A structural alignment model of noun-noun compound interpretation

by

Jer Hayes

School of Computer Applications,

Dublin City University,

Glasnevin,

Dublin 9.

Supervisor: Prof. Josef Van Genabith

A dissertation submitted for the degree of Master of Science

February 2003

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Master of Science is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my own work.

Signed: Ger Hayes ID Number: 97970603

Date: 05/02/03

Permission to lend and/or Copy

D.C.U. may lend or copy this thesis on request.

Signed: Fer Hayes ID Number: 93970603

Date: 05/02/03

Dedicated to Faye for always being there...

Abstract

The interpretation of noun-noun compounds is complex, yet compounds such as 'web surfer' and 'beef baron' are generated and interpreted easily by native English speakers. Concept combination is the core process in the generation and interpretation of noun-noun compounds. Such compounds may be read literally or metaphorically suggesting that the combination process is capable of both literal and metaphoric interpretations.

The motivation for this thesis is to tackle three problems which occur in concept combination. These problems are: (1) compounds are often polysemous, (2) compounds often appear to be understood by evoking a context (or world knowledge) and (3) compounds can be interpreted figuratively. We suggest that adopting structural alignment allows us to deal with each of these problems.

Structural alignment is a process whereby conceptual structures are placed into correspondence and similarities are found. The structural alignment model proposed in this thesis suggests that there are six core combination types and that an interpretation of a noun-noun compound will fall into one of these combination types. Some of these combination types are figurative and some rely on finding a context.

We provide an implementation of the model, the INCA system. The INCA system is a program where a user can find interpretations for noun-noun compounds. INCA has a knowledge base and attempts to find fixed patterns in a network representation of concepts. Depending on the type of pattern found, several types of interpretation can be generated.

The performance of INCA is compared with that of a number of human subjects in a brief evaluation study. The study shows that combination types proposed by our structural alignment model to offer a good coverage of the interpretations that people generate. Finally we set out proposals for developing INCA further and outline directions for future research.

Acknowledgements

This section is the part that everyone reads and generally, only reads. This is unfortunate, as all the effort has gone into the rest of this short tome.

There are a number of people I have to thank. I would especially like to thank Josef Van Genabith and Tony Veale. My experience with Josef and Tony has taught me a lot. I have found them both to be in no short measure equally exact and exacting. I have learned more about research and writing for an academic audience in the last six months than I had in the previous number of years. I have also had the fortunate experience of having a German (but with a Dutch name?) correcting my appalling English. I guess at one level its all Germanic. It is Josef's input which has led to this thesis actually being here in existence. In the final stages of this thesis Josef put in a huge effort to push me past the last post. Without Josef's corrections to (many) earlier drafts and the very slow improvements I made there would be no thesis. Whether you think that is a good thing or not I will leave up to you, gentle reader. Tony Veale is the person who set this train in motion, and I will be ever grateful to him for that but I'm more than glad that we have finally reached the station.

A quick thanks must go to John Kelleher and Hyowon Lee who have helped me by giving good advice at times when it was especially needed. I could also not have completed this work without help and support from many people who've passed through DCU over the years: Owen, Rachel, Mark, Eamon, Ronan, Jerh, Fran, Quinn, Tony, Aidan, Paul, Keiran, Q, Steve, Cathal and of course last but not least Tom.

Finally, most important of all, I would like to express my eternal gratitude to my family. Special thanks to Stella, Jerry, Deirdre and Dara for putting up with me and of course Faye, for really putting up with me.

Contents

Typographical Conventions

This thesis uses a number of terms that are not distinguishable from each other when displayed in plain text. As a result the following typography is used:

CONCEPTS - concepts and categories are displayed as italicised small caps

word - the mention of a term or a word is displayed underlined

"Quotations" - are placed within double quote marks and italicised. Quotations will include those from written sources and all that which may occur in spoken language.

'compounds' - compounds are placed within single apostrophe marks.

properties, features, & attributes - are italicised.

>>meanings<< - are placed within the following brackets: >>, <<.

List Of Tables

List Of Figures

Chapter 1 - Understanding Noun-Noun Compounds

1.1 Introduction

New words enter our language everyday. Sometimes these words exist within a certain context for the briefest amount of time; sometimes these words enter the common parlance and last for centuries. So what, if people generate new words? To ask this question is to ignore the subtle beauty of human thought. Somehow people can create new words and somehow other people can readily understand them. The juxtaposition of existing words in new combinations can create a novel unit termed a compound. These compounds are different from other grammatical combinations of words in that, to a first approximation, they can be manipulated and reused as a single unit. This is how combinations such as 'arms race' and 'web surfer' have entered our language. The search for how people combine words to make novel combinations and how these are interpreted is the motivation for the work that follows.

This chapter will introduce the object of study in this thesis: the nominal compound and its interpretation in terms of concept combination and structural alignment. The interpretation of compounds can be complex, and a number of theories have been proposed. From these, two major problems can be drawn: (1) polysemy – the notion that compounds can have more than one interpretation; and (2) world knowledge – the background of facts and expectations that serve as a foundation for situated thinking in the real world. In this thesis we consider a further problem, that of figurative or metaphoric interpretations, which stretch the boundaries of conventional word meaning. The problem of literal combination is already so challenging that current theories of concept combination generally do not account for figurative interpretations. However, we argue that figurality underlies the whole enterprise of concept combination, and consequently the model developed in this thesis will primarily deal with this problem. A research methodology that borrows from Cognitive Psychology and Artificial Intelligence is proposed. This borrowing is used as it accounts for metaphor and analogy, exemplars of figurative language. Using the same mechanism that accounts for metaphor we will offer an account of concept combination that deals with both literal and figurative combinations.

1.2 Research Methodology

Combinations such as 'arms race' and 'web surfer' are made up of two different words but appear to form a linguistic construct that, to a first approximation, functions like a single word. They often refer to a single concept or entity. Linguistically such constructs are known as compounds (O'Grady, Dobrovolsky & Aronoff, 1993). There are various ways in which such combinations can be studied.

1.2.1 Analysing Combinations

The way in which combinations like 'arms race' and 'web surfer' are formed and interpreted falls under the broader study of language. The study of language is of interest to various academic disciplines, including, but not limited to, Psychology, Philosophy, Computer Science and Linguistics (Aitchison, 1997). However, in each of these academic disciplines only certain aspects of language are investigated, and the very same linguistic phenomena can be viewed in different ways in different disciplines.

One sub-area of Psychology, called Psycholinguistics, looks specifically at language. It has been defined as *"... the study of language behaviour: how real (rather than ideal) people learn and use language to communicate ideas"* (Taylor & Taylor, 1990, p3). One of its primary topics is language acquisition, the study of how children acquire language. Constructions such as the combination 'web surfer' can be analysed in terms of when and how children begin to learn combinations.

Another area of Psychology that researches language is Cognitive Psychology. Cognitive Psychology is interested in the higher thought processes, e.g. memory, attention and language (Eysenck & Keane, 1995). In looking at combinations, a cognitive psychologist may be interested in how mental representations of the component word meanings interact to give meaning to the combination as a whole.

One of the major areas of philosophical investigation is philosophy of language (e.g. see Lycan, 1999). This area is largely concerned with theories of word meaning as reference, and attempts to account for the relationship between words and the world. For example, the combination 'arms race' can be analysed in terms of its referents, and whether these referents have an actual counterpart in the real world.

The analysis of combinations in Computer Science often concerns their use in information retrieval research. Hoenkamp and de Groot (2000) argue that combinations are of more use in content description than single words, since they can carry more information than single words on their own. They also argue that a combination such as 'horse race' is a better description (or key phrase) to retrieve a text about horse racing than either of the component words on their own.

Unsurprisingly, the broadest approach to the study of language is that taken by Linguistics. Linguistic research on language is done with respect to different levels of analysis, including phonology, morphology, syntax, semantics and pragmatics (Saeed, 1995; Aitchison, 1992). Each of these levels comprises an area of sophisticated research in its own right and some are briefly outlined below.

Phonetics and phonology is the branch of linguistics concerned with sounds and sound systems in language (Aitchison, 1992). A combination like 'web surfer' can be analysed in terms of the number of sounds or phones from which it is constituted. A further area of interest may be in analysing the stress pattern that occurs when a combination like 'web surfer' is uttered.

In contrast, morphology is the study of word formation and the analysis of words into morphemes (Brown, 1980). Morphemes are the smallest, distinct, meaningful units that a word can be broken into (O'Grady et al., 1993). A combination like 'arms race' would be analysed as consisting of three morphemes: (1) arm, (2) -s, (3) race. The second morpheme in this case indicates a plural.

Syntax is the study of how words are structurally combined to create well-formed patterns such as phrases and sentences (Givon, 1979). From a syntactic perspective several questions can be asked of a combination like 'arms race'. What does a combination consist of? How does it fit into a sentence? Does it act like a noun? Note that a sentence can be well-formed in syntactic terms because it conforms to the various word-order constraints imposed by the language, yet be almost nonsensical, e.g. "colorless green ideas sleep furiously".

Semantics is concerned with analysing meaning in language (Saeed, 1995). A theory of semantics should describe how the meaning of a composite structure, like a phrase or sentence or a compound, is derivable from its the constituent parts. Clearly then, it is this aspect of word use in relation to compounds that forms the primary concern of this thesis.

Semantics, however, does not provide a complete picture of word meaning and how it contributes to the meaning of compounds. This is because we can distinguish between word meaning and world meaning. The former concerns the properties of word meanings as they relate to a linguistic theory of sentence meaning, while the latter concerns the meanings of words as they relate to the plausibility of actions and beliefs in the real world. A combination can be semantically well-formed if it leads to an interpretation that is congruent with the properties of the words involved. However, the interpretation may be highly implausible in the context of the real world. For example, a 'stone lion' is generally interpreted as >> a statue of a lion << and not as >> a lion made of stone <<.

In short, in studying language, the choice of a research methodology is of critical importance. Each academic discipline involved in the study of language employs differing research methodologies. As a result, a single aspect of language can be examined in quite different ways, and the research methodology adopted will affect the manner in which the aspect of language under discussion is treated. The interpretation of nominal compounds falls within the general remit of semantics since word meaning is involved, so a general research methodology that allows the researcher to consider aspects of semantics is required. However, to allow our model to appeal to the idea of plausibility in the real world, we shall need to look beyond semantics and invoke other areas that study meaning, e.g. Psychology, Cognitive Psychology. Ideally, we need a research methodology that draws insights from several of the academic research disciplines outlined above.

1.2.2 A Cognitive Science Based Research Methodology

The research methodology adopted in the present dissertation falls within the area of Cognitive Science. Cognitive Science is an interdisciplinary approach to the study of mind and intelligence that embraces Philosophy, Psychology, Artificial Intelligence (AI), Neuroscience, Linguistics, and Anthropology (Gardner 1985). The inclusion of AI as a main branch marks Cognitive Science as a relatively new area of research as its date of origin has been set as 1954 (Gardner, 1985).

Table 1.1 - Main branches of Cognitive Science

Main Branches of Cognitive Science
Psychology
Artificial Intelligence
Linguistics
Philosophy
Neuroscience
Anthropology

Table 1.1 shows the main branches of Cognitive Science. Typically a researcher will work predominantly in one particular branch but draw information from other branches. For example, a researcher interested in problem solving in AI might take interest in how subjects

solve problems and attempt to model this. The researcher would use a research methodology largely based on AI but would also draw on insights from Cognitive Psychology.

The methodology adopted in this dissertation is largely based on work from: (1) Cognitive Psychology, a sub area of Psychology, and (2) AI, a subarea of Computer Science. In terms of Cognitive Psychology, the area of concept combination (which is outlined in detail in Chapter 3) is applied to the problem of interpreting nominal compounds. Research in this area makes assumptions about knowledge and how it is represented in the mind. Cognitive Psychology proposes that there are basic constituents to thought; these constituents include such things as images, concepts, categories, schema and scripts (e.g. see Eysenck & Keane, 1995).

The AI elements of our research methodology involve the application of structural alignment to the problem of interpreting combinations. Structural alignment is a computational theory that has been implemented in various forms (e.g. Falkenhainer, Forbus & Gentner, 1989; Veale, 1995). This theory assumes that the representation of meaning is inherently structural, so that structure itself imparts meaning on the elements it connects together. If structure is the essence of meaning, it follows that structural operations are the essence of meaning manipulation and the means through which new meanings are constructed. The theory of structural alignment (or structure mapping) claims that the most useful operation one can perform on structured meanings is alignment, in which correspondences are derived between the elements of two meaning structures. Structural alignment is examined in more detail in Chapter 4.

The goal of this thesis is to demonstrate the centrality of structural alignment to the process of concept combination, and to construct a model of concept combination for the interpretation of nominal compounds around this theory (Chapter 5). The computational foundations of structural alignment lead to a computational implementation of this model (Chapter 6).

Cognitive Science allows us to view a problem from various academic disciplines, but this multiplicity has advantages and disadvantages. The drawback is that different researchers may use the same technical terms but hold them to mean different things. As a foundation for the work to follow, it is necessary to be explicit about the terminology used in this thesis, especially terminology related to meaning.

1.3 Accounting For Meaning

The object of study of this dissertation is the interpretation of nominal compounds. It is not easy to define meaning, and this dissertation does not attempt to offer a definitive answer to what "meaning" is. A number of assumptions about word meaning are made throughout this dissertation, however. We first distinguish between different types of meaning.

1.3.1 Sense Versus Reference

Word meaning involves both sense and reference; this distinction was first made by Frege (1984/1892). Sense concerns the logical description of the meaning and the inferences, assumptions, expectations it leads a subject to make. For instance, the sense of the word unicorn is the logical description of a horse with a horn on its head, and what that entails (mythical, magical, rare, etc.) The reference aspect of meaning concerns how a word is used to refer to the real world. For instance, the referent of man can be the set of men in the world. Not all word meanings have both sense and reference: a unicorn has a sense but no reference in this world. Indeed different words can have the same referent and different senses (Frege 1984/1892), e.g. the morning star and evening star, where the former refers to the last star to disappear in the morning and the latter is the first star to appear in the sky. Both refer to the same object in the sky, the planet Venus.

1.3.2 Extension Versus Intension

Extension and intension are more mathematical (set-theoretic) notions, but have been applied to meaning. Above we suggested that man refers to the set of men in the world, this is the extension of the word, man. In general the extension of a simple property is the class or set of individuals it refers to. In contrast, intension concerns the logical properties of meaning independent of the entities that actually instantiate that meaning. For example, *"the intension of mammal...is a definition, such as 'warm-blooded animal, vertebrate, having hair...' "* (Sowa, 1984, p 11) without reference to particular individual mammals. The extension of mammal, however, is the set of all mammals.

1.3.3 Concept Versus Category

The dichotomy between intension and extension exists for concepts and categories. Concepts are the meaning units from which complex meanings are constructed. Categories are used for categorisation, where instances of a concept (members of the category) are recognised and classified (hence the term categorisation). So to deal with concepts is to take an intensional perspective on meaning. In contrast, to deal with categories is take an extensional perspective on meaning.

Concepts and categories are structured differently. A concept is structured from attributes, features and properties, and it guides how we think and reason, e.g. how we expect instances of the concept to behave. For example, the concept *BACHELOR* contains all the expectations we have of being a bachelor – being unmarried and never having been married, being male. Instances of the category *BACHELOR* will match these expectations to varying degrees, hence the members of the *BACHELOR* category will have differing levels of membership. This imposes a radial structure on categories, where the best examples of the category occupy the focal point or hub of the category. The less typical examples inhabit the peripheral ground. For instance, a man who is unmarried but can get married is more a *BACHELOR* than a man who is, say, a priest. The structure of a category is thus defined by the structure of the corresponding concept.

Different theories of combination employ categories and concepts, depending on whether the intensional or extensional qualities of the theory are emphasised. To provide a common ground for evaluating these theories, we will sometimes use the terms concept and category interchangeably in this thesis. However, it should be clear from the context whether an intensional or extensional perspective is intended.

1.4 Compounds And Combinations

The main problem in defining compound, is that there are competing views as to what a compound may be. For example, in concept combination (Smith, Osherson, Rips & Keane, 1988) the linguistic form adjective-noun, e.g. 'red apple' is considered a compound. In Linguistics this is seen as distinct from a compound and is known as a collocation or simply as an adjective noun sequence.

The theories of concept combination discussed in Chapter 3 take a very broad definition of what a compound is. Throughout this thesis the following definition for compounding is adopted:

> Compounding *"is the combination of two or more free morphemes to form a new word"*, (O'Grady et al., 1993, p573).

This definition suggests that compounds can consist of all word types, e.g. adjectives, verbs and nouns. Compounding usually involves adjectives, nouns and verbs and no other word types (O'Grady et al., 1993). The definition above mirrors the one adopted in concept combination.

1.4.1 Noun-Noun Compounds

A nominal compound is a compound that is solely made up of nouns. It is made up of two or more elements. Only one type of compound, of all the compounds the definition above allows, is generally researched in concept combination: the noun-noun compound. In concept combination research (e.g. Costello & Keane, 1997) the term noun-noun compound refers to a compound that is made up of exactly two nouns. Throughout this thesis we assume that the term noun-noun compound refers a compound consisting of exactly two nouns.

1.4.2 Nominal Compounds And Combinations

A distinction must be made between the term combination and the term nominal compound. Language can be viewed as consisting of a syntactic level and a semantic level. The syntactic element can be seen as the form of the language and the semantic element as the meaning associated with syntactic forms. In the present dissertation the term combination refers to the interpretation of a nominal compound. This interpretation operates at the semantic and conceptual levels. The nominal compound is the syntactic form of the combination, and care must be taken not to confuse both levels and create a fundamental category error.

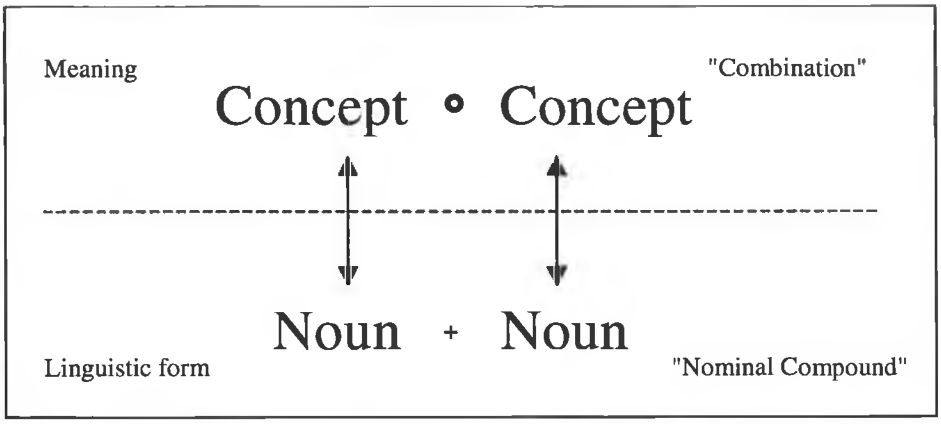


Figure 1.1 - Difference between nominal compound and combination

Figure 1.1 shows the distinction between the term combination and compound more clearly. The interaction at the semantic level is represented by the symbol, ∘ while + indicates the process of syntactic composition (e.g. concatenation, insertion of a blank space or a hyphen in English). The goal of this thesis is to develop an account of semantic and conceptual composition, ∘, in terms of concept combination and structural alignment. Figure 1.1 is a simplistic representation as the same noun structure may give rise to distinct concepts, e.g. the noun bank. When presenting examples of combinations throughout this thesis we will largely assume that a noun points to a single concept. This concept would be akin to the most prevalent sense of a word.

1.5 Interpreting Nominal Compounds

The object of study in this dissertation is the interpretation of nominal compounds. Nominal compounds are linguistic structures made up of two or more nouns. The following are nominal compounds:

> 1. 'Prize fighter', 2. 'abortion clinic', 3. 'folk rock', 4. 'police academy', 5. 'air rage', 6. 'laser therapy', 7. 'army hospital'...

Throughout this thesis, when interpretations of nominal compounds are given, these are intended to capture the meaning that most native speakers would give. These interpretations are approximations. This is in common with other areas in Cognitive Science, e.g. metaphor research, where when meanings (or readings) are given, these are presumed to be the most reasonable (e.g. Turner, 1997).

The first three nominal compounds listed above can be interpreted in order as follows:

1. >> A person who fights for money <<
2. >> A clinic that performs abortions <<
3. >> rock music which uses elements from folk music <<

The three interpretations all involve integrating the meanings of the constituent parts of the nominal compound in some way. In the first, an interpreter has to relate the concept *PRIZE* and the concept *FIGHTER* in some way. A 'prize fighter' is a type of *FIGHTER*. In 'abortion clinic' the interpreter has to relate the concepts *CLINIC* and *ABORTION*. An 'abortion clinic' is a type of clinic where abortions are performed.

1.6 Problems In Interpreting Nominal Compounds

The interpretation of nominal compounds has been examined in various ways. The area of concept combination, in particular, has pointed to a number of problems in the interpretation of noun-noun compounds. The two main problems are:

1. Polysemy
2. World Knowledge

These problems are drawn from the work of Costello and Keane (1997) and Murphy (1988) respectively.

1.6.1 Polysemy

Combinations can have more than one interpretation. The combination 'bed pencil' has been reported as having several meanings (Costello & Keane, 1997). The following interpretations are drawn from an experiment carried out by Costello and Keane (1997):

(1) >> a pencil that you put beside your bed for writing some messages <<

(2) >> a pencil used to draw patterns on bed-clothes <<

(3) >> a bed made out of pencils <<

(4) >> a pencil shaped like a bed <<

(5) >> a thin bed <<

(6) >> a big, flat pencil that is a bed for a doll <<

(7) >> a pencil case <<

The interpretations above are in no particular order, i.e. they are not ranked according to frequency or from best to worst. The range of interpretations becomes a major problem if we wish to analyse the interpretation of combinations. Although some research has been carried out into this area (e.g. Costello & Keane, 1997; Keane & Costello, 1997) the focus of much concept combination research has been on single interpretations.

1.6.2 World Knowledge

The term world knowledge is used in this thesis to refer to information that appears to fall outside of a typical representation of a concept. The importance of world knowledge in concept combination was first pointed to by Murphy (1988). For example, consider the combination 'arms race'. A simple representation of the term race might point out that it is a competition based on speed. A competition will involve a number of competitors, thus in 'horse race' the competitors are the horses (and their jockeys).

'Arms race' seems more complex than 'horse race'. It is a race between countries to develop both the most arms and the most dangerous arms. This is a *RACE* where the competitors are countries. Yet a reasonable representation of *RACE* would not include this. The information on countries does not clearly belong to either *ARMS* or *RACE*. In some way, world knowledge is used in interpreting this combination. It may be that *ARMS* acts as a pointer to a broader context which is used to interpret the overall combination. It could be that *COUNTRIES* come in through *ARMS*, e.g. *COUNTRIES own arms*. This problem is discussed further in Chapter 3.

1.6.3 A New Problem: Figurative Interpretations

Lakoff and Johnson (1980) suggest that much of everyday language involves figurative language. Nominal compounds occur frequently in everyday language. This suggests that in some cases compounds may have figurative interpretation. If this is the case, it poses a new problem for the interpretation of nominal compounds. Most of the current models of concept combination treat combinations as literal (Constraint Theory: Costello, 1996, Costello & Keane, 1998; Dual Process Theory: Wisniewski, 1997a, 1997b; Concept Specialisation: Murphy, 1988; Selective Modification: Smith et al., 1988)

Veale (1995) and Hayes & Veale (1999) point out that figurative combinations do exist. Consider the combination 'math clinic', which is typically used to describe a remedial course or centre for failing math students. Because a 'math clinic' is not literally a *CLINIC* in the established, medical sense, and the *STUDENTS* are not literally *PATIENTS* and the *TEACHERS* are not literally *DOCTORS*, this suggests that concept combination can give rise to figurative interpretations that stretch the conventional meanings of words and their implications. To date only one model attempts to give an account of figurative interpretations, that of (Gagné, 2001) which will be discussed further in Chapter 3.

Most theories of concept combination do not account for literal *and* figurative interpretations. This can be remedied by adding additional mechanisms to current theories. However, this causes further problems as it is not clear how one can draw a clear distinction between the literal and figurative. Consider the combination 'browser war' which refers to the conflict in the mid-1990s between Microsoft and Netscape. This combination seems to be figurative as a 'browser war' is not literally a type of *WAR*. The combination 'guerrilla war' seems to be literal as a 'guerrilla war' does seem to be a type of *WAR*. Now consider the combination 'gang war', a fight between street gangs. This seems to fall between the figurative and literal. On the one hand the 'gang war' is an armed conflict but it differs from *WAR* in terms of the actual conflict. There are likely to be sporadic shootings and killings but not large scale battles and it does not involve armies.

Needing to distinguish between the literal and the figurative places a large burden on any theory that does so. Presumably, as most of the main theories of concept combination treat the interpretations of compounds as being literal, these will need a mechanism that recognises deviation from literal language. The borderline between the literal and figurative is not clear-cut as the examples involving *WAR* should demonstrate.

1.7 A New Approach

Previous research with regard to some forms of cognition, e.g. metaphor (Veale, 1995) and analogy (Gentner & Markman, 1997), has suggested that structural alignment may be the underlying process in the generation of interpretations. Structural alignment or structure mapping is the process through which two conceptual structures are aligned and the constituent parts of both structures are compared. If it is the case that structural alignment is the underlying process in some forms of cognition (such as metaphor and analogy), it may be useful to apply it to concept combinations, especially in relation to figurative interpretations.

The application of structural alignment to concept combination is not new. Dual Process theory (Wisniewski, 1997a, Wisniewski, 1997b) suggests that alignment is involved. This theory is outlined in detail in Chapter 3. It should be noted that Wisniewski applies structural alignment to literal combinations and only to those where some property transfer is involved. What is new in our present research is the claim that literal and figurative interpretations of nominal compounds in terms of concept combination can be accounted for solely in terms of structural alignment. Such an account is given in Chapter 5.

1.7.1 Why Apply Structural Alignment To This Problem?

The most basic reason for attempting to apply structural alignment to concept combination is that it may be possible to account for both the literal and figurative interpretation of compounds in terms of a single process, i.e. there is no need for additional mechanisms. If concept combination involves a comparison between the component parts, then structural alignment would be a good model to apply given its pedigree in other areas. In Chapter 5, it will be shown that structural alignment can be used to account for figurative and literal combinations. Our view of structural alignment is set out in Chapter 4 and is based on the work of Veale (1995).

Furthermore, using this single mechanism we propose that world knowledge, polysemy as well as figurative combinations can be accounted for. An approach such as the one proposed is in contrast to other theories of concept combination which suggest that more than one process is involved, e.g. Concept Specialisation (Murphy, 1988). An advantage of our approach is that it conforms to Ockham's razor.[1] The model proposed in Chapter 5 will have one underlying process, structural alignment. Compared to previous theories it has a greater explanatory power as it accounts for polysemy, world knowledge and figurative interpretations (see Chapter 3 for a more in-depth discussion of previous theories).

---

[1] "Pluralitas non est ponenda sine neccesitate": which loosely translates as "entities should not be multiplied unnecessarily".

1.8 Chapter Summary

This chapter has introduced the topic that is the basis for study in this thesis: the nominal compound and its interpretation in terms of concept combination and structural alignment. A research methodology that borrows from Cognitive Psychology and AI is adopted. Broadly speaking this is a Cognitive Science approach. Taking a Cognitive Science research methodology can involve conflicting definitions. The terms compound and combination were defined and the term concept was elaborated.

The interpretation of compounds in terms of concept combination is complex. A number of theories have been proposed. From these theories two major problems can be drawn: (1) polysemy and (2) world knowledge. In this present dissertation a further problem is considered, that of figurative interpretations. Current theories of concept combination, except for Gagné (2001), do not account for figurative interpretations.

1.9 The Road Ahead

This chapter has touched on many of the chapters that follow in this thesis. In Chapter 2 we examine the basic properties of nominal compounds. In Chapter 3 the major theories of concept combination are outlined. This is done in detail in relation to one example. The problems that arise in each theory are also discussed. Structural alignment is presented in detail in Chapter 4. Our view of structural alignment is based on the Sapper model of Veale (1995). Chapter 5 develops a new theory that deals with the three problems, polysemy, world knowledge and figurative interpretations, that this chapter has discussed. The theory introduces the new idea of combination types. In Chapter 6 the design and implementation of a program, INCA (interpreting nominal compounds using alignment) is outlined. This program is the implementation of the theory presented in Chapter 5. In Chapter 7, INCA is evaluated in a study that compares the output of INCA against a group of human subjects interpreting the same combinations. In Chapter 8 a discussion is undertaken on all the topics presented in the thesis. A broader debate is also entered into how our model can integrate a number of the current models.

Chapter 2 - Basic Properties of Nominal Compounds

2.1 Introduction

This chapter examines nominal compounds. The basic properties of nominal compounds in English, duality and asymmetry, are set out. Then a corpus based study of nominal compounds in the Penn Treebank (Linguistic Data Consortium, CD-Rom, 1995 Release 2) is presented. This provides information on the frequency of compounds, the number of noun-noun compounds consisting of exactly two nominals and those that consist of more than two. This study is used as motivation for examining noun-noun compounds, the entities most widely examined by researchers in concept combination. This chapter also serves as a basis for the discussion of the theories of concept combination in Chapter 3.

2.2 Compounds in English

The properties of compounds in English can be examined in a number of distinct ways. This section first examines the orthography of compounds in English. Compounds in English tend to take one of three orthographic forms. The syntactic and semantic properties of compounds are then examined.

2.2.1 Orthography of Compounds

As was stated in Section 1.2 a compound can be made up of adjectives, nouns and verbs. The way compounds are written in English is not straightforward. There is no single way of writing them. Rather, there are three main ways in which they are written. They are either written as one word, written as two separated words or finally written together but separated by a hyphen. Table 2.1 shows some examples of this. The more accepted a compound becomes, the more likely it is to occur as two words written together, which gives the impression that it is one word (O'Grady et al., 1993).

Table 2.1 - Typographical forms of compounds

One Word: girlguide
Two words & a hyphen: girl-guide
Two words: girl guide

2.2.2 Basic Syntactic & Semantic Properties of Compounds

A compound can be broken into two parts: the head and the modifier. The modifier is usually the first element; the head is the second element. In a compound the first element is generally assumed to act by modifying the second in some way, hence the name modifier. In the compound 'house boat', house is the modifier and boat is the head. Semantically speaking, to a first approximation, the head often denotes the type of object referred to. 'house boat' is a type of boat but with some of the properties of a house, e.g. it provides shelter and is a place to live. The modifier appears to place constraints on the type of boat that 'house boat' can be. There are a plethora of interesting and challenging exceptions to this: to give but one, a 'plastic dog' e.g. is not "a dog" but "plastic shaped like a dog".

There are also syntactic and morphological features that mark the second element in a noun-noun compound as the head. The head is the point of inflection. In English a nominal head can be inflected to capture number (singular or plural) or the possessive case. The compound 'assistant lecturer' has two elements, the head, lecturer and the modifier, assistant. Consider the case where utterances involve the possessive case. The first is acceptable in English, the second is not:

1. "It's the assistant lecturer's turn"
2. ? "It's the assistant's lecturer turn"[1]

In utterance (2) it appears that the noun-noun compound 'assistant lecturer' has been broken up such that "lecturer turn" is possessed in some way by "assistant" which makes the utterance difficult to interpret.

According to O'Grady et al. (1993) the two main properties of compounds are: (1) duality and (2) asymmetry.

(1) Duality

Compounds are always structured as pairs. These pairs themselves can be part of other compounds. The result of this duality is that a combination such as 'conservative party leader' is one compound but has two possible compounds as elements. Figure 2.1. demonstrates that there are two possible ways of representing the structure of 'conservative party leader'. One way is to view the head of 'conservative party leader' as leader with the modifier being the compound 'conservative party'. 'conservative party' itself seems to consist of a head party and a modifier, conservative. The second way of analysing 'conservative party leader' is to view the compound 'party leader' as the head and conservative as the modifier. In this case the head

[1] The symbol ? is placed before an utterance to mark it as an unacceptable construction.

itself is a compound which consists of a modifier, party, and head, leader. All compounds consisting of more than two nominals exhibit this kind of structural ambiguity.

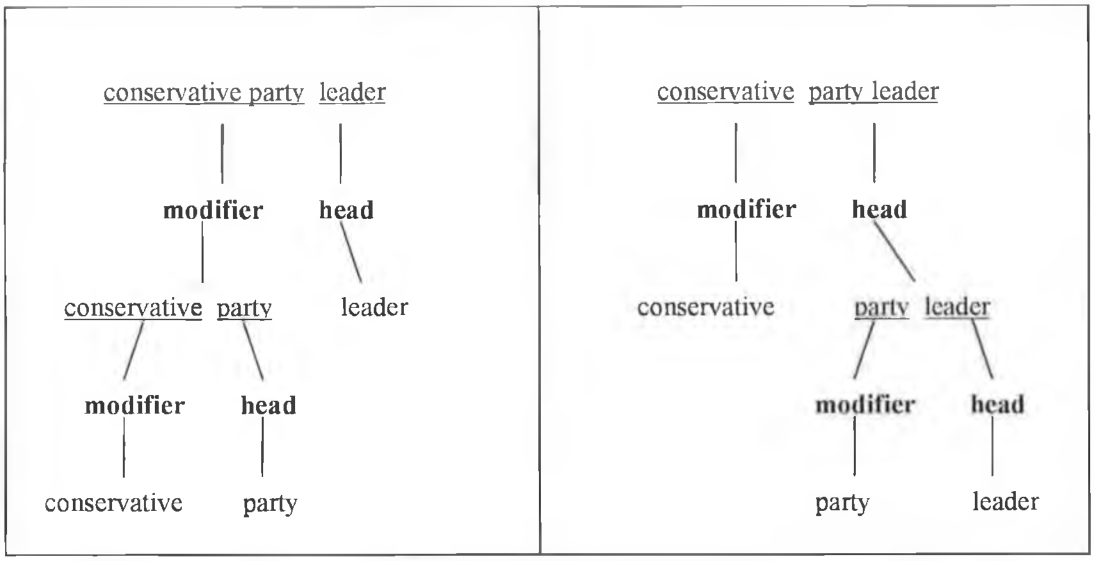


Figure 2.1 - Duality in action

'conservative party leader' can be interpreted in at least two ways. One is that a 'conservative party leader' is >>a party leader who is conservative<<. Another interpretation is that a 'conservative party leader' is >>a leader of the conservative party<<. Figure 2.1 demonstrates that the two interpretations are reflected in the different parses for the compound.

Ultimately, to interpret compounds we need to model how a modifier affects a head. Of course, in larger compounds, such as 'conservative party leader' the process must be applied recursively to the modifier or to the head before being applied to the rest of the compound.

(2) Asymmetry

A compound will in general either be nonsensical or gain a different meaning when reversed. For example, 'boathouse' and 'houseboat' have two very different meanings. Previously, it was stated that in a compound the head often marks out the type of object denoted by the compound as a whole. In these cases reversing a compound changes the head and thus the type of object denoted. This results in two compounds that are orthographical reverses of each other having different meanings. This is known as asymmetry. Certain combinations do not display the asymmetry property, e.g. 'servant girl', but these exceptions appear to be rare.

2.3 A Corpus based Examination of Nominal Compounds

To gather empirical information on the frequency of nominal compounds, an English language corpus was examined. The Wall Street Journal section of the Penn Treebank (Linguistic Data Consortium, CD-Rom, 1995 Release 2) was chosen as the focus of the study. The Penn Treebank (PT) is a corpus of annoted text marking out linguistic structures. It includes tagged and parsed data from the Department of Energy abstracts, IBM computer manuals and the Wall Street Journal.

One part of the PT provides tagged texts. The tagged section contains the entire contents of the Wall Street Journal from 1988. The PT has been automatically tagged and then checked and corrected manually by trained "taggers" under the supervision of professional linguists. The Wall Street Journal section (WSJ) consists of one directory named "wsj" which contains 25 sub-directories. Each of these numbered subdirectories contains 100 text files. The length of these files is not uniform; in one directory the smallest file contains 17 words while the largest contains 1,857 words. The text files are tagged as in Figure 2.2.

```
[ Tom/NNP Panelli/NNP ]
had/VBD
[ a/DT ]
perfectly/RB
[ good/JJ reason/NN ]
for/IN not/RB using/VBG
[ the/DT $/$ 300/CD rowing/NN machine/NN ]
```

Figure 2.2 Example of tagged corpus

The tag /NNP is used to refer to "Proper noun, singular" e.g. Tom. had is tagged /VBD to indicate that is a "Verb, past tense". Adverbs are marked with /RB. Other tags include: /IN for "Preposition/subordinate", /CD for "Cardinal number" and so on (the full set of tags is listed in Appendix A, Section 1). Note that simple phrases are generally placed on a separate line in the text file.

Nouns are classified into four types: (1) /NN, (2) /NNPS, (3) /NNS and (4)/NNP. These types are outlined in Table 2.2. Nouns are classified as either (a) proper or (b) common or mass. This division of common and proper is then further subdivided into singular and plural.

Table 2.2 - Noun types

Tag	Type of Noun	Example
/NN	noun, common, singular or mass	shed, thermostat, investment
/NNPS	noun, proper, plural	Americas, Amusements
/NNS	noun, common, plural	bodyguards, facets, coasts
/NNP	noun, proper, singular	Shannon, Meltex, Liverpool

As mentioned in Section 2.1.1, established noun-noun compounds are often represented orthographically with the constituent parts separated by a hyphen. This is also the case in the PT, where these established hyphenated compounds were tagged as single nouns. Table 2.3 gives examples of these compounds. As Table 2.3 shows, the constituent elements in the compounds were not tagged, e.g. 'Pianist-comedian' is not tagged as "Pianist/NN-comedian/NN". Presumably, this marks a distinction between compounds that the original authors already considered to be a part of the common parlance and those that were not. Compounds that are tagged as single nouns were analysed separately in Section 2.3. A rough count of these forms was undertaken.

Table 2.3 - Establish hyphenated compounds

Term	Tag
Pianist-comedian	Pianist-comedian/NN
Chinese-American	Chinese-American/NNP
public-relations	public-relations/NNS
Tele-Communications	Tele-Communications/NNPS

2.3.1 Performing the Analysis

Initially a Java program and a shell script were written (see Appendix A, Section 2) to find nominal compounds in the tagged Wall Street Journal section of the Penn Treebank. The program works on a line-by-line basis. The structure of the text files in the WSJ ensures that compounds do not wrap-around onto the next line (as can be seen in Figure 2.2). The program identifies a noun as a string of characters that ends with one of the following nominal tags: "\NN", "\NP", "\NNS", and "\NNP". When the program has identified a noun, it checks the noun's neighbour. If the neighbour is also identified as a noun then both the initial noun and its neighbour are saved as a new string, which represents the compound. This procedure continues iteratively to see if further immediate neighbours are also nouns. If more nouns are identified, they are appended to the saved compound string. So the string "lung cancer deaths" would be extracted as a compound with three elements. The (possible) sub-compounds of this particular compound, "lung cancer", and "cancer deaths", are not extracted separately. When the next neighbour is not a noun or the end of a line is reached the compound string is saved and the program continues searching for a new compound.

The program gathers all nominal compound tokens with information on the file from which they originate in the WSJ and stores these in a file. This file is also used to perform error checking. From this initial file an unsorted list of compound tokens is computed (by removing the information on the source files in the WSJ of the compounds). These compound tokens were then further broken up into three other lists. In total the software computes the four following lists:

1. a list of all nominal compound tokens
2. a list of all nominal compound types
3. a list of all noun-noun compound tokens (consisting of exactly two nouns)
4. a list of noun-noun compound types

2.3.2 Error Checking

The analysis undertaken in this study is based on two assumptions:

1. that PT contains no errors.
2. that the program and script work correctly.

The first of these is difficult to check but it should be noted that the PT has been automatically tagged followed by a manual correction phase and so should be relatively error-free (Marcus, Santorini & Marcinkiewicz, 1993). The second assumption was checked in the following way. One hundred random numbers were generated. The range of these random numbers was between 1 and 75695. This range covered the total number of nominal compound tokens collected by the program. Each random number generated was used to refer to an element in the list of all nominal compound tokens. For example, if the number 5000 was generated, this would taken to refer to the 5000^{th} nominal compound token. Taking one hundred compounds and looking at the original source, no errors were found. All the compounds were extracted correctly.

The program could still have missed compounds. The error checking above only examines the output of the program; it does not check to see that the output generated finds all compounds in the input file. To check this, a number of files from the WSJ were selected and the output of the program was manually checked against these files. It appears that the program does find all noun-noun compounds in files that are tagged in the format of Figure 2.2

2.4 Results

The program collected 75,695 nominal compound tokens of varying length from the WSJ. If the program came across a compound consisting of three or more nouns the whole entity was counted as a single compound. Counting of the subparts of these large compounds was not undertaken. Out of approximately 1,009,471 words and hence adjacent word pairs, at least 7.5% are nominal compounds of the type searched for. There are two things to note about this percentage:

1. the number of words in the actual corpus is open to debate.
2. there are actually more than 75,695 nominal compounds in the corpus.

The first of these reflects the fact any tagged item in the WSJ was counted as a word. Some of these are better described as morphemes than words (e.g. 's/POS or punctuation markers). So the number of actual words should be smaller. This suggests that the percentage of nominal compounds out of the total number of word pairs would actually be higher. Point (2) relates to the fact that many "established compounds" are tagged as single words in the corpus and compounds consisting of more than two nouns contain sub-compounds which were not counted separately.

Of the 75,695 compound tokens, there are 41,403 nominal compound types. Of the 75,695 nominal compound tokens 58,745 are composed of just two nouns. Out of these 58,745 noun-noun compound tokens, there are 29,660 noun-noun compound types. These results are summarised in Table 2.4. In relation to the size of the nominal compound type/token sets, the noun-noun compound is the most frequent.

Table 2.4 - Summary of Results

Compounds	Total Number
compound tokens	75,695
compounds types	41,403
noun-noun compound tokens	58,745
noun-noun compounds types	29,660

A rough count was undertaken of the number of established hyphenated compounds in the corpus, i.e. those that were tagged as single nouns. There are approximately 3,369 established hyphenated compound tokens in the corpus. This number is an over-estimate as this figure really refers to the number of nouns that contain hyphens. This results in a number of false positives, e.g. re-election. Despite this, the number of established hyphenated compounds is small compared to the number of non-hyphenated compounds found in the corpus.

2.4.1 How Big Can a Nominal Compound Get?

Nominal compounds can consist of more than two elements. Table 2.5 lists the number of compound types according to length.

Table 2.5 - How big nominal compound types can get

Compound types	Number	Examples
two elements	29,660	"abortion counselling"
three elements	10,621	"lung cancer deaths"
four elements	809	"California Health Facilities Authority"
five elements	226	"Defense Advanced Research Agency contract"
six elements	72	"state bar association policy making body"
seven elements	11	"General Electric Co. National Broadcasting Co. Unit"
eight elements	3	"New York Stock Chairman John J. Phelan Jr."
nine elements	0	
ten elements	0	

Compounds can be become rather large, e.g. "New/NNP York/NNP Stock/NNP Chairman/NNP John/NNP J./NNP Phelan/NNP Jr./NNP". It should be noted that close to three-quarters of nominal compound types consist of two elements.

A careful reader will note that the total number of compound types in Table 2.5 does not add up to the number of compound types given in Table 2.4. One large compound was found that had eleven elements. On closer examination it appeared that this was a result of a typographical error in an article that was left uncorrected but still tagged.

2.4.2 Caveat Lector

Although the Penn Treebank has been thoroughly checked there are a number of errors in the tagging. These would be small compared to the database as a whole. Indeed they may not be noticed at all. As the originators of the PT state: "Many texts are not models of good prose, and some contain outright errors and slips of the tongue" (Santorini, 1990, p1). Thus the results above are suggestive rather than conclusive.

2.4.3 Implications of Results

Nominal compounds occur frequently in the corpus. The actual number of nominal compounds found in our experiment is an underestimation. This finding reinforces the discussion of compounds in O'Grady et al. (1993) who suggests that noun-noun compounds are widespread. It also ties in with Canon (1987) who reported that up to 55% of new items in a given corpus were compounds of existing words.

The interpretation of nominal compounds will be important for any semantic approach to language. As a starting point we focus on noun-noun compounds consisting of exactly two nominal elements (rather than nominal compounds in general) for two reasons:

1. the frequency of noun-noun compounds
2. duality.

Taking the first reason, clearly, in the corpus examined, noun-noun compounds are the most prevalent of all nominal compounds. The second reason suggests that if we know how to model how a modifier and head interact in a noun-noun compound, we should be able to deal with larger compounds recursively.

2.5 Chapter Summary

This chapter presents an examination of nominal compounds. Compounds have certain properties: asymmetry and duality. A study of compounds in the Penn Treebank, an English language corpus was undertaken. This study suggests that nominal compounds are relatively frequent in the corpus examined (and extrapolating from this perhaps in English in general). The noun-noun compound appears to be the most common of nominal compounds. This is interesting as this particular form is the focus of research on concept combination. Two reasons as to why noun-noun compounds should be examined were outlined. These reasons are: their relative frequency and the duality of compounds.

Chapter 3 - Theories and Models of Concept Combination

3.1 Introduction

This chapter outlines the main theories of concept combination: Selective Modification (Smith, Osherson, Rips, & Keane, 1988), Concept Specialisation (Murphy, 1988), Dual Process Theory (Wisniewski, 1997a, 1997b), Constraint Theory (Costello, 1996; Costello & Keane, 1998; Costello & Keane, 2000) and the CARIN model (Gagné & Shoben, 1997, Gagné, 2001). They are briefly set out in chronological order before being examined in more detail in individual sections. Each theory is examined in relation to at least one noun-noun compound to demonstrate how each theory approaches the process of concept combination in a distinct way. The outline of these theories will reveal that each is incomplete in some way. At the end of this chapter the need for a new theory is set forth.

3.2 The Theories Of Concept Combination - A Brief Chronology

The theories of concept combination are now briefly outlined in chronological order. They are examined in more detail in Section 3.4. The chronological order of the theories can be seen in Figure 3.1. In this figure, "onwards" highlights the fact that some of these theories are still being actively researched.

The first theory is Selective Modification (Smith, Osherson, Rips, & Keane, 1988). The model that Smith et al. proposed was to account for adjective-noun combinations but they suggested that it could be extended to account for noun-noun combinations. Their proposal is that in a combination the modifier selects an aspect of the head and modifies it. In a combination such as ‘green apple’, an aspect of the concept *APPLE* is modified. This aspect is presumably the *colour* of *APPLE*.

According to Murphy (1988) Smith et al.'s theory is limited because it could not be extended in a straightforward way to account for noun-noun combinations. Murphy pointed out that nouns do not often act like adjectives and this was the motivation for his theory of Concept Specialisation. One example that Murphy (1988) offers is that of 'apartment dog'; here it is not clear what exactly *APARTMENT* selects and modifies. This is a result of the concept's relative complexity. Concept Specialisation suggests that world knowledge plays a role in the interpretation of nominal compounds.

The next theory was proposed by Costello (Costello, 1996; Costello & Keane, 1998; Costello & Keane, 2000) and is known as Constraint Theory. This approach was largely motivated by the need for a theory that accounted for the polysemy of combinations. For example, 'chocolate bar' could be interpreted as >>a candy bar made of chocolate<< or as >>a public bar that is predominantly brown in colour<<. In this case bar is ambiguous but multiple interpretations can be generated from unambiguous words, e.g. 'pencil bed' in Section 1.5.1. Constraint Theory suggests that there are three constraints on the process of concept combination: diagnosticity, plausibility, and informativeness. Constraint Theory is also the only theory that has been implemented.

Dual Process Theory (Wisniewski, 1997a, 1997b) is the first model to introduce the idea of structural alignment into the area of concept combination. Initially this theory was named The Structural Alignment View by its originator but is referred to as Dual Process Theory by other theories (e.g. Costello & Keane, 1998; Gagné , 2001). Dual Process Theory posits that the interpretations that subjects give to combinations can be classified into three different categories: property-based, relation-based, or hybrids.

The CARIN model (Gagné & Shoben, 1997) draws heavily on the work of Levi (1978) and is the most linguistic approach of all the theories. Gagné and Shoben (1997) propose that there are a number of relations associated with each noun and that interpreting combinations is a question of selecting the most appropriate relation to link the head and modifier. They suggest that a native speaker has knowledge of these relations and how frequently they occur with certain words. We will argue that this model is not a full theory of concept combination.

Selective Modification –1988
Concept Specialisation – 1989
Constraint Theory – 1996 onwards
Dual Process Theory – 1997 onwards
CARIN model – 1997 onwards

Figure 3.1 – Main theories of concept combination

3.3 Theory Components

In this section we outline the various component ideas that are needed to describe the theories reviewed in the following section in detail. These component ideas are: prototypes, properties, frames, slots, values, structured graph representations. They are all concerned with the representation of knowledge. Theories of concept combination attempt to account for how different concepts interact to form an interpretation. These concepts are often represented as frames as will be seen in Section 3.4.

3.3.1 Frames And Conceptual Graphs

Luger and Stubblefield (1998) make a distinction between frames and network representations, e.g. conceptual graphs. They say that knowledge can be organised into complex units which represent situations or objects in a domain. These units are termed frames (originally coined by Minsky (1975)). A frame is a data structure used to represent well-understood stereotyped situations or objects. An individual frame can be seen as a "record" data structure. A frame is made up of a number of slots. These slots have values (or fillers), e.g. Figure 3.2 displays a frame for 'hotel bed'.

hotel bed
Superclass: bed
Use: sleeping
Size: king
Part: (mattress frame)

Figure 3.2 - Frame for "hotel bed" (adapted from Luger & Stubblefield, 1998)

When subjects are asked to describe objects they often do in terms of features, properties or attributes as well as relations. For example, a *CAR* may be described in terms of having *four wheels*, *an engine*, *a driver*, *passengers* and *being used for transport*. Some of these could be described as either a property, a feature or an attribute. We will treat the terms features, properties and attributes as synonymous and we will use the term property to cover all three terms. In a frame the slot and filler attempts to represent the properties of the concept, e.g. a hotel bed in Figure 3.2 has the property king size. The slot is the name of the property and the filler is the value for this property. So in this example, size is a slot and king is a filler. The value can be another frame and frames are rarely used in isolation.

