

```
<s snum=1>
<wf cmd=ignore pos=DT>The</wf>
<wf cmd=done rdf=group pos=NNP lemma=group wnsn=1 lexsn=1:03:00::
    pn=group>Fulton_County_Grand_Jury</wf>
<wf cmd=done pos=VB lemma=say wnsn=1 lexsn=2:32:00::>said</wf>
<wf cmd=done pos=NN lemma=friday wnsn=1 lexsn=1:28:00::>Friday</wf>
<wf cmd=ignore pos=DT>an</wf>
<wf cmd=done pos=NN lemma=investigation wnsn=1 lexsn=1:09:00::>investigation</wf>
<wf cmd=ignore pos=IN>of</wf>
<wf cmd=done pos=NN lemma=atlanta wnsn=1 lexsn=1:15:00::>Atlanta</wf>
<wf cmd=ignore pos=POS>'s</wf>
<wf cmd=done pos=JJ lemma=recent wnsn=2 lexsn=5:00:00:past:00>recent</wf>
<wf cmd=done pos=NN lemma=primary_election wnsn=1
    lexsn=1:04:00::>primary_election</wf>
<wf cmd=done pos=VB lemma=produce wnsn=4 lexsn=2:39:01::>produced</wf>
<punc>``</punc>
<wf cmd=ignore pos=DT>no</wf>
<wf cmd=done pos=NN lemma=evidence wnsn=1 lexsn=1:09:00::>evidence</wf>
<punc>"</punc>
<wf cmd=ignore pos=IN>that</wf>
<wf cmd=ignore pos=DT>any</wf>
<wf cmd=done pos=NN lemma=irregularity wnsn=1 lexsn=1:04:00::>irregularities</wf>
<wf cmd=done pos=VB lemma=take_place wnsn=1 lexsn=2:30:00::>took_place</wf>
<punc>.</punc>
</s>
```

Fig 7.16 Fragment of SemCor

As the addition of structural information to SemCor is more difficult than adding sense data to SUSANNE (achieved by simply adding fields), SUSANNE was augmented by senses drawn from SemCor. The sentence shown above is displayed in its final sense-tagged form in Fig. 7.19, and also shows that the sense field for closed-class words is, in SUSANNE style, filled by a hyphen (unless it is an easily identified personal pronoun).

However, before attempting to sense-tag SUSANNE, the structural and representational differences between the two corpora must be considered.

7.3.1 Unique representation of WordNet Sense Keys

The word sense information of SemCor is encoded by the fields *lemma* and *lexsn*, which must be combined to form a WordNet *Sense Key* of the form *lemma%lexsn*. Thus the Sense Key of the following SemCor entry is irregularity%1:04:00.

```
<wf cmd=done pos=NN lemma=irregularity wnsn=1 lexsn=1:04:00::>irregularities</wf>
```

The WordNet API provides search functions to locate the WordNet entry identified by any Sense Key. To make location of WordNet entries faster, and to provide a more compact representation, we propose that each WordNet Sense Key be transformed into a Sense Identifier String (SIS) of form *DB*, where *D* is a single character drawn from {n,v,j,r} signifying the noun, verb, adjective or adverb (resp.) database to which the Sense Key points, and *B* the byte offset within that database at which the entry indexed by the Sense Key begins[3], and referred to previously as the SnsID. Thus the Sense Key irregularity%1:04:00 becomes the SIS 'n475542'. Such a representation has an advantage over Sense Keys when directly comparing senses of synonyms as it eliminates all traces of the surface form from the representation. For example the sense of *irregularity* pointed to by the Sense Key irregularity%1:04:00 is also pointed to by its synonym abnormality%1:04:00; the SIS identifies the senses as identical, whereas the WordNet Sense Keys do not.

[3] The byte offset (WordNet *hereiam* field) is included as part of the WordNet synset structure as supplied by Princeton University and is commonly employed by researchers using WordNet to discriminate between synsets of homonyms. The byte offset is not transportable between different knowledge bases or even between different versions of WordNet.

7.3.2 Compounds

SemCor represents compounds by concatenating the lemmas of the constituent words by use of the underscore character. Each SemCor compound has been assigned the most appropriate sense from WordNet. Thus SemCor presents one sense per line, as may be seen by reference to Figure 7.16, particularly the entry for the compound verb took_place. SUSANNE, on the other hand, presents one word per line, representing compounds by bracketing, for example the bracketing of 'The Fulton County Grand Jury' in Fig. 7.15. This presents a structural mismatch between the two corpora that must be resolved.

Dealing with SemCor compounds is relatively simple: by splitting each SemCor compound into its constituent words and assigning the compound's SIS to each constituent, a 1:1 corespondence between SemCor and SUSANNE is achieved, and the SemCor SIS may then be appended to the appropriate SUSANNE line. For completeness, an additional index field is added to each SUSANNE entry: a hyphen for closed-class words other than easily identifiable personal pronouns, a zero for simple words, and an incrementing number, beginning at one, for consecutive words from a SemCor compound, resulting in the structure shown in Fig. 7.17. An example of the final encoding is shown in Figure 7.19.

```
A01:0030.03 - DDy any any [Np:s. - -
A01:0030.06 - NN2 irregularities irregularity .Np:s] 0 n475542
A01:0030.09 - VVDv took take [Vd.Vd] 1 v235191
A01:0030.12 - NNL1c place place [Ns:o.Ns:o]Fn]Ns:o]Fn:o]S] 2 v235191
```

Fig. 7.17 SUSANNE corpus with appended compound-count and WordNet sense indicating fields

7.3.3 An algorithm for appending sense indicators to SUSANNE

The 33 overlapping documents were processed using the following algorithm:

```
For each SemCor document represented in SUSANNE
    Open the corresponding SUSANNE document
    For each line in SemCor document
        If the line expresses a closed-class word
            Set strLemma to the closed class word
            If the word is a singular pronoun
                Set strCompCnt to 0
                Set strSIS to the SIS of 'Person'
            Else If the word is a plural pronoun
                Set strCompCnt to 0
                Set strSIS to the SIS of 'Group'
            Else
                Set both strCompCnt and strSIS to "-"
            SuzAppend(strLemma, strCompCnt, strSIS)
        Else
            Set strSIS to correspond to the SemCor Sense Key
            Set strCompCnt to "0"
            If word is not compound
                Set strLemma to the encoded lemma
                SuzAppend(strLemma, strCompCnt, strSIS)
            Else
                For each constituent word of the compound
                    Increment strCompCnt
                    Set strLemma to the current constituent word
                    SuzAppend(strLemma, strCompCnt, strSIS)
```

Fig. 7.18 Algorithm for appending WordNet sense information to SUSANNE

The function SuzAppend(`strLemma`, `strCompCnt`, `strSIS`) seeks strLemma in the current SUSANNE file, starting at the current position and stepping down for up to five lines until either a match is found, in which case `strCompCnt` and `strSIS` are appended, or not found, in which case they, along with the SUSANNE line number (i.e. the first SUSANNE field) of the current line are written to an error log. Limiting the search to five lines from the current position allows the stepping-over of the SUSANNE -specific mark-up lines, such as <bmajhd>, <emajhd>, <majbrk>, <minbrk>, whilst limiting the possibility of accidental matches caused by descending too far into the document. The error log, which contained around 200 entries after executing the above

algorithm, was used to guide a manual repair of the resulting sense tagged SUSANNE corpus.

The result of the above is a structurally annotated SUSANNE corpus augmented by two additional fields that indicate compound constituency and WordNet sense, as shown in Figure 7.19 below:

```
A01:0010.06  -    AT     The          the          [O[S[Nns:s.      -        -
A01:0010.09  -    NP1s   Fulton       Fulton       [Nns.            1        n17954
A01:0010.12  -    NNL1cb              County       county           .Nns]    2        n17954
A01:0010.15  -    JJ     Grand        grand        .                3        n17954
A01:0010.18  -    NN1c   Jury         jury         .Nns:s]          4        n17954
A01:0010.21  -    VVDv   said         say          [Vd.Vd]          0        v682542
A01:0010.24  -    NPD1   Friday       Friday       [Nns:t.Nns:t]    0        n10883362
A01:0010.27  -    AT1    an           an           [Fn:o[Ns:s.      -        -
A01:0010.30  -    NN1n   investigation investigation .             0        n4470360
A01:0020.03  -    IO     of           of           [Po.             -        -
A01:0020.06  -    NP1t   Atlanta      Atlanta      [Ns[G[Nns.Nns]   0        n6609298
A01:0020.09  -    GG     +<apos>s     -            .G]              -        -
A01:0020.12  -    JJ     recent       recent       .                0        j1666848
A01:0020.15  -    JJ     primary      primary      .                1        n119389
A01:0020.18  -    NN1n   election     election     .Ns]Po]Ns:s]     2        n119389
A01:0020.21  -    VVDv   produced     produce      [Vd.Vd]          0        v1465746
A01:0020.24  -    YIL    <ldquo>      -            .                -        -
A01:0020.27  -    ATn    +no          no           [Ns:o.           -        -
A01:0020.30  -    NN1u   evidence     evidence     .                0        n4485976
A01:0020.33  -    YIR    +<rdquo>     -            .                -        -
A01:0020.39  -    CST    that         that         [Fn.             -        -
A01:0030.03  -    DDy    any          any          [Np:s.           -        -
A01:0030.06  -    NN2    irregularities irregularity .Np:s]        0        n475542
A01:0030.09  -    VVDv   took         take         [Vd.Vd]          1        v235191
A01:0030.12  -    NNL1c  place        place        [Ns:o.Ns:o]Fn]Ns:o]Fn:o]S]    2    v235191
```

Fig. 7.19 Structure of the sense-tagged SUSANNE corpus.

7.3.4 Selecting the test data

Having created a sense-tagged version of the SUSANNE corpus, all that remains is to extract sense-relation data from it. This is accomplished by seeking logical subject tags (s), logical object tags (o), surface subject tags (S), surface object tags (O), logical indirect object tags (i), verb tags (V), and pronoun tags (p) within clauses identified by the clause tags: S, F, T, Z, L, A, and W. Passive verbs are marked by the tag ‘p’, which

here is used to indicates that, should the verb's arguments be marked as surface, then its arguments should switch roles, that is, the object become the subject and the subject the object. In total, 32658 relations were extracted from the sense-tagged SUSANNE corpus, consisting of 18397 verb-subject, 13870 verb-object, and 391 verb-indirect object relations. Each relation was written to its own file, again using the {relation, verb sense-list, noun sense-list} structure shown in Fig. 7.8.

Following Resnik [RESN98], the evaluation will be performed on the verb-direct object relation, and a random sample of 331 items from the data extracted above was automatically selected, comprising approximately 13 items for each alphabetic character on the basis of the verb's initial character.

7.4 Comparing WSD Performance

After construction of the WordNet and Abridged WordNet training datasets and the evaluation dataset as described above, the evaluation was performed as follows (using 'Z' notation):

Given:

[NLEMMA]	The set of WordNet noun lemmas
[VLEMMA]	The set of WordNet verb lemmas
[NCLASS]	The set of WordNet noun classes
[VCLASS]	The set of WordNet verb classes
assocSns: ℙ(NCLASS x VCLASS)	associated noun/verb senses
nSensesOf: NLEMMA → ℙNCLASS vSensesOf: VLEMMA → ℙVCLASS	Functions to return the senses of the given noun/verb lemma
nHypSenses: ℙNCLASS → ℙNCLASS vHypSenses: ℙVCLASS → ℙVCLASS	Functions to return all subsuming classes of the noun/verb senses.

MaxSA: $(\mathbb{P}NCLASS, \mathbb{P}VCLASS) \rightarrow assocSns$ Function to return the most strongly associated noun and verb classes.

The nSensesOf function uses the findtheinfo_ds function of the WordNet API to return all WordNet noun senses associated with the supplied noun lemma; vSensesOf is similarly defined.

The nHypSenses function is defined as follows, using the hypernym function H from Section 6.2. The vHypSenses function is similarly defined:

nc: $\mathbb{P}NCLASS$ Senses of a noun lemma

nHypSenses (nc) $\widehat{=}$

$$\exists\, h: NCLASS \bullet \forall\, c: NCLASS; x: \mathbb{N} \mid c \in nc, x \geqslant 0 \bullet h = H^x(c)$$

The MaxSA function is defined as follows, using a function FindSA(n,v) which looks up the Selectional Association value between the supplied arguments in the calculated selectional association dataset for a given relation:

nh: $\mathbb{P}NCLASS$ Hypernym senses of a noun lemma
vh: $\mathbb{P}VCLASS$ Hypernym senses of a verb lemma

MaxSA(nh, vh) $\widehat{=}$

$$\exists (n: NCLASS, v: VCLASS) \mid n \in nh, v \in vh \bullet MAX(FindSA(n, v))$$

Sense disambiguating the lemmas from each line of evaluation data, within the context of a relation, is then a matter of applying the given functions as follows:

nl: NLEMMA a noun lemma
vl: VLEMMA a verb lemma
result: assocSns set of most associated senses

result = MaxSA(nHypSenses(nSensesOf (nl)), vHypSenses(vSensesOf(vl)))

7.4.1 Metrics

The basic comparison was the accuracy to which the noun and verb senses given on each line of the evaluation data were reproduced by application of the MaxSA function to the noun and verb lemmas on that line, and this was made by the Recall/Precision measure.

The second metric was the time taken to perform the evaluation, which is relevant to the need for real-time performance. Note that due to the large size of the SA data files calculated from the BNC (the verb-object SA data for the full WordNet hypernym taxonomy is over 631Mb, whilst the equivalent for the abridged WordNet is nearly 4 Mb), the SA data could not be read into memory. Both datasets therefore remained as disk files, and retrieval of specific association data involved a binary search and consequent disk-access time penalty.

The third metric was the number of comparisons made, that is, how many times the FindSA function was called.

7.4.2 Results

The results of applying the MaxSA function to the evaluation data for both the original WordNet SA values and the abridged WordNet SA values are presented in Table 7.1 below.

The table shows an across the board improvement in a WSD task, using the Specialisation Classes based abridged WordNet hypernym taxonomy over the original WordNet hypernym taxonomy, where Selectional Association is the disambiguation method; fewer noun and verb senses are recalled using SCs, but more of them are correct, verbs showing the greatest improvement.

Metric	WordNet SA	Abridged SA
Nouns recalled	667	551
Nouns correct	189	198
Noun Recall	57.10%	59.82%
Noun Precision	28.34%	35.93%
Verbs recalled	474	356
Verbs correct	92	114
Verb Recall	27.79%	34.44%
Verb Precision	19.41%	32.02%
Time	20min 48s	2min 40s
Comparisons	219,481	31,962

Table 7.1. Results of Selectional Association-based WSD evaluation using original and abridged WordNet. Both WordNet and Abridged SA data stored on disk.

The improvement in recall and precision is perhaps more interesting in the context of execution time and number of comparisons performed; the original WordNet taxonomy allows on average 0.27 disambiguations to be performed per second, whereas the abridged version manages 2.07. Similarly, 880.61 comparisons are made per disambiguation using the original taxonomies, whereas only 96.56 are required by the abridged version. Considering the size of the two SA datasets as well, 631Mb vs. 4Mb, the SC based disambiguator has produced better results by doing far less work with much less (but presumably higher quality) knowledge. This is an encouraging result as the reduction in workload brings the desired real-time processing ever closer.

Correct	Original WordNet SA	Abridged WordNet SA
Neither	104	101
All Nouns	189	198
All Verbs	92	114
Noun + Verb	54	82

Table 7.2. Breakdown of disambiguation results.

As Table 7.1 shows, the number of nouns and verbs correctly disambiguated are different using both datasets, meaning that in some cases, one is disambiguated correctly whilst the other is not. This information is presented in Table 7.2.

The table shows very little difference in the number of entirely incorrect disambiguations between the original and abridged WordNet SA methods. However, the number of correct noun or verb disambiguations is higher for the abridged version, as is the number of disambiguations where both noun and verb are both correct.

7.5 Conclusions

The results of the WSD exercise show that Selectional Association between Specialisation Classes is able to more accurately model the association between a verb and an argument than an equivalent system built around the full set of WordNet senses. We propose that this can be explained by consideration of the selection mechanism: in general, disambiguations are made by identifying from the possible senses those that associate most strongly, those senses being identified by novel Sense Indicator Strings. When using the original taxonomies, the simple maximum SA value is found, that is, every permutation of possible senses is explored and the pair(s) with the highest overall SA value is/are returned. Using the abridged taxonomies however, the search is limited to a subset of the possible senses that have been previously identified as exhibiting the greatest degree of sense specialisation within their hypernym chain. By exploring associations between SCs only, the system is able to ignore the simple maximum and instead seek the maximum SA between highly sense-discriminatory senses.

From the evidence presented in Chapter 6, that SCs are able to both compactly represent the senses of the fine-grained WordNet and retain the sense distinctions of the lemmas contained therein, and from Chapter 7, that SCs are better able to model Selectional Association between a predicate and argument and greatly reduce the number of calculations necessary to disambiguate noun/verb pairs, it is concluded that a cognitively-oriented language understanding system with a requirement for real-time operation will benefit from the use of Specialisation Classes as the underlying sense-representation. Note however that we do not consider a Specialisation Class to be the *actual* representation – what can 1595188 tell us about dogs? – but more of a *surrogate* or *primary key* that activates some representational structure.

8 The Grammar Element

This chapter proposes that the standard, discrete CG syntactic categories are an impediment to incremental interpretation, and hence to the early resolution of sense and structural ambiguity needed to minimise the workload of the Construction Integration Model. Type-raising is shown not to be a viable solution, and the grammatical theories of Chapter 4 are revisited for inspiration, and we propose that the acquisition of grammar through configuration of an innate language facility provides those insights: A calculation of the syntactic category problem space size is presented and used in a comparison with the number of categories extracted from a configured system (English verbs). The comparison not only shows that that a very small proportion of the innate (unconfigured) category problem space is actually expressed in a configured system, but also that the configured categories exhibit structural inheritance. This finding is used to formulate a view of syntactic categories expressed as trees rather than distinct categories. Using this new 'Inheritance Model of Syntactic Categories' it is shown that incremental interpretation is possible in that partial semantic interpretations can be produced incrementally, and without modification to the overall CG derivation process. In order to achieve this, the description of semantic categories is extended to incorporate a notion of sense.

Chapters 5 and 6 have presented two elements of the pre-processor needed to bridge the gap between printed (or electronic) text and the logical representation of that text required by the Construction Integration Model. These are a chunking model that allows parallel shifts into the parser, and compact sense representations in the form of abridged

WordNet noun and verb taxonomies. Parallel shifting supports the expression of alternative word groupings (compounds) within the parser, necessary because identification of the most appropriate grouping can require integration with additional knowledge supplied by long term memory, an activity performed by the Construction Integration Model proper. Sense representations are needed by the pre-processor because coherence is taken as the mechanism by which the Construction Integration Model builds *truthful* (in terms of reflecting real-world objects, actions and relations) interpretations of input text, and, as has been shown in Chapter 4, coherence is sought between senses, not surface forms. The description of the pre-processor to the CIM is not yet complete as the grammar necessary to drive the logical-form construction as performed by the chart parser has not yet been included.

8.1 Lexicalised Grammars

As CG is a lexicalised grammar, the surface form of a word is used as a key into a lexicon that associates surface forms with syntactic and semantic categories. After lookup, the categories are retrieved and shifted into the chart parser. CCGBank [HOCK03] is a wide-coverage set of automatically extracted syntactic categories that has been shown to perform well in evaluations [CLAR02], Table 8.1 shows the fragment of CCGBank associated with the surface-form ‘drink’, and presents categories associated with three parts-of-speech: common, singular or mass noun (NN), base form of verb (VB), and present tense, singular (not 3rd person) verb (VBP). Parts-of-speech are expressed using the Penn-Treebank tag-set.

Term	PoS	Syntactic Category
drink	NN	N
drink	VB	(S[b]\NP)/NP
drink	VB	(S[b]\NP)/PP
drink	VB	S[b]\NP
drink	VBP	(S[dcl]\NP)/NP
drink	VBP	(S[dcl]\NP)/PP
drink	VBP	S[dcl]\NP

Table 8.1. Fragment of CCGBank showing PoS and Syntactic Categories for the word 'drink'

When presented with the word 'drink' from a sentence, the seven categories from Table 8.1 are shifted into the chart. The CKY algorithm then combines these categories with those shifted previously, along with those to come, through application of the combinatory rules to ideally arrive at a single derivation for the whole sentence. That derivation will therefore involve just one of the seven categories from Table 8.1.

The process described above is reasonable and, given appropriate lexical and grammatical resources, readily implemented. However, as it produces sentential parses in a non-incremental fashion, as demonstrated by Figure 8.1, we question the accuracy of the procedure as an analogue of the cognitive process employed by humans when reading, for whom the experience of reading *is* incremental.

8.2 Incremental Interpretation

Intuitively, humans appear to process sentences incrementally, that is, as each word is read it is added to the interpretation of the sentence read so far, thereby building up an interpretation incrementally. Incremental processing of sentences has received some attention, for example Milward describes Dynamic Dependency Grammars as a means

to provide incremental parsing and interpretation for Lexicalised Dependency Grammars [MILW92], and Costa et al propose a recursive neural network, trained on a parsed corpus (Penn Treebank), to predict the correctness of partial syntactic structures as they are discovered in incremental fashion [COST01].

The CKY algorithm initially appears to follow the incremental interpretation intuition because words are shifted in to it starting with the first word of a sentence and ending with the last. However, it is easily demonstrated that the chart parser does not build

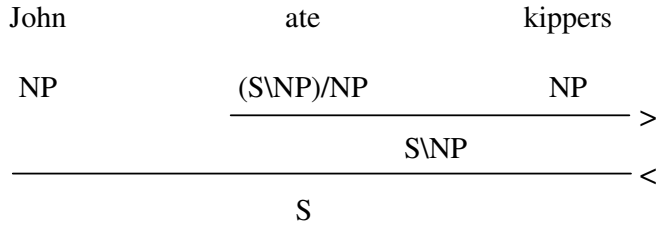


Fig. 8.1. The parse of ‘John ate kippers’ does not proceed incrementally.

interpretations incrementally by parsing the simple sentence shown in Figure 8.1. Note that the structure of the verbal category dictates that the verb ‘ate’ must first combine with its object ‘kippers’ - the last word of the sentence - *before* the first word ‘John’

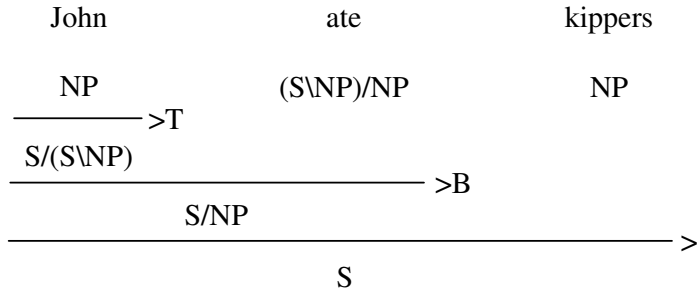


Fig. 8.2. Type-raising can force incremental parsing.

enters into any combination, resulting in a non-incremental parse. This is counter-intuitive as it is not necessary for humans to read the word 'kippers' before understanding that 'John ate'.

Type-raising may be used to coerce incremental parsing, as shown in Figure 8.2. The type-raised category S/(S\NP) of 'John' combines with the verbal category (S\NP)/NP via the forward functional composition rule (>B) to give S/NP, which in turn combines with the NP category of 'kippers' and results in an incremental parse; the right-branching interpretation of Fig. 8.1 has been converted into a left-branching interpretation in Fig. 8.2. Of course, to apply type-raising freely would result in all manner of new derivations in the chart, some desirable, other not, and ultimately would lead to overgeneration. A further argument against type-raising freely is that to do so would result in additional resource usage to support and evaluate the multitude of type-raised categories and their derivations, impacting on the memory/processing limitations of the human cognitive system.

In the following sections we demonstrate that the Inside-Out theory of grammar acquisition, in particular the *configuration* of an innate language system to a local linguistic environment, provides insights into the problem of incremental interpretation with respect to syntactic categories

8.3 Configuration

In Section 4.7.4 it was shown by comparison of English and German transitive verbs that CG embodies the notion of configuration in its syntactic categories. Here we follow

the argument presented earlier, that the syntactic abilities of the innate language system must encompass all human languages, by proposing that the syntactic component of the innate language system be viewed as a problem space, and that configuration comprises the selection/activation of certain points within that space. Firstly, the degree of syntactic configuration necessary for any human language may be found by comparing the size of the innate problem space with the number of categories expressed by that language.

8.3.1 Size of problem space

The size of the syntactic category problem space may be calculated for complex categories composed of a number, *a*, of atomic categories by consideration of the components of a complex category:

1. There are three atomic categories: NP, PP and S;
2. There are two functor slashes '\' and '/';
3. There are two brackets '(' and ')';

Given that the structure of a complex category may be recursively defined as:

$(\alpha|\beta)$ where α and β are atomic or complex categories,
and | a slash functor.

The above definition states that a complex category always consists of two components, which may themselves be atomic of complex. Two components, separated by a slash, shall be referred to here as a *binary bracketing*.

The calculation must take the following factors into consideration:

1. The number of atomic categories in any given complex category (henceforth the *length*);
2. The number of binary bracketings of any category of given length;
3. The number of categories of a given length that may be generated from the set of atomic categories;
4. The number of variations of the bracketed string due to the slash operators.

The number of binary bracketings of a string of length *a* is given by its *Catalan Number*, C(n), calculated as:

$$C(n) = \frac{(2n)!}{(n+1)!n!} \qquad \text{where } n = a - 1$$

Thus for strings of length 3, the Catalan Number is:

$$C(2) = \frac{(4)!}{3!(2)!} = \mathbf{2}$$

i.e. ((x x) x) and (x (x x))

Given that there are three atomic categories, the number of permutations of *a* atomic categories, P(*a*), may be calculated as:

$$P(a) = 3^a$$

So for complex categories of length 3, the number of strings generated is:

$P(3) = 3^3 = \mathbf{27}$ i.e.

xxx	yxx	zxx
xxy	yxy	zxy
xxz	yxz	zxz
xyx	yyx	zyx
xyy	yyy	zyy
xyz	yyz	zyz
xzx	yzx	zzx
xzy	yzy	zzy
xzz	yzz	zzz

Finally, as the slashes interleave with the atomic categories, there are a-1 slashes in a category of length a. As the 'alphabet' here consists of only two symbols, the number of category variations due to slashes, $V(a)$, is calculated as:

$$V(a) = 2^{a-1}$$

Again, for a category of length 3, the number of slashed categories generated is:

$V(3) = 2^{3-1} = 2^2 = \mathbf{4}$ i.e. //, /\, \/, \\

giving x/x/x, x/x\x, x\x/x, x\x\x, …

The size of the language encompassing all complex categories of length a, $L(a)$, that is, the size of the problem space, is given by the product of the Catalan number for a, the number of atomic category combinations for a, and the number of slash-variants for a:

$$L(a) = C(a\text{-}1) * P(a) * V(a)$$

Thus for strings of length 3, the size of the problem space is therefore:

$$L(a) = C(3\text{-}1) * 3^3 * 2^{3-1}$$

$$= 2 * 27 * 4$$

$$= \mathbf{216}$$

Thus 216 unique bracketed categories are possible using any three of the three atomic categories and any two of the two slashes.

