Identifying Nocuous Ambiguity in Natural Language Requirements

by Francis J. Chantree

Bachelor of Science (Honours),
The Open University (1997).
Masters in Artificial Intelligence,
University of Edinburgh (1999).

Submitted to the Faculty of Maths and Computing
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computing

THE OPEN UNIVERSITY

29 September 2006

Abstract

This dissertation is an investigation into how ambiguity should be classified for authors and readers of text, and how this process can be automated. Usually, authors and readers disambiguate ambiguity, either consciously or unconsciously. However, disambiguation is not always appropriate. For instance, a linguistic construction may be read differently by different people, with no consensus about which reading is the intended one. This is particularly dangerous if they do not realise that other readings are possible. Misunderstandings may then occur. This is particularly serious in the field of requirements engineering. If requirements are misunderstood, systems may be built incorrectly, and this can prove very costly. Our research uses natural language processing techniques to address ambiguity in requirements. We develop a model of ambiguity, and a method of applying it, which represent a novel approach to the problem described here.

Our model is based on the notion that human perception is the only valid criterion for judging ambiguity. If people perceive very differently how an ambiguity should be read, it will cause misunderstandings. Assigning a preferred reading to it is therefore unwise. In text, such ambiguities should be located and rewritten in a less ambiguous form; others need not be reformulated. We classify the former as *nocuous* and the latter as *innocuous*. We allow the dividing line between these two classifications to be adjustable. We term this the *ambiguity threshold*, and it represents a level of intolerance to ambiguity. A nocuous ambiguity can be an *unacknowledged* or an *acknowledged* ambiguity for a given set of readers. In the former case, they assign disparate readings to the ambiguity, but each is unaware that the others read it differently. In the latter case, they recognise that the ambiguity has more than one reading, but this fact may be unacknowledged by new readers.

We present an automated approach to determine whether ambiguities in text are nocuous or innocuous. We use heuristics to distinguish ambiguities for which there is a strong consensus about how they should be read. These are innocuous ambiguities. The remaining nocuous ambiguities can then be rewritten at a later stage. We find consensus opinions about ambiguities by surveying human perceptions on them. Our heuristics try to predict these perceptions automatically. They utilise various types of linguistic information: generic corpus data, morphology and lexical subcategorisations are the most successful. We use *coordination ambiguity* as the test case for this research. This occurs where the scope of words such as *and* and *or* is unclear.

Our research contributes to both the requirements engineering and the natural language processing literatures. Ambiguity is known to be a serious problem in requirements engineering, but has rarely been dealt with effectively and thoroughly. Our approach is an appropriate solution, and our flexible ambiguity threshold is a particularly useful concept. For instance, high ambiguity intolerance can be implemented when writing requirements for safety-critical systems. Coordination ambiguities are widespread and known to cause misunderstandings, but have received comparatively little attention. Our heuristics show that linguistic data can be used successfully to predict preferred readings of very diverse coordinations. Used in combination, these heuristics demonstrate that nocuous ambiguity can be distinguished from innocuous ambiguity under certain conditions. Employing appropriate ambiguity thresholds, accuracy representing 28% improvement on the baselines can be achieved.

Thesis Supervisor: Anne de Roeck
Title: Professor

Thesis Supervisor: Bashar Nuseibeh
Title: Professor

Thesis Supervisor: Alistair Willis
Title: Dr.

Contents

List of Figures

List of Tables

Acknowledgements

Firstly I wish to acknowledge the help and support of my supervisors: Anne de Roeck, Bashar Nuseibeh and Alistair Willis. Special thanks also go to Adam Kilgarriff, who provided a seed from which this research grew. David Jenkinson helped with some of the statistical analysis, and Marian Petre and Trevor Collins gave invaluable advice about completing a PhD. Chief amongst my colleagues I would like to thank Katerina “e-Kat” Tzanidou, Avik “Babu” Sarkar and Andrea Capiluppi, who have been such good company over the years. Others who deserve special mention are Debra, Geke, Armstrong and Mohammed, not least for having put up with my tapping feet — for which, thanks go to ţambal/cimbalom/tsymbaly players everywhere. Final appreciation goes to Dana Boancă, who gave love and light during the final stages of this research.

Preface

We have published results on distinguishing nocuous from innocuous ambiguities in requirements at a specialist requirements engineering conference (Chantree, Nuseibeh, de Roeck, and Willis 2006). We have also addressed coordination ambiguities from an NLP perspective in a conference paper (Chantree, Kilgarriff, de Roeck, and Willis 2005) and a book chapter (Chantree, Willis, Kilgarriff, and de Roeck 2006). Ideas we developed on ambiguity in natural language generation were presented in an earlier paper (Chantree 2004). This dissertation represents a definitive and unifying account of the research presented in these publications. Additionally, we have presented software we developed to chunk text in a manner suitable for analysing our test case ambiguities. This software was demonstrated at the Seventh International Conference on Text, Speech and Dialogue (TSD) in Brno, Czech Republic, in September 2004 (Sojka et al. 2004) [1].

[1] http://www.tsdconference.org/tsd2004/tsd_prog.html.

Chapter 1

Introduction

Processing ambiguous sentences and expressions is one of the most challenging tasks in natural language processing (NLP). Much NLP research has therefore been directed towards disambiguation, for example (Agarwal and Boggess 1992; Okumura and Muraki 1994; Resnik 1999). This is usually achieved by determining the *preferred* reading of any ambiguity, and many techniques have been proposed to achieve this. However, such a reading may not be preferred by some readers, and indeed it may not be the author's *intended* reading. Some ambiguities are particularly susceptible to this, and misunderstandings will result. Few NLP researchers have addressed this problem and how it might be solved. This issue is the focus of this thesis.

Our approach to ambiguity takes into account the vagaries of human perception that make disambiguation unwise for some linguistic constructions. We introduce a novel model of ambiguity to account for this. Instead of disambiguating ambiguities, we determine instead whether they are likely to lead to misunderstandings. We call ambiguities *nocuous* if this is the case. The opposite situation is when linguistic constructions are ambiguous in theory but in fact are generally read in the same way. We term these ambiguities *innocuous.* Misunderstandings between readers or between an author and readers are the result of nocuous ambiguity: in either case there is no clearly preferred reading. On the other hand, an innocuous ambiguity does not result in misunderstandings because it is easily disambiguated, and there is generally only one reading made of it: the preferred reading is therefore generally the same as the intended one. The aim of this research is to determine whether any given ambiguity is nocuous or innocuous.