The information represented in Figure 3.2 can also be treated as a graph. In a graph based representation the slots can be represented as arcs and the fillers as nodes. This type of representation is given in Figure 3.3. Russell and Norvig (1995) suggest that any distinction between frames and networks is more apparent than real. This is the view that is also adopted throughout this thesis. Both frame and network representations see objects as being nodes in a graph, with nodes organised in a taxonomic structure with links between nodes representing binary relations. The difference between the two is that in a frame, the binary relations are seen as slots in one frame that are filled by another frame. However, in a semantic network, the binary relations are thought of as arcs between nodes. Russell and Norvig (1995) suggest that the meaning and implementation of both is often identical. Thus a frame can also be viewed as a network based representation. Section 3.4 will show that the majority of theories treat concepts in terms of frames rather than networks.

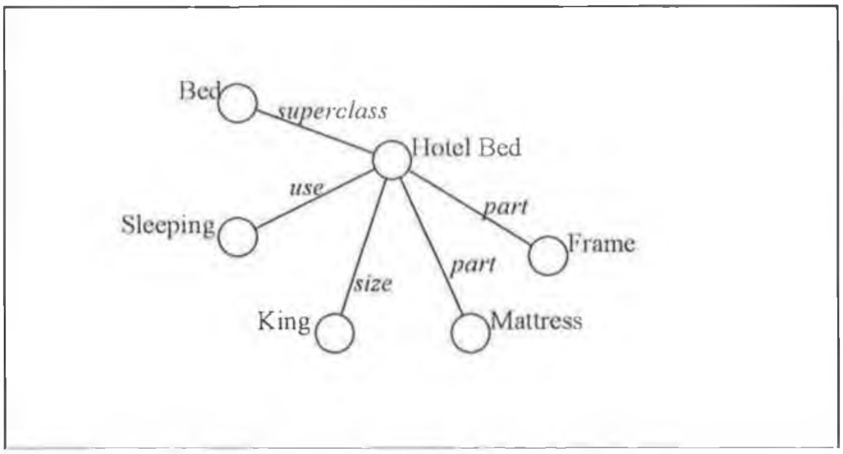


Figure 3.3 - Network representation of Hotel Bed

3.3.2 Prototypes

It is impossible to talk of prototypes without discussing concepts and categories, which were first touched on in Section 1.2.1. Prototype theory is a theory of concepts, a theory that says something about the structure of concepts. Discussion of prototypes is set often against the backdrop of categorization. Categorization is the process through which a person places a concept into a particular category, e.g. is a *TOMATO* a *FRUIT* or a *VEGETABLE*? More broadly it involves the discrimination between inputs and so is a fundamental cognitive process.

In the mid-to-late 1970s Rosch (Rosch, 1975; Rosch & Lloyd, 1978) proposed that categorization involves comparisons with a prototype of the category in question and the prospective member. Some concepts are better examples of a category than others. For example a *SPARROW* is a better example of the category *BIRD* than an *OSTRICH* (Aitchison, 1997). This goodness of example effect led researchers to suggest that there must be a system of family resemblances which determine whether a concept is a good example of another concept or not. This ties in with earlier philosophical investigations by Wittgenstein, e.g. his discussions of the concept *GAME* (Wittgenstein, 1968).

This view of family resemblance proposed by Rosch (Rosch, 1975; Rosch & Lloyd, 1978) broke with the prevalent thinking at the time that membership of a category was based on defining features. A view that accepts defining features (necessary and sufficient features) also holds there cannot be "good examples" (Aitchison, 1997). A concept is either a member of a category or it is not. If membership is determined by defining features, all members must have these features so no member can be a better member than another. Yet subjects still rate items as good examples of concepts or not.

Rosch suggested (Rosch, 1975; Rosch & Lloyd, 1978) that a prototype consists of features, attributes and properties but, importantly, each of these is weighted for typicality. Diagnosticity is a measure of how typical a feature, an attribute or property is for a group as a whole.[1] This can be calculated in various ways, e.g. by comparing a feature in one concept with all other concepts. If a property is fairly uncommon across all concepts then its diagnosticity should be high for the category in question. Conversely, a feature that is common across all concepts should have a low diagnosticity as it will prove useless in determining membership. A good example of a concept should have the most typical features, those that have the highest rating for diagnosticity. To be a member of a category an item just needs enough typical features to distinguish it from other examples. Both the classic approach and the more modern prototype approach do agree on one thing: concepts are made up of features, attributes or properties. According to Eysenck and Keane (1995) prototype theory suggests that concepts should also have information on the diagnosticity of these features, attributes or properties. Prototypes can be represented in terms of frames. Frames do not usually include information on the diagnosticity of a property. However, a weighting can be given to every slot with respect to how typical it is to the category the prototype belongs to.

3.4 The Theories In Detail

This section will examine the main theories of concept combination in detail but this examination is constrained by the original presentations of these theories. Constraint Theory has had the most detailed exposition in the literature while some models such as Selective Modification seem less detailed in comparison. In outlining these theories there will be a difference in the amount of time spent on each theory. Each theory is being examined in as much detail as possible; it just happens that some theories are worked out in more detail than others. A difference in the depth of examination does not reflect a bias towards certain theories over others.

Each theory is presented in terms of its own representations. At least one example is given to demonstrate how each theory views the process of concept combination. For each theory we outline the problems that it faces. In some cases these problems are substantial. We also list combinations that cannot be handled by each theory.

---

[1] Conversely diagnosticity could also be viewed as a measure of atypicality as the diagnostic attribute, feature or property would be atypical for non-members of the group.

3.4.1 Selective Modification

Selective Modification (Smith et al., 1988) is a theory that developed from research on categorisation, most notably prototype theory, and so focuses on the combination of prototypes. There are two main views as to what a prototype may be; both are probabilistic and recognise that concepts do not have clear-cut boundaries and are often ill-defined (Eysenck & Keane, 1995). From the Selective Modification viewpoint (Smith et al, 1988) a prototype is *"...roughly, a description of the best examples or central tendency of a concept"* (p486).

Smith et al. (1988) focus largely on adjective-noun combinations, completely ignoring the class of noun-noun combinations. The model they propose, Selective Modification, consists of three components:

1. a model for prototype representation
2. procedures for modifying simple prototypes to form combined concepts
3. A mechanism for determining the similarity between a prototype of a category and an instance of the same category

Our focus will be on the first two components as these are the most important with respect to concept combination.

The first component details how prototypes are represented in their theory. A prototype is a pre-stored representation of the usual properties associated with a concept's instances. So a prototype for the concept *APPLE* would include information on properties that are part of subjects' common knowledge of *APPLES*, e.g. *has seeds*. The properties that make up a prototype need not be necessary or sufficient so a representation of apple may include information such as, is *red, round* and *smooth*. Properties can be broken into two parts: attributes and values. An *APPLE* prototype would include attribute-value pairs such as: *colour=red, shape=round, texture=smooth.*

A prototype may also have information on the salience of each value. A salient property is a property that subjects strongly associate with a concept. Subjects are generally faster in deciding that *"an apple is red"* over *"an apple is round"*. The conclusion drawn from this is that the attribute *red* is more salient than *round* with respect to the concept *APPLE*. Further to this a prototype may also include information on the diagnosticity of each attribute. According to Smith et al (1988) the diagnosticity of an attribute is the measure of how useful the attribute is in discriminating instances of a category from other instances of contrasting concepts.

The representation of a concept in Selective Modification (Smith et al, 1988) includes the following:

1. An attribute value structure
2. The salience of a value
3. The diagnosticity of an attribute

A representation of a prototype for *APPLE* is given in Table 3.1. This prototype is made up of three attributes: *colour, shape and texture*. As far as Smith et al (1988) are concerned each attribute has various values, e.g. an *APPLE* can be *red, green* or *brown*. The number to the left of the attribute represents the diagnosticity of the attribute. The salience of each value is given as a number to the right of the value. For this example, the scores given for salience and diagnosticity have been arbitrarily devised. In Table 3.1 the most diagnostic attribute is that of *colour* with a score of 1. The least diagnostic attribute is *texture* with a score of 0.25. Note that each attribute value has a salience score, even ones with a low diagnosticity. The salience of *smooth* is 25, which is the highest salience score along with that for *red*. From Table 3.1 it can be seen that the most salient features of an *APPLE* are *smoothness* and the *colour red*. There may be an overlap between salient properties and diagnostic ones. With Selective Modification a distinction is drawn between the attributes and values of a property. Only attributes can be diagnostic and only attribute values can be salient.

Table 3.1 - A prototype representation of *APPLE* (Smith et al. 1988)

APPLE	
Diagnosticity - Attribute	Value - Salience
1 colour	{ red 25, green 5, brown... }
0.5 shape	{ round 15, square, cylindrical 5... }
0.25 texture	{ smooth 25, rough 5, bumpy... }

The basic proposal of Selective Modification is that each attribute in the adjective concept selects the corresponding attribute in the noun concept. Every attribute selected in this way in the noun has an associated increase in its salience and diagnosticity. In a combination such as 'shrivelled apple', if the attributes of *SHRIVELLED* are *shape* and *texture* then these are the attributes of *APPLE* that are modified. These attributes would have their diagnosticity scores increased. However, the outline of how adjective modification works is given in relation to adjectives which are assumed to have only one attribute (Smith et al., 1988). Adjectives are not described as being rich in attributes and properties, as nouns are, but are instead modelled as single properties. So Selective Modification will have great difficulty in being applied to noun-noun compounds as has been originally noted by Murphy (1988).

In Selective Modification the adjective selects the relevant attribute in the noun. In our view this appears to be done via matching between the modifier and the head. If this is the case then presumably the attribute associated with the adjective must exist in the head. All the salience scores in the head are shifted so that the saliency value of the attribute selected increases and the diagnosticity value for the attribute is also increased. In Figure 3.4 there is a representation of *RED* and a partial representation of *APPLE* (see Table 3.1 for the full representation). In *APPLE* the colour attribute has three values, *red, green and brown*, two of these have salience values. When the combination process occurs all the salience scores are moved to *red*, so *red* becomes the most salient attribute of 'red apple'.

RED	*APPLE*	"red apple"
Color { red }	1 color { red 25, green 5, brown... }	2 color { red 30, green, brown... }

Figure 3.4 - Shifting the diagnostic and salience values through combination

The selection of the attribute and the changing of its salience value suggest that the attribute should already exist in the head concept. But what happens when the attribute does not exist? In that case Smith et al. (1988) suggest that the attribute may be temporarily added in. In that case, the salience and diagnosticity must still be worked out. Perhaps the salience scores of the other attributes are given to this new attribute; Smith et al. (1988) do not go into detail. The model deals only with modifiers that can be represented as single attributes. There is the possibility that an adjective may involve more than one attribute and so modify more than one attribute in the noun. Again, this is not dealt with explicitly in Smith et al. (1988).

3.4.1.1 Problems With Selective Modification

We argue that as it stands Selective Modification cannot deal with noun-noun compounds. The structure of nouns cannot be represented in terms of a single attribute. Selective Modification implicitly acknowledges this. The representation of *APPLE* in Table 3.1 is not comprised of a solitary attribute. Rather, this concept has several attributes. A combination involving noun-noun compounds immediately poses the problem of which attribute(s) is (are) selected to be modified. As suggested above, Selective Modification offers no account of how this is to be done.

At a broader level, the combination process may involve the changing of the diagnosticity and salience of attributes but it also involves much more than this. Concepts can be linked to each other via a third concept or a property of one concept may be applied to another. Selective Modification does not account for this as its notion of concept combination is focused on accounting for changing salience and diagnostic scores. This is understandable as it is concerned with accounting for subjects' typicality judgements for combinations but it ignores the problem of generating interpretations for these compounds.

3.4.1.2 Combinations Selective Modification Cannot Deal With

Murphy (1988) suggests that Selective Modification cannot deal with noun-noun compounds. A case might be argued that where concepts are strongly associated with a property this salient property is the attribute that is involved in the Selective Modification process. However, other models such as Constraint Theory (Costello, 1996; Costello & Keane, 1998; Costello & Keane, 2000), which attempt to find the properties of concepts that play a role in interpretation, still assume complex conceptual structure. By making this case further questions arise, e.g. how do we find the relevant property? This is a question which is dealt with by Constraint Theory and Dual Process Theory below. As a model of concept combination Selective Modification never really gets off the ground, although it does suggest that a slot-filling model may be useful in explaining concept combination.

3.4.2 Concept Specialisation

Murphy's (1988) theory of Concept Specialisation can be seen as attempt to improve on the weaknesses in Selective Modification. His view of concept combination is one of a process that takes simple concepts, e.g. *ENGINE*, *REPAIR*, to form complex concepts, e.g. "engine repair". Murphy defines simple concepts as those that are generally represented as single lexemes. A complex concept is made up from these simple concepts and so will be represented by two lexemes. His term, complex concept, would be similar to our term, combination.

Murphy (1988) suggests that world knowledge is involved in concept combination. Concept combination is not a closed process in that information outside of what would typically be considered to be a part of either concept is often involved. This information is associated with a concept but may not be contained in a representation of that concept. The nature of this association is not outlined by Murphy (1988). However, this extra information effects concept combination in two ways:

1. it picks out the appropriate slot to be specialised.
2. it cleans-up (elaborates) the combination.

The term specialised should be explained further. From the Concept Specialisation perspective, concept combination involves the modifier specialising an aspect of the head. The head is represented as a frame with a number of slots. The modifier acts by picking out the most appropriate slot. Consider the combination 'apartment dog'. With regards to Concept Specialisation the modifier may change the value of a possible *location/habitat* slot of the concept *DOG*. So, an 'apartment dog' can be interpreted as >>a dog that lives in an apartment<<.

Murphy (1988) argues that by treating adjective-noun combinations and noun-noun combinations in the same way Selective Modification ignores:

1. The primary features of a noun (when acting as a modifier) do not always carry over to that head.
2. Modifiers do not have the same effect on every head they modify.
3. That point (1) becomes stronger the more complex the structure of the modifier.

The primary features of a noun are probably best described as those that are the most associated with a concept. For the concept *APPLE*, the feature *colour green* might be salient. The combination ‘apple grower’, is not however >>a grower who is green<<. In this combination the salient property of *APPLE* has not been carried over. Rather it seems that the concept *APPLE* fits into the concept *GROWER* and forms the new concept 'apple grower'. An 'apple grower' being >>a person who grows apples<<.

Also, it is clear that a modifier can have different effects on different heads even though it occupies the same modifier role as outlined by the Table 3.2.

Table 3.2 - A modifier having different effects on different heads

Combinations with advertising as a modifier
'advertising executive' is >>an executive in an advertising company<<   'advertising time' is >>the broadcast duration of an ad<<   'advertising recession' is >>the economic state where advertising companies reduce the amount of ads they place on T.V., radio and newspapers<<

The representation that Murphy (1988) uses for concepts is frame-based. Figure 3.5 displays a representation of *DOG*. The thrust of Concept Specialisation can be found in the "apartment dog" example that Murphy uses. This combination could be interpreted as >> a dog that lives in an apartment <<. For Murphy, an interpretation such as this involves the modifier selecting a slot in the head and replacing it. In many ways his approach could be described as *"selective modification"* (if a theory did not already bear this name), as a slot is selected and modified. In fact, to distinguish between the two theories Murphy refers to Selective Modification as the Feature Weighting Approach.

Name:	dog
Body-parts:	Legs: 4,3
	Head: 1
	Hair
	Eyes: 2
	Colour: brown, white, black...
	(etc.)
Habitat:	home, streets
Functions:	best friend, guard home
Behaviours:	bark, bite, eat, sleep, chase cats...

Figure 3.5 - A representation of dog (Murphy, 1988)

Murphy suggests that when nouns act as modifiers they do so in a more complex manner than adjectives. For example, an adjective such as *RED* appears to refer to a single dimension and generally has the same effect on every head noun (unless *RED* is part of an idiom, e.g. 'red herring'). A "red apple" is >> an apple that is red <<, a "red car" is >> a red coloured car <<. He suggests that concept combination involves finding an appropriate slot to be filled. In concept combination, the modifier replaces the filler of the most appropriate slot.

The selection of this slot is guided by the world knowledge that an interpreter has. For example, an 'apartment dog' could be interpreted as >> a dog that functions as apartment <<. This is an unlikely interpretation as dogs do not usually function as apartments; people cannot live in dogs. It is seemingly obvious information like this that participants have available when interpreting noun-noun compounds. From this perspective concept combination is not a closed process.

Murphy also suggests that the fact that concept combination is an open process can be demonstrated in other ways. Using adjective-noun examples he suggests that sometimes combinations have properties which belong to neither of the constituents. Consider the example, "empty store": a person may report that one of the properties of empty store is that it is *losing money* but *losing money* is not a property of *STORE* and does not seem to be a property of *EMPTY* either. Murphy (1988) suggests that this information is provided by applying world knowledge to the process of concept combination.

3.4.2.1 Problems With Concept Specialisation

The central problem with the Concept Specialisation theory is lack of detail. In the Concept Specialisation model the modifier selects and specialises an aspect of the head. However, how this is actually done is never clearly explained except in vague terms such as, it is "guided by world knowledge". As with Selective Modification there is no given mechanism for explaining how the modifier selects and specialises (or modifies) some aspect of the head. So again this theory misses the point somewhat. In its favour it suggests that a slot-filling approach may be used to model concept combination but then it offers no explanation as to how this slot-filling process will take place. Overall, this model suggests that interpretations of combinations can be viewed in terms of a slot-filling process but does not really deal with the generation of these interpretations.

3.4.2.2 Combinations Concept Specialisation Cannot Deal With

Concept Specialisation will have problems in dealing with most types of combinations. Although it attempts to tackle the noun-noun compounds that Selective Modification cannot deal with it ends up in the same position as this model. The reason is that it does not provide details of how the modifier and head interact. In effect it has been left up to later models (e.g. Constraint Theory and Dual Process Theory) to fill this gap.

3.4.3 Constraint Theory

Constraint Theory (Costello, 1996; Costello & Keane, 1998; Costello & Keane, 2000) marks a new direction in research on concept combination. Previous theories focus exclusively on a single interpretation paradigm. Constraint Theory, however, examines the polysemy of combinations and takes a multiple interpretation approach. Many combinations, especially novel ones, can be interpreted in a number of ways. For example, consider the combination 'chocolate door'. This could be interpreted in numerous ways: >>a chocolate coloured door<<, >>a bar of chocolate shaped like a door<<, >>a door made of chocolate<<. Although some combinations may seem more coherent or more likely to occur in the real world than others, this does not take from the fact that subjects can generate interpretations for things that may not exist in the real world. The area of concept combination has largely focused on finding a single interpretation for combinations despite the fact that combinations can be polysemous.

Although Constraint Theory attempts to examine polysemy, the pragmatics of concept combination appear to be the guiding force behind the theory. According to Costello and Keane (2000) an interpreter when presented with a combination will make a number of pragmatic assumptions about the intentions of the speaker who made the phrase. Assuming that the speaker and the listener are co-operating Costello & Keane (2000) make a number of assumptions:

1. The combination is one the listener should understand
2. The intended concept is best identified by the two concepts in the combination
3. The intended concept is one for which both words in the combination are necessary and sufficient.

Costello (1996) proposes that there are three main constraints on concept combination. Each of the three constraints is supposed to reflect the above pragmatic assumptions. The plausibility constraint ensures the production of something that should be more-or-less already known to the listener. The diagnosticity constraint ensures that interpretations contain some properties diagnostic of both the constituent properties. The informativeness constraint ensures that an interpretation conveys the appropriate amount of new information. To interpret a combination correctly the listener constructs an interpretation which best satisfies all three constraints.

The diagnosticity constraint requires that an interpretation contains some predicates which are diagnostic of the modifier concept and some predicates which are diagnostic of the head concept. The diagnosticity constraint predicts that of the following two interpretations the first is the more acceptable:

1. A 'cactus fish' is >> a prickly fish <<
2. A 'cactus fish' is >> a green fish <<

This is because *prickly* is more diagnostic of *CACTI* than *green*. In contrast, *green* is typical of most *PLANTS* and it is not particularly diagnostic of *CACTI*. The constraint does not demand that all the diagnostic predicates of each constituent concept are used; only enough predicates to identify an instance of a concept well. So the combination 'cactus fish' would probably not contain the property, *grows-in-desert*.

Research on single interpretations of combinations (e.g. Wisniewski, 1996) suggests that there are three basic types of interpretations. These are:

1. Relational - attempting to find a relation between two concepts.
2. Property-based - suggesting that a property of the modifier concept holds for the head concept.

3. Conjunctions or Hybrids - where the features of the modifier concept and head concept are conjoined to form a new entity.

To this Costello & Keane (1997) add:

4. Direct-reference - combinations that refer to already existing concepts, e.g. interpreting 'pencil bed' as >>a pencil case<< (Costello & Keane, 1997). This may already be subsumed under (1), (2) and (3).
5. Reversals - combinations where the head and modifier appear to exchange roles, i.e. the head modifies the modifier, e.g. interpreting 'plastic dog' as >>plastic shaped like a dog<<.

The diagnosticity constraint is applied equally to all interpretation types, however the effect of this principle is different for each interpretation type. For each interpretation type the following occurs:

1. Property: diagnostic properties of one concept are asserted of the other.
2. Conjunctive and hybrid: a concept is constructed that contains both the diagnostic properties of both constituent concepts.
3. Relational: the diagnostic predicates of the constituent concepts occur implicitly in different parts of the interpretation.

Although Costello & Keane (Costello, 1996; Costello & Keane, 1998; Costello & Keane, 2000) are not explicit as to what happens with reversals and direct reference combinations we suggest that the following could occur. Reversals involve reversing the roles of the head and modifier. So that (1), property interpretation, and (3), relational interpretation, above could also be true of reversals. For example in Section 1.6.1, "bed pencil" was interpreted as >> a thin bed <<. Here a property of the head has been asserted of the modifier. This would involve (1) but with the modifier and head roles reversed. The direct reference would be found when an interpretation matches a concept that exists in memory.

The plausibility constraint ensures that the interpretation should contain properties that are consistent with prior experience. When a person interprets a novel combination the speaker will assume that the intended combined concept is describing one that a person more or less already knows. This seems to rule out combinations which refer to concepts that may not have existed for the listener before the combination process; where it may be the intention of the speaker to highlight something new. Consider the combination, 'academic polyfilla', this combination could refer to academics on temporary contracts.[2] This is a novel combination

[2] This combination was used in conversation the author took part in.

and should impart some novel information about the way academics on temporary contracts are treated, i.e. they are a stop-gap measure to fix any apparent holes in a department's teaching timetable. People are playful and creative in their use of language; theories of concept combination should not ignore this.

The plausibility constraint predicts that of the two combinations below the first is more acceptable than the second:

1. A 'shovel bird' is >> a bird with a flat beak it uses to dig for food <<
2. A 'shovel bird' is >> a bird which uses a shovel for digging for food <<

The first is more in tune with what happens in everyday experience in the world, where *BIRDS* often dig for food. The second could only be understood in terms of a special context, e.g. a bird in a cartoon. Plausibility is determined in relation to all the information in the knowledge base so information from outside of the original constituents in the combination can be brought into play.

The informativeness constraint ensures that an interpretation should convey new information such that both words are necessary and sufficient for that information. New information should come from both the head and the modifier, not just one of the concepts. The following would be unacceptable when the informativeness constraint is applied:

1. A 'head hat' is >> a hat worn on the head <<
2. A 'car vehicle' is >> a car <<

The first offers no new information in relation to *HAT*, while the second provides no new information about *CAR*. People have difficulty in interpreting redundant modifiers where the modifier provides no new information beyond that already contained in the head. Informativeness is only used to determine if a combination is informative or not. The three constraints do not contribute equally to the process of concept combination. The constraints of diagnosticity and plausibility govern the general acceptability of an interpretation. These are, thus, the most important constraints.

When people interpret combinations they do so quickly and efficiently. For many combinations a large number of interpretations are possible. An important requirement on theories of concept combination is that they should be computationally tractable - they should produce interpretations in a reasonable time. Plausibility itself may not be a computable notion considering how it draws in open-ended world knowledge. However, the notion of plausibility, as marked out by the plausibility constraint in Constraint Theory, reduces the problem by looking at the overlap of properties between a concept in memory and a potential interpretation. The greater the overlap the more plausible the interpretation. Constraint Theory, however, does not account for figurative combinations (see Section 3.3.3.2).

The exact details of how Constraint Theory works are fleshed out in the CT3 program, an implementation of this theory (Costello & Keane, 2000). Using the compound, 'finger cup', an analysis of Constraint Theory will be undertaken. Table 3.3 contains representations needed for interpreting finger and cup. Note that this model uses more than two representations. In the table there is not just an abstract representation of *FINGER* and *CUP*; there are also representations of instances, e.g. there are two types of *CUP*. Unlike the previously mentioned models, the CT3 model uses a large knowledge base.

Table 3.3 - Representation of instances of finger and cup (Costello & Keane, 2000)

Finger	Name[F, "Finger"]	Name[H, "Hand"]
	Shape[F, Tubular]	Consistency[H, Solid]
	Size[F, Small]	Size[H, Medium]
	Consistency[F, Solid]	Shape[H, Flat]
	Part-of[F, H]	Holds[H, _]
Cup	Name[C, "Cup"]	Name[L, _]
	Shape[C, Hemispherical]	Consistency[L, Liquid]
	Size[C, Small]	
	Consistency[C, Solid]	
	Contains[C, L]	
Cup	Name[D, "Cup"]	Name[E, "Expresso"]
	Shape[D, Hemispherical]	Consistency[E, Liquid]
	Size[D, Small]	Colour[E, Brown]
	Consistency[D, Solid]	Taste[E, Bitter]
	Contains[D, E]	Temperature[E, Hot]
Bowl	Name[B, "Cup"]	Name[K, _]
	Shape[B, Hemispherical]	Consistency[K, Solid]
	Size[B, medium]	Wash-in[K, B]
	Consistency[B, Solid]	
	Contains[B, K]	

There are three stages to generating an interpretation in the CT3 model:

1. Constructing partial interpretations using diagnosticity
2. Plausibility and the generation of full interpretations
3. Informativeness and overall acceptability

The diagnosticity of each attribute is a function of its relative occurrence in members against non-members of a category. The more frequently an attribute is shared by instances and the more frequently it is not shared by non-instances the more diagnostic it is. Diagnosticity was a major concern for Selective Modification (Smith, Osherson, Rips, & Keane, 1988) but was calculated in a wholly arbitrary way. From viewing the representations we can suggest the most diagnostic attributes of *FINGER* are that its *shape is tubular*. Remember that the interpretation of a compound must have features that are diagnostic of both the concepts. The diagnostic features of *FINGER* and *CUP* in this example are: *shape(small) contains(liquid), shape(tubular)*. Partial interpretations are created based on the diagnostic predicates.

An interesting twist that Constraint Theory puts on diagnosticity is that diagnostic attributes can be grouped together. In other models, e.g. Selective Modification, diagnostic attributes are treated in isolation. In Constraint Theory attributes can be grouped together if they occur within a category but do not occur in other concepts. Table 3.4 lists some of the diagnostic attributes for *FINGER* and *CUP*. Scores have been given for each attribute. This score is a ratio between its occurrence in a concept (which will always be one) and the number of other concepts it occurs in. If the attribute only occurs once it is given a value of one.

Table 3.4 - Some diagnostic features of *FINGER* and *CUP* (Costello & Keane, 2000)

FINGER	*CUP*
Shape[F, tubular] 1	Size[C, small] & Contains[C, L] 1
Part-of[F, H] 1	Contains[C, L] 2/3
Size[F, Small] 1/3	Consistency[C, solid] 1/4

The diagnostic features are then used to form partial interpretations. These partial interpretations are then fleshed out by the plausibility constraint. The diagnostic and plausibility constraints can be seen as a construction process that creates interpretations. One partial interpretation is listed in Table 3.5 where F is a finger and C is a cup. Although only one partial interpretation is presented here, several partial interpretations would be generated by combining the diagnostic predicates in Table 3.4 exhaustively.

Table 3.5 - A partial interpretation of 'finger cup' (Costello & Keane, 2000)

'finger cup'
Shape[F, tubular] 1
Size[C, small] & Contains[C, L] 1
Contains[C, L] 2/3

A partial interpretation such as the one in Table 3.5 is then examined in relation to prior experience, this is the plausibility constraint. Plausibility in the CT3 model is measured by examining the degree to which the predicates of a partial interpretation exist in stored concepts. If an interpretation already exists in the knowledge base there will be a complete overlap in the shared predicates between the two. The intuition at work here is that the more predicates a partial interpretation shares with an existing concept the more plausible it is. Where there is an overlap between a partial interpretation and an existing concept, predicates from the existing concept are added to the interpretation. This is the mechanism by which partial interpretations are fleshed out. In the example, 'finger cup', this new information may be taken from *BOWL*, e.g. *Wash-in[K, B].*

The informativeness constraint takes the full interpretation and compares it against the prototypes of the constituent parts. For example, if an interpretation contains all predicates that already exist in one of the component parts, i.e. is a subset of this component, it is uninformative. This constraint captures the uninformativeness of combinations such as the 'head hat' example above.

3.4.3.1 Problems With Constraint Theory

Although Constraint Theory is an extremely powerful model and also the most detailed, there is one major weakness with this approach: it places too much emphasis on diagnosticity. There may be combinations where diagnosticity does not play a large role. Combinations which are generally classified as relation-based do not seem to rely upon diagnosticity. We do not claim however, that Constraint Theory cannot deal with relation-based interpretations e.g. it can interpret 'finger cup' as >> a cup for washing fingers in <<. We do wish to suggest that diagnosticity may not play a large role in relation-based interpretations. Consider an interpretation of 'robin snake' which may be >>a snake which hunts robins <<. Here, the interpretation suggests that there is a third concept, *HUNTING*, which links the two in the combination. Both the head and modifier appear unchanged, so they will satisfy the diagnosticity constraint. Yet we still have to explain where the third concept comes from and how the head and modifier can both satisfy the concept, *HUNTING*.

In the CT3 model new information is given by the plausibility constraint, where new predicates may be added. Indeed, the first two constraints are the generative mechanism of this model; they are responsible for the creation of interpretations. The addition of new predicates to an interpretation is based on finding similar concepts to a partial interpretation in the knowledge base. There appears to be no mechanism for judging the appropriateness of this new additional information except in terms of informativeness. Informativeness by itself is not necessarily a good judge of appropriateness. As informativeness is conceived by Constraint Theory it only compares an entire interpretation against the knowledge base and disqualifies those interpretations with redundant information. However, the information added may not be redundant (i.e. already exist in a category or concept) but may not be appropriate. Suppose *BOWL* in Table 3.3 contained some information that something is drunk from a bowl, *Drunk[K, B]* should this be added to 'finger cup'? So that 'finger cup' could mean >> a cup from which fingers are drunk <<. This is a possible interpretation that Constraint Theory could generate and demonstrates that informativeness may not be a good measure of appropriateness.

3.4.3.2 Combinations Constraint Theory Cannot Deal With

Constraint Theory appears to be able to deal with many types of combination. We suggest though, that it cannot account for combinations which are :

(1) Figurative
(2) Which evoke a particular context.

It appears difficult to envisage how Constraint Theory can deal with figurative combinations such as 'drug baron'. Consider this combination in terms of its constituent concepts, *DRUG*, *BARON* and the diagnostic properties of both. The diagnostic properties of a *DRUG* may be: *relief-giver, medication*. While the diagnostic properties of *BARON* may be: *powerful, wealthy*. An interpretation of 'drug baron' as >> a powerful drug dealer << does not seem to rely on the diagnostic properties of *DRUG*.[1] In this interpretation the concept appears to refer to another concept, 'drug dealer'. Constraint Theory is ill-equipped to deal with these types of combination.

Another way of viewing the 'drug baron' combination is to suggest that the concept *DRUG* draws in the context 'drug dealer'. A 'drug baron' is really a 'drug dealer baron'. The CT3 model adds new predicates to an interpretation but does not allow for a concept to refer to another concept. It cannot draw in context in this manner. Combinations in which one concept refers to related context cannot be dealt with by Constraint Theory.

Constraint Theory may also have difficulty with more mundane combinations. This difficulty is a result again of its diagnosticity constraint. Consider the combination 'table leg'. The diagnostic property of *TABLE* is probably that it *has a flat surface*. While the diagnostic properties of *LEGS* may be that it is a *limb used for support*. Note that *LEGS* could already exist

---

[1] This is one possible interpretation of 'drug baron'. Other interpretations are possible, e.g. one reviewer suggested that 'drug baron' could be treated as a >> person of some-but-not-so-much status and wealth (and power) derived from an association with drugs.<< This interpretation does not seem suitable for providing the extensions that are available with our interpretation, e.g. a serf is like user (see Section 5.3.3 for a more detailed interpretation).

within the concept *TABLE*, as *parts* of that concept. Perhaps this combination refers to something that already exists in one of its elements, namely that *LEGS* will already be included in a representation of *TABLE*. This type of interpretation has little to do with the diagnostic values of attributes of the constituent concepts. So it appears that not all combinations will satisfy this constraint of diagnosticity. We call these interpretations referential.

3.4.4 Dual Process Theory

Dual Process Theory (Wisniewski, 1997a, 1997b) is the first theory to apply structural alignment to concept combination. The theory arises from Wisniewski's (1996) examination of interpretation types, and accounts for relational and property-based interpretations as well as hybridizations. It suggests that concept combination involves two distinct processes:

1. scenario-creation
2. comparison/construction.

The first accounts for relation-based interpretations and the second accounts for property-based interpretations. The model has not been implemented and is not as detailed as other models, e.g. Constraint Theory (Costello, 1996; Costello & Keane, 1998; Costello & Keane, 2000).

The pragmatics of concept combination have been the motivation for at least one model of concept combination, Constraint Theory, so it is worth examining Wisniewski's (1997a) view on this subject. He suggests that there are at least two assumptions that both the speaker and listener may make:

1. The combination refers to a new type of category.
2. That the new information conveyed is carried by / marked out by the modifier.

Comparing these with the Constraint Theory, both Dual Process Theory and Constraint Theory agree on assumption 1: concept combination gives rise to new information. The second is not assumed by Constraint Theory. It suggests that there is a focal concept in concept combination. Although in general it may be the head, the head is not always the focal concept. Dual Process Theory places great importance on the modifier.

Dual Process Theory (Wisniewski, 1997a; 1997b) is essentially an extension of the frame based approach with the novel idea that the slot-filling process is carried out by a comparison construction model. Wisniewski (1997a; 1997b) makes the same representational assumptions about concepts as previous models. However, we should note now that no explicit examples of frame representations are ever given in his work, e.g. Wisniewski (1997a; 1997b). This is a big drawback for a frame-based approach and has forced researchers to supply their own representations when analysing Wisniewski's theory, e.g. Costello & Keane (1998).

In Wisniewski's (1997a) augmented schema approach (his term) a schema or frame contains a scenario which corresponds to a verb with information on actions, events, states with various roles. The concept *SOAP* may have the scenario *CLEANING* associated with it. This scenario would be made up of roles such as recipient, agent and instrument. Indeed some combinations appear to involve the form of a nominalized verb e.g. "blood donor".

Generating a relation-linking interpretation involves finding the most plausible scenario. The most plausible scenario is that in which all the concepts can be bound to different roles. The roles in the scenario may have preconditions that limit which concept can fill which role. So in 'truck soap', *TRUCK* is the recipient and *SOAP* is the instrument. Consider this quote: *"The use of scenarios allows the augmented model to explicitly indicate the different roles that the modifier and head noun are playing in an action, event, or state. In contrast, the slot-filling process of schema models derives a relation-linking relation interpretation by finding a role or slot for the modifier to fill, without explicitly indicating the role that the head noun fills"* (p174, Wisniewski 1997a). This is true if slot-filling only allows the direct replacement of an aspect of the head with the modifier but a more elaborate model may allow for "scenario creation" if a mechanism can go beyond the initial constituents.

Slot-filling for Wisniewski (1997a) only works on the constituents, whereas relation-linking allows for associates of concepts to become involved. The term associates is not elaborated on formally. Taking his "truck soap" example an associate is a sub-part of the head concept as this is where the cleaning scenario is found. Yet if related concepts are allowed to enter the process then perhaps slot-filling can account for both property-based interpretations and relation-based interpretations. For example, "truck soap" could be described as involving the filling of slots in the concept *CLEANING*, where *TRUCK* and *SOAP* can both fit slots.

Wisniewski has also suggested that, *"In order to derive a property interpretation, there must be an important difference between the modifier and head concepts that forms the basis of the interpretation"* (1997a, p175). In his view property interpretations are based on differences. Interpreting a "zebra clam" as >> a clam with stripes << suggests that the crucial difference between *CLAM* and *ZEBRA* is *stripes*. This idea is drawn from Wisniewski's view of the computational aspect of concept combination - people create new combinations to mark out new members of a category where the difference between the head category and combination is marked out by the modifier.

The type of difference marked out by property-based interpretations is known as an alignable difference. Consider Table 3.6 which displays frame-based representations of the concepts *CAR* and *BIKE*. In terms of Dual Process Theory a similarity would be a slot with the same slot name and value that occurs in both representations. In Table 3.6, *Is-a: VEHICLE* would be a similarity between *CAR* and *BIKE*. A difference would be a slot (incl. slot name and value) which does not occur in one of the two representations. An alignable difference would be where both representations have the same slot but with different values. For example in Table 3.6 both *CAR* and *BIKE* have parts, represented by the *part* relation. The representation for *CAR* has *DOOR* as a slot value while *BIKE* does not have this value; this would be an alignable difference.

Table 3.6- Representation of *CAR* and *BIKE*

CAR	*BIKE*
Is-a: VEHICLE, AUTOMOBILE *Performs: TRANSPORT* *Parts: DOOR, WHEELS, ENGINE* *Wheels: 4*	*Is-a: VEHICLE,* *Performs: TRANSPORT* *Parts: HANDLEBAR, FRAME, WHEELS,* *Wheels: 2*

The comparison process would involve taking every aspect of one representation and contrasting it against every aspect of the second representation. An aspect in terms of Table 3.6 would be a slot consisting of slot name and value, e.g. *Is-a: VEHICLE* consists of a slot name *Is-a* and a value *VEHICLE*. In comparing *CAR* and *BIKE*, *Is-a: VEHICLE* (an aspect of *CAR*) would be contrasted with every aspect of *BIKE*. Table 3.7 lists the results of a comparison process between *CAR* and *BIKE*. It displays similarities in which slot names and slot values are identical. Alignable differences are in a manner of speaking based on a weaker notion of similarity. For example, both *CAR* and *BIKE* have parts although they differ in the parts they have. Both *CAR* and *BIKE* have wheels but differ in the number.

Table 3.7 - Results of a comparison process between *CAR* and *BIKE*

Similarities		Alignable differences	
CAR	*BIKE*	*CAR*	*BIKE*
Is-a: VEHICLE *Performs: TRANSPORT* *Parts: WHEELS*	*Is-a: VEHICLE* *Performs: TRANSPORT* *Parts: WHEELS*	*Is-a: AUTOMOBILE* *Parts: DOOR* *Parts: DOOR* *Parts: ENGINE* *Parts: ENGINE* *Wheels: 4*	*Is-a: VEHICLE* *Parts: HANDLEBAR* *Parts: FRAME* *Parts: HANDLEBAR* *Parts: FRAME* *Wheels: 2*

The basic assumption at the heart of this approach is that the interpretation of some noun-noun compounds is similar to the interpretation of some nominal metaphors, "implying that similar processes operate in both domains" (Wisniewski, 1997a, p168). However, structural alignment in this model is only used for property-based interpretations; it is not applied to other interpretation types.

There are always multiple differences between concepts (unless they are identical). So if an interpretation is based on difference - which difference is selected, which one is the most appropriate? Wisniewski (1997a) suggests that there are a number of factors which may come into play:

1. Context - information in the context may point to the relevant property.
2. Nouns sometimes refer to salient properties - 'elephant' seems to refer to large in "elephant garlic".
3. Cue validity / category validity - people may pick properties that have a high cue validity. The cue validity of a property is a probabilistic measure that a concept with this property belongs to a certain category. For example a concept with the property *wing* is likely to be (but not necessarily) a *BIRD*.
4. Plausibility - some properties may not make sense when applied to the head, e.g. "fork ball", >> a ball with prongs <<.

Point 2 is interesting as it is not accounted for in Wisniewski's description of the comparison / construction process and seems more reminiscent of Constraint Theory. Also, nouns may refer to salient properties but this does not automatically mean these salient properties form the basis of an interpretation. For example, an 'elephant painting' is not so much >> a large painting << (although, it could be) but more probably >> a painting of an elephant <<.

Wisniewski (1997a) also suggests that if two concepts are highly similar there is a possibility that a hybridization interpretation will occur. The concept *ZEBRA* and *HORSE* are quite similar compared to say, that of *ZEBRA* and *CLAM*. When two concepts are highly similar a subject may generate an interpretation such as >> something that is both a horse and zebra << for 'zebra horse'. This seems a sensible interpretation, esp. when compared to a hybridization interpretation for 'zebra clam', >> something that is both a clam and a zebra <<. However, Dual Process Theory (Wisniewski, 1997a, 1997b) does not provide a clear account of how hybridization is invoked, e.g. is there a threshold of similarity that must be crossed before a hybridization interpretation can be given?

The relationship between the scenario creation mechanism and comparison / construction has not been outlined by us. Concept combination involves these two processes but do these processes compete or do they occur serially? One approach is to argue that scenario creation occurs first - people attempt to find a relation that links the two concepts before making a property-based interpretation. However, if both concepts are similar, they may have the same associated scenario and may fill the exact same role and so would not give rise to relation-based interpretation. This may explain why highly similar concepts do not tend to give rise to relation-based interpretations. But it appears that people often attempt to compare the properties of the concepts first. A person is likely to treat a "mourner dancer" as >> a dancer who is also a mourner << rather than, say, >> a person who performs mourning by dancing <<.

According to Wisniewski (1997a) in the interpretation of a compound, if a property based-interpretation is not generated, the second mechanism, scenario creation, kicks in. How the actual related scenario is found is quite simple. Wisniewski suggests that every concept has a scenario associated with it. However, it appears from the examples that Wisniewski (1997a, 1997b) uses, scenarios that are only associated with the head and that scenarios associated with the modifier are ignored. The interpretation of a compound in terms of scenario creation involves fitting the modifier and head into the roles associated with a scenario. Thus, an interpretation that a 'zebra clam' is >> a clam eaten by zebras << may arise because clams have the scenario eating associated with them. This is the weakest part of the model as it seems to imply that only one scenario is associated with each concept. As there is no mechanism for deciding which is the most appropriate scenarios, there is no hint that there may be competing scenarios.

3.4.4.1 Problems With Dual Process Theory

There are several problems with the Dual Process Theory approach, among these are:

1. A lack of detailed representations for the frames
2. The use of augmented frames
3. Concepts may have more than one scenario
4. Implicit assumptions about polysemy

The first of these relates to Wisniewski's (1997a) description of his model as an extension of the frame-based approach. If Dual Process Theory truly is a frame-based approach then examples of the frames should be given; other theories such as Selective Modification, Concept Specialisation do so. Related to this problem is that it makes it difficult for other researchers to adequately analyse and test his theory.

Although no precise descriptions of frames are given, Wisniewski (1997a) does suggest that they would contain information on scenarios (which he calls augmented schema). These scenarios are activities that may be associated with a concept. We suggest, in theory, that these scenarios could be accounted for by using frames. Since activities can also be represented in terms of frames, these type of representations are not solely for objects. By placing the scenarios directly with the frames the problem of finding the correct scenario is avoided but the representations have been weakened. They are not like the representations other theories have used and Wisniewski (1997a) offers no psychological evidence for adopting these new modified frames.

Another problem with scenarios is that the question of how to deal with multiple scenarios is not answered (nor posed). A concept may easily have several associated scenarios. Given a combination involving a concept with multiple scenarios there may be competition between the related scenarios. Indeed, what if the modifier concept and head concept have related scenarios? For example, "truck soap" as >> soap transported by truck << where the scenario is transportation as opposed to the earlier example where it was >> a soap used to clean trucks <<. The interpretation of combinations appears to allow for scenarios to be drawn from either the head or the modifier. Wisniewski's account suggests that scenarios are associated only with the head.

Overall this model is let down by its lack of detail but it does offer at least one important insight into concept combination. For the first time, the comparison process in concept combination has been dealt with in terms of structural alignment. It suggests that the mechanics of the slot-filling process may be best explained in terms of structural alignment. Other models, e.g. Concept Specialisation (Murphy, 1988), are not clear on how slot-filling takes place.

3.4.4.2 Combinations Dual Process Theory Cannot Deal With

Dual Process Theory (Wisniewski, 1997a; 1997b) can deal with large classes of property-based and relation-based combinations. It cannot however deal with:

(1) Figurative combinations
(2) Referential combinations
(3) Some types of property-based combinations

In some models of metaphor which use alignment (Sapper: Veale, 1995; SME: Falkenhainer, Forbus & Gentner , 1989), the interpretations generated are based on finding multiple points of similarity. In Dual Process Theory, when two concepts are compared, they are compared to find an alignable difference. Alignment in Dual Process Theory is used in quite a confined way. It does not give rise to the types of mapping that are associated with metaphors. This will be dealt with in more detail in the next chapter but for the time being we note that Dual Process Theory applies alignment to solely find alignable differences. This is quite different to how metaphors and analogies are interpreted.

Referential combinations are combinations where one concept refers to a part of another concept. The combination 'library book' can be interpreted in this way. A representation of library may contain information on books, e.g. that a library stores books. A 'library book' may refer to this aspect of library. Both processes in Dual Process Theory cannot account for this type of interpretation as it is not a property-based interpretation nor a relation-based one.

The way in which Dual Process Theory accounts for property-based interpretations is to assume that a property of the modifier is inserted into the head. In some combinations the modifier itself (and not just a property) can be inserted into the head. For example, in 'fur coat' which could be interpreted as >> a coat made from fur <<, *FUR* could replace the default values of what a *COAT* is typically made from. Here, the entire modifier concept is inserted. The comparison / construction part of Dual Process Theory does not account for combinations such as 'fur coat'.

3.4.5 CARIN Model

The CARIN model (Gagné & Shoben, 1997) views concept combination as the process of finding the appropriate relation between the modifier and the head. The acronym stands for: competition among relations in nominals. For the CARIN model, the combination 'mountain stream' would be interpreted in terms of a location relation, >>a stream located in a mountain<<. This approach proposes a limited number of relations that all combinations will fall into.

The complete list of relations is shown in Figure 3.6. Gagné and Shoben (1997) claim that these relations have been picked to cover the largest amounts of interpretation possible. The first fourteen are drawn from Gagné and Shoben (1997) the fifteenth relation is taken from Gagné (2001) and does not appear in the earlier work. They suggest that each nominal has a set of relations associated with it when acting as a modifier. Given a particular nominal, each native speaker has knowledge about the frequency of these relations. When presented with a novel combination they can choose the most frequently occurring relation first.

1. Noun causes modifier – 'flu virus'
2. modifier causes noun – 'college headache'
3. noun has modifier – 'picture book'
4. modifier has noun – 'lemon peel'
5. noun makes modifier – 'milk cow'
6. noun made of modifier – 'chocolate bird'
7. noun for modifier – 'cooking toy'
8. modifier is noun – 'dessert food'
9. noun uses modifier – 'gas antiques'
10. noun about modifier – 'mountain magazine'
11. noun located modifier – 'mountain cloud'
12. noun used by modifier – 'servant language'
13. modifier located noun – 'murder town'
14. noun derived from modifier – 'oil money'
15. noun like modifier - 'beehive hairdo'

Figure 3.6 – Relations in the CARIN model

As two models of concept combination Dual Process Theory (Wisniewski, 1997a, 1997b) and Constraint Theory (Costello, 1996; Costello & Keane, 1998; Costello & Keane, 2000) agree largely on the types of interpretation people generate it is worth comparing the relations against these interpretation types. In Dual Process Theory and Constraint Theory there are two main types of interpretation: property-based and relation-based. In a real sense all the interpretations that the CARIN model is concerned with are relation-based interpretations. A simple question that can be posed of the CARIN model and any approach that relies on relations is: can it deal with property-based interpretations? The clear answer is no. Consider the compound 'bumblebee moth' and the interpretation >> a moth with black and yellow stripes <<. In this interpretation a property of *BUMBLEBEE* is applied to a *MOTH*. Figure 3.7 lists all possible interpretations with respect to the relations in Figure 3.6.

1. moth *causes* bumblebee
2. bumblebee *causes* moth
3. moth *has* bumblebee
4. bumblebee *has* moth
5. moth *makes* bumblebee
6. moth *made of* bumblebee
7. moth *for* bumblebee
8. bumblebee *is* moth
9. moth *uses* bumblebee
10. moth *about* bumblebee
11. moth *located* bumblebee
12. moth *used by* bumblebee
13. bumblebee *located* moth
14. moth *derived from* bumblebee
15. moth *like* bumblebee

Figure 3.7 – Possible interpretations for 'bumblebee moth' using thematic relations

The closest interpretation to >> a moth with black and yellow stripes << in Figure 3.7 is 15. Some of the others may be understandable but they do not give the property-based interpretation we are searching for. For example, perhaps 11 could be >> a moth that lives in the same habitat as the bumblebee <<. Indeed, 15 even seems a rather poor interpretation. To describe a 'bumblebee moth' as >> moth *like* bumblebee (a moth that is like a bumblebee) << is hardly a description at all. This particular interpretation does not say in what respect the two are alike.

The reason for the failure of a relation-based approach to deal with property-based interpretations is that these interpretations involve concepts associated with the compound. In interpreting a 'bumblebee moth' as a >> a moth with black and yellow stripes <<, the concept *BUMBLEBEE* is used to refer to a property of itself, namely having *black and yellow stripes*. The relation-based approach (as taken from Figure 3.6) can only deal with the actual constituents and not other concepts related to one of the constituents.

Accepting that a thematic relation model will have difficulty in accounting for property-based interpretations we now examine the CARIN model in more detail. The range of the relations and the different ways in which they interact with the head and modifier point to the complexity of concept combination. Gagné and Shoben (1997) propose that subjects actually have information on the way modifiers generally act, e.g. when *MOUNTAIN* is used as a modifier in a combination it is generally in association with *located*. This relation can be used to interpret combinations such as 'mountain stream', 'mountain goat'.