8.3.2 Problem space size for given category lengths

As a grammar is not limited to complex categories of any one length (CCGBank ranges from 2 to 48 atomic categories in any complex category), the total problem space, T(*a*), includes all categories up to and including a given length:

$$T(a) = \sum_{n=1}^{a} L(n)$$

The size of the problem space, that is, the total number of grammatical categories, for category lengths 1 to 8 are listed in Table 8.1:

Category Length (*a*)	C(*a*)	P(*a*)	V(*a*)	L(*a*)	T(*a*)
1	1	3	1	3	3
2	1	9	2	18	21
3	2	27	4	216	237
4	5	81	8	3240	3477
5	14	243	16	54432	57909
6	42	729	32	979776	1037685
7	132	2187	64	18475776	19513461
8	429	6561	128	360277632	379791093

Table 8.1. Problem space size for categories of up to 8 atomic categories

8.3.3 Problem space reduction through merging of N and NP

In Section 5.5 an argument was presented to justify the merging of the N and NP categories, and it was reasoned that this would result in a reduction in the number of categories necessary to parse a language such as English. Using the above calculation, it is possible to show the degree to which the total problem space has been reduced. Table 8.2 presents total problem space sizes for grammars using three and four atomic

categories for category lengths of 8 or less, the problem space reduces from approximately 3.7 billion to around 380 million categories.

Category Length (*a*)	T(*a*) 3 atoms	T(*a*) 4 atoms
1	3	4
2	21	36
3	237	548
4	3477	10788
5	57909	240164
6	1037685	5745188
7	19513461	144157220
8	379791093	3742870052

Table 8.2. Comparison of problem space sizes when using 3 and 4 atomic categories.

8.3.4 Comparison of Innate and Configured syntactic problem space

CCGBank (sections 2-21) provides the raw syntactic category data for the configured problem space. As the syntactic categories of CCGBank include the N and NP atomic categories, all instances of N were replaced by NP, thereby aligning the grammar with the three-atom system in use here. Additionally, all category features were eliminated for clarity, as they play no part in this argument. Finally, following [CLAR02], all categories with a frequency < 10 were ignored. Eliminating duplicate categories from the resultant set showed that only 155 unique syntactic classes were expressed, each of which comprised between 1 and 4 four atomic categories.

The 155 categories of maximum length 4 occupy approximately 4.46% of the 3477 (from Table 8.2) points in the innate problem space for categories of length 4 or less. These results show that only a small proportion of the problem space categories are actually needed in a basic description of the syntax of English. However, the results say nothing about why any particular category is selected over another. The driving force behind configuration is, according to the Principles and Parameters theory [CHOM81],

the external linguistic environment; exposure to certain categories in the environment leads to those categories being selected from the set of innate categories. This explanation is convenient, but still does not explain why any particular category is fit for inclusion in a grammar of a natural language. To answer this question, it is necessary to examine the selected categories themselves.

8.3.5 Selection of syntactic categories for a grammar

Two facts relevant to category selection are that:

1. Languages can be described in terms of the order in which verbal arguments are presented relative to the verb: English is an SVO (subject-verb-object) language, in which the verb is placed after the subject and before the object, whereas German is an SOV language, placing the verb after the subject and object;
2. Verbs can be described in terms of the number of arguments they take; Intransitive verbs take the subject only, Transitive a subject and direct-object, and Ditransitive a subject, direct-object and indirect object. The argument order of point 1 above is adhered to throughout for a given language.

Both points are illustrated by consideration of some basic verbal categories for intransitive, transitive, ditransitive, and ditransitive-preposition categories:

Intransitive:	S\NP	SV
Transitive:	(S\NP)/NP	SVO
Ditransitive:	((S\NP)/NP)/NP	SVO
Ditransitive-PP:	((S\NP)/PP)/NP	SVO

Shown this way, it is evident from the categories presented that the intransitive category is the leftmost constituent of the transitive category, itself the leftmost constituent of the ditransitive category, that is, the ditransitive form inherits the transitive form, which in

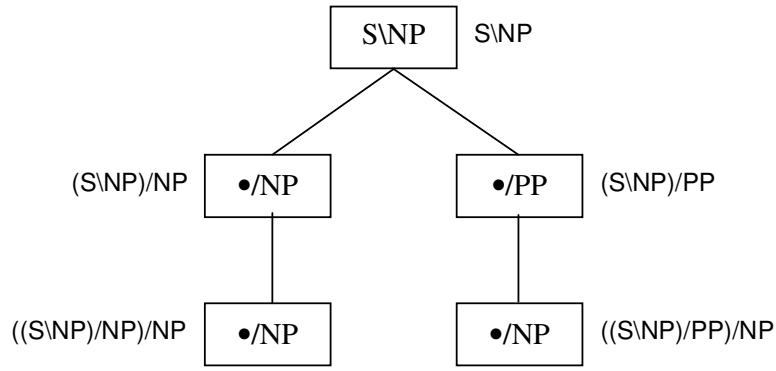


Fig. 8.3. Inheritance structure of intransitive, transitive and ditransitive verbal categories. • signifies the inherited category.

turn inherits the intransitive form, which we shall refer to as the Inheritance Model of Syntactic Categories. It is therefore possible to represent left-branching categories such as these, not as discrete categories, but as a tree, as shown in Fig. 8.3.

Although outside the remit of this work, we postulate that the inheritance of syntactic categories is relevant to the learning of syntax by reasoning that without inheritance, there is no regularity (other than by chance) between intransitive, transitive and ditransitive forms, leading to the situation where each is a random selection of a point in the problem space, thereby giving rise to sentences like the following within one language:

John ate.

Kippers ate John.

Kippers John for breakfast ate.

Extracting syntactic information from a linguistic environment populated by sentences such as these would seem far more difficult than from an environment in which a regular structure is the norm; a pattern can be extracted only if a pattern is there to extract.

8.3.6 Evidence from CCGBank for configuration as syntactic inheritance

If inheritance is an aspect of configuration, then it should be evident in a configured system such as the syntactic categories of English. CCGBank again provides the raw data from which evidence of configuration may be sought: Working with verbal data identified by the VB, VBD, VBN, VBP and VBZ parts of speech, all category occurrences having a frequency < 10 were eliminated, resulting in the 31 unique verbal syntactic categories shown, with their frequency of occurrence, in Table 8.3.

Category	Freq	Category	Freq
(S\NP)/(S\NP)	27285	((S\NP)/(S\NP))/PP	81
(S\NP)/NP	22798	(S\(S\NP))/NP	59
S\NP	8924	S/NP	56
(S\NP)/S	7225	(S/S)/NP	49
(S\NP)/PP	3817	(S/(S\NP))/NP	44
(S\S)\NP	1610	((S\NP)/NP)/(S\NP)	43
(S\S)/NP	1384	(S\PP)/NP	41
((S\NP)/PP)/NP	1223	NP	37
((S\NP)/(S\NP))/NP	1179	((S\NP)/PP)/(S\NP)	34
NP/NP	780	((S\NP)/(NP\NP))/NP	31
((S\NP)/NP)/NP	364	(NP/NP)\NP	29
((S\NP)\(S\NP))/PP	190	(((S\NP)/PP)/PP)/NP	25
((S\NP)/(S\NP))/(S\NP)	134	((S\S)\NP)/NP	18
((S\NP)/S)/(S\NP)	113	((S\NP)\(S\NP))/NP	13
((S\NP)/S)/NP	101	(((S\NP)/S)/(S\NP))/NP	11
(NP\NP)/PP	82		

Table 8.3. Verbal syntactic categories extracted from CCGBank parts 2-21.

The tabled categories can be represented by two trees, one rooted in S, the other in NP. For clarity, the tree rooted in S is split into two subtrees, one for the initial complex category (S|NP) and the other for the remaining categories (Figures 8.3b and 8.3c).

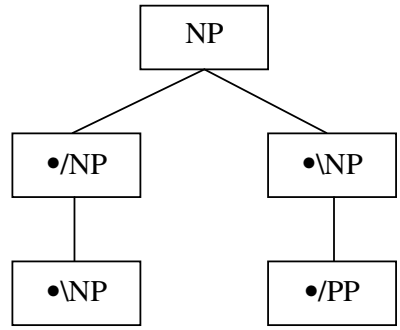


Fig. 8.3a. Verbal syntactic categories rooted in NP.

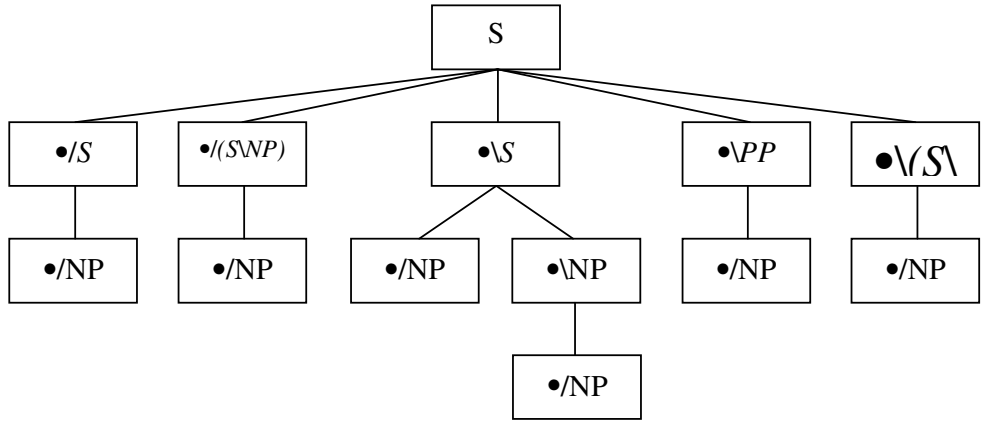


Fig. 8.3b. Verbal categories rooted in S but not (S|NP)

Figures 8.3a to 8.3c show that the verbal syntactic categories extracted from CCGBank do indeed form tree structures, each child category inheriting its left-hand category from its parent. We therefore propose that, given the evidence that only 31 categories in a total space of 57909 categories (from Table 8.2) are present in the configured verbal syntactic categories as defined by CCGBank, and that those 31 categories are related in structure through inheritance of the parent category, a configured verbal category space

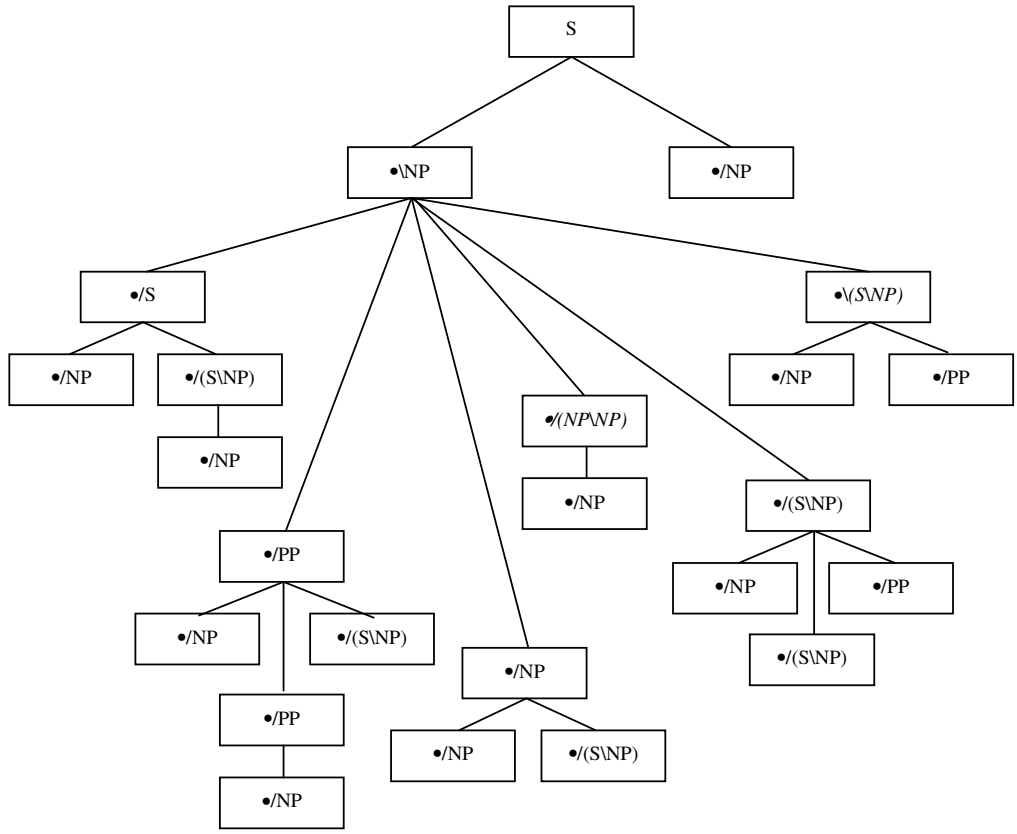


Fig. 8.3c. Verbal categories rooted in S

is a subset of that space in which the points are related structurally and from which the discrete syntactic categories as traditionally used by CG are derived.

Representing syntactic categories as trees has the effect of exposing the inner structure of categories, making it available to the parser, and having implications for incremental interpretation.

8.4 Incremental interpretation using a tree-representation of a configured syntax

In Section 8.2 it was shown that to parse the sentence 'John ate kippers' incrementally, it was necessary to introduce type-raising (Figs. 8.1 and 8.2). In that example, only the

correct verbal category (S\NP)/NP was shown for brevity, however, it is understood that all verbal categories (such as those of Table 8.3) would be identified by lexicon lookup of the verb's surface form, and would be shifted into the chart with the verb.

For simplicity of argument, the verbal syntactic categories to be used here will consist of the basic intransitive, transitive and ditransitive categories described by the tree in Figure 8.3. The first derivation of the sentence, involving the words 'John' and 'ate' are presented in Figure 8.4, and shows that there is no difference in the result; only NP and S\NP can combine through backward functional application to yield the category S. The

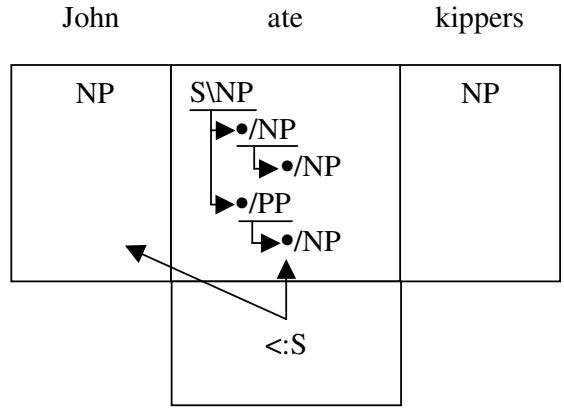


Fig. 8.4. Lexicalised shifts into the parser, verbal categories in tree representation.

difference, however, is in what this derivation means. As the (S\NP) root of the verb syntactic category tree has successfully combined with NP, the implication is that all descendants of the root have the potential to combine with NP, once all right-hand categories have been stripped-away through future combinations; involving the now exposed innermost (left embedded) category of the distinct S\NP, (S\NP)/NP, ((S\NP)/NP)/NP and ((S\NP)/PP)/NP categories has *licensed* those categories for use in future combinations.

The rule-to-rule hypothesis [GAZD85] maintains that each syntactic rule is associated with a semantic rule. Thus the licensing of syntactic categories will be reflected in the semantic categories; if the syntax can be licensed then so can the semantics. The intuition behind semantic licensing is that the *sense* of an intransitive verb is the same as that of its transitive and ditransitive forms; drinking is drinking, regardless of what is being drunk and why. If the intransitive sense of a verb can be found, or at least the possible senses reduced, then the sense (or senses) will be applicable in the transitive and ditransitive cases. The relevance of licensed syntactic categories will become apparent when they are used in conjunction with the parallel-shifting Chunker of Chapter 5 and the Specialisation Classes of Chapter 6, that is, when all elements are integrated into a pre-processor to the Construction Integration Model. A worked example is given in Chapter 9.

8.5 Indicating Sense in Semantic Categories

So far, only the syntactic elements of the grammar have been discussed. However, it is necessary to incorporate a notion of *sense* to the semantic categories that partner the syntactic ones. This is easily accomplished by assigning a list of appropriate senses to each *instantiated* variable in a semantic category as follows:

A typical semantic category is initially a template of the form λy.λx.*(y, x), where * takes the place of the predicate, and x and y are the argument variables. The predicate is assigned its value by the lexicon as each input term is looked-up, so given the word 'drink', the lexicon returns the semantic category λy.λx.drink'(y, x). However, the

lexicon also assigns possible senses to the predicate, requiring a list containing those senses to be associated with that predicate. We propose the following notation:

$\lambda y.\lambda x.drink'_{\{\#1, \#2, \ldots, \#n\}}(y, x)$ where #1 etc are sense indicators.

In the work to follow the list of senses will comprise Specialisation Classes, and as the list may be lengthy, it will be abbreviated to {SC}, hence:

$\lambda y.\lambda x.drink'_{\{SC\}}(y, x)$

8.6 A criticism of the Inheritance Model

The Inheritance Model presented here is applicable only to a special class of syntactic categories, those that are left branching and rooted in a compound with a single backward slash, that is, from the series:

a\b, (a\b)/c, ((a\b)/c)/d, (((a\b)/c)/d)/e, …

In each element of the series, a\b is the leftmost embedded category, and all slashes, other than the one in a\b, are forward slashes unless they are embedded in their own complex category such as:

((S\NP)/(S\NP))/NP

The category above still has the form (a\b)/c, the bracketing ensuring the overall structure is correct.

Examining the categories given in Table 8.3, it can be seen that of the 31 presented, 20 are of this form (64.5%), including 8 of the 10 most frequently used CCGBank verbal categories, suggesting that the inheritance model will be applicable more times than it is not, however, the relevance of inheritance to categories from outside the given series

will not be discussed here as this thesis intends to present a general framework for future work, not a fully detailed and implemented system.

8.7 Conclusions

Recognising that incremental interpretation is hampered by the structure of syntactic categories, this chapter has proposed that semantic categories occupy points in a problem space of all possible categories. A method of calculating the size of that problem space for all categories comprising up to a given number of atomic categories is presented and used to compare this to the number of actual verbal categories used in English (as extracted from CCGBank). It was found that very few of the available categories were actually used. Furthermore, in examining the structure of the extracted categories it was found that they are closely related through inheritance. Applying this to the configurational theories of the Inside-Out theorists, it was proposed that configuration of an innate grammar system consists of selection of related categories expressed in the problem space. This proposal was supported firstly by showing that very few syntactic categories are present in a configured grammar (that is, the verbal categories extracted from CCGBank), and secondly that the syntactic categories of a configured grammar are related structurally, each category inheriting its parent in its entirety. It was also shown that this 'Inheritance Model of Syntactic Categories' allows left-embedded rightward-looking categories to be accessed early and used in combinations that promote incremental parsing, thereby addressing the inhibitory nature of the standard syntactic category forms to incremental processing. However, there is some question as to whether this technique is generally applicable; not all categories are of the correct form, although they are in the minority. Standard CG semantic categories have no mechanism whereby sense may be expressed, but as has been shown in Chapter

4, sense is instrumental in the determination of correct grammatical structure. To address this, a proposal for an extended CG semantic category structure able to accommodate sense indicators was presented.

9 Combining the Elements

Chapters 5, 6 and 8 present the individual (but interoperating) elements of the proposed CIM pre-processor. This chapter unites those elements and presents a walk-through of the processes as they parse a sentence. Like the CIM, the pre-processor is inspired by psychological evidence, and it is proposed that a system such as this should therefore exhibit similar properties to a real psychological/cognitive language processor. The Garden Path effect is selected as a suitable property and a parallel pair of garden path/non-garden path sentences is used to show the unified elements of the pre-processor perform in accordance with expectation. The results also show that incremental interpretations are built without interfering with the CG derivation process or invoking the criticisms levelled at other approaches to incremental interpretation, and that a correctly grammatically structured derivation results, even when parallel shifts are involved.

The derivation is also shown to be sense tagged with Specialisation Classes that give an overall impression of the sense of the sentence. The process of selecting senses from the initial set of Specialisation Classes assigned to each term involves plausibility testing, that is, testing the senses for *coherence* with a world model, simulated here by Selectional Association values. Here then is the evidence that the Construction Integration Model is involved in the translation of text into a logical representation, coherence determination being a component of the CIM (tacitly referred to in point 3 of the 'Construction' phase of the model – Section 3.3).

A pre-processor to the Construction Integration Model that is built using elements inspired by theories and evidence concerning the cognitive functions of the human brain might reasonably be expected to exhibit similar properties to its human counterpart. To study this aspect of the pre-processor elements, a Garden Path sentence shall be processed; the ability of such sentences to induce parse errors in human readers, as discussed in Chapter 5, is an effect that should be reproducible by an analogue of the human system.

The sentence pairs (5) and (6) of Chapter 5, reproduced below as (1) and (2), shall be used to evaluate the performance of the pre-processor elements.

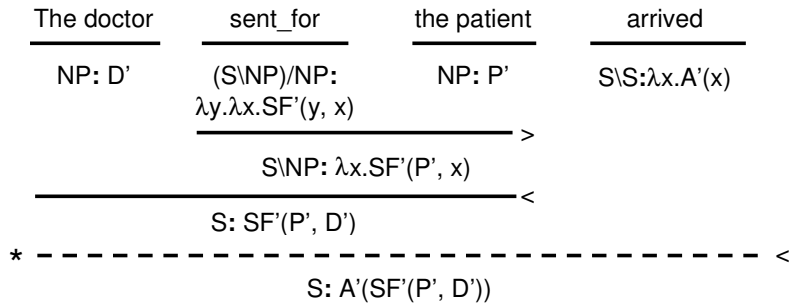
The doctor sent for the patient arrived. (1)

The flowers sent for the patient arrived. (2)

9.1 Standard CG parse of the Garden Path Sentences

Before proceeding with the evaluation, the standard CG parse of sentences (1) and (2) are presented below as (3) and (5):

The doctor sent for the patient arrived.



(3)

The doctor	sent	for	the patient	arrived
NP: D’	(S\NP)/PP: λz.λy.S’(z,y,(ana’ x))	PP/NP:λx.x	NP: P’	S\S:λx.A’(x)
		PP: P’ (>)		
	S\NP: .λy.S’(P’,y,(ana’ x)) (>)			
	S\NP: .λp.S’(P’,A’(p),(ana’ x)) (<B)			
S: .S’(P’,A’(D’),(ana’ x)) (<)				

(4)

The flowers sent for the patient arrived.

The flowers	sent	for	the patient	arrived
NP: F’	(S\NP)/PP: λz.λy.S’(z,y,(ana’ x))	PP/NP:λx.x	NP: P’	S\S:λx.A’(x)
		PP_{io}: P’ (>)		
	S\NP: .λy.S’(P’,y,(ana’ x)) (>)			
	S\NP: .λp.S’(P’,A’(p),(ana’ x)) (<B)			
S: .S’(P’,A’(F’),(ana’ x)) (<)				

(5)

As discussed in Section 5.2.3, the transitive constituent representing a doctor requesting a patient appears correct. However, at the disambiguating term ‘arrived’ a parse failure occurs and an alternative parse must be sought, resulting in the parse shown as derivation (4).

Sentence (2) is not a Garden Path, the sense of ‘flowers’ as plants being incompatible with the required animate subject sense of the verb ‘send_for’. The sentence therefore parses without failure, as shown by derivation (5), which, apart from the word ‘flowers’ is identical to (4).

9.2 Parsing using the pre-processor

For comparative and explanatory purposes, the proposed elements of a pre-processor to the CIM shall now be used in concert to parse the Garden Path sentence 'the doctor sent for the patient arrived'.

9.2.1 The action of the Chunker

The terms of the sentence are streamed into the Chunker which identifies the potential compound terms within the sentence, and outputs all possible grouping permutations of adjacent terms as items to be shifted into the parser. With reference to Figures 9.1 and 9.2, the Chunker firstly identifies the noun phrase 'the doctor'. Following this, two parallel elements comprising the two individual elements 'sent' and 'for', and the compound verb 'sent_for' are identified. Next, another noun phrase 'the patient' is found, followed by the single term 'arrived'. During this process, extents based on the individual terms' indices (Fig. 9.1) are assigned to the Chunker-identified terms (Fig. 9.2).

Firstly, the space-delimited terms are indexed:

Sentence: The doctor sent for the patient arrived.
Term Index : 1 2 3 4 5 6 7

Fig. 9.1. Sentence with term indices.

The Chunker then identifies the chunks to be shifted, adding extents:

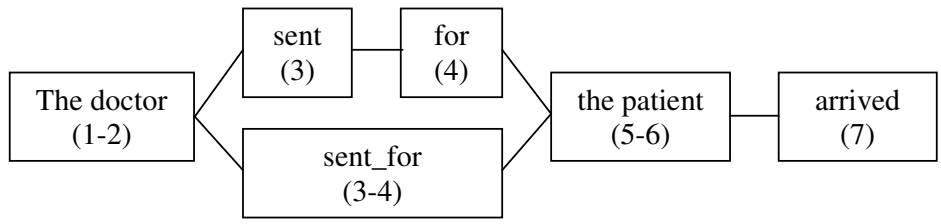


Fig. 9.2. Identified chunks with extents.

9.2.2 Specialisation Class assignment

The possible senses of identified chunk heads are identified via the lexicon as follows:

Sense #	Doctor	Sent	Sent_for	Patient	Arrived
1	Physician	Cause to go	Request	Person requiring medical care	Reach destination
2	Theologian	Transmit			Succeed
3	Role in game	Mail			
4	A Ph.D	Transport			
5		Posting			
6		Transfer			
7		Commit to institution			
8		Broadcast			

Table 9.1. WordNet senses of sentence words.

Specialisation Classes are assigned on the basis of the possible senses according to the procedure presented in Chapter 6:

Sense #	Doctor	Sent	Sent_for	Patient	Arrived
1	Person	Move	Order	Person	Get
2	Person	Move			Succeed
3	Diversion	Move			
4	Person	Move			
5		Move			
6		Move			
7		Transfer			
8		Tell			

Table 9.2. Specialisation Classes assigned to sentence words.

9.2.3 Category Assignment

Along with Senses and Specialisation Classes, the lexicon also supplies the possible syntactic and semantic categories for each word. For clarity of argument, a restricted set of categories will be used here:

Noun: NP**:** *

Preposition: PP/NP**:** λx.x

Verb: S\NP: λx.*(x)
(S\NP)/NP: λy.λx.*(y, x)
(S\NP)/NP: λz.λy.*(z,y,(ana' x))
(S\NP)/PP: λy.λx.*(y, x)
(S\NP)/PP: λz.λy.*(z,y,(ana' x))
S\S: λx.*(x)

With reference to Chapter 8, the verbal categories are expressed by the following tree:

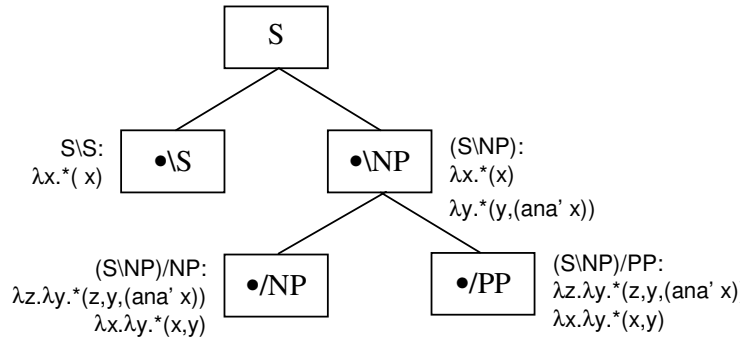


Fig. 9.3. Verbal category tree.