In our model of ambiguity, an ambiguity can be nocuous for two reasons. Firstly, it is nocuous if it is experienced by readers as having more than one reading. We refer to this situation as *acknowledged* ambiguity. Secondly, it is nocuous if it is read differently by different people, but they do not recognise that other readings are possible. We refer to this situation as *unacknowledged* ambiguity. Unacknowledged ambiguity is the most dangerous situation as it is more liable to lead to unresolved misunderstandings. Acknowledged ambiguity may not be a problem as misunderstandings may get resolved because people are aware of their ambiguity. However, it is potentially dangerous because it can also lead to unacknowledged ambiguity. This happens because somebody may still not recognise that an ambiguity generally acknowledged to have more than one reading has this property. Acknowledged and unacknowledged ambiguity can only ever be determined for a fixed group of readers. So, an acknowledged ambiguity for any such group may be unacknowledged for a new reader.

We distinguish nocuous from innocuous ambiguities by first obtaining a consensus of human judgements about them. This is fundamental to our model of ambiguity: ambiguity is only realised via human perception (Wasow, Perfors, and Beaver 2003), and so it should be judged accordingly. We therefore survey the opinions of a group of human judges in order to capture the variety of possible perceptions. We use a sufficient number of judges to give a reliable indication of this variety.

There is a question concerning how many people must recognise an ambiguity as being ambiguous for us to consider it an acknowledged ambiguity. Similarly, how many people must assign different readings to an ambiguity for it to be considered, on balance, an unacknowledged ambiguity? These issues will affect whether an ambiguity is nocuous or innocuous. We use the concept of an *ambiguity threshold* to establish where the dividing line between nocuous and innocuous ambiguity lies. This threshold represents a level of *intolerance* to ambiguity.

We then try to automatically predict whether ambiguities are nocuous or innocuous. (The nocuous ambiguities, once they have been detected and flagged up, can be dealt with at a later stage. Depending on the application, this may involve rewriting them in a less ambiguous form. We do not deal with this later process in this research.) The prediction is achieved using heuristics which we have developed and found to be effective. These heuristics use various types of linguistic information obtained from a corpus and from surface features of the text. We experiment with optimising these heuristics, and we combine them to obtain increased predictive power.

We use three methods of implementing our heuristics, each one using ambiguity thresholds in a different way. The method we term the Weighted Method uses a fixed ambiguity threshold that represents the idea that unacknowledged ambiguity is of more immediate concern than acknowledged ambiguity. The Flexible Method considers unacknowledged and acknowledged ambiguity to be equally indicative of nocuous ambiguity, but it allows for a flexible intolerance towards them. Thirdly, we employ the Unacknowledged Method to analyse solely unacknowledged ambiguity. This third method also employs a flexible intolerance to ambiguity. We use the Weighted Method to ascertain our heuristics' optimal effectiveness and to test the hypotheses on which they are founded. We use the Flexible Method to test our heuristics' ability at distinguishing nocuous from innocuous ambiguity. This method demonstrates the usefulness of a flexible intolerance to ambiguity. We use the Unacknowledged Method to examine whether unacknowledged ambiguity — the most immediately concerning type — can be characterised and predicted.

We use text from the field of *requirements engineering* (RE). This is the branch of software engineering concerned with specifying systems. Loosely speaking, each specification is termed a *requirement*. It is a very suitable domain, as requirements must be as free as possible from ambiguity (Gause and Weinberg 1989). A misunderstanding of something written in a requirement can result in very costly mistakes: a system may then be built incorrectly, incurring large and unrecoverable development costs (Boehm 1981). This may happen due to a nocuous ambiguity in the requirement. The problem of ambiguity is known to be potentially serious in RE (Berry and Kamsties 2005). However, there is a competing argument that sweeping disambiguation is often unnecessary. Many ambiguities are easily disambiguated by humans, and hence the intrusion of an automated disambiguation system can be unwelcome (Kamsties, Berry, and Paech 2001). Our ambiguity threshold addresses this issue. If the consequences of ambiguity are likely to be serious, as in requirements for a safety-critical system, then intolerance of ambiguity will be high. Alternatively, it may be considered important to avoid the effort required to check lots of ambiguities, many of which will not lead to misunderstandings. Flagging up unnecessary numbers of ambiguities can be avoided by implementing a low intolerance of ambiguity.

We test our approach on coordination ambiguity. This is a structural (i.e. syntactic) ambiguity, that is both widespread and potentially dangerous. Resnik (1999) states that coordinations are a “pernicious source of structural ambiguity in English”. However, coordination ambiguity has received little attention in the NLP literature compared to similar types such as prepositional phrase (PP) attachment ambiguity. We attempt to analyse coordinations of a wide range of word and phrase types. Our research therefore makes a useful contribution to the literature in this regard. We focus upon a narrowly-defined manifestation of coordination ambiguity. However, our language model, and our method of applying it, are generally applicable to ambiguity.

Our individual heuristics achieve variable success in distinguishing nocuous from innocuous ambiguities. In combination they achieve good performance when some appropriate ambiguity thresholds are implemented. It is impossible in practice for computers to model ambiguity exhaustively, largely because context plays such a large part when humans disambiguate text (Bar-Hillel 1960). Any approach such as ours therefore offers assistance as opposed to a total solution to the problem. In the RE community this is deemed to be the most appropriate approach (Ryan 1993), and also to be both feasible and useful (Gervasi and Nuseibeh 2000; Gervasi and Nuseibeh 2002). Ultimately, we envisage our approach being implemented as a *wizard*, operating in conjunction with a word processor. This would determine whether ambiguities of many types are nocuous or innocuous.

1.1 Introduction to Ambiguity

Here we introduce the concept of ambiguity, and the definition of it that we use for this research. To further explain and refine the scope of this definition, we also discuss concepts which are similar to ambiguity. Where appropriate, we explain why we do not consider these in our analysis, or why they may overlap with our understanding of ambiguity.

1.1.1 What is Ambiguity?

Dictionary definitions of *ambiguous* can be as various as “doubtful”, “undetermined”, “indistinct”, “wavering”, “equivocal”, and “admitting of more than one meaning” (Simpson et al. 1988). The definition generally adopted in linguistics and artificial intelligence, and the one on which we base our own, is the last of these.

Many variations on this definition can be used, generally involving differing consideration of context. For instance, in the field of RE, ambiguity can be defined as having "multiple interpretations despite the reader's knowledge of the RE context" (Kamsties, Berry, and Paech 2001). Many other definitions exist for the RE domain alone (Kamsties 2001). Sometimes in RE the term "ambiguity" is assigned a specialised meaning whereby a requirements specification is ambiguous if it is non-deterministic[1]. This implies that the computer program specified by the requirements can execute in more than one way and still be consistent with the requirements specification.

Such is the "ambiguity of ambiguity", that Gillon (1990) concludes that all attempts to formulate precise definitions and tests of ambiguity have been problematic. Our task is simplified somewhat as we ignore most considerations of context. Context has infinitely large scope and many facets, and many of these facets are either not computationally tractable or not knowable from the information at hand. Also, we wish to develop a general solution that is applicable for requirements describing many types of system. Ultimately, we wish it to be usable for application domains other than requirements engineering. It is therefore appropriate that our consideration of context is limited.