Choosing the most appropriate relation is the central problem in the CARIN model. The relation chosen for an interpretation depends on the strength of a candidate relation against the weight of the alternatives. In other words, the relations compete against each other to be selected (hence the acronym, CARIN). The method for selecting the relation can be Luce's choice rule (Luce & Raffia (1957), as cited in Gagné and Shoben (1997)). This is not the only possible way but is a relatively straightforward one. Taking an example directly from Gagné and Shoben (1997), consider the two concepts: *MOUNTAIN* and *JUVENILE*, and what thematic relations may be associated with them. Table 3.8 lists the thematic relations associated with each concept. Note that the relations differ from those listed in Table 3.6, e.g. there are actually two *has* relations but only one is presented in Table 3.8. This is not a deliberate mistake by the present author but how Gagné and Shoben (1997) name the relations associated with *MOUNTAIN* and *JUVENILE*. The percentages in Table 3.8 do not add up to 100%, we assume that the missing values are associated with other relations.

Table 3.8 - Relation scores for mountain and juvenile

Mountain	**Juvenile**
Locative - 82%	For - 34%
About - 10%	Has - 20%
Uses - 2%	About - 15%
Made of - 1%	Is - 10%

When interpreting a combination, each thematic relation is measured against the others using the Luce choice rule. To work out the score of *locative* against *about* for *MOUNTAIN* see Table 3.9. The difference between the *locative* relation and *about* relation is quite large, especially when compared to the situation with the concept *JUVENILE*. The difference between the scores between the *has* and *for* relations is a lot smaller in the concept *JUVENILE*. The Luce choice rule can be described as follows (Wills, Suret and McLaren, 2000):

$$P(i) = \frac{v_i}{\sum_{j=1}^{n} v_j}$$

P(i) is the probability of choosing a relation *i* from *n* alternative relations and v_j is relation magnitude term for the *jth* alternative.

Table 3.9 - Calculating competing thematic relations

Mountain:	
Locative	0.82/(0.82 + 0.1 + 0.02 + 0.01) = 0.86
About	0.1/(0.82 + 0.1 + 0.02 + 0.01) = 0.11
Juvenile:	
For	0.34/(0.34 + 0.2 + 0.15 + 10) = 0.43
Has	0.20/(0.34 + 0.2 + 0.15 + 10) = 0.25

Essentially, Gagné and Shoben (1997) are attempting to explain ease of interpretation but the central point of concept combination seems to be interpretation generation. Strangely this theory seems to be lacking in this area. Yes, the interpretation of a combination involves selecting the correct relation. A sympathetic reading would suggest that the strongest relation is chosen first but we still do not know if it is appropriate. CARIN is not currently a model of concept combination. It is concerned with ease of interpretation over the actual generation of the interpretation. It could be adapted to be a model of concept combination but this would involve dealing with plausibility.

More fundamentally, thematic relations may not give the most appropriate interpretations. Consider, 'picture book' in Figure 3.6, which is interpreted as >> a book that has pictures <<. But this is not what a 'picture book' is. For example a children's version of a classic novel may have pictures but it is not a picture book. A 'picture book' is >> a book where the usual text is largely replaced with pictures <<.

3.4.5.1 The CARIN Model And Figurative Combinations

The CARIN model can deal with figurative combinations (Gagné, 2001), primarily by positing the addition of a similarity relation signifying that 'modifier is like noun'. For example 'beehive hairdo' can be interpreted in terms of relation (15) in Figure 3.6, >> a hairdo that's like a beehive <<. However, these types of interpretations raise serious questions. In metaphor research, interpretations involve more than the finding of a relation. For example, consider the metaphor "John is a fox". Is it enough to suggest that this means that >>John is like a fox in some sense<<? Clearly interpretations such as these are just too vague to stand on their own as interpretations; they require further elaboration. It is this elaboration which bears the brunt of giving real meaning to the interpretation.

The vagueness of the interpretations is not just a problem with how the CARIN model deals with figurative combinations, literal combinations are also given vague interpretations. Treating a combination such as 'bumblebee moth' as >> a moth like a bumblebee << is not an adequate interpretation. In terms of the relations in Figure 3.6, 'prize fighter' could be interpreted as >> a fighter *for* prize <<. This interpretation still needs some work to be fully understood. Another problem for an approach with a fixed set of relations is that all interpretations must be created in relation to this set. A slot-filling based approach is limited to the names of the slots and the fillers, however these need not be drawn from a fixed set and may allow more creativity in the generation of an interpretation.

3.4.5.2 Problems With The CARIN model

Ultimately, the CARIN model is not really a model of concept combination but more a model of ease of interpretation. To qualify as a model for concept combination there would have to be a number of additions to the model, such as:

1. Explaining plausibility
2. Dealing with concepts outside of the immediate constituents of the compound

It may well be the case that subjects have knowledge on past combinations and may apply this knowledge during the combination process. But knowledge of relation frequency is only part of the explanation. A thematic-relation based approach should also explain why certain thematic relations are appropriate, especially where more than one is possible. The CARIN model does not deal with the plausibility of its relations.

As the model currently stands, it also has difficulty in accounting for combinations where concepts other than the constituents are involved. This is often the case with property-based interpretations. For Constraint Theory the diagnostic features of the modifier are brought into play. In Dual Process Theory other concepts are involved in the process when there are alignable differences. This lack of concepts outside the constituent concepts seriously hampers the CARIN model as an explanation of concept combination.

3.4.5.3 Combinations CARIN Cannot Deal With

The CARIN model can deal with most combinations. However, the problem here is with how it deals with these combinations. The manner in which figurative combinations and some literal combinations are dealt with invites further elaboration. It could be argued that this further elaboration is post the concept combination process but this is not something assumed by the other models which attempt to find the most informative interpretations immediately. Saying a 'beehive hairdo' is >> a hairdo like a beehive << is true but hardly informative.

3.5 The Need For A New Theory?

The five theories outlined above do share some similarities as well as a great deal of differences. Of the five theories, four rely on frame-based representations with the CARIN model (Gagné & Shoben, 1997) being the only one that does not. The theories also range in the amount of detail with which they describe the process of concept combination. The five theories of concept combination can also be compared and contrasted as to how they deal with the three problems of combination outlined in Chapter 1, namely: polysemy, world knowledge, and figurative interpretations. Table 3.10 shows these comparisons. The table shows that none of the current models deal with all three problems simultaneously. Where a question mark occurs in the table, it is the opinion of the present author that a problem could be dealt with in some way by the model, even the if the original theories have not dealt explicitly with this problem.

Table 3.10 - Comparison of models and theories

Models	Problem 1: Polysemy	Problem 2: World knowledge	Problem 3: Figurative combinations
Selective Modification	No	No	No
Concept Specialization	No	Yes	No
Constraint Theory	Yes	Yes	No
CARIN model	?	No	Yes
Dual Process Theory	?	Yes	No

Accounting for world knowledge was first suggested by Murphy (1988). Combinations can draw in outside information. Our review suggests that this problem has yet to be successfully tackled. Murphy (1988) does not specify how exactly world knowledge plays a role. Constraint Theory (Costello, 1996; Costello & Keane, 1998; Costello & Keane, 2000) can draw in outside information but cannot account for combinations where one of the constituent concepts appears to act as a reference to another concept. Dual Process Theory (Wisniewski, 1997a, 1997b) does allow for some world knowledge to be brought in terms of a scenario in which the components of the combination may be involved. Yet this is still quite limited. However, Dual Process Theory and Concept Specialisation and Constraint Theory are marked in Table 3.10 as attempting to deal with world knowledge.

The problem of polysemy is only dealt with explicitly by one model, Constraint Theory (Costello, 1996; Costello & Keane, 1998; Costello & Keane, 2000). In theory a number of the current theories, however, could deal with polysemy. Dual Process Theory (Wisniewski, 1997a, 1997b) can give rise to several interpretations of the combination 'robin snake'. In the CARIN model (Gagné & Shoben, 1997) polysemy can be seen to result from more than one relation being involved in a combination. These theories would need some modification to account for polysemy more fully. There is no mechanism for ranking combinations in Dual Process Theory. The Luce choice rule could be used to rank polysemous combinations in the CARIN. Constraint Theory does apply constraints to decide on the rankings of possible interpretations. In Table 3.10 only Constraint Theory is marked as dealing with polysemy, while Dual Process Theory and CARIN model are marked with a question mark.

Only one model deals with figurative interpretations, the extended CARIN model. To do so Gagné (2001) has had to modify the number of relations that the original model proposed (i.e. relation 15 was added to Figure 3.6). The types of interpretation this model generates are meagre. These interpretations necessitate some form of elaboration. This problem affects the CARIN model's ability to deal with world knowledge as well. Consider the combination 'drug baron': this is a figurative combination but it does not involve the *like* relation in the sense that a 'drug baron' is not >>a baron that's like a drug in some way<<. It appears that *DRUG* draws in the context of 'drug dealer', so the interpretation >>a drug dealer that's like a baron in some way<< seems better. The CARIN model does not allow for the drawing in of world knowledge, in this case 'drug dealer'. Even if it could come up with the interpretation, >>a baron that's like a drug dealer in some way<<, it would still need further elaboration.

This suggests that a new model is needed, one that attempts to deal simultaneously with all three problems. If concept combination can give rise to interpretations that draw in world knowledge (e.g. 'drug dealer' in 'drug baron'), then a model of concept combination should account for this. Also, if concept combination can give rise to multiple interpretations, then again a model of concept combination should account for this. Finally, if concept combination can give rise to interpretations that are figurative, then a model of concept combination should certainly account for this. A model that attempts to do this is presented in Chapter 5.

3.6 Borrowing From The Existing Models

Although our review of the theories of concept combination may have been largely critical we see a number of important points in each of these theories. The model we offer in Chapter 5 is based on structural alignment which is a model of a comparison process. Without getting into the minutiae of structural alignment (which are dealt with in the next chapter) we will briefly outline here why the comparison process is important for concept combination.

A comparison process already exists implicitly in a number of the models. Slot filling, e.g., seems to involve a comparison process. The calculation of diagnosticity also seems to involve a comparison process, especially as described by Constraint Theory (Costello, 1996; Costello & Keane, 1998; Costello & Keane, 2000). The plausibility constraint of Constraint Theory involves a comparison process where partial interpretations are compared with existing combinations. With the exception of the CARIN model (Gagné and Shoben, 1997), all the models rely on a comparison process in some way. Of course, the comparison process does not have to be dealt with in terms of structural alignment but we will show that it can be effective in modelling concept combination in the next chapter and especially in Chapter 5.

Another way to view the model that we will propose is to see it as partially borrowing from the existing models. The elements of previous models that are borrowed are the following:

(1) Use a knowledge base for representations
(2) Use frame-based representations
(3) Apply structural alignment

Constraint Theory (Costello, 1996; Costello & Keane, 1998; Costello & Keane, 2000) makes use of a large knowledge base to derive interpretations. There are several reasons for using a large knowledge base, e.g. in Constraint Theory it will allow for the calculation of diagnostic scores. From our viewpoint, an extensive knowledge base can allow for the introduction of related concepts. It is this what Murphy (1988) alludes to as world knowledge. Wisniewski (1997a) also maintains that concepts have associates, namely scenarios. These associates are part of the representations of concepts, so perhaps 'associates' is a misnomer. The use of a knowledge base allows us to suggest that where combinations involve outside knowledge this information can be found in the knowledge base.

In our model, the representation of knowledge that will exist in the knowledge base will be frame-based. The model of concept combination we will propose in Chapter 5 could be described loosely as a slot-filling one, following in the footsteps of Selective Modification and Concept Specialisation. The process of slot-filling will be carried out using structural alignment. While the application of structural alignment to concept combination could be said to be borrowed from Wisniewski (1997a), we envisage it as a more fundamental process. In reality, Dual Process Theory only deals with property-based interpretations in terms of structural alignment. Furthermore, the actual process of structural alignment is not well specified (if at all) by Dual Process Theory. Even within Dual Process Theory there are several ways it could be implemented. In contrast, our model will be specific in its implementation.

3.7 Chapter Summary

In this chapter the current theories of concept combination were reviewed. The theories varied considerably in the detail with which each examines concept combination. Constraint Theory is perhaps the most detailed and formally explicit. Dual Process Theory is only informally represented in the available literature. The theories were examined in chronological order. The first theory, Selective Modification, is not suitable for noun-noun compounds but can deal with some adjective noun compounds. Concept Specialisation points to a number of interesting aspects of concept combination, e.g. the use of world knowledge, but is not explicit enough in how world knowledge is used. Constraint Theory sees the combination process as governed by three constraints, although only two generate the actual interpretations. This model may fail to account for a number of compounds due to its reliance on the diagnosticity constraint. The CARIN accounts for ease of interpretation but not concept combination.

It was also pointed out that most of the theories are successful in relation to one or two of the problems arising in the interpretation of compounds outlined in Chapter 1, but not to all three. No theory can deal simultaneously with the role of polysemy, world knowledge and figurative interpretation. Constraint Theory can deal with polysemy. Constraint Theory, Concept Specialisation and Dual Process Theory attempt to deal with the role of world knowledge. A number of the models, e.g. Constraint Theory, can possibly be extended to deal with all three problems, but how this extension is to occur is not completely clear. The lack of a model that deals with all three problems arising in the interpretation of combinations suggests that each is, in an important respect, incomplete, and that a new, more comprehensive model is required. Such a model is presented in Chapter 5. As this model is based on structural alignment, our approach to this comparison process is outlined in the next chapter.

Chapter 4 - Structural Alignment

4.1 Introduction

This chapter examines structural alignment (SA), which models a comparison process. Initially, the area of similarity is discussed. Simple feature-based models do not account for some of the phenomena in similarity judgements. Subjects seem to use structural information as well as simple features in similarity judgements. Structural similarity is discussed in relation to analogy and examples of how subjects take account of structural information are given. The principles of structural alignment, one-to-one mapping, parallel connectivity, and a systematicity principle are then set out. With these principles in mind, rules for mapping are discussed. Different approaches (e.g. Veale, 1995; Falkenhainer, Forbus & Gentner, 1989) apply different mapping rules but still adhere to the principles of structural alignment.

Following our introduction to structural alignment we present the view of structural alignment adopted in this present dissertation. This view, which is based on the work of Veale (1995), suggests that the comparison process gives rise to two types of mapping: shallow similarity and deep similarity. To conclude this chapter, arguments against applying structural alignment to concept combination are examined. It has been suggested that SA should not be applied to concept combination (Keane and Costello, 2001) but we argue that the evidence does not support this view.

4.2 Similarity

We cannot talk about a comparison process without mentioning similarity and its related topics, such as discrimination. To compare and contrast two things is to both look for similarities (a similarity process) and differences (a discrimination process). As mentioned in Chapter 3, some researchers suggest that finding a difference between the head and modifier may be crucial in understanding some types of combination, especially in property-based interpretations.

Similarity however, is a rather empty notion, as is illustrated from this quote by William James: *"The moon is similar to a gas-jet, it is also similar to a football; but a gas-jet and a football are not similar to each other. When we affirm the similarity of two compound things, we should always say in what respect it obtains. Moon and gas-jet are similar in respect of luminosity, and nothing else; moon and foot-ball in respect of rotundity, and nothing else... Similarity, in compounds, is partial identity".* (p579, James, 1890/1995). James suggests that when comparing one object with another, e.g. a gas-jet and a football, it appears we do so with respect to the properties that these objects have. It is generally these properties which make the entities similar.

A further problem is that any two objects in principle will share some common properties. For example, let us say an *ACTRESS* is like a *HORSE*. They both exist, both are physical, both eat, both inhabit the earth. The other side to this is that the number of differences between the two entities is also enormous. One is a human, the other is not, one performs on the stage while the other typically does not, one generally lives in a stable the other does not. Yet subjects can quickly find similarities and differences between concepts.

One of the most widespread models of similarity is the contrast model of Tversky (1977). In short, when comparing an object A and an object B the similarity of one to the other is a function of the features they share minus both the features that A has but B does not and the features that B has but A does not. (The model allows for particular weightings to be included). This is a set based approach to similarity. Similarities are the intersections of sets, e.g. the red intersections in Figure 4.1.

The contrast model makes at least one prediction on similarity, namely that the more different two objects are, the easier it should be to find differences. Consider the dissimilar pair in Figure 4.1 below, in these two sets anything that is not in the intersection is a difference. The same is true of the similar pair but clearly the former has more differences. The intersections in Figure 4.1 are highlighted in red, the greater the area of this red zone the greater the number of similarities.

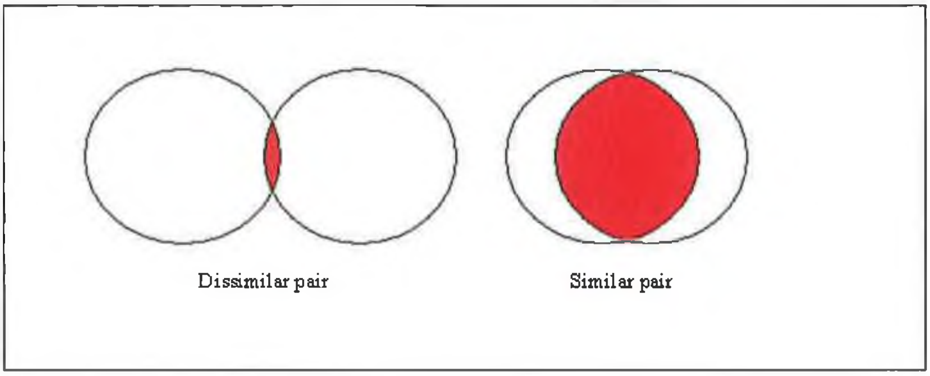


Figure 4.1 - Similarity as the Intersection of Sets

Recent research, however, has suggested that the more similar two objects are, the more differences subjects can report between them (Markman & Gentner, 1993). This may seem counter-intuitive and is actually counter to the contrast model (Tversky, 1977). In Figure 4.1 there are clearly more differences between the dissimilar pair and so it should give rise to more reported differences. There should also be less differences to report for the similar pairs as there are much less differences between them. This has not been found to be the case in empirical studies. Markman and Wisniewski (1997) examined subjects' ability to list the differences between various objects. Interestingly, the more similar two items were the more differences the subjects were able to list. This finding does not fit the contrast model. Markman and Wisniewski (1997) suggested that the type of differences subjects return are alignable differences (see Section 3.3.4). This type of difference is based on similarity and so the more similar two items are, the more likely they are to have alignable differences.

4.2.1 Similarity And Structure

Similarity is of great interest to researchers on metaphor and analogy. Lakoff and Johnson (1980), in their influential work, suggested that metaphors occur frequently in everyday language. Metaphors involve finding similarities between seemingly dissimilar entities. For example, speakers of English often describe the concept *ARGUMENT* in terms of the concept *WAR*. They speak of arguments as if they were literally wars. This can be seen as a generative metaphor schema: "argument is war". Below are some utterances that rely on this metaphor (adapted from: Lakoff & Johnson, 1980, p4)

1. *"He attacked every weak point in my argument".*
2. *"Your criticisms are right on target".*
3. *"If you use that strategy, you'll get wiped out".*
4. *"He shot down all of my arguments".*

In the metaphor "argument is war", there are a number of similarities between the two concepts. Firstly though we should note that arguments have points, but these points may be called arguments themselves. These points are positions that can be defended, attacked or abandoned. The opposite sides of the argument are opposing forces. Arguments can be the weaponry that attacks, as in (2) above. Arguments can be physical positions that are defended and attacked. They can also be weaponry that is attacked, (4). This suggests that the subparts of one domain, e.g. points in an argument, are similar in some way to the positions armies hold in a war.

Early models of metaphor suggested that metaphors can be understood in terms of simple feature-matching. For example, Ortony (1979) proposes a Salience Imbalance approach to metaphor. The problem with models such as Ortony's is that concepts may in fact be best represented in terms of structure rather than a simple set of features (simple in that these features may be unconnected with one another). This is a problem, as it appears that phenomena such as metaphor and analogy involve similarities based on structure. The similarities are not just simple feature commonalities but involve complex groups of interconnected features.

Lakoff and Johnson (1980) argued that metaphors such as "argument is war" can be understood in terms of cross domain mappings. In short, one domain is described in terms of another and in doing so a hearer must place both domains in correspondence to see what part of what domain maps onto another. The work of Gentner (e.g. Markman & Gentner, 2000; Bowdle & Gentner, 1999; Gentner 1998; Ferguson, Forbus, & Gentner, 1997; Gentner & Markman, 1997) on analogy offers a rich model of how cross-domain mapping occurs. These cross-domain mappings can be viewed as points of similarity.

4.2.2 Analogy and Structural Alignment

Markman and Gentner claim that similarity and analogy can be modelled in terms of the alignment and mapping of structured representations (Markman & Gentner, 2000). These structured representations *"...consist of hierarchical systems that encode objects, attributes of objects, relations between objects, and relations between relations"*, (p152, Gentner & Markman, 1997). To understand how a comparison process works we present an example from Markman & Gentner (1993). The example is based around a question: why is Jupiter like Mars? Consider the representation in Table 4.1. Both concepts are alike because apart from sharing common features (which are not shown) they share the same place in a similar relational structure. Viewing (1) and (2) in Table 4.1 *JUPITER* and *MARS* fill the same roles. This is one of the central insights of structure mapping: similarity can be based in structure (structural similarity) and not just feature matching.

Table 4.1 - Why is Jupiter like Mars?

1. cause(greater(mass(Sun), mass(Jupiter)), revolve(Jupiter, Sun))
2. cause(greater(mass(Sun), mass(Mars)), revolve(Mars, Sun))

The importance of the relational structure can be seen using a further example where one object occurs twice but in different roles, see Table 4.2. Consider the situation where we ask what is *JUPITER* similar to? If we focus solely on attribute relations then in Table 2 then we would say that *JUPITER* maps onto *JUPITER* (as they are identical). However, it should be clear that *JUPITER* maps onto the *SUN* as they both occupy the same place (role) in the overall structure. Similarity, then, is a question of relational structure and not just simple features.

Table 4.2 - Comparing the relationships between the Sun and Jupiter, and Jupiter and Europa

1. cause(greater(mass(Sun), mass(Jupiter)), revolve(Jupiter, Sun))
2. cause(greater(mass(Jupiter), mass(Europa)), revolve(Europa, Jupiter))

The similarities found between different domains, e.g. *JUPITER* and *MARS*, can be described as mappings between the two domains. The term mapping is used to suggest some sort of conceptual connection between aspects of the two domains and these mappings are similarities.

4.2.3 Principles Of Structure Alignment

When performing matches between two domains there should be a one-to-one mapping. A feature should not be mapped onto several features in another domain. For example, when asked to find comparisons between the Gulf War and World War Two (WW2), the types of mappings subjects give can be predicted (Spellman & Holyoak, 1992). Subjects do not generally break from one-to-one mappings. Consider the four questions in Table 4.3. These questions asks a subject to draw an analogy between the Gulf War and WW2 and then to make some specific mappings. Taking the questions in this table, a one-to-one mapping suggests that subjects should only give one answer to these questions. For example, taking question 4, subjects are unlikely to say the *CHURCHILL* is like *GEORGE BUSH* and *JOHN MAJOR*. They are likely to only take one of these persons to be like *CHURCHILL*.

Depending on a subject's political views a person is likely to suggest that perhaps *SADDAM HUSSEIN* is like *HITLER* as answer to Q1.[1] If this answer is given then the answer to question 2 should be *IRAQ*. Although, *GERMANY* did take part in the Gulf War it seems that by saying *SADDAM HUSSEIN* is like *HITLER* we have to say that *IRAQ* is like *GERMANY*. This is likely because of the relationships between these leaders and their countries. For example, Figure 4.2 gives a simplistic representation of the concepts *HUSSEIN* and *HITLER*. Both are leaders but more importantly the relationship between these leaders and their countries is similar. It is this relationship that leads subjects to suggest that if *HITLER* is like *HUSSEIN* then *GERMANY* is like

[1] The English version of his full name is "Saddam Hussein al-Majd al-Tikriti", with "al-Majd" being the equivalent of a surname. We have chosen to refer to him in our examples as Hussein.

IRAQ. This finding suggests that when a mapping is made, subjects take note of structure to find further mappings.

Table 4.3 - Points of comparison between WWII and the Gulf War

1.	If the Gulf War is like WW2, who is Hitler?
2.	If the Gulf War is like WW2, who are Germany?
3.	If the Gulf War is like WW2, who is Churchill?
4.	If George Bush is like Churchill, then who is like FDR?

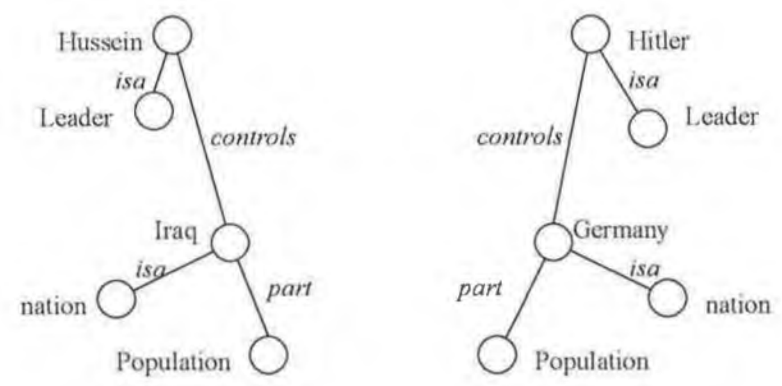


Figure 4.2 - Simple representations of Hitler and Hussein

Furthermore, subjects exhibit a preference for similarities that involve more related structure to those that do not. This may reflect the fact that similarities based on related structure will be richer (carry more information) than simple feature based ones. Where there is a choice of interpretations, subjects prefer the one that preserves the largest connected relational structure (Gentner, Rattermann & Forbus, 1993). This is known as the systematicity principle.

Table 4.4 - Principles of Structural Alignment

1. One-to-one mapping
2. Parallel connectivity
3. Systematicity principle

The basic principles of SA are outlined in Table 4.4 (Gentner & Markman, 1997). Thus far only 1 and 3 have been discussed. The other principle, parallel connectivity, is related to the principle of a one-to-one mapping, as they both ensure structural consistency. Parallel connectivity suggests that when two predicates are matched their arguments should be matched as well. The crux of parallel connectivity is an attempt to define what further mappings should be made with respect to initial mappings. For example, in Figure 4.2 if *IRAQ* is mapped onto *GERMANY* then *HUSSEIN* should be mapped onto *HITLER*. How this parallel connectivity is actually worked out depends on the mapping rules adopted. In the next section two views of parallel connectivity are examined.

4.3 Rules For Mapping

In general, SA can be viewed as involving mappings between domains. These domains can be represented as graphs. SA then becomes a process whereby graphs and sub-graphs are placed into correspondence. The main problem in SA, from this perspective, is to find out what parts of each graph should be placed in correspondence, Figure 4.3. This includes what parts should be initially mapped and how parallel connectivity should be organised.

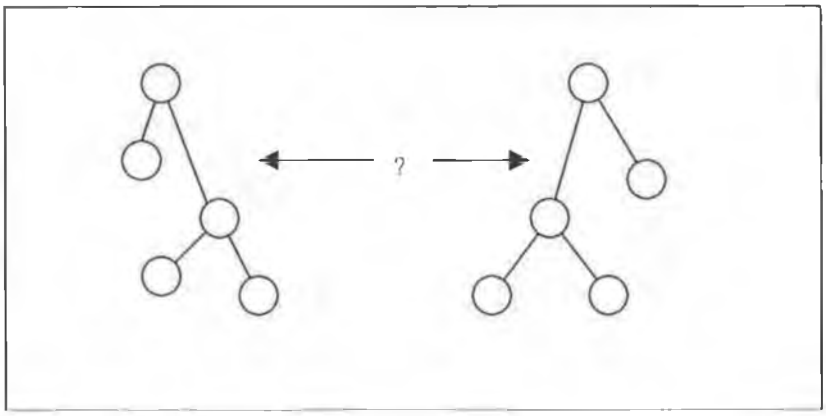


Figure 4.3 - Main problem of structural alignment

The short answer to the question - what is mapped in SA? - is that it is the similarities between the two domains that gets mapped. We will briefly look at how two different models have suggested how mappings occur. Both have been implemented as computer programs. The first is based on the work of Gentner (e.g. Markman & Gentner, 2000; Bowdle & Gentner, 1999; Gentner 1998; Ferguson, Forbus, & Gentner, 1997; Gentner & Markman, 1997), the second is based on a broader conception of cognition and how central SA is to human thinking (Veale, 1995). Both attempt to account for metaphor and analogy, where an interpretation is the largest isomorphic sub-graph found. However, we are not so much interested in how a final interpretation is created but with how each model initially finds mappings. We will only examine the processes that find the initial mappings and we will not delve into procedures for cleaning mappings, i.e. the removal of irrelevant mappings, or how an actual interpretation is generated for a metaphor or analogy.

4.3.1 SME (The Structure Mapping Engine)

SME is short for the structure mapping engine and was devised to account for analogy (Forbus, Ferguson & Gentner, 1994; Falkenhainer, Forbus & Gentner, 1989). The way the alignment works in the structure mapping engine is set out in Table 4.5. Returning to the question of similarity raised in Section 4.1, similarity becomes a structural question. Cross-domain mappings should take place between corresponding sub-structures.

Table 4.5 Alignment Rules

1. Objects in the base and target are placed in correspondence.
2. Relations between objects in the base are mapped across to similar relations in the target.
3. Map the arguments of the relations mapped in 2. Above.
4. Apply the principle of systematicity and gather the largest of mappings

The rules for finding the largest isomorphic subset involve the removal of a number of mappings. Only one-to-one mappings between the initial relations and the arguments of relations are allowed. Mappings, which break with this one-to-one mapping, are discarded. Mappings between objects that are not involved in any other system of relations are also discarded.

The rules become clearer when the representations of the domains is understood. Figure 4.4 shows the representation of two domains: simple water flow and heatflow. The analogy behind these representations is that in simple terms heat flow is like water flow. Note that the domains are represented in terms of objects and more importantly the relations between objects. Objects are referred to as entities. Expressions involve relations between objects and other expressions.

Water flow:	Heat Flow:
(defDescription water-flow entities (water beaker vial pipe) expressions (((flow beaker vial water pipe) :name wflow) ((pressure beaker) :name pressure-beaker) ((pressure vial) :name pressure-vial) ((greater pressure-beaker pressure-vial) :name >pressure) ((greater (diameter beaker) (diameter vial)) :name >diameter) ((cause >pressure wflow) :name cause-flow) ((flat-top water)) ((liquid water))))	(defDescription heat-flow entities (heat coffee ice-cube bar) expressions (((flow coffee ice-cube heat bar) :name hflow) ((temperature coffee) :name temperature-coffee) ((temperature ice-cube) :name temperature-ice-cube) ((greater temperature-coffee temperature-ice-cube) :name >temperature) ((flat-top coffee)) ((liquid coffee))))

Figure 4.4 - Representations of water flow and heat flow

The exact implementation of SME (Forbus et al., 1994; Falkenhainer et al., 1989) adheres broadly to the notions of SA that were outlined in the parent section. To sum up the mapping rules of SME there are three basic steps:

1. Objects in a domain are mapped
2. Relations are mapped
3. The arguments of relations are mapped recursively

In SME all possible mappings, according to the rules (1), (2) and (3) are made. From this large set of mappings a great number are culled or removed. Taking the examples in Figure 4.4 the following mappings would occur. Initially all objects are mapped against each other. So water maps onto heat, beaker, vial etc. Initially there are no one-to-one mappings for objects. Relations with similar predicate names and the same number of arguments are mapped. For example the relations named *wflow* maps onto *hflow* (in Figure 4.4 relations are given specific names or abbreviations), as both are relations with the predicate name *flow* and both have the same arity. The relations *>pressure* and *>diameter* both map onto *>temperature*. The next stage is to map the arguments of relations. This is done recursively, so mapping *>pressure* onto *>temperature* results in mapping *pressure-beaker* onto *temperature-coffee* and *pressure-vial* onto *temperature-ice-cube*. The arguments of these relations are then mapped, this results in beaker being mapped onto coffee and vial being mapped onto ice-cube.

So far these mappings appear to have little in common with the principles of SA which we outlined. There seems to be unconstrained mapping between the two representations in Figure 4.4. However, in the cleaning up stage mappings which break with the principles are removed. We will not go into further details as to how an interpretation is generated. But essentially the largest connected set of mappings is sought. At the start of the mapping process all objects are mapped onto each other. However, when the arguments of relations are mapped then further object mappings are found, e.g. mapping beaker onto coffee. These mappings are selected over ones that are not related to other mappings. When there are two or more conflicting mappings (i.e. there is a one-to-many mapping) the mapping which harmonises with the most other mappings is kept, all the others are removed. Both the principles of systematicity and parallel connectivity are upheld by favouring mappings that involve the largest set of mappings between connected relations.

Mapping or the alignment process initially involves attempting to create all possible mappings between the two representations. Rules are then applied to tidy up these mappings. Obviously, the larger the representations, the larger the number of mappings. In general, all models of structural alignment have to deal with large numbers of mappings.

4.3.2 Sapper

The Sapper model (Veale, 1995) is more than just a model of metaphor; Sapper is based around a parallel network model of memory. The process of SA results in new links being created in the network. The central idea behind using a model of memory is that the cross-domain mappings which SA models actually occur in memory and so a prospective model should reflect this. This in turn makes the approach cognitively plausible. It is also similar to the approach that Constraint Theory (Costello, 1996; Costello & Keane, 1998; Costello & Keane, 2000) takes, where concept combination involves a large knowledge base.

Apart from the fact that cross-domain mappings occur in memory and that this should be modelled, one other crucial point should be noted. Allowing cross-domain mappings to occur across a model of memory allows the researcher to draw in knowledge outside of the concepts being compared but which is still part of the network. This cannot be done with SME (Falkenhainer, Forbus & Gentner, 1989), the initial representations are the sole information that SME works on.

The name Sapper is derived from the fact that the model attempts to recognise cross-domain bridges. These cross-domain bridges are really types of similarity that are used to create interpretations later on. They exist in the network as a part of memory. There are two types of cross-domain bridges: ones resulting from the "triangulation rule", ones resulting from the "squaring rule". Before the rules are outlined the nature of the representations used should be examined. The types of representation that Sapper uses are presented in Figure 4.5, which gives a representation for *SPORTSCAR*. The network will consist of numerous other concepts, e.g. representations of 'car paint', 'car seat', 'combustion chamber'. The whole network is the input that the squaring rules and triangulation rules take.

```
Defconcept(sports_car,
    [[isa, automobile, vehicle, mode_of_transportation],
        [surface, car_paint],
        [part, car_bodywork, car_wheel, car_hood],
        [interior, chassis, car_seat],
        [attr, big, fast, sleek],
        [interior, combustion_chamber, catalytic_converter, engine]]).
```

Figure 4.5 - **Representation** of *SPORTSCAR*

The triangulation rule builds a bridge between concepts that share similar properties. Consider the concept *SCALPEL* and the concept *CLEAVER*. Both are knives, both are sharp and both are instruments. In essence, the representations of concepts are checked to see if the same slot and slot value occur in both concepts. For example, if both representations had the relation *attribute SHARP* then a similarity would be found. This similarity would result in a link being built between the concept *SCALPEL* and the concept *CLEAVER*. In Figure 4.6 this link is marked (1). The squaring rule builds on the links created by the triangulation rule. When a similarity is found between concepts, a link is also placed between parent concepts. A parent concept for *SCALPEL* could be a concept such as *SURGEON*. A parent concept for *SCALPEL* might be *BUTCHER*. A link would be placed in memory between the concepts *SURGEON* and *BUTCHER*. The concept *SURGEON* must link to the concept *SCALPEL* with the same link as *BUTCHER* to *CLEAVER*. For example, both might *control* the respective *KNIFE* concepts. The bridge built by the squaring rule is marked (2) in Figure 4.6.

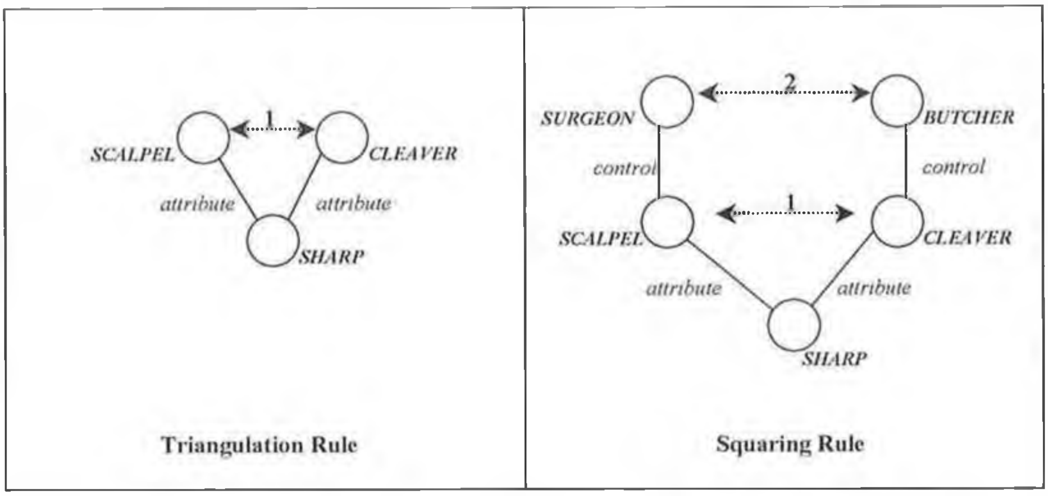


Figure 4.6 - Squaring and triangulation rules in Sapper (Veale, 1995)

Taking the metaphor "A general is a surgeon" as a starting point, we will outline the mapping process of Sapper, although we will not outline a full interpretation for this metaphor. It should also be noted that in Figure 4.7 and 4.7 the representations form part of a larger network and so are incomplete (incomplete in that more information is required for Sapper to succeed). By contrast, in Figure 4.4 the representations are complete, these are all SME requires.

General:	Surgeon:
defconcept(general, [[isa, 0.5, soldier, person], [prototype, george_patten, napoleon], [attr, 0.6, old, 0.7, influential, control, powerful, 0.5, intelligent, educated, military_school, warlike, arrogant, agressive, battlefield], [control, 0.5, army, nerve_gas, atomic_bomb, command_centre, snub_fighter, bomber_plane, soldier], [depend, 0.5, army], [create, 0.5, military_propaganda, plan], [affect, 0.5, army, -0.7, enemy_army, enemy_soldier, 0.8, soldier], [perform, 0.5, bombing_raid, 0.1, cavalry_charge], [part, 0.5, military_uniform]]).	defconcept(surgeon, [[isa, 0.5, profession, doctor, person], [depend, -0.7, cardiac_arrest, 0.5, medical_textbook], [attr, 0.7, intelligent, 0.85, educated, 0.9, precise, 0.6, dextrous, 0.8, skilful, 0.6, influential, 0.75, well_paid, 0.5, operating_theatre, 0.8, delicate, 0.25, blood, 0.9, medical_school], [affect, -0.7, cancer, 0.5, life, 0.05, corpse, 0.8, patient, 0.5, circulatory_system, 0.7, human_flesh], [perform, 0.5, hypocratic_oath, pre_op, 0.9, surgery], [control, 0.5, circulatory_system, radiation_therapy, disinfectant, scalpel], [part, 0.5, surgical_glove, surgical_mask, white_smock]]).

Figure 4.7 - Representation for *GENERAL* and *SURGEON*

Soldier:	Patient:
defconcept(soldier, [[isa, 0.5, element, combatant, person], [part, 0.5, torso, arm, leg, head], [attr, 0.7, healthy, 0.5, military_uniform, combat, expendable]]).	Defconcept(patient, [[isa, 0.5, person], [part, 0.5, torso, arm, leg, head], [attr, 0.7, healthy]]).

Figure 4.8 - Representation for *SOLDIER* and *PATIENT*

We have described two rules for mapping, the triangulation and squaring rules. These are applied to all concepts in the network. These rules note the similarity between concepts and are build what are known as bridges. For this example such a bridge may be built between *SOLDIER* and *PATIENT*. Both share a number of similarities, they *are persons* and share similar parts, e.g. torso, leg. As a result of the triangulation rule a bridge is built between these two concepts. The squaring rule builds bridges between the parents of these concepts. For example, a bridge would be built between *SURGEON* and *GENERAL*. However this would not be the only bridge created, any concept that is one link from either *SOLDIER* and *PATIENT* would also have a bridge built between them. Depending on the size of the network the number of bridges could be huge. For example, if the concept *SURGERY* is linked to *PATIENT* and *WAR* linked to *SOLDIER* then a bridge would be built between *SURGERY* and *WAR*. Considering the metaphor "a general is a surgeon" this may be an interesting mapping. However, other bridges would be built, e.g. perhaps a bridge could be made between the concept *TANK* and the concept *HOSPITAL*.

If structural alignment is a comparison process that gives rise to mappings then the products of the triangulation rules and the squaring rules, the bridges, are the mappings. Of course the actual interpretation of the metaphor will be more complex than we have shown. Particular mappings or bridges will have to be favoured over others. For example, although *SOLDIER* maps onto *PATIENT* it would also map onto *SURGEON* and *DOCTOR*, but these bridges are the ones that Sapper creates. The interpretation of a metaphor will involve the selection of the most appropriate bridge.

4.4 Comparing SME and Sapper

SA was introduced as a process that generates cross-domain mappings or correspondences between domains. This notion of cross-domain mappings is central to the Contemporary Theory of Metaphor (Lakoff, 1993). As SA has been modelled by Sapper (Veale, 1995) and SME (Forbus et al., 1994; Falkenhainer et al., 1989) SA is a comparison process that places conceptual structures in correspondence. Dual Process Theory (Wisniewski, 1997a; Wisniewski, 1997b) has attempted to apply this process to concept combination, albeit in a limited way.

Sapper (Veale, 1995) and SME (Forbus et al., 1994; Falkenhainer et al., 1989) can be examined more abstractly in terms of a process that has inputs and outputs. For example, SME takes two inputs and generates an output. Figure 4.8 shows a very simple representation of this. The main point to note is that the comparison can only access the two input boxes. These boxes would contain information such as that displayed in Figure 4.9, no other knowledge is accessed. Dual Process Theory (Wisniewski, 1997a; Wisniewski, 1997b) treats alignment in the same way. Both the head and modifier are the only inputs and no further information is drawn. Alignment in Dual Process Theory is only involved in property-based interpretations.

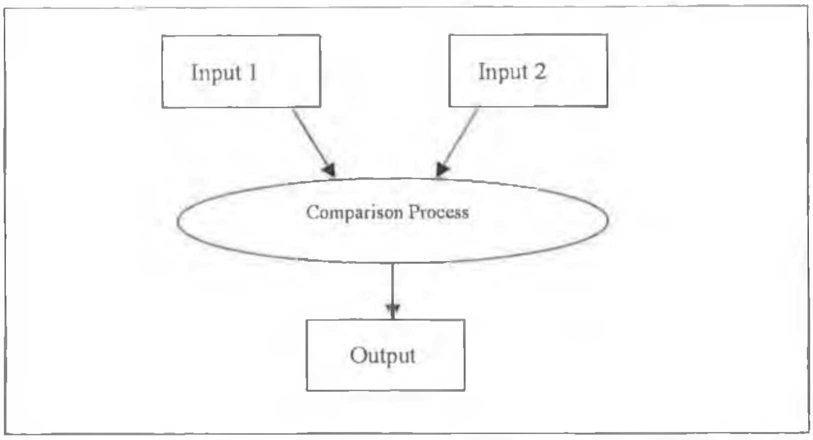


Figure 4.9 - Inputs in SME

Sapper (Veale, 1995) introduces a clever variation on this. In Sapper, SA does involve two inputs but in a sense there is just one input: the model of memory. In Figure 4.10 the large rectangular box with the dotted lines represents the model of memory in terms of a network. The inputs are part of this network. The process of alignment involves activating parts of this network. The actual inputs used depend on what has been activated. Information outside of a usual representation of a concept is found within the range of activation. In general, this leads to the possibility of world knowledge being brought into play.

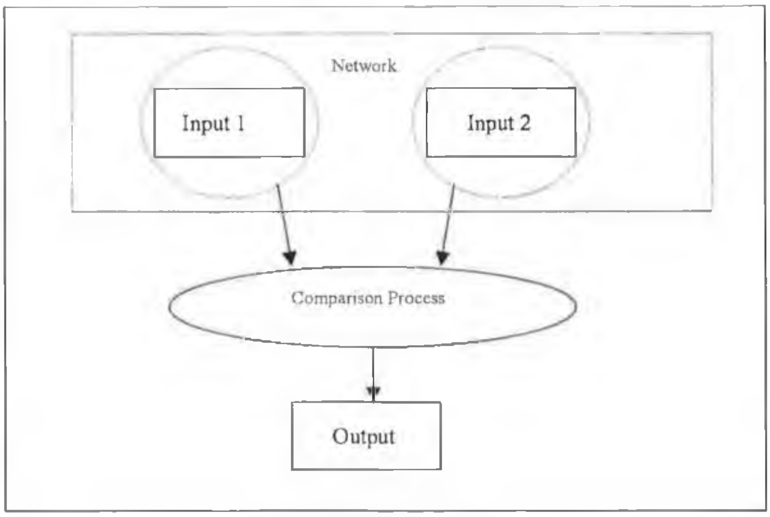


Figure 4.10 - Inputs in Sapper

The type of representation that Sapper (Veale, 1995) uses offers advantages over the other approaches. Using a network model of memory, world knowledge can be brought to bear on the interpretation process. This world knowledge would consist of information that is related to the inputs. This related information can be found by seeing what the inputs are linked to. We suggest that models of SA should adopt this type of representation.

What are to be the representations that will make up the network or knowledge base? Again there is a choice between Sapper (Veale, 1995) and SME (Forbus et al., 1994; Falkenhainer et al., 1989). SME assumes that representations are structured and that they are made up of entities, attributes, functions and relations. These "building blocks" can be defined as follows:

1. Entities - objects (e.g. building-1) .
2. Attributes - representational elements that provide descriptive information (e.g. tall[building-1]) .
3. Functions - map values onto the other "building blocks" (e.g. color(building-1) = gray).
4. Relations - these are representational elements that relate two or more entities, attributes or other relations (e.g. taller[building-1, building-2]).

This is quite different to the types of representations that have been used in accounting for concept combination (e.g. Table 3.1, Figure 3.4). The difference can be highlighted by examining Tables 4.1 and 4.2. The representations here are not representations of the planets themselves; they are representations of relations between these concepts. SME has a tendency to not so much provide a representation of an object but of situations. Objects are only treated within this context of situations. This suggests it may be unsuitable with respect to concept combination. The representations that Sapper uses are similar to the ones used by models of concept combination.

The representations that are chosen also affect the type of similarity rules that are applied. Simply speaking, it is difficult for SME (Forbus et al., 1994; Falkenhainer et al., 1989) and Sapper (Veale, 1995) to have the same similarity rules as the rules reflect the representations. As both have different representations, it is not surprising that they have different similarity rules. SME for example, places little emphasis on what is termed attributes but emphasises similarities based on relations to capture structural similarity. Sapper also emphasises structural similarity but does so using the attributes that SME largely disregards.

As noted in Section 3.5 we have actually already committed ourselves to a type of representation. We have adopted a frame-based approach to representation (following Selective Modification (Smith et al., 1988) and Concept Specialisation (Murphy, 1988)). These frames will be organised into a knowledge base (following Constraint Theory (Costello, 1996; Costello & Keane, 1998; Costello & Keane, 2000)). This is actually the same approach that Sapper adopts to nominal metaphor interpretation. This suggests that the mapping rules of Sapper or something very similar will be suitable for our approach to concept combination.

4.5 Deep and Shallow Similarities

Having settled on what inputs the comparison process takes, a new question arises: what does the comparison process result in? We argue that the comparison process gives rise to two types of similarity: one based on simple features (commonalities) and one based on structural similarity. In Section 4.2, similarity was discussed, as was the contrast model of Tversky (1977), which is based on contrasting the simple features that exist between concepts. However, we also pointed out that in relation to similarity-based comparisons, simple **feature-based** approaches were not sufficient to account for subjects' similarity judgements (e.g. Markman & Wisniewski, 1997). Research on metaphor and analogy, e.g. Markman & Gentner (1993), suggests that subjects can also find **structural similarities** between concepts. Ultimately, subjects can find similarities based on both simple features and structure. We suggest that the comparison process gives rise to at least two types of mapping:

1. Shallow similarities - based on simple features (commonalities)
2. Deep similarities - based on structural similarity

Interestingly, both these types of similarity are captured to some extent by the Sapper model (Veale, 1995). The first type of similarity is one where an aspect of one structure exists in another. In our work it involves the same slot existing in both the contrasted representations. Consider the concepts *CAT* and *DOG*. How are these two concepts similar? One response may be to say that both are *PETS*. To represent that a *CAT* or a *DOG* is a *PET* an *Is-a* slot may be used where the slot value is *PET*. If both representations contained this same slot name and slot value, then a similarity has been found. This corresponds to a simple feature based model of similarity and to the triangulation rule of Sapper (Veale, 1995). We will to this as a shallow similarity and elaborate further on this component of our model at the end of this section.

Structural similarity can be captured in many ways. In Section 4.3, two models, SME (Forbus et al., 1994; Falkenhainer et al., 1989) and Sapper (Veale, 1995), each with different mapping rules were outlined. We favour the approach of Sapper as it already captures similarity based on commonality (via the triangulation rule). The squaring rule of Sapper notes structural similarity between conceptual representations, we refer to this type of similarity as a deep similarity. We will now consider both types of similarity, shallow similarity and deep similarity, in relation to two examples: a comparison between *CAT* and *DOG*, and an extended comparison between *SCALPEL* and *CLEAVER* (first introduced in Section 4.3.2).

In Table 4.6 the concepts *CAT* and *DOG* are represented. These concepts share a large number of commonalities. Apart from both being *PETS*, both *CAT* and *DOG* share similarities in relation to body parts. Overall, this suggests that the two concepts are actually quite similar. The commonalities between the concepts *CAT* and *DOG* are presented visually in Figure 4.11. On the basis of these commonalities we suggest that a shallow similarity exists between the concepts *CAT* and *DOG*.