9.2.4 Shifting into the chart

The sentence has now been prepared for shifting into the parser; chunks have been identified and assigned extents, categories and potential senses. The first chunk to be shifted is 'The doctor', the chunk comprising the categories, extent and list of SCs:

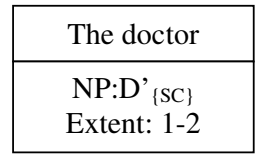


Fig. 9.4. Initial shift into the chart contains the syntactic and semantic categories of the chunk words, plus the word's Specialisation Class list.

Next, a parallel shift is necessary to accommodate the two paths corresponding to 'sent for' and 'sent_for'. Both parallel elements also include the categories, Specialisation Class list and extent.

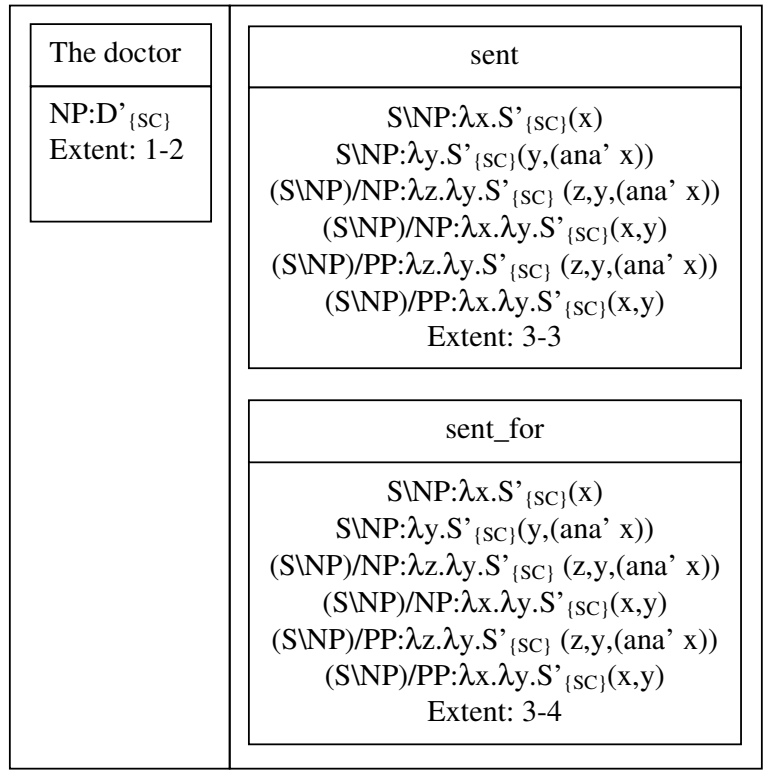


Fig. 9.5. Chart state after one shift and one parallel shift.

9.3 The initial combination

With the first two cells of the chart populated, the initial combination can be made. As column 2 has two dimensions, one for 'sent' and the other for 'sent_for', two combinations are necessary, although as the same categories have been applied to both verb forms, the categorial combinations will be the same.

The only derivations possible using the verbal categories listed above involve backward functional application between the noun and verb:

$$\frac{\text{NP:D'}_{\{SC\}} \qquad \text{S\NP: } \lambda x.\text{V'}_{\{SC\}}(x)}{\text{S: V'}_{\{SC\}}\ (\text{D'}_{\{SC\}})}\ < \qquad \frac{\text{NP:D'}_{\{SC\}} \qquad \text{S\NP: } \lambda y.\text{V'}_{\{SC\}}(y,(\text{ana' } x))}{\text{S: V'}_{\{SC\}}\ (\text{D'}_{\{SC\}},(\text{ana' } x))}\ <$$

Fig. 9.6. The first derivations. Verbal predicate V' refers to either 'send' or 'send_for'.

Although as has been seen in derivations (3), (4) and (5) of Section 9.1, this derivation plays no part in the final parse of the sentence; it does however license those categories in which (S\NP) is left-embedded. The effect of licensing, that of promoting word sense disambiguation, is demonstrated below.

9.3.1 Licensing promotes sense-disambiguation.

The combinations above produced the semantic interpretations $\mathsf{sent'}_{\{SC\}}(\mathsf{doctor'}_{\{SC\}})$ and $\mathsf{sent\_for'}_{\{SC\}}(\mathsf{doctor'}_{\{SC\}})$ for the cases where 'doctor' is the subject, and $\mathsf{sent'}_{\{SC\}}(\mathsf{doctor'}_{\{SC\}},(\mathsf{ana'}\ x))$ and $\mathsf{sent\_for'}_{\{SC\}}(\mathsf{doctor'}_{\{SC\}},(\mathsf{ana'}\ x))$ for the cases where the subject is unspecified and 'doctor' is the direct object. Each of these propositions is available for plausibility testing by comparing their possible senses (specified by the list $_{\{SC\}}$) with real-world knowledge.

Plausibility testing, or *coherence determination,* at this point is important in terms of this thesis as, in a full discourse comprehension system, it is the Construction Integration Model that 'performs' the tests by matching input propositions against those in working memory and/or long-term memory either directly or through inference [KINT78]. As it is reasonable to expect that there is only one plausibility checking mechanism for linguistic information, it follows that this mechanism is used by the Construction Integration Model *and* the proposed pre-processor, thereby providing evidence that elements of the CIM are used in the pre-processor.

World-scale knowledge bases are currently not available for use in this work, however it is possible to simulate a suitable knowledge base for the purposes of this thesis. In Chapter 7, Selectional Association as a method of Word Sense Disambiguation was

used to assess the performance of Specialisation Classes. The Selectional Association data will be used here to determine the semantic fit of predicates and their arguments in the propositions created above, that is, to simulate the plausibility testing performed by the CIM (point 3 of the 'Construction' phase - Section 3.3)

Taking sent'$_{\{SC\}}$(doctor') first, the Selectional Association values for verbs and their objects were calculated for all permutations of the associated predicate and argument Specialisation Classes, the values shown in Table 9.3. The table shows the strongest association is between 'move' and 'person', which includes all senses of 'doctor' except 'diversion', and all senses of 'send' except 'commit to an institution' and 'broadcast' (i.e. the senses having nothing to do with moving).

SC	move	transfer	tell
adult			
person	**0.060**		0.0218
religionist			
diversion			
intellectual			

Table 9.3. Selectional Association values for all SV Specialisation Classes of 'doctor' and 'sent'. Empty cells indicate no association.

Similarly, the Selectional Associations values for sent_for'$_{\{SC\}}$(doctor'$_{\{SC\}}$) were calculated:

SC	order
adult	
person	**0.068**
religionist	
diversion	
intellectual	

Table 9.4. Selectional Association values for all SV Specialisation Classes of 'doctor' and 'sent_for'.

Here, the strongest (and only) association is between 'person' and 'order', and as the association value is higher than that obtained in Table 9.3 for the 'sent' form of the verb, 'send_for' is currently the preferred interpretation, that is, $\text{sent\_for}'_{\{SC\}}(\text{doctor}')$ is the most plausible proposition.

The Selectional Association values where the noun 'the doctor' is taken as the direct-object of either verb form were similarly calculated, the result shown in Tables 9.5 and 9.6.

SC	move	transfer	tell
adult	0.000		
person	**0.010**		0.005
religionist			
diversion			
intellectual			

Table 9.5. Selectional Association values for all VO Specialisation Classes of 'doctor' and 'sent'.

SC	order
adult	
person	**0.041**
religionist	
diversion	
intellectual	

Table 9.6. Selectional Association values for all VO Specialisation Classes of 'doctor' and 'sent_for'.

Of these results, the association between 'order' and 'person', that is, send_for' and 'doctor', is the highest. However, it does not beat the value obtained for 'doctor' as the subject of the verb 'sent_for' (Table 9.4), and so the proposition $\text{sent\_for}'_{\{SC\}}(\text{doctor}')$, derived from the categories $S\backslash NP: \lambda x.V'_{\{SC\}}(x)$ is selected as the most plausible. This has two effects: firstly, the categories where 'doctor' is the direct-object of the verb (i.e.

S\NP:λy.V'$_{\{SC\}}$(y,(ana' x))) can be deactivated, removing them from further consideration, and secondly, the list of Specialisation Classes associated with the selected categories can be modified such that the verbal SC list contains only 'order', and the nominal SC list just 'person'. The licensing of categories, made possible through exposure of the left-embedded (S\NP), permits this information to be

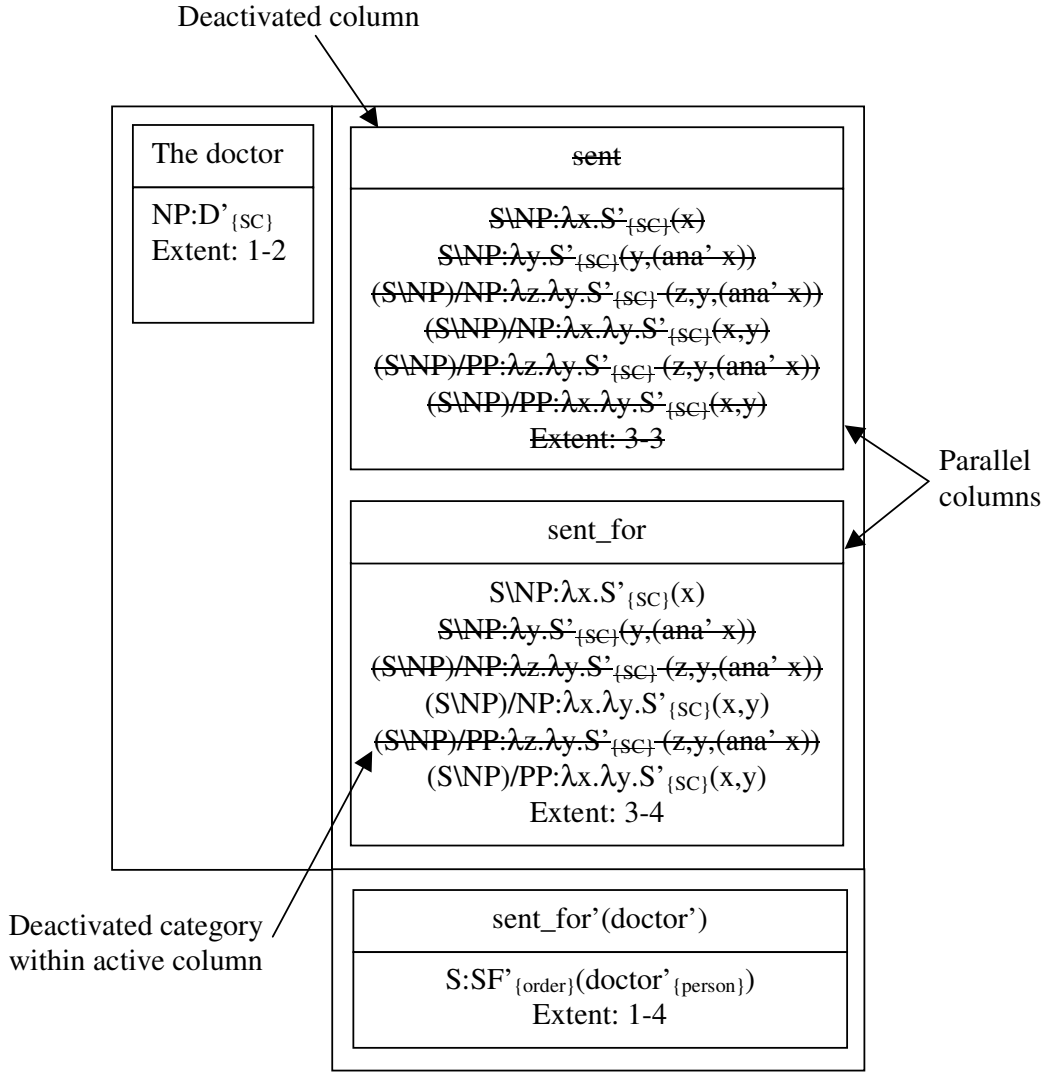


Fig. 9.7. Chart state after first combination. Senses have been identified and an entire parallel path has been deactivated.

propagated through future combinations. The state of the chart after this step is shown in Fig. 9.7, and shows that an interpretation, $SF'_{\{order\}}(doctor'_{\{person\}})$, has been constructed.

9.4 The second combination

The second combination requires 'the patient' to be shifted into the chart. Figure 9.8 depicts the chart state after the shift. For clarity, the deactivated components are not shown.

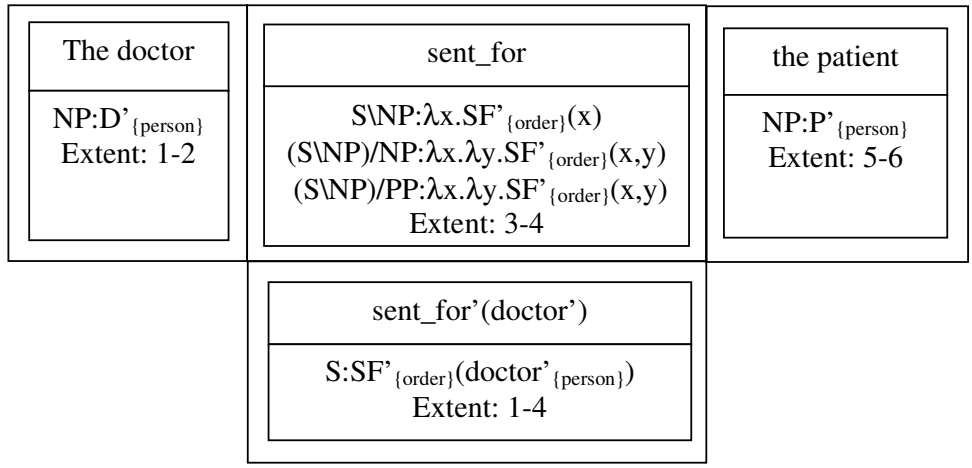


Fig. 9.8. Chart state after third shift, showing only active columns.

The only possible combination is between the categories $(S\backslash NP)/NP:\lambda x.\lambda y.SF'_{\{order\}}(x,y)$ and $NP:P'_{\{person\}}$, resulting in $S\backslash NP:\lambda y.SF'_{\{order\}}(patient_{\{person\}},y)$. Table 9.6 shows that 'person' is a suitable direct object argument class for the verb 'send_for', and so this derivation is plausible. The chart state shown in Figure 9.9 shows the result after completing all derivations following this shift.

Because of the inheritance structure of the syntactic categories, it is possible to build the semantic interpretations, shown in emboldened boxes, in an incremental fashion: the construction of sent_for'$_{\{order\}}$(patient'$_{\{person\}}$,doctor'$_{\{person\}}$) is a direct descendant of sent_for'$_{\{order\}}$(doctor'$_{\{person\}}$). The derivation however follows the standard right-branching approach of combining the direct-object with the verb, combining that derivation with the subject.

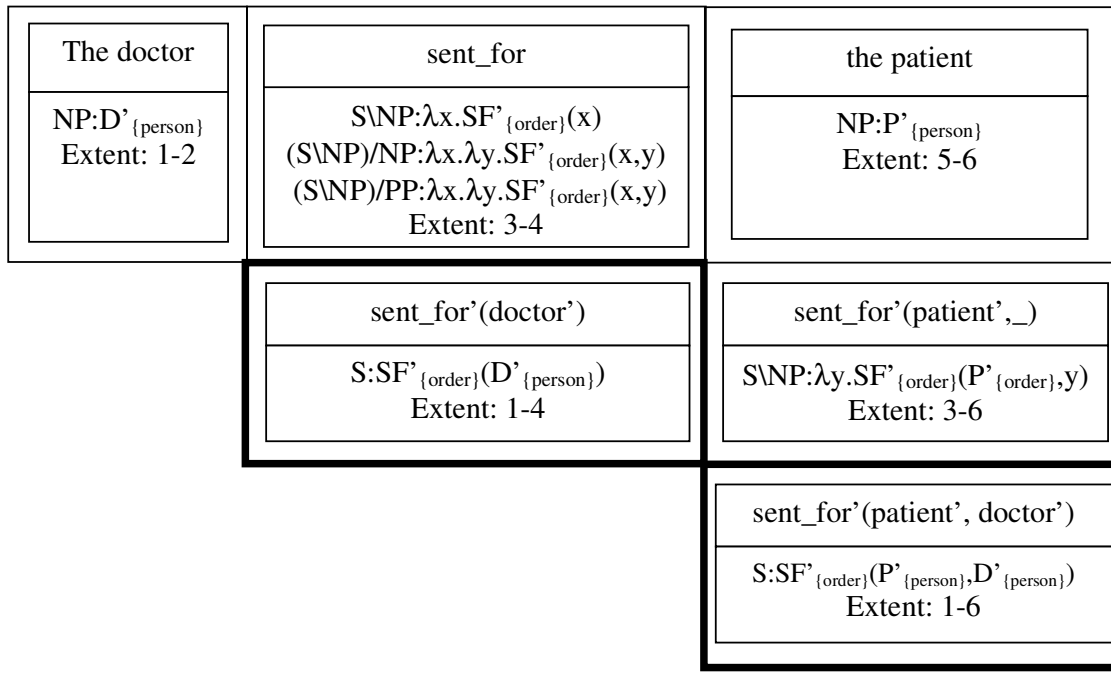


Fig. 9.9. Chart state after third shift, showing all derivations, including complete interpretations in emboldened chart cells.

9.4.1 The parse failure

On attempting to combine 'arrived' with the derivation of 'the doctor sent_for the patient', a parse failure results; although the syntactic categories can combine, as shown in derivation (3), the semantic categories fail as it makes no sense to say that 'the act of sending_for something' has arrived. As we have not yet extracted Selectional

Association data for verbs taking verbs as arguments, we cannot demonstrate the failure here and must fall back on our intuitions.

Having recognised a parse failure, we reactivate the chart elements previously deactivated after selection of the most plausible parallel path, as described in Chapter 5. The next most plausible interpretation on the basis of Selectional Association is presented in Table 9.3, corresponding to the 'send' version of the verb with 'doctor' once again as the subject. However, this parse will fail semantically as well – the failure is caused by the inappropriate attachment of the verb 'arrived' to the sentence 'the doctor sent for the patient'. The same is true of the second repair attempt, which selects 'doctor' as the direct object of the verb 'sent_for' on the basis of its Selectional Association Value being the next plausible, having a SA value of 0.041 (Table 9.6).

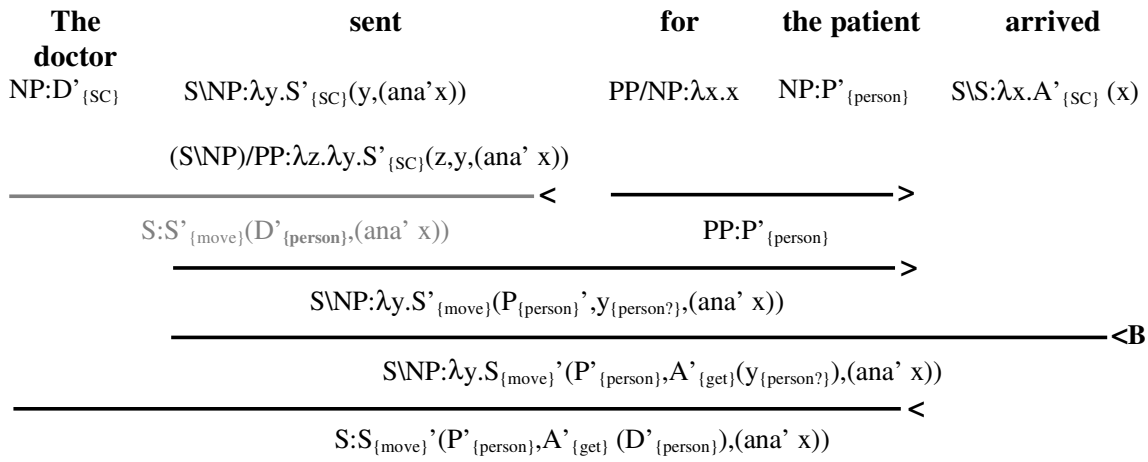


Fig. 9.10. Successful chart parse of 'the doctor sent for the patient arrived'. Note that the initial derivation from 'The doctor' and 'sent' is not involved in any additional derivations, but can be used to predict that the sense of lambda-variable **y** of 'send' will be PERSON. This is indicated by the sense-tag,{PERSON?}.

The third repair attempt is successful; having the next highest Selectional Association value of 0.010 (Table 9.5), 'doctor' is selected as the direct object of the verb 'send'. The full chart parse is shown in Figure 9.10 and shows all participating categories and derivations. 'Arrive' has been assigned the sense 'get' (i.e. arrive at destination) on the basis of the Selectional Association between itself and 'doctor$_{\{person\}}$' as its subject. Although the sentence is not fully sense disambiguated, it does carry sense indicators in the form of Specialisation Classes, which have been instrumental in defining that grammatical structure of the sentence by enabling plausibility testing at the predicate-argument level, and which now give a general sense of the sentence,

$$S_{\{move\}}'(P'_{\{person\}},A'_{\{get\}}\ (D'_{\{person\}}),(ana'\ x))$$

which may be interpreted as:

> An unknown agent (ana' x) moved (sent) a person (doctor) to another person (patient) and that person (doctor) has got there (arrived).

9.4.2 Parsing a non-garden path sentence

Processing the non-garden path sentence 'the flowers sent for the patient arrived' proceeds in similar fashion to the garden path sentence. After the chunks 'the flowers', 'sent' and sent_for' have been shifted into the parser, the plausibility tests are made, this time resulting in 'sent' being selected as the most plausible verb, with 'the flowers' as its object (Table 9.7). No SA values are available for the verb 'send_for' as plants cannot order or be ordered. Clearly, the parser has this time selected the correct path on the first attempt as expected, and the parse proceeds without the need for repair.

Verb	Flowers as subject	Flowers as object
Send (SC=move)	0.002 (SC=plant)	**0.013** (SC=plant)
Send for (SC=order)		

Table 9.7. Maximum SA values between SCs of the verbs 'send'/'send_for' and the noun 'flowers'.

The chart state after the first combination of the non-garden path sentence is shown in Figure 9.11.

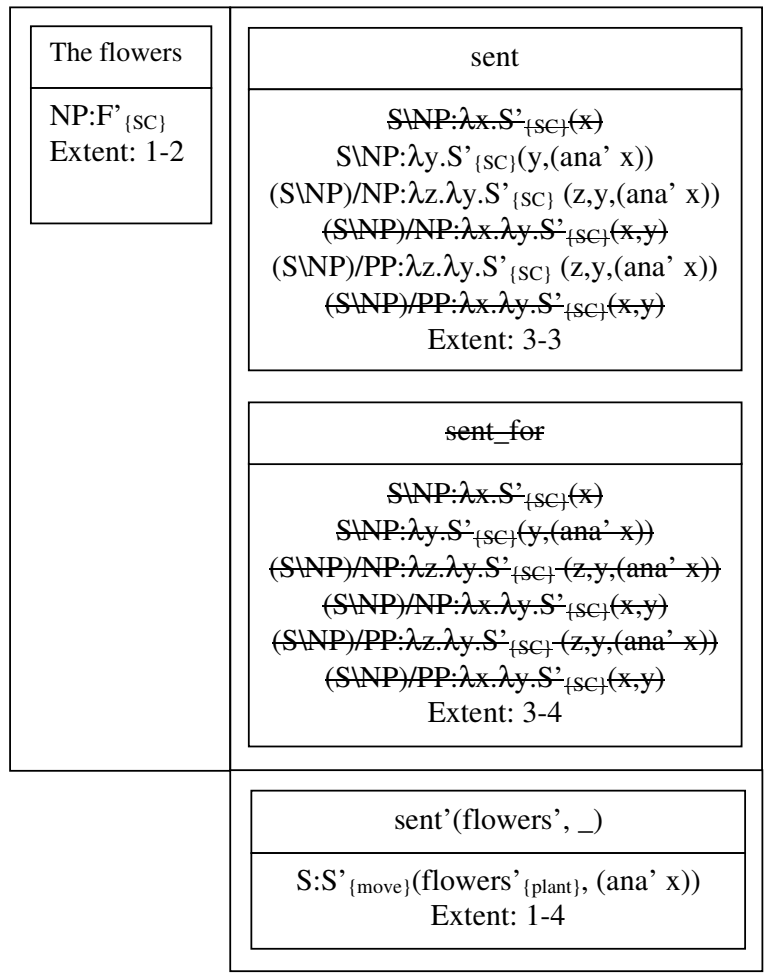


Fig. 9.11. Chart state after first combination. Senses have been identified and an entire parallel path has been deactivated.

9.5 Conclusions

It was proposed that a pre-processor to the Construction Integration Model, if constructed in accordance with functional components that experimental data would suggest are in operation in the human equivalent, would exhibit similar properties to the human version. The property examined here was the parse error induced by garden path sentences.

Using Selectional Association data as a means of determining plausibility, and the proposal that left-embedded syntactic categories can be represented as a tree thereby exposing their innermost sub-category, a typical garden path sentence was processed.

The results obtained mirrored the actions of the human processor; parse errors resulting in repair attempts at the point of ambiguity.

The results show that by exposing left-embedded categories, plausibility decisions can be made earlier than would be expected from examination of a standard complete parse of the sentence in which derivations to the right of the verb are obtained before incorporating categories to the left. Although these early derivations are not used in the remainder of the parse, the plausibility information they afford propagates throughout all related categories, that is, those sharing the left-embedded category. Thus given a verb with intransitive, transitive and ditransitive forms, the sense of the intransitive form, made available here through left-embedded category exposure, will also be that of the transitive and ditransitive forms, and so is already available when those verb forms are considered by the parser. We propose that the construction of intransitive partial interpretations (e.g. $\text{John}_{\{\text{person}\}}$ $\text{ate}_{\{\text{consume}\}}$), which lead to transitive partial interpretations (e.g. $\text{John}_{\{\text{person}\}}$ $\text{ate}_{\{\text{consume}\}}$ $\text{kippers}_{\{\text{food}\}}$), which in turn lead to ditransitive interpretations (e.g. $\text{John}_{\{\text{person}\}}$ $\text{ate}_{\{\text{consume}\}}$ $\text{kippers}_{\{\text{food}\}}$ for $\text{breakfast}_{\{\text{meal}\}}$) is

an expression of incremental interpretation because of the elimination of superfluous senses and categories at each level of transitivity.

Stabler [STAB91] has previously addressed the incremental interpretation problem, drawing an analogy between incremental interpretation and preparing a meal; the individual processes involved in meal preparation may proceed asynchronously as long as they all complete at the same time. Stabler's view is that syntactic and semantic processes run incrementally but in parallel. Ultimately, the results of the processes are combined to provide the final parse of the sentence. However, Steedman [STEE00] argues against the theory of asynchronous processes, showing that Stabler does not attempt to handle any non-constituents of the type 'John loves', which would occur when attempting to process a sentence incrementally.