We wish to analyse how differently humans perceive ambiguity. To obtain a clear picture of this, we use examples of ambiguity with equal numbers of possible interpretations. We therefore require that they have a *discrete* number of these interpretations. This definition distinguishes ambiguity from *vagueness*, for instance, where a continuum of interpretations exist. (Concepts such as vagueness which are similar but not identical to ambiguity are discussed in the next subsection). Our definition is similar to Kamsties' (2001) notion of *genuine ambiguity.* We talk of *interpretation* rather than *meaning*, as

[1] Connie Heitmeyer: personal communication

the former requires an interpreter, which the latter does not. This is important for us, as requirements are a communication between two or more parties. We are concerned only with written text as this is the medium in which most requirements are formulated. Also, we are interested in passages of text rather than individual words or whole documents, as this is the unit which most usually corresponds to individual requirements. Therefore we formulate the following definition of ambiguity for use in our research:

A passage of text is ambiguous if it has two or more discrete interpretations

Ambiguities are most often classed as either *lexical*, *semantic*, *syntactic* (or *structural*) or *pragmatic*. Lexical ambiguity occurs when a word has several meanings. Semantic ambiguity can be simply lexical ambiguity. Or, it can refer to the situation where combinations of words can have different meanings. Structural ambiguity refers to the situation where a sequence of words can be grammatically structured in different ways. This will generally yield different meanings. Pragmatic ambiguity occurs when the context of an expression has an influence, allowing several alternative meanings of the expression. Our definition of ambiguity, and our ambiguity model, could be applied to ambiguities of all these types. However, we look only at one particular type of ambiguity in order to focus our research.

1.1.2 Concepts Akin to Ambiguity

Here we discuss concepts that are akin to ambiguity, but are not exactly the same as it. We do not seek to analyse these other concepts specifically in this research. However, some of them overlap with our concept of ambiguity, and examples of them will inevitably be captured together with the ambiguities we analyse. Others can be seen as similar to ambiguity in that they lead to similar consequences. *Inconsistency* and *incompleteness* are of particular interest to requirements engineers, and are often discussed in the literature in the same breath as ambiguity (Rupp 2000; Shull, Rus, and Basili 2000). We approach the others – *generality*, *vagueness* and *clarity* – from a linguistics standpoint. Lastly, we introduce the concepts of *indefiniteness*, *indeterminacy* and *correctness*. These subsume, or overlap with, the concepts already mentioned here, so we do not discuss them in depth.

Inconsistency (or Conflict)

Statements "are inconsistent if they state facts that cannot both be true, or if they express actions that cannot be carried out at the same time" (Kamsties 2001). Zwicky and Sadock (1975) contend that inconsistent statements count as ambiguities. As with ambiguity, *undetected inconsistency* is recognised as a particularly dangerous source of costly errors (Gervasi and Zowghi 2005). Nuseibeh, Easterbrook, and Russo (2001) state that problems are caused not by inconsistency *per se*, but by undetected inconsistency. These views concur with our own regarding unacknowledged ambiguity. However, we are concerned with text that can be read in more than one way. Inconsistency is revealed when a *single* reading causes conflicting truth conditions, so we do not look explicitly for it.

Incompleteness

Incompleteness refers to the omission of necessary information in a communication. This includes failure to provide further information needed to determine or define information already present in the communication (Zowghi and Gervasi 2002). Kamsties, Berry, and Paech (2001) make the distinction that "Incompletenesses and ambiguities can be distinguished by the type of required correction activity. The former require adding information, while the latter just require rephrasing the present information". However, addition of information can disambiguate ambiguities as well as incompletenesses. We believe therefore that there is an overlap, and so we may well capture many of the latter in our quest to capture the former.

It can be argued, at least from a linguistics standpoint, that statements are never complete. There is always a cut-off point regarding how much specifying information is given. Boehm (1984) realises the difficulty this poses in requirements. However, rather than having to consider "all possible worlds", requirements engineers can usually relate their texts to real-world denotations.

Generality

"An expression is general if and only if the expression's connotation is a genus of more than one species" (Gillon 1990). For instance, *cousin* is general with respect to gender: it can refer to both male and female cousins. The word *kid* is ambiguous between two senses, *baby goat* and *baby human*, whereas *horse* is not ambiguous between *mare* and *colt* but rather has a sense that subsumes both. The latter instance is another example of generality. General expressions can always be made more precise, if necessary, by adding supplementary information.

Looking from an RE perspective, Kamsties (2001) states that generality "usually occurs as a pragmatic ambiguity". This means that the context of the generality, as opposed to the general expression itself, is the factor that determines how it is read. Kamsties also claims that generality is "usually used deliberately" and is "often acceptable for a while". This will tend to mean that generality is not a fruitful source of unacknowledged, and therefore nocuous, ambiguity. These factors mitigate against the use of generalities as the test case for our research.

Vagueness

Some researchers, for example Allen (1995), consider *vagueness* to be no different from *generality*. However, it may be distinguished in the following way: "In the case of a vague sentence, one is uncertain of its truth relative to a specified state of affairs. No further knowledge of the state of affairs relieves the uncertainty" (Gillon 2003). In philosophy it is argued that vague statements admit borderline cases (Bach 1998). For instance, the word *tall* is vague because, for example, it cannot be decided whether a man of 1.80m is clearly tall or clearly not tall (Berry, Kamsties, and Krieger 2003). This is in contrast to generality. For instance, regardless of which other factors are left unspecified, it is always clear whether or not somebody is your cousin. Virtually all statements contain vagueness, as they don't give all the possible determining factors of that which they describe. It can be argued that some expressions, for example "fast" and "user-friendly", are vague by nature and cannot be made precise (Kamsties 2001).

As with generality, Kamsties (2001) claims that vagueness is usually used deliberately and is often acceptable for a while. It will also therefore not be so fruitful a source of unacknowledged, and therefore nocuous, ambiguity. Also, vagueness is acknowledged to be more unavoidable and more a fact of life than ambiguity. For these reasons, we believe that it should be tackled by a different type of research project to the one presented here.

Lack of Clarity

Clarity, as described by Walton (1996), is based on Grice's maxims for the successful conduct of a conversation (Grice 1975). These maxims insist upon avoidance of hindrances to understanding such as obscurity, irrelevance and verbosity. Clarity is a pragmatic concept, generally dependent on context and on a wider consideration of text than we attempt. We describe it here as it conveniently brings consideration of the *recipients* of communication, as well as the *originators*, into focus. Clarity, and therefore also the lack of it, are closely related to ambiguity. However, they emphasise the idea that misleading the recipients is key, and that ambiguity is only a problem when this occurs. This accords with our use of human perception as the criterion for judging ambiguity, and our view that unacknowledged ambiguity is the most dangerous source of misunderstandings. However, it does not capture the danger of acknowledged ambiguity: text which has not so far misled the recipients may also be dangerous. This is because if it has more than one possible interpretation, it may mislead future recipients.