Table 4.6 - Representation of *CAT* and *DOG*

DOG	*CAT*
Is-a: PET, CANINE *Parts: LEG, HEAD, TORSO, TAIL* *Noise: BARK* *Pesonality: LOYAL*	*Is-a: PET, FELINE* *Parts: LEG, HEAD, TORSO, TAIL* *Noise: MEOUW* *Personality: INDEPENDENT*

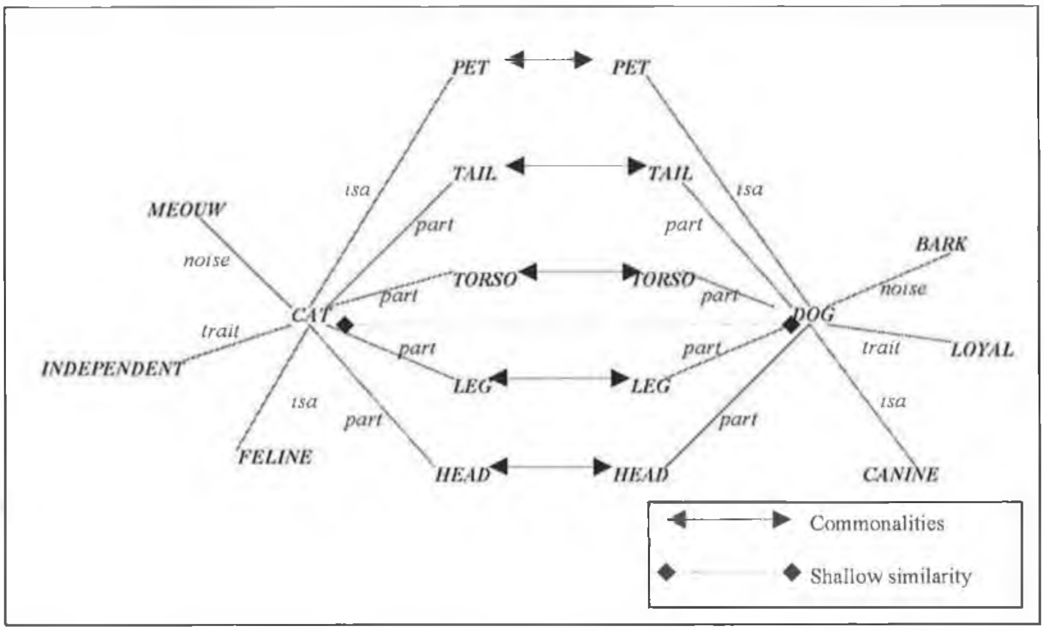


Figure 4.11 - Shallow similarity between *CAT* and *DOG*

The second type of similarity is one that is found in Sapper (Veale, 1995), and is based on the notion that though initially one concept may be mapped onto another, further mappings can be created from these initial mappings based on the larger structure that the initial mappings are part of. For example, in terms of a frame representation, suppose a concept A is like another concept B if both A and B share a number of identical attributes and value pairs, i.e. both concepts share a shallow similarity. Two further concepts that link to concept A and B may also now be linked together, if they share the same link label. This should become clearer as we look at an example involving the concepts *SCALPEL* and *CLEAVER*, and the concepts *BUTCHER* and *SURGEON*. This similarity encompasses shallow similarity but captures the structural similarities between representations.

Table 4.7 describes the concepts *SCALPEL* and *CLEAVER* in terms of a frame, with properties described in terms of slot name and slot value pairs. The concepts represented in Table 4.7 should be part of a wider network that has information on what a *KNIFE* is and who uses a *SCALPEL* and *CLEAVER* and so on. It is difficult to define concepts in isolation as they exist and are understood in relation to other concepts.

Table 4.7- Representations for *SCALPEL* and *CLEAVER*

SCALPEL		*CLEAVER*	
Isa:	*KNIFE, INSTRUMENT*	*Isa:*	*KNIFE, INSTRUMENT*
Part:	*BLADE, HANDLE*	*Part:*	*BLADE, HANDLE*
Attribute:	*SMALL, NARROW, SHARP*	*Attribute:*	*BIG, WIDE, SHARP*
Used_for:	*CUTTING*	*Used_for:*	*CUTTING*

Viewing the contents of Table 4.7 it can be said that a scalpel is like a cleaver in the following ways:

1. Both are knives
2. Both are instruments
3. Both are sharp
4. Both are used for cutting

All these similarities (1 - 4) are commonalities. For example, (1) can be found because both the representations for *SCALPEL* and the concept *CLEAVER* have the same slot name *is-a* and same slot value. (3) can be found because both have the slot name *attribute* and slot value "sharp". Figure 4.11 shows that the concept *SURGEON* links to the concept *SCALPEL* via the relation *control*. The link between the concepts *BUTCHER* and *CLEAVER* is also the *control* relation.

The deep similarity mapping suggests that on the basis of the shallow similarity between *SCALPEL* and *CLEAVER* the concepts *SURGEON* and *BUTCHER* should also be linked. This mapping is termed deep similarity as it builds on the first type of similarity and it captures structural similarity. The appropriateness of these mappings is only apparent when they are used in a suitable context. In interpreting the metaphor "My surgeon is a butcher" it may be profitable to see the similarities between the concepts *SCALPEL* and *CLEAVER* and the concepts *SURGEON* and *BUTCHER*. Consider also Figure 4.10 again, many deep similarities can be created on the basis of the shallow similarity between the concepts *CAT* and *DOG*. For example, *CANINE* can map onto *FELINE* as both have the same relation (*isa*) and occupy the same corresponding place in the structure. However, the deep similarity mappings that are found for Figure 4.12 should also include mappings between *NARROW* and *BIG* which seem inappropriate for any context. What mappings are used depends on the context of the interpretation. In the next chapter we will suggest that the interpretation of noun-noun combinations are based on deep similarities or shallow similarities.

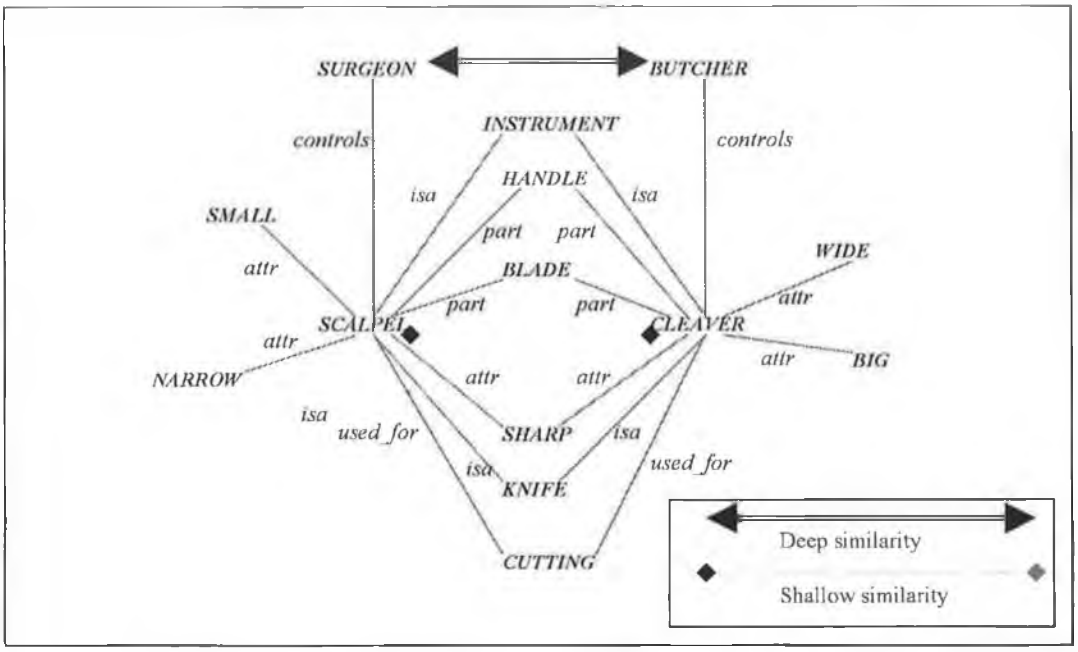


Figure 4.12 - Creation of deep similarity between *SCALPEL* and *CLEAVER*

Having given an outline of shallow and deep similarity we now present in Table 4.8 our conception of the rules for how these two similarities should be computed between concepts. We also suggest that deep similarities should be treated recursively, so they can encompass as much structure as possible and also ultimately be grounded in shallow similarities or if possible, other deep similarities. These rules also reflect an interesting distinction between the two. Deep similarities capture structural similarity while shallow similarities are based on shared attributes and features between concepts.

Figure 4.13 the concepts A1 and CI share two commonalities, X1 and Y1 via the links *p* and *k*, respectively. As the concepts E1 and D1 link to A1 and C1 via the same relation, *g*, a deep similarity can be found between the two concepts E1 and D1. Our mapping rules differ from Sapper (Veale, 1995) as discussed in Section 4.3.2 above, in that a shallow similarity requires two or more commonalities, whereas in Sapper a bridge can be based on a single commonality. Sapper, however, includes numerical weights to indicate the strength of a bridge (mapping) based on the number of supporting commonalities. Note, though, the similarity between Figure 4.13 and Figure 4.6.

Table 4.8 - Rules for finding shallow similarities and deep similarities

Shallow Similarity: Where a concept A and a concept B share at least two commonalities a shallow similarity exists between both concepts.
Deep Similarity: Where a concept A and a concept B share at least two shallow similarities, a deep similarity can be created between any two concepts that link to A and B respectively through the same relation. Further to this, when a concept A and a concept B share a deep similarity, a further deep similarity can be created between any two concepts that link to A and B respectively through the same relation. This can be applied recursively.

One of the basic tenets of structure-mapping theory is that there should be a one-to-one correspondence between concepts and/or relation names. When SME (Forbus et al., 1994; Falkenhainer et al., 1989) is run, many mappings are generated. Mappings, which break with this tenet, are dropped. In Sapper (Veale, 1995) the first dormant bridge (mapping) becomes the basis for the interpretation. We suggest that shallow similarities can be given a score based on the number of commonalities they are based on (which would be a score of at least two). Where there is a choice between a mapping based on a deep similarity and a shallow similarity then the deep similarity will be favored. If a concept can be mapped onto more than one concept via different deep similarity relations, then the deep similarity built on the largest structure will be used to generate an interpretation.

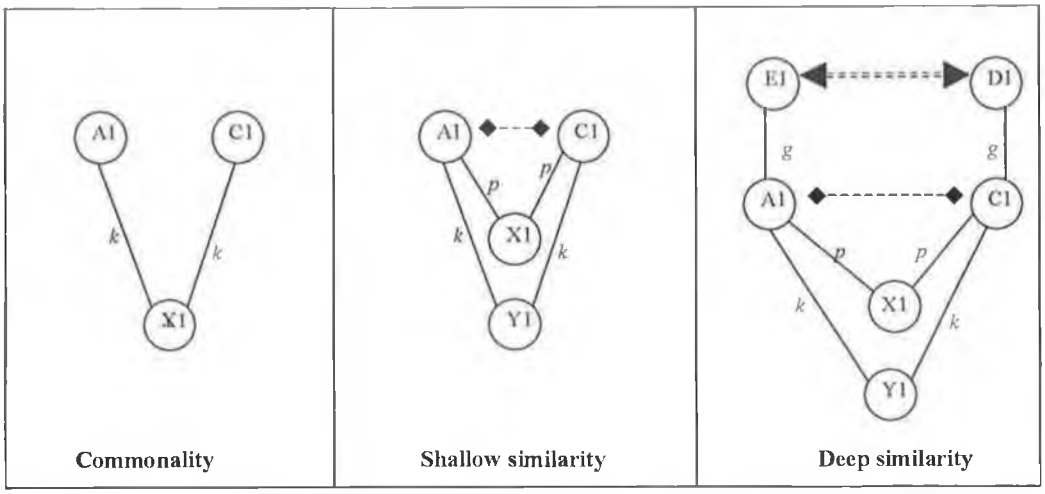


Figure 4.13 - Commonality, shallow similarity and deep similarity

4.6 An Example: Applying SA To Concept Combination

Our discussion of SA has suggested that it gives rise to two types of similarity: shallow and deep. However, it has not been stated how these either of these two types of similarity can account for concept combination. As already mentioned, the next chapter attempts to answer this question in detail but as a first pass the combination 'robin snake' is examined in SA terms. The combination 'robin snake' can be interpreted as a >> a snake that hunts robins <<. Taking this as our example, we will outline how SA can account for this interpretation. Firstly the concepts *ROBIN* and *SNAKE* are represented as frames in Table 4.9. These representations attempt to capture general knowledge about these concepts. It also contains prototypical information about these concepts, e.g. robins generally hunt worms, snakes generally eat rodents, such as mice.

Table 4.9 - Representations of *ROBIN* and *SNAKE*

ROBIN	*SNAKE*
Is-a: ANIMAL, BIRD,	*Is-a: ANIMAL, REPTILE*
Performs: FLIES, 0.6, HUNTS, 0.4	*Performs: SLITHERS, HUNTS*
Eats: WORM	*Hunts: MOUSE*
Parts: RED_BREAST, WINGS,	*Parts: FORKED_TONGUE, SCALES,*
Attr: SMALL, WEAK	*Attr: LONG, THIN, CYLINDRICAL, STRONG*

The alignment process would compare *ROBIN* and *SNAKE*. This would be achieved by comparing every sub-concept of *ROBIN* with every sub-concept of *SNAKE*. This process can give rise to shallow and deep similarities (if any exist). We suggest that *ROBIN* could be mapped onto the *MOUSE* concept of *SNAKE*. Why would this mapping take place? Depending on the representation of *MOUSE* there may be a number of commonalities between *MOUSE* and

ROBIN. Perhaps both are small and weak and a shallow similarity could be mapped between the two. Thus, *ROBIN* could replace this concept in *SNAKE*. Alternatively, an interpretation of 'robin snake', such as >> a snake eaten by a robin << may involve the concept *SNAKE* replacing *WORM* in *ROBIN*. Note that we have found these interpretations via the similarities that exist between the concepts and their related concepts. In the next chapter we describe six combination types in terms of SA. SA gives rise to both deep and shallow similarities and both are important for interpreting combinations.

4.7 Is Concept Combination the Same as Metaphor?

One of the central ideas behind the SA view of concept combination is that concept combination and metaphor have the same underlying process but are distinct cognitive entities. Both Sapper (Veale, 1995) and SME (Forbus et al., 1994; Falkenhainer et al., 1989) deal with similar classes of problems: metaphor and analogy. Yet concept combination is not metaphor nor is it analogy. SA involves a comparison process between conceptual structures. The conceptual structures can be treated as graphs or trees. In metaphor the largest isomorphic sub-graph is found. This is probably not the case with a large class of noun-noun compounds. Our assumption about concept combination is that it does not always give rise to the largest isomorphic sub-graphs, rather several distinct things can happen with respect to the alignment of the modifier and the head. In the next chapter we will show that concept combination can be treated in terms of fixed patterns of alignment. It is these fixed patterns of alignment that distinguishes concept combination from metaphor and other processes based on SA. At its simplest the interpretation of a combination will be based on a deep similarity or a shallow similarity.

In terms of the two types of similarity we have introduced we suggest that the interpretation of a metaphor can be conceived as building the largest set of linked deep similarities. For example, interpreting "a composer is a general" would involve finding deep similarities between subconcepts of the base and target but also probably deep similarities between the daughters of these subconcepts. Of course this set of nested deep similarities would have to conform to the principles of structural alignment, i.e. involve one-to-one mappings, display parallel connectivity and systematicity. The interpretation of combinations, however, may rely on shallow similarities as well as deep similarities. We suggest that this is one of the critical differences between metaphor and concept combination.

4.8 Arguments Against Applying Structural Alignment To Concept Combination

At this stage it should be acknowledged that here have been a number of arguments put forward against SA being involved in concept combination (Keane & Costello, 2001). These arguments arise from discussions between the proponents of Dual Process Theory (Wisniewski, 1997a, 1997b) and Constraint theory (Costello, 1996; Costello & Keane, 1998; Costello & Keane, 2000). Although these arguments are directly applied to Dual Process Theory they should be examined here as we are arguing that SA is central to concept combination. There are six main arguments in Keane & Costello's (2001):

1. Speed of interpretation
2. Analogies can be extended
3. Analogy and concept combination serve different functions
4. Combinations have alternative meanings which analogy does not have
5. Limited pragmatic context
6. Empirical findings

We outline each of these briefly and suggest that they are, in the main, weak arguments against applying SA to concept combination.

1 Speed of interpretation

Keane & Costello (2001) point out that the interpretation of combinations is rapid whereas the interpretation of analogies is often slow and considered. This is interesting as it suggests that we cannot have fast interpretations via SA. Taking this to its logical conclusion, we could argue that any metaphors or figurative language that occur in common discourse cannot be the result of SA. This seems to contradict much of the work done in the area.

2 Analogies can be extended

It is quite clear that many analogies can be extended. In terms of SA we can suggest that further mappings can occur. Keane & Costello (2001) say this is not true of concept combination. However, there are some combinations that can be extended. These combinations are often figurative, e.g. when interpreting 'maths clinic', it could be said that a "teacher is like a doctor" and so on. The fact that many combinations cannot be extended, which we concede to Keane & Costello (2001), does not by itself mean that the interpretation of combinations is based on SA.

3 Analogy and concept combination serve different functions

Keane and Costello (2001) suggest that concept combination and analogy serve different functions in language. However, this just suggests that concept combination and analogy are distinct, not that SA cannot occur in concept combination. Consider metaphor and analogy, it could be argued that they both serve a different purpose in language. An analogy is more pedagogical than metaphor and often people are explicitly given more of the mappings in an analogy than in a metaphor. So when a subject uses the 'Rutherford atom' analogy we should expect them to say things such as "the nucleus is like the sun". This is to be expected because analogies are often used to teach people. Yet we would not say that because of this different purpose metaphor does not involve SA.

4 Combinations have alternative meanings which analogy does not have

Many combinations are extremely polysemous, in that they will have multiple meanings. Keane & Costello (2001) say that this is not true of analogies where there is generally only one interpretation available. Again, we can simply say that they have demonstrated that analogy and concept combination are different.

Related to this is the fact the metaphors can also be quite polysemous, although context may often serve to hide this. Consider that when Romeo says 'Juliet is the sun', we gather from the context of the story that Juliet is not a large fiery mass. But in a situation without a context this is only one possible interpretation. Other interpretations are possible, e.g., >> Juliet is the sun - because she never comes out at night <<, >> Juliet is the sun, if you get too close you'll be destroyed <<. Considering this, we would not say that metaphor cannot involve SA. Polysemy in itself offers no evidence of this. We must remember that we use combinations in everyday language where the context must remove much of the polysemy. If this were not the case, we would have extreme difficulty in even carrying out the simplest conversation due to the large number of meanings that could be derived from the speaker's utterances.

5 Limited pragmatic context

Here, we find ourselves disagreeing with Keane and Costello (2001) in the strongest terms. They suggest that the pragmatic context of combinations is limited in that they are only used to convey information in a compact and concise way. This just simply is not the case. Combinations occur wherever language is used and so must be used in practically every context - contexts, where people are serious, where people are playing and where people are lying.

The clearest example that we can offer of this is in the use of euphemism. A euphemism is generally defined as the use of a word or a phrase in substitution of something more offensive. Sometimes these euphemisms are used to hide the truth, e.g. we now have terms like 'collateral damage' for >> civilian deaths << and 'friendly fire' for >> troops killed by their own side <<. These combinations do not involve the conveying of information in a compact and concise way.

6 Empirical findings

The sixth point that Keane and Costello (2001) make is based on empirical findings. However, these findings involve comparisons between Dual Process theory and Constraint Theory. We would like to suggest that Dual Process Theory is the straw man of SA. It does involve SA but proposes an additional process of scenario-creation. As such it is a step in the right direction but is in itself unwilling to see SA as the fundamental underlying process. Over the last number of years Costello and Keane have largely shown that their model offers the best fit to the phenomena of concept combination when compared to Dual Process Theory (e.g. Costello & Keane, 1997; Costello & Keane, 1998; Costello & Keane, 2000). However, this in itself does not absolutely rule out that SA is involved. It merely suggests that Dual Process Theory is not as successful as Constraint Theory. There is also some evidence for SA that Keane and Costello (2001) did not consider, the existence of figurative combinations.

In essence, we suggest that concept combination is not analogy. This does not mean though, that concept combination cannot involve SA. If we were to take this view then we put ourselves in a strange position where someone could argue that the same goes for metaphor: metaphor is not analogy, and so cannot involve SA. Even when many researchers (e.g. Veale, 1995, Markman & Gentner, 2000) assume that it does seem to share the same underlying process.

4.9 Chapter Summary

This chapter opened by introducing similarity. A comparison process gives rise to similarities and differences. We suggested that similarity involves more than simple features but also structural similarity. Examples from analogy were given to show how structural similarity is important. The principles of SA were then outlined. They are: one-to-one mapping, systematicity, parallel connectivity. Different rules for implementing these principles were given. Our view of SA was then set forth. SA should be set against a background of a knowledge base or a network. The representations in this network should be like the ones used in concept combination research already, i.e. they should be frame-based. The alignment process gives rise to shallow similarity and deep similarity. The process of concept combination relies on both. An example of SA in concept combination was given where an interpretation relied on deep similarity. In the next chapter, more examples will be given when six combination types are introduced. The arguments against applying SA in concept combination were looked at. Although interesting, these arguments do not really suggest that SA cannot be applied to concept combination.

Chapter 5 - A Structural Alignment View of Concept Combination

In this chapter a new approach to concept combination is set out. This approach sees structural alignment as the core process in concept combination. This model is motivated by an attempt to account for world knowledge, polysemy and figurative language in concept combination with a single mechanism. This model also suggests that a single syntactic form, the noun-noun compound, can give rise to several distinct types of semantic interpretations. These distinct types of interpretation involve different patterns of structural alignment. Some interpretations of noun-noun compounds require little processing, some require a great deal.

We explain how this new structural alignment approach deals with the three main problems of concept combination. With respect to the problem of world knowledge a number of the combination types depend on finding a suitable context to generate an interpretation. This context is found in the knowledge base (a network of concepts) and we outline how it can be found. Polysemy in the structural alignment approach largely results from a combination falling into more than one combination type. In addition, the polysemy of an interpretation could also result from the lexical polysemy of constituent nouns. Figurative combinations are found in the same way that literal ones are, i.e. one mechanism can generate both. After accounting for the three problems we outline how a rule-based implementation of our model can be created.

5.1 Introduction

The structural alignment view of concept combination suggests that structural alignment (SA) is the underlying mechanism in concept combination. This entails that the interpretations that are generated for the concept combination process are the result of SA. SA has already been used to model cognitive processes, e.g. metaphor and analogy. We now argue that it can also be used to account for concept combination.

The division of compounds into property-based and relation-based interpretations is one that is not found in our view of concept combination. It is clear that the interpretations of compounds can be divided into groups but this does not necessarily mean that the process itself can be similarly divided. That is, there does not have to be two distinct processes in concept combination because there are distinct types of interpretation. [1]

[1] as, e.g. Dual Process Theory (Wisniewski, 1997a, 1997b), suggest.

5.1.1 A Pure SA Based Approach vs. Dual Process Theory

The careful reader will have noted that SA was first introduced into the area of concept combination by Dual Process Theory (Wisniewski, 1997a, 1997b). Dual Process Theory suggests that SA is involved in the generation of property-based interpretations. Another mechanism, scenario-creation, is used to account for relation-based interpretations. This division of concept combination into multiple processes suggests that SA cannot on its own account for concept combination, otherwise no additional mechanisms would be required. Our approach disagrees with Dual Process Theory and on this point suggests that SA (among others) can account for a large group of combinations, including those that may be classified as property-based or relation-based interpretations. Our perspective on SA was given in the last chapter. Essentially, SA is a comparison process that gives rise to two types of similarity: shallow or surface similarity and deep or structural similarity. These similarities are used to support different interpretations when concepts are compared.

5.2 A New Model Based On SA

The main burden in proposing that SA underlies concept combination is to describe how one process can generate a range of different interpretations. Consider the compounds in Table 5.1 and their respective interpretations.

Table 5.1 A range of interpretations

1. 'book cover' - >>The front cover part of a book<<
2. 'book review' - >>A review of a book<<
3. 'book shop' - >>A shop that primarily sells books<<
4. 'book pirates' - >>People who illegally copy and sell books<<

The interpretation for (1) in Table 5.1 suggests that the combination refers to an aspect of *BOOK*. It is not so much a type of *COVER* but more a part of a *BOOK*. Interpretation (4) refers to *PIRATES* but does not involve pillage on the open sea, it rather seems to refer to illegally copying and selling books. All the compounds above have the same syntactic structure, they are made up of a noun-noun sequence. Although they have the same syntactic structure, the semantic range is not so neat. The interaction between the constituent elements is not the same across every combination. It is proposed that there are several distinct types of combination at the semantic level, and for each of these there is a schematic pattern in terms of SA.

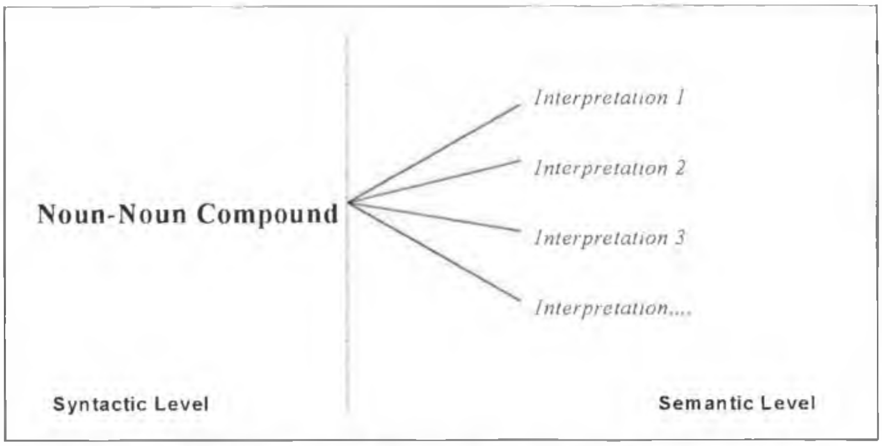


Figure 5.1 - Differences between syntactic and semantic level

Figure 5.1 shows that compounds operate at both the syntactic and the semantic level. The semantic level may be more complex than the syntactic. In the figure, the noun-noun compound can give rise to several interpretations. The straight lines from the noun-noun compound to what we will refer to as the "combination types" suggest that one or more of several interpretations may be possible. We suggest that the semantic form of compounds (the combinations) can be categorised into combination types. Most approaches to concept combination have to deal with problems related to complexity. To reduce this complexity, we propose that most interpretations of noun-noun compounds will fall into just one of several combination types. Some of these combination types will require little processing while others will require a great deal. This assumption is examined in more detail in Section 5.5.

Each combination type reflects a schematic pattern of alignment between the concepts referred to by the head and the modifier. If the alignment falls into one of the six patterns we specify in Section 5.2, then an interpretation is generated. This schematic pattern of alignment may involve only one or both types of similarity (deep and shallow) that we mentioned in Section 4.3. Some of the schematic patterns of alignment involve accessing of world knowledge, some do not.

Before analysing the various combination types separately, we classify all interpretations into one of two broad groups: the referential and the constructive. The division of interpretations into these two groups is motivated by our desire to show that two distinct but similar phenomena appear to be occurring in concept combination. Sometimes, interpretations appear to refer to structures that already exist in memory, e.g. 'book cover'. At other times interpretations are constructive: they construct something new that may be added to memory, e.g. 'book pirates'. Of course, when an interpreter understands a combination, its meaning is added to memory. We expect that when the same combination has to be understood at a future date, the meaning is retrieved from memory rather than reconstructed anew. At its broadest the structural alignment model attempts to account for the two broad types of combination: the referential, and the constructive.

5.2.1 Combination Types

To summarise, we propose that there are a number of combination types that operate at the semantic level. These combination types involve fixed patterns of alignment. Six combination types are proposed. These combination types are not assumed to be exhaustive but are expected to offer a good coverage of all interpretations. The combination types are listed in Table 5.2. From this section onwards we use the terms modifier concept and head concept in our discussion of combination types. The modifier concept is the concept referred to by the modifier while the head concept is the concept referred to by the head. We wish to emphasise the conceptual approach of our work and to reinforce that the focus of this current research is aimed at the semantic level, not the syntactic. Each of the combination types listed in Table 5.2 is outlined in the next section.

Table 5.2 - Combination types

Referential:		
	1.	Meronymous Combinations, e.g. 'book cover'
	2.	Catachrestic Combinations, e.g. 'table leg'
Constructive:		
	3.	Contextual Combinations, e.g. 'drug baron'
	4.	Insertion Combinations, e.g. 'fur coat'
	5.	Context Insertion Combinations, e.g. 'mink coat'
	6.	Conjoined Combinations, e.g. 'house boat'

5.2.1.1 Meronymous Combinations

This type of combination is the simplest of the referential type, where the head concept is already a part of the modifier concept. The combination is used to refer to a part of the modifier concept. Consider that in the combination 'library book', a representation of *LIBRARY* would have information regarding books, e.g. books are stored in a library and people borrow these books. So a 'library book' would be >>a book from a library that can be borrowed<<. Other examples of this type of combination would be: 'book cover', 'chicken wing'. Meronymous combination is taken from the term meronymy.[2] Meronymy is the lexical relation between words that denote parts and wholes of the same concept, e.g. this

[2] This relation is also known as parlonymy.

relationship holds between shirt and sleeve since a sleeve is part of a shirt. A sleeve is a meronym of shirt (and a shirt is a holonym of sleeve).

5.2.1.2 Catachrestic Combinations

These types of combination are slightly more complicated than the meronymous combination type, since the head concept is actually a metaphorical reference to an aspect of the modifier concept. So as in meronymous combinations, the combination refers to a feature that already exists in the modifier concept. However, in this case we do not have a clear lexical term for this feature and use the head concept as a metaphorical reference to it. These type of combinations are usually so commonplace that we are often unaware of the metaphors involved. An example of this type of combination would be 'book spine', where *SPINE* is an exemplar of back support and is used to name a part of *BOOK* that fills a similar role. Other examples of this combination are: 'bottle neck', 'table leg'. This type of combination is named after the term catachresis, which is a term used in traditional rhetoric for the misapplication of a word for another. However, Black (1962) uses this term for the metaphors that people use to fill lexical gaps which appears to be the case with combinations such as 'table leg'.

5.2.1.3 Contextual Combinations

One of the assumptions behind the proposed model is that certain types of combinations are knowledge intensive, in that they often appear to require a great deal of outside information or world knowledge. So, it appears that some combinations draw in larger contexts. This is not surprising if one adopts a representational approach to meaning where words refer to concepts and these concepts are in turn made up of other concepts. From this perspective a seemingly simple concept can become quite complex once analysed.

In a contextual combination the modifier concept refers to a larger conceptual structure which is then aligned with the head concept. Consider the combination 'loan shark'. It appears that the modifier concept *LOAN* acts as a reference to the broader context of *MONEY-LENDING*, with an aspect of this, *MONEY-LENDER*, being mapped onto *SHARK*. In a sense the combination is a 'money lender shark'. Subjects appear to cram as much meaning into the noun-noun lexical structure as they can. Other examples of this combination type include: 'drug baron', and 'book pirate'. We suggest that a contextual combination can always be paraphrased in terms of the imported context e.g. 'drug baron' can be paraphrased as 'drug dealer baron' and 'book pirate' can be paraphrased as 'book publisher pirate'.

5.2.1.4 Insertion Combinations

This type of combination does not involve drawing large amounts of information; instead the modifier concept refers to an aspect of the head concept that the modifier concept then replaces. Consider the combination 'fur coat'. Here the prototypical *made-of* feature value of *COAT*, *CLOTH*, is replaced by *FUR*. We understand a 'fur coat' as >> a coat that's made from fur <<. This process involves comparing the modifier concept with every aspect of the head concept. A suitable aspect for insertion and replacement is found when the aspect most similar to the head concept is discovered. The modifier concept then replaces this aspect.

This combination type is a very broad category and it can encompass some relation-based interpretations and some property-based interpretation of compounds. The example, 'fur coat' is a property-based interpretation. It may not be clear how this combination type also covers relation-based interpretations. Ultimately, it depends on how concepts are represented. Consider the combination 'robin snake' and the interpretation that it is >>a snake that hunts robins<<. If the representation of *SNAKE* contains information that the snake *is-a PREDATOR* and *hunts PREY* then *ROBIN* may be similar to the *PREY* aspect of this representation. The *ROBIN* concept and its subparts then could replace this aspect. This combination type is similar to the types of interpretations that Concept Specialisation (Murphy, 1988) attempts to deal with. We will, in contrast to Murphy (1988), demonstrate how slots are selected (see Section 5.3.4).

As this combination type gives rise to what could be seen as property-based and relation-based interpretations, it suggests that there is not a neat correspondence between the distinctions that Dual Process Theory (Wisniewski, 1997a, 1997b) makes and our approach. This could be because in our approach the classification of the interpretation of compounds into property-based and relation-based occurs *after* the process of concept combination, and does not reflect the internal mechanics of the process of concept combination itself.

5.2.1.5 Context Insertion Combinations

This type of combination is similar to the insertion type except that the modifier concept now refers to another concept or context which is then inserted into the head concept replacing a corresponding structure in that head. If we consider the combination 'video recorder', *VIDEO* can refer to a large number of contexts e.g. 'video games', 'video conferencing', 'video cassette' and 'video phone'. In this combination we replace the prototypical *cassette* of *RECORDER* with 'video cassette'. Again like the contextual combination types, the context insertion types can be paraphrased in terms of the context. 'Video recorder' seems to be the conceptual shorthand for 'Video-cassette recorder'.

Now consider the combination 'mink coat', if this interpreted as >>a coat made of mink<< it seems that a subpart of *MINK*, namely *FUR*, is aligned with an aspect of the head concept. The interpretations that this combination type gives rise to could be regarded as property-based by Dual Process Theory (Wisniewski, 1997a, 1997b). However, note that the context in these insertion combinations may not just involve sub-properties of the modifier concept, but also concepts in which the modifier concept itself is a sub-property.

5.2.1.6 Conjoined Combinations

This combination generally involves the conjunction of features between the modifier concept and the head concept. This combination when interpreted would be classified as a conjunction or hybridization. A conjoined combination is in some sense both a member of the modifier concept category as well as that of the head concept category. For example, in the combination 'servant girl' the person referred to is in some sense both a *GIRL* and a *SERVANT*.

This does not mean that all conjoined combinations involve the simple conjunction of all the properties of both modifier concept and head concept. It is still possible that a certain amount of accommodation between the two concepts must occur. In the case of 'house boat' which is in some sense both a house and a boat, a subject would map the structure of *BOAT* onto *HOUSE*, perhaps even replacing the structure of *floor* in house with *hull*. This type of conjoined structure is more akin to hybridization in Dual Process Theory (Wisniewski, 1997).

5.3 The Combination Types In Terms of SA

In this section each of the combination types is modelled in terms of SA. In each of the examples a figure or number of figures will be presented and then used to discuss how the combination type works. Each of the examples below can be treated as a schema or template for how the combination type works in structural alignment terms. To understand these examples properly, the typography used should be explained. Figure 5.2 shows a frame based representation of the concept and a visual representation of the same concept. Note that both representations convey or carry the same information. The visual representation allows us to see the structure of the representation in a more graph-like way, where concepts are nodes and relations are arcs. In general, the concept that is being represented will be placed in the centre of the graph and its properties will surround the concept. For example in Figure 5.2, the visual representation shows the concept *BANK* and so it is placed in the centre. The properties of *BANK* are placed around it, e.g. *TELLER, MONEY* and so on.

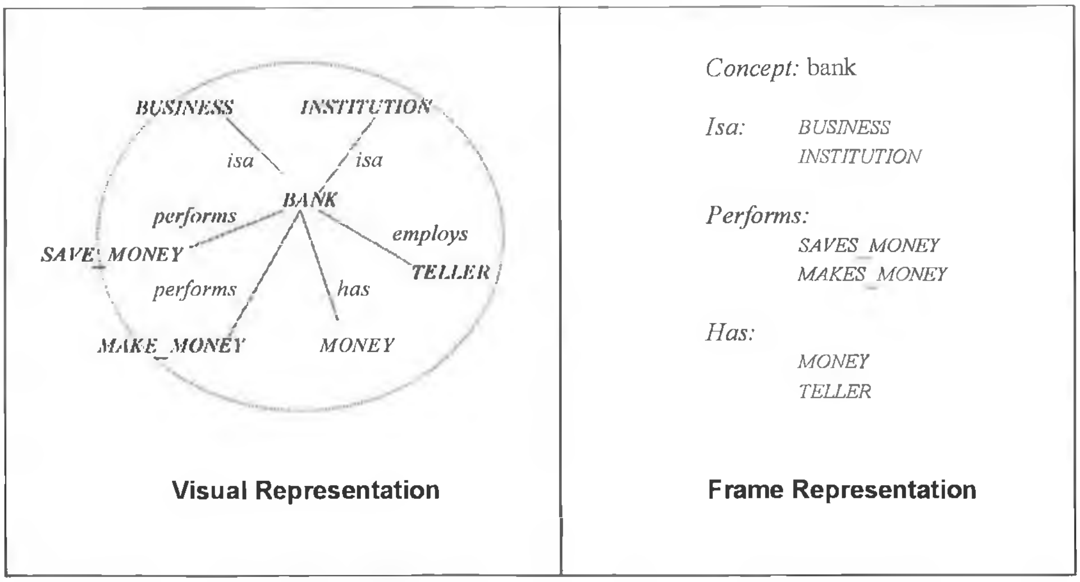


Figure 5.2 – Frame and Visual representations of the concept *BANK*

It should also be noted that the representations in Figure 5.2 are token representations. As we envisage that concepts should be represented within a semantic network, it is the network that holds all the information. The representation of bank in Figure 5.2 would only be a selective representation when compared to the network as whole. When giving examples of how the combination types work in SA terms we will present information that exists in the network selectively. Not every concept that is linked to another will be presented, as in space terms this would prove impossible. Rather the most relevant parts of the semantic network will be presented. In Chapter 6 it will be seen that more elaborate representations are actually used.

To demonstrate the structural alignment process, the typography in Figure 5.3 is used. As previously described, SA involves a comparison process. This comparison process can give rise to two types of similarity: shallow and deep similarity. These similarities are based on commonalities and so these are also shown. The arrows in Figure 5.3 signal that a shallow similarity has been found based on two commonalities. The graphical conventions used for similarities are shown in Figure 5.4. Figure 5.3 shows that the *COVER* aspect of *BOOK* is in a shallow similarity relation with the *COVER* concept. The arrows in Figure 5.3 can also be described as mappings. All mappings in the various diagrams mark out one of the two types of similarity that SA can give rise to or the commonalities that give rise to these similarities.

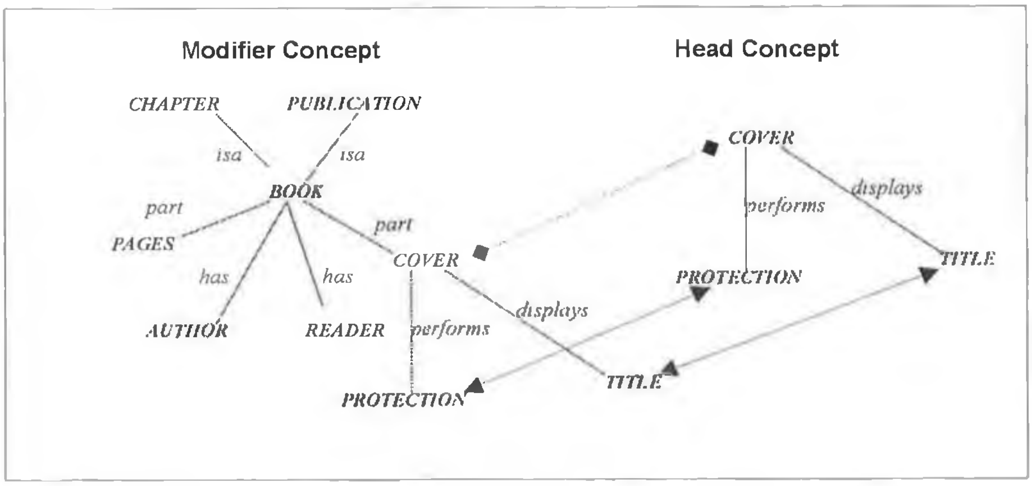


Figure 5.3 - A representation of alignment

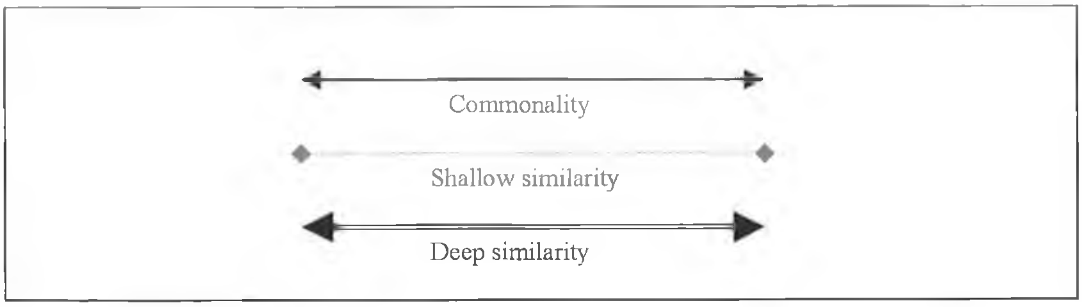


Figure 5.4 - Graphical representations of similarity types

A number of terms are also used within the coming sections that should be explained. The term subconcept refers to concepts such as, *BUSINESS, INSTITUTION, MONEY, TELLER* etc. with respect to the representation of *BANK* in Figure 5.2. In essence all concepts that are linked to another concept are subconcepts of the concept they are linked to. The term aspect is used interchangeably with subconcept and so both are considered to have the same meaning in this present context.

As mentioned in the last chapter, a deep similarity is based on at least two shallow similarities or another deep similarity. Figure 5.5 shows a graphical comparison of both. We suggest that some of the combination types we have outlined could involve both types of similarity. The more information that occurs in the knowledge base the easier it would be to find deep similarities. When outlining each combination type we will state whether it is based on a shallow similarity, a deep similarity or, in some cases, it can involve both. In Figure 5.5 a shallow similarity is formed between the concepts A1 and C1 because they have at least two commonalities. A deep similarity can be found between E1 and D1 because they each have the same relation, *g*, with A1 and C1 respectively.

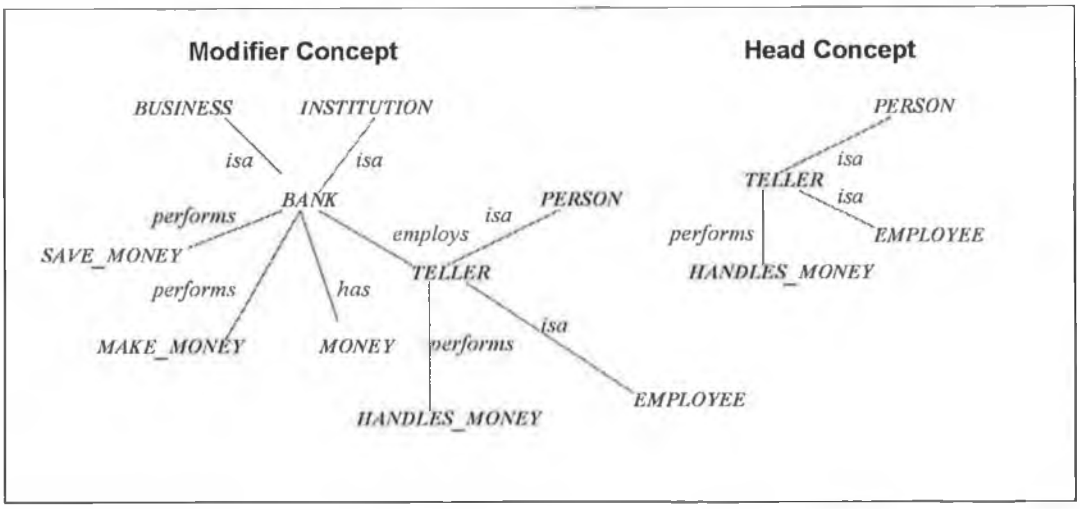


Figure 5.6 - Concepts *TELLER* and *BANK*

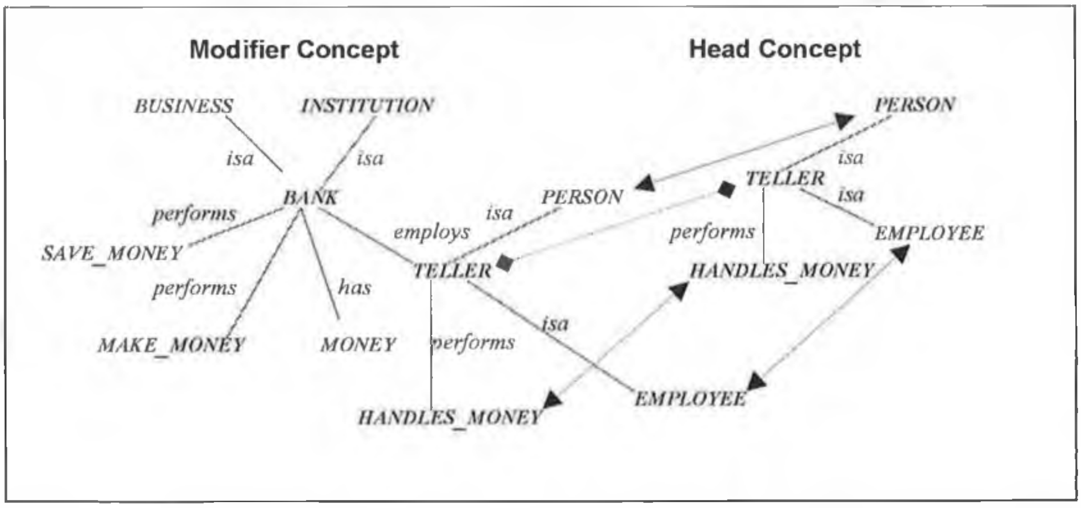


Figure 5.7 - Mappings in 'bank teller'

Abstracting from the example in Figure 5.7, meronymous combinations appear to work in a particular schematic way. If the modifier concept is seen as a graph structure, called A, then the head concept, B, is a sub-graph of A. This sub-graph is found by matching every sub-node of B with every sub-node of the graph structure A. The head concept and the sub-graph of A will be identical. This is shown in Figure 5.8 where the head concept already exists in the modifier concept. It should also be noted that it was suggested that the relationship between the modifier concept and the head concept is one where the head concept fills a *part* role (see Section 5.2.1.1). Thus in terms of Figure 5.8 the link between the node A1 and A3 in the modifier concept would probably be the *part* relation, although other relations are possible, e.g. in Figure 5.6 and 5.7 the relation is *employs*. We expect the *part* relation to be the most frequently found in this type of interpretation and this may allow for a shortcut in the processing of these types.

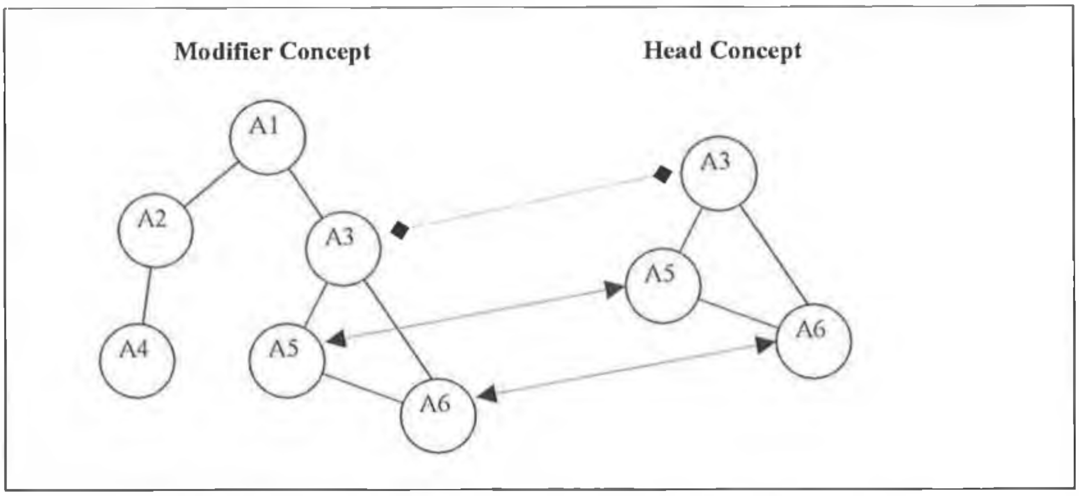


Figure 5.8 - Alignment schema for meronymous combination type

5.3.2 Catachrestic Combinations

This combination is similar to the meronymous combination but with one critical difference. In this case the head concept is a metaphorical reference to an aspect of the modifier concept. It was noted in Section 5.2.1.2 that catachrestic combinations fill lexical gaps, where the aspect of the modifier concept they refer to may not be easily lexicalized. In Figure 5.9 the concepts *BOOK* and *SPINE* are represented. In the interpretation of the compound 'book spine' the concept *SPINE* is aligned with every subpart of the concept *BOOK*. If *SPINE* is similar to an aspect of the modifier concept then the head concept is a reference to this aspect of the modifier concept.

In aligning *BOOK* and *SPINE* it should be found that the concept *SPINE* can be placed into correspondence with *LONG_THIN_PART*. The concept *LONG_THIN_PART* may have a strange name but we need to name a part of *BOOK* which cannot be easily lexicalized.