This criticism cannot be levelled against the system described here; the formation and use of non-constituents is instrumental in the early elimination of less-plausible senses and categories. The parsing method described here is able to build incremental interpretations by integrating notions of sense and plausibility with the syntactic and semantic categories of the standard CG parsing mechanism, and by allowing the parser to select between chunks rather than having it build them.

We conclude then that the actions of the described pre-processor not only avoid criticisms of asynchronous processes, but are also consistent with expectations with

regard to Garden Path sentences, and hence its operations in this context are consistent with those of the human processing element.

Regarding the main thesis, that the translation of text into logical form is not handled by a separate process but involves elements of the Construction Integration Model, this work presents evidence that the thesis is correct. It has been demonstrated that the element of the CIM involved in the translation is that of coherence determination, and that it assists the translation process by testing the plausibility of sense-tagged propositions generated by the grammar-parsing element. In order to accomplish this effectively it has been necessary to introduce the Inheritance Model of Syntactic Categories, which permits interpretations to be built incrementally without affecting the overall CG parsing strategy, and to extend CG semantic categories to carry sense indicators.

10 Conclusions

The main contributions of this thesis are presented here, along with suggested future work.

The Construction Integration Model is not a complete model of discourse comprehension because it does not actually process text (or natural language); instead it accepts a *logical representation* of text as its input.

The main thesis is that the translation of text into logical form is not handled by a separate process, but actively involves elements of the Construction Integration Model (CIM) itself. The thesis demonstrates this to be the case by drawing on current linguistic resources and techniques to build a system (henceforth called the pre-processor) capable of performing the translation, thereby revealing the elements of the CIM at work. The two philosophies underlying this work are that:

1. A psychologically oriented model of discourse comprehension would be served best by a psychologically oriented pre-processor;
2. A web page summariser should work in real-time, and so the linguistic/summarisation processes cannot be overly elaborate or resource-hungry.

This thesis has identified and explored three main elements of such a pre-processor in terms of these philosophies, justifying the inclusion of any technique that addresses these elements in terms of their psychological/cognitive validity, and processing effort

and quantity of 'knowledge' required to implement, presenting improvements or novel alternatives as necessary.

10.1 Conclusions relating to the field of Linguistics

The work presented here does not necessarily relate solely to the study of linguistic systems that involve the CIM; we believe it holds general implications for the field of Linguistics, and specifically for Parsing and for Word Sense Disambiguation (WSD).

Firstly, this thesis demonstrates that the derivation of logical form is not independent from the processes of syntactic parsing and WSD. The implication here is that any one of the processes **must** take the other two into consideration if ambiguity is to be overcome. For instance, Section 4.1.1 demonstrates that the *sense* of sentential terms is a factor in determining the correct site of prepositional attachment – if sense is not taken into account, a syntactic parser can only suggest an ambiguous set of possible attachments.

Similarly, Section 2.5.5 demonstrates that when attempting to discover the sense of terms through coherence (or any other WSD technique), false positives can be avoided if the terms in question are not randomly selected but chosen on the basis of a possible relation, that is, by selecting those terms that, in some logical derivation, hold (for example) a predicate-argument relation.

We therefore propose that the processes of logical form derivation, syntactic parsing and WSD be viewed as interrelated and mutually constraining processes.

Secondly, it has been demonstrated in Section 5.2.3 that the mechanism of shifting space-delimited terms into a chart parser is incomplete once *sense* is added to the 'standard' set of *syntactic* and *semantic* categories these parsers have been designed to process. This is because compositionality of syntactic (or semantic) categories and of term senses are different problems; syntactic categories such as N and N can be composed to give N, whereas the senses of individual terms of an idiom (for example) cannot be composed to give the sense of that idiom. If, as is accepted here, it is the lexicon that supplies not only the syntactic and semantic categories, but also possible senses to any recognised term, then the lexicon must also recognise compounds and idioms (Section 5.2.2) if the non-compositionality problem above is to be avoided. Simply put, the chart parser must support both compound and individual term paths in order to present all reasonable senses to the sense-selection process. The use of a parallel-shift enabled chart parser of the type proposed here is therefore a necessary step in integrating sense with syntactic and semantic categories into the parsing process.

Thirdly, in order for the lexicon to supply possible term sense indicators, and for those senses to be carried by the CG semantic categories, the semantic categories must be extended to accept those indicators. This is accomplished here by attaching a list of possible senses to the lambda-variables of semantic categories in much the same way as feature bundles are attached to syntactic categories for control of number, person and gender agreement (although these agreement feature bundles have not played a part in this thesis). By carrying possible senses within the semantic categories it becomes possible to evaluate and rate the alternative categorial derivations for plausibility. By selecting those senses within a logical representation that conform most closely to the

'world model', both Word Sense Disambiguation and Structural Disambiguation are achieved, and single-terms/compound-term ambiguity is resolved.

We believe that the Inheritance of Syntactic Categories is an important property of grammars when one considers both the difficulty of grammar acquisition and the penchant of the (human) brain for pattern detection. If, as discussed in Chapter 4, the process of grammar acquisition can be summarised as the configuration of an innate language facility, achieved by detection of regularities in the input signal, then acquisition is very much more achievable if the signal is constructed so as to make the regularities a feature. This is expressed at the E-Language level by relative word orders, in for example active and passive sentences, and in the inheritance of intransitive verb structures by transitive verbs, and so on. We believe that recognition of inheritance would be advantageous when attempting to induce grammars from, say, corpus analysis, where constituent that express inheritance could be selected over those that do not (although we appreciate that there will always be exceptions). By the same token, inheritance may also be a useful element of error recovery. As an E-Language can potentially consist of an infinite number of sentences, it is likely that any induced or crafted grammar will not provide 100% coverage and will fail to parse some sentences. Should such a gap in the grammar be detected, inheritance provides a principled means of automatically extending the grammar. Plausibility testing can of course moderate the addition to the grammar of categories obtained in this way.

Sense tag-sets could be said to come in two kinds: low granularity (such as the LDOCE) and high granularity (such as WordNet). The low granularity tag-sets comprise

relatively few sense tags (~2000) which in use enable high(er) precision sense-tagging; with fewer tags to choose from there is less chance for error. High granularity tag-sets consist of many sense tags (WordNet has 66025 noun and 12127 verb senses), and these offer the prospect of more informative sense assignments. However, as sense assignment algorithms for use with high-granularity tag-sets are necessarily more complex and difficult to devise, they have a low(er) sense-assignment precision. We have shown that low granularity Specialisation Classes can be mapped with high precision into high-granularity senses when used in conjunction with lemmas to form a key into the original taxonomy from which the Specialisation Classes were derived. Processes employing Specialisation Classes can therefore work at a reasonably high level of abstraction when applying low complexity reasoning methods, for example the Selectional Association between the verb RUN and a subject ANIMAL instead of, say, HORSE or DOG. Sense-tags so obtained can then be mapped onto detailed senses without recourse to more complex reasoning methods. Alternatively, the more detailed senses obtained through mapping can be used with more complex reasoning methods to further refine sense and structural ambiguity if necessary.

Finally, at the time of writing, the 32 files from the SUSANNE corpus that have been sense-tagged with WordNet 1.6 senses drawn from the SemCor corpus are being prepared for inclusion on Geoffrey Sampson's resources website, and will be available for download from **http://www.grsampson.net/Resources.html** by the end of 2005.

10.2 Main Conclusions

Chapter 2 reviews available summarisation techniques, contrasting them in terms of their applicability to the summarisation of web pages, and their knowledge requirements and processing effort required to realise them. The chapter favours the psychologically inspired approaches because, although they require huge linguistic, grammatical and knowledge resources when compared to the surface feature oriented statistical techniques, in mimicking human language processing they offer better prospects for processing the diversity of document types and contents found on the web.

Chapter 3 looks at the Construction Integration Model in detail, presenting the background evidence for its acceptance as a reasonable model upon which to build a discourse comprehension system. The CIM is selected as the model of discourse comprehension as it has psychological validity, addresses both local and global coherence, and utilises summarisation as an integral part of its processing. In presenting the individual processes of the model however it becomes evident that the initial step, that of converting text into a logical representation, is not part of the model as it is described. A contribution of this thesis is therefore the rectification of this situation.

Chapter 4 identifies the main elements of the pre-processor as those dealing with logical form transformation, sense, and coherence. These seemingly distinct elements are demonstrated to be interrelated by presenting example sentences that show that the site of prepositional attachment, and hence grammatical structure, is dependent on sense (Section 4.1.1), and with reference to [KINT78] that coherence may only be sought

between senses, not surface forms (Section 4.1.2). From this it is concluded that these elements cannot be treated as distinct. The grammar parser is posited as the site of intersection of the elements on the basis that it is the parser that actually generates the logical forms, and logical forms are affected by sense (Section 4.1.1) and coherence testing [KINT78]. By uniting the elements at the grammar parser it is further proposed that they become interrelated, mutually constraining processes, and demonstration of these propositions (Chapter 9) are contributions of this thesis.

As a grammar parser has been proposed as the site of element interaction, an experimental or implemented pre-processor requires a grammar parser that is compatible with those elements. A review of grammar acquisition theories results in the selection of the Coalition Model, which in accepting that differences in the Outside-In and Inside-Out theories of grammar acquisition are more of degree than of kind, offers a spectrum of attributes against which psychologically oriented systems of grammar may be compared. Categorial Grammar (CG – Section 4.6) is shown to be consistent with the Coalition Model as it is sensitive to input elements and their arrangement, can cope with a variety of linguistic phenomena, and is shown to be configurable to any e-language.

Chapter 5 builds on the CG parser introduced in Chapter 4. In keeping with the guiding philosophy that resource usage should be as low as possible, this chapter begins by concerning itself with the chart parser at the heart of an implemented CG parser. Two problems are identified and addressed. Firstly, the cubic complexity of the chart parsing algorithm results in high resource usage when processing longer sentences, and secondly non-compositionality of compounds (in a categorial sense) can lead to parse

failures if the constituent words of the compound are shifted into the parser one at a time. The technique of Chunking is presented as a solution (Section 5.1), and justification for its inclusion in a psychological model of text comprehension is presented by reference to Visual Acquisition (Section 5.2.1), the Parallel Recognition Model (Section 5.1.2), and the processing of Garden Path sentences. When processing the Garden Path sentences, it is noted that the induced error involves the incorrect identification of a compound, whereas when processing a parallel, non Garden Path sentence, the individual words (of the compound) are correctly identified. This evidence is used to propose that the chart-parsing algorithm be modified to accept both a compound and its constituent words in parallel (Section 5.2.3). The modified chart and the additional information required by each 'term' shifted into it (i.e. the extent) are defined in Section 5.4. The modified chart is then shown to provide resource-efficient support for 'Parallel Shifts' (Section 5.4) and to allow deactivation of less plausible paths and their reactivation in the event of a parse failure (Section 5.4). Finally, the modified chart is demonstrated to operate in accord with expectations with regard to the sense assignments and structural decisions made and to the occurrence of parse failures and repairs when presented with a Garden Path sentence (Chapter 9).

As a side effect of moving compound construction away from the chart parser and on to the Chunker, it is argued that the CG atomic categories of N and NP may be merged into one category, say NP, which can be used for individual nouns and noun phrases alike. This is used in Section 8.3.3 to show a reduction in the number of syntactic categories needed to support a natural language like English.

Chapter 6 turns its attention away from the chart parser and toward the senses that are ultimately needed for coherence determination. Recognising that senses from a fine-grained sense representation (such as WordNet) will impact heavily on processing resources when combined into knowledge representations that must be evaluated against each other, a more compact sense representation is sought. A novel tree-cut model for the WordNet noun and verb hypernym taxonomies, based on maximum change of information, is presented (Section 6.2). The intuition behind the model is that for any hypernym chain, an increase in information between a class and its descendant corresponds to a specialisation of sense, and the maximum change indicates the class exhibiting the greatest specialisation. The classes exhibiting the greatest information change in any hypernym chain are here named a Specialisation Class (Section 6.3). The technique is used to create heavily abridged versions of those taxonomies, which are shown to retain the sense distinctions of the original taxonomies to a high degree through a retrieval exercise (Section 6.4). It is demonstrated that the abridged taxonomies present a significantly reduced search space (Section 6.3.2) for use by class-based reasoning processes such as Word Sense Discrimination, thereby allowing those processes to operate much more quickly, but with no (or little) loss of information, thereby addressing the real-time processing requirement without incurring a high cognitive loading.

Chapter 7 evaluates the Specialisation Classes of chapter 6 in a Word Sense Disambiguation task using Selectional Association as the disambiguation mechanism. Recognising that Resnik's method of Selectional Association determines the association between a noun class and a verb lemma, a novel extension to the method is presented to

enable noun-class/verb-class associations to be calculated from non sense-tagged training data (Section 7.1.1). In order to reliably compare two senses, a new Sense Indicator String is introduced which eliminates the ambiguity of the WordNet Sense Keys (Section 7.3.1). Two sets of Selectional Association values are generated, one for the original WordNet taxonomies, the other for the abridged versions, and both are used in a comparative WSD task using relations drawn from pre sense-tagged SemCor as the evaluation data. The improved disambiguation results obtained when using Specialisation Classes (Section 7.4.2) shows that Specialisation Classes are better able to model the associations between a verb and its argument than the full range of classes expressed in WordNet. The explanation offered is that by eliminating less informative classes, the maximum SA value search is performed on highly sense-discriminatory classes between which the association is more strongly expressed. It is also shown that the improved disambiguations resulted from fewer calculations, again addressing the real-time and cognitive loading requirements.

Chapter 8 returns to the CG parser and proposes that the structure of syntactic categories as used by CG are an impediment to incremental interpretation by reference to example CG derivations (Section 8.2), where it is shown to be caused by an inability to access the left-embedded left-looking categories until the outer, right-looking categories have been dispatched.

Type-raising allows incremental parsing, but is accompanied by overproduction caused by the introduction of new categories, and hence possible combinations, into the chart. Theories of grammar acquisition are reviewed for inspiration, and an analogy is drawn between the innate but unconfigured grammatical knowledge of a human and a problem

space that expresses all possible syntactic categories (Section 8.3). It is reasoned that configuration involves selection of categories from that space, and experimental evidence is presented to show that very few verbal syntactic categories are used in English when compared with the problem space (Section 8.3.4).

Regularity of verbal form within a human language is used to reason that the selection of categories is not random, but follows a pattern (Section 8.3.5), and data is presented to show that 'configured' syntactic categories can be expressed as trees rather than as discrete categories, and the nodes of the tree exhibit inheritance (Section 8.3.6). This 'Inheritance Model of Syntactic Categories' is presented as evidence for a configuration process (Section 8.3.6), as a means to promote incremental processing (Section 8.4), and to propose that structurally-related syntactic categories are easier to extract from the environment (i.e. learn) than a random selection (Section 8.3.5).

It is shown that by expressing syntactic categories as trees and thereby exploiting the Inheritance Model, the left-embedded left-looking category may be accessed earlier than would be possible when using distinct categories (Section 8.4). In accordance with the rule-to-rule hypothesis, early semantic processing is performed (Section 8.4). This permits plausibility testing of derivations involving the left-embedded category, the resultant sense discrimination and category selection propagating down the tree.

In order to support a notion of sense, a novel extension to the standard CG semantic categories is proposed that allows possible predicate/argument senses to be expressed within the semantic category (Section 8.5).

Chapter 9 combines the Chunking, Sense and Grammar elements and presents a worked example of their operation. To evaluate the system it is proposed that a psychologically oriented pre-processor should exhibit the Garden Path effect when processing garden path sentences. The evaluation showed that:

- The Chunking element presents the input text in suitable chunks, including parallel paths for compounds;
- The chart parser is able to represent the parallel paths, deactivating and reactivating them as necessary;
- The lexicon supplies categories for each term and compound term, and represents all possible senses in the semantic categories;
- Early access to left-embedded left-looking categories enables sense and category discrimination to occur in an incremental fashion, without affecting the overall derivation-building strategy;
- Early access also permits interpretations of partial constituents to be built incrementally, mimicking the human experience of reading and comprehending incrementally;
- The Garden Path effect leads to incorrect selection of the compound and the consequent repair attempts as expected.
- The final derivation is partially sense-disambiguated and correctly grammatically structured.
- A parallel, non-garden path version of the sentence also performed as expected, correctly identifying the non-compound path at the first attempt.

The evaluation shows that the proposed pre-processor meets expectations when processing a Garden Path sentence. More importantly, it is shown during the walkthrough that (im)plausibility of propositions is used to: a) reduce the number of Specialisation Classes (that is, senses) attached to each predicate, thereby disambiguating the words of the sentence, b) deactivate categories in the chart that are deemed implausible, thereby conserving processing resources whilst reducing the possibility of overproduction.

10.3 Summary of Contributions

The overall aim of the thesis was to determine whether elements of the Construction Integration Model are at work in the conversion of text into logical form, which is glossed over in the description of the CIM. The result of the walkthrough (Chapter 9) is presented as evidence that the coherence determination element of the model is instrumental in both sense and grammatical structure discovery, where it is shown that intermediate propositions of a CG parse are checked for plausibility and deactivated if not.

This result has implications for grammar parsing and Word Sense Disambiguation: Research into grammar parsing must take account of the possible senses of the sentential terms, which must themselves be presented as both single terms and compounds wherever possible. Similarly, research into Word Sense Disambiguation techniques will benefit from consideration of the structural relations between the term to be disambiguated and its cohorts that provide the context for the disambiguation decision. In fact, the effect of not taking both aspects into consideration has already been shown in this thesis; Section 2.5.5 presents the disambiguation of the dictionary

definition of ‘Alarm Clock’ through lexical chaining. As grammatical relation is not considered, ‘Alarm’ and ‘Sleeper’ are sense-tagged as ‘Alarm Clock’ and ‘Railway Sleeper’, as both these senses share the hypernym ‘device’. This error could have been avoided if the grammar of the sentence had been taken into consideration.

It is shown that a chart parser cannot necessarily build compounds from individually shifted words, and so cannot take responsibility for the single-words/compound-word decision.

A new, extended version of the chart structure is presented that allows ‘Parallel Shift’ of single-words/compound-word combinations provided by a Chunker, and does so in a resource-economical fashion. The new chart structure also allows deactivation/reactivation of chart columns and categories, allowing sections of it to be effectively ignored on grounds of plausibility (Chapter 5).

A Chunker is employed to identify noun phrases by grouping input terms on the basis of their Part of Speech (for example ‘the:NP/N’ + ‘dog:N’ become ‘The_dog:NP’), controversially removing the need to distinguish between the syntactic categories N and NP when assigning categories to terms prior to shifting into the chart parser. This allows the N and NP atomic categories to be merged into the NP category, leading to a simplification of the lexicon and a reduction in the number of chart operations necessary to process a sentence (Chapter 5).

In order to make plausibility judgements in real-time, the fine-grained WordNet noun and verb hypernym taxonomies are abstracted onto 'Specialisation Classes' through application of a novel tree-cut model based on maximum change in information. Specialisation Classes are shown to significantly reduce the size of the taxonomies (Chapter 6), whilst performing better than the full range of WordNet senses in a Selectional Association based Word Sense Disambiguation task (Chapter 7).

To take full advantage of coherence determination/plausibility testing in a grammar-parsing situation, a re-evaluation of syntactic categories is necessary; by identifying the Inheritance Model of Syntactic Categories it is possible to build interpretations of sentences incrementally (Chapter 8). The Inheritance Model provides access to left-embedded, right-looking categories that would otherwise not be available to participate in derivations until all external, right-looking categories have been dispatched. Incremental interpretation due to the Inheritance Model is begun in Chapter 8 and completed in Chapter 9. A justification of the model in terms of configuration of an innate grammar system is presented in Chapter 8.

A modification to the structure of CG semantic categories is presented that allows those categories to carry sense information with respect to each of their arguments. These are shown to be effective in making plausibility decisions regarding derivations (Chapter 8).

By uniting the sense coherence and grammar elements of the pre-processor at the grammar parser it is proposed that those elements become interrelated, mutually constraining processes. This is demonstrated in Chapter 9.

10.4 Future Research

The work in this thesis presents a 'proof of concept' for a future natural language processor capable of transforming plain text into a sense-tagged logical representation. We have demonstrated the viability of the proposed processor by applying it to two parallel sentences, one of which was a Garden Path sentence. The processor was shown to behave as expected, that is, its behaviour matched that of a human reader; the garden path sentence was initially parsed incorrectly, inducing re-evaluation, whilst the non-garden path sentence was parsed without incident.

We realise that additional testing will be necessary in order to determine the properties of the parser, and propose the following evaluations to achieve this:

10.4.1 Further testing

The sense-tagging (sense disambiguation) abilities of the parser described here requires quantification and comparison with other sense disambiguation schemes. This can be achieved by applying the parser to test datasets such as Senseval 2 [SENSEV], which uses WordNet senses and provides lexical sample (i.e. selected words) and all words tagging exercises, together with software to score the evaluation results.

The logical-form building (structural disambiguation) abilities is perhaps more difficult to evaluate. Basic testing using pre-parsed sentences culled from CG (and other

grammar schemes) textbooks is one possibility. However, comparison of parses against structurally annotated corpora would provide a larger and more realistic sample. Both Clark [CLAR02] and Hockenmaier [HOCK03] have used section 23 of CCGBank as their test corpus, and to follow their example would again provide qualitative and quantitative data. However, as there are differences in structural annotation between a CG logical form and the structural markup used in CCGBank, evaluation of structural parsing is not straightforward. See Hockenmaier [HOCK01] and Clarke and Hockenmaier [CLAR02b] for a discussion of evaluation issues.

As the Parallel-Shift chart parser relies on the Chunker/Recogniser to feed it with all viable phrasal elements as derived from the input terms, the abilities of the Chunker/Recogniser to identify those units requires evaluation. Again this may be achieved by processing texts previously marked-up for grammatical structure. However, as we need only to confirm that the required unit is present in the collection of possibilities, and are not concerned with the structure between the units, evaluation will be easier than that for logical structure.

Given parallel phrasal units, the Parallel-Shift chart parser must select the most viable from those units. Again, no structural relationships are involved - we need only compare the selected unit to that marked in the evaluation texts.

Finally, comparing the effect on processing time (or effort) the introduction of sense, parallel shift, Specialisation Class and Syntactic Inheritance has on a set of texts compared with a standard parser would be interesting. Our hope is that, although the parser presented here would seem to have more work to do in implementing the additional processes, the mutually constraining processes the parser embodies will lead

to early elimination of erroneous derivations and senses, thereby reducing the overall workload.

10.4.2 Follow-up work

In addition to further testing there are four aspects of this work that we find particularly interesting and hope to follow-up in the future:

The first concerns the determination of coherence. This work employed Selectional Association as the means to determine coherence, but it is not the only method available. For example, lexical chains, dictionary definition overlap, and semantic nets have all been employed in this respect. The real interest for us however would be to use a sense-tagged corpus such as SemCor, where disambiguation decisions have been made by human annotators, and determine whether different lexical/sense/grammatical situations call for different approaches to coherence determination. Metadata of this type would be very useful in the design of both coherence determination algorithms and the knowledge bases they use.

The second concerns the Specialisation Classes developed during the course of this work. They have been shown to heavily abridge the WordNet noun and verb classes whilst retaining the sense distinctions expressed hypernymically, and to give improved performance over the full set of WordNet senses in a WSD task. Although they have been used as static knowledge in the work presented here, the Specialisation Class selection algorithm is capable of generating sets of SCs at different levels of abstraction. These could be employed in a sense-tagging system that initially makes course-grained sense tagging decisions by using a highly abstract dataset, followed by repeated

operations at lower levels of abstraction. This would enable the system to 'home-in' on fine-grained sense tags. It would also be interesting to discover whether this 'divide and conquer' approach to sense tagging reduces the overall workload as, at high levels of abstraction there is little knowledge to process, whereas at later, higher levels more knowledge is available, but operates on the fewer remaining word senses.

The third concerns the Inheritance Model of Syntactic Categories. As noted in the thesis (Section 8.6), the model is currently defined only for categories in the series a\b, (a\b)/c, ((a\b)/c)/d …, and it must be determined if this is a limitation of the model or a special case of a more general model before the it can be usefully deployed as a component of CG.

The fourth again concerns the Inheritance Model, but here the fact that categories are related structurally might be useable by allowing the viability of automatically extracted categories to be determined by their structure rather than frequency of occurrence

Having demonstrated that coherence-determination acting on sensed propositions is an important factor in producing sense-tagged, correct grammatical structures, of general interest would be a parallel-shift enabled, Inheritance Model aware CG parser that accepts plug-in grammar and coherence modules. This would then provide a useful research tool for both grammars and knowledge bases for use in coherence determination.

11 References

ABNE90 Abney, S. 1990. Rapid Incremental Parsing with Repair In: Proceedings of the 6th New OED Conference: Electronic Text Research, pp.1-9. University of Waterloo, Waterloo, Ontario.

ABNE91 Abney, S. 1991. Parsing by Chunks. In: Berwick, R., Abney, S. and Tenny, C. (eds.). Principle based Parsing, p257-278. Dordrecht: Kluwer.

ABNE96 Abney, S. 1996. Chunk Stylebook. Manuscript. 1996. Unpublished manuscript. http://www.vinartus.net/spa/96i.pdf (Accessed Jan 2005)

ALTM88 Altmann, G. 1988. Ambiguity, Parsing Strategies, and Computational Models. Language and Cognitive Processes, 3, pp. 73-98.

AMIT00 Amitay, E. and Paris, C. 2000. Automatically summarising web sites – is there a way around it? Proceedings of the ninth international conference on Information and knowledge management, McLean, Virginia, United States, pp. 173-179

APPE93 Appelt, D.E., Hobbs, J.R. Bear, J., Israel, D. and Tyson, M. 1993. FASTUS: A finite-state processor for information extraction from real-world text. In: Proceedings of the IJCAI'98, Chambery, France.

BAR53 Bar-Hillel, Y. 1953. A Quasi-arithmetical Notation for Syntactic Description. Language, 29, 47-58.

BARZ97 Barzilay, R. and Elhadad, M. 1997. Using lexical chains for text summarisation. MANI97, pp 10-17.

BATE87 Bates, E. and MacWhinney, B. 1987, Competition, variation and language learning. InMacWhinney (ed.), Mechanisms of language acquisition. Lawrwnce Erlbaum Associates, Hillsdale, NJ.

BATE89 Bates, E. and MacWhinney, B. 1989. Functionalism and the Competition Model. In MacWhinney, B. and Bates, E. (eds.), The Crosslinguistic Study of Sentence Processing. Cambridge University Press, Cambridge.

BERG00 Berger, A. and Mittal, V. 2000. OCELOT: A system for summarising web pages. Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, p.144-151, Athens, Greece.

BERR92 Berry, M.W. 1992. Large scale singular value computations. International Journal of Supercomputer Applications, 6(1), pp. 13-49

BERR95 Berry, M.W., Dumais, S.T., O'Brien, G.W. 1995. Using linear algebra for intelligent information retrieval. SIAM: Review, 37(4), pp. 573-595

BERN00 Berners-Lee, T., Hendler, J. and Lassila, O. 2000. Semantic web. Scientific American, 1(1): pp. 68-88.

BERZ79 Berzon, V.E., Brailovskii,A.B., 1979. The classification of connectors and conversational systems for automatic abstracting. In: Automatic Documentation and Mathematical Linguistics, 13,6, pp 32-40.