Other Concepts

Other terms exist to describe concepts similar to ambiguity. For instance, Pinkal (1995) uses the notion of *indefiniteness* (Pinkal 1995), which subsumes ambiguity and vagueness but not generality. Poesio states that "A sentence is 'semantically indefinite' if and only if in certain situations, despite sufficient knowledge of the relevant facts, neither *true* nor *false* can be clearly assigned as its truth value" (Poesio 1996). This appears to align with Gillon's definition of vagueness, given above. Whichever definition of indefiniteness is preferred, this concept adds no information not contained in the concepts we have already defined. Therefore we do not discuss it further.

Indeterminacy is a term sometimes used by both linguists and those using linguistics in practical applications. However, it can mean different things. Gillon (2003) states that "the noun 'nail' is indeterminate with respect to the longness or shortness of the things in its denotation". He claims that indeterminacy is a distinct concept as "every common noun is indeterminate but not every common noun is general" or, indeed, ambiguous (Gillon 1990). However, this sounds like vagueness, according to our discussion of it above. Pinkal (1995) would seem to confirm this. Alternatively, indeterminacy is sometimes considered, for example by requirements engineers such as Kamsties (2001), to be the same as generality. Because of these confusions, we prefer not to consider indeterminacy as a separate concept.

Correctness is a term sometimes used in RE, though it is not always clearly defined. Zowghi and Gervasi (2002) state that it can be the "combination of consistency and completeness", or it can refer to the "satisfaction of certain business goals". In the former usage it subsumes concepts that we have already defined, in the latter it refers to aspects out of the scope of our research. Therefore we do not discuss it further.

1.2 The Application Domain

In this section we introduce the domain in which we implement our approach to ambiguity, and from which we take our examples of ambiguity. We explain what requirements engineering is, the problems that ambiguity causes within it, and what our response is to these problems.

1.2.1 Definitions

RE, or *software systems requirements engineering* as it is sometimes called, can be defined as the process of discovering the purpose for which a software system is intended (Nuseibeh and Easterbrook 2000). This involves "identifying stakeholders[2] and their needs, documenting these in a form that is amenable to analysis, communication, and subsequent implementation" (Nuseibeh and Easterbrook 2000). As we are analysing text, we are concerned with the documents that carry this information, and refer to these as *requirements documents*[3]. In this thesis we make no use of common distinctions such as that between documents of *functional* and *non-functional* requirements. This because we are concerned with short passages of text, and not with the wider significance of the documents that contain them.

[2] *Stakeholders* are all the people who have something to gain or lose from the system.

[3] Other RE practitioners, e.g. (Berry, Kamsties, and Krieger 2003), refer to such documents as *(software) requirements specifications.* We prefer to follow those, including Jackson (1995), who reserves the word *specifications* for other usages.

1.2.2 Ambiguity in Requirements

As ambiguity is endemic in natural language, it is known to be a considerable problem for all requirements written in natural language (Berry, Kamsties, and Krieger 2003; Goldin and Berry 1997). Gause and Weinberg (1989) recognise the crucial position that ambiguity has in requirements engineering, and Berry, Kamsties, and Krieger (2003) suggest that unintended ambiguity is the "Achilles' heel of software requirements specifications". If stakeholders interpret a requirement in different ways, this can result in an incorrect implementation and severe costs may be incurred. Ambiguity can be more intractable than other defects, such as incompleteness, and research has shown that it can more frequently result in misunderstandings (Kamsties 2001). Unacknowledged ambiguity, the situation that we hope to address by locating nocuous ambiguity, is the same as *unrecognised disambiguation*. This is one of Gause's five most important sources of requirements failure (Gause 2000), as reported by Berry, Kamsties, and Krieger (2003).

The problems inherent in natural language have led Hanks, Knight, and Strunk (2001) to state that "the way in which humans innately use language is not conducive to effective communication between domain experts and software engineers". Several strategies have therefore been developed to help requirements engineers avoid ambiguity by not using everyday natural language. However, Gervasi (2000) states that natural language is the only language that one can confidently assume is shared among everyone involved in the software development process. Recently Berry, Kamsties, and Krieger (2003) could still observe that the great majority of documents available for requirements analysis were written in natural language. Ambiguity is a problem because requirements are usually not written by those who build and implement the systems they specify. As a result of *outsourcing*, they are increasingly written by employees from different cultural backgrounds. It is known to cause problems in RE when author and reader have differing levels of skill in the language being used (Berry, Kamsties, and Krieger 2003). Additionally, as a result of *offshoring*, employees may not work in the same countries as the other stakeholders. Organisational differences add to the possibilities of misunderstanding ambiguity, making the possibility of misunderstanding requirements even more likely.

1.2.3 Motivation for a Solution

Our work is driven by the desire to locate ambiguities in requirements that will lead to misunderstandings, and so potentially to incorrect implementations. “Ambiguity is characteristic of poor quality requirements, and poor quality requirements are characteristic of challenged projects” (Boyd, Zowghi, and Farroukh 2005). Carrying this out during requirements analysis is a relatively cheap solution to the problem, as the cost of fixing errors at later stages of a system’s development process can be orders of magnitude higher (Boehm 1981). However, locating errors is nontrivial, and even requirements that have been checked many times can still contain defects (Gervasi and Nuseibeh 2000). We therefore aim to offer a technique that assists requirements engineers with this process. We concur with the general opinion in the RE community that a partial solution, such as ours, is more appropriate than one claiming to have full understanding of the ambiguity problem (Ryan 1993). och Dag et al. (2005) cite recent industrial experience that motivates the need for automated support for requirements management.

Kamsties, Berry, and Paech (2001) have considered the fact that not all ambiguities in RE documents need be disambiguated. Also, Mich and Garigliano (2000) have suggested using an ambiguity threshold to distinguish these from more dangerous ambiguities. However, none of these researchers develop further the idea of discriminating between these types of ambiguity. Kamsties, Berry, and Paech do not formulate the idea of an adjustable intolerance to ambiguity. Mich and Garigliano do not base their idea on the human perceptions which, we believe, are necessary to analysis of ambiguity. They provide no conclusive evidence that their way of classifying ambiguities is effective. We are not aware of any other RE researchers who have tried empirically locating ambiguities that might cause misunderstandings.

1.3 Introduction to Coordination Ambiguity

The type of ambiguity we use as the test case for all the research in this thesis is coordination ambiguity. Because coordinations are known to be dangerous (Resnik 1999), nocuous ambiguity should be sufficiently in evidence. We are only interested in coordination in English, and make no comparisons with research based on other languages.