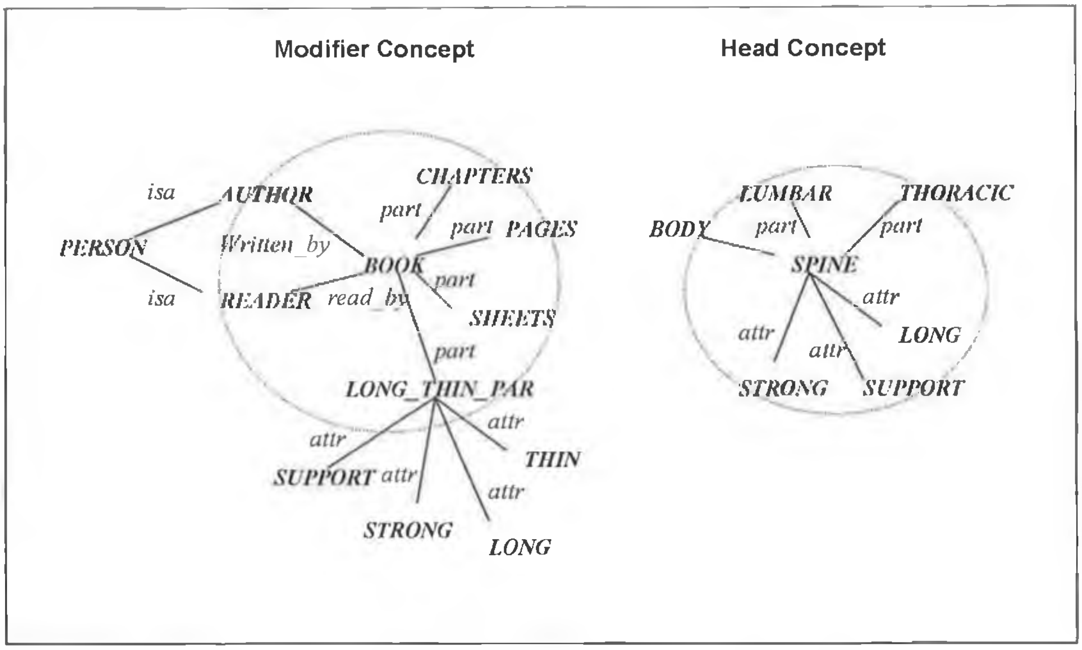


Figure 5.9 - Representation of concepts in 'book spine'

The alignment process would compare *SPINE* with all concepts linked with *BOOK*. This means that according to Figure 5.9, *SPINE* would be aligned with: *CHAPTERS, SHEETS, PAGES, AUTHOR, READER,* and *LONG_THIN_PART*. In Figure 5.9 only the subparts of *LONG_THIN_PART*, *AUTHOR* and *READER* are given. In reality, each of the subconcepts of *BOOK* would also be represented. Comparing *SPINE* and *LONG_THIN_PART* there are a number of shallow similarities. Both are strong, both offer support, and both are long. From these similarities we suggest that *SPINE* can be matched with *LONG_THIN_PART*, since there is a shallow similarity between the two. These mappings can be seen in Figure 5.10. Checking to see where *LONG_THIN_PART* fits into *BOOK* would generate the interpretation for 'book spine'. It is a part of *BOOK* and has the following attributes: *strong, support,* and *long*. So 'book spine' could be interpreted as >>a part of a book that offers support, is strong and long<<.

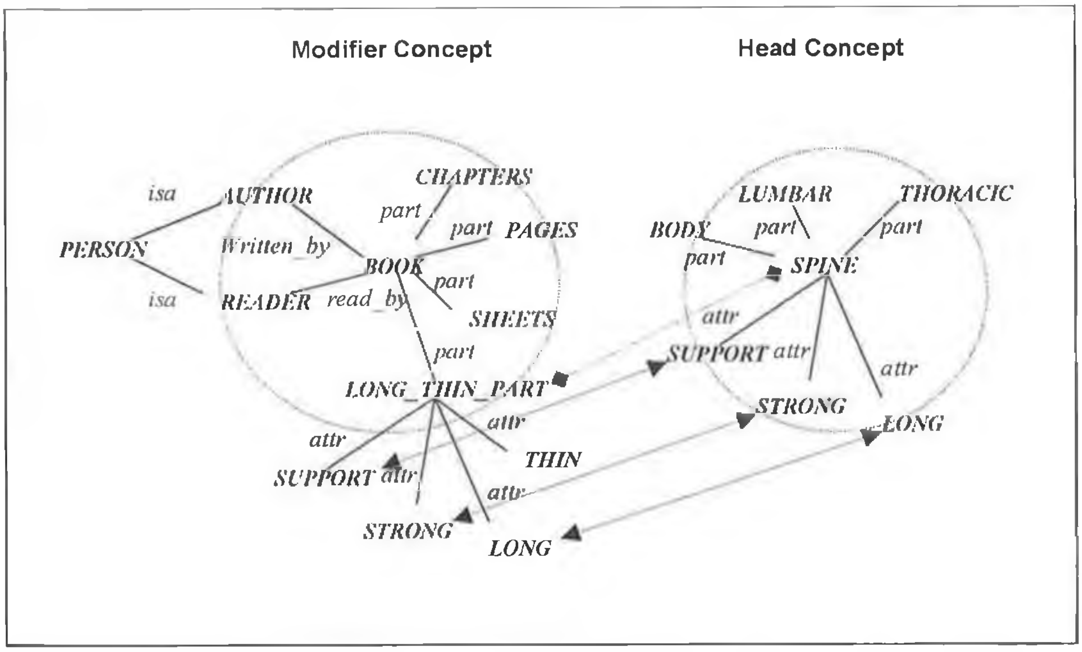


Figure 5.10 - Mappings in 'book spine'

Taking the example 'book spine', we can abstract as to how a catachrestic combination works in relation to SA. In a catachrestic combination, if the modifier concept is a graph A, then the head concept can be mapped onto a subpart of A in terms of a shallow similarity. In Figure 5.11 node A3 is mapped onto node B2 as a result of shallow similarities between similar nodes in the modifier concept and head concept. The difference between this combination and the meronymous one is that when the shallow similarity is found between the head and an aspect of the modifier, every subconcept of each is identical. This is not necessarily the case with the catachrestic combination type.

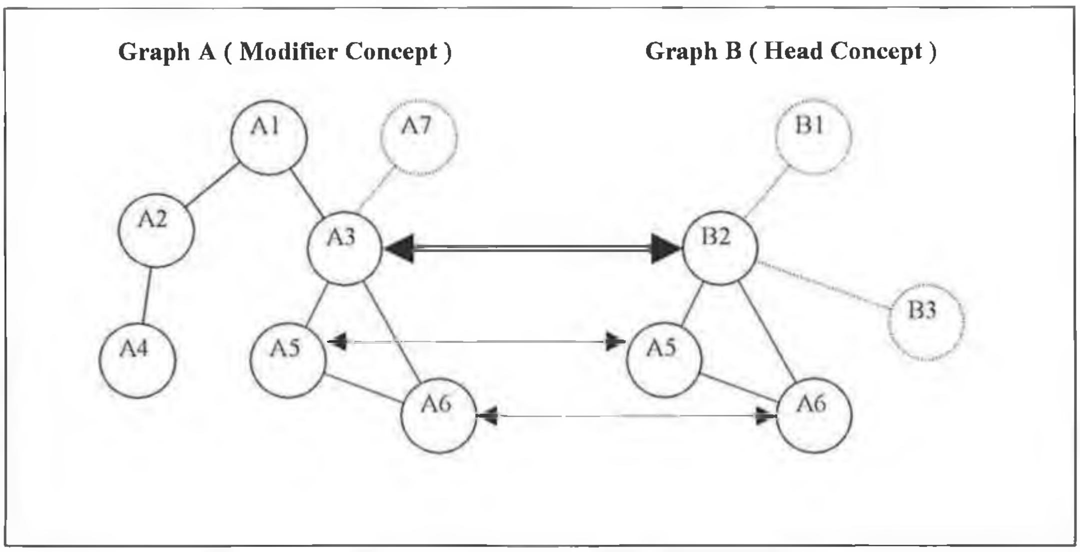


Figure 5.11 - Alignment schema for catachrestic combination types

5.3.3 Contextual Combinations

In contextual combinations more than just the modifier concept and head concept are involved in the alignment process. In this combination type the modifier concept refers to a context that is then aligned with the head concept. One example of this type of combination is 'drug baron' where *DRUG* appears to refer to *DEALER* or a *PUSHER* and it is this that is aligned with *BARON*. But one may very well ask how we know that *DRUG* refers to a *PUSHER* in this context? In the previous combination type the head concept was aligned with all concepts linked to the modifier concept. A context may be found by finding all concepts that have the concept *DRUG* as a feature, e.g. depending on the representations used this could include: *PUSHER*, *DOCTOR*, *NURSE*, *PATIENT*, *CHEMIST*, *USER*, or *MEDICINE*. To see this more clearly, examine Table 5.3 which lists representations of the concepts *DOCTOR* and *PUSHER* in terms of slot names and slot values.

Table 5.3 - Representations of *DOCTOR* and *PUSHER*

DOCTOR		*PUSHER*	
Is-a	*PERSON, ANIMAL*	Is-a	*PERSON, ANIMAL*
Attribute	*SKILLED*	Attribute	*POWERFUL, DANGEROUS*
Perform	*DIAGNOSIS, MEDICINE*	Perform	*BUYING, SELLING*
Prescribe	*DRUG*	Buys	*DRUG*
		Sells	*DRUG*

In Table 5.3 both concepts refer to the concept *DRUG*. The concept *DOCTOR* involves the prescription of drugs, while a *PUSHER* buys and sells drugs. These may be suitable contexts for the concept *DRUG*. Figure 5.12 provides a visual representation of this.

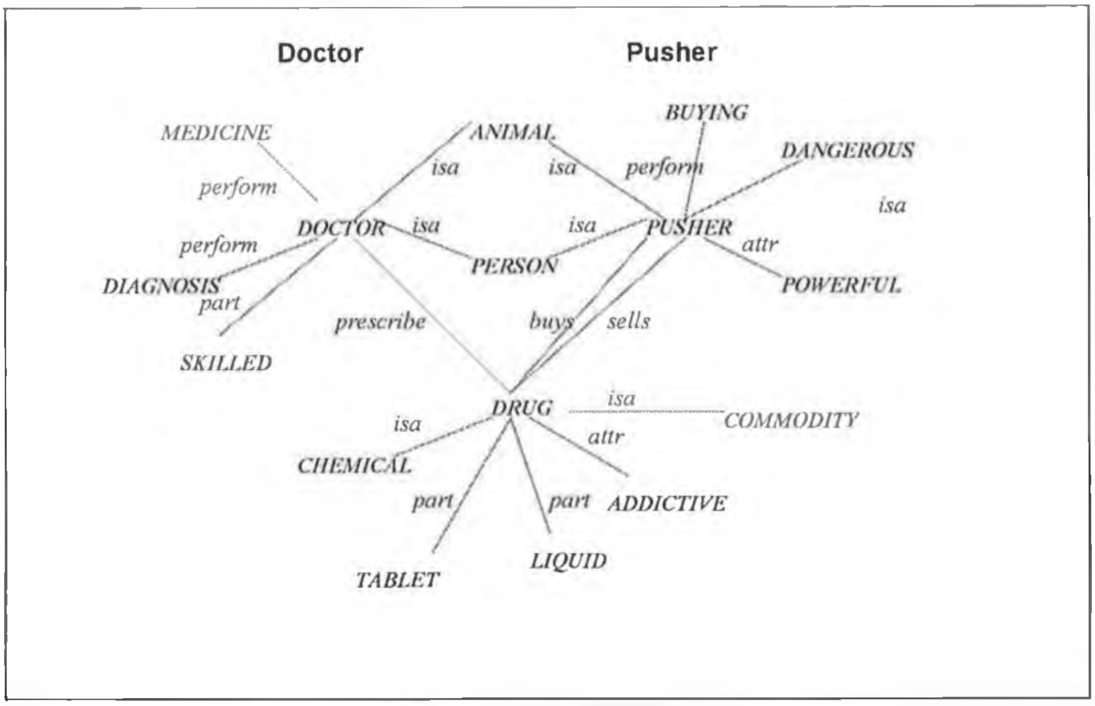


Figure 5.12 - Possible contexts for *DRUG*

The alignment in a contextual combination primarily involves finding all suitable contexts of the modifier concept. The most simple approach to this is to assume that any concept that lists *DRUG* as a feature or attribute is a suitable context. These contexts are then aligned with the head concept. This can be computationally expensive when a large number of contexts occur. Furthermore, the compound 'drug baron' may involve several contexts. All these contexts are aligned with the head concept, *BARON*. This involves looking for similarities between the context and *BARON*. A *PUSHER* may provide the most similarities with *BARON* since a *PUSHER* exercises control of the *USER*, and a *BARON* exercises control of his *SERFS*. Both are powerful, both are dangerous and both control other people. In Figure 5.13 there are a number of possible matches between *BARON* and *PUSHER*. The most important is between *SERF* and *USER*. This match is based on the fact that both the serf and user are weak and poor. Importantly, both are also controlled. There is a deep similarity between *PUSHER* and *BARON*. So ultimately a contextual combination depends on deep similarity between the head concept and the context.

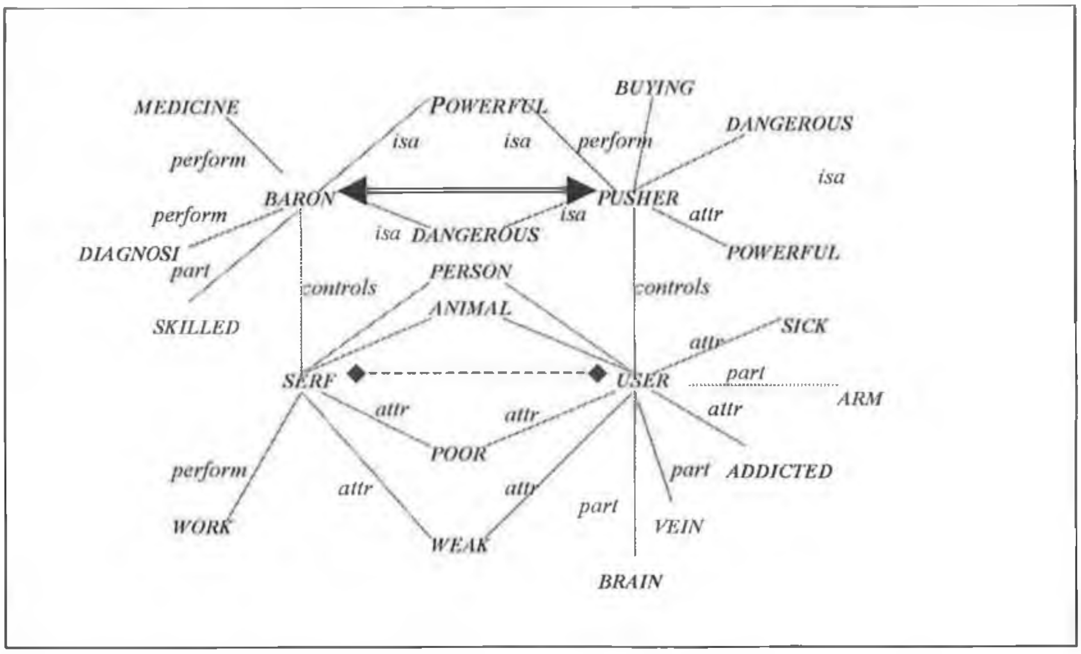


Figure 5.13 - Representation of *PUSHER* and *BARON*

It may be possible that the contexts are evoked in different ways, another possible way is to see if the modifier concept exists in a taxonomy where perhaps concepts above it may be possible contexts. Another way may be to reverse this, to see what concepts below the modifier concept may provide suitable contexts.

The example of 'drug baron' can be abstracted into the alignment schema in Figure 5.14. This alignment is difficult to capture visually. In this diagram the node A1 represents the modifier concept. In reality the modifier concept would be a graph with many nodes. Some of these have not been entered to avoid cluttering the figure. The nodes that A1 is linked to are considered possible contexts. As was stated above, there are certain criteria for what constitutes a possible context in this example: where A1 is a node in a larger graph, the node it is linked to in the larger graph is the possible context. If a context can be mapped onto the head concept via a deep similarity, then a context combination can be found. In this example, C1 is mapped onto E1 due the shallow similarity between C2 and C2. This relies on finding one or more deep similarities between C1 and E1.

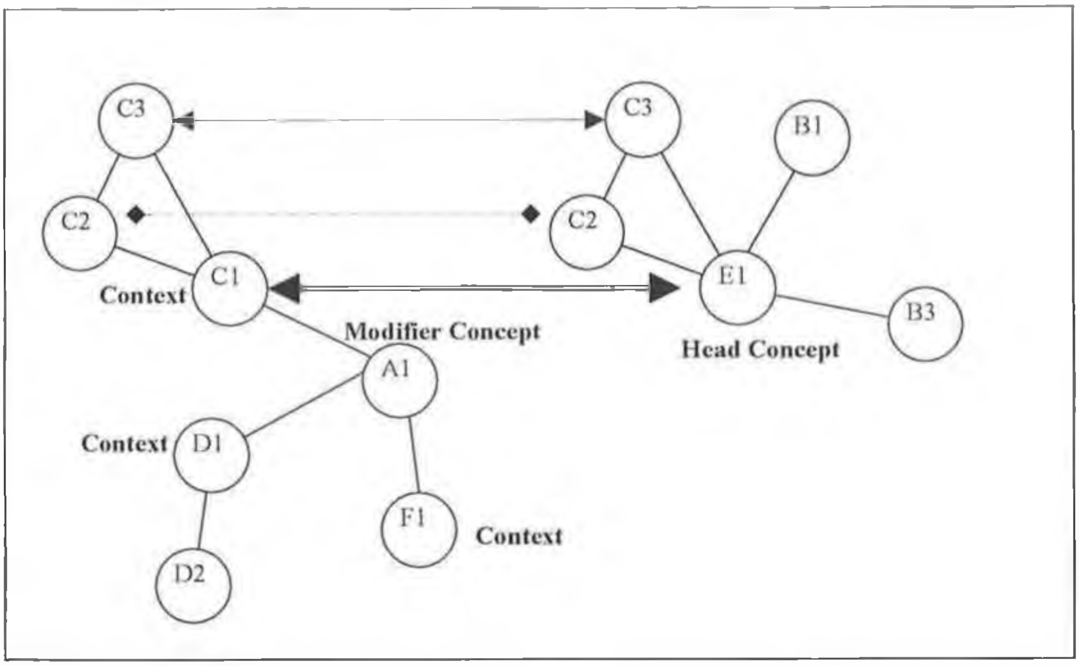


Figure 5.14 - Alignment schema for contextual combination type

5.3.4 Insertion Combinations

In the insertion combination type the modifier concept refers to an aspect of the head concept that the modifier concept then replaces. This is similar to the slot-filling approach of a number of models, e.g. Selective Modification (Smith et al., 1988) and Concept Specialisation (Murphy, 1988). In Figure 5.15, there are two representations, one for the concept *FUR* the other for the concept *COAT*. A suitable aspect of the head concept has to be found and replaced. This is done by mapping the modifier concept onto every concept linked to the head concept: *SLEEVES, GARMENT, CLOTH, PROTECTION, POCKETS,* and *BUTTONS*.

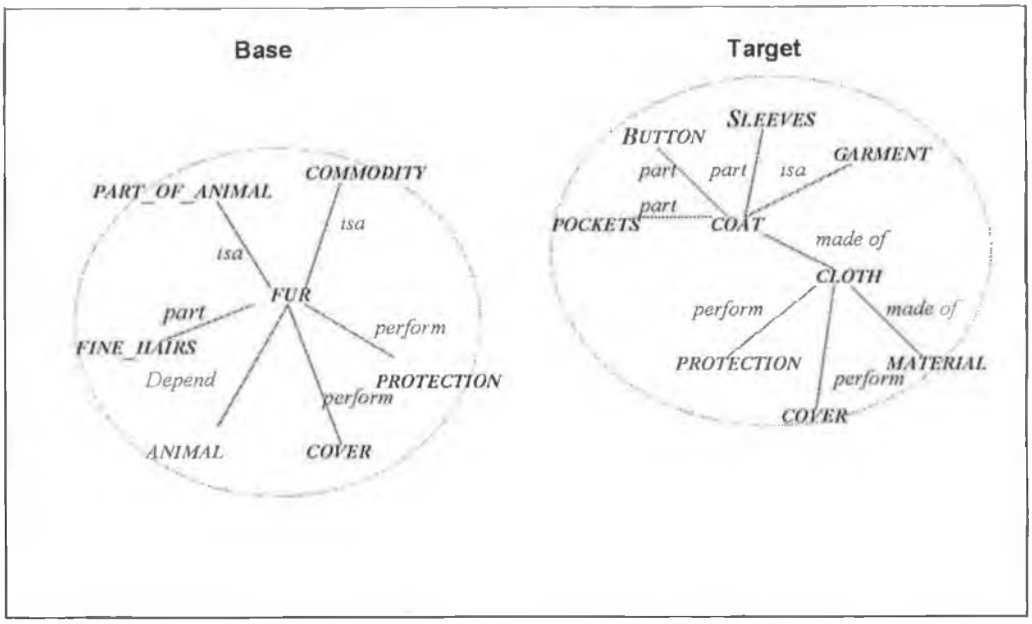


Figure 5.15 - Representation of *FUR* and *COAT*

The aspect which will be replaced is found by discovering a similar subconcept. In this case *COAT* has been represented as being made from *CLOTH*. This is more similar to *FUR* than the other aspects of *COAT* (see Figure 5.15). We envisage that, depending on the representation used, this similarity will either be a deep similarity or a shallow one, although a deep similarity would be preferred for generating an interpretation. Consider that in Figure 5.16 there is a shallow similarity between *FUR* and *CLOTH*. However, in Figure 5.17 there is a deep similarity between *FUR* and *CLOTH* based on a shallow similarity between fine-hairs and fibres. We would argue that the interpretations based on shallow similarities are more common, since to find a deep similarity there has to be an already existing shallow similarity between a subconcept of the modifier concept and a subconcept of an aspect of the head concept. Comparing the two Figures 5.16 and 5.17 there is clearly some similarity between cloth and fur. This is reinforced by the deep similarity in Figure 5.17.

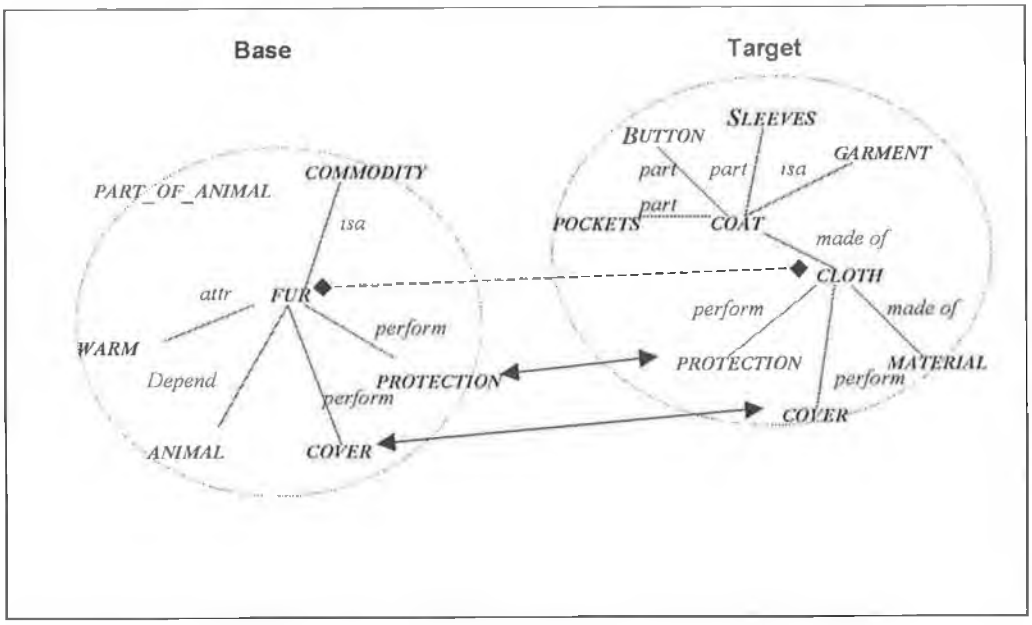


Figure 5.16 - Mapping between *FUR* and *CLOTH* based on a shallow similarity

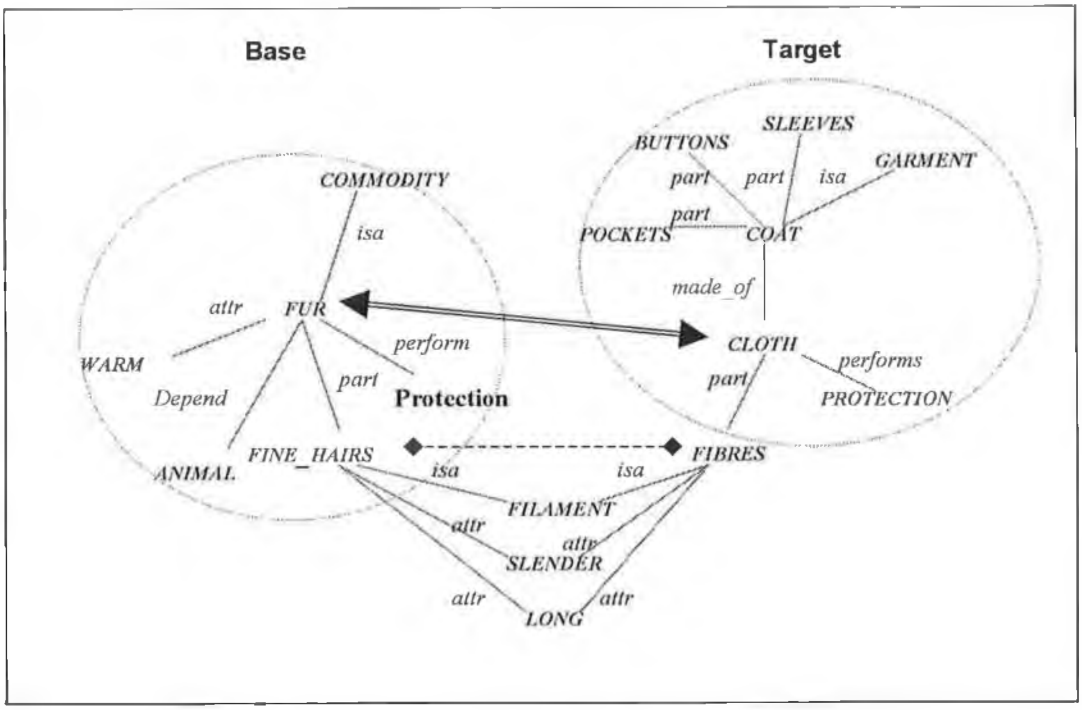


Figure 5.17 - Mapping between *FUR* and *CLOTH* based on a deep similarity

Whether an interpretation is based on a deep similarity or shallow similarity the modifier concept then replaces the most similar aspect of the head concept in the exact same way. This will result in a new combination being created (see Table 5.4). In Table 5.4 the slot value of *made_of* has been replaced by *FUR*.

Table 5.4 - Representations of 'fur coat' and *COAT*

COAT		'fur coat'	
Isa	*GARMENT*	Isa	*GARMENT*
Part	*GUTTONS,* *SLEEVES,* *POCKETS*	Part	*BUTTONS,* *SLEEVES,* *POCKETS*
Made_of	*CLOTH*	Made_of	*FUR*

Abstractly this combination type involves replacing an aspect of the head concept with the modifier concept. This aspect is found by aligning the modifier concept with every subconcept of the head concept and finding the subconcept with the most similarities with the modifier concept, this similarity will either be a deep similarity or a shallow similarity. In Figure 5.18, the modifier concept has a number of matches with B4 in terms of either a shallow similarity or a deep similarity. This provides a suitable aspect for the modifier concept to replace. When the modifier concept replaces B4 it replaces B4 and its substructures. A new structure is created. Thus, a situation such as Figure 5.18 arises.

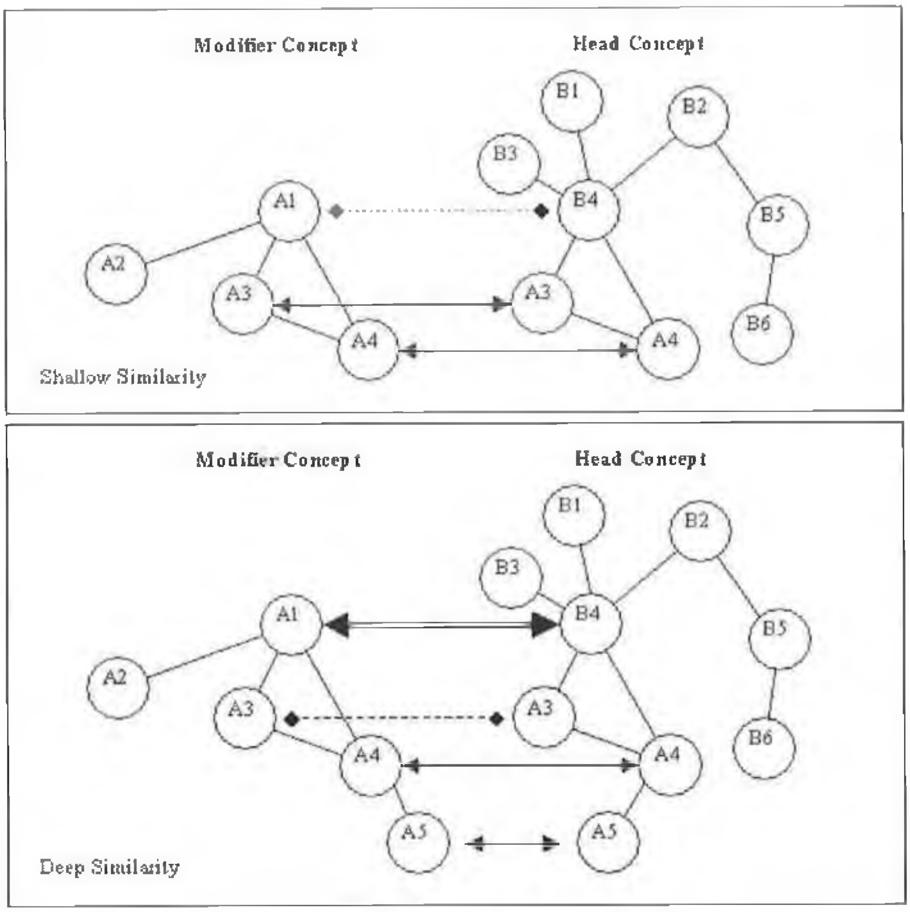


Figure 5.18 - Alignment schema for insertion combination types

In Figure 5.19 the modifier concept actually replaces part of the head concept. However, the original head concept is kept in the network and a new concept is added, one that reflects the new structure created by the insertion combination. Figure 5.18 shows the creation of a new concept based on shallow similarity as well as deep similarity.

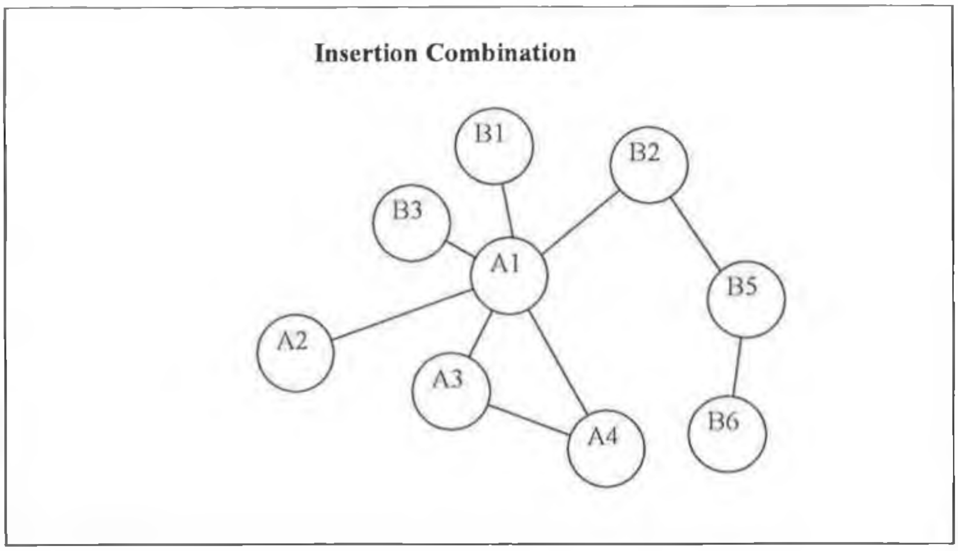


Figure 5.19 - The result of an insertion combination

5.3.5 Context Insertion Combinations

In this combination the modifier concept refers to a context which is then inserted into the head concept. Compared with the last combination we outlined (the insertion combination) there is a further step, that of drawing in a suitable context. Remember thought that we found two scenarios where an insertion combination could be found, one via a deep similarity and one via a shallow similarity. The finding of a context is an extra stage on top of the interpretation of the insertion combination. Consider the combination 'mink coat'. In this particular combination an aspect of *MINK* is aligned with the an aspect of *COAT*. This process would involve taking every subconcept of the modifier concept and aligning it with every subconcept of the head concept.

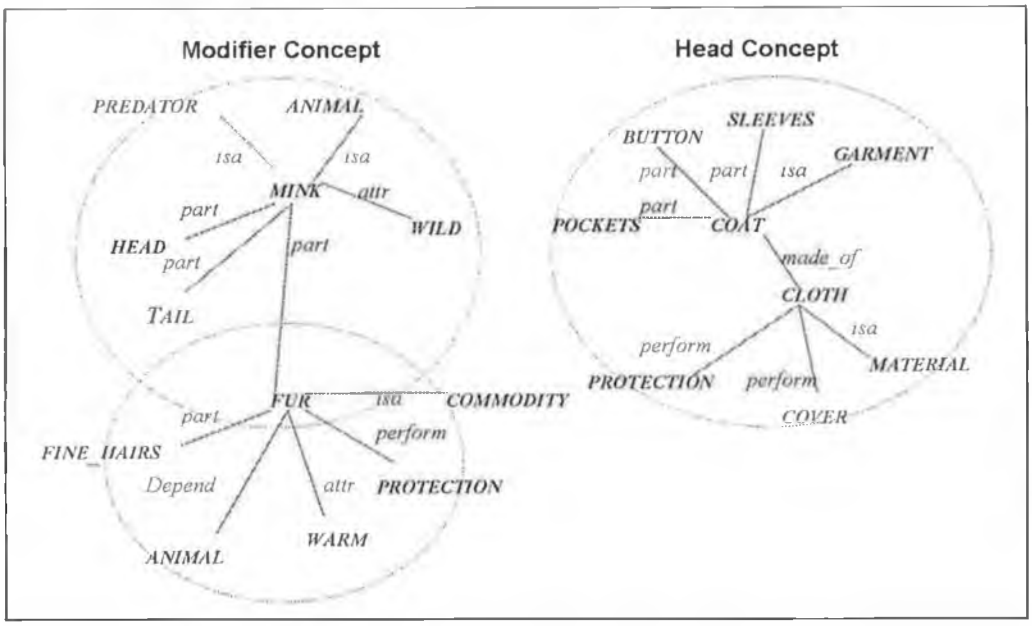


Figure 5.20 - Representation of concepts in 'mink coat'

In Figure 5.20 there are really three concepts represented: *MINK*, *FUR* and *COAT*. Every subpart of *MINK* is compared with every subpart of *COAT*. As was stated in Section 5.3.4, the concept *FUR* shares a number of similarities with one aspect of *COAT*, *CLOTH*. The similarities were shown in Figure 5.15. Thus in comparing a context insertion combination and an insertion combination the only significant difference is that the modifier concept refers to a context which then acts as an insertion combination. These type of combinations can be paraphrased in terms of the context, a 'mink coat' is really shorthand for 'mink fur coat'.

This combination type can be examined more abstractly in terms of SA. In Figure 5.21 the nodes numbered from E1 to E4 represent the modifier concept. In a context insertion combination type a context of the modifier concept is aligned with the head concept. This context will be a node that is linked with the modifier concept or is part of the modifier concept. This context is aligned with the head concept where it replaces the aspect of the head concept with which it is the most similar. In this example the context forms a shallow similarity with the part of the head concept it replaces but in a context insertion combination this replacement could also be based on a deep similarity. In a case where rival interpretations are based on a shallow similarity and a deep similarity, the deep similarity would be preferred.

The new information that this combination type adds to the network is a compound. This compound is essentially the structure of the head concept with the context replacing the aspect of the head the context was aligned with. For example, in 'fur coat' the cloth aspect of

COAT would be replaced by *FUR*. This is shown in Figure 5.22 where the head concept of Figure 5.20 has had the concept *CLOTH* replaced by the concept *FUR*. The modifier concept remains unchanged, as does the head concept. However, a new entry is created in memory for 'fur coat'.

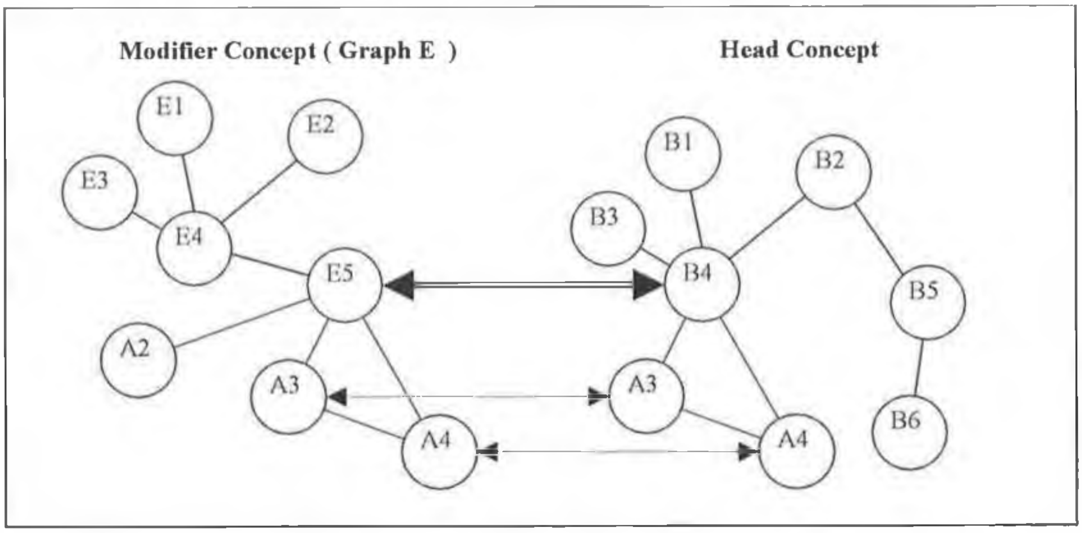


Figure 5.21 - Alignment schema for context insertion combination type

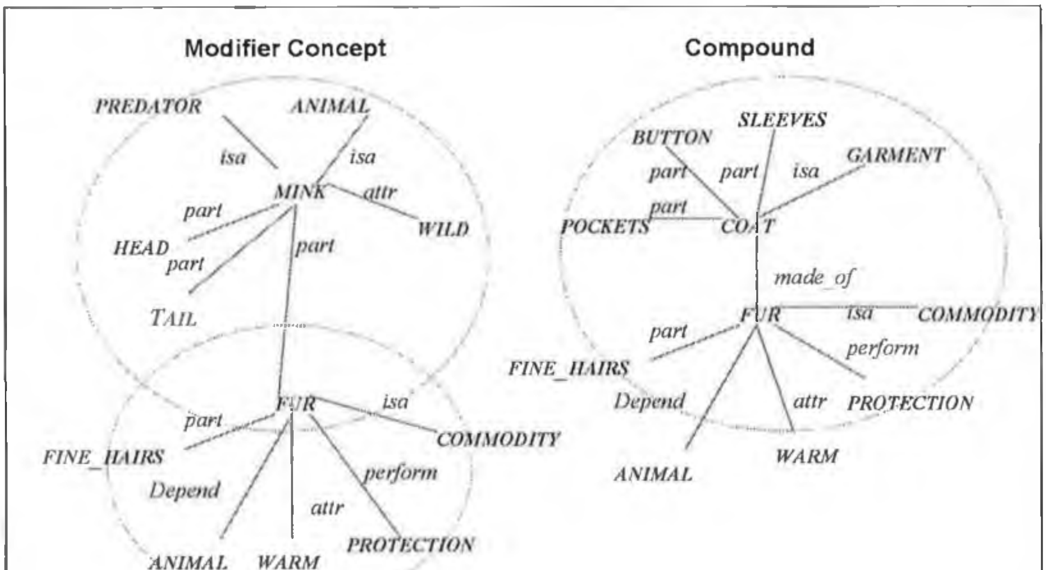


Figure 5.22 - Creating the compound: 'mink coat'

5.3.6 Conjoined Combinations

This combination is the most difficult to analyse in terms of structural alignment. It involves establishing an equivalence between the modifier concept and the head concept. It is probable that this equivalence is based on shared category membership. For example, a 'servant girl' which is interpreted as >> a person who is both a girl and a servant << would be both a *PERSON* and an *ANIMAL*. In the case of 'house boat' both concepts are *STRUCTURES*. Moreover though, it seems that there cannot be a conflict between the modifier concept and the head concept in this combination type. The lack of a conflict could be marked by a number cross-domain mapping(s) between the modifier concept and the head concept. The mappings could be of two types: deep similarities or shallow similarities. So perhaps where little or no mappings can occur no conjoined interpretation is possible. Figure 5.23 displays a representation of 'house boat'. The concept *HOUSE* and the concept *BOAT* are both *STRUCTURES*. This is a shared category. A subpart of each concept also share similarities, *FLOOR* and *DECK* share a number of similarities. We suggest that these similarities are either shallow or deep similarities. The 'house boat' example involves a deep similarity.

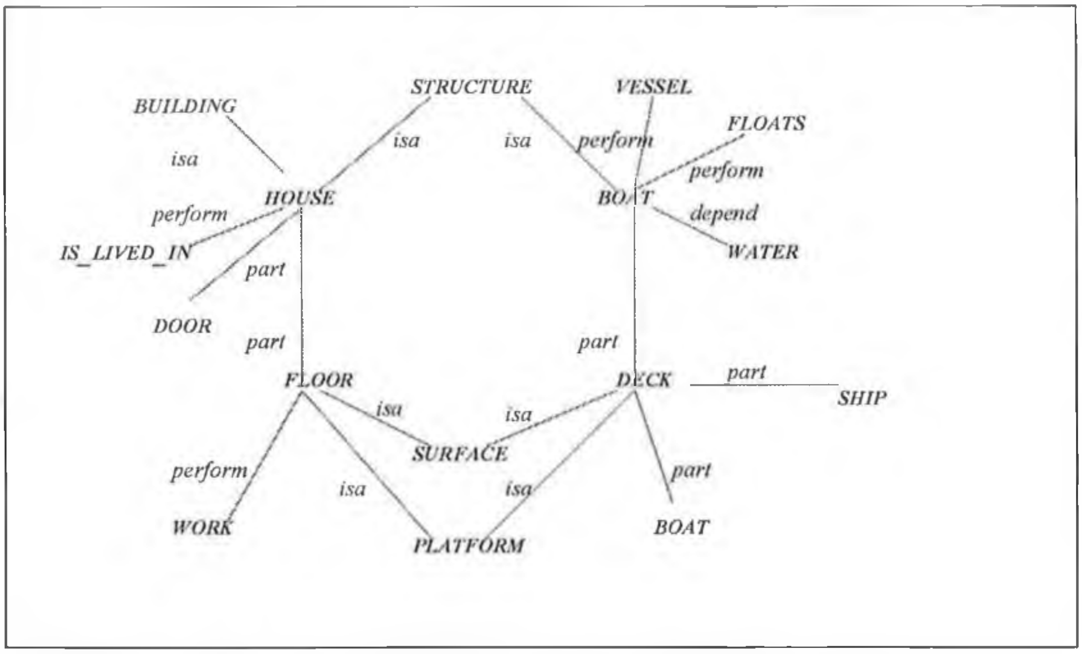


Figure 5.23 - Representation of concepts in 'house boat'

The structure of the representation in Figure 5.23 is similar to that of Figure 5.13. This suggests that the combination 'drug dealer baron' could also be interpreted as >>a person who is both a drug dealer and a baron<<. Although the existence of such a person is unlikely the interpretation is still understandable. This is just one place where polysemy can arise. Polysemy and the other two problems of concept combination are examined in next section.

The abstraction of the conjoined combination type is given in Figure 5.24. The nodes A1 and B1 share a common category, C. In a conjoined combination type the concepts share a common parent. We also suggest that a subconcept of both the modifier and the head should share a deep similarity. In Figure 5.24 the nodes A1 and B1 form a deep similarity based a shallow similarity between A3 and B2. We have not shown the mapping but have marked out some of the shallow similarities, i.e. nodes D and E.

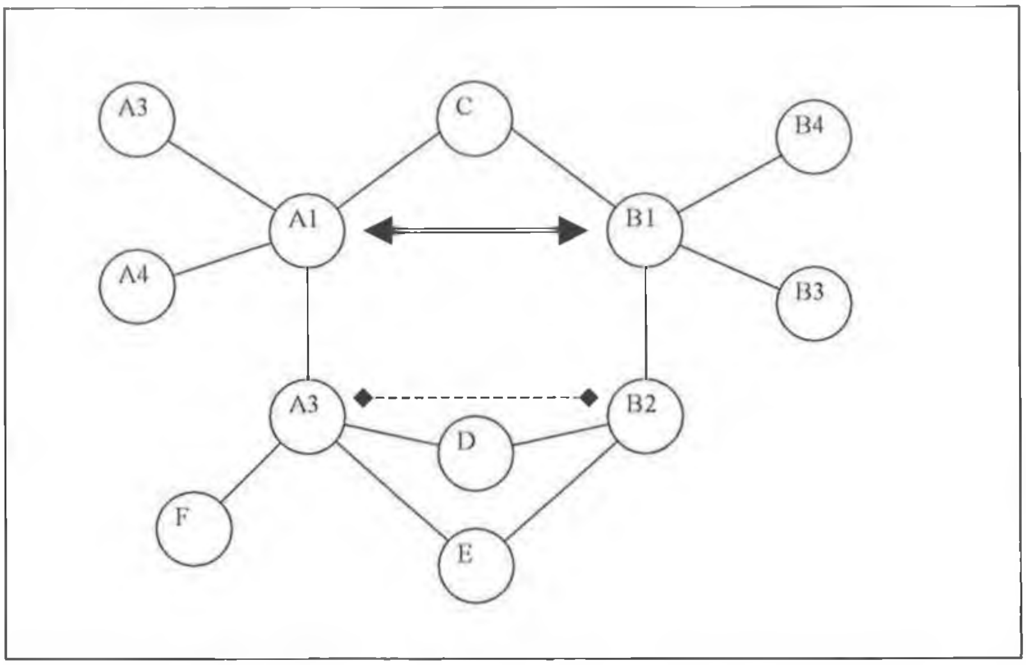


Figure 5.24 - Alignment schema for the conjoined combination type

5.3.7 Summary

The relationship between the combination types and the types of similarity are summarised in Table 5.5 Where a combination is followed by the phrase "(or Deep similarity)" in Table 5.5 we are suggesting that the finding of a deep similarity can also give rise to an interpretation. Only one combination is based solely on deep similarity and that is the contextual combination. This is interesting, as it is the most like nominal metaphor of any of the combinations. Deep similarities are used in nominal metaphors to give rise to interpretations (Veale, 1995). We would expect combinations based on deep similarities to give rise to figurative interpretations. Yet, with some combination types, e.g. the insertion combination type, a deep similarity does not necessarily result in a figurative interpretation. In addition to this, the catachrestic combination type which will always give rise to a figurative interpretation is based on a shallow similarity, not a deep similarity. The distinction between literal and figurative interpretations of combinations is not clear-cut when based on assumptions about the roles of either shallow or deep similarities.

Table 5.5 - Combination types and the similarity types

Combination type:	Interpretation based on:
Meronymous	Shallow similarity
Conjoined	Shallow similarity (or Deep similarity)
Insertion	Shallow similarity (or Deep similarity)
Context insertion	Shallow similarity (or Deep similarity)
Catachrestic	Shallow similarity
Contextual	Deep similarity

5.4 Dealing With The Three Problems of Concept Combination

So far we have not explicitly said how our model can deal with the three problems in concept combination we outlined in Chapter 1. The three problems are:

1. Accounting for world knowledge.
2. Accounting for polysemy.
3. Accounting for figurative combinations.

We have suggested that all these problems can be overcome by using SA. Let us deal with each of these problems.

5.4.1 Accounting For World Knowledge

In our model world knowledge is not set out as, e.g. a naïve physics, rather it is accounted for by involving associated concepts. Firstly, rich conceptual structures are used, secondly, all concepts are treated as occurring in a wider domain, with aspects of this domain being drawn in as context. This model has worked relatively well in the field of metaphor where the interpretations that models have to generate are often extremely complex, e.g. Sapper (Veale, 1995).

Two of the combination types, the context insertion combination and contextual combination, explicitly attempt to find a context (another concept) that can be drawn in to flesh out the interpretation. The way in which these contexts are found assumes that all the concepts exist within some type of semantic network memory. A context is found by looking at the links between concepts in the semantic network. Consider the combination 'computer wizard'. An interpretation of this combination is >> a computer user with good computer skills<<. Note that *COMPUTER* seems to refer to 'computer user'. There are in fact many concepts that *COMPUTER* could refer to, e.g. 'computer company', 'computer printer', 'computer programming', 'computer language'. When interpreting 'computer wizard' which of the various concepts that *COMPUTER* refers to is actually used to generate an interpretation? The assumption that structural alignment theory makes in relation to suitable contexts is that the modifier concept will in some way be similar to the head concept. In 'computer wizard', whatever context the modifier concept, *COMPUTER,* refers to, it is likely to be similar to the head concept, *WIZARD*. This similarity will be based on cross-domain mappings.

In our model world knowledge is brought into play in terms of retrieving appropriate contexts. Treating concepts as graphs, contexts are retrieved by examining the connection between nodes. Any node that is linked to another node may be a possible context. These connections between nodes will in reality take two forms: they will either link to parent concepts or to child concepts. If a concept A is linked to a concept B by an *isa* relation, i.e. A is a type of B, then B can be considered a parent of A. When a concept A is linked to a concept B by a *part* relation, i.e. B is part of A, then B can be considered a child of A. In terms of the combination types, the context insertion combination appears to largely draw in parent concepts, i.e. concepts which the modifier concept is a subconcept of. The contextual combination type appears to rely on child concepts, i.e. subconcepts of the modifier concept. This is a point that could be useful when computationally treating these types.

Some combination types involve insertion into the head concept. In the insertion combination type the modifier concept is inserted into the head concept whereas in the context insertion combination type a concept related to the modifier concept is inserted into the head concept. Fitting one part of a concept into another is done purely in terms of the structure of the two concepts. Consider the example of 'fur coat' again. This combination is understood by inserting the modifier concept into the head concept. The point of insertion is found in terms of cross-domain mappings. Structural alignment theory suggests that the modifier concept replaces that aspect of the modifier concept with which it is the most similar.

Reasoning about the integration of concepts is dealt with purely in terms of SA. No information other than the representation of concepts is used. There are no separate but associated scenarios (as e.g. in Dual Process Theory) or extra functional models. What aspect is to be inserted into what is dealt with using cross-domain mappings. What context is the most appropriate is found by aligning possible contexts with the head concept and picking the one which gives rise to cross-domain mappings. In a sense, all reasoning is dealt with in terms of the structure of the concepts. Using SA, we have a powerful tool for accounting for world knowledge.