BEVE70 Bever, T. 1970. The Cognitive Basis for Linguistic Structure. In: Hayes, J. (ed.), Cognition and the Development of Language.pp. 279-362. New York: Wiley.

BEVE88 Bever, T. and McElree, B. 1988. Empty categories access their antecedents during comprehension. Linguistic Enquiry. 19, 34-43.

BLOO75 Bloom, L., Lightbrown, P. and Hood, L. 1975. Structure and Variation in Child Language. Monographs of the Society for Research in Child Development. 40 (Serial No. 160).

BNC http://www.natcorp.ox.ac.uk/ (Accessed Feb 2005)

BRUC94 Bruce, R. and Weibe, J. 1994. Word-sense disambiguation using decomposable models. In: Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, pp. 139-145. Las Cruces, New Mexico.

BRUN75 Brunner, J. 1975. The ontogenesis of speech acts. Journal of Child Language. 2, 1-19.

CATT86 Cattell, J. 1886. The time taken up by cerebral operations. Mind, 11. pp. 277-282, 524-538.

CHOD88 Chodorow, M., Ravin, Y., Sachar, H. 1998. Tool for investigating the Synonymy Relation in a Sense Disambiguated Thesaurus. In: Proceedings of the Second Conference on Applied Natural Language Processing, pp, 144-155.

CHOI99 Choi, F. 1999. A flexible distributed architecture for NLP system development and use. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (Student Session), pp. 615-618. College Park, USA.

CHOI00 Choi, F. 2000. Advances in Domain Independent Linear Text Segmentation. Proceedings of NAACL'00. Seattle, USA.

CHOM57 Chomsky, N. 1957. Syntactic Structures, Mouton, The Hague.

CHOM72 Chomsky, N. 1972. Studies on semantics in generative grammar. The Hague, Moulton.

CHOM75 Chomsky, N. 1975. Reflections on Language. New York. Random House.

CHOM81 Chomsky, N. 1981. Lectures on government and binding. Dordrecht: Foris.

CHOM82 Chomsky, N. 1982. Some Concepts and Consequences of the Theory of Government and Binding, Cambridge, Mass.: MIT Press.

CHOM88 Chomsky, N. 1988. Language and Problems of Knowledge: The Managua Lectures. The MIT Press, Cambridge, MA.

CHOM96 Chomsky, N. 1996 Powers and Prospects, Boston, MA: South End Press.

CLAR02 Clark, S. 2002. A Supertagger for Combinatory Categorial Grammar. In: Proceedings of the TAG+ Workshop, p.19-24. Venice, Italy.

CLAR02b Clark, S. and Hockenmaier, J. 2002. Evaluating a wide-coverage CCG parser. In: Proceedings of the LREC Beyond PARSEVAL workshop, Las Palmas, Spain.

CLIM61 Climenson, W.D., Hardwick, N.H., Jacobson, S.N., 1961. Autiomatic syntax analysis in machine indexing and abstracting. In: American Documentation 12,3, pp 178-183.

COHE01 Cohen, W. and Jensen, L. 2001. A structured wrapper induction system for extracting information from semi-structured documents. In: Proceedings of the Workshop on Adaptive Text Extraction and Mining (IJCAI'01).

COHE93 Cohen, L.B., and Oakes, L.M. 1993. How infants perceive a simple causal event. *Developmental Psychology* 29, 421-433.

CRAI85 Crain, S. and Steedman, M. 1985. On not being led up the garden path: the use of context by the psychological parser. In: Kartunnen, L., Dowty, D. and Zwicky, A. (eds.), Natural Language Parsing: Psychological, Computational and Theoretical Perspectives. pp. 320-385. Cambridge: Cambridge University Press.

CRYS85 Crystal, D., 1985. A Dictionary of Linguistics and Phonetics. Basil Blackwell Ltd.

DEER90 Deerwester, S., Dumais, S.T., Furnas, G.W., Landaur, T.K., Harshman, R. 1990. Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science, 41(6), pp. 391-407

DILL97 Diller, K.C. 1997. Representation and Reality: Where Are the Rules of Grammar? New Trends in Cognitive Science 1997.

DILL03 Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A., Tomlin, J.A., and Zien, J.Y. 2003. Semtag and seeker: bootstrapping the semantic web via automated semantic annotation. In: Proceedings of the 12th International World Wide Web Conference, pp. 178-186. ACM Press.

DEJO82 DeJong, G. 1982. An overview of the FRUMP system. In: Lehnert, W.G., Ringle, M.H. (eds.) Strategies for natural language processing. Hillsdale, NJ, Earlbaum.

DUDA75 Dudani, S. 1975. The distance-weighted k-nearest-neighbour rule. IEEE Transactions on Systems, Man and Cybernetics, SMC-6(4): pp. 325-327.

DUMA91 Dumais, S.T. 1991. Improving the retrieval of information from external sources. Behaviour Research Methods, Instruments and Computers. 23(2), pp. 229-236

DUMM86 Dummett, M.A.E. 1986. Comments on Davidson and Hacking. In Lepore, E. (ed.) Truth and Interpretation, Oxford: Blackwell.

EARL70 Earley, J. 1970. An efficient context-free parsing algorithm, Communications of the ACM, v.13 n.2, pp.94-102

EDMU61 Edmundson, H.P., 1961. Automatic abstracting and indexing: Survey and recommendations. In: Communications of the ACM 5:5, pp 226-235.

EDMU63 Edmundson, H.P., 1963. Automatic abstracting. TRW Computer Division, Thompson Ramo Wooldridge Inc., Canoga Park, CA.

EDMU69 Edmundson, H.P., 1969. New methods in automatic extracting. Journal of the ACM, 16(2), 1969.

EDWA02 Edwards,P., Grimnes, G.A. and Preece, A.D. 2002. An Empirical Investigation of Learning From the Semantic Web. In: Proceedings of ECML/PKDD-2002 Semantic Web Mining Workshop.

EISN96 Eisner, J. Efficient Normal-Form Parsing for Combinatory Categorial Grammar. In: Proceedings of the 34th Meeting of the ACL, p.79-86, Santa Cruz. USA.

EJER83 Ejerhed, E. and Church, K. 1983. Finite state parsing, Papers from the Seventh Scandanavian Conference on Linguistics, Universitu of Helsinki, Finland.

ENDR98 Endres-Niggemeyer, B. 1998. Summarising Information. Springer-Verlag.

ENGB02 Engbert, R., Longtin, A., and Kliegl R. 2002. A dynamical model of saccade generation in reading based on spatially distributed lexical processing. Vision Research, 42, 621-636.

ERIK86 Eriksen, C.W. and St James, J.D. 1986. Visual attention within and around the field of focal attention: A zoom lens model. Perception and Psychophysics, 40, pp. 225-240.

FELL98 Fellbaum, C. 1998. WordNet: An Electronic Lexical Database. MIT Press, Cambridge, MA.

FERR90 Ferreira, F. 1990. The use of verb information in syntactic parsing: Evidence from eye movements and word-by-word self-paced reading. Journal of Experimental Psychology: Learning, Memory and Cognition. 16, pp. 555-569.

FIRT57 Firth, J,R, 1957. A synopsis of linguistic theory 1930-1955. In: Studies in Linguistic Analysis. Oxford: Philological Society. Reprinted in in Palmer, F.R. (ed), Selected Papers of J.R. Firth 1952-1959, London: Longman, 1968. pp, 1-32

FODO87 Fodor, J.D. and Crain, S. 1987. Simplicity and generation of rules in language acquisition. In MacWhinney, B. (ed.), Mechanisms of language acquisition. Hillsdale, NJ.: Lawrence Erlbaum Associates.

FOLT99 Foltz, P.W., Laham, D., and Landauer, T.K. 1999. The Intelligent Essay Assessor: Applications to Educational Technology. Interactive Multimedia Electronic Journal of Computer-Enhanced Learning, 1(2).

FRAN82 Francis, W.N. and Kucera, H. 1982. Frequency Analysis of English. Houghton Mifflin Company, Boston.

FUM82 Fum, D., Guida, G. and Tasso, C. 1982. Forward and backward reasoning in automatic abstracting. COLING82. Proceedings of the 9th International Conference on Computational Linguistics. pp 83-88. Prague.

FUM84 Fum, D., Guida, G. and Tasso, C. 1984. A propositional language for text representation. Bara, B. and Guida, G. (eds.). Computational models of natural language processing. pp 121-150. Amsterdam.

FUM85a Fum, D., Guida, G. and Tasso, C. 1985. Evaluating Importance: A step towards text summarisation. IJCAL85. Proceedings of the 9th International Joint Conference on Artificial Intelligence. Pp 840-844. Los Altos, CA. Kaufmann.

FUM85b Fum, D., Guida, G. and Tasso, C. 1985. A rule-based approach to evaluating importance in descriptive texts. Proceedings of the 2nd Conference of the European Chapter of the Association for Computational Linguistics. pp 244-250. Geneva.

GALE92 Gale, W., Church, K., Yarowsky, D. 1992. One Sense Per Discourse., In Proceedings of the ARPA Workshop on Speech and Language Processing, pp. 233-237.

GAZD85 Gazdar, G., Klein, E., Pullum, G.K., Sag, I.A. 1985. Generalized Phrase Structure Grammar, Blackwell, Oxford.

GIBS92 Gibson, E. 1992. On the adequacy of the competition model. Language. 68, 812-830.

GLEI90 Gleitman, L.R. 1990. The structural sources of verb meanings. Language Acquisition. 1, 3-55.

GLUC85 Gluck, M.A. and J.E. Corter, J.E. 1985. Information, uncertainty and the utility of categories. In: Proceedings of the Seventh Annual Conference of the Cognitive Science Society: Lawrence Erlbaum Associates, 1985, pp. 283-287.

GOLI81 Golinkoff, R.M. 1981. The case for semantic relations: Evidence from the verbal and nonverbal domains. *Journal of child language* 78, 413-438.

GOLI84 Golinkoff, R.M., Harding, C.G. Carlson-Luden, V., and Sexton, M.E. 1984. The infant's perception of causal events: The distinction between animate and inanimate objects. In: Lipsitt, L.P. (ed.), *Advances in infancy research.* Vol. 3. Norwood, N.J. : Ablex.

GOMU56 Gomulicki, B.R. 1956. Recall as an abstractive process. Acta Psychologica, 12, pp77-94.

GREF92 Grefenstette, G. 1992. Use of syntactic context to produce association lists for text retrieval. In: Proceedings of the SIGIR'92, Copenhagen, Denmark.

GONZ99 Gonzalo, J., Penas, A., Verdejo, F. 1999. Lexical ambiguity and Information Retrieval revisited. In Proceedings of EMNLP/VLC'99..

GONZ01 Gonzalo, J., Fernandez-Amoros, D., and Verdejo, F. 2001. The Role of Conceptual Relations in Word Sense Disambiguation. Proceedings of the 6th International Workshop on Applications of Natural Language for Information Systems. (NLDB'01).

GURN04 Gurney, K., Prescott, T.J., Wickens, J.R. and Redgrave, P. 2004. Computational models of the basal ganglia: from robots to membranes. Trends in Neurosciences, 27, pp. 453-459.

HAEG94 Haegeman, L. 1994. Introduction to Government and Binding Theory (2nd ed.), Blackwell, Oxford.

HAHN90 Hahn, U. 1990. TOPIC Parsing: Accounting for text macro structures in full text analysis. Information Processing and Management. 1:2,131-153

HAHN98 Hahn, U. and Schnattinger, K. 1998. Towards text knowledge engineering. In: AAAI'98, Proceedings of the 15th National Conference on Artificial Intelligence.

HARA98 Harabagiu, S.M. and Moldovan, D.J. 1998. Knowledge Processing on an extended WordNet. In: Fellbaum, C. (ed), WordNet: An Electronic Lexical Database. 305-332. MIT Press.

HARR51 Harris, Z.S. 1951. Structural Linguistics. Chicago: University of Chicago Press.

HALL76 Halliday, M. A. K., and Hasan, R. 1976. Cohesion in English. London: Longman.

HEAR94 Hearst, M.A. 1994 Multi-paragraph segmentation of expository text. In: Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics.

HEND88 Henderson, J.M. 1988. Visual Attention and the acquisition of extrafoveal information during eye fixations. Doctoral dissertation, University of Massachusetts, Amherst, MA.

HEND91 Henderson, J.M. 1991. Stimulus discrimination following covert attentional orienting to an exogenous cue. Journal of Experimental Psychology: Human Perception and Performance, 17, pp. 91-106.

HEND95 Henderson, J.M. and Ferreira, F. 1995. Eye Movement Control During Reading: Fixation measures Reflect Foveal but not Parafoveal Processing Difficulty. In: Henderson, J.M., Singer, M., and Ferreira, F. (eds), Reading and Language Processing. Lawrence Erlbaum Associates, NJ.

HIRS87 Hirst, G. 1987. Semantic Interpretation and the Resolution of Ambiguity: Cambridge: Cambridge University Press.

HIRS98 Hirst, G. and St-Onge, D. 1998. Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms. In: Fellbaum, C. (ed), WordNet: An Electronic Lexical Database. 305-332. MIT Press.

HIRS99 Hirsh-Pasek, K. and Golinkoff, R.M. 1999. The Origins of Grammar: Evidence from Early Language Comprehension. The MIT Press, Cambridge, MA.

HOCK01 Hockenmaier, J. 2001. Statistical parsing for CCG with simple generative models. In: Proceedings of Student Research Workshop, 39th Annual

Meeting of the Association for Computational Linguistics and 10th Meeting of the European Chapter, pp. 7–12, Toulouse, France.

HOCK03 Hockenmaier, J. 2003. Data and Models for Statistical Parsing with Combinatory Categorial Grammar. Ph.D. thesis, University of Edinburgh.

HOVY97 Hovy, E. and Lin, C.Y. 1997. Automated text summarisation in SUMMARIST. MANI97, pp 18-24.

JACO90 Jacobs, P.S. and Rau, L.F. 1990. SCISOR: Extracting information from online news. Communications of the ACM 33:11, pp 88-97.

JIAN97 Jiang, J. and Conrath, D. 1997. Semantic Similarity based on Corpus Statistics and Lexical Taxonomy. Proceedings of the International Conference on Research in Computational Linguistics, Taiwan.

JOSH75 Joshi, A. Levy, L. and Takahashi, M. 1975. Tree adjunct grammars. Journal of. Computing Systems Science. 10(1), pp.136-163

KASA65 Kasami, I. 1965. An efficient recognition and syntax algorithm for context-free languages. Technical Report AFCRL-65-758, Air Force Cambridge Research Lab., Bedford, MA.

KARM89 Karmiloff-Smith, A. 1989. Commentary. Human Development. 32, 272-275.

KAY96 Kay, M. 1996. Chart Generation. In: Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics. pp 200-204.

KEEN88 Keenan, E.L. and Timberlake, A. 1988. Natural language motivations for extending categorial grammar. In: Oehrle, R., Bach, E. and Wheeler, D.

(eds), 1987, Categorial Grammars and Natural Language Structures, Dordrecht: Reidel, pp 265-299.

KINT73 Kintsch, W. and Keenan, J, M. 1973. Reading rate and retention as a function of the number of propositions in the base structure of sentences. Cognitive Psychology, 5, pp 257-274.

KINT74 Kintsch, W. 1974. The representation of meaning in memory. Hilsdale, NJ: Lawrence Erlbaum Associates Inc.

KINT78 Kintsch, W and van Dijk, T.A. 1978. Towards a model of text comprehension and production. Psychological Review 85, pp363-394.

KINT88 Kintsch, W. 1988. The role of knowledge in discourse comprehension: A construction-integration model. Psychological Review, 95, pp163-182.

KINT90 Kintsch. W., Welsch, D., Schmalhofer, F. and Zimny, S. 1990. Sentence memory: A theoretical analysis. Journal of Memory and Language, 29, pp133-159.

KINT92 Kintsch, W. 1992. A cognitive model for comprehension. In H.L.Pick, P. van den Broek, and D.C. Knill (Eds.), Cognition: Conceptual and methodological issues. Washington, DC: American Psychological Association.

KINT94 Kintsch, W. 1994. The psychology of discourse processing. In M.A. Gernsbacher (Ed.), Handbook of psycholinguistics. London: Academic Press.

KOZI93a Kozima, H. 1993. Text segmentation based on similarity between words. In: Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics, pp. 286-288

KOZI93b Kozima, H., and Furugori, T. 1993. Similarity of words computed by spreading activation on an English dictionary. In: Proceedings of the 6th Conference of the European Chapter of the Association for Computational Linguistics, pp. 232-239

KOZI94 Kozima, H., and Furugori, T. 1994. Segmenting narrative text into coherent scenes. Literary and Linguistic Computing, vol. 9, pp. 13-19

KROV97 Krovetz, R. 1997. Homonymy and Polysemy in Information Retrieval. In: Proceedings of ACL/EACL'97.

KROV98 Krovetz, R. 1998. More than One Sense Per Discourse. NEC Princeton NJ Labs., Research Memorandum.

KUCE67 Kucera, H., and Francis, W. N. 1967. Computational Analysis of Present-Day American English. Providence, Rhode Island: Brown University Press.

KUCZ77 Kuczaj, S.A. 1977. On the acquisition of regular and irregular past tense forms. Journal of Learning and Verbal Behavior. 16, 589-600.

KUHL89 Kuhlen, R., Hammwoehner, R. and Theil, U. 1989. TWRM-TOPOGRAPHIC. Informatik Forschung und Entwickling. 4:89-107

KUPI95 Kupiec, J., Pedersen, J. and Chen, F. 1995. A trainable document summarizer. SIGI95, pp 68-73.

KUSH97 Kushmerick, N., Weld, D.S., and Doorenbos, R.B. 1997. Wrapper induction for information extraction. In: International Joint Conference on Artificial Intelligence (IJCAI), pp. 729-737.

LAMB58 Lambek, J. 1958. The mathematics of sentence structure. American Mathematical Monthly, 65, pp.154-170.

LAND96 Landaur, T.K., Dumais, S.T. 1996. How come you know so much? From practical problem to theory. In: Hermann, D., McEvoy, C., Johnson, M., and Hertel, P. (Eds.), Basic and applied memory: Memory in context. Mahwah, NJ: Erlbaum, pp. 105-126

LAND97 Landaur, T.K., Dumais, S.T. 1997. A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction and representation of knowledge. Psychological Review, 104(2). pp. 211-240

LAND98 Landes, S., Leacock, C., Tengi, R. 1998. Building Semantic Concordances. In: Fellbaum, C. (ed), WordNet: An Electronic Lexical Database, 199-216. MIT Press, Cambridge, MA. pp. 199-216.

LARS04 Larson, K. 2004. The Science of Word Recognition ,or how I learned to stop worrying and love the bouma. Microsoft (July 2004). http://www.microsoft.com/typography/ctfonts/WordRecognition.aspx. Accessed 26 January 2005

LEAC98 Leacock, C. and Chodorow, M. 1998. Combining Local Context and WordNet Similarity for Word Sense Identification. In: Fellbaum, C. (ed), WordNet: An Electronic Lexical Database. 265-283. MIT Press.

LERM01 Lerman, K., Knoblock, C. and S. Minton. 2001. Automatic data extraction from lists and tables in web sources. In: IJCAI-2001 Workshop on Adaptive Text Extraction and Mining, August 2001.

LESK86 Lesk, M**.** 1986. Automatic Sense Disambiguation using Machine Readable Dictionaries: How to tell a Pine Cone from an Ice Cream Cone. Proceedings of SIGDOC ’86.

LI95a Li, H. and Naoki, A. 1995. Generalising Case Frames using a Thesaurus and the MDL Principle. Proceedings of Recent Advances in Natural Language Processing, 239-248.

LI95b Li, X., Szpakowics, S. and Matwin, S. 1995. A WordNet-based Algorithm for Disambiguation. Proceedings of IJCAI-95. Montreal, Canada.

LI96 Li, H. and Naoki, A. 1996. Learning Dependancies between Case Frame Slots. Proceedings of the Sixteenth International Conference on Computational Linguistics, 10-15.

LIN93 Lin, D. 1993. Principle Based Parsing without Overgeneration. In 31st Annual Meeting of the Association for Computational Linguistics. Columbus, Ohio. pp 112-120. Downloadable from: http://www.cs.ualberta.ca/~lindek/minipar.htm (Accessed Feb. 2005)

LIN97 Lin, D. 1997. Using Syntactic Dependancy as a Local Context to Resolve Word Sense Ambiguity. Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, 64-71, Madrid, 1997.

LUHN58 Luhn, H. P. 1958. The Automatic Creation of Literature Abstracts. . IBM Journal of Research and Development, 2, 159-165.

MAND88 Mandler,J.M. 1988. How to build a baby: On the development of an accessible representational system. *Cognitive Development* 3, 113-136.

MAND92 Mandler,J.M. 1992. How to build a baby: II. Conceptual primitives. *Psychological Review* 99, 587-604.

MATH72 Mathis, B.A. 1972. Techniques for the evaluation and improvement of computer-produced abstracts. Ohio State University Technical Report OSU-CISRC-TR-72-15. Columbus, Ohio.

MATH73 Mathis, B.A., Rush, J.E., Young, C.E., 1973. Improvement of automatic abstracts by the use of structural analysis. In: Journal of the American Society for Information Science, 24, 1973 (2), pp101-109.

MCCO75 McMonkie, G.W. and Rayner, K. 1975. The span of the effective stimulus during a fixation in reading. Perception and Psychophysics, 17, 578-586.

MCKO80 McKoon, G. and Ratcliff, R. 1980. Priming in item recognition: The organisation of propositions in memory for text. Journal of Verbal Learning and Verbal Behaviour, 19, pp369-386.

MEGY02 Megyesi, B. 2002. Shallow Parsing with PoS Taggers and Linguistic Knowledge. Journal of Machine Learning Research. 2

MELC88 Melcuk, I. A. 1988. “Dependency Syntax: Theory and Practice”, State University of New York Press, Albany.

MILL56 Miller, G.A. 1956. The magic number seven, plus or minus two: Some limits on our capacity for processing information. Psychological Review, 63, 81-93.

MILL90 Miller, G. A. (Ed). 1990. WordNet: An on-line lexical database. Special issue of International Journal of Lexicography, 3(4).

MILL95 Miller,G.A. 1995. WordNet: A Lexical Database. Communication of the ACM. 38(11): 39-41.

MILW92 Milward, D. 1992. Dynamics, Dependancy Grammar and Incremental Interpretation. In: Proceedings of the 14th International Conference on Computational Linguistics, Coling 92, Nantes, p1095-1099.

MITC97 Mitchell, T. 1997. Bayesian Learning, chapter 6, Machine Learning, pp. 154-200. McGraw-Hill.

MITR97 Mitra, M., Singhal, A. and Buckley, C. 1997. Automatic text summarisation by paragraph extraction. In: Mani, I. And Maybury, M. (eds.), Proceedings of the ACL/EACL '97 Workshop on Intelligent Scalable Text Summarization, Madrid, Spain.

MOLI02 Molina, A. and Pla, F. 2002. Shallow Parsing using Specialised HMM. Journal of Machine Learning Research. 2

MORR84 Morrison, R.E. 1984. Manipulation of stimulus onset delay in reading: Evidence for parallel processing of saccades. Journal of Experimental Psychology: Human Perception and Performance, 10, pp. 667-682.

MORR88 Morris, J. 1988. Lexical cohesion, the thesaurus, and the structure of text. Master's thesis, Dept. of Computer Science, University of Totonto. (Tech. Report No. CSRI-219)

MORR91 Morris, J., Hirst, G. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of the text. In: Computational Linguistics, 18(1) pp. 21-45.

NELS85 Nelson, K. 1985. Making Sense: The acquisition of shared meaning. Academic Press, Orlando, Fla.

NIST71 Nistor, E., Roman, E., 1971. Constructing automatic abstracts from kernel sentences. In: Cahiers de Linguistique Theorique et Appliquee, 8,8, pp 249-256.

OLSE97 Olsen, M.B. and Resnik, P. 1997. Implicit object constructions and the (in)transitivity continuum. In: Papers from the 33rd Regional Meeting of the Chicago Linguistic Society. Chicago: University of Chicago, Chicago Linguistic Society.

OPENDP Open Directory Project. http://dmoz.org

OWL W3C. Web ontology language. http://www.w3.org/2001/sw/WebOnt/.

PAVL92 Pavlidis, T. and Zhou, J. 1992. Page segmentation and Classification. CVGIP: Graphical Models and Image Processing, Vol. 54, 6. pp 484-496.

PERL84 Perlmutter, P. and Postal, P. 1984. The 1-Advancement Exclusiveness law. In: Perlmutter, D. and Rosen, C. (eds) Studies in Relational Grammar 2, Chicago, Chicago University Press.

PINK84 Pinker, S. 1984. Language learnability and language development. Harvard Academic Press, Cambridge, MA.

PLUN95 Plunkett, K. 1995. Connectionist approaches to language acquisition, In Fletcher, P. and MacWhinney, B. (eds.). The Handbook of Child Language. Blackwell, Cambridge, MA.

POLL75 Pollock, J., Zamora, A., 1975. Automatic Abstracting Research at Chemical Abstracts Service. Journal of Chemical Information and Computer Sciences, 15,4.

POLL90 Pollatsek, A. and Rayner, K. 1990. Eye movements and lexical access in reading. In: Balota, D.A., Flores d'Arcais, and Rayner, K. (eds.), Comprehension Processes in Reading. Hillsdale, NJ., Erlbaum

POLL94 Pollard, C., Sag, I. 1994. Head-Driven Phrase Structure Grammar, CLSI, Chicago.

PORT80 Porter, M. F., An Algorithm for Suffix Stripping. Program, Vol. 14, No. 3, July 1980, pp 130-137.

POWE00 Powell, C., Zajicek, M. and Duce, D., 2000, The Generation of Representations of Word Meanings from Dictionaries. ICSLP2000. International Conference on Spoken Language Processing, Beijing.

PROC78 Procter, P. (Ed.). 1978. Longman Dictionary of Contemporary English. Longman Group Ltd., Essex, UK.

QUIN89 Quinlan, J. R. and Rivest, R. L. 1989. Inferring Decision Trees using the Minimum Description Length principle. Information and Computation, 80:227-248.

RADA89 Rada, R., Mili, H., Bicknell, E., and Blettner, M. 1989. Development and Applications of a Metric on Semantic Nets. IEEE Transactions on Systems, Man and Cybernetics, 19(1):17-30.

RATC78 Ratcliff, R. and McKoon, G. 1978. Priming in item recognition: Evidence for the propositional structure of sentences. Journal of Verbal Learning and Verbal Behaviour, 20, pp204-215.

RATN96 Ratnaparkhi, A., 1996. A Maximum Entropy Model for Part-of-Speech Tagging, Proceedings of EMNLP-1, University of Pennsylvania, PA, USA.

RAU89 Rau, L.F., Jacobs, P.S. and Zernik, U. 1989. Information extraction and text summarisation using linguistic knowledge acquisition. Information Processing and Management, 25:4, pp 419-428.

RAYN94 Raynar, J.C. 1994. An automatic method of finding topic boundaries. In: Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics.

REIC98 Reichle, E.D., Pollatsek, A., Fisher, D. L. and Rayner, K. 1998. Toward a model of eye movement control in reading, Psychological Review 105, 125-257.