Coordination occurs when two or more linguistic units of equal rank or importance are *coordinated*, i.e. are linked or yoked together (Davidson 1996). In English, many different types of linguistic unit can be coordinated: words, phrases, clauses, sentences, etc. This is frequently achieved using *coordinating conjunctions*, such as *and* and *or*. *Coordination ambiguity* occurs when the scope of the coordination is in doubt. This is the scope of the coordinating conjunction, when one is used. The doubt can arise when it is unclear if a modifying word or phrase modifies one of the linguistic units being coordinated or both of them. This is demonstrated by the following phrase:

> *Assumptions and dependencies that are of importance*

It is unclear whether the clause *that are of importance* modifies (in other words *attaches to*) *assumptions and dependencies* or only *dependencies*. We concentrate solely on configurations of this type, involving a single conjunction and a single modifying word or phrase whose attachment is ambiguous. We evolve *test case criteria* to ensure that the examples of coordination ambiguity we use are of this type.

1.4 Methodology

The hypothesis we wish to test in this thesis is whether ambiguities that cause misunderstandings can be distinguished from those that do not. We test firstly whether this can be witnessed in the perceptions that humans have of ambiguities. Secondly we test whether such perceptions can be predicted automatically by the use of heuristics. We discuss below how we implement these two ways of validating our hypothesis.

1.4.1 Validation Using a New Model of Ambiguity

The first way in which we validate our hypothesis involves postulating a novel Model of Ambiguity. This model accounts for our views about how ambiguity comes into being, and how it can lead to misunderstandings. We take a pragmatic consideration of ambiguity. Because we are concerned with misunderstandings that result from ambiguity, we wish to locate ambiguities that produce them. These are what we term nocuous ambiguities: they are given more than one reading. People can either acknowledge that more than one reading is possible, or they can each assign a reading and not realise that different readings are being made. Alternatively, an ambiguity can always be read in the same way: it is an innocuous ambiguity. Although such an ambiguity may have a structure (i.e. syntax) that permits multiple readings, only one is actually made. Our Model of Ambiguity distinguishes between the *structural* and *interpretation* (i.e. *reading*) aspects. The latter aspect is solely determined by human perception. We wish to validate whether human perceptions do indeed result in nocuous and innocuous ambiguity.

We determine whether this is the case by creating a database of ambiguities and associated human judgements. Firstly we collect a set of examples of text containing ambiguities having the same number of possible readings resulting from their structure. (These are all coordination ambiguities and are taken from a corpus of requirements.)

We then ask a set of human judges to give their interpretations of each ambiguity. We infer from these whether each ambiguity is nocuous or innocuous, applying an ambiguity threshold to determine this. We consider that our Model of Ambiguity is validated if significant quantities of both types of classification are witnessed. Using different ambiguity thresholds results in differing quantities of ambiguities considered nocuous and innocuous. Validation of our Model of Ambiguity will show that ambiguities leading to misunderstandings can be distinguished from those that will not. From an RE perspective, it will also show that we are right to think that the former are a potential problem in requirements.

1.4.2 Validating Using Heuristics

The second way in which we validate our hypothesis involves automatically predicting human perceptions about ambiguity. We use the same dataset of ambiguities as we used for the first type of validation. We seek to categorise the same ambiguities as nocuous and innocuous as the human judges did. Proving that this can be achieved automatically will show the validity of our heuristics.

All our heuristics attempt to predict whether an ambiguity is likely to have a single reading. If they predict this strongly enough, they indicate that the ambiguity is innocuous. It is not the same single reading that all the heuristics predict: we look for heuristics that predict the different readings afforded by the syntax of the examples of ambiguity in our dataset.

We evaluate heuristics of many different types. Some use external information in the form of statistical data from a generic corpus. Others use information obtainable from the surface features of the text containing the ambiguities. Each heuristic is developed from a hypothesis about a linguistic phenomenon that might signify a single reading. To evaluate our individual heuristics, we train them to predict a single reading (and therefore innocuous ambiguity) with high accuracy. Their relative efficacies can then be compared. This process also tests the validity of their underlying hypotheses.

Next, we combine the heuristics. We hope that they will have much greater powers of prediction when used in combination. This is partly reliant on them having complementary coverage. In other words, we hope that they predict different innocuous ambiguities. We use two methods of combination: simple disjunction and logistic regression. We use the latter to determine accuracy at distinguishing innocuous from nocuous ambiguities, as opposed to just predicting the former. During this process, we vary the ambiguity threshold. This shows us the efficacy of the combined heuristics when there are different proportions of nocuous and innocuous ambiguities to predict. Our hypothesis is validated if our heuristics can distinguish innocuous from nocuous ambiguities effectively. It will show that this process can be automated. From an RE perspective, it will also show that a useful tool can be developed to assist authors with determining which ambiguities in their texts need to be addressed.

1.5 Outline of this Thesis

In Chapter 2 of this thesis we discuss previous work that either partially concurs with our approach to ambiguity or demonstrates how the problem has traditionally been tackled. This work provides the starting point for our own research, and demonstrates that we are filling a gap in the literature. In Chapter 3 we outline our model of ambiguity, which we believe represents a novel and appropriate approach to the problems presented by ambiguity. In Chapter 4 we present a detailed presentation of the ambiguity that we use as our test case, and discussion of the ways in which it can become nocuous or innocuous. Chapter 5 explains the process whereby we implement our model of ambiguity using coordination ambiguities taken from a corpus of requirements. This involves explaining how we obtain human judgements about these ambiguities, and then introducing the heuristics that we use to predict these judgements automatically. In Chapter 6 we present the results of our empirical study. We use a variety of statistical methods to prove the efficacy of our heuristics and our key idea that nocuous ambiguities can be distinguished from innocuous ones. Our conclusions are presented in Chapter 7; these are followed by ideas for future work in Chapter 8.

1.6 Summary

In this chapter we have summarised this thesis in the form of an extended introduction. We have then discussed the concept of ambiguity, and how it is different from and similar to other allied concepts. We have introduced the application domain, in which we perform all our empirical investigations. Then we have introduced the type of ambiguity we use as the test case for all the research presented here. Also, we have described the methodology we use to validate the hypotheses which this thesis attempts to prove. Lastly, we have outlined the structure of this thesis.

Chapter 2

Previous Approaches to Ambiguity

Most researchers addressing the problem of ambiguity in text have sought to *disambiguate*. This is a big field, so we survey only the research focusing on the same type of ambiguity that we use as our test case. We then survey the smaller quantity of NLP research where the benefits of *preserving* ambiguity are considered. This is followed by a discussion of the ways in which ambiguity has traditionally been addressed in RE.