5.4.2 Accounting For Polysemy

Combinations can be polysemous; they can have more than one interpretation. There are three ways in which polysemous combinations can occur within our model. These are:

1. A combination fits into two or more combination types.
2. A combination can be interpreted in similar but distinct ways in terms of the same combination type.
3. A combination involves a mixture of (1) and (2).

As the list above suggests, lexical ambiguity is not considered here. In general, theories of concept combination assume that concept combination occurs using either the most prevalent sense or the sense that has been marked out by the context. As our model attempts to work within the area of concept combination research, we also adopt this view. This topic is discussed further in Section 7.5.5 in relation to the major findings of this thesis.

The first form of polysemy listed above is that a combination may fit into more than one combination type. Consider the combination 'robin snake'. This could be interpreted as a >>snake that hunts robins<<, or as >>a snake with a red underbelly<<. The first interpretation would result from treating 'robin snake' as an insertion combination. The second interpretation would result from treating 'robin snake' as a context insertion combination. Indeed, more bizarre interpretations can be generated, e.g. >> an animal that is both a robin and snake << which would be an conjoined combination type.

The second form of polysemy involves a combination that gives rise to slightly different interpretations of the same combination type. This could happen in the insertion combination type if the modifier concept matched more than one aspect of the head concept. Then interpretations based on each of these aspects could be generated. For example, if a 'bumblebee moth' is interpreted as >> a moth with a sting << or >> a moth with black and yellow stripes <<, these would both be context insertion based interpretations. The difference between the two is the context, in one it is a *STING*, in the other is the *STRIPES* of the *BUMBLEBEE*.

The third form of polysemy is combination of the first two. This type of polysemy is in theory possible but we can offer no examples. We suggest that the first type of polysemy is probably the most likely. The second might only occur with certain combination types, e.g. insertion combinations and contextual combinations.

5.4.3 Accounting For Figurative Combinations

In our model figurative language is accounted for in the exact same way as literal language. The model makes no distinction between the two at the semantic level as structural alignment is assumed to be able to account for all combinations. The separation of combinations into literal and figurative combinations is assumed to be done after the process itself. However, looking at the classification of combination types it is clear that some by their definition will give rise to figurative interpretations: contextual combinations and catachrestic combinations.

Both the contextual combination type and the catachrestic combination type will always be figurative. In the first type the modifier concept refers to a context with is then aligned with the modifier concept, e.g. 'drug baron'. This alignment is very similar to metaphor. If the context for *DRUG* in 'drug baron' is 'drug dealer' then we are suggesting an interpretation such as >>drug dealer is like a baron<<. In the catachrestic combination the head concept is a metaphorical reference to an aspect of the modifier concept and so is by definition figurative, e.g. 'table leg'. Both combination types give rise to combinations that can become fossilised and they may not often be perceived as being figurative.

Our approach to concept combination applies the same mechanism of SA to literal combinations. The following combination types should give rise to literal interpretations: meronymous combinations, insertion combinations, context insertion combinations and conjoined combinations. All these combinations are understood in terms of SA. An insertion combination works by aligning the modifier concept and head concept and finding a suitable aspect of the head concept to replace. The suitable aspect is found by finding that aspect of the head concept which will provide the most cross-domain mappings with the modifier concept. In short, in our model both literal and figurative interpretations can be accounted for using SA.

It may appear that any deep similarity will give rise to a metaphorical interpretation. This is not really the case. Although both catachrestic and context insertion combinations involve deep similarity, deep similarity does not in itself make interpretations metaphorical. One example in the last chapter referred to similarities between the concepts *SCALPEL* and *CLEAVER*. A deep similarity was found between the two but to say a *SCALPEL* is like a *CLEAVER* is not figurative. In a larger context of "my surgeon is a butcher" this similarity may be seen as figurative.

5.5 Concept Combination As A Process

So far we have outlined six combination types and suggested how SA can account for the three problems of world knowledge, polysemy and figurative language. Now we describe how the process of concept combination works according to our model. Taking a combination such as 'robin snake' we argue that there are several interpretations of this combination. The alignment process will attempt to see if the prospective combination can fit into any of the combination types. The order in which the combination is checked against the combination types is assumed to proceed in from the computationally simplest to the most complex. In general the order will be:

1. Meronymous
2. Conjoined
3. Insertion
4. Context insertion
5. Catachrestic
6. Contextual

Regardless of the order, the process will generate several interpretations if such are available. These interpretations have to be ranked. Ultimately, the process of concept combination can give rise to several interpretations and subjects seem to able to rank interpretations. We suggest that given several interpretations, subjects will favour the most simple over the most complex. The most complex in this context will be those interpretations that involve a great deal of mapping. We suggest that people will favour interpretations in the same order that we have suggested that combination occurs in, i.e. meronymous first.

The principle behind this assumption is that subjects are likely to favour interpretations that they first come across. In discourse, interpretations must be made quickly and the simplest should be found first. However, it is probably the case that when interpretations have to be ranked, not every combination type will be reflected in the interpretations under consideration. The ranking will probably never involve all six combination types, as it is hard (if not impossible) to imagine a combination that could be interpreted in six different ways according to the combination rules.

Ultimately, concept combination in the structural alignment view of concept combination sees the combination process as one of conceptual integration. When presented with a noun-noun compound a subject will try to fit the conceptual structure of the compound into each of the combination types. If a fit is found, then an interpretation is generated. If more than one interpretation is found, we hypothesise that the most simple interpretation will be the one that is ranked as being the most appropriate.

5.6 Towards An Implementation Of The SA View Of Concept Combination

The model we have outlined should lend itself (easily) to the development of a computer program. Each of the combination types can be viewed in terms of a schema or a template which describes how structural alignment works for that particular combination. The approach lends itself to the creation of a rule-based mechanism. When a combination is processed it can be checked against the alignment templates for each combination type. If a combination satisfies one of these templates, then an interpretation is returned. Each template can be used as a rule and the order of the application of these rules will reflect the order given in the last section, where the simplest is tried first.

This rule-based approach will, at first glance, add a certain rigidity to the model. But we suggest that the range of the combination types will give enough cover to produce good interpretations. "Good interpretations" are, in this context, interpretations that subjects would generate. Our approach will be outlined in detail in the next chapter when the INCA (Interpreting Nominal Compounds using Alignment) program is outlined.

5.7 Chapter Summary

We outlined our approach to concept combination, the structural alignment view. This view is substantially different from Dual Process Theory, the first model to introduce the application of SA to concept combination. Our model suggests that at the semantic level there are six types of combination: meronymous, catachrestic, contextual, insertion, conjoined, context insertion. These combination types range from those that are referential to those that are the constructive.

The process of concept combination is a process of conceptual integration. When presented with a noun-noun compound a subject will try to fit the conceptual structure of the compound into each of the combination types. If a fit is found, then an interpretation is generated. If more than one interpretation is found, the most simple interpretation will be the one that is ranked the most appropriate.

Each of the combination types was outlined in terms of how they operate in relation to SA. This outline will form the basis for a rule-based computer program (INCA) presented in the next chapter which implements the model presented this chapter.

Chapter 6 - INCA: Interpreting Nominal Compounds using Alignment

6.1 Introduction

In this chapter an implementation of the structural alignment model outlined in Chapter 5 is presented. This implementation is dubbed INCA: interpreting nominal compounds using alignment. INCA is a computer program and employs a knowledge base or semantic network to represent concepts. The knowledge base also includes information on the potential similarity between all the representations of concepts in the knowledge base. In the previous two chapters we have mentioned deep and shallow similarity. Our knowledge base contains information on the deep and shallow relationships between all the representations of concepts in the knowledge base.

There are two algorithms at the heart of INCA, each involving the testing of an input against rules which model each combination type. One algorithm searches for all interpretations (the order of types is not important for this algorithm), the second returns the simplest interpretation and only this interpretation (the order of types is important for this algorithm). Each of the combination rules is briefly explained and the algorithmic design of INCA is then set forth. A sample interaction with the program is then given. INCA is user driven: the user poses queries to the system and INCA attempts to return several interpretations based on the combination rules. We conclude this chapter by discussing possible ways of testing INCA.

6.2 INCA

The structural alignment model of Chapter 5 is here implemented as a computer program, named INCA. It is implemented in PROLOG, a language that has proved popular among European AI and natural language processing researchers (Bratko, 1990). PROLOG allows (procedural) knowledge to be represented in a declarative fashion. The outline of INCA will be broken into five parts:

1. Representing concepts
2. Building the knowledge base
3. Modelling the combination types
4. Returning an interpretation
5. Algorithmic design

This breakdown will include some sample code (in PROLOG). Further sample code for INCA can be found in Appendix C. The INCA system operates over a knowledge base which represents the structure of various concepts. The program employs a number of combination schemas implemented as logical procedures which are used to generate interpretations. The user can interact directly with INCA by posing different queries. When a query to interpret a combination is posed, the system attempts to find as many interpretations as it can for the combination, unless a special "simplest interpretation first" mode is used (we assume that this will be an informative interpretation and this assumption is examined in the next chapter).

6.2.1 Representing Concepts

Concepts in INCA are represented via frames. Frames are made up of slots and fillers. These slots and fillers in turn are represented via PROLOG assertions that take four arguments. This structure can capture many relations and properties. The basic form is as follows: relation(concept1, concept2, relation, weight). Figure 6.1 below shows an example of such an assertion which states that the concept *DRUG is a commodity*. The fourth argument is a rating of diagnosticity for the property or relation and these are always set to 0.5 by default. Initially no property or relation is marked as being more diagnostic than another. However, when the entire network has been set up, the diagnosticity scores for each property can be worked out. These scores are not used in the current implementation of INCA.

relation(drug, commodity, isa, 0.5)

Figure 6.1 - Example relation

Each concept is thus represented as a collection of relation/4 assertions.[1] In Table 6.1 the representation of the concept *ROBIN* is provided, together with an equivalent paraphrase in English for each relation. *ROBIN* is considered a concept as it occupies the first argument in several relation/4 assertions. Other concepts may link (using relation/4) to *ROBIN* but as robin will be listed as the second argument these links are not considered part of the concept *ROBIN*. (This is the case for all concepts, e.g. like *TELLER*, in Section 5.3.1). In general for every concept the most common and typical attributes are listed. This information usually involves the categories that a concept belongs to (via the *isa* relation), the parts the concept comprises, and the functions that are associated with a concept. There are a core number of relations that are used. In INCA these relations are not exhaustive but sufficiently high-level to be of use in many domains of interest. The relations are listed in Figure 6.2.

[1] The /4 marks the arity of the assertion, so relation/4 is a PROLOG assertion that has four arguments.

Table 6.1 - Example representation of the concept robin

Relations	English equivalents
relation(robin, bird, isa, 0.5)	A robin is a bird.
relation(robin, small_animal, isa, 0.5)	A robin is a small animal.
relation(robin, animal, isa, 0.5)	A robin is an animal
relation(robin, flying, perform, 0.5)	A robin can fly.
relation(robin, red_breast, attr, 0.5)	A robin has a red breast.
relation(robin, wing, part, 0.5)	A robin has a wing.
relation(robin, breast, part, 0.5)	A robin has a breast.
relation(robin, beak, part, 0.5)	A robin has a beak.

isa (X is a type of Y)
attr (X is an attribute of Y)
part (X is a part of Y)
used_for (X is used for Y)
effect (X has an effect on Y)
depend (X depends on Y)
control (X controls Y)

Figure 6.2 -Types of relations in the system

6.2.2 Building The Knowledge Base

The knowledge base mainly consists of relation/4 assertions that represent the component parts of concepts. However, other information is also stored in the network capturing the similarities that hold between concepts. These are stored in the form of two types of assertions: shallow_similarity/3 and deep_similarity/3. We will now discuss the computation of these assertions which capture the similarities between concept representations and reflect our understanding of the commonalities that drive structural alignment as set out in Chapter 4.

Firstly, a commonality is asserted wherever two concepts share a third concept as a property via the same relation. In Figure 6.3, C is this third concept, as it exists in both concept A and concept B and so is a point of similarity between the two. Figure 6.3 also suggests that concept A and concept C should not be the same, as is also the case with concept B and concept C (otherwise the similarity process is trivialised). Two versions of the commonality rule are shown in Figure 6.3, commonality/3 returns the point of similarity as a third argument.

```
commonality(A, B) :-
        relation(A, C, F, _),
        relation(B, C, F, _),
        A \== C,
        B \== C.
```

```
commonality(A, B, [C, F]) :-
        relation(A, C, F, _),
        relation(B, C, F, _),
        A \== C,
        B \== C.
```

Figure 6.3 - Commonality rules

Shallow similarity is based directly on commonalities and Figure 6.4 lists the shallow_similarity/3 procedure as defined in INCA. As the definition in Figure 6.4 suggests shallow similarities are sets of commonalities. We suggest that there must be at least two commonalities for a shallow similarity to be found. This threshold can be changed, but for the purpose of this thesis it is set to two, as anything less than this could give rise to similarities that may not be useful for our purposes.

Shallow similarity in INCA:

```
shallow_similarity(A, B, L) :-
        setof(X, commonality(A, B, X), L),
        length(L, Num2),
        Num2 > 1.
```

Figure 6.4 - Shallow similarity rule

Deep similarity can be seen as an implied correlation arising from a richness of surface similarities. In INCA a deep similarity is found by seeing if any shallow similarities or deep similarities exist between two concepts (see Figure 6.5). As was discussed in Section 4.1, any two concepts can in theory have a number of similarities. To take account for this, a threshold is can be set where a deep similarity is not found unless there is at a given number of shallow similarities between the concepts being compared. At present we have set this threshold to two or more shallow similarities.

Two concepts share a deep similarity if they link to the same concept with the same link name.

```
deep_similarity(A, B, L) :-
        relation(A, C, F, _),
        relation(B, D, F, _),
        similar(C, D, R),
```

Where

```
similar(X, Y, Z) :-
        shallow_similarity(X, Y, Z).

similar(X, Y, Z) :-
        deep_similarity(X, Y, Z).
```

Figure 6.5 - Deep similarity rule

These rules give rise to the similarity facts that are computed and then asserted into the knowledge base. They are applied as the INCA program is loaded, rather than being applied on-the-fly as the program runs. Indeed, when the system is first loaded all the original relation/4 and the computed deep_similarity/3 and shallow_similarity/3 facts are asserted into the PROLOG database. This ensures a faster processing at run-time when a query is posed to the program.

6.2.2.1 Properties of The Knowledge Base

In this section a number of the properties of the knowledge base are set out. Firstly, the knowledge base is quite flat; by that it is meant that if the *isa* relation is used to analyse the network in terms of a tree there are a small number of levels in that tree. In the sample run-through in Section 6.3 there are only 4 levels. In this knowledge base the average concept will most likely have one parent but have several siblings. The representations of concepts in the knowledge base are created solely around relations and properties. This is in contrast with other models, e.g. WordNet (Miller, Beckwith, Fellbaum, Gross and Miller, 1993), which are built around hyponymous, meronymous, antonymous and synonymous relations that hold between concepts. Meronymous relations exist in the INCA network but they are not used to structure the network. In fact if the representations used in the knowledge base are compared with WordNet our focus is on attributes, an aspect that is for the most part downplayed by WordNet (or at least confined to the glosses).

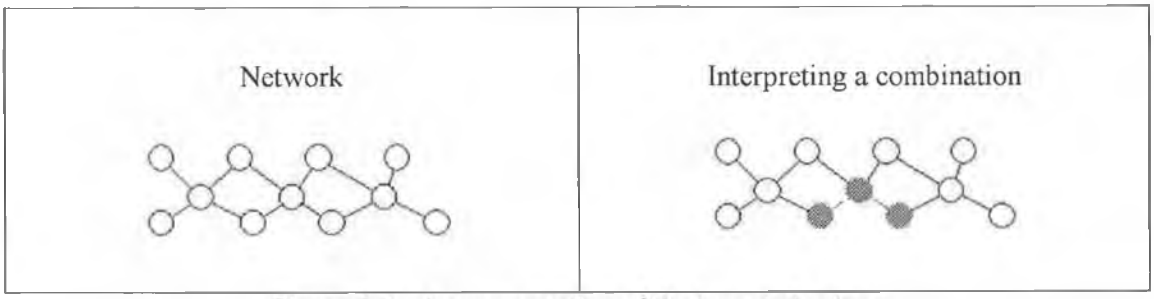


Figure 6.6 - A representation of the knowledge base

Figure 6.6 shows a simplistic representation of the network where concepts are represented by circles, and these concepts are connected by links to each other. In reality the representation of concepts in the knowledge base is highly interconnected, but this is difficult to represent visually. Figure 6.6 shows that concepts in the knowledge base are quite closely related while the network itself is quite flat. Owing to this property of the network, we view concept combination as a process that finds particular groups of links (hence, the structural schema in Chapter 6) between concepts. In Figure 6.6 a combination is interpreted by finding particular links between concepts, this group of links and concepts are marked out in bold. We have proposed six ways conceptual structures can be linked together, namely the combination types. This leads us to implement INCA as a rule-based program that attempts to find the links between concepts that best match the structural schema with the contents of the knowledge base.

6.2.3 Modelling The Combination Types

The last chapter suggested that there are six types of combination, and for each of these combinations an alignment schema was suggested. These schemas have been implemented using logical procedures in PROLOG in a declarative fashion This approach is highly dependent on the knowledge base. This also means that if written correctly (in terms of PROLOG), the logical procedure that can interpret combinations can also be used to generate combinations that will satisfy the procedure.

Initially, each combination type was designed to handle at least one paradigmatic example. For example, the insertion combination type was developed with respect to the combination 'mink coat'. Concepts were thus created for *MINK*, *COAT* and other related concepts. The idea behind this was to model a combination type in relation to a paragon of that same combination type. Subsequently, the combination rules were tested with examples that fell outside the original specific examples. New combinations were introduced into the knowledge base and the combination rules were tested with these new concepts.

6.2.3.1 Meronymous

The implementation of the meronymous combination type can be summarised as the question: does the head concept exist within the modifier concept? There are at least two ways of finding this out. The first is by seeing if a part of the modifier concept is identical with the head concept. Identicality consists of every part of one concept aligning with every part of another concept. All these correspondences would be shallow similarities. A rule is shown in Figure 6.7 for this combination type. The second way is a shortcut (cheat) that sees if the head satisfies the following relation: relation(Modifier, Head, Part, _). We refer to this as a cheat since it does not rely on structural alignment.

```
Meronymous(A, B) :-
        subconcept(A, C),
        identical(B, C),
        relation(A, B, _Rel, _W).
```

Figure 6.7 - Basic rule for the catachrestic combination type in INCA

The actual interpretation given to the user is based on finding out what the relationship is between the head concept and the modifier, within the head concept. In Figure 6.7, *relation(A, B, _Rel, _W).*, is used to find the relationship between the modifier and the head. If the combination was 'snake fangs' and *FANGS* was already part of *SNAKE* then the relationship would be based on *part*. This relation would be returned as part of the interpretation.

6.2.3.2 Conjoined

In Section 5.3.6 we suggested that a conjoined combination may be based on shared category membership. This approach is implemented in INCA. Firstly, we see if there is a related context that both the concepts share. This related concept is assumed to be the shared category. If such a shared category is found then there should also be at least one deep or shallow similarity between one of the subconcepts of the modifier and the head. This is the idea behind the code in Figure 6.8.

conjoined(Base, Target) :- *context(Base, Scenario),* *context(Target, Scenario),* *subconcept(Base, C),* *subconcept(Target, C1),* *deep_similarity(C, C1, _).*	*conjoined(Base, Target) :-* *context(Base, Scenario),* *context(Target, Scenario),* *subconcept(Base, C),* *subconcept(Target, C1),* *shallow_similarity(C, C1, _).*

Figure 6.8 - Basic rule for the conjoined combination rule

6.2.3.3 Insertion

In the insertion combination type the modifier concept is inserted into the structure of the head concept. This is done by finding an aspect of the head that can be replaced by the modifier concept. This aspect will have a deep similarity or shallow similarity relation with the modifier concept. So the basic rule for this combination type will involve finding a subconcept of the head that possesses a deep similarity or similarity with the modifier concept, as in Figure 6.9.

insertion(A, B) :- *subconcept(B, C),* *deep_similarity(A, C, _).*	*insertion(A, B) :-* *subconcept(B, C),* *shallow_similarity(A, C, _).*

Figure 6.9 - Basic rule for insertion combination type in INCA

6.2.3.4 Context Insertion

In a context insertion combination the modifier concept refers to a larger context which is then inserted into the head concept. Here, we envisage that the context could be one of two types: (a) a subconcept of the modifier, or (b) a concept of which the modifier is a subconcept. Both these situations are accounted for in INCA as shown Figure 6.10. As with the insertion combination there are two forms of the rule, one for finding a shallow similarity and another for finding a deep similarity.

Context as a subconcept of A	Context as a super-concept of A
context_insertion(A, B) :- *subconcept(B, C),* *subconcept(A, D),* *deep_similarity(D, C, _).*	*context_insertion(A, B) :-* *subconcept(B, C),* *subconcept(D, A),* *deep_similarity(D, C, _).*
context_insertion(A, B) :- *subconcept(B, C),* *subconcept(A, D),* *shallow_similarity(D, C, _).*	*context_insertion(A, B) :-* *subconcept(B, C),* *subconcept(D, A),* *shallow_similarity(D, C, _).*

Figure 6.10 - Basic rule for a context insertion combination in INCA

6.2.3.5 Catachrestic

The implementation of the catachrestic combination checks to see if a deep similarity exists between the head concpet and the modifier concept. To locate this point of similarity, a subconcept of the modifier concept must be found which has a deep similarity with an aspect of the head concept. The basic rule for this is shown in Figure 6.11.[1] A subconcept of the concept A is any concept B that satisfies the following goal: relation(A, B, _, _).

```
catachrestic(A, B) :-
        subconcept(A, C),
        \+(subconcept(A, B)),
        shallow_similarity(C, B, _).
```

Figure 6.11 - Basic rule for catachrestic combinations in INCA

6.2.3.6 Contextual

In the contextual combination type the modifier concept appears to act as a context which is then compared with the head concept, and this comparison may be quite complex. Apart from a deep similarity between the concept and the head, there may also be deep similarities between some of their respective subconcepts. This combination type is very common in metaphors, especially nominal metaphors. Figure 6.12 outlines the base rule for this combination type in INCA.

```
contextual(A, B) :-
        context(A, C),
        deep_similarity(C, B, _).
```

Figure 6.12 - Basic rule for contextual combinations in INCA

[1] '\+' is the built-in predicate for 'not' in SICStus.

6.2.4 Returning an Interpretation

Each of the combination types that are listed above are implemented in two different logical procedures. One procedure merely returns true or false as to whether the corresponding combination type is satisfied. These procedures are the ones listed in Figures 6.7, 6.8, 6.9, 6.10, 6.11 and 6.12. The alternate procedures return an interpretation as an argument based on satisfaction of the procedure. Table 6.2 lists the information that is returned by the second type of combination procedure. For each combination only the most important information is returned.

Table 6.2 - Information given for an interpretation

Meronymous: return the relation that links the modifier concept to the head, this is usually the part relation.
Identity: return the shared category and at least one similarity between the subconcept of each.
Insertion: return the concept that the modifier replaces and the relation that exists between this concept and head concept.
Context Insertion: return the context and the subconcept of the head concept that the subconcept replaces.
Catachrestic: return the concept that the head concept is a metaphoric reference to.
Contextual: return the context that is referred to by the modifier concept and the similarities that exist between this context and the head concept.

Figure 6.13 displays two versions of the procedures or rules for the meronymous combination type, with the second procedure returning an output as a third argument. All the procedures that return interpretations are of this form. The output takes the form of a list with the type of combination listed as the first element. This first element is used later to aid in providing the user with an insight into interpretation.

Procedure 1	Procedure 2
meronymous(A, B) :-	*meronymous(A, B, [meronymous, Rel, B, A]) :-*
subconcept(A, C),	*subconcept(A, C),*
identical(B, C),	*identical(B, C),*
relation(A, B, _, _).	*relation(A, B, Rel, W).*

Figure 6.13 - Two types of procedure. The first only takes inputs. The second takes two inputs but also returns an output as the third argument

6.2.5 Algorithmic Design

The design of the INCA algorithm is a reflection of both the combination rules used and the specific properties of the knowledge base. The knowledge base has a relatively flat structure. Since the application of the rules works best on this type of structure, a more general structure would result in the need for more dynamic rules or ones that allowed for a greater depth of links between concepts.

The design of the basic algorithm at the heart of INCA is motivated by the desire to return coherent interpretations for a combination. Firstly, the conceptual inputs to the program should exist within the knowledge base, and this input is expected to be the names of two concepts. The first argument is the modifier concept and the second is the head concept. If these do not denote existing concepts then the program terminates. Otherwise the concepts are compared against every combination rule exhaustively. The rules used are of the type which return an interpretation, i.e. rule 2 in Figure 6.14.

```
INCA:
IF Inputs exist within the knowledge base
        Return interpretations
ELSE
        Return false
Return interpretations:
        WHILE(Meronymous Rule satisfied)
                Return an interpretation
        WHILE(ConjoinedRule satisfied)
                Return an interpretation
        WHILE(Insertion Rule satisfied)
                Return an interpretation
        WHILE(Context Insertion satisfied)
                Return an interpretation
        WHILE(Catachrestic Rule satisfied)
                Return an interpretation
        WHILE(Contextual Rule satisfied)
                Return an interpretation
```

Figure 6.14 - Pseudocode of INCA algorithm

The overall design of the INCA program is a rule based one, which involves taking a relatively simple approach to a complex problem. Concept combination becomes a process of taking a concept pairing and seeing if it satisfies any of our combination rules. In Figure 6.14, a combination is passed to each of the combination rules and if an interpretation is found then it is returned. Each rule is applied to the combination exhaustively, allowing INCA to return different interpretations for the same combination.

Our intuition is that some combinations will require more processing than others, and that when combinations are interpreted, the simplest interpretation is the best. We term this the simple-first rule. When people are interpreting a combination, the first generated interpretation is the best. As we imagine that the process interpretation starts from the simplest to the most complex, the first combination found will be simpler than other subsequent combinations found. In general use, INCA is run without the simple-first rule as we are interested in how many possible meanings a given combination can possess. The INCA algorithm which implements the simple-first rule is shown in Figure 6.15.

```
INCA:
IF Inputs exist within the knowledge base
        SWITCH

        CASE (Meronymous)
                Return Interpretation

        CASE (Conjoined)
                Return Interpretation

        CASE (Insertion)
                Return Interpretation

        CASE (Context Insertion)
                Return Interpretation

        CASE (Catachrestic)
                Return Interpretation

        CASE (Contextual)
                Return Interpretation
        DEFAULT
                Return false

ELSE
        Return false
```

Figure 6.15 - Pseudocode of the INCA algorithm (simple first)

In the last chapter, when discussing a number of combination types we noted that novel interpretations can be absorbed into the semantic network. Procedures exist for this within our current implementation (as will be seen in the sample interaction) but have not been included in the version of INCA presented here, i.e. both algorithms do not add new compounds to the semantic network. The reason for this is a desire to keep the semantic network relatively static, so that any new concepts or compounds are added under the guidance of the user. INCA does not automatically add new compounds to the network, although it could do so.

Rather, it is for users to decide what compounds should enter the network and manually insert these themselves.

6.3 INCA: A Sample Interaction

Initially in creating INCA each combination type was tested and developed with respect to one paradigmatic example. These types of examples, however, do not always give rise to polysemous interpretations. To counter this, concepts were added which had nothing to do with the original development of the combination type, e.g. concepts related to 'robin snake'. This combination was chosen as it was already clearly described as being polysemous in the literature (Wisniewski, 1997a).

A run-through of the INCA program will now be presented. In this run-through we will describe the initial set up of the program and then see how it treats the combinations 'robin snake' using the basic algorithm which returns all interpretations (see Section 6.1.5). Before we perform this run-through, some information on the knowledge base is given in Table 6.3. There are forty-eight concepts in the network (see Appendix C for a complete list). These range from quite complex representations to those that are quite simple. Also note that when creating representations for concepts, our model depends on these representations possessing a rich structure. This entails creating representations for most of the subconcepts as well.

Table 6.3 - Basic information on the knowledge base

Number of concepts: 48
Average number of relations per concept: 7.125
Range in number of relations in each concept over entire network: 1 - 17

Table 6.3 states that there were 48 concepts used in the run through. It should be noted that there are only 24 concepts which have more than six relations. So the information in the knowledge base is not extensive, but it should provide an adequate basis for running the model. When the systems is loaded, the knowledge base is set up. This knowledge base consists of a large body of relation/4 assertions and the shallow_similarity/3, deep_similarity/3 assertions computed (see Section 6.1.1).

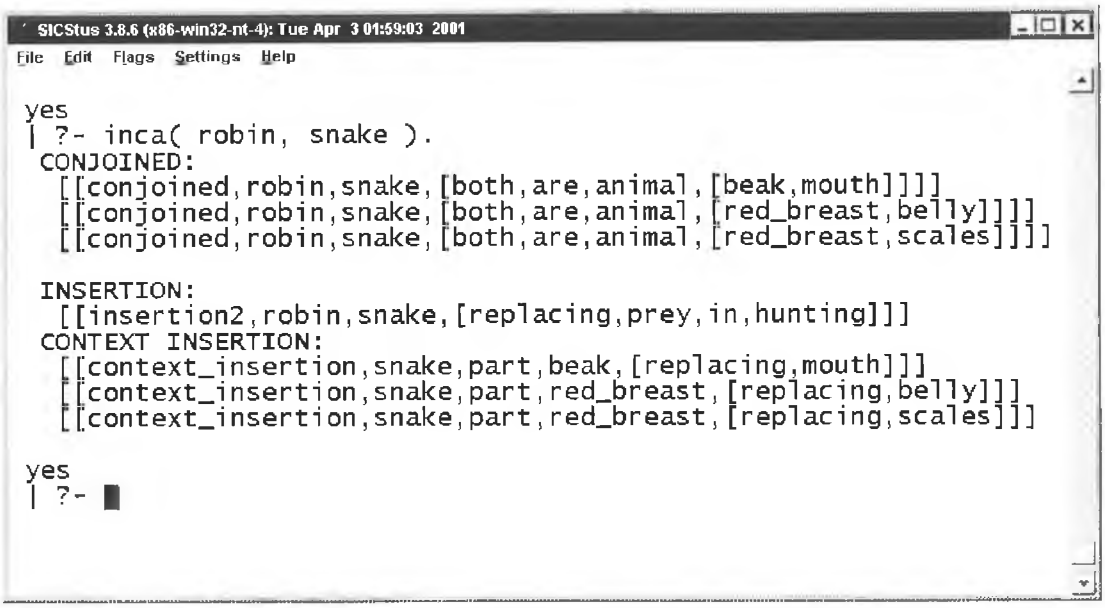


Figure 6.16 - Sample interaction with INCA

INCA is a PROLOG program and generally such programs involve an a goal-driven interaction between the user and the underlying theorem-prover. Typically the user will pose a goal and the system will attempt to resolve or satisfy this query goal. In Figure 6.16 the user asks for an interpretation for the compound 'robin snake'. This results in a number of interpretations which are listed in Table 6.4.

Table 6.4 - Interpretations for 'robin snake'

Context Insertion	
	'robin snake' is a snake with a beak (replacing mouth)
	'robin snake' is a snake with a red breast (replacing belly)
	'robin snake' is a snake with a red breast (replacing scales)
Insertion	
	'robin snake' is a snake that hunts robins
Identity	
	'robin snake' is both a robin and a snake, since both are animals and a beak is like a mouth.
	'robin snake' is both a robin and a snake, since both are animals and a red breast is like a belly.
	'robin snake' is both a robin and a snake, since both are animals and a red breast is like scales.

The variety of interactions that the user can invoke is quite broad. Consider the examples in Figure 6.17. In this figure the user queries a combination such as 'drug baron'. The response suggests that a 'drug baron' can be understood by using drug as a context for 'drug dealer' or user (as in a 'drug user'). But 'drug dealer' has a higher score and so is the more favourable interpretation. The user interacting with INCA can also find any concepts that may give rise to a catachrestic interpretation when a particular concept is used as modifier. Indeed this can be done with all combinations types, by supplying the head concept or modifier concept in question. In Figure 6.17, the query first sees if there is a concept which forms a catachrestic combination with the concept *ROBIN*. One is found, using the concept *MOUTH*. The catachrestic relationship between these two can be then examined further by finding which aspect of *ROBIN*, *MOUTH* is referencing.

```
SICStus 3.8.6 (x86-win32-nt-4): Tue Apr  3 01:59:03  2001
File  Edit  Flags  Settings  Help

yes
| ?- inca( drug, baron ).
 CONTEXTUAL:
   [[contextual,drug_dealer,baron,4]]
   [[contextual,user,baron,2]]

yes
| ?- concept_name( X ), catachrestic( robin, X ).

X = mouth ?

yes
| ?- concept_name( X ), catachrestic( robin, X, Z ).

X = mouth,
Z = [catachrestic,mouth,=,part,beak] ?
```

Figure 6.17 - A number of different queries put to INCA

In Section 6.2.5 the simple-first rule and algorithm were described. This returns the first combination found by the INCA system. Remember that the application of rules is done in a particular order, from the simplest to the most complex. This order is listed in Figure 6.13 and Figure 6.14 in the description of the algorithms. For the user to apply the simple-first rule they simply enter "simple_first" as a third argument to INCA. This can be seen in Figure 6.17.

```
SICStus 3.8.6 (x86-win32-nt-4): Tue Apr  3 01:59:03  2001
File  Edit  Flags  Settings  Help
yes
| ?- inca( robin, snake, simple_first ).

 SIMPLEST FIRST:
    [identity,robin,snake,[both,are,animal,[red_breast,belly]]]

yes
| ?-
```

Figure 6.18 - Returning the simplest interpretation first

In Section 5.3 we noted that several combination types give rise to new combinations which are stored in memory. In Figures 6.19 and 6.20 an interaction is shown where an interpretation is given for 'bottle mouth' and this compound is then added to the network. At present these compounds are not kept in the network (i.e. when INCA is closed this information is not saved). Initially, 'bottle mouth' is interpreted as a catachrestic combination where *MOUTH* refers to the opening part of *BOTTLE*. This interpretation is returned by the catachrestic/3 procedure. To check and see if a concept is part of the network the concept/1 procedure is used. As Figure 6.19 demonstrates, this procedure merely checks to see if a concept is part of a relation assertion in the network. To add this compound to the network another procedure, load_concept/2, is used.

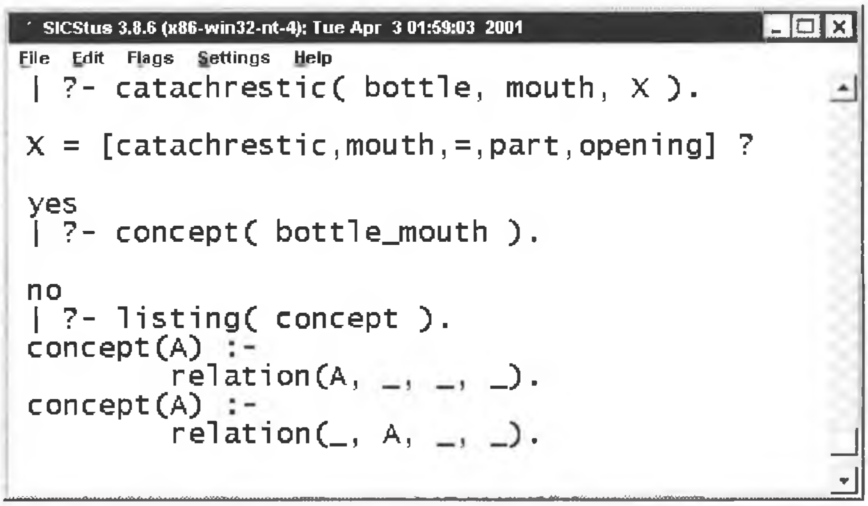


Figure 6.19 - Interpreting 'bottle mouth'

Figure 6.20 demonstrates how the load_concept/2 procedure is used. It takes two arguments and creates a group of relation facts for the compound. The compound becomes part of the network when the load_concept/2 procedure succeeds. Note in Figure 6.19 the concept/1 procedure now registers 'bottle mouth' as being part of the network. To create an entry for a catachrestic combination the properties of the concept it refers to are asserted for the new compound. So for example the compound 'bottle mouth' has the property *perform entrance*. Compounds are treated within in the INCA system as atoms linked together by an underscore, so the compound 'bottle neck' becomes bottle_neck. A further link is created in the concept *BOTTLE* where 'bottle neck' now exists as a part.

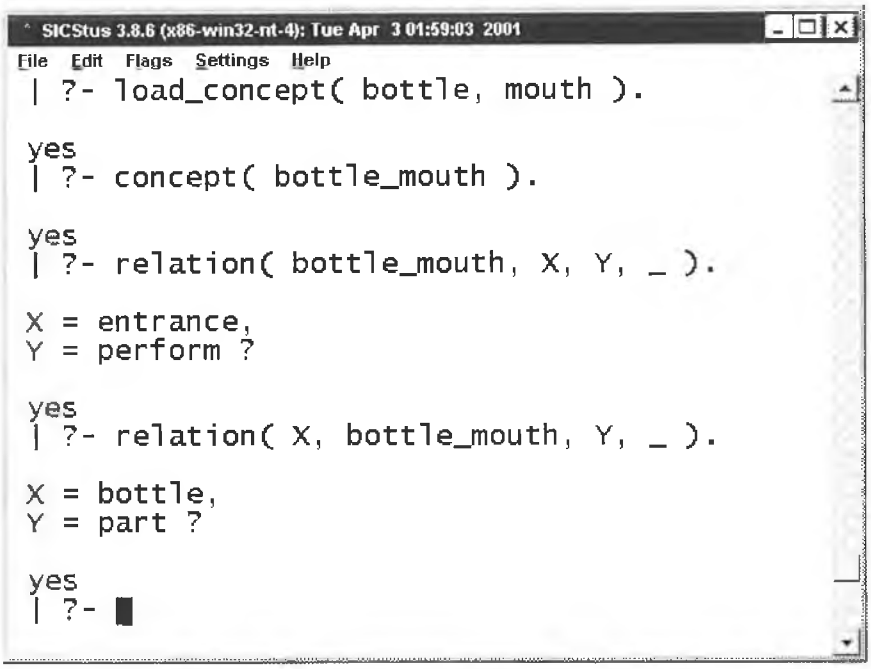


Figure 6.20 - Adding 'bottle neck' to the semantic network

6.4 How best to test INCA?

INCA is designed to be a model of concept combination and so, at the very least, it should produce interpretations that are similar to those that a human subject might give. If the interpretations that INCA gives are radically different to what people give, then this is a serious drawback to this current implementation of the structural alignment view. For example, suppose a study indicates that nearly all interpretations fall completely outside any of the combination types that have been listed. Clearly then, INCA could not explain these combination types. This would also have implications for the theory outlined in Chapter 5.

INCA should be tested on a large number of concepts, especially ones that have not been hand-coded by the author. Programs such as INCA can perform well when the coders hand-code the representations as invariably they know what information or type of information to add to aid their programs. Fortunately, it is actually possible to test INCA on concepts which have not been hand-coded by the author. Sapper (Veale, 1995) has an extensive knowledge base and concepts from it can be used to test the combination type rules. There are some possible drawbacks with this approach however. Although the Sapper knowledge base can be adapted and used, the representation of the concepts may not be the exact kind that INCA requires, as different researchers may represent the same concept in different ways.

In the next chapter a study is outlined where the responses of INCA are compared with those of 10 subjects over 20 combinations.[2] The representations that INCA will use for this study are taken from Sapper and have not been hand-coded by the author. The combinations used have also been chosen from a long list (over 2650) by three judges. A number of analyses are performed on the data from this study.

6.5 Chapter Summary

In this chapter an implementation of the structural alignment model was presented. This implementation is named INCA (interpreting nominal compounds using alignment). The knowledge base or semantic network around which INCA is built is was also described. In the INCA knowledge base concepts are represented as frames with slot:filler structures. The knowledge base also includes information on the a priori potential similarity between all the concepts, it precomputes the shallow and deep similarities that can exist between all the representations of concepts. The algorithm at the heart of INCA involves testing an input against rules which model each combination type. Each of these rules was briefly explained, some have different versions for generating interpretations based on shallow similarities or deep similarities. The algorithmic design of INCA was then set forth and a sample interaction with the program given. Two versions of the algorithm exist, one returns the simplest interpretation, and the other returns all possible interpretations. We also discussed possible ways of testing INCA, especially the idea of testing INCA using a different knowledge base. In the next chapter a study is presented which investigates INCA with respect to how it compares with human performance.

---

[2] Results are based on 19 combinations, one combination was eventually excluded from the study.

Chapter 7 - Evaluation

7.1 Introduction

This chapter tests the INCA program by comparing its interpretations of certain combinations against those of a number of human subjects. This study serves two purposes: the output of INCA can be analysed and the correctness of the combination types can be ascertained. If the output of INCA bears no resemblance to that of human subjects then the model is clearly flawed. Also, if during the study subjects generate interpretations which fall outside of the combination types suggested by the structural alignment view, then this view will have to be broadened to include new combination types.

A description of a study is given, which involved a number of subjects generating interpretations for twenty combinations. The responses of the subjects were classified to allow a comparison between INCA and human performance, where classification was based on the combination types outlined in the previous chapter. To this a further category was added, "not classifiable". Interpretations that fell outside of the combination types were assigned to this new category. The categorised responses of the subjects and INCA were then compared and analysed. Observations were also drawn from this experiment's raw data (the subject's interpretations). This chapter has two parts, the first outlines the study and its results. The second examines observations drawn from the study and is thus more qualitative in nature.

Part I: A Study

7.2 Description Of The Experiment

Having outlined INCA in the previous chapter we now wish to test the program's output. One way of doing this is to compare the responses of test subjects against the responses of INCA to a number of given combinations. This is the main principle behind the study that we now outline. The study essentially involved taking the knowledge base of Sapper (Veale, 1995) and creating a number of combinations that were given to subjects to interpret; the responses of these subjects was compared with those of INCA, which was run without the "simple- first rule" (see Section 6.1.5.1) so that as many combinations as possible would be found.

7.2.1 Organising The Experiment

There are 600 concepts in the Sapper knowledge base. Most of these concepts were combinations, e.g. 'eighteen twelve overture', and of the 600 concepts only 242 were not compounds. These non-compound concepts were selected to be the experimental building blocks of the combinations presented to the subjects. The range in complexity of the concepts was large, with the number of properties ranging from 2 to 32. Concepts with a small number of relations are not suitable to test INCA (as INCA will not have enough information to generate an interpretation), so all the concepts with more than six relations were selected, leading to a list of 59 concepts. From this list proper-nouns were removed, resulting in a final list of 50 concepts. The number of possible combinations that can be created from this list of 50 concepts is quite large (2500).

Using these concepts as a base, all possible juxtapositions of words into combinations were formed. A total of 2256 combinations were generated. The list of combinations was then repeatedly shuffled. Two judges initially picked the first fifteen combinations from this list that they both agreed were not nonsensical. However, in conducting the initial pilot study it became clear that in certain cases INCA would not return an interpretation for certain combinations. It was then decided that the first ten combinations from the list selected by the judges would be used, with an additional ten combinations selected by the author. These ten were combinations for which INCA would give at least one response.

7.2.2 Method

The subjects were given a form listing the compounds. They were then instructed to write as many interpretations for each combination as possible, where these interpretations were to be ones that the subjects felt were reasonable. Subjects were also asked to mark the interpretation that they felt best suited the combination:

> In this survey you will be presented with a number of novel compounds. For each compound you should write down as many meanings as possible. You may find that some compounds have only one reasonable meaning or you may find that there is no reasonable meaning for the compound.

A copy of the complete form is in Appendix D (Section 2).

7.3 Collation Of Data

The purpose of this study was to compare the performance of the INCA program against human performance. Initially, all interpretations were classified into seven categories, which are listed in Table 7.1 This categorisation was done so that a comparison would take place between the numbers of interpretation types that subjects produced, as opposed to the quality of the interpretations (which is obviously more subjective; the quality of interpretations is discussed in Part II of this chapter).

Table 7.1 - Classification of interpretations

1.	Meronymous
2.	Conjoined
3.	Insertion
4.	Context Insertion
5.	Catachrestic
6.	Contextual
7.	Not Classifiable

The categories listed in Table 7.1 list all the combination types plus one new category, "not classifiable", which is for interpretations that fall outside of the combination types proposed by the structural alignment view of concept combination. This category is quite important, as the more interpretations that fall into this category the less successful the alignment view appears.

Categorisation was done in accordance with the following criteria:

1. A meronymous interpretation suggested that the head concept was part of the modifier concept.
2. An conjoined interpretation suggested that the combination was in some way both a member of the head concept and the modifier concept.
3. An insertion interpretation suggested that the modifier concept could be inserted into the head concept.
4. A context insertion interpretation suggested that the modifier concept referred to another concept that was part of the head concept which was replaced.
5. A catachrestic interpretation used the head concept to make a figurative reference to a part of the modifier concept.
6. A contextual interpretation used the modifier concept to refer to a context that was then aligned with the head concept.
7. Any interpretation that does not meet the criteria in 1 through 6 above is categorised as "not classifiable".

There were twenty questions in the form (see Appendix D, Section 2). The responses of ten subjects and INCA to *nineteen* questions from the form were classified. Recall that the form originally consisted of ten questions selected by two judges and a further ten selected by the author. Due to a typographical error one of the questions from the ten selected by the author was not included in the analysis. These two groups of questions will be analysed separately.

For each question the number of interpretations for each combination was calculated and the response of each subject after classification was placed in a table. From this the average for the 10 subjects was determined. The responses to the same combinations from INCA were also entered into this table, and the average score of the ten subjects was compared with the results for INCA. An example of this data is given in Table 7.2, while the complete data can be found in Appendix D (Section 3). Table 7.2 shows the data for question six on the form. From this it can be seen that subject 7 (S7) only responded with one interpretation and that this interpretation was classified as a context-insertion type. At the bottom in italics is the data that was compared, i.e. INCA's responses against the average for the ten subjects.

Table 7.2 - Example of collected data

Q6	Meronymous	Conjoined	Insertion	Context Insertion	Cataohrestic	Contextual	N.C
S1	1	0	0	0	0	0	0
S2	0	0	2	0	0	0	1
S3	0	0	2	0	0	0	0
S4	0	0	1	0	0	0	0
S5	0	0	1	0	0	0	0
S6	0	0	0	0	0	0	3
S7	0	0	0	1	0	0	0
S8	0	0	2	0	0	0	0
S9	1	0	0	0	0	0	0
S10	0	0	2	0	0	0	0
INCA	*0*	*2*	*1*	*2*	*0*	*0*	*0*
S/Avg	*0.2*	*0*	*1*	*0.1*	*0*	*0*	*0.4*

7.4 Results

As already mentioned, in conducting the initial pilot study it became clear that INCA would not return an interpretation for certain combinations. It was therefore decided that the first ten combinations from the list selected by the judges be used, with an additional ten combinations selected by the author. These ten were combinations for which INCA gave at least one response. So the examination of the results will take place in two steps. First the combinations picked by the judges, and then the combinations selected by the author, will be examined in sequence.

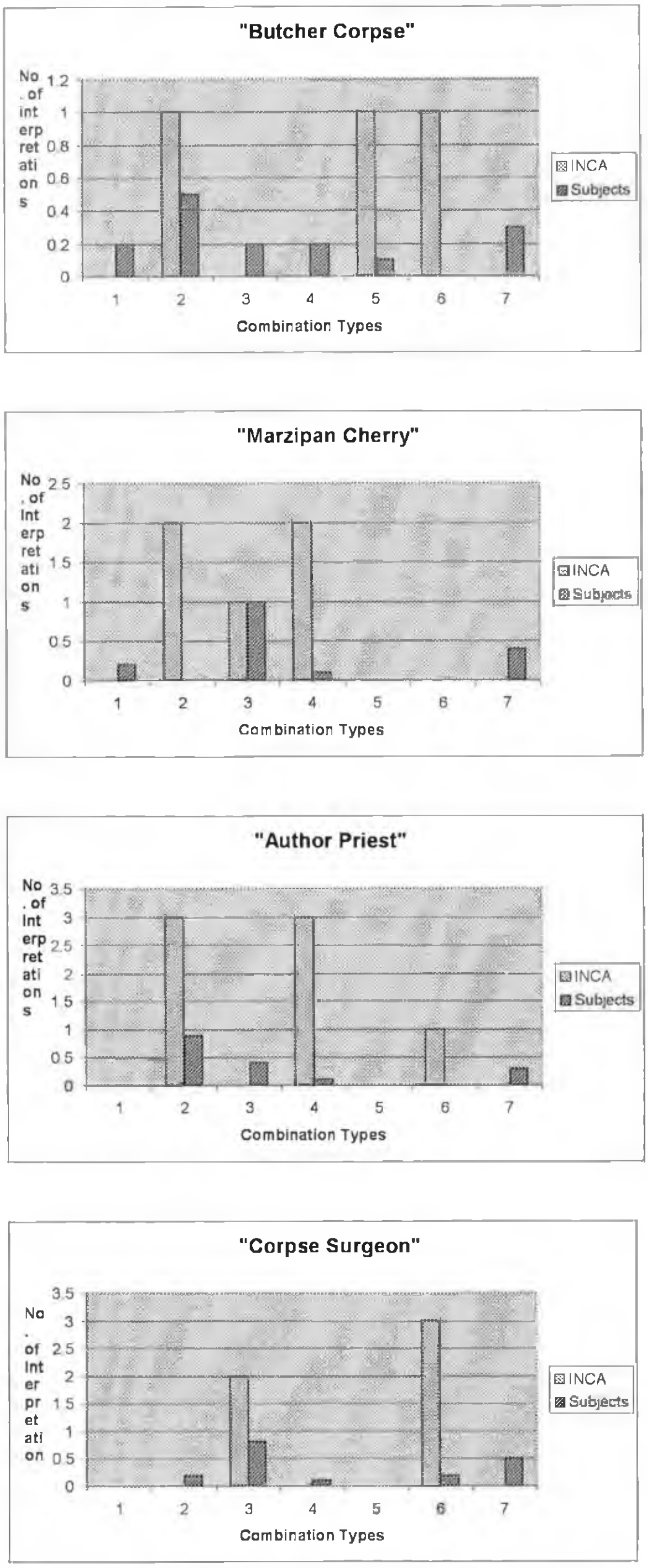
7.4.1 Results: The First Ten Combinations

To summarise, the combinations we first examine are the ones picked by the two judges. INCA failed to respond with interpretations for a number of these combinations. Of the ten combinations INCA could only generate four interpretations, which are listed in Table 7.3. This is a possible result of at least two factors: problems with the INCA system itself, or the nature of the representations used. The INCA program might be flawed so that few interpretations are possible given any combination, or the nature of the representations may cause information that INCA needs to find an interpretation to be omitted.