RESN95a Resnik, P. 1995. Using information content to evaluate semantic similarity in a taxonomy. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence, pp 448-453. San Francisco: Morgan Kaufmann.

RESN95b Resnik, P. 1995. Disambiguating noun groupings with respect to WordNet senses. In: 3rd Workshop on Very Large Corpora.

RESN96 Resnik, P. 1996. Selectional Constraints: An information theoretic model and its computational realization. Cognition, 61. 127-159.

RESN97 Resnik, P. 1997. Selectional preference and sense disambiguation. In: Proceedings of ACL Siglex Workshop on Tagging Text with Lexical Semantics, Why, What and How? Washington, April 4-5, 1997.

RESN98 Resnik, P. 1998. WordNet and Class-Based Probabilities. In: Fellbaum, C. (ed) WordNet, An Electronic Lexical Database. 239-264. MIT Press.

ROSE84 Rosen, C. 1984. The interface between semantic roles and initial grammatical relations. In: Perlmutter, D. and Rosen, C. (eds) Studies in Relational Grammar 2, Chicago, Chicago University Press.

ROSE97 Rose, T., and Wyard, R. 1997. A Similarity-Based Agent for Internet Searching. Proceedings of RIAO'97.

RUME75 Rumelhart, D.E. 1995. Notes on schemata for stories, In D.G. Bobrow, A. Collins (Eds), Representation and Understanding. Studies in Cognitive Science. New York Academic Press, pp. 211-236.

RUME77 Rumelhart, D.E. 1977. Understanding and summarising brief stories. In Laberge D. and Samuels S.J. (eds). Basic processes in reading: perception and comprehension. pp. 265-303. Lawrence Erlbaum Associates.

RUSS79 Russell, P. 1979. The Brain Book. Routledge and Kegan Paul, London. ISBN: 0 7100 0386 2

SAMS95 Sampson, G. 1995. English for the Computer: The SUSANNE Corpus and analytic scheme. Oxford University Press. Corpus available from: http://www.grsampson.net/RSue.html (Accessed Feb. 2005)

SALT97 Salton, G, Singhal, A., Mitra, M. and Buckley, C. 1997. Automatic Text Structuring and Summarisation. Information Processing and Management, 33:2, pp 193-208.

SCHL71 Schlesinger, I.M. 1971. Production of Uterences and Language Acquisition. In Slobin, D.I. (ed.), The ontogenesis of grammar. Academic Press, New York.

SCHL79 Schlesinger, I.M. 1979. Cognitive and linguistic structures: The case of the instrumental. *Journal of Linguistics* 15, 307-324.

SCHL88 Schlesinger, I.M. 1988. The Origins of Relational Categories. In Levy, Y., Schlesinger, I.M., and Braine, M.D.S., (eds). Categories and Processes in Language Acquisition. Lawrence Erlbaum Associates, Hillsdate, NJ.

SENSEV SENSEVAL-2 website: **http://www.sle.sharp.co.uk/senseval2**

SKOR72 Skorodoch'ko, E. F. 1972. Adaptive method of automatic abstracting and indexing. In: Information Processing 7: Proceedings of the IFIP Congress 71. Freiman, C.V. (ed). North-Holland Publishing Company. pp 1179-1182

SMAL82 Small, S. and Reiger, C. 1982. Parsing and comprehension with word experts. LEHN82, pp 89-147

STAB91 Stabler, E. 1991. Avoid the Pedestrian's Paradox. In: Berwick, R., Abney, S. and Tenny, C. (eds.). Principle based Parsing, p199-238. Dordrecht: Kluwer.

STEE96 Steedman, M. 1996. Surface Structure and Interpretation. MIT Press, Cambridge, MA.

STEE00 Steedman, M. 2000. The Syntactic Process. MIT Press, Cambridge, MA.

TEFU97 Tefuel, S. and Moens, M. 1997. Sentence extraction as a classification task. MANI97, pp 58-65.

TJON02 Tjong Kim Sang, E.F. 2002. Memory Based Shallow Parsing. Journal of Machine Learning Research. 2.

TRAB85 Trabasso, T. and Sperry, L.L. 1985. Causal relatedness and importance of story events. Journal of Memory and Language, 24, 595-611.

TWAD48 Twaddle,W.F. 1948. Meanings, habits and rules. Education vol. 4.

VAND77 Van Dijk, T.A, 1977. Complex semantic information processing. In: Walker, D.E. et al. (eds.) Natural language in information science Stockholm: Skriptor, pp.127-163.

VAND79 van Dijk, T.A., 1979, Recalling and summarising complex discourse. In: Burchart, W., Hulker, K., (Eds.), Text Processing, Walter de Gruyter, Berlin.

WALK91 Walker, M. 1991. Redundancy in collaborative dialog. In: AAAI Fall Symposium on Discourse Structure in Natural Language Understanding and Generation. Hirschberg, J., Litman, D., McCoy, K., Sidner, C. (eds), Pacific Grove, CA

W3C99 World Wide Web Consortium. 1999. Resource Description Framework (RDF) Schema Specification. Brickly, D. and Guha, R. (eds) http://www.w3.org/TR/1998/WD-rdf-schema/

W3CSWI World Wide Web Consortium Semantic Web Initiative, http://www.w3.org/2001/sw/

WEIN87 Weinberg, A. 1987. Comments on Borer and Wexler. In Roeper, T., and Williams, E. (eds.). Parameter Setting. Reidel, Dordrecht.

WEXL85 Wexler, K. and Chein, Y.C. 1985. The development of lexical anaphors and pronouns. Papers and Reports on Child Language Development. 24, 138-149.

WINO72 Winograd, T. 1972. Understanding Natural Language. Edinburgh: Edinburgh University Press.

WOOD93 Wood, M. 1993. Categorial Grammars. Routledge, London. ISBN: 0-415-04954-7

WYLL68 Wyllys, R.E., 1968. Extracting and abstracting by computer. In: Borko, H. (Ed.) Automated Language Processing. Wiley, New York.1968, pp 127-179.

XTAG The XTAG Group 1995. A Lexicalised Tree Adjoining Grammar for English. Technical Report IRCS Report 95-03, The Institute for Research in Cognitive Science, University of Pennsylvania.

YARO92 Yarowsky, D. 1992. Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In Proceedings of COLING92, Nantes, France.

YARO95 Yarowski, D. 1995. Unsupervised word-sense disambiguation rivalling supervised methods. In: Proceedings of ACL95.

YOUN67 Younger, D. 1967. Recognition and parsing of context-free languages in time n3. Information and Control, 2(10): 189-208.

ZAJI97a Zajicek M, Powell C. 1997. Enabling Visually Impaired People to Use the Internet, IEE Colloquium: Computers helping people in the service of mankind, London.

ZAJI97b Zajicek, M. and Powell, C. 1997. The use of information rich words and abridged language to orient users in the World Wide Web. IEE Colloquium: Prospects for spoken language technology, London.

ZAJI98 Zajicek, M., Powell, C. and Reeves, C. 1998. A Web Navigation Tool for the Blind. 3rd ACM/SIGAPH on Assistive Technologies, California.

ZAJI99 Zajicek M., Powell C., Reeves C. 1999. Web search and orientation with BrookesTalk. California State University Northridge, CSUN '99, Technology and Persons with Disabilities, Los Angeles.

Appendix 1: Glossary

Coherence – A principle of organisation relating to the functional connectedness or identity of elements of a text.

Construction Integration Model (CIM) – A theoretical model of discourse comprehension based on psychological evidence.

E-Language – The external realisation of a language in the form of Natural Language.

Garden Path sentence – A sentence that leads the reader to make an error when parsing its grammatical structure, thereby forcing a re-evaluation of the sentence.

I-Language – The internal realisation of a language in the form of some universal logical representation.

Incremental Interpretation – The perceived or actual process by which the interpretation of a sentence is constructed word by word in the order the words are presented.

Inheritance Model of Syntactic Categories – A model that specifies that the syntactic categories of a verb of higher transitivity inherit the structure and properties of lower transitivity versions of that verb.

Logical Form – A representation of a natural language text in which the relations between its terms are made explicit through use of predicate-argument structure.

Logical Form Transformation – The conversion of a natural language text into logical form.

Parallel Path – Two or more paths through a segment of a text, where each path supports an alternate grouping (or non grouping) of the words in that segment, thereby allowing compounds and their individual terms to be processed separately.

Parallel Shift – The process by which a compound term and its constituent terms are entered into a chart parser in parallel.

Plausibility Testing – The evaluation of logical forms with respect to a World Model.

Pre-Processor – The 'missing' component of the CIM, that is, the component that takes a natural language text and returns a representation of that text in logical form.

Sense – A notion of the meaning of a term implemented as a reference to an entry in a dictionary of senses (WordNet).

Sense Indicator String (SIS) – A string used to specify the sense of a word with respect to the WordNet lexical database. The SIS promotes like-sense detection as it removes all reference to surface form, thereby eliminating problems associated with synonymy and homonymy.

Specialisation Class (SC) - A class node in a noun or verb hyponym taxonomy at which the greatest change in Mutual Information is exhibited with respect to that of the parent node.

Surface Form – The natural language representation of a text (i.e. the text itself)

Appendix 2: Publications

- The generation of representations of word meanings from dictionaries.

  International Conference on Spoken Language Processing, Beijing. 2000.

- Similarity Based Document Keyword Extraction Using an Abridged WordNet Noun Taxonomy.

  Journal of Literary and Linguistic Computing: Special Issue on Keywords. 2005. In Press.

Publications: ICSLP2000. International Conference on Spoken Language Processing, Beijing. 2000.

The generation of representations of word meanings from dictionaries.

Chris Powell, Mary Zajicek, Prof. David Duce

Oxford Brookes University

Abstract

This paper describes the generation of *iconic* and *categorical* representations of word meaning, in propositional form, from the WordNet lexical database. These are derived from the list of synonyms, the descriptive gloss, and from the hypernym and meronym relations of each WordNet word sense. We demonstrate that these representations promote *identification* and *discrimination*, these being suggested qualities of representations of meaning, and finally suggest that these representations have further applications in language engineering

1. Introduction

Systems are available for the conversion of speech into text, such as IBM ViaVoice® and Dragon Dictate®. For text entry into a word processor this may be sufficient, but when extraction of the underlying semantics of the utterance is required by the task, in query expansion for example, further processing is necessary.

The construction-integration model proposed by Kintsch and van Dijk [1, 2, 3, 4] models cognitive processes involved in story comprehension. Experimental evidence is available to support this model, particularly for the existence of arguments, propositions and the micro and macro-structural levels of representation [5]. The general nature of the model also makes it robust and applicable in virtually all situations when compared to other systems requiring complex and specific rules [6]. However, it has not yet been fully implemented, perhaps due to its enormous grammatical and domain knowledge requirements [7].

Kintsch and van Dijk define an argument as the representation of the meaning of a word. Unfortunately, what a representation of word meaning might be is not given. We address this issue here.

2. Representations

Harnad proposes that symbolic representations, such as words, must be grounded in iconic and categorical representations [8]. He suggests that the non-symbolic sensory data we receive is processed, possibly by some connectionist approach, such that features of that data emerge and are used as the basis of these representations. Iconic representations are shown to be sufficient for discrimination (how different or alike things are) and categorical representations for identification (naming things). He also suggests that the features can be processed symbolically.

WordNet [9, 10] is an electronic lexical database and has become an important tool for the linguistics researcher. The basic unit of WordNet is the *synset*. This consists of a set of synonyms of a word sense (e.g. speech, oral communication), a descriptive gloss much like a dictionary description, and a set of semantic relations that link the synset with a number of other synsets. The two relations used here are *hypernymy* (the 'is a' relation) and *meronymy* (the 'has part' relation). Each synset also has a *sense key*, which serves as the primary key to that synset.

We propose that a WordNet synset is a suitable basis for the formation of a representation as the additional descriptive information outlined above can be viewed as analogous to the features discussed by Harnad, and which in concert define that synset. Lesk uses this property in his maximal overlap of dictionary definition terms method of word sense disambiguation [11]. For future compatibility with the construction-integration model, the representations generated consist of sets of propositions.

2.1. Categorical Representation

The gloss of *zebra* reads:

> Any of several fleet black and white striped African equines.

Which becomes in propositional form:

(OF, ANY, (SEVERAL, EQUINE))
(FLEET , EQUINE)
(STRIPED, EQUINE , BLACK , WHITE)
(AFRICAN , EQUINE)

Referring to figure 1 it can be seen that *equine* is a hypernym of *zebra* and so its representation will contain the argument EQUINE. Thus EQUINE is not a unique (categorical) feature of *zebra* and is replaced by the null argument ∅. No further words from the synset of *equine* conflict with *zebra*, and so the categorical representation of *zebra* becomes:

(OF, ANY, (SEVERAL, ∅))
(FLEET , ∅)
(STRIPED, ∅ , BLACK , WHITE)
(AFRICAN , ∅)

This representation now contains only the information necessary to distinguish a zebra from a general equine.

2.2. Iconic Representation

The *iconic* representation contains features that allow discrimination between entities. As any feature may be called upon as a criterion for discrimination, the iconic representations should comprise all features. These can be formed from the conjunction of categorical representations along a hypernym chain. For example:

Iconic(zebra) = Categorical(zebra) +
Categorical(equine) + … + Categorical(entity)

3. Generation of word-sense representations

The procedure for generating word sense representations consists of 4 steps to be followed in sequence. The procedure was implemented in C++, and was used to generate the representations used in the evaluation. The functions described in the evaluation were also implemented in C++.

3.1. Step 1: Segmenting WordNet Glosses

A WordNet gloss consists of a general dictionary-style definition, optional example sentences, and occasional bracketed or quoted embedded sentences. For example, the gloss for one sense of the word *speech* reads:

> (Communication by word of mouth; "His speech was slurred"; "he uttered harsh language"; "he recorded the language of the streets")

Obviously, the raw form of the gloss is unsuitable for presentation to a part-of-speech (POS) tagger, and the gloss is segmented, breaking it into its constituent sentences and extracting any embedded sentences:

> Communication by word of mouth.
> His speech was slurred.
> He uttered harsh language.
> He recorded the language of the streets.

Each gloss segment is now in a suitable form for POS tagging.

3.2. Step 2: Tagging WordNet Glosses

The POS tagger used is probablistic and was constructed from word frequency/part-of-speech and part-of-speech/part-of-speech bigrams derived from the British National Corpus. The Viterbi algorithm [12] is employed to determine the most probable path through the lattice of possibilities. Although generally not as good as, say, the Brill tagger [13], considering the relatively simplistic forms of the gloss sentences, and our requirement to identify only basic syntactic categories rather than the finer-grained categories of the CLAWS5 tagset used by the BNC, it is entirely acceptable. To date, no incorrect taggings have been detected.

3.3. Step 3: Compound Noun Detection

Nouns are replaced by their WordNet lemmas after identifying them from noun phrases obtained from the tagged gloss segments. Each noun phrase is presented to WordNet, and if recognised, is replaced by its WordNet lemma. If not, it repeatedly undergoes head decomposition until it is recognised. In this way both simple and compound nouns are recognised.

3.4. Step 4: Proposition Extraction

Propositions from the gloss. The procedure described by Kintsch [14] is used to extract propositions from each gloss segment. The propositions formed consist of a *predicate* derived from verbs, adjectives, adverbs and sentence connectives, and the *arguments*, representing items such as agent, subject and goal (replaced by ∅ as necessary). Predicates and arguments are shown in uppercase to distinguish them from words. Currently, our proposition generator is limited in that fragments such as *black and white striped equine* are represented by the proposition set:

{(STRIPED, ∅) , (BLACK , ∅) , (WHITE , ∅)}

and not the more accurate:

(STRIPED , ∅, BLACK , WHITE)

So far this is not a problem, but will need to be rectified before the representations are used in conjunction with the construction-integration model.

Participles and Gerunds. The use of participles (verbal adjectives) and gerunds (verbal nouns) is common in English. For example, *striped horse* is equivalent to *horse with stripes*. Thus it is desirable to capture the equivalent noun form of any participle or gerund used in the gloss to assist argument matching. As these often contain their noun-form in their gloss (e.g. wheeled = having *wheels*, containing = include or *contain*), this is possible: The participle or gerund is stemmed using the Porter stemming algorithm [15], and is compared to the stemmed words from the gloss. If a match is found, a new proposition is added to the relation, predicated by EQUIV, e.g. (EQUIV, WHEELED, WHEEL).

Synonyms, Hypernyms and Meronyms. Additional propositions are generated to reflect the synonyms and WordNet relations of hypernymy and meronymy, predicated by SYN, IS_A, and HAS_PART respectively:

(SYN , RUBBER , PENCIL_ERASER , RUBBER_ERASER)
(IS_A , ZEBRA , EQUINE)
(HAS_PART , COAT , SLEEVE)

4. Word-Sense Representation

The representation of a word consists of the set of propositions generated from its synset as described above. This can be formally stated as follows:

Let π be a proposition generated from a synset, and ρ a representation of a synset, then:

$$\rho = \{\pi\}$$

Now let ρ_s be the instance of a representation generated for synset *s*. The representation of *zebra* generated from its gloss is as follows (note, *zebra* has no synonyms):

ρ_{zebra} = {(SYN , ZEBRA) ,
(IS_A , ZEBRA , EQUINE) , (FLEET , EQUINE) ,
(BLACK , EQUINE) , (WHITE , EQUINE) ,
(STRIPED , EQUINE) , (EQUIV , STRIPED , STRIPE) ,

(AFRICAN , EQUINE) ,
(EQUIV, AFRICAN , AFRICAN) }

The function H (hypernym) can now be defined as:

$$H(\rho_s) = \{\rho_x \mid \rho_s \text{ IS\_A } \rho_x\}$$

The function M (meronym) is similarly defined.

Applying the hypernym function to the representation of *zebra* yields the representation of *equine*:

$$H(\rho_{zebra}) = \{\rho_{equine}\}$$

The function E (entire) generates the entire representation of a synset, which consists of the union of all representations on the hypernym chain of that synset, i.e. the reflexive transitive closure of H over ρ_s:

$$E(\rho_s) = \rho_s \cup \{n : \mathbb{N} \mid n \geq 1 \bullet H^n(\rho_s)\}$$

Thus the entire representation or *zebra* is:

$$E(\rho_{zebra}) = \{\rho_{zebra} , \rho_{equine} , \rho_{mammal} , \rho_{animal} , \rho_{entity} \}$$

5. Evaluation

As discussed earlier, the properties of *identification* and *discrimination* have been proposed as desirable qualities of meaning representations. Thus the evaluation attempts to discover these properties in the representations generated by the procedure above. Representations for the animal words *Horse*, *Zebra*, *Donkey*, *Panda*, *Parrot* and for the marking words *Stripe* and *Dapple*, were generated.

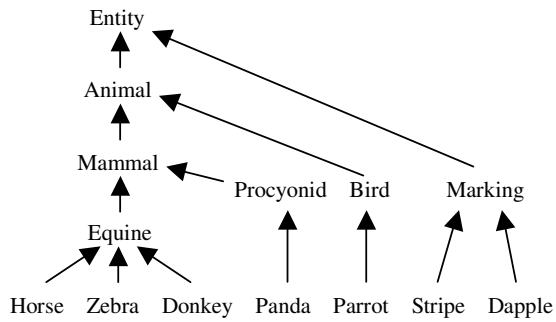


Figure 1: Hypernym relationships of WordNet evaluation synsets (condensed for brevity).

5.1. Identification

Harnad uses the example that a Zebra can be identified from the expression: Horse + Stripes. If identification is possible, then combining these representations should lead to the selection of the representation of *zebra* from all representations.
Firstly, we define the L function (like) to return all representations containing the immediate hypernym of the given representation, i.e. its siblings and their descendants. The hypernym itself is not returned:

$$L(\rho_s) = \text{ran}(H^{-1}(\rho_s \lhd H))$$

This essentially selects all categorical representations that are like the given representation, i.e. have the same hypernymic subsumer. Thus, given the representation of the synset of *horse*, function L produces the following:

$$L(\rho_{horse}) = \{\rho_{horse} , \rho_{zebra} , \rho_{donkey} \}$$

The function Y (identify) produces a set of representations for which the function's arguments are satisfied:

$$Y(\rho_a , \rho_b) = \{\rho_x \mid \exists \pi_x \in \rho_x , \pi_b \in \rho_b \bullet \rho_x \in L(\rho_a) \Lambda C(\pi_x , \pi_b)\}$$

The function C tests for referential coherence, which is implemented by Kintsch and van Dijk [1] as argument overlap between propositions. Thus proposition (X, Y, Z) is referentially coherent with (F, Z, T) as they share argument Z. Null arguments are ignored.

When supplied with the representations of *horse* and *stripe*, the Y function firstly applies the L function to *horse* as shown above. Referential coherence is sought between the resulting set and ρ_{stripe}, and is found only when $\rho_x = \rho_{zebra}$, where π_x = (EQUIV, STRIPED, **STRIPE**) and π_b = (SYN, **STRIPE**, STREAK), both from categorical representations, giving:

$$Y(\rho_{horse} , \rho_{stripe}) = \{\rho_{zebra} \}$$

Thus *Zebra* has been identified from the representations of *horse* and *stripes*.

5.2. Discrimination

Given a representation drawn from the entire set of items under consideration, the discrimination function (Δ) should accept or reject that representation on the basis of a second representation, the discrimination criterion:

$$\Delta(\rho, \rho) : \text{boolean}$$

If the representations are sufficient for discrimination, then they should allow, for example, the partitioning of the set ANIMALS into feathered and non-feathered types.

Positive discrimination is attained when the iconic form of a representation contains the feature *feather*. For example, when ρ_{parrot}, which contains the proposition (CHARACTERISED, FEATHER) from the synset of *bird*, is tested against $\rho_{feather}$, which contains the proposition (SYN, FEATHER, PLUME, PLUMAGE):

$$\Delta (\rho_{parrot} , \rho_{feather}) = \text{true}$$

Negative discrimination is attained when the iconic form of a representation does not contain the feature FEATHER, e.g.:

$$\Delta (\rho_{zebra} , \rho_{feather}) = \text{false}$$

Formally:

$$\Delta(\rho_a, \rho_b) = (\exists \rho_x \in E(\rho_a) \mid \exists \pi_x \in \rho_x, \pi_b \in \rho_b \bullet C(\pi_x, \pi_b))$$

6. Applications

One application of this form of meaning representation that has been investigated is anaphoric resolution. Consider the two sentences:

> Before inserting the cassette into the vcr, make sure **it** is plugged-in/blank.

Using the representations of the words *cassette, vcr, plugged-in,* and *blank*, the anaphor **it** can be resolved by the Δ function in each case:

$$\Delta(\rho_{vcr}, \rho_{plugged\_in}) = \text{true}$$
$$\Delta(\rho_{cassette}, \rho_{blank}) = \text{true}$$

which are respectively satisfied by:

C((recording , **TV**), (connect , **TV**)).

C((HOLDS, **CONTAINER** , MAGNETIC_TAPE), (EQUIV, CONTAINING, **CONTAINER**))

The other two combinations are not coherent:

$$\Delta(\rho_{vcr}, \rho_{blank}) = \text{false}$$
$$\Delta(\rho_{cassette}, \rho_{plugged\_in}) = \text{false}$$

Note also that the adjective *blank* has three senses in WordNet, but only one sense satisfies the referential coherence function, suggesting applications in word sense tagging.

7. Conclusion

Using a model system we have demonstrated that it is possible to construct representations of word meaning from WordNet synsets which promote the properties of *identification* and *discrimination*. It has also been demonstrated that these representations have applications in language engineering, namely anaphoric resolution and word sense tagging. However, a larger scale evaluation is required to confirm the robustness of the representations and the defined functions, and to determine any requirement for further functionality.

The work presented here concentrates on noun representations. Other syntactic classes will require slightly different representations, for the incorporation of transitivity (or otherwise) and selectional preference of verbs for example.

An extension to this work will use WordNet sense keys as propositional arguments. This will allow inheritance of representations indexed on sense key, that is the import of additional related features, enabling the representations to be extended in a constrained manner.

7. References

1. Kintsch, W & van Dijk, T.A. Towards a model of text comprehension and production. Psychological Review 85, 363-394. 1978.
2. Kintsch, W. The role of knowledge in discourse comprehension: A construction-integration model. Psychological Review, 95, 163-182. 1988.
3. Kintsch, W. A cognitive model for comprehension. In H.L.Pick, P. van den Broek, & D.C. Knill (Eds.), Cognition: Conceptual and methodological issues. Washington, DC: American Psychological Association. 1992.
4. Kintsch, W. The psychology of discourse processing. In M.A. Gernsbacher (Ed.), Handbook of psycholinguistics. London: Academic Press. 1994.
5. McKoon, G. & Ratcliff, R. Priming in item recognition: The organisation of propositions in memory for text. Journal of Verbal Learning and Verbal Behaviour, 19, 369-386. 1980.
6. Kintsch. W., Welsch, D., Schmalhofer, F. & Zimny, S. Sentence memory: A theoretical analysis. Journal of Memory and Language, 29, 133-159. 1990.
7. Hahn, U. & Mani, I. Tutorial T6: Automatic Text Summarisation. ECAI '98. 1998.
8. Harnad, S. The Symbol Grounding Problem. In: S. Harnad (Ed.) Categorical perception: The groundwork of cognition. New York: Cambridge University Press. 1990.
9. Miller, G. A. (Ed). WordNet: An on-line lexical database. Special issue of International Journal of Lexicography, 3(4). 1990.
10. Miller, G. A. WordNet: A lexical database for English. Communications of the ACM, 38(11), 39-41. 1995.
11. Lesk, M. *Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone.* Proceedings ACM SIGDOC Conference, 1986, 24-26. Toronto, Canada.
12. Viterbi, A.J. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory. IT-13:1 260-269. 1967.
13. Brill, E. A simple rule-based part of speech tagger. In: Proceedings of the Third Conference on Applied Natural Language Processing, ACL, Trento, Italy. 1992.
14. Kintsch, W & van Dijk, T.A. The representation of meaning in memory. Hillsdale, N.J.: Erlbaum. 1974.
15. Porter, M.F., "An Algorithm For Suffix Stripping," Program14 (3), July 1980, pp. 130-137

Publications: Journal of Literary and Linguistic Computing: Special Issue on Keywords. 2005. In Press.

Similarity Based Document Keyword Extraction Using an Abridged WordNet Noun Taxonomy.

Chris Powell
Oxford Brookes University.

Abstract

We present a novel method of abridging the WordNet noun and verb taxonomies through selection of classes corresponding to *Specialization Classes* (SCs) by examining change in information content along hyponym chains. The abridged taxonomies are evaluated through their ability to disjointly partition polysemous senses of all WordNet noun and verb lemmas; the proposed abridgement method is shown to have high precision (95% for nouns, 83% for verbs) at 100% recall, superior to baseline and random abridgements.

Key Classes are identified by selecting those SCs that occur more frequently than in a reference document collection. Experiments collecting noun Key Classes from SemCor documents, each tagged with all possible noun senses, and with a single sense selected by the First Sense heuristic, are compared with those obtained using the SemCor -annotated senses. Over 80% accuracy is found only for single senses, demonstrating the increased frequency of the 'correct' SCs is not sufficient to rank them higher than erroneously selected SCs, and hence the procedure cannot select the correct sense of keywords automatically.