Then we introduce suggestions other researchers have made about classifying ambiguity according to how dangerous it is. Some of this research has provided us with the motivation for our nocuous/innocuous distinction. Other research, carried out in parallel with our own, has arrived at ideas similar to ours, but has not investigated them empirically.

2.1 Disambiguation

We discuss here the research of NLP researchers who have sought to disambiguate coordination ambiguity. (A full analysis of coordination ambiguity, and our focus on a narrowly-defined manifestation of it, is made in Chapter 4.) In English, many types of linguistic unit can be coordinated using the same coordinating conjunctions (Okumura and Muraki 1994). The literature[1] reflects this diversity of possibilities and the inherent ambiguities. We firstly discuss research that attempts disambiguation of a wide range of these possibilities. We then focus on research disambiguating only coordinations of nouns, as these have attracted the most attention.

Note that none of the research discussed here seeks to distinguish ambiguities in the way that ours does. However, the results achieved show how amenable coordination ambiguity is to computational analysis. The discussion also serves to highlight the various ways in which coordinations can cause ambiguity, and thereby misunderstandings.

2.1.1 Disambiguation of Wide Range of Coordinations

Coordination in English can be of words of almost any part of speech, and occur at any level of syntactic structure (Okumura and Muraki 1994). An analysis of all such possibilities promises maximal practical application for real-life texts. However, only a few research projects have attempted to analyse a wide range of coordination structures.

Agarwal and Boggess 1992

Agarwal and Boggess (1992) present an algorithm that attempts to identify the scope of coordinating conjunctions in running text. They thereby hope to disambiguate the coordinations in the text. For instance, they attempt to find which phrases are coordinated by each *and* in:

[1] We look only at research on coordination in English. Realisation of coordination varies from language to language, and research on other languages would not be comparable to ours. For instance, Japanese requires the matching of coordinating conjunction to syntactic level, giving a lower potential for ambiguity (Okumura and Muraki 1994). Related work we do not consider includes Kurohashi and Nagao's (1992) analysis of Japanese conjunctive structures, and Park and Cho's (2000) disambiguation of coordinations in Korean.

> *The mites live on the surface of the skin of the ear and canal, and feed by piercing the skin and sucking lymph, with resultant irritation, inflammation, exudation, and crust formation.*

Though there are many syntactic possibilities, the phrases coordinated by the third *and* are probably *piercing the skin* and *sucking lymph*. The method employed to determine this matches parts of speech and case labels of the candidate phrases' head words. Coordinating conjunctions and head words are located, and prepositional phrases are attached, using customised software. This software semi-parses a sentence and pushes all the chunks of words that it creates onto a stack. When a coordinating conjunction is found, candidate pre-conjunction phrases are popped off the stack until a match with the post-conjunction phrase is found.

Agarwal and Boggess test their technique on a tagged and parsed chapter of the Merck Veterinary Manual. They achieve an accuracy of 81.6% on this task. Their method is a straightforward and potentially useful way of matching candidate coordinated phrases. They have also shown that pre-processing software custom-built for coordination disambiguation can be effective and have wide coverage. However, by their own admission, their technique does not extend to coordination ambiguity arising from modifier attachment.

Okumura and Muraki 1994

Okumura and Muraki (1994) develop a model for analysing coordinations in English. They use this to disambiguate coordinations as part of an English-Japanese machine translation system. They claim to cover all types of coordinating conjunction and coordinations of many types of linguistic units. Their approach is to use the *parallelism* found in coordinations in English. They identify three levels of syntactic pattern in order to do this: phrase/clause, word and morphology. Let us consider the phrase:

Inspect the cockpit indicators and levers

Cockpit will be judged to apply to the phrase *indicators and levers.* This is because the latter phrase is judged to be highly syntactically parallel. Both *indicators* and *levers* are concrete nouns and include the same suffix. They are also phrases containing sequences of identical word types — in this case, trivially, a noun matching with a noun.

Because parallelism is exhibited in many ways, coordinations of many different types of words and phrases can be covered. Okumura and Muraki report an accuracy of 75% at predicting conjunction scopes. They appear to use a dataset of 15,000 conjunctive sentences as their test data. No baselines are given, so it is not clear how much of an benefit their technique is. Application of the technique also increases processing efficiency considerably. However, ambiguity arising from modifier attachment will not be adequately addressed. This is because the reading that preserves the parallelism of phrase length will be preferred. In the example above, for instance, *levers* will always be judged to be modified by *cockpit.* In many other examples this reading will be clearly false.

2.1.2 Disambiguation of Coordinations of Nouns

Nouns (and noun phrases) are the most commonly coordinated linguistic units. Also, there is more information available, for example in WordNet[2], on the behaviour of nouns than for other types of word. Focusing on noun coordinations therefore gives the possibility of more satisfactory results, and much research has concentrated on these.

Goldberg 1999

Goldberg (1999) uses unsupervised learning to determine the attachment of noun phrases in ambiguous coordinations of the form *noun1 prep noun2 coord noun3.* Let us consider the phrase:

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

salad of lettuce and tomatoes

It is unclear whether the *tomatoes* are included in the *salad.* Goldberg considers examples like this to be syntactically analogous to the classic PP-attachment dilemma:

I saw the man with the telescope

Tomatoes corresponds to *with the telescope*, and it can attach to *salad of lettuce* (*saw*) or to *lettuce* (*the man*). She accordingly adapts a PP-attachment disambiguation method, that of Ratnaparkhi (1998). She locates unambiguous occurrences of *noun1 coord noun3* and *noun2 coord noun3* in a large unannotated corpus — the Wall Street Journal. (A chunker[3] is used to locate the nouns that are head words of the coordinated phrases.) She then uses these, and other probabilities, to predict preferred interpretations of her test case coordination ambiguities.

Her system predicts with an accuracy of 72% the annotated attachments of a dataset drawn from the Wall Street Journal. The baseline is 64%. She suspects that these results are adversely affected by ineffectiveness of her chunking heuristics. Her PP-attachment re-implementation is a solution that might extend to attachments of words other than nouns. However, it does not make use of information, such as any form of parallelism or word similarity, that is useful specifically for coordination disambiguation. Goldberg's use of coordination probability in a corpus is of interest to us as a simple prediction metric. Also, using a chunker to simplify text and extract head words could be developed into a more effective lightweight pre-processing solution.

[3] Chunkers group words of running text into coarse-grained units such as noun phrases and verb phrases. This can be done to simplify parsing or for NLP tasks not requiring full parsing. Chunking is discussed in depth in Section 5.2.3.