Table 7.3 - Combinations picked by the judges

Combination	Interpretation found?
1. 'Eye plot'	No
2. 'Raisin Sculptor'	No
3. 'Butcher Corpse'	Yes
4. 'Cake Machete'	No
5. 'Corpse Surgeon'	Yes
6. 'Marzipan Cherry'	Yes
7. 'Man Criminal'	No
8. 'Author Priest'	Yes
9. 'Surgery Priest'	No
10. 'Experiment Eye'	No

As only four combinations resulted in an interpretation being given by INCA we will compare the interpretations of INCA against human subjects for these combinations. These combinations are: 'butcher corpse', 'corpse surgeon', 'marzipan cherry ', 'author priest'. This makes a comparison against all ten combinations rather redundant as the lack of values for INCA in six of the ten combinations will skew any average score for INCA. So the four combinations where INCA does make a response will be examined. The lack of responses to the first ten combinations in the study is of critical concern.



Combination Types: 1. Meronymous, 2. Conjoined, 3. Insertion, 4. Insertion, 5. Catachrestic, 6. Contextual

Figure 7.1 - Comparison of INCA and subjects on four combinations

Figure 7.1 outlines a comparison of INCA against the average of 10 human subjects. It is clear that INCA finds more interpretations than human subjects, the most likely reason for this is because INCA attempts to look for all possible interpretations. The program looks for all possible interpretations available within the confines of the combination type rules. When an interpretation is found for one combination type INCA also seems to generate more than one interpretation of this type. For example, 'author priest' is interpreted as a context insertion combination three times. Examining INCA's responses, the following context insertion interpretation were made:

1. >> a priest who affects readers, reader replacing believer <<
2. >> a priest who effects readers, reader replacing heretics <<
3. >> priest who depends on novels, novels replacing bible <<

All three interpretations are based on a perceived similarity related to the modifier concept and an aspect of the head concept. It appears that a *READER* is both like a *BELIEVER* and a *HERETIC* and that a *BIBLE* is like a *NOVEL*.

From studying Figure 7.1, we can also predict how INCA's simple first rule would effect the comparisons. In each bar chart the combination nearest the left edge (or Y axis) would be chosen. Now, for each of the combinations we can argue that the combination type with the highest value as scored by the subjects is the best interpretation. In that case, INCA would find the best interpretation in three out of the four cases. Where it would fail is on 'mazipan cherry' which is interpreted by INCA as >> something that is both a marzipan and a cherry since both are a type of food and almond mush is like cherry pulp <<.

7.4.1.1 Missing Information

The danger in taking representations that the authors had not constructed is that information that one might typically include or expect to be present may actually be left out. This is the case with a number of combinations drawn from Sapper (Veale, 1995). For example, consider the combination 'surgery priest', which could be interpreted as: >> a priest who councils people undergoing surgery <<. This would be a context-insertion combination type where the *PATIENT* of *SURGERY* replaces the *CHURCHGOER* of *PRIEST*. Both *PATIENT* and *CHURCHGOER* may be similar as both *are people* and both *seek relief*; albeit they may have different types of relief. This information is not contained in the representations that were used to test INCA. An approach which looks for interpretations based on structural correspondences needs information to be present to be matched. The lack of interpretations can be interpreted as a reflection of the lack of certain information in some of the concepts used for this study.

7.4.1.2 Summary

The first part of the study did not provide a lot of information with respect to how INCA compares with human subjects. The reason for this failure appears due to what we have dubbed "missing information", i.e. information that if placed in the knowledge base would allow INCA to generate interpretations. Four combinations alone are not sufficient to provide a suitable basis for comparison (hence the combinations used in the next section). From the combinations that did result in an interpretation, there do, however, appear to be a number of interesting points to glean.

Firstly, INCA produces more combinations than human subjects on average do, even when these subjects are asked to provide as many interpretations as possible. Also, when an interpretation is given which falls into one combination type, subjects to not generally generate another interpretation which will also fall into the same combination type. INCA, in contrast, consistently does this. More importantly it also shows a major flaw in these types of programs. INCA and programs of its ilk rely crucially on good input (in this case very rich representations). When the representations of concepts in the knowledge base are not rich the program falters easily. Consequently, of the ten combinations given to INCA only four could be interpreted.

7.4.2 Results: The Remaining Combinations

INCA's lack of success in generating any response in the earlier pilot study has led the author to add ten combinations to the experimental materials so INCA might at least provide a response and some comparison could be made between the system and the human subjects. One of these ten combinations one was excluded due to a typographical error leading to the following nine combinations being used:

> 'accountant author', 'architect composer', 'surgery composer', 'criminal composer', 'cake marzipan', 'artery blood', 'accountant office', 'cancer disinfectant', 'author scalpel',

For each of these combinations the interpretations of the subjects were again classified into one of seven categories. The average for the ten subjects was calculated and this was then compared against INCA's responses. Figure 7.2 provides a macro view of this analysis. From this figure it can be seen that again INCA provides more interpretations than the average subject, which fits well with what was found in with the first group of combinations. In terms of the first three combination types (meronymous, conjoined and insertion) INCA does provide more interpretations but the difference is not as noticeable when compared to the remaining three combination types. For instance, INCA produces a large quantity of context-insertions when subjects clearly produce very little. This warrants further examination. Also worthy of investigation are the large number of contextual combinations found by INCA compared to the small number found by the subjects.

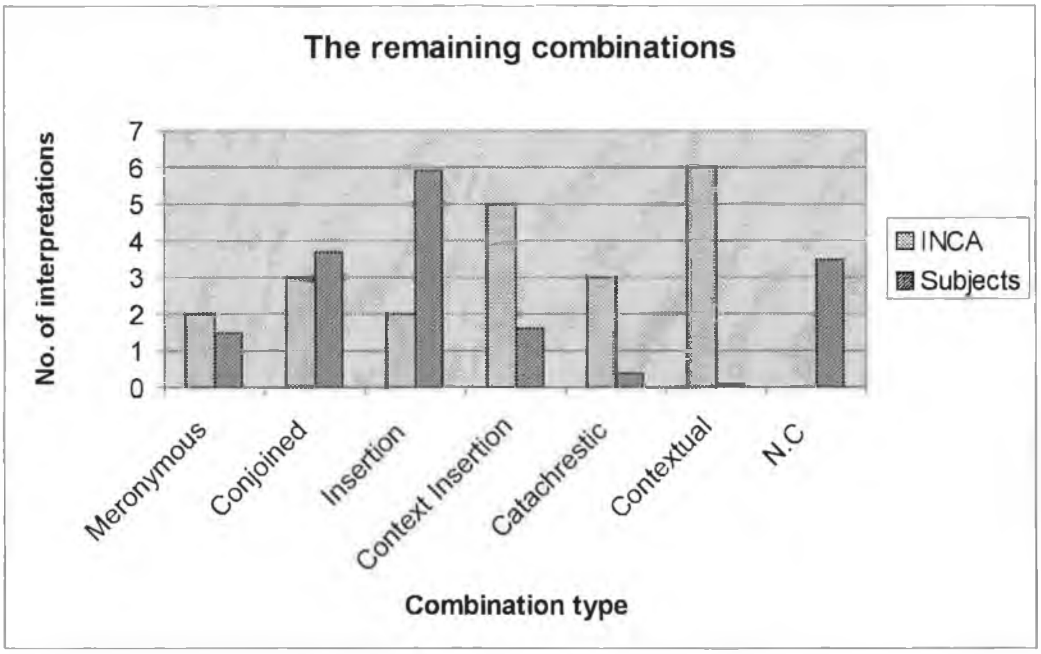


Figure 7.2 - Comparison of INCA and subjects over the second group of combinations

The comparison between INCA and the subjects can also be examined question by question. Figures 7.3 to 7.12 outline the results in such a way. An examination at this level highlights areas where INCA has been successful while also pointing out possible problem areas. In each of these figures we will suggest that the best type of interpretation is the one that occurs with the most frequency among the subjects, e.g. in Figure 7.3 a conjoined combination appears to be the best type.

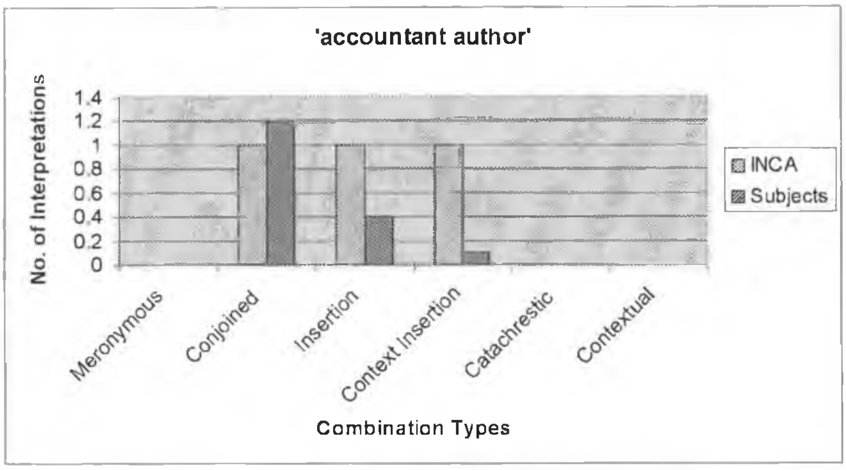


Figure 7.3 - Comparison for 'accountant author'

As already noted, the conjoined combination type seems to be the best interpretation type for 'accountant author'. Applying the simple-first rule, INCA would return this result also. Note however that INCA, unlike human subjects, generates multiple interpretations for the same combination type. But this can be remedied by picking only one interpretation for each combination type. Note also that INCA generated multiple insertion based interpretations. This was because the concept *ACCOUNTANT* was similar in some key aspects to *AUTHOR*. For example, *ACCOUNTANT* was found to be similar to the concept 'Norman Mailer', since both *use ink*, both *use a pen* and both *are people*.

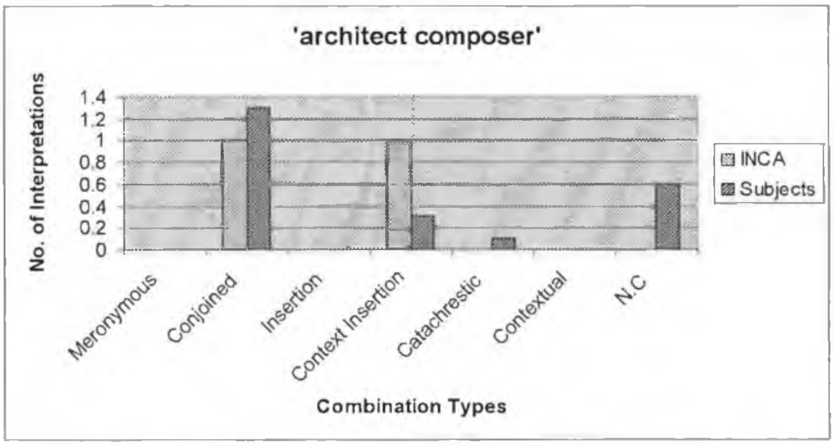


Figure 7.4 - Comparison for 'architect composer'

In Figure 7.4 the conjoined combination seems to be the best interpretation. Again, if INCA only found the simplest interpretation then it would also find this interpretation to be the best. However, INCA appears yet again to have found a number of context insertion combinations, which although also found by some subjects, not many were found overall. INCA found a number of similarities between concepts related to *ARCHITECT* and concepts related to

COMPOSER. For example, a *CONSTRUCTION-CREW* was seen to be similar to an *ORCHESTRA* and a composer's *LIBRETTO* was seen to be similar to an architect's *BLUEPRINT*.[1] These are good examples of deep similarity and these became the contexts that were used to generate the context-insertion combinations. For example an 'architect composer' is >> a composer who creates blue prints (replacing librettos) <<.

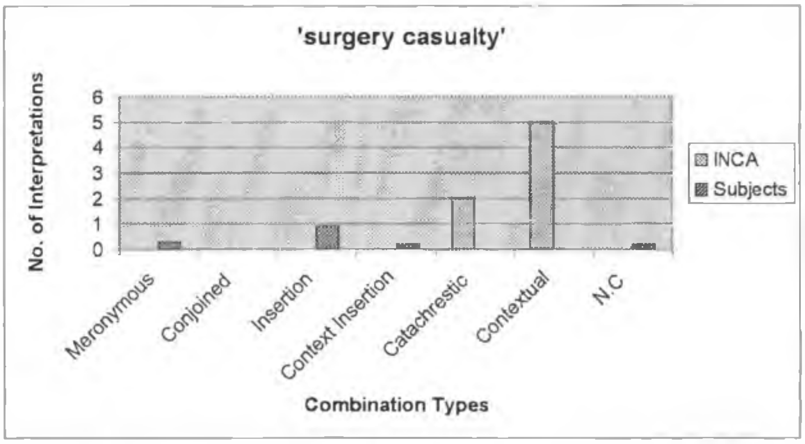


Figure 7.5 - Comparison for 'surgery casualty'

From Figure 7.5 we can see that INCA falters quite badly with this combination. Although the insertion type seems to be the best interpretation type INCA could not find any insertion interpretations. On closer examination this appears to be another case of missing information. Simply put, there was not enough information in the knowledge base for certain key similarities to be found. For example, a representation of *CASUALTY* should perhaps include information on how a person becomes a *CASUALTY*, i.e. due to an accident. If *SURGERY* and *ACCIDENT* are similar then *SURGERY* could be inserted, but this information was not in the knowledge base.

[1] Libretto = the text of a work (as an opera) for the musical theater.

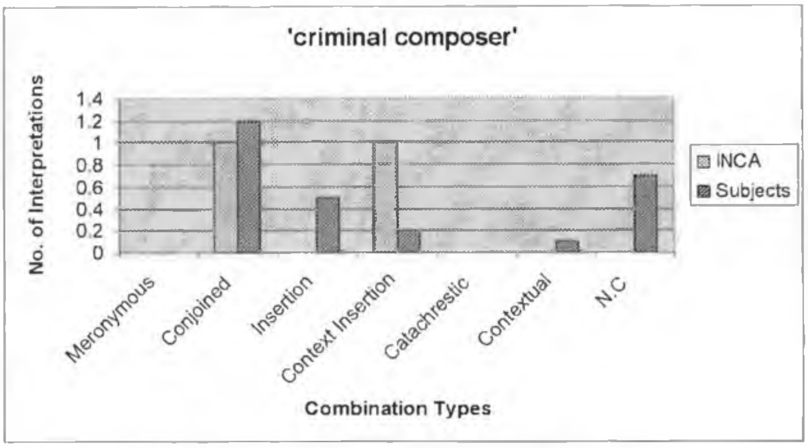


Figure 7.6 - Comparison for 'criminal composer'

In Figure 7.6 is another example of where the simple-first rule finds the best interpretation. Again however, INCA seems to find context-insertion combinations when subjects do not. In this particular case, this was because the representation of the concept *CRIMINAL* is related to the concepts *LOCK-PICK* and 'thermal lance'.[2] Both of these concepts are found by INCA to be similar to the *BATON* that a *CONDUCTOR* uses, since all are instruments and each has a similar shape, i.e. *are long* and *narrow*. These similarities formed the basis for the context insertion combinations INCA found.

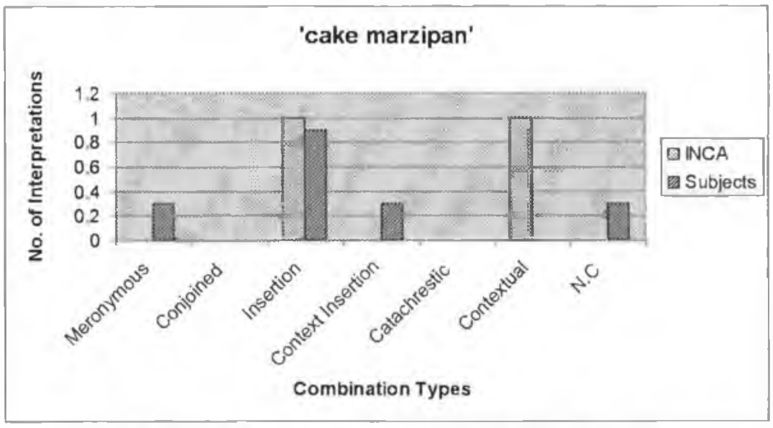


Figure 7.7 - Comparison of 'cake marzipan'

With regards to Figure 7.7, again when the simple-first rule is applied, then INCA finds the best interpretation type, which for this combination is the insertion type. This is also an interpretation that gives rise to a number of reversals, discussed later in Section 7.5.1. This combination also gives rise to a number of contextual interpretations which subjects clearly did not find, showing a bias in INCA towards the generation of these combination types.

[2] thermal lance = A device used to burn through metal, e.g. in safe-cracking.

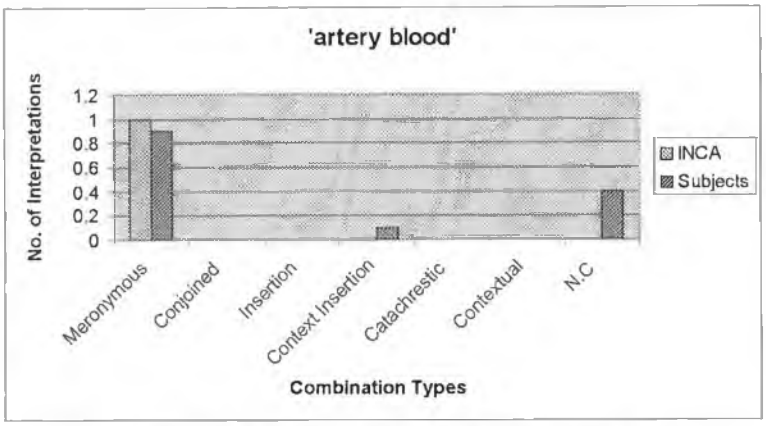


Figure 7.8 - Comparison of 'artery blood'

Figure 7.8 shows an unusual combination with respect to this study, as this combination does not give rise to a great number of distinct interpretations. It appears that there is only truly one acceptable interpretation for this compound. Again, when the simple-first rule is applied then INCA finds the best interpretation.

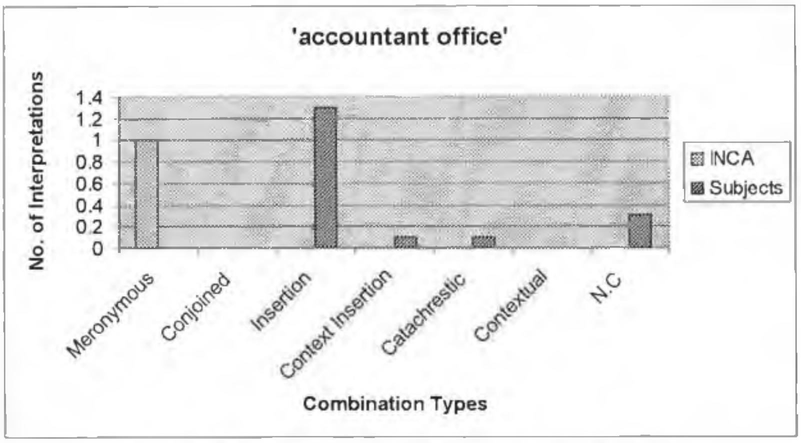


Figure 7.9 - Comparison of 'accountant office'

The combination in Figure 7.9 appears to have caused great difficulty for INCA as the best interpretation for 'accountant office' is of the insertion type. A closer examination reveals that the concept *OFFICE* is related to the concept *ACCOUNTANT* in the knowledge base and this becomes a point of insertion. This is not surprising as *ACCOUNTANTS* typically work in offices and this information is in the representation of the concept *ACCOUNTANT*. Furthermore, the concept *OFFICE* does not have a representation of *WORKER* or *OFFICE-WORKER* that *ACCOUNTANT* can replace, giving rise to an insertion type. Again INCA's failure results from missing information in the knowledge base.

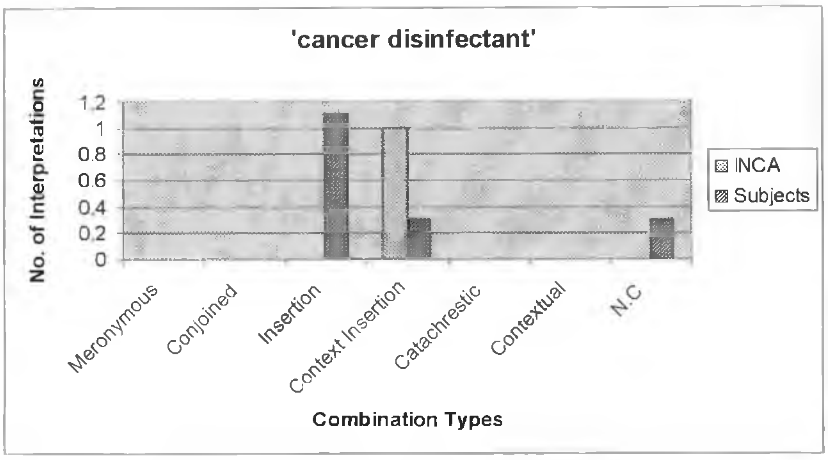


Figure 7.10 - Comparison of 'cancer disinfectant'

From Figure 7.10 it can be seen that for the subjects, 'cancer disinfectant' is clearly an insertion combination. INCA, however, finds a context insertion combination. In fact, INCA suggests that the *CANCER* acts a context for the concept *CANCER-CELL*, so that *CANCER-CELL* replaces an associate of *DISINFECTANT*, namely *BACTERIUM*. So the combination was interpreted as >> a disinfectant which removes cancer-cells <<. For most subjects, 'cancer disinfectant' was a >> a disinfectant which removes cancer <<. Here it seems that INCA is being more specific than the subjects.

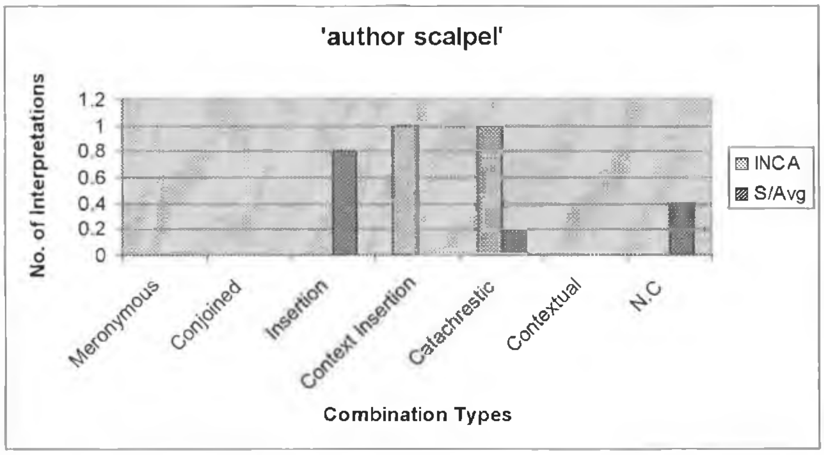


Figure 7.11 - Comparison of 'author scalpel'

In Figure 7.10, INCA again does not generate an insertion combination but generates a context insertion and a catachrestic interpretation. Even when the simple-first rule is applied, here INCA does not find the best interpretation. However, the interpretations that INCA does generate are worth considering. The catachrestic interpretation marks out that INCA has found the head concept to be similar to a part of the modifier concept. In this example it suggests that a *SCALPEL* is like an author's *PEN*. This results in an interpretation such as >> an 'author scalpel' is the pen that an author uses <<. The same similarity between *PEN* and *SCALPEL* is also used to form the context insertion interpretation.

7.4.2.1 Summary

This section reveals that the second group of combination types produce similar findings to those of the initial ten combinations. Firstly, INCA produces more interpretations than subjects do. Secondly, INCA produces more than one interpretation for a particular combination type. So INCA is actually consistent across both groups of combinations.

Picking the simplest interpretation first, or applying the simple-first rule (as we have dubbed it), seems to generally correspond with the best interpretation given by human subjects. When this heuristic fails, it often appears to be as a result of deficiencies in the knowledge base, i.e. missing information. It is these deficiencies that make the generation of an interpretation for the first set of combinations so difficult.

Having said this, INCA is sometimes led astray by similarities between certain concepts. For example, in the case of 'criminal composer' where interpretations are returned on the basis of how a *BATON* (drawn in from the context *CONDUCTOR*) is like a *LOCK-PICK*, it appears that subjects either do not either find this information or if it is found it is not used during the combination process. This is especially true when other interpretations can be found.

7.4.2.2 Simple First Rule Algorithm

Figure 7.12 below shows a comparison of INCA for the second group of combinations when only one interpretation, the simplest, is allowed for each combination type. This provides a closer fit when compared to Figure 7.2. However, it is clear that INCA still generates interpretations that subjects do not, especially contextual interpretations. This could of course be an artefact of the small number of subjects but may also reflect a larger problem, namely a bias towards the generation of context insertion combinations.

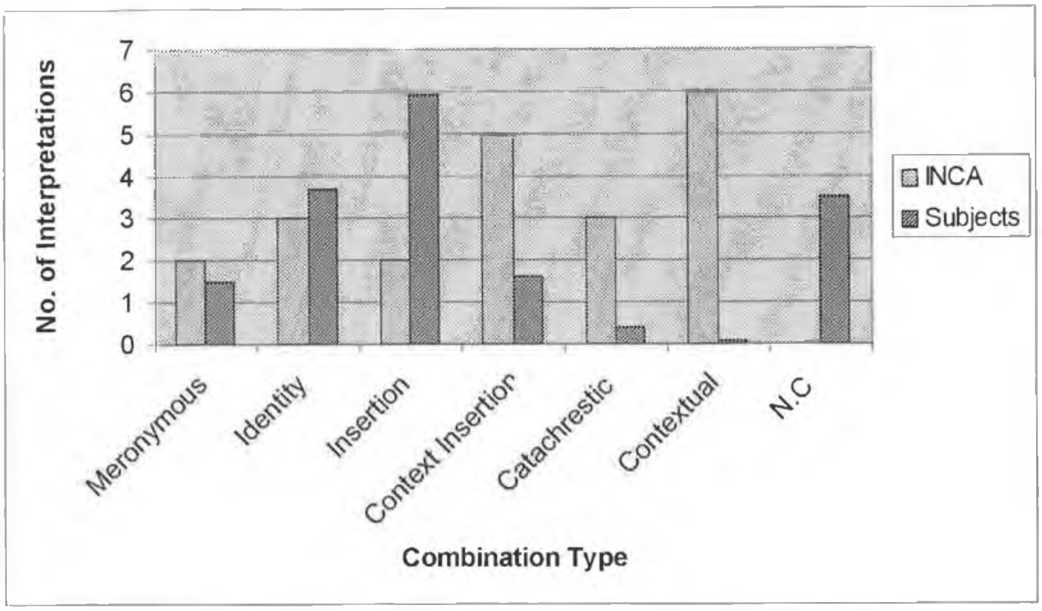


Figure 7.12 - Comparison of INCA and subjects over the second group of combinations where INCA returns only one interpretation per combination type

Part II: Observations On The Raw Data

7.5 Observations

The observations outlined in this section are drawn from the raw data generated by the subjects in the experiment. These observations deal mainly with the correctness of our approach and with results which were unexpected in light of previous research on concept combination. The observations are based largely on the actual interpretations of the subjects but we will refer back to the comparisons made in the last section.

7.5.1 Dealing With "Not classifiable" Interpretations

The interpretations given by the subjects in the study were classified into different combination types. However, a number of interpretations fell outside of the standard combination types. Especially when the interpretations were nonsensical or strange. For example, 'architect composer' was interpreted as >> just an architect <<. This type of interpretation does not agree with any of the theories on concept combination. Within these "not classifiable" interpretations, however, there may reside new combination types. For at least two subjects some interpretations gave rise to what we term a reverse-insertion. For example, Subject 6 and Subject 2 both interpreted:

'cake machete' as >> a machete-shaped cake <<

This is what is known as a reversal, i.e. it has been interpreted as 'machete cake'. The head is now the first element and modifier is the second element, which is not the default way that combinations work in English work. Although only two subjects gave rise to this type of interpretation, they found this type of interpretation more than once. The interesting aspect of this new combination type is that is appears to be a variant of the insertion combination type but where the head concept is the concept that is inserted, with the modifier concept being the recipient of the insertion.

A further problem with the "not classified" group relates to what the words denote. Sometimes interpretations relied on concepts that were not clearly related to the combination. For example, 'Corpse Surgeon' was interpreted as >> a funeral home person, making the dead look proper for burial << by subject 7. This was not the subject's choice as a best interpretation for this combination but it shows that combinations can refer to concepts not directly related to the constituents. Even in terms of the structural alignment view, where contexts can be evoked, this particular interpretation is quite complex and falls outside our view. Another example was given by subject 2, who interpreted a 'butcher corpse' as >> a killer corpse, e.g. a zombie <<. This type of interpretation also falls outside the structural alignment view and in fact all other theories of concept combination. A question arises as to whether such an interpretation is worth investigating as only one of the ten subjects generated such an interpretation.

Some interpretations were not classified as it appeared that the interpretation of the combination was treated as a 'adjective noun' and not 'noun noun'. For example, 'criminal composer' was sometimes interpreted as >> a bad composer <<. Here criminal appears to be used as a synonym of the word bad (this occurs in English in utterances such as "his piano-playing is criminal"). When an interpretation appeared to be based on an adjectival modifier the interpretation was placed in the "not classifiable" category. Homonymy is not treated within the confines of concept combination research but clearly combinations can involve homonymous words.

Overall, we suggest that the combination types outlined by the structural alignment view appear to cover a good range of interpretations but they do not have complete coverage. For at least two subjects, combinations can be interpreted as a reverse insertion. Another type that might be marked out is the 'killer corpse' example (when interpreted as >> a zombie <<) is that when an interpretation is similar to an existing concept, then the interpretation is based on this existing concept.

7.5.2 The Workhorse: Insertion Combination Type

The range of different interpretations that can be given by the insertion combination type can be large. Consider the combination, 'criminal composer', and the two interpretations listed below drawn from the study.

> Subject 2: >> a composer that writes music / poems about criminals <<
> Subject 7: >> someone who composes for criminals <<

In the first, the subject matter of the *MUSIC* is replaced with *CRIMINAL*. In the second, the *AUDIENCE* of *MUSIC* is replaced with *CRIMINAL*. These types of combinations are thus quite likely to be highly polysemous. The structural alignment view offers a good explanation of why this is the case. If the modifier is similar in multiple respects to the head, then an interpretation can be generated for each of these similarities. For example, the concept *CRIMINAL* may match parts of the *COMPOSER* that link to the members of the *PERSON* concept. Consider that a *COMPOSER* often has a *SPONSOR*, so that a 'criminal composer' could be >> a composer who is supported by criminals << where criminal replaces sponsor. The more matches the modifier has with parts of the head, the more interpretations that can be generated.

7.5.3 Dealing With The Conjoined Combination Type

Although a reading of the literature on concept combination suggests that when presented with a combination, such as a 'robin snake' it will be interpreted as >> a snake which is both a robin and a snake <<, it actually appears that when a conjoined combination arises, subjects can pick one element of the combination to be the head. For example: 'accountant author' was interpreted as:

> >> an accountant who is also an author <<
> >> an author who is also an accountant <<

It may be that the treatment of conjoined combinations needs to be revised. Wisniewski (1997a) suggests that these types of combination should be treated either as conjunctions or hybridizations but neither of these appears to capture a situation where a subject picks one element to be the head and the SA view does not fully account for this particular type of interpretation.

7.5.4. The Quality of INCA's Interpretations

INCA's interpretations were on the whole not as concise and direct as those of the human subjects. However, the output of INCA was generally in agreement with human subjects' interpretations. Consider Table 7.4 which compares some interpretations generated by INCA with some interpretations given by the subjects. 'Butcher corpse' was interpreted as a conjoined combination and most subjects interpreted it in a similar manner. However, insertion interpretations in INCA mention the aspect of the head concept that is replaced. This is not the case with human subjects. This information is given by the system so that a user can see where the basis for the interpretation originates. INCA also provides two interpretations for 'corpse surgeon' that are highly similar. Human subjects (in this study at least) did not give interpretations that are highly similar.

Table 7.4 - Comparisons between INCA and human subjects

Combination	INCA	Subject
'Butcher Corpse'	"A person who is both a butcher and a corpse"	"the corpse of a butcher"
'Corpse Surgeon'	"A surgeon who effects corpses, replacing patient" or "A surgeon who performs surgery on corpses, replacing patients"	"A surgeon who works on corpses"
'Cancer disinfectant'	"A disinfectant that effects cancer cells, bacterium is replaced by cancer cell"	"product that disinfects cancer"

These interpretations are at best adequate, but INCA will often produce interpretations that a human subject will not. This can be seen more clearly in terms of the actual interpretations themselves. Consider the interpretations that INCA generated for certain combinations when human subjects produced nothing equivalent, in Table 7.5 below:

Table 7.5 - Interpretations that INCA gives but subjects do not give

'Butcher Corpse'	Catachrestic: "Corpse refers to the carcass that a butcher effects"
'Marzipan Cherry'	Contextual Insertion: "A cherry made from almond mush" [The almond mush part of Marzipan replaces the cherry pulp part of cherry] Contextual Insertion: "A cherry made from sugar" [The sugar part of Marzipan replaces the cherry pulp part of cherry]
'Author Priest'	Contextual: "A gothic author priest" [The concept 'gothic author' was selected as a context for priest] Contextual: "A romantic author priest" [The concept 'romantic author' was selected as a context for priest]

In Table 7.5, three interpretations have been selected. These are ones which INCA generated but for which none of the human subjects generated a similar interpretation. The first interprets a *CORPSE* as referring to the *CARCASS* that a *BUTHCER works on*. This arises since a *CARCASS* and a *CORPSE* are actually quite similar concepts. This could be a case where INCA has found a similarity that most subjects may not notice. The second combination 'marzipan cherry' gives to two types of contextual insertion combination. This is found because both the almond-mush and sugar part of *MARZIPAN* is similar to the *CHERRY-PULP* of *CHERRY*. The generation by INCA of contextual interpretations begins to become cognitively implausible when we reach the combination 'author priest'. It seems unlikely that subjects would interpret an 'author priest' as >> a romantic author priest << because a *BIBLE* is like a *ROMANTIC NOVLEL*.

It could be said that similarities are on one level merely a reflection of the knowledge base, i.e. the problem lies within the contents of the knowledge base. However, we suggest that the combination rules as outlined in Chapter 6 may need some reworking. For example, what may count as a suitable context for the contextual combination type should be tightened. Perhaps a threshold for similarity should be added, where only contexts which are highly similar to the head concept can be brought into play.

7.5.5 Nouns and Senses

A problem that is not dealt with in the literature on concept combination but which clearly affects this study is the polysemy of the nouns in the noun-noun compounds. Many nouns have several senses, while an interpretation may be based on just one sense. For example, an entry for the word surgery in WordNet (Fellbaum, 1998) contains the information listed in Figure 7.13. The representations used by INCA for this study were based around just one sense and did not include the extra senses that subjects use when making interpretations. (This was noted briefly in Section 1.4.2).

The **noun** "surgery" has 4 senses in WordNet.

1. **surgery** -- (the branch of medical science that treats disease or injury by operative procedures; "he is professor of surgery at the Harvard Medical School").
2. **surgery** -- ((British) a room where a doctor or dentist can be consulted; "he read the warning in the doctor's surgery").
3. operating room, OR, operating theater, operating theatre, **surgery** -- (a room in a hospital equipped for the performance of surgical operations; "great care is taken to keep the operating rooms aseptic").
4. operation, **surgery**, surgical operation, surgical procedure, surgical process -- (a medical procedure involving an incision with instruments; performed to repair damage or arrest disease in a living body; "they will schedule the operation as soon as an operating room is available"; "he died while undergoing surgery")

Figure 7.13 - WordNet entry for surgery

In fact, the polysemy of nouns is a problem for all models of concept combination. The structural alignment view does not favour one sense over another and will attempt to find interpretations for any sense that is located in our knowledge base. However, it should be noted that the initial construction of our knowledge base did not include concepts which have multiple senses. The burden is on the architect of the knowledge base to include information on word forms which can give rise to multiple senses.

7.5.6 Summary

This section dealt with a number of observations drawn from the study described earlier in Section 7.1 There are five broad observations that could be drawn from this study:

1. there may be two new combination types (reverse insertion and referential)
2. the insertion combination gives rise to a large range of interpretations
3. a subject's dealing with conjoined combinations was more subtle than had been anticipated or suggested by previous research on concept combination.
4. INCA's interpretations vary in some respects with those of subjects for the same combination type.
5. INCA like other models of concept combination does not deal with the possibility that nouns will have multiple senses.

7.6 How Does INCA Really Compare With Human Performance?

Having tested INCA, looked at an adjusted version of INCA and examined some observations of the raw data, we can now pose the question: how does INCA really compare with human subjects? The answer is not entirely positive. In terms of interpretations, INCA produces more than most subjects do. This can be altered by only allowing one possible interpretation for each combination type. The quality of the interpretations INCA gives, although basic, are not completely different to those that a subject would give. The combination types that the SA model predicts do, however, appear to offer a good coverage of the interpretations that human subjects give.

However, there should be a limit placed on the contextual and context insertion combination rules. It is clear that INCA finds these interpretations when subjects do not. This is a major difference between INCA and humans. Fortunately, there is a way to reduce the number of interpretations found by these combination rules. A threshold can be placed on the similarities that are needed for these combinations. Only interpretations which involve concepts that have a very high degree of similarity should be allowed.

Of course, a simple program will never be identical to a human especially in relation to a problem that requires a great deal of information. Even giving INCA access to over 600 concepts from Sapper did not ensure that interpretations would be generated. In many cases the information needed was missing. For example, in a number of cases insertion interpretations would have been possible if more information had been in the knowledge base. The INCA system, despite its bias to generating certain interpretations that human subjects did not, worked quite well. The interpretations that INCA produced, although more crude than human subjects, were not altogether different from those that the subjects prodcuced. Earlier in Section 7.4.1 we stated that INCA's lack of success in the first part of the study was of major concern. It appears though, that this lack of success was due to information not present in the database, rather than a problem with INCA itself. The use of outside knowledge was always going to be a potential problem with the study, but the results from the nine compounds chosen by the author reflect the findings from the first part of the study. We suggest that with a larger knowledge base created by the author, INCA would succeed and cut down the chances of INCA not generating a response. The use of the simple-first rule would also seem to aid INCA's effectiveness in generating the best interpretation for most combinations. The overall implications of this study for the structural alignment view are outlined in the next chapter.

7.7 Chapter Summary

This chapter outlined a study on concept combination. This study's function was to compare the output of INCA against that of human subjects. The study involved a number of subjects generating interpretations for twenty combinations. The responses of these subjects were classified to allow a comparison between INCA and human performance. It was suggested that INCA should always apply the simple-first rule. The possibility of new combination types was discussed. The discovery of new combination types would result in the creation of new rules. But this should only be undertaken when the existence of these combination types is examined further.

Overall, this chapter showed that programs like INCA are slaves to the representations they use. If information is omitted from a representation, it cannot be retrieved later. The combination type rules that INCA uses do cover a good deal of interpretations that human subjects made. When non-classifiable interpretations do occur they tend to be either nonsensical or involve interpretations based on treating a noun as an adjective. However, the INCA system did display a bias towards context-insertion combinations, while human subjects did not often find these types of interpretations. It also repeatedly generated more than one interpretation of the same combination type when subjects generally responded with just one interpretation of the same combination type. In the next chapter we discuss the implications of the present study for the structural alignment view of concept combination.

Chapter 8 - Discussion & Conclusions

8.1 Introduction

This chapter will offer a summary of the thesis and suggest where the SA approach may be taken. We suggest that there are a number of advantages in adopting the SA approach. The most important advantage is that we can attempt to account for the three problems of concept combination (polysemy, world knowledge and figurative interpretations) which were originally outlined in Chapter 1. The implications of the study in the last chapter will be discussed in relation to the SA view and INCA. In relation to the study it appears that there may be new combination types. We also point out a number of problems with INCA. Possible future work is then outlined. This future work includes removing the problems with INCA that the study threw up. Other possible future work involves the automatic generation / formation of the knowledge base from existing knowledge bases. In the conclusions section we argue that SA can be viewed as a framework and that this thesis is an attempt at examining concept combination in terms of SA.

8.2 Summary of Thesis

Concept combination is a highly complex problem. There are several competing theories and these theories disagree on fundamental aspects of concept combination, e.g. the nature of the generation of interpretations. For some theorists the generation of interpretations is best described as a slot-filling process but these theorists disagree over the exact mechanism for choosing the right slot. Others have suggested that combinations are interpreted with respect to a closed set of relations.

Our approach to concept combination is to suggest that SA is the underlying process. This is a process whereby conceptual structures are aligned and similarities found. Essentially, as SA was set out in this thesis it is a process that finds similarities between conceptual structures. These similarities are of two forms: shallow similarity and deep similarity. The former is based on collections of commonalties between concepts and the later is based on shallow similarity (or other deep similarities) and is more structural in nature. Assuming this we outlined how a SA model of concept combination would work.

Our approach suggested that there are six basic combination types (see Section 5.2.1). Each combination type has an associated SA schema (see Section 5.3). The SA schemata are based solely on shallow similarity and deep similarity. These combination types were: meronymous, conjoined, insertion, context insertion, catachrestic and the context combination. They range from combinations that are referential, i.e. reference parts of other concepts, or are constructive and involve the addition of new properties to concepts. The difference between each combination type results from what aspects of the modifier concept and the head concept are aligned and whether this alignment gives rise to a shallow or deep similarity.

We suggested that there were three problems which affect concept combination: (1) the problem of polysemy, (2) the problem of world knowledge and (3) the problem of figurative combinations. In Section 3.4 we outlined how the current theories of concept combination have difficulty in dealing with all three of these problems. Some of the theories may succeed with the problems if they were further modified. But this modification would begin to make the theories more cumbersome, e.g. they would have difficulty in judging when combinations were not literal.

In Section 5.4 we suggested that the SA view can deal with these three problems. With regard to the problem of world knowledge, rich conceptual structures are used. Also all concepts are treated as occurring in a wider domain, with aspects of this domain being evoked as a context. This model has worked relatively well in the field of metaphor where the interpretations that models have to generate are often extremely complex, e.g. Sapper (Veale, 1995). Of course two of our combination types explicitly look for a context, the context insertion combination type and the contextual combination type. Reasoning about the integration of concepts is dealt with purely in terms of SA. No information other than the representation of concepts is used. There are no separate but associated scenarios, e.g. in Dual Process Theory (Wisniewski, 1997a, 1997b), or extra functional models.

There are currently three ways in which polysemous combinations can occur within our model. These are:

1. A combination fits into two (or more) combination types.
2. A combination can be interpreted in similar but distinct ways in terms of the same combination type.
3. A combination involves a mixture of 1 and 2.

The first type is accounted for by applying each of the combination rules to a prospective combination. If a combination satisfies more than one rule then more than one interpretation will be generated. As noted in Section 7.5.2 many interpretations can be classified as insertion combinations. This is an example of the (2) above, although it appears that this holds true more for insertion combination type than any other combination type.

Having suggested how multiple interpretations can be generated we also stated that in general we assumed the best interpretation would be the simplest or from our perspective the first interpretation found. This was dealt with in Section 5.5. However, it is difficult to rate what a best interpretation is, especially when no context is given. When a context is given there would likely be only one interpretation generated and there would be no need to choose a best interpretation.

This model makes no distinction between the figurative and the literal at the semantic level as SA is assumed to be able to account for all combinations. So, the problem of figurative language is actually accounted for in the exact same way as literal language. The separation of combinations into literal and figurative combinations is assumed to be done after the process of concept combination itself. However, looking at the classification of combination types it is clear that some by their definition will give rise to figurative interpretations: context combinations and catachrestic combinations.

8.2.1 Advantages Of The SA Approach

We suggest that overall there are two basic advantages to adopting a SA approach:

1. We can deal with the three problems of concept combination
2. We can take an eirenic approach to a problem that has given rise to many diverse theories

The first advantage has already been dealt with in the previous section. The second advantage is that we can build on the current models. In a sense, the SA view can seen as suggesting that concept combination is a slot-filling process with the mechanics being guided by similarity. That is, the slot-filling is guided by SA between the modifier concept and the head concept. What is new to our approach is that the interaction can involve related concepts (contexts) and we suggest that sometimes the head concept fills a part of the modifier which the previous models do not suggest.

8.3 Implications of Study

One basic question we can ask in relation to the study is: does the SA view offer a good explanation of the data? Before answering this question we should distinguish between the model, the SA view, and the implementation of it, INCA. We will largely focus on the implications for the SA view. Leaving aside weaknesses in the INCA program (e.g. its over-finding of contextual and context insertion combinations) the SA view does seem to offer a good explanation of the data. The combination types offer a good coverage of subjects' interpretations. The study could have shown that human subjects give rise to many interpretations that do not fall within our model's set of combination types but this was not the case.

Where the SA view does seem to require possible modification is with respect to: the possibility of new combination types. One new combination type could be the reverse insertion. This combination type seems to involve a reversal of the head and the modifier roles. Interestingly, when it occurred it did so as an insertion combination, not as any other combination type. It simply seems to involve interpreting a combination, "X Y" as a "Y X". Another possible combination type is the one which seems to check if an interpretation is captured by an existing concept. For example, when a "killer corpse" is interpreted as a *ZOMBIE*. Presumably because the properties of a "killer corpse" are the same or highly similar to a *ZOMBIE*. It is not clear how prevalent each of these combinations is but the latter does seem to be a legitimate combination type. If a combination is interpreted as having properties which are strongly associated with another concept, then perhaps the whole combination refers to this concept. The other combination type could be an artefact of the study as reversals are generally assumed to be rare and it is consequently difficult to come with reasonable examples of this combination type.

8.3.1 Implications For INCA

The study showed that INCA had great problems in dealing with interpretations in the absence of information. This is big problem for these types of programs. INCA needs huge representations and needs information on a concept's attributes, properties and functions. Given the difficulty in hand-coding these representations INCA can only deal with toy examples. The study in the last chapter, though, did involve a network that consisted of over 600 representations. The extension of INCA requires larger knowledge bases. The creation of a larger knowledge base is not a simple task. From a simply practical perspective it would make more sense to work with an existing and accepted and knowledge base, e.g. WordNet (Fellbaum, 1998). In a sense, this would remove the author from having to create the knowledge base and allow him to focus on interpreting combinations and refining the rules which generate the interpretations, this is mentioned in the future work section.

As a side note on the adoption of a large knowledge base, especially WordNet (Fellbaum, 1998), this is not a simple task either. It would involve a complete redesign of INCA as the information that INCA relies on is currently not directly available in WordNet. INCA relies on attributes and functions, and these are listed in the natural language glosses in WordNet. For example, in Figure 7.3 the bracketed information in the sense of surgery lists the attributes. As this information is not directly available and it would have to be extracted from the gloss.

When comparing human performance and INCA we suggest that INCA needs to be changed in two crucial areas:

1. Reduce multiple interpretations of the same combination type.
2. Reduce the bias in INCA for creating contextual and context insertion combinations.

During the study it became clear that INCA will sometimes produce several interpretations that fall into the same combination type category. For example, in Figure 7.1, it is clear that INCA finds more than one interpretation which can be classified as a conjoined combination. Subjects do not appear to do this. When more than one interpretation is found and each interpretation can be classified as a similar combination type the best should be picked. This requires a mechanism for choosing the best interpretation. We suggest that the best interpretation may be the one based on the largest number of similarities.

INCA also appears to generate interpretations when subjects do not. The clearest example of this is in Figure 7.2 where INCA produces a large number of contextual and context insertion combinations compared with the average for the subjects. The contextual and context insertion interpretations are based on similarities that do exit between concepts but subjects are either not aware of them or do not use them as a basis for interpretation. INCA should be limited in the number of interpretations it generates for these combination types. We suggest that a threshold be set where only interpretations based on a large number of similarities are used. The setting of this threshold will require further investigation.

Another implication of the study for INCA is that a model of word sense disambiguation is needed. Theories of concept combination do not account for words that might refer to more than one concept, e.g. presumably the concept *SURGERY* which refers to an activity is not the same as the concept *SURGERY* when it seems to mean the location where GPs work. In fact, a cursory glance of Chapter 3 at the theories of concept combination will show that the examples used tend not have multiple senses, e.g. the concept *ROBIN*. Our study made it quite clear that distinguishing between multiple senses is of critical importance to the interpretation of a combination.

8.4 Suggestions For Future Work

The computer program INCA is in many ways a crude model. This is mainly a result of the program being designed to take advantage of the properties of the knowledge base. However, it could be improved in at least one way, by creating or using a more dynamic model of SA. At present the program uses rules which like all rules are brittle. They are either satisfied or not. These rules also only look for a set of fixed patterns in the network. It is possible to change this set of fixed patterns.

The SA view has little to say on the role of diagnosticity in concept combination. However, we have suggested that the range of interpretations that can arise in the insertion combination type can be large. Perhaps diagnosticity could act as a constraint for this particular combination type. Indeed, INCA can already work out the diagnosticity of each property in the network but this information is not currently used. The next version of INCA should investigate the role of diagnosticity in concept combination. It may be the case that diagnosticity only has an impact on certain combination types.

Making a more dynamic program and examining diagnosticity does not, however, deal with the problem of having to create the initial knowledge base. Perhaps a more "macro" approach is more suitable, e.g. say, taking a large knowledge base and work with this. There are no large collections of frames or frame-like representations available publicly for research. So these frames must be created. Ideally methods should be developed for automatically creating the frame representations that INCA needs. This could possibly be done by data mining a knowledge base such as WordNet (Fellbaum, 1998).

Last but not least is the massive problem of word sense disambiguation, which was mentioned in the previous section. Clearly, in this work we have assumed that nouns have a primary sense and we tended to use examples where there is only sense for each noun in the compound. This is in line with the current theories of concept combination but some development needs to be made on this thorny issue. The knowledge base does not bar a word having multiple senses. All that is required is that each sense is given a different name (in our approach each sense should be marked by a different atom). A further level could be added where a word points to a list of different concepts. For example, the word surgery could point to, *SURGERY1*, *SURGERY2* and so on. Entries in the knowledge base like *SURGERY1* could capture one particular concept associated with the word surgery. This would add another dimension to the outline of polysemy in our model.