Example documents from the SemCor collection show that the Key Classes group lexically dissimilar but semantically similar terms, thereby defining the sense of the Key Class lexically, unlike the traditional Term Frequency approach where each Key Term is represented by a single lexical item that may be sense-ambiguous in isolation.

1 Introduction

1.1 Motivation

We have been investigating the application of natural language processing techniques as an assistive technology for blind and visually impaired (BVI) people, particularly with respect to web page navigation and content determination. BVI people generally use refreshable Braille displays or synthesised speech to render the text of a web page. These approaches make the content accessible to the user, but the serial presentation limits their ability to determine rapidly the *relevancy* of the web page to their line of enquiry, and requires that a reasonable amount of the page be rendered in

speech/Braille in order to present sufficient information for topic inference to occur. This contrasts with the technique employed by the sighted user who rapidly and randomly visually scans the page, quickly forming a judgement as to the content and hence relevancy of the document.

BrookesTalk, a web browser for BVI people, has been developed at Oxford Brookes University (Zajicek *et al.,* 1997; Zajicek *et al.,* 1998); its main innovation being the presentation to the user of keywords automatically extracted from the currently loaded page. The intuition behind this is that the user's own cognitive abilities will identify the context in which the keywords are most strongly related - this context will then, in a general sense, inform them of the topics covered by the page. This is particularly useful when selecting potentially relevant pages, from those returned by a search-engine. For example, given the query 'RED DWARF', a number of pages are returned and keywords generated for each of them. Keywords such as STAR, PULSAR and TELESCOPE suggest an astronomical theme, whereas KRYTEN, LISTER and SCUTTER are indicative of the eponymous BBC TV series.

Currently, BrookesTalk uses Luhn's (Luhn, 1958) Term Frequency (TF) method to generate the keywords: words from the web page are stopword-filtered, stemmed, and the stems ranked on decreasing frequency. The top 10 stems, after being mapped back to their original form, are then presented as the keywords. TF is attractive as it is easily implemented, requires no resources other than a stemmer and a stopword list, and importantly for user satisfaction, operates in real-time.

1.2 Defining the Senses of Keywords

The efficacy in suggesting page topics hinges on the user's ability to identify the context in which particular meanings of the extracted terms make sense. Humans are of course very good at this kind of task. Nevertheless, we are concerned that by relying solely on high frequency tokens drawn from a text, additional contextual information provided by low frequency near-synonyms of those high frequency tokens is being discarded. For example, suppose the word DISC is extracted as a keyword. What kind of disc is it? If one of the other keywords is COMPUTER we can deduce FLOPPY DISC or HARD DISC, whereas coocurrence with MUSIC would suggest a PHONOGRAPH RECORDING. Now consider the situation where a keyword is presented together with its near synonyms that have also been extracted from the document: DISC might be presented with FLOPPY and HARD in the first instance, and with LP in the second, making the sense in each case clear. Presenting sense-related words in this way thus suggests the sense of the near synonyms before the other extracted keywords (i.e. COMPUTER and MUSIC) are taken into consideration.

In order to implement such a system, a definition of *near-synonym* is required. Obviously, synonyms of extracted keywords form a subset. As seen above, FLOPPY DISC and HARD DISC are not synonyms, but they are *similar* in that they are both direct *hyponyms* of MAGNETIC DISK. Therefore, we equate near-synonymy with similarity, and propose to extract words of similar meaning under each keyword.

1.3 Overview

We hypothesis then that by collecting groups of keywords on the basis of *similarity,* rather than purely on frequency of occurrence, will more strongly suggest senses for those keywords, and the consequent reduction in ambiguity will simplify the task of identifying a context into which they fit. The remainder of this article describes the process of selecting keywords on similarity.

Section 2 presents a novel method of abridging the WordNet noun and verb hypernym taxonomies, using change in information content to identify Specialization Classes – essentially, points at which cuts in the taxonomy are to be made – thereby reducing the size of the taxonomies to 6.8% (noun) and 13.4% (verb) of their original size. All procedures are formally defined using the Z notation.

Section 3 demonstrates that the abridged noun and verb taxonomies are capable of accurately discriminating between senses of polysemous lemmas – a recall/precision evaluation revealing a precision of 95% (noun) and 83% (verb) at a recall of 100%, showing that few sense distinctions are lost by the abridgement process.

Section 4 proposes that *Key Classes* may be used to replace the *Key Lemmas* selected by *Term Frequency* methods. Recognising classes subsumed by a Specialization Class as similar, it compares the Key Classes selected when no sense disambiguation occurs, and when rudimentary sense disambiguation is attempted, against those selected from fully sense disambiguated document nouns drawn from SemCor (Landes, 1998). Results indicate that better results are obtained when sense disambiguation is attempted (over 80% accuracy for the 10 highest ranked Key Classes). Examples are given demonstrating the augmentation of lexically dissimilar but semantically similar document terms extracted under each Specialization Class. The increase in frequency due to this augmentation however is not sufficient to rank the desired senses of polysemous lemmas more highly than erroneous senses. Again, all procedures are formally defined using the Z notation. Conclusions are presented in Section 5.

2 Similarity

Various methods of assessing *word similarity* have been proposed: Using a taxonomy such as WordNet (Miller 1995), two nodes are similar if they share a hypernymically related node. The degree of similarity may be determined by counting edges (Rada *et al.,* 1989; Leacock *et al.,* 1998). *Semantic Similarity* (Resnik, 1995) measures similarity as *information content* of the common subsumer, obtained from taxonomy node probabilities assigned through corpus frequency analysis. This approach as been augmented by factoring-in path length (Jiang, 1997), itself similar to the *Similarity Theorem* based *Lin Measure* (Lin, 1997). Relations other than hypernym/hyponym have been used, employing defined sequences of directed relational types (Hirst *et al.,* 1998). Tree-Cut Models (TCM) employing Minimum Description Length (Quinlan *et al.,* 1989) have been used to partition noun taxonomies on similarity of case-frame slot fillers (Li *et al.,* 1995a; Li *et al.,* 1996). As an alternative to these approaches, Lesk proposes *dictionary definition overlap* (Lesk, 1989), where increasing definition-word overlap indicates greater similarity.

The similarity metrics above, with the exception of the Tree-Cut Model, all produce a measure of how similar two senses are (or will state that they are not similar). So, given CAR and LORRY, these metrics will report that they are very similar, and share the hypernym MOTOR VEHICLE. CAR and SKATEBOARD are less similar, but similar nonetheless, and share the hypernym ARTEFACT. However, by the same token, CAR and PENCIL are also similar, again sharing the hypernym ARTEFACT. To avoid this unacceptable result, a similarity threshold would be required; those cases where the similarity value was found to be above the threshold accepted as similar, and those below rejected. This presents yet another problem in that a suitable threshold must be selected. The Tree-Cut Model on the other hand partitions the hypernym/hyponym taxonomy, thereby collecting similar senses under each cut. Using this scheme it is possible to give a yes/no answer to the question 'are these senses similar?'. However, the proposed TCM is designed to identify senses that are similar with respect to their roles in case frames, requiring consideration of their coocurrence probabilities with some predicate. Nevertheless, having a preselected set of cuts, and hence groups of similar senses, is attractive considering the real-time application we have in mind.

2.1 A method for Predefining Groups of Similar Senses

A method of defining sets of similar senses presents itself if one considers Resnik's procedure for calculating the *information content* (IC) of nodes in the WordNet noun hypernym taxonomy (Resnik, 1995; Resnik 1998). Recall that in the construction of the probabilistic model of WordNet, the frequency of any class *c* is calculated recursively as the number of

occurrences of that class plus the sum of the frequencies of its hyponyms, shown in equation 1 below.

$$\text{freq}(c) = \sum_{w \in words(c)} \frac{1}{|\, classes(w) \,|} \text{freq}(w) \qquad (1)$$

where *words*(c) is the set of words in any synset subsumed by *c*, and where *classes(w)* is the set $\{c \mid w \in words(c)\}$ (Resnik, 1998)

Two factors are involved in the calculation of the IC value of a WordNet class: Firstly, the raw frequency of occurrence of terms, as derived from corpus analysis, is assigned to appropriate classes. This results in the more frequently occurring classes having a higher frequency score than less occurring classes, as illustrated by node α in Fig. 1a. In some way, this echoes Luhn's observation that term frequency and term significance are related (Luhn, 1958). Secondly, the frequency scores are cumulatively propagated along the hypernym relation, resulting in the summed class frequency being additionally influenced by its hyponyms, as shown by node β in Fig. 1b, which is reminiscent of a *spreading-activation network*.

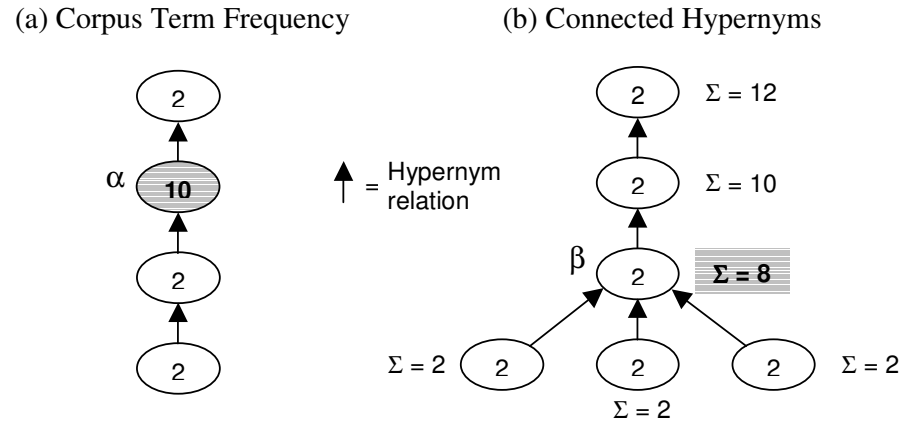


Fig. 1. The derived frequency of a class depends upon both term frequency (a), and on number of hyponyms (b). Individual term frequencies are shown within nodes, sum indicates cumulative class frequency.

In the two examples above, it can be said that the labelled nodes form *abstract classes*; node α is similar to a term frequency based *keyword* within its hypernym chain, and node β is highly activated by its subordinate nodes. Observe in each case, there is a large change in value (frequency and

summed frequency respectively) between the labelled node and its immediate hyponym(s). This effect is shown clearly in Table 1, the cumulative frequency data for the hypernym chain of DOG (canine). Word frequency data is derived from the 100 million word British National Corpus (BNC) and applied to the noun taxonomy of WordNet 1.6.

Class	ΣFreq
ENTITY	963909
ORGANISM	385594
ANIMAL	38913
CHORDATE	21502
VERTEBRATE	21496
MAMMAL	13657
PLACENTAL	13391
CARNIVORE	2803
CANINE	1203
DOG	995

Table 1 Cumulative frequencies of hypernyms of DOG (canine)

Note that in Table 1 the summed frequencies do not change smoothly; there are particularly large changes when moving from ANIMAL to ORGANISM (Δ=346681), and from ORGANISM to ENTITY (Δ=578315). These are caused by the summation of frequencies of *all* subordinates of ORGANISM (including ANIMAL), and of *all* subordinates of ENTITY (including ORGANISM) respectively, of which there are many. From this we deduce that ORGANISM and ENTITY strongly abstract the hypernyms of dog. However, in an ideal situation we would prefer just the right level of abstraction, not strong abstraction - clearly ORGANISM does not discriminate between DOG and CAT, or even PLANT and ANIMAL. Worse still, ENTITY cannot discriminate between such as DOG and BICYCLE.

Following Resnik (Resnik 1998), the information content value *I* for each class *c* was calculated using equation 3, after first deriving the class probabilities *p(c)* from the cumulative frequencies via equation 2.

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

$$\text{where } N = \sum_{c'} freq(c') \text{ for } c' \text{ ranging over all classes} \qquad (2)$$

$$I_c = -\log p(c) \qquad (3)$$

Table 2 shows that, as expected, the classes near the top of the taxonomy express relatively little information (column **IC**). Calculating the change

(increase) in information (column **ΔIC**) reveals the greatest change takes place in the move from ORGANISM to ANIMAL.

Class	ΣFreq	Prob	IC	ΔIC
ENTITY	*963909*	*0.03962*	*1.40212*	
ORGANISM	*385594*	*0.01585*	*1.80003*	*0.39790*
ANIMAL	38913	0.00160	2.79606	**0.99603**
CHORDATE	21502	0.00088	3.05367	0.25761
VERTEBRATE	21496	0.00088	3.05380	0.00013
MAMMAL	13657	0.00056	3.25081	0.19701
PLACENTAL	13391	0.00055	3.25933	0.00852
CARNIVORE	2803	0.00012	3.93848	0.67915
CANINE	1203	0.00005	4.30596	0.36748
DOG	995	0.00004	4.38817	0.08221

Table 2 Class-based probability and information values for the hypernym chain of *dog*

If ENTITY and ORGANISM are strong abstractions of DOG, then it can be said that the classes ANIMAL..DOG are *specialisations* of the strong abstractions, Further, as the move from ORGANISM to ANIMAL presents the greatest change in IC, then the greatest specialisation happens at ANIMAL. We have chosen to designate the node that incurs the greatest positive change in IC a *Specialization Class* (SC). Thus ANIMAL is the SC of those classes within the DOG hypernym chain. Intuitively, ANIMAL does seem to present a *plausible abstraction* of DOG, and it certainly discriminates between DOG and BICYCLE. By applying cuts to the WordNet noun hypernym taxonomy at the SCs we can construct an abridged WordNet noun hypernym taxonomy; the nodes of the taxonomy will be the SCs, and each SC will 'contain' all subordinate similar senses.

An SC can be formally defined as follows:

Given: (4)

[CLASS] the set of WordNet noun classes.

c: CLASS c is of type CLASS

I: c → REAL Function I returns the information content of class c.

The hypernym function H can be defined as: (5)

H: CLASS ↔ CLASS

$H(c) = c_h \mid c$ IS_A c_h

Note that:

(6)

$H^n(c)$ represents the reflexive transitive closure of H over c, and

H^0 represents the identity, that is, $H^0(c) = c$

The Specialization Class selection function SC can now be defined: (7)

$$SC: CLASS \rightarrow\!\!\!\!\!\!\rightarrow CLASS$$

$$SC(c) = H^n(c) \text{ where}$$
$$\exists\, n : \mathbb{N} \mid n \geq 0 \bullet MAX(I(H^n(c)) - I(H^{n+1}(c)))$$

2.2 Identifying the Specialization Classes

Using the BNC as the reference source, the information content of each WordNet noun class was calculated as per equations 1 to 3 above. The specialization class selection function *SC*, defined in equation 7, was then applied to identify the subset of WordNet noun classes that constitute the Specialization Classes, as shown in equation 8. Initial examination of the results showed that for many nouns, the immediate hypernym of a root class was selected as the SC - an unsatisfactory result precipitated by the fact that these classes are the focus of many subordinate classes. To counteract this, the roots and their immediate hypernyms were disallowed as candidates for selection. SUBSTANCE, a third-level class, was also found to have a very high change in information content, leading to its preferential selection, and so was similarly disallowed. This resulted in 145 noun *base classes* being disallowed.

[CLASS]	The set of WordNet noun classes
SCLASS: ℙCLASS	The set of noun Specialization Classes

$$SCLASS = \{\forall\, c: CLASS \bullet SC(c)\} \quad (8)$$

The verb taxonomy was similarly processed, with the exception that as no bias was found towards the top of the taxonomy, possibly due to the shallow, bushy nature of the verb taxonomies, there was no need to disallow any verb classes from the selection process. However, as the selection mechanism can never select the root of a taxonomy, 618 verb base classes were nevertheless ignored. We will return to this issue in Section 2.3**.**

2.2 Abridging Hypernym Chains

It is interesting to note that a class *c* selected as the SC of a noun sense *s* is not necessarily selected as the SC of all hyponyms of *s* Take for example the classes DOG and HAMSTER. As has been seen above, ANIMAL is the SC of DOG, and is also a hypernym of HAMSTER. However, the SC of HAMSTER is RODENT. This observation permits an abridged representation of HAMSTER to be constructed by selecting only the identified SCs, as shown in Fig. 2.

HAMSTER: **RODENT** → PLACENTAL → MAMMAL → VERTEBRATE →
CHORDATE → **ANIMAL** → LIFE_FORM → ENTITY

HAMSTER: **RODENT** → **ANIMAL**

Fig. 2 Abridged hypernym representation of HAMSTER using SCs

Complex taxonomic structures, such as that for BEER, see Fig. 3, are easily accommodated by traversing each hypernym path from leaf to root separately. Table 3 gives the change in information content values for the three paths associated with BEER, and shows that BEVERAGE, FLUID and DRUG are directly selected as SCs of BEER.

Path A			Path B			Path C		
Class	**Info**	**ΔInfo**	**Class**	**Info**	**ΔInfo**	**Class**	**Info**	**ΔInfo**
ENTITY	1.40212		ENTITY	1.40212		ENTITY	1.40212	
OBJECT	1.59641	0.19429	OBJECT	1.59641	0.19429	OBJECT	1.59641	0.19429
SUBSTANCE	2.30612	0.70971	SUBSTANCE	2.30612	0.70971	ARTEFACT	1.83769	0.24128
FOOD	2.76016	0.45404	**FLUID**	**3.45593**	**1.14981**	**DRUG**	**3.22856**	**1.39088**
			LIQUID	3.47550	0.01957	D ABUSE	3.59402	0.36545
BEVERAGE	**3.64836**	**0.88820**	BEVERAGE	3.64836	0.17286			
ALCOHOL	3.78927	0.14092	ALCOHOL	3.78927	0.14092	ALCOHOL	3.78927	0.19526
BREW	4.57581	0.78654	BREW	4.57581	0.78654	BREW	4.57581	0.78654
BEER	4.87778	0.30197	BEER	4.87778	0.30197	BEER	4.87778	0.30197

Table 3 Change in information content for hypernyms of BEER

Processing the entire noun taxonomy in this way selects 4373 of the available 66025 WordNet noun classes. Similarly, 931 of the 12127 verb classes were selected.

2.3 A Fully Abridged Taxonomy

Recall that the *base classes* disallowed by the selection process do not appear in the set of extracted SCs. Nevertheless, they may be encountered in texts, and are required in order to reconstruct (in abridged form) the original

noun and verb taxonomies. For these reasons the base classes were added to the set of SCs, resulting in a total of 4518 noun and 1625 verb classes in the abridged WordNet noun and verb taxonomies. This corresponds to an abridged noun taxonomy 6.8% of the size of the original, and an abridged verb taxonomy 13.4% the size of the original.

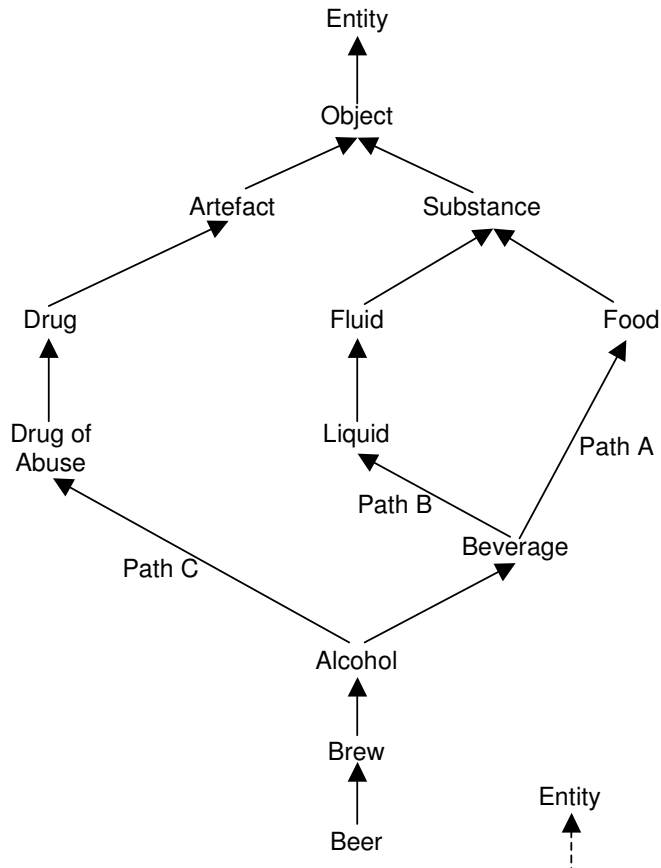


Fig. 3. WordNet Hypernym representation of BEER

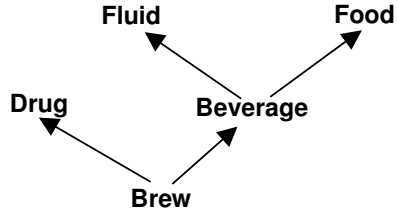


Fig. 4a Abridged representation of Beer

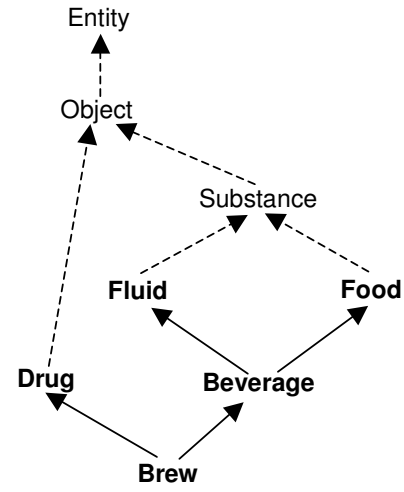


Fig 4b Full abridged representation of Beer

The abridged representation of BEER, constructed only of SCs, is shown without base classes in Fig. 4a, and with base classes in Fig. 4b. Note that BREW and DRUG are selected as SCs by processing other senses not shown here.

2.4 Discussion

Tables 4 and 5 show the distribution of noun and verb classes within their respective specialization classes. **Size** indicates the number of classes subsumed by an SC, and **Freq** the number of occurrences of an SC containing **Size** classes.

Table 4 shows that 54 noun SCs do not subsume any other class, and consequently only synonymous lemmas can be grouped by these SCs. This is also true of the 271 instances of single-class verb SCs. The most frequent number of classes subsumed by a noun or verb SC is 2, corresponding to the SC class and one hyponym of that class. Although SCs containing few senses are frequent, the tables show that some SCs subsume a high number of senses – the highest containing 1920 nouns and 830 verbs. Examination of the data revealed that the PLANT (*flora*) SC held the most noun senses, closely followed by ANIMAL (*animate being*), FAMILY (*taxonomic*) and COMPOUND (*chemical*). For verbs, CHANGE (*transform*) was the most populous SC, followed by CHANGE (*undergo change*), MOVE (*locomote*) and MOVE (*displace*). Considering the large numbers of animals, plants, and taxonomic classifications, together with ways to or be changed or moved, contained within a dictionary such as WordNet, it is entirely predictable that highly subsuming SCs will exist. However, as the senses within an SC are not distinguished from each other in any way other than by identity, no subdivisions within an SC exist. This results in no distinction being made between PARROT and DOG for example - both map on to the SC ANIMAL. This may be problematic if SCs are to be used as the basis for selectional association calculations; where it would only be possible to state FLY(ANIMAL), and not FLY(BIRD) for example.

A solution to the above problem, should one be necessary, would be to add all populous SCs to the base classes during SC extraction; as these are disallowed by the selection process, the class scoring the next highest change in information would be selected in its stead. So in the case of DOG (Table 2), ANIMAL would be disallowed, and CARNIVORE would be selected, and as a consequence the SC for ANIMAL would no longer contain any carnivores directly. The process could be repeated until all SCs contained less than a predefined number of senses. On completion, base classes are combined with the selected classes to form the abridged taxonomy, and so ANIMAL, and all other senses shunted into the set of base classes, would again become available as an SC, albeit with fewer subsumed senses.

Size	Freq	Size	Freq	Size	Freq	Size	Freq	Size	Freq	Size	Freq
1	54	31	9	61	3	96	3	146	1	255	1
2	1022	32	17	62	6	97	3	148	2	261	1
3	663	33	15	63	2	99	1	150	1	276	1
4	460	34	11	64	2	100	1	152	1	286	1
5	322	35	11	66	3	101	1	153	1	288	1
6	231	36	5	67	2	103	1	155	1	299	1
7	183	37	11	68	4	105	1	160	2	300	1
8	168	38	5	69	4	106	2	162	1	303	1
9	144	39	8	70	2	107	1	169	1	306	1
10	104	40	5	71	1	110	2	170	1	308	1
11	105	41	10	72	4	111	2	178	1	313	1
12	82	42	8	73	1	112	1	179	2	322	1
13	79	43	6	74	1	115	1	183	2	324	1
14	65	44	5	75	3	116	1	190	1	333	1
15	56	45	7	76	3	118	1	191	1	334	1
16	47	46	5	78	4	120	1	193	1	364	1
17	43	47	10	79	2	122	2	198	1	367	1
18	30	48	6	80	2	127	1	199	1	370	1
19	38	49	3	81	1	129	1	202	3	385	1
20	30	50	10	82	2	130	2	204	3	401	1
21	34	51	4	83	2	133	1	206	1	423	1
22	23	52	3	84	2	134	1	207	2	524	1
23	20	53	4	85	3	135	2	208	1	558	1
24	16	54	6	87	1	136	2	215	1	607	1
25	18	55	10	88	1	138	1	218	1	774	1
26	14	56	1	89	1	140	2	227	1	860	1
27	18	57	6	91	2	141	1	229	1	1070	1
28	23	58	4	92	1	143	2	239	1	1824	1
29	18	59	2	93	1	144	1	242	1	1920	1
30	19	60	3	94	3	129	1	245	1		

Table 4 Number of noun classes subsumed by noun SCs

Size	Freq	Size	Freq	Size	Freq	Size	Freq
1	271	17	15	33	3	71	1
2	425	18	10	35	4	74	1
3	234	19	12	36	2	75	1
4	130	20	4	38	1	80	2
5	107	21	4	40	1	87	1
6	93	22	12	41	2	91	1
7	51	23	7	43	1	143	1
8	48	24	1	45	1	150	1
9	30	25	2	49	1	152	1
10	19	26	7	50	1	154	1
11	22	27	2	53	1	187	1
12	21	28	4	55	1	236	1
13	12	29	5	58	1	295	2
14	14	30	3	61	1	376	1
15	7	31	1	64	1	830	1
16	9	32	3	65	1		

Table 5 Number of verb classes subsumed by verb SCs

3 Evaluation of SC Sense Distinctions

To determine the degree to which sense distinctions have been preserved in the abridged noun and verb hypernym taxonomies, a precision/recall experiment was devised to evaluate the ability of SCs to disjointly partition the senses of polysemic lemmas: by recognising that the function *SC* simply maps a given class on to itself or one of its hypernyms it can be seen that, ideally, the *n* senses of a polysemic lemma should map on to *n* SCs. The senses of that lemma may thus be considered *query terms*, and the mapped SCs the *target set*. Recall will always be 100% as the SCs will always be hypernyms of the query terms (or the query terms themselves), whereas Precision may be reduced if two or more query terms map on to the same SC. Precision is therefore calculated as follows:

Let Λ be a lemma, $\sigma(\Lambda)$ a function returning the set of senses of Λ, and $\chi(\Lambda)$ a function returning the set of SCs for all senses of Λ.

$$\text{Precision} = \frac{\#\sigma(\Lambda)}{\#\chi(\Lambda)} \tag{9}$$

3.1 Evaluation datasets

To evaluate the ability of SCs to discriminate between senses of a lemma, all 94474 noun (10319 verb) lemmas from the WordNet NOUN.IDX (VERB.IDX) tables were processed. Along with the set of 4518 noun (1625 verb) SCs extracted by the above method, for comparative purposes two additional sets of SCs were generated: (a) a baseline containing only the 145 noun (618 verb) base classes, and (b) a randomly selected set of 3907 noun (1605 verb) SCs (including the base classes).