Resnik 1999

Resnik (1999) investigates the role of number agreement, semantic similarity and noun-noun compounding frequency in disambiguating noun coordinations. He looks specifically at the form *noun1 and noun2 noun3*, which can be exemplified by the sentence:

bank and warehouse guard

This is similar to our test case, introduced in Section 1.3, in that it comprises one coordination and one attachment decision. *Noun3* attaches either to both *noun1* and *noun2*, or just to *noun2*. Resnik uses three heuristics to predict the preferred attachment. Firstly, he hypothesises that in the former scenario *noun1* and *noun2* are likely to have equal number; in the latter scenario *noun1* and *noun3* are likely to have equal number. Secondly, using WordNet, he determines whether *noun1* and *noun2* have higher semantic similarity than *noun1* and *noun3*. He hypothesises that this will indicate the former scenario; the reverse will indicate the latter scenario. Thirdly, he calculates the *selectional association* (Resnik 1996) of *noun1* and *noun3* to indicate whether they can form a compound. This metric utilises co-occurrence frequencies from a corpus and semantic class membership from WordNet. Using experimentally obtained thresholds, high selectional association indicates the former scenario and low selectional association indicates the latter scenario.

Resnik combines these heuristics in various ways. He also includes strategies of defaulting to the most likely interpretation, backing off, and choosing the majority verdict of the heuristics. He achieves accuracies of between 65% and 82% for different combinations of heuristics and strategies. This performance is on a hand-disambiguated dataset drawn from the Wall Street Journal. Resnik's work presents several ideas of interest to us: his test case ambiguity is of a similar structure to ours, and he develops useful heuristics for predicting preferred interpretations of it. However, it is unclear if success on the narrowly-defined example of noun coordination transfers to other types of coordination.

Nakov and Hearst 2005

Nakov and Hearst (2005) try to disambiguate compound noun coordinations of the form *noun1 coord noun2 noun3*. They consider only *and* and *or* as the coordinating words. Like Resnik (1999), they use heuristics based on noun number agreement and on co-occurrence frequency between the nouns. They also employ heuristics based on obviously disambiguating elements, such as punctuation and other orthographic features. The following examples illustrate two of these:

buy and sell, orders

(buy and sell) orders

Their heuristics predict that in both these examples *orders* applies to both *buy* and *sell*. Also used are some heuristics from Rus, Moldovan, and Bolohan (2002). These include metrics based on parts of speech of words in the context, requiring a narrowly defined range of these.

Combining their heuristics using a majority voting strategy, Nakov and Hearst achieve precision of 80.61% above a baseline of 56.54%. Their dataset is a collection 428 examples from the annotated Penn Treebank. As with Resnik, being concerned only with nouns allows them to factor in more information than would be possible with a less narrowly defined task. Also, their punctuation and orthographic heuristics provide added precision from what might be considered trivial disambiguation problems.

2.2 Ambiguity Preservation

Researchers in diverse fields have recognised that eliminating ambiguities may not always be necessary, or even desirable. For instance, human translators frequently preserve ambiguity to keep translations faithful, and some machine translation systems attempt to mimic this. In natural language generation, ambiguity can be preserved to avoid the trouble of ensuring that the input is totally unambiguous. Both these applications require awareness of the interpretations that an ambiguity permits. This is necessary so that the ambiguity can be re-presented in a new way that also permits these interpretations. In both machine translation and natural language generation, inaccuracies result from failing to represent all possible interpretations of the input in the output. Note, however, that no account is necessarily taken of whether such preserved ambiguities lead to misinterpretations or not.

Alternatively, the ease with which most ambiguities are understood by humans means that a lot of them can be *ignored*. These ambiguities are always interpreted in the same way, and so they will not lead to misunderstandings. They can therefore be preserved. If such ambiguities do not have to be reformulated, it is not necessary to enumerate all possible readings of them. If these ambiguities can be located in text, they can merely be forgotten about. This is exactly what we are trying to capture with respect to innocuous ambiguities in all contexts.

Several practical benefits accrue from preserving ambiguity. From a semantic standpoint, preserving ambiguity limits the need for world knowledge to disambiguate (in language understanding) or for full specification (in language generation). From a syntactic standpoint, Allen (1995) advocates the encoding of uncertainty as a means of improving parsing efficiency. Allowing uncertainty means that a decision can be delayed until further data is encountered. The backtracking required to change that decision can then be avoided. Encoding of uncertainty can be achieved by use of underspecification (Reyle 1993). To implement this, Allen outlines the use of *packed representations*. These avoid the need for encoding duplicate information by factoring as much common information as possible from the various syntactic readings of an ambiguity. The syntactic alternatives are represented as local disjunctions without conversion to disjunctive normal form. Packed representations are popular in ambiguity preserving approaches, and we refer to them in our discussions below of ambiguity preserving implementations.

We discuss below research in two fields, machine translation and natural language generation, where ambiguities can usefully be preserved.

2.2.1 Natural Language Generation

Ambiguities can be preserved in natural language generation to enable the use of ambiguous input. This is sometimes done with a view to providing one side of a machine translation process. Shemtov (1997) articulates the core premise that language generation is "a many to many relation, between an input representing alternative meanings and an output consisting of multiple renditions of these meanings". This presumes that there are generally multiple meanings contained within the semantics of a single input (in addition to those within any output). So, due to this inevitable ambiguity, full disambiguation may not be a realistic or even a desirable goal. Shemtov uses packed representations to capture the semantic commonalities between different interpretations within the source semantics. These representations are then passed on to the generation stage of the process, thereby preserving the ambiguity. Additional sources of information may then be used to specify them more fully (and more efficiently). Shemtov appears not to have implemented his ideas empirically, so their viability cannot be ascertained.

Nitrogen (Knight and Langkilde 2000) is a natural language generation system primarily intended for use in machine translation. It incorporates the capability for preserving ambiguity. It uses a hybrid approach to implement this. Firstly, its knowledge base is shallow and its inputs are underspecified, so it over-generates considerably. Fluent sentences are then extracted from the resultant parse forest by use of statistical information.

The ambiguity preservation aspect of this process is in the finding of a sentence in the parse forests that represents the many possible meanings of the input. Knight and Langkilde claim that their system avoids the pitfalls encountered by other more deterministic systems. They state that it has been “very easy to adapt Nitrogen to perform a wide range of ambiguity preservations”. That this is possible is encouraging news for us in our work. Unfortunately, however, no results are presented to confirm these claims.

2.2.2 Machine Translation

An ambiguity can be preserved in order to ensure veracious translation from source language to target language. This is only possible when equivalently ambiguous phrases exist in both languages. For instance, in the following sentence, the prepositional phrase *in Madrid* can attach to either the verb *visit* or the noun phrase *our ambassador*:

> *We visit our ambassador in Madrid*

It is ambiguous whether the visit takes place in Madrid, or Madrid is the ambassador’s current posting (regardless of where the visit takes place). This ambiguity is also expressed in the German translation:

> *Wir besuchen unseren Botschafter in Madrid*

The ambiguity can be clarified in both English and German by rephrasing the sentences. But often clarification of the original meaning is not possible. It may also not be necessary if, for instance, the ambassador to Madrid is in Madrid. In both cases, preserving the ambiguity in translation may be desirable.