To summarise, the future work that we envisage as an immediate consequence of the current thesis is the following:

1. Create / Use a more dynamic model of SA.
2. Investigate the role of diagnosticity.
3. Create large knowledge base.
4. Incorporate multiple senses

The four areas of future work are relatively straightforward to implement, although they would require a great deal of work, e.g. (1) involves a near total re-write of INCA. The development of a new knowledge base does not necessarily entail developing a new version of INCA but we feel that both tasks (1) and (3) should be undertaken together. Task (2) relates more to the SA model presented in this thesis rather INCA, however, the role of diagnosticity could be studied separately from INCA. The fact that diagnosticity scores are available in INCA for all properties in the knowledge base will prove a valuable tool in studying this problem. The incorporation of multiple senses is something that should be examined in more detail, as theories of concept combination avoid this problem. Indeed, task (4) could be carried out while task (3) is being undertaken.

8.5 Conclusions

The SA model of concept combination is an attempt to take a multidisciplinary approach to the complex problem of concept combination. This model suggests that the interpretations of combinations can be broadly classified into one of six combination types. These combination types were chosen to offer the largest coverage. With respect to the study it appears that the combination types are successful in dealing with most interpretations that people generate. For each combination type a SA schema was proposed to explain how interpretations are generated. These schemas were based on deep similarities and shallow similarities. In the study, the interpretations generated by INCA were similar to those that human subjects generated, although they were not as concise.

The SA model of concept combination outlined in this thesis does not offer a complete solution to concept combination. It does, however, offer the researcher a framework to examine concept combination while also dealing with some of its thorniest problems. There are many models of metaphor and analogy based on SA. Other approaches could be taken that apply SA but do so in radically different ways to the one we have suggested in this thesis. By this we mean that analogy has been investigated in at least three different ways: ACME (Holyoak & Thagard, 1989), Sapper (Veale 1995) and SME (Forbus, Ferguson & Gentner, 1994; Falkenhainer, Forbus & Gentner, 1989). Each claims to be a SA approach but each is quite different and so SA can be viewed as a framework to investigate concept combination.

If SA is a framework for the investigation of concept combination then other SA approaches are possible. The work in this present thesis could be seen as the tentative first steps in exploring concept combination through SA. A researcher could quite easily create another model of concept combination which uses different mappings rules (in truth, probably more sophisticated ones). This new model and the model we outline could be quite different entities. But both would have the advantage of being able to deal with: the problem of polysemy, the problem of world knowledge and the problem of figurative interpretations.

With respect to the current approach outlined in this thesis it appears that the combination types do offer a wide coverage of the combinations that people generate. At the same time, it is clear that other combination types may exist. The INCA implementation of the SA view has a number of quirks (see Section 8.3.1) but that these could be dealt with (see Section 8.4). The implementation of the SA approach could also be done using a less-rule based model. However, the biggest hurdle in creating a new implementation is the creation of a larger knowledge base. This knowledge base would capture the multiple senses of words.

8.6 Chapter Summary

This chapter offered a brief recap on the main topics of this thesis, most importantly the SA model of concept combination and the implications of the study. We suggest that there are at least two advantages in adopting the SA model. The most important advantage is that we can deal with the three problems of concept combination which were originally outlined in Chapter 1. The other advantage is that the model can incorporate and develop parts of the current theories of concept combination, e.g. providing a similarity-based mechanism for slot-filling.

The implications of the study in the last chapter were also set out in relation to the SA model and INCA. One implication was that there may be new combination types. It was also pointed out that a number of problems exist with INCA. The possible future work we feel is necessary with respect to this thesis was then outlined. This work includes removing the problems with INCA that the study threw up. Other suggested future work involved the automatic generation / formation of the knowledge base from existing knowledge bases. Finally, in the conclusion section we argue that SA can be viewed as a framework and that this thesis can be described as one attempt at examining concept combination in terms of SA.

Bibliography

Aitchison, J. (1992). *Linguistics*. - London : Hodder & Stoughton.

Aitchison, J. (1997). *The language web: the power and problem of words*. Cambridge: Cambridge University Press.

Black, M. (1962). *Models and Metaphors*. Ithaca, NY: Cornell University Press.

Bowdle, B., & Gentner, D. (1999). Metaphor comprehension: From comparison to categorization. *Proceedings of the Twenty First Annual Conference of the Cognitive Science* Society, 90-95. Mahwah, NJ: Lawrence Erlbaum Associates.

Bratko, I. (1990). *Prolog: programming for artificial intelligence*. Wokingham, UK: Addison Wesley.

Brown, E. K. (1980). *Syntax: a linguistic introduction to sentence structure*. London: Hutchinson.

Canon, G. (1987). *Historical change and English Word-formation*. New York: Lang.

Costello, F. (1996). *Noun-Noun conceptual combination: the polysemy of compound phrases*. PhD Thesis, University of Dublin, Dublin, Ireland.

Costello, F. J. & Keane, M.T. (1997). Polysemy in Conceptual Combination: Testing the Constraint Theory of Combination. In *Nineteenth Annual Conference of the Cognitive Science Society*, M. G. Shafto, P Langley (Eds.). Hillsdale, N.J.: Erlbaum. pp 137-142.

Costello, F. J., & Keane, M. T. (1998). Alignment versus diagnosticity: A critical test of two theories of conceptual combination. Published in the Proceedings of *AICS 98, 9th Irish Conference on Artificial Intelligence & Cognitive Science*. pp. 29-45.

Costello, F. J. & Keane, M. T. (2000). Efficient creativity: Constraint guided conceptual combination. *Cognitive Science,* 24(2), pp. 299-349.

Eysenck, M. W. & Keane, M. T. (1995). *Cognitive psychology: a student's handbook*. UK: Hove.

Falkenhainer, B., Forbus, K.D. & Gentner, D. (1989). The structure-mapping engine: algorithm and examples, *Artificial Intelligence*, 41, pp. 1-63.

Fellbaum, C. (1998). *WordNet, An Electronic Lexical Database*. The MIT Press.

Ferguson, R. W., Forbus, K. D., & Gentner, D. (1997). On the proper treatment of noun-noun metaphor: A critique of the Sapper model. *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society*, pp 913.

Forbus, K.D., Ferguson, R.W. & Gentner, D. (1994). Incremental structure-mapping. In Procedingss. *Sixteenth Annual Conference of the Cognitive Science Society*, pp.313--318.

Frege (1984/1892). *Collected papers on mathematics, logic, and philosophy*. Oxford: B. Blackwell.

Gardner, H. (1985). *The mind's new science : a history of the cognitive revolution*. New York : Basic Books

Gagné, C. L. (2001). Metaphoric interpretations of comparison-based combinations. *Metaphor and Symbol*. (in press).

Gagné, C. L., & Shoben, E. J. (1997). The influence of thematic relations on the comprehension of modifier-noun combinations. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 23, 71-87.

Gentner, D., Rattermann, M. J., & Forbus, K. D. (1993). The roles of similarity in transfer: Separating retrievability from inferential soundness. *Cognitive Psychology*, 25, 524-575.

Gentner, D., and A. B. Markman. (1997). Structure-mapping in analogy and similarity. *American Psychologist,* 52, 45-56.

Gentner, D. (1998). Analogy. In W. Bechtel & G. Graham (Eds.), *A companion to cognitive science* (pp. 107-113). Oxford: Blackwell.

Givon, T. (1979). *On understanding grammar.* New York : Academic Press,

Hayes, J. & Veale, T. (1999). A Structure Mapping Model of Conceptual Combination. Presented at *ICLA'99: International cognitive linguistics association conference*, Stockholm University, Stockholm, Sweden.

Hoenkamp, E., & de Groot, R. (2000). Finding relevant passages using noun-noun components. In M. Hearst, F. Gey, & R. Tong (Eds.), Proceedings of *the 23rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 385–387). New York: ACM.

Holyoak, K. J. and P. Thagard (1989). Analogical mapping by constraint satisfaction. *Cognitive Science*, 13, 295-355.

James, W. (1890/1995). *Principles of Psychology: Vol. 1.* New York: Dover Publications.

Keane, M.T. & Costello, F. J. (1997). Where do "Soccer Moms" Come From? : Cognitive Constraints on Noun-Noun Compounding in English. In Tony Veale (Ed.) *Proceedings of MIND II. Dublin: Dublin City University.*

Keane, M. T., & Costello, F. J. (2001). Setting Limits on Analogy: Why Conceptual Combination is Not Structural Alignment. In *The Analogical Mind: Perspectives from Cognitive Science*, D Gentner, K. J. Holyoak, B. N. Kokinov (Eds.). Cambridge, MA: MIT Press. pp 172-198.

Lakoff, G. & Johnson, M. (1980). *Metaphors we live by.* Chicago: University of Chicago Press.

Lakoff, G. (1993). The contemporary theory of metaphor. In Ortony, A., editor, *Metaphor and Thought.* Cambridge University Press, Cambridge. pp 202-251

Levi, J. N. (1978) *The Syntax and Semantics of Complex Nominals.* New York: Academic Press.

Linguistic Data Consortium (1995). Penn Treebank, CD-Rom, Release 2.

Luce, R.D. & Raffia, H. (1957). *Games and Decisions: Introduction and Critical Survey.* New York: John Wiley & Sons. (As cited in Gagné & Schoben, 1997).

Luger, G. F. & Stubblefield, W. A. (1998). *Artificial Intelligence: Structures and Strategies for Complex Problem Solving.* Reading, Mass: Addison-Wesley.

Lycan, W (1999). *Philosophy of Language: A Contemporary Introduction.* London: Routledge.

Marcus, Santorini & Marcinkiewicz (1993). Building a large annotated corpus of English: The penn treebank. *Computational Linguistics*, 19, 313-330.

Markman, A. B., & Gentner, D. (1993). Structural alignment during similarity comparisons. *Cognitive Psychology*, 25, 431-467.

Markman, A.B, & Wisniewski, E.J. (1997). Similar and different: The differentiation of basic level categories. *Journal of Experimental Psychology: Learning, Memory, and Cognition,* 23(1), 54- 70.

Markman, A. B., & Gentner, D. (1997). The effects of alignability on memory storage. *Psychological Science*, 8(5), 363-367.

Markman, A. B., & Gentner, D. (2000). Structure-mapping in the comparison process. *American Journal of Psychology*, 113(4), 501-538.

Miller, G., Beckwith, R., Fellbaum, C., Gross, D. & Miller, K. (1993). Five papers on WordNet. *Technical report, Stanford University.*

Minsky, M. (1975). A framework for representing knowledge. In P. H. Winston, Ed., *The Psychology of Computer Vision.* New York: McGraw-Hill, pp. 211-277.

Murphy, G. L. (1988). Comprehending complex concepts. *Cognitive Science*, 12, 529-562.

O'Grady, W. D.; Dobrovolsky, M & Aronoff, M (1993). *Contemporary linguistics : an introduction.* New York : Longman.

Ortony, A. (1979). Beyond literal similarity. *Psychological Review*, 161-180.

Rosch, E. (1975). Cognitive representations of semantic categories. *Journal of Experimental Psychology: General,* 104:192-232.

Rosch, E., and Lloyd, B. B. (1978). *Cognition and Categorization*. Hillsdale, NJ: Erlbaum.

Russell, S.J. & Norvig, P. (1995), *Artificial Intelligence: A Modern Approach*. Englewood Cliffs, N.J.: Prentice Hall.

Santorini, B. (1990). Part-of-Speech tagging guidelines for the Penn Treebank Project (3rd revision, 2nd Printing). Download Spring 2002, from:
http://www.cis.upenn.edu/~treebank/tagguide.ps.gz

Saeed, J. (1995). *Semantics*. Oxford: Blackwell Publishers.

Smith, E. E., Osherson, D. N., Rips, L. J. and Keane, M. (1988). Combining Prototypes: A Selective Modification Model. *Cognitive Science*, 12(4): 485-527

Sowa, J. F. (1984). *Conceptual Structures: Information Processing in Mind and Machine*. London: Addison-Wesley.

Spellman, B. A., & Holyoak, K. J. (1992). If Saddam is Hitler then who is George Bush? Analogical mapping between systems of social roles. *Journal of Personality and Social Psychology*, 62, 913-933.

Taylor, I. & Taylor, M. M. (1990). *Psycholinguistics: Learning and Using Language*. Prentice-Hall.

Turner, M. (1997). *Death is the mother of beauty: Mind, Metaphor, Criticism*. Chicago: University of Chicago Press.

Tversky, A (1977). Features of similarity. *Psychological Review*, 327-352.

Veale, T. (1995). *Metaphor, Memory and Meaning: Symbolic and Connectionist Issues in Metaphor Interpretation*. PhD Thesis, University of Dublin, Dublin, Ireland.

Wills, A. J., Reimers, S., Stewart, N., Suret, M., & McLaren I. P. L. (2000). Tests of the ratio rule in categorization. *Quarterly Journal of Experimental Psychology, 53A*, 983-1011.

Wisniewski, E. J. (1996). Construal and similarity in conceptual combination. *Journal of Memory and Language*, 35, 434-453.

Wisniewski, E. J. (1997a). When concepts combine. *Psychonomic Bulletin & Review*, 4(2), pp. 167-183.

Wisniewski, E. J. (1997b). Conceptual combination: Possibilities and esthetics. In T.B. Ward, S.M. Smith, & J. Vaid (Eds.), Creative thought: An investigation of conceptual structures and processes,(pp. 51-81) Washington, D.C.: American Psychological Association.

Wittgenstein, L. (1968). *Philosophical investigations*. Oxford : Basil Blackwell.

Appendix A - Corpus Study

Section 1 - Tags used in Penn Treebank corpus

Tag	Description
$	dollar
``	opening quotation mark
''	closing quotation mark
(	opening parenthesis
)	closing parenthesis
,	comma
--	dash
.	sentence terminator
:	colon or ellipsis
CC	conjunction, coordinating
CD	numeral, cardinal
DT	determiner
EX	existential there
FW	foreign word
IN	preposition or conjunction, subordinating
JJ	adjective or numeral, ordinal
JJR	adjective, comparative
JJS	adjective, superlative
LS	list item marker
MD	modal auxiliary
NN	noun, common, singular or mass
NNP	noun, proper, singular
NNPS	noun, proper, plural
NNS	noun, common, plural
PDT	pre-determiner
POS	genitive marker
PRP	pronoun, personal
PRP$	pronoun, possessive
RB	adverb
RBR	adverb, comparative
RBS	adverb, superlative
RP	particle
SYM	symbol
TO	"to" as preposition or infinitive marker
UH	interjection
VB	verb, base form
VBD	verb, past tense
VBG	verb, present participle or gerund
VBN	verb, past participle
VBP	verb, present tense, not 3rd person singular
VBZ	verb, present tense, 3rd person singular
WDT	WH-determiner
WP	WH-pronoun
WP$	WH-pronoun, possessive
WRB	Wh-adverb

Section 2 - Sample programs used in Corpus study

1. "NounNoun" program

This program searches through a file and looks for nouns. If nouns are found then subsequent neighbours are checked to see if they too are nouns. If they are then they are printed to the screen.

NounNoun.java

```
import java.io.*;
import java.util.*;


public class NounNoun {

        static Vector lineSpace = new Vector();

        public static void main(String[] args) throws IOException {

                if( args.length != 1 )
                        System.err.println( "Usage: NounNoun <source file> " );

                else{
                  System.out.println( "\n\nResults taken from: " + args[0] + "\n\n");
                        processFile( args[0] );
                }
        }


        // Process the file that was passed in as the 1st arg
        public static void processFile( String in ){ //throws IOException{



                try{

                        BufferedReader source = new BufferedReader( new FileReader( in) );

                        String line;


                        int note = 0;
                        while( ( line = source.readLine() ) != null )
                        {

                          // System.out.println( "At line: " + ++note + "\n");
                                if( line.length() > 1 ) {

                                        StringTokenizer st = new StringTokenizer(line, " \n\r\t\f;,:)(=[]");

                                        while( st.hasMoreTokens() ) {

                                          String temp = st.nextToken();
                                          if( temp != "/" )
                                                lineSpace.addElement( temp );


                                        }

                                        // printVector( lineSpace );
                                        processLine( lineSpace );
                                        lineSpace.removeAllElements();
                                }
                        }

                        // Close Streams
                        source.close();
                        // target.close();
```

```
                }
                catch( java.io.IOException ioe) {
                        System.out.println (ioe.getMessage());
                }

        }


// Process each in the given file
public static void processLine( Vector line ){

        Vector compound = new Vector();

        // System.out.println( " Word space processed..." );

        // How many of these words are nouns
        for( int i = 0; i < line.size() -1 ; i++ ){

          String temp = ( String ) line.elementAt( i );

          if( isNoun( temp ) ){

                String neighbour = getNeighbour( line, i );

                if( neighbour != null && isNoun( neighbour ) ) {

                  compound.addElement( temp );
                  compound.addElement( neighbour );
                  int mark = i + 1;

                  String neighbour2 = getNeighbour( line, mark++ );

                  while( neighbour2 != null && isNoun( neighbour2     ) ){
                        compound.addElement( neighbour2 );
                        neighbour2 = null;
                        neighbour2 = getNeighbour( line, ++mark );
                  }
                i =  ( mark -1 )  ;
                }



                if( compound.size() > 1 ) {
                  printVector( compound);
                  compound.removeAllElements();
                }
          }


        }
  }

        // Get a token's neighbour
        public static String getNeighbour( Vector Space, int mark ){

                        if( mark >= Space.size() - 1 )
                                return null;
                        String temp = ( String ) Space.elementAt( ( mark + 1 ) );
                        return temp;
        }

        // Is a string a noun?
         public static boolean isNoun( String temp){

                if( temp.endsWith( "/NN" ) ||
                  temp.endsWith( "/NNS" ) ||
                  temp.endsWith( "/NNPS" ) ||
                  temp.endsWith( "/NNP" ) ) {
                  return true;
        }

        return false;
```

```
            }

            // Prints the compound
            public static void printVector( Vector compound){

                        for( int i = 0; i < compound.size(); i++ ){

                                    System.out.print( compound.elementAt( i ) + " " );
                        }
                        System.out.println( "\n");

            }
}

// End of file
```

2. "collect_nouns" script

This c-shell script calls the "NounNoun" program on a series of files in the Penn treebank. The directories in the script refer to the actual directories where the Penn treebank is stored locally.

collect_nouns

```
#!/bin/csh -f

# Info is stored here:
cd /cl_resources/penn_treebank/tagged/wsj

# Collect Info & put here:
touch ~/Corpus/Results/Noun_compound_list

foreach d (* )
            cd /cl_resources/penn_treebank/tagged/wsj/$d

            foreach f (* )
                        echo $f
                        java -classpath ~/Corpus/Corpus_tools NounNoun $f >>~/Corpus/Results/Noun_compound_list
            end

            cd /cl_resources/penn_treebank/tagged/wsj
end

# End of file
```

Further processing was done using the UNIX commands SORT and UNIQ to organise the generated files.

Appendix B - INCA (Sample Code)

Sample Concepts:

```
/* Some more concepts... */

% bike

:- create_concept( bike,
      [
          [ isa, 0.9, vehicle ],
        [ perform, transport ],
        [ part, handlebar, frame, wheels ],
      [ wheels, two ]
          ] ).


:- create_concept( frame,
      [
          [ isa, 0.9, structure ],
        [ perform, support ],
        [ part,  paint ]
         ] ).
```

Main Algorithm:

"inca.prolog"

```
%=======================================================
% Filename: inca.prolog
%
% Purpose: the is the main procedure for interpreting
% combinations. The order in which combinations are
% found is as follows:
%
%       1. Meronymous
%       2. Conjoined
%       3. Insertion
%       4. Context insertion
%       5. Catachrestic
%       6. Contextual
%
% Date: Summer 2002
%=======================================================


inca( M, H, [ A1, A2, A3, A4, A5, A6 ], 1) :-
      % 1. Meronymous
      find_meronymous( M, H, A1 ),
      % 2. Conjoined
      find_conjoined ( M, H, A2 ),
      % 3. Insertion
      find_insertion( M, H, A3 ),
      % 4. Context insertion
      find_context_insertion( M, H, A4 ),
      % 5. Catachrestic
      find_catachrestic( M, H, A5 ),
      % 6. Contextual
      find_contextual( M, H, A6 ).

% 1. Meronymous
```

```
find_meronymous( M, H, A1 ) :-
      meronymous( M, H, A1 ).

find_meronymous( M, H, [] ).

% 2. Conjoined
find_conjoined( M, H, A1 ) :-
      conjoined( M, H, A1 ).

find_conjoined( M, H, [] ).

% 3. Insertion
find_insertion( M, H, A1 ) :-
      insertion( M, H, A1 ).

find_insertion( M, H, [] ).

% 4. Context insertion
find_context_insertion( M, H, A1 ) :-
      context_insertion( M, H, A1 ).

find_context_insertion( M, H, [] ).

% 5. Catachrestic
find_catachrestic( M, H, A1 ) :-
      catachrestic( M, H, A1 ).

find_catachrestic( M, H, [] ).

% 6. Contextual
find_contextual( M, H, A1 ) :-
      contextual( M, H, A1 ).

find_contextual( M, H, [] ).



% Given a combination - generate an interpretation

inca( M, H, [ A1, A2, A3, A4, A5, A6 ], 2) :-
      % 1. Meronymous
      find_meronymous2( M, H, A1 ),
      % 2. Conjoined
      find_ conjoined2( M, H, A2 ),
      % 3. Insertion
      find_insertion2( M, H, A3 ),
      % 4. Context insertion
      find_context_insertion2( M, H, A4 ),
      % 5. Catachrestic
      find_catachrestic2( M, H, A5 ),
      % 6. Contextual
      find_contextual2( M, H, A6 ).

% 1. Meronymous
find_meronymous2( M, H, A1 ) :-
      gen_meronymous( M, H, A1 ).


% 2. Conjoined
find_conjoined( M, H, A1 ) :-
      gen_conjoined( M, H, A1 ).
```

```
% 3. Insertion
find_insertion2( M, H, A1 ) :-
      gen_insertion( M, H, A1 ).

% 4. Context insertion
find_context_insertion2( M, H, A1 ) :-
      gen_context_insertion( M, H, A1 ).

% 5. Catachrestic
find_catachrestic2( M, H, A1 ) :-
      gen_catachrestic( M, H, A1 ).


% 6. Contextual
find_contextual2( M, H, A1 ) :-
      gen_contextual( M, H, A1 ).


% This version just prints out the results
% neatly to the screen...


inca( M, H ) :-
      inca( M, H, X, 2 ),
      print_results2( X ).

print_results([]).

print_results( [null|T] ) :-
      print_results2( T ).

print_results2( H ) :-
      H = [C1, C2, C3, C4, C5, C6 ],
      write_interpretation1( C1 ),
      write_interpretation2( C2 ),
      write_interpretation3( C3 ),
      write_interpretation4( C4 ),
      write_interpretation5( C5 ),
      write_interpretation6( C6 ).

write_interpretation1( null ).

write_interpretation1( X ) :-
      write( ' Meronymous: ' ), nl,
      write_list_2( X ).

write_interpretation2( null ).

write_interpretation2( X ) :-
      write( ' CONJOINED: ' ), nl,
      write_list_2( X ).


write_interpretation3( null ).

write_interpretation3( X ) :-
      write( ' INSERTION: ' ), nl,
      write_list_2( X ).
```

```
write_interpretation4( null ).

write_interpretation4( X ) :-
      write( ' CONTEXT INSERTION: ' ), nl,
      write_list_2( X ).

write_interpretation5( null ).

write_interpretation5( X ) :-
      write( ' CATACHRESTIC: ' ), nl,
      write_list_2( X ).

write_interpretation6( null ).

write_interpretation6( X ) :-
      write( ' CONTEXTUAL: ' ), nl,
      write_list_2( X ).

write_list_2( [] ).

write_list_2( [H|T ] ) :-
      H = [_,_ | X ],
      write( '   ' ), write( X ), nl,
      write_list_2( T ).

% Version of INCA that applies the "simple first rule"
inca( M, H, X, simple_first) :-
      simple_first( M, H, X ).

inca( M, H, simple_first) :-
      inca( M, H, X, simple_first), nl,
      write( ' SIMPLEST FIRST: ' ), nl,
      write( '    ' ), write( X ), nl.

simple_first( A, B, C ) :-
      % 1. Meronymous
      meronymous( A, B, C ).

simple_first( A, B, C ) :-
      % 2. Conjoined
      conjoined ( A, B, C ).

simple_first( A, B, C ) :-
      % 3. Insertion
      insertion( A, B, C ).

simple_first( A, B, C ) :-
      % 4. Context insertion
      context_insertion( A, B, C ).

simple_first( A, B, C ) :-
      % 5. Catachrestic
      catachrestic( A, B, C ).

simple_first( A, B, C ) :-
      % 6. Contextual
      contextual( A, B, C ).
```

The Combination Types:

"meronymous.prolog"

```
/* Meronymous combination type... */

% Version 1 - Meronymy based on structure

meronymous( A, B ) :-
      subconcept( A, C ),
      identical( B, C ),
      relation( A, B, _, _ ).

meronymous( A, B, [meronymous1, Rel, B, A ] ) :-
      subconcept( A, C ),
      identical( B, C ),
      relation( A, B, Rel, W ).

identical( A, B ) :-
      % They should have the same number of relations
      how_detailed( A, Z ),
      how_detailed( B, Z ),
      % Each subconcept should map onto one subconcept
      % in the other...
      the_same( A, B ).
      % Important this still has to be done!

the_same( A, B ) :-
      all_subconcepts1( A, List1 ),
      all_subconcepts1( B, List2 ),
      aux_the_same( List1, List2 ).

aux_the_same( [], _ ).

aux_the_same( [ H | T ], L ) :-
      member( H, L ),
      aux_the_same( T, L ).

aux_the_same( _, _ ) :-
      fail.

subconcept( A,B ) :-
      relation( A, B, _, _ ).

% Version 2 - Meronymy based on certain relations

%% It appears that only certain relations may be involved
%% indexical combinations

meronymous( A, B, [meronymous2, Rel, B, A ] ) :-
      relation( A, B, Rel, W ),
      meronymous_relations( Rel ).

meronymous_relations( control ).
meronymous_relations( works_in ).
meronymous_relations( part ).
meronymous_relations( affect ).
meronymous_relations( uses ).
```

"conjoined.prolog"

```
/* Conjoined */

% Establish an equivalence between the two concepts

% Deep similarity version

conjoined( Base, Target, [ conjoined, Base, Target, [ both, are,
Scenario, [C, C1] ] ] ) :-

      % For an equivalence to be established
      % then the base and target must share
      % a related scenario
      find_related_scenario_1( Base, Target, Scenario ),
      subconcept( Base, C ),
      subconcept( Target, C1 ),
      deep_similarity( C, C1, Z ).

% Shallow similarity version

conjoined( Base, Target, [ conjoined, Base, Target, [ both, are,
Scenario, [C, C1] ] ] ) :-

      % For an equivalence to be established
      % then the base and target must share
      % a related scenario
      find_related_scenario_1( Base, Target, Scenario ),
      subconcept( Base, C ),
      subconcept( Target, C1 ),
      shallow_similarity( C, C1, Z ).

find_related_scenario_1( Base, Target, Scenario ) :-
   relation( Base, Scenario, isa, _ ),
   relation( Target, Scenario, isa, _ ).

% Special rules for people

conjoined( Base, Target, [ conjoined, Base, Target, 'Both people?' ]
) :-
      relation( Base, person, isa, _ ),
      relation( Target, person, isa, _ ),
      diff( Base, Target ).
```

"insertion.prolog"

```
%=================================================
% Filename: insertion.prolog
%
% Purpose: desribes the basic rule(s) for the insertion
% combination type.
%
% Date: Summer 2002
%=================================================

% Deep similarity version

insertion( A, B ) :-
      subconcept( B, C ),
```

```
        deep_similarity( A, C, Z ).

insertion( A, B, [ insertion, B, Rel, A, [replacing, C] ]) :-
        % A matches a subpart of B
        subconcept( B, C ),
        deep_similarity( A, C, Z ),
        relation( B, C, Rel, _ ).

% Shallow similarity version

insertion( A, B ) :-
        subconcept( B, C ),
        shallow_similarity( A, C, Z ).

insertion( A, B, [ insertion, B, Rel, A, [replacing, C] ]) :-
        % A matches a subpart of B
        subconcept( B, C ),
        shallow _similarity( A, C, Z ),
        relation( B, C, Rel, _ ).


insertion( A, B ) :-
        % A matches a subpart of B

        subconcept( B, C ),
        shallow _similarity( A, C, Z ).

% Another version...

insertion( A, B, [ insertion2, B, A, [replacing, D, in, C] ]) :-
        % A matches a subpart of B

        subconcept( B, C ),
        subconcept( C, D ),
        relation( C, D, _, _ ),
        deep_similarity( A, D, Z ).

insertion( A, B, [ insertion2, B, A, [replacing, D, in, C] ]) :-
        % A matches a subpart of B

        subconcept( B, C ),
        subconcept( C, D ),
        relation( C, D, _, _ ),
        shallow_similarity( A, D, Z ).
```

"context_insertion.prolog"

```
/* Insertion... */

% Context Insertion

context_insertion( A, B ) :-
        % a subpart of A matches a subpart of B

        subconcept( B, C ),
        subconcept( A, D ),
        deep_similarity( D, C, _ ).

context_insertion( A, B, [ context_insertion, B, Rel, D, [replacing,
C] ]) :-
        % a subpart of A matches a subpart of B
```

```
      subconcept( B, C ),
      subconcept( A, D ),
      deep_similarity( D, C, _ ),
      relation( B, C, Rel, _ ).

context_insertion( A, B ) :-
      % a subpart of A matches a subpart of B

      subconcept( B, C ),
      subconcept( A, D ),
      shallow_similarity( D, C, _ ).

context_insertion( A, B, [ context_insertion, B, Rel, D, [replacing,
C] ]) :-
      % a subpart of A matches a subpart of B

      subconcept( B, C ),
      subconcept( A, D ),
      shallow _similarity( D, C, _ ),
      relation( B, C, Rel, _ ).
```

"catachrestic.prolog"

```
/* Catachrestic combination... */

%
% Filename: catachrestic.prolog
%
% Purpose: defines the catachrestic combination type
% where the head is metaphoric reference to an aspect
% of the modifier.
%
% Date: Summer 2002
%

catachrestic( A, B ) :-
      subconcept( A, C ),
      \+( subconcept( A, B ) ),
      shallow_similarity( C, B, _ ).

catachrestic( A, B, [ catachrestic, B, =, Rel, C ] ) :-
      Subconcept( A, C ),
%     \+( subconcept( A, B ) ),
      shallow_similarity( C, B, L ),
      relation( A, C, Rel, _ ).
```

"context.prolog"

```
/* Contextual */


contextual( A, B ) :-
      context( A, C ),
      deep_similarity( C, B, _ ).

% The bigger the similarity the greater the score
```

```
contextual( A, B, [ contextual, C, B, Score ]  ) :-
      context( A, C ),
      deep_similarity( C, B, Z ),
      length( Z, Score ).

% What's a context

context( A, B ) :-
      relation( B, A, _, _ ).


context( A, B ) :-
      relation( A, B, _, _ ).
```

Procedure for gathering interpretations:

"generator.prolog"

```
%
% Filename: generator.prolog
%
% Purpose: finds examples of particular combination
% types. These procedures show off waht INCA can find
% or create.
%
% Date: Summer 2002
%

% Meronymous Combinations

gen_meronymous( A, B, L ) :-
      setof( [ A, B, Z ], meronymous( A, B, Z ), L ).

gen_meronymous( A, B, null ).

% Identity Combinations

gen_identity( A, B, L ) :-
      setof( [ A, B, Z ], identity( A, B, Z ), L ).

gen_identity( A, B, null ).

% Insertion Combinations

gen_insertion( A, B, L ) :-
      setof( [ A, B, Z ], insertion( A, B, Z ), L ).

gen_insertion( A, B, null ).

% Context Insertion Combinations

gen_context_insertion( A, B, L ) :-
      setof( [ A, B, Z ], context_insertion( A, B, Z ), L ).

gen_context_insertion( A, B, null ).

% Catachrestic Combinations

gen_catachrestic( A, B, L ) :-
      setof( [ A, B, Z ], catachrestic( A, B, Z ), L ).
```

```
gen_catachrestic( A, B, null ).

% Context Combinations

gen_contextual( A, B, L ) :-
        setof( [ A, B, Z ], contextual( A, B, Z ), L ).

gen_contextual( A, B, null ).
```

Procedure for finding similarities:

"comparison.prolog"

```
%
% Filename: comparison.prolog
%
% Purpose: find similarities between concepts in the
% network.
%
% Date: Summer 2002
%

%============================================================


% Similarities


% Shallow similarities are groups commonalities


commonality( A, B ) :-
      relation( A, C, F, _ ),
      relation( B, C, F, _ ),
      A \== C,
      B \== C.


commonality( A, B, [ C, F ] ) :-
      relation( A, C, F, _ ),
      relation( B, C, F, _ ),
      A \== C,
      B \== C.


% Shallow similarity


shallow_similarity( A, B, L ) :-
      setof( X, commonality( A, B, X ), L ).


shallow_similarity ( A, B, L ) :-
      setof( X, commonality( A, B, X ), L ),
      length( L, Num2 ),
```

```
    Num2 > 1.

shallow_sim( A, B, _ ) :-
    fail.

% Deep similarity

deep_sim2( A, B, C ) :-
    deep_similarity ( A, B, C ).

deep_similarity ( A, B, R ) :-
    A \== B,
    relation( A, C, F, _ ),
    relation( B, D, F, _ ),

    % This is to help stop loops while backtracking
    % ...some of these checks are repetitious?
    C \== D,
    C \== B,
    C \== A,
    A \== C,
    A \== D,
    A \== B,
    D \== B,
    D \== A,
    D \== C,

    similar1( C, D, R ).

similar1( X, Y, Z ) :-
    shallow_similarity ( X, Y, Z ).

similar1( X, Y, Z ) :-
    deep_similarity ( X, Y, Z ).
```

Appendix C - List of concepts

List of concepts with which are in the knowledge base of INCA on loading.

1. bike
2. car
3. wheel
4. robin
5. snake
6. scales
7. skin
8. mouth
9. beak
10. belly
11. red_breast
12. hunting
13. prey
14. hunter
15. drug
16. user
17. baron
18. serf
19. dealer
20. drug_dealer
21. fur_dealer
22. dealing
23. seller
24. buyer
25. commodity
26. house
27. building
28. dwelling
29. wall
30. roof
31. door
32. window
33. floor
34. boat
35. fires_weapons
36. porthole
37. deck
38. coat
39. fur
40. mink
41. leather
42. bottle
43. long_narrow_piece
44. bottle_part
45. opening
46. neck
47. subpart
48. whole

Appendix D - The Concept Combination Study

This appendix is divided into three sections:

Section 1 - The forty eight combinations drawn from Sapper
Section 2 - The form
Section 3 - The data

Section 1 - The forty-eight combinations drawn from Sapper

accountant architect author believer brain butcher
cake carcass casualty chef cherry composer
corpse country criminal dough edifice experiment
eye face general hacker illusion machete
magic magician man martyr marzipan
modernism novel opera plot politician
priest raisin religion scalpel scientist sculptor skiing
slaughter snow soldier surgeon surgery sword tree

Section 2 - The form

(Question 18 was the one that was omitted from the study).

Survey

Background:

Many ideas in English are expressed in the form of compounds, e.g. "web surfer", "arms race". Compounds which are novel to speakers often have more that one meaning. For example, the novel compound "robin snake" may mean:

"a snake that hunts robins"
"a snake with a red underbelly"
"a snake that is hunted by robins"
"a snake with a beak instead of a mouth"

Instructions:

In this survey you will be presented with a number of novel compounds. For each compound you should write down as many meanings as possible. You may find that some compounds have only one reasonable meaning or you may find that there is no reasonable meaning for the compound. If you do find more than one meaning for a compound please tick or mark the one which you think is the best interpretation.

In the box below four interpretations for "robin snake" are given and the best one is marked. The choice of which meaning is the best one is completely subjective. There is no right or wrong answer - in the survey just mark the meaning which you think is the best description of the compound in question.

Robin Snake

1. *a snake that hunts robins **
2. *a snake with a red underbelly*
3. *a snake that is hunted by robins*
4. *a snake with a beak instead of a mouth*

Please write your answers neatly and take as much time as you wish with each compound.

1. Eye Plot

2. Raisin Sculptor

3. Butcher Corpse

4. Cake Machete

5. Corpse Surgeon

6. Marzipan Cherry

7. Man Criminal

8. Author Priest

9. Surgery Priest

10. Experiment Eye

11. Accountant Author

12. Architect Composer

13. Surgery Casualty

14. Criminal Composer

15. Cake Marzipan

16. Artery Blood

17. Accountant Office

18. Anglicanism Religion

19. Cancer Disinfectant

20. Author Scalpel

Thanks for your time!

Section 3 - The data

Q1	Meronymous	Identity	Insertion	Context Insertion	Catachrestic	Contextual	N.C	Number of interpretations
S1	0	0	2	0	0	0	0	2
S2	0	0	2	0	0	0	0	2
S3	0	0	1	0	0	0	0	1
S4	0	0	1	0	0	0	0	1
S5	0	0	2	0	0	0	0	2
S6	0	0	0	0	0	0	2	2
S7	0	0	1	0	0	0	1	2
S8	0	0	3	1	0	0	0	4
S9	0	0	1	0	0	0	0	1
S10	0	0	0	0	0	0	1	1
INCA	0	0	0	0	0	0	0	0
S/Avg	0	0	1.3	0.1	0	0	0.4	1.8

Q2	Meronymous	Identity	Insertion	Context Insertion	Catachrestic	Contextual	N.C	Number of interpretations
S1	0	0	1	0	0	0	0	1
S2	0	0	3	0	0	0	0	3
S3	0	0	2	0	0	0	0	2
S4	0	0	1	0	0	0	0	1
S5	0	0	0	2	0	0	0	2
S6	0	0	1	0	0	0	1	2
S7	0	0	1	0	0	1	0	2
S8	0	1	2	0	0	0	0	3
S9	0	0	0	0	0	0	0	0
S10	0	1	1	0	0	0	0	2
INCA	0	0	0	0	0	0	0	0
S/Avg	0	0.2	1.2	0.2	0	0.1	0.1	1.8

Q3	Meronymous	Identity	Insertion	Context Insertion	Catachrestic	Contextual	N.C	Number of interpretations
S1	0	0	0	1	0	0	0	1
S2	1	1	0	0	0	0	1	3
S3	0	0	1	0	0	0	0	1
S4	0	1	0	0	0	0	0	1
S5	0	1	0	0	0	0	0	1
S6	1	0	0	0	0	0	0	1
S7	0	0	1	0	0	0	1	2
S8	0	1	0	0	0	0	0	1
S9	0	0	0	1	0	0	1	2
S10	0	1	0	0	1	0	0	2
INCA	0	1	0	0	1	1	0	3
S/Avg	0.2	0.5	0.2	0.2	0.1	0	0.3	1.5

Q4	Meronymous	Identity	Insertion	Context Insertion	Catachrestic	Contextual	N.C	Number of interpretations
S1	1	0	0	0	0	0	0	1
S2	0	0	3	0	0	0	1	4
S3	0	0	2	0	0	0	0	2
S4	0	0	2	0	0	0	0	2
S5	0	0	1	0	0	0	0	1
S6	0	0	1	0	0	0	1	2
S7	0	0	1	0	0	0	1	2
S8	0	0	1	0	0	0	1	2
S9	0	0	1	0	0	0	0	1

S10	0	0	1	0	0	0	1	2
INCA	0	0	0	0	0	0	0	0
S/Avg	0.1	0	1.3	0	0	0	0.5	1.9

Q5	Meronymous	Identity	Insertion	Context Insertion	Catachrestic	Contextual	N.C	Number of interpretations
S1	0	0	1	1	0	0	0	2
S2	0	0	1	0	0	2	0	3
S3	0	0	1	0	0	0	0	1
S4	0	0	1	0	0	0	0	1
S5	0	0	1	0	0	0	0	1
S6	0	1	1	0	0	0	0	2
S7	0	1	1	0	0	0	1	3
S8	0	0	0	0	0	0	1	1
S9	0	0	1	0	0	0	0	1
S10	0	0	0	0	0	0	3	3
INCA	0	0	2	0	0	3	0	5
S/Avg	0	0.2	0.8	0.1	0	0.2	0.5	1.8

Q6	Meronymous	Identity	Insertion	Context Insertion	Catachrestic	Contextual	N.C	Number of interpretations
S1	1	0	0	0	0	0	0	1
S2	0	0	2	0	0	0	1	3
S3	0	0	2	0	0	0	0	2
S4	0	0	1	0	0	0	0	1
S5	0	0	1	0	0	0	0	1
S6	0	0	0	0	0	0	3	3
S7	0	0	0	1	0	0	0	1
S8	0	0	2	0	0	0	0	2
S9	1	0	0	0	0	0	0	1
S10	0	0	2	0	0	0	0	2
INCA	0	2	1	2	0	0	0	5
S/Avg	0.2	0	1	0.1	0	0	0.4	1.7

Q7	Meronymous	Identity	Insertion	Context Insertion	Catachrestic	Contextual	N.C	Number of interpretations
S1	0	0	1	0	0	0	0	1
S2	0	1	1	0	0	0	0	2
S3	0	0	2	0	0	0	0	2
S4	0	1	0	0	0	0	0	1
S5	0	0	1	0	0	0	0	1
S6	0	1	1	0	0	0	0	2
S7	0	1	1	0	0	0	0	2
S8	0	0	2	0	0	0	0	2
S9	0	0	1	0	0	0	0	1
S10	0	1	0	0	0	0	1	2
INCA	0	0	0	0	0	0	0	0
S/Avg	0	0.5	1	0	0	0	0.1	1.6

Q8	Meronymous	Identity	Insertion	Context Insertion	Catachrestic	Contextual	N.C	Number of interpretations
S1	0	2	0	0	0	0	0	2
S2	0	0	0	1	0	0	0	1
S3	0	1	0	0	0	0	0	1
S4	0	1	0	0	0	0	0	1
S5	0	0	0	0	0	0	0	0
S6	0	1	1	0	0	0	1	3
S7	0	1	2	0	0	0	1	4

S8	0	1	0	0	0	0	1	2
S9	0	1	0	0	0	0	0	1
S10	0	1	1	0	0	0	0	2
INCA	0	3	0	3	0	1	0	7
S/Avg	0	0.9	0.4	0.1	0	0	0.3	1.7

Q9	Meronymous	Identity	Insertion	Context Insertion	Catachrestic	Contextual	N.C	Number of interpretations
S1	0	2	2	0	0	1	0	5
S2	0	1	0	0	1	0	1	3
S3	0	0	1	0	0	0	0	1
S4	0	0	1	0	0	0	0	1
S5	0	0	1	0	0	0	0	1
S6	0	1	0	0	0	0	0	1
S7	0	0	2	0	0	0	0	2
S8	0	0	2	0	0	0	0	2
S9	0	0	1	0	0	0	0	1
S10	0	0	1	0	0	0	1	2
INCA	0	0	0	0	0	0	0	0
S/Avg	0	0.4	1.1	0	0.1	0.1	0.2	1.9

Q10	Meronymous	Identity	Insertion	Context Insertion	Catachrestic	Contextual	N.C	Number of interpretations
S1	0	0	0	1	0	0	0	1
S2	0	0	1	1	1	0	2	5
S3	0	0	0	0	0	0	0	0
S4	0	0	0	2	0	0	0	2
S5	0	0	0	1	0	0	0	1
S6	0	0	0	0	0	0	2	2
S7	0	0	0	0	0	0	1	1
S8	2	0	0	0	0	0	0	2
S9	0	0	0	1	0	0	0	1
S10	1	0	0	0	0	0	0	1
INCA	0	0	0	0	0	0	0	0
S/Avg	0.3	0	0.1	0.6	0.1	0	0.5	1.6

Q11	Meronymous	Identity	Insertion	Context Insertion	Catachrestic	Contextual	N.C	Number of interpretations
S1	0	2	0	1	0	0	0	3
S2	0	4	0	0	0	0	1	5
S3	0	0	0	0	0	0	0	0
S4	0	2	0	0	0	0	0	2
S5	0	1	0	0	0	0	0	1
S6	0	2	1	0	0	0	0	3
S7	0	0	1	0	0	0	1	2
S8	0	0	1	0	0	0	0	1
S9	0	0	1	0	0	0	0	1
S10	0	1	0	0	0	0	1	2
INCA	0	2	4	2	0	0	0	8
S/Avg	0	1.2	0.4	0.1	0	0	0.3	2

Q12	Meronymous	Identity	Insertion	Context Insertion	Catachrestic	Contextual	N.C	Number of interpretations
S1	0	2	0	1	0	0	0	3
S2	0	2	0	1	0	0	1	4
S3	0	1	0	0	0	0	1	2
S4	0	2	0	0	0	0	0	2
S5	0	0	0	0	0	0	0	0

S6	0	2	0	0	1	0	0	3
S7	0	1	0	0	0	0	2	3
S8	0	1	0	0	0	0	2	3
S9	0	1	0	0	0	0	0	1
S10	0	1	0	1	0	0	0	2
INCA	0	2	0	3	0	0	0	5
S/Avg	0	1.3	0	0.3	0.1	0	0.6	2.3

Q13	Meronymous	Identity	Insertion	Context Insertion	Catachrestic	Contextual	N.C	Number of interpretations
S1	0	0	1	0	0	0	0	1
S2	0	0	1	1	0	0	1	3
S3	0	0	1	0	0	0	0	1
S4	0	0	1	0	0	0	0	1
S5	1	0	1	0	0	0	0	2
S6	1	0	0	1	0	0	0	2
S7	1	0	1	0	0	0	0	2
S8	0	0	1	0	0	0	0	1
S9	0	0	1	0	0	0	0	1
S10	0	0	1	0	0	0	1	2
INCA	0	0	0	0	2	5	0	7
S/Avg	0.3	0	0.9	0.2	0	0	0.2	1.6

Q14	Meronymous	Identity	Insertion	Context Insertion	Catachrestic	Contextual	N.C	Number of interpretations
S1	0	2	0	0	0	1	0	3
S2	0	2	1	1	0	0	0	4
S3	0	1	0	0	0	0	0	1
S4	0	2	0	1	0	0	1	4
S5	0	0	1	0	0	0	1	2
S6	0	1	1	0	0	0	0	2
S7	0	1	1	0	0	0	1	3
S8	0	1	1	0	0	0	2	4
S9	0	1	0	0	0	0	1	2
S10	0	1	0	0	0	0	1	2
INCA	0	2	0	2	0	0	0	4
S/Avg	0	1.2	0.5	0.2	0	0.1	0.7	2.7

Q15	Meronymous	Identity	Insertion	Context Insertion	Catachrestic	Contextual	N.C	Number of interpretations
S1	0	0	0	0	0	0	1	1
S2	0	0	2	3	0	0	0	5
S3	1	0	0	0	0	0	1	2
S4	0	0	2	0	0	0	0	2
S5	0	0	1	0	0	0	0	1
S6	1	0	0	0	0	0	1	2
S7	1	0	1	0	0	0	0	2
S8	0	0	1	0	0	0	0	1
S9	0	0	1	0	0	0	0	1
S10	0	0	1	0	0	0	0	1
INCA	0	0	3	0	0	1	0	4
S/Avg	0.3	0	0.9	0.3	0	0	0.3	1.8

Q16	Meronymous	Identity	Insertion	Context Insertion	Catachrestic	Contextual	N.C	Number of interpretations
S1	1	0	0	0	0	0	1	2
S2	1	0	0	1	0	0	0	2
S3	1	0	0	0	0	0	0	1

S4	1	0	0	0	0	0	0	1
S5	1	0	0	0	0	0	0	1
S6	1	0	0	0	0	0	1	2
S7	1	0	0	0	0	0	1	2
S8	1	0	0	0	0	0	0	1
S9	0	0	0	0	0	0	1	1
S10	1	0	0	0	0	0	0	1
INCA	2	0	0	0	0	0	0	2
S/Avg	0.9	0	0	0.1	0	0	0.4	1.4

Q17	Meronymous	Identity	Insertion	Context Insertion	Catachrestic	Contextual	N.C	Number of interpretations
S1	0	0	1	0	0	0	0	1
S2	0	0	2	0	0	0	2	4
S3	0	0	1	0	0	0	0	1
S4	0	0	2	0	0	0	0	2
S5	0	0	1	0	0	0	0	1
S6	0	0	2	0	0	0	0	2
S7	0	0	1	0	1	0	0	2
S8	0	0	1	0	0	0	0	1
S9	0	0	1	0	0	0	0	1
S10	0	0	1	1	0	0	1	3
INCA	1	0	0	0	0	0	0	1
S/Avg	0	0	1.3	0.1	0.1	0	0.3	1.8

Q18 Removed

Q19	Meronymous	Identity	Insertion	Context Insertion	Catachrestic	Contextual	N.C	Number of interpretations
S1	0	0	0	1	0	0	0	1
S2	0	0	2	1	0	0	0	3
S3	0	0	1	0	0	0	0	1
S4	0	0	1	0	0	0	0	1
S5	0	0	0	1	0	0	0	1
S6	0	0	2	0	0	0	0	2
S7	0	0	1	0	0	0	1	2
S8	0	0	2	0	0	0	0	2
S9	0	0	1	0	0	0	1	2
S10	0	0	1	0	0	0	1	2
INCA	0	0	0	1	0	0	0	1
S/Avg	0	0	1.1	0.3	0	0	0.3	1.7

Q20	Meronymous	Identity	Insertion	Context Insertion	Catachrestic	Contextual	N.C	Number of interpretations
S1	0	0	0	0	1	0	0	1
S2	0	0	0	0	1	0	2	3
S3	0	0	1	0	0	0	0	1
S4	0	0	1	0	0	0	0	1
S5	0	0	1	0	0	0	0	1
S6	0	0	2	0	0	0	0	2
S7	0	0	0	0	0	0	0	0
S8	0	0	2	0	0	0	0	2
S9	0	0	0	0	0	0	1	1
S10	0	0	1	0	0	0	1	2
INCA	0	0	0	1	1	0	0	2
S/Avg	0	0	0.8	0	0.2	0	0.4	1.4