Precision was calculated for each lemma obtained from the noun (verb) index according to equation 9 and recorded in an array indexed on $\#\sigma(\Lambda)$. For completeness, monosemous lemma occurrences were recorded and, as only one sense is available to its SC, assigned a precision of 100%.

3.2 Results

The precision values for the three evaluations, for both nouns and verbs, are presented in Table 6. Column $\mathbf{\#\sigma(\Lambda)}$ indicates the number of senses obtained for a lemma, **Count** the number of lemmas contained in each of the above groups, and **Bases**, **Rnd+Bases**, and **SC+Bases** the precision of the three abridgement sets.

In calculating the average precision, monosemous lemmas ($\#\sigma(\Lambda) = 1$) were ignored, as were those values of $\#\sigma(\Lambda)$ for which no data was seen (Count = 0)**,** resulting in 21 noun and 39 verb precision values. Of the 4518

noun (1625 verb) SCs, 432 (28) corresponded to monosemous lemmas, the remaining 4086 (1597) to polysemous lemmas.

The relatively low number of noun bases presents a coarse-grained abridgement, which is reflected in its low precision (0.5381) in the polysemous lemma discrimination task. The random selection, covering more classes lower in the taxonomy, provides a better precision (0.7574), but the best precision is obtained using the extracted SCs (0.9464). The situation is similar for the verb discrimination task, the extracted SCs producing the highest precision (0.8328). On three occasions the random verb precision equalled the SC verb precision ($\#\sigma(\Lambda)$ = 14, 17, 48) and on one occasion bested it ($\#\sigma(\Lambda)$ = 30). No SC noun precision was equalled or beaten by a random noun precision.

It is interesting to note that although the precision is not 100%, it does not necessarily follow that the SC selection procedure is somehow flawed. Take for example the lemma MUSIC, for which WordNet 1.6 lists 6 senses. Of these, senses MUSIC#2 and MUSIC#5 are described as '*any agreeable (pleasing and harmonious) sounds*' and '*the sounds produced by singers and musical instruments*' respectively. Both of these senses map on to the SC PERCEPTION#3. Further, these two senses of MUSIC share the immediate hypernym SOUND#2 (auditory sensation). It is therefore not surprising, and should be expected, that a certain number of one-to-many mappings between SCs and senses will be encountered. Considering the fact that one-to-many mappings occur when the senses of a lemma share a hypernym, something that becomes more likely towards the root of a taxonomy, it is perhaps more surprising that so few occur.

4 Selecting Keywords on Similarity

A typical TF keyword extractor selects the most frequently occurring terms as the keywords, either by comparison with the frequencies of other terms in that document, or with the frequencies of those document terms when compared with their frequencies in a reference corpus. We have selected the latter, again using the BNC as the reference corpus.

4.1 Noun Key Lemmas

Before examining the keyword expansion properties of SCs, it will be useful to see the *Key Lemma* output of a keyword generator based on noun lemmas. This requires a noun lemma reference corpus and a set of documents for keyword extraction.

The noun reference corpus was prepared by extracting all nouns from the BNC, lemmatizing them using the WordNet morphological normalizer, and counting the frequencies of each lemma form, resulting in (lemma, frequency) pairs.

	Noun Precision				Verb Precision			
Size		145	3907	4518		618	1605	1625
#σ(Λ)	**Count**	**Bases**	**Rnd + Bases**	**SC + Bases**	**Count**	**Bases**	**Rnd + Bases**	**SC + Bases**
1	81910	1.000	1.0000	1.0000	5752	1.0000	1.0000	1.0000
2	8345	0.7901	0.8991	0.9434	2199	0.9038	0.9293	0.9611
3	2225	0.7112	0.8661	0.9426	979	0.8488	0.8931	0.9302
4	873	0.6804	0.8411	0.9444	502	0.8237	0.8675	0.9268
5	451	0.6718	0.8483	0.9512	318	0.7679	0.8277	0.8931
6	259	0.6274	0.8346	0.9556	188	0.7660	0.8333	0.9069
7	140	0.5898	0.8102	0.9541	102	0.7507	0.8305	0.8978
8	82	0.5762	0.7835	0.9482	75	0.7333	0.7767	0.8867
9	68	0.5376	0.7598	0.9493	39	0.7009	0.7664	0.8803
10	42	0.5476	0.7429	0.9286	39	0.7359	0.7897	0.8769
11	23	0.5415	0.7312	0.9605	32	0.7358	0.8097	0.8835
12	18	0.5602	0.7500	0.9861	15	0.7444	0.8056	0.8833
13	9	0.5470	0.7436	0.9487	16	0.6827	0.7692	0.8606
14	8	0.4643	0.6696	0.9464	5	0.7571	0.8429	0.8429
15	7	0.4952	0.7333	0.9333	8	0.7667	0.7917	0.9167
16	3	0.4583	0.7083	0.9167	8	0.6641	0.7422	0.8359
17	6	0.4412	0.6275	0.9216	4	0.6324	0.7647	0.7647
18	1	0.5556	0.8333	1.0000	4	0.6528	0.7222	0.8333
19	1	0.4737	0.8421	0.8947	2	0.5789	0.7632	0.8158
20	0				2	0.5000	0.6250	0.7000
21	0				3	0.8095	0.8413	0.9048
22	0				3	0.5758	0.6212	0.8182
23	0				1	0.4783	0.5652	0.6087
24	1	0.2500	0.4583	0.9167	3	0.6528	0.7083	0.7222
25	0				2	0.6800	0.7800	0.9000
26	0				3	0.5769	0.7308	0.7821
27	0				1	0.5185	0.6296	0.6667
28	0				1	0.7500	0.7857	0.8571
29	1	0.4138	0.6897	0.9655	1	0.6552	0.6897	0.8966
30	1	0.3667	0.7333	0.9667	1	0.7333	0.8333	0.8000
32	0				1	0.5313	0.6875	0.7500
33	0				1	0.5455	0.7273	0.8182
36	0				1	0.6944	0.7778	0.8889
37	0				1	0.6757	0.7838	0.8378
38	0				1	0.6316	0.7105	0.7632
41	0				2	0.6341	0.7195	0.8293
42	0				1	0.5476	0.6667	0.8095
45	0				1	0.5556	0.6667	0.9333
48	0				1	0.5625	0.6667	0.6667
63	0				1	0.4921	0.6349	0.7302
Total	94474				10319			
	21 polysemous lemma groups				39 polysemous lemma groups			
Average		**0.5381**	**0.7574**	**0.9464**		**0.6671**	**0.7533**	**0.8328**

Table 6 Precision of sense distinctions of three abridgements for both nouns and verbs

The evaluation documents selected are the Brown 1 section of the SemCor corpus, which was chosen as the WordNet sense tags it incorporates will

allow keyword senses to be evaluated alongside keyword surface form. A document is processed by calculating the frequency of each lemma identified as a noun by its POS-tag, again resulting in (lemma, frequency) pairs. The frequencies of the lemmas in both sets are then normalised to sum to one, restricting the normalisation of the reference set to those with lemmas occurring in the document set, as shown in equations 10 to 13 below. Note that when processing SemCor documents, those terms mapped onto PERSON, LOCATION, and GROUP were excluded, and the first sense of those terms identified as 'difficult' by the SemCor taggers (and therefore having two or more sense tags) was taken.

Given:

[LEMMA]	the set of lemmas
docSum: $\mathbb{N}$	sum of document frequencies
refSum: $\mathbb{N}$	sum of corresponding reference frequencies
doc: LEMMA $\pfun$ REAL	document lemma to frequency
ref: LEMMA $\pfun$ REAL	reference corpus lemma to frequency

Then:

$$\text{docSum} = \Sigma\ \text{ran}(\text{doc}) \tag{10}$$

$$\text{refSum} = \Sigma\ \text{ran}(\text{dom}(\text{doc}) \lhd \text{ref}) \tag{11}$$

$$\text{doc} \oplus \{\forall\ l: \text{LEMMA} \mid l \in \text{dom}(\text{doc}) \bullet l \mapsto \text{doc}(l) \div \text{docSum}\} \tag{12}$$

$$\text{ref} \oplus \{\forall\ l: \text{LEMMA} \mid l \in \text{dom}(\text{doc}) \bullet l \mapsto \text{ref}(l) \div \text{refSum}\} \tag{13}$$

Key Lemma selection is made by calculating the difference in normalised frequencies between the document and reference sets, as shown in equation 14. Ranking on decreasing difference then places at the top of the list those lemmas that occur more frequently than predicted by the reference corpus.

diff: LEMMA $\pfun$ REAL	Lemma to difference in normalized frequency relation

$$\text{RANK}(\text{diff} \oplus \{ \forall \text{ l: LEMMA} \mid \text{l} \in \text{dom doc} \bullet \text{l} \mapsto \text{doc(l)} - \text{ref(l)} \}) \quad (14)$$

The top ten lemmas from three SemCor documents, BR-A01, BR-D03, and BR-N05 selected by this method are presented in Table 7, frequencies shown bracketed.

Rank	BR-A01	BR-D03	BR-N05
1	jury(20)	england(20)	wilson(18)
2	election(12)	catholic(16)	girl(9)
3	resolution(9)	church(18)	fire(7)
4	bonds(8)	priest(7)	half-breed(4)
5	fund(8)	clergyman(6)	wrist(4)
6	funds(8)	unity(6)	cheek(4)
7	legislator(6)	non-catholic(5)	grass(4)
8	georgia(6)	protestant(5)	scar(3)
9	petition(6)	article(6)	dish(3)
10	county(7)	archbishop(5)	horse(4)

Table 7 Key Lemmas from three SemCor documents

The first two documents provide reasonably indicative keywords as they are drawn from categories that cover single domains (Press Reportage and Religion respectively). The third document is drawn from the General Fiction category, which of course is *story* based, and consequently the extracted keywords have nothing to be indicative of.

4.2 Specialization Class Keyword Groups

Here we follow the same basic procedure as for Key Lemma selection, but replace the lemma with the specialization class of the lemma in all procedures, thereby generating *Key Classes*. The consequence of this is that similarly sensed lemmas (that is, those sharing the same specialization class) will be grouped together regardless of their lexical form. See Tables 8 to 10 below for examples of this. The procedure is formally defined by equations 10 to 14 above (including the 'givens'), but with LEMMA replaced by SPECIALIZATION CLASS. Additionally, the reference corpus, which now must comprise (SC, frequency) tuples, is constructed by assigning all appropriate WordNet senses to each noun and verb lemma, as the BNC is not sense-tagged. This introduces noise, but again we rely on that noise being randomly distributed throughout the corpus to minimize its effect.

A further problem with the move from lemma to class is that appropriate senses must be assigned to each document lemma, a non-trivial task. We therefore have the choice of either pre-processing a document using some sense-tagging procedure to obtain one sense per lemma, or

attempt no sense-tagging at all and assign all appropriate senses to each lemma. Note however that, as each lemma is now augmented by similarly sensed but low frequency terms drawn from the document, the frequency of each SC is the sum of its component terms. For example, the lemma *election*, with a frequency of 12, is included in the SC VOTE along with primary(3), reelection(1), general_election(1), primary_election(1) as shown in Table 8, giving a total frequency of 18. If all senses of a lemma are taken, then other erroneous SCs will be present and will also gain frequency counts from their low frequency associates. This poses the question of whether the increase in frequency of actual associations of intended senses can outweigh the increase in frequency of chance associations between erroneous senses. If this proves to be the case, the intended senses of SCs will rank more highly and consequently there will be no need to sense-disambiguate a document before extracting key senses.

4.3 Comparison of Key Class Selections

To evaluate the possibility of automatic selection of correctly sensed Key Classes, a comparison was made between noun Key Classes selected from the Brown 1 SemCor document set nouns (excluding those mentioned in 4.1 above) tagged with all noun senses, and with senses assigned using the *First Sense* heuristic in which the first listed WordNet sense (i.e. the most frequently encountered) of any polysemic lemma is assigned. This heuristic produces reasonable accuracy – (Li *et al.,* 1995b) report 57% accuracy, whereas (Gonzalo *et al.,* 2001) report that 70% of the noun senses in SemCor correspond to the first WordNet sense, but suggest that the higher result may be due to the annotator selecting of the first matching sense rather than the best matching sense, thereby preferring those senses at the top of the sense list.

The SCs selected by equation 7 for the All Sense and First Sense taggings are compared with a baseline comprising the SCs selected (again using equation 7) when using the senses supplied as part of the SemCor annotation. The comparison was performed on the ranked difference between the All Sense (or First Sense) SCs and the baseline, using the SPECIALISATION CLASS version of equation 14. Precision/Recall is again used to determine the degree of correspondence between the All/First sense sets and the baseline, and is performed over a variable number of top-ranked SCs to simulate the selection of the n top SCs as *Key Classes*. The first n items of the ranked test sets are compared with the first n items of the baseline set, where $1 \leq n \leq 30$. As Recall = Precision, the results are reported as 'F-Score' values. On average, 590 SCs were returned from each of the 102 documents in the SemCor Brown 1 collection when inspecting all senses of the document lemmas, whereas on average 190 were extracted

when considering only the first WordNet sense. The results for the first 30 items are presented in figure 4.

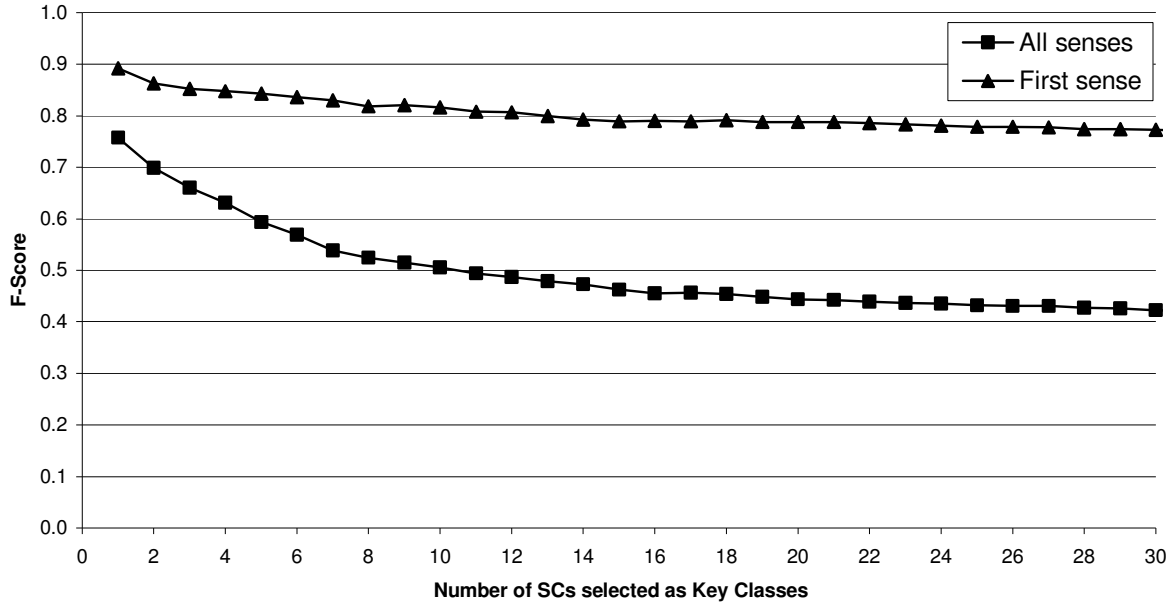


Fig 4 F-Score of top ranked *n* SCs from first and all WordNet senses with respect to top ranked SemCor annotated SCs

4.4 Discussion

The results show that 89% of the noun SCs ranked first in the First Sense set are found in the base data, and that a recall/precision of over 80% may be obtained by selecting up to the top 13 ranks. Even in the All Sense set, the first ranked SC occurs over 75% of the time in the base data, but rapidly drops - the top eleven ranks scoring below 50%. Evidently, with the set of SCs used, the First Sense heuristic set performs better overall than the All Sense set, and the theory that correct SC senses would gravitate to the top of the ranked list does not hold. It therefore appears that some sense disambiguation is necessary in order to take advantage of the similarity grouping provided by SCs.

Tables 8 to 10 present the top 10 SC ranks, using the First Sense heuristic, of the documents BR-A01, BR-D03, and BR-N05 respectively. Rows marked with an asterisk indicate erroneous SCs not represented in the original document, caused by the heuristic selecting the wrong sense. Emboldened lemmas indicate Key Lemmas that also occur in the baseline data, and figures in brackets indicate lemma frequency.

Rank	SC Sense	Lemma
1	money	**fund**(8), money(2), revolving_fund(1)
2	vote	**election**(12), primary(3), reelection(1), general_election(1), primary_election(1)
3	funds;finances; monetary_resource; cash_in_hand; pecuniary_resource	**funds**(8)
4	body	**jury**(15), grand_jury(3),
5	calendar_day;civil_day	monday(5), friday(4), sunday(2), tuesday(2), saturday(1), wednesday(1)
*6	chemical_bond;bond	**bond**(8)
7	law;jurisprudence	law(5), laws(2), enabling_legislation(1)
8	contest;competition	campaign(5), race(3)
9	document;written_document ;papers	**resolution**(9), ballot(1)
10	lawgiver;lawmaker	**legislator**(6), congressman(1)

Table 8 Top 10 SCs of document BR-A01

Rank	SC Sense	Lemma
1	religionist;religious_person	**catholic**(16), **non-catholic**(5), roman_catholic(2), nun(1), christian(1), tractarian(1)
2	religion;faith	**church**(18), catholic_church(2), church_of_rome(1), religious_order(1)
3	england	**england**(20)
*4	old_age;years;age	year(8); years(8)
5	year;twelvemonth;yr	year(8); years(8)
6	religion;faith;religious_belief	catholicism(4), faith(4), high_Anglicanism(1), roman_catholicism(1)
7	leader	**priest**(7), **clergyman**(6), minister(2), cleric(1), parson(1), shepherd(1), vicar(1)
8	protestant	**protestant**(5), anglican(1), nonconformist(1)
9	denomination	church_of_England(4), anglican_church(2)
10	integrity;unity;wholeness	**unity**(6)

Table 9 Top 10 SCs of document BR-D03

Rank	SC Sense	Lemma
1	situation;state_of_affairs	thing(7); things(5)
*2	property;belongings;holding; material_possession	thing(5); things(5)
3	female;female_person	**girl**(9); woman(4)
3	whip	quirt(7)
4	fire	**fire**(7)
5	joint;articulation	**wrist**(4), knee(2)
6	real_property;real_estate;realty	acre(2); acres(2); land(2)
7	leather_strip	reins(2); rein(2); thong(1)
8	organ	eye(4); eyes(4)
9	topographic_point;place;spot	place(7)
10	belief	eye(4); eyes(4)

Table 10 Top 10 SCs of document BR-N05

Comparing the top ten lemmas (Key Lemmas) in Table 7 with the top ten SCs lemmas (*Key Class Lemmas*) in Tables 8 and 9 shows that for documents BR-A01 and BR-D03, the highest ranked 7 (8 resp.) Key Lemmas are to be found in the Key Class Lemmas - shown emboldened in the tables - and are the most frequent of each Key Class Lemma set. The tables also show that the Key Lemmas have been augmented with low frequency but related lemmas. For example, the Key Lemma *election* is included in the Key Class *VOTE* – where *VOTE* is a hypernym (@) of *election*, where *primary_election* and *reelection* are hyponyms (~), *general_election* is a sibling, and *primary* and *primary_election* are synonyms, as shown in Fig. 5. This is true for other lemmas, for example *fund* is augmented with *revolving_fund*, *resolution* with *ballot*, *church* with *catholic_church, church_of_rome* and *religious_order*.

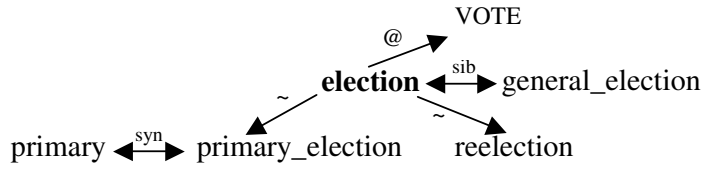


Fig 5 Relationships of lemmas associated with the Key Lemma **election** under the SC VOTE.

Only three of the Key Lemmas of document BR-N05 are present in the Key Class Lemmas of Table 10, although they do occur in the top five ranks. Again, some degree of augmentation is present in that *girl* is augmented with *woman*, and *wrist* with *knee*.

Some lemmas appear in more than one Key Class Lemma group, for example, *funds, years, things,* and *eyes*. Multiple entries such as these occur where a document term maps on to two or more lemmas in WordNet, occurring in this case because the term is both a lemma in its own right *and* the plural form of another lemma, e.g. the term *funds* has both the lemmas *funds* and *fund.*

12 5 Conclusion

We have presented a novel method of abridging the WordNet noun and verb taxonomies, which utilises change in class information content to identify those classes at which major specializations occur. Although few in number when compared to the total number of classes in the taxonomy, these specialization classes have been shown to discriminate between polysemous lemmas when their senses are not closely related by hypernymy. It has also been proposed that scaleable abridgements may be produced by repeated application of the abridgement algorithm.

As the abridgement algorithm effectively produces cuts in a taxonomy at specialization classes, the classes under each cut may be viewed as similar, and therefore allows words of similar meaning to be grouped together. By applying standard keyword selection techniques to specialization classes rather than lexically similar terms, each key specialization class contains a number of lexically dissimilar terms which, when read together, point to the sense of the specialization class. Additionally, these terms do not necessarily occur frequently in their source documents. This is different from lexical keywords, which are disambiguated by their mutual context, and always occur frequently in their source documents.

For a polysemous lemma, the addition of low frequency terms similar to the usually selected high frequency terms for the desired sense does not increase the overall frequency of its specialization class sufficiently to increase its ranking it above the alternative and erroneous specialization classes of that lemma, and hence specialization classes cannot be used to automatically select correctly sensed Key Classes from non sense-tagged documents. However, it has been shown that keywords sense-tagged to an accuracy of over 80% may be obtained by pre-assigning senses to nouns using the First Sense heuristic.

The use of abridged taxonomies allows keywords to be expanded into groups of similarly sensed words drawn from a document. However, as the keyword selection procedure is essentially frequency based, as used in lexical keyword selection procedures, the efficacy of the of the procedure to extract useful keywords is dependent on the document type, better results being obtained from single-subject documents. In the case of story-based or

multi-topic documents, identifying discourse segments through *linear text segmentation* (Choi, 2000) may prove fruitful.

Specialization Classes have two obvious applications that will be investigated in the future. Firstly, they may prove useful as a mechanism for query expansion; query terms, mapped to a specialization class, can be expanded into a group of similarly sensed but lexically different terms, thereby increasing the scope of each query term. Secondly, the accuracy to which specialization classes partition the senses of polysemous lemmas, along with the high degree of abridgement they afford, suggests that specialization classes may be used as surrogates for noun and verb senses, effectively reducing the search space of any sense-discrimination procedure. Our work in progress is using specialization classes in this fashion, using them to describe selectional association profiles, which in turn are involved in incremental parsing and decision making during processing of predominantly right-branching sentences with a Combinatory Categorical Grammar.

6 Acknowledgements

I would like to thank Marilyn Deegan for inviting me to 'The Use of Computational Linguistics in the Extraction of Keyword Information from Digital Library Content' workshop, Kings College London, Feb. 2004, and Dawn Archer and Tony McEnery of Lancaster University for their useful and encouraging comments.

7 References

BNC http://info.ox.ac.uk/bnc URL accessed 1 May 2004.

Choi, F. (2000). Advances in Domain Independent Linear Text Segmentation. *Proceedings of NAACL'00.* Seattle, USA.

Gonzalo, J., Fernandez-Amoros, D., and Verdejo, F. (2001). The Role of Conceptual Relations in Word Sense Disambiguation. *Proceedings of the 6th International Workshop on Applications of Natural Language for Information Systems.* (NLDB'01).

Hirst, G. and St-Onge, D. (1998). Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms. In: Fellbaum, C. (ed), *WordNet: An Electronic Lexical Database.* 305-332. MIT Press, 1998.

Jiang, J. and Conrath, D. (1997). Semantic Similarity based on Corpus Statistics and Lexical Taxonomy. *Proceedings of the International Conference on Research in Computational Linguistics,* Taiwan, 1997.

Landes, S., Leacock, C., Tengi, R. (1998). Building Semantic Concordances. "WordNet: An Electronic Lexical Database" , MIT Press, Cambridge, MA.

Leacock, C. and Chodorow, M. (1998). Combining Local Context and WordNet Similarity for Word Sense Identification. In: Fellbaum, C. (ed), *WordNet: An Electronic Lexical Database.* 265-283. MIT Press, 1998.

Lesk, M. (1998). Automatic Sense Disambiguation using Machine Readable Dictionaries: How to tell a Pine Cone from an Ice Cream Cone. *Proceedings of SIGDOC '86*, 1986.

Lin, D. (1997). Using Syntactic Dependancy as a Local Context to Resolve Word Sense Ambiguity. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, 64-71, Madrid, 1997.

Li, H. and Naoki, A. (1995a). Generalising Case Frames using a Thesaurus and the MDL Principle. *Proceedings of Recent Advances in Natural Language Processing*, 239-248.

Li, H. and Naoki, A. (1996). Learning Dependancies between Case Frame Slots. *Proceedings of the Sixteenth International Conference on Computational Linguistics*, 10-15.

Li, X., Szpakowics, S. and Matwin, S. (1995b). A WordNet-based Algorithm for Disambiguation. *Proceedings of IJCAI-95*. Montreal, Canada.

Luhn, H. P. (1958). The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, **2**, 159-165.

Miller,G.A. (1995). WordNet: A Lexical Database. *Communication of the ACM*. **38**(11): 39-41.

Quinlan, J. R. and Rivest, R. L. (1989). Inferring Decision Trees using the Minimum Description Length principle. *Information and Computation,* **80**:227-248.

Rada, R., Mili, H., Bicknell, E., and Blettner, M. (1989). Development and Applications of a Metric on Semantic Nets. *IEEE Transactions on Systems, Man and Cybernetics*, **19**(1):17-30, 1989.

Resnik, P. (1995). Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence,* Montreal, August 1995.

Resnik, P. (1998). WordNet and Class-Based Probabilities. In: Fellbaum, C. (ed) WordNet, An Electronic Lexical Database. MIT Press.

Zajicek, M. and Powell, C. (1997). The use of information rich words and abridged language to orient users in the World Wide Web. IEE Colloquium: *Prospects for spoken language technology*, London.

Zajicek, M., Powell, C. and Reeves, C. (1998). A Web Navigation Tool for the Blind. *3rd ACM/SIGAPH on Assistive Technologies*, California.

Publications: Journal of Literary and Linguistic Computing: Special Issue on Keywords. 2005. In Press.