However, such equivalently ambiguous phrases are not always available, especially with highly dissimilar languages. Ambiguity preservation therefore does not always have wide application in machine translation. Emele and Dorna (1998) explain the process of ambiguity preservation between languages, and try to overcome this lack of scope. They use packed representations to preserve PP-attachment ambiguities such as the example given above. However, if the ambiguities can only partly be preserved in the target language, they unpack these representations only as much as necessary. Human translators then make the final decisions about the translations, choosing between the interpretations remaining in the representations. Emele and Dorna present their ambiguity preservation technique as a theoretical architecture, and offer no empirical validation of it.

2.3 RE-Specific Approaches to Ambiguity

Ambiguity has traditionally been tackled in RE using techniques either for *avoiding* or *detecting* it. We are solely concerned with written requirements, so do not consider other types of communication involved in the requirements part of the software life-cycle. Requirements elicitation is an example of this. Preventing ambiguities during such processes is also an issue, but beyond the scope of this research. Additionally, ambiguities can be *preserved* in requirements, though this has received scant attention compared to approaches classifying ambiguity as a defect to be avoided.

Most previous work on ambiguity in RE has focused upon detecting and disambiguating *RE-specific* ambiguities. These are defined as ambiguities arising from the RE context. This context is composed of the application, system and development domains. Analysing RE-specific ambiguities therefore pre-supposes a great deal of background knowledge (Kamsties, Berry, and Paech 2001). Compared to more generic *linguistic* ambiguities, such as our test case ambiguity, they are reported to account for more ambiguities in requirements (Kamsties 2001). But many RE researchers, for instance Berry, Kamsties, and Krieger (2003), also realise the danger of more generic linguistic ambiguities in RE.

Below we discuss some ideas from the RE literature about preserving ambiguities temporarily, including parallel ideas that recommend ignoring some inconsistencies. Then we look at the bulk of research on ambiguity in RE, which attempts to either avoid or detect it.

2.3.1 Preserving Ambiguity in RE

The idea of *deliberate ambiguity* (or *intentional ambiguity*) is discussed by Kamsties (2001). He claims that this situation can arise when a stakeholder, for example a customer, leaves aspects of a requirement purposely ambiguous. This could be for the perfectly valid reason that the way the requirement is implemented is a *design decision*. It could also be for the perhaps less valid reason of "diplomacy" (Goguen 1994). In the former case, the customer, being interested only in the finished product, might want to leave that decision to other stakeholders. It would be desirable to preserve such an ambiguity in the requirements — at least in the *early-phase* requirements. The ambiguity is then resolved at the stage of the software life-cycle when design decisions are made. This means that it is only temporarily preserved. However, we feel that deliberate *vagueness* (or *underspecification*) is much more likely than deliberate ambiguity. Gause and Weinberg (1989) offer the following example, containing many underspecifications that might be preserved until design decisions are made or further information is obtained:

> *Create a means for protecting a small group of human beings from the hostile elements of their environment*

Temporary ambiguity preservation is considered to be effective for other pragmatic reasons. Goguen (1994) observes that it facilitates the gradual evolution of requirements, and avoids sub-optimal prejudgement of cost trade-offs. Despite this, we are unaware of any empirical research determining which ambiguities are worth preserving temporarily in requirements. We suspect, however, that they will be more likely to be ambiguities of specific types rather than any which do not cause misunderstandings. We are also unaware of any empirical research which investigates more permanent preservation of ambiguities in requirements.

A project addressing a parallel problem can also usefully be discussed. Nuseibeh, Easterbrook, and Russo (2001) consider the advantages of preserving *inconsistency* in requirements. As explained in Section 1.1.2, inconsistency can have implications similar to ambiguity. Nuseibeh et al. argue that if the cost of fixing an inconsistency outweighs the risk of ignoring it, then it makes no sense to fix it. It can be preferable to let some known inconsistencies remain in text. They present a framework within which inconsistencies can be detected and their cause discovered. Three handling strategies are then offered. These are to resolve the inconsistency, tolerate it temporarily, or ignore it completely. The second of these strategies corresponds to temporary preservation, as recommended by Goguen. The third goes one stage further, and is of more interest to us as it relates to our treatment of innocuous ambiguity. The framework of Nuseibeh et al. has been developed in various forms. However, we are unaware of any empirical work that tests how inconsistencies worth preserving might be distinguished from others.

2.3.2 Avoiding Ambiguity in RE

Ambiguity can be avoided in RE by use of formal methods, controlled languages and less formal reference-based approaches. All these techniques promise safe and manageable techniques for writing documents free from ambiguity. However, they can require considerable effort to learn or to use. Kamsties (2001) concluded in 2001 that many RE practitioners were reluctant to utilise such methods. In the same year, Mich (2001) cites an internal study at Trento University that surveyed "documents used for requirements analysis". She reports that only 16% of these documents were couched in "structured natural language (e.g. templates, forms) and only 5% in formalised language". The remainder still chose to use “common natural language”.

Formal Methods

For some time, attempts have been made to use formal techniques to express requirements: for example, in terms of finite state machines (Whitis and Chiang 1981) and program description languages (Caine and Gordon 1975). This type of approach has been brought to prominence by the development of formal (or semi-formal) requirements specification languages such as Z (Spivey 1992) and UML (Rumbaugh, Jacobson, and Booch 1999). A primary aim of such methods is to avoid anything ambiguous being written. However, using such languages can also be seen as a form of disambiguation. This is because it can force ambiguous informal specifications to be translated into unambiguous but sometimes incorrect formal specifications. (Hanks, Knight, and Strunk (2001) have pointed out that natural language is still essential to understand the formalism of any formal method.) This might not become so serious a problem if all the stakeholders understand the formal specification language used. But, as already mentioned, this is not to be relied upon.

Other formalisms use non-linguistic elements to avoid ambiguity. For instance, the tabular specifications in the SCR (Software Cost Reduction) requirements method aim to specify software requirements unambiguously and concisely. They are claimed to be easily understood and developed by engineers (Heitmeyer, Jeffords, and Labaw 1996). However, a “large amount of detail” is still required to apply such formal methods, and this is difficult to obtain in practice without automation, which is not always available (Heitmeyer, Jeffords, and Labaw 1996).

Controlled Languages

A controlled language is a subset of a language with a smaller grammar and (usually) a smaller vocabulary. It is carefully designed, and its use is controlled, so that ambiguity is minimised. Ideally, this is not at the expense of reducing readability and expressiveness unreasonably. The linguistic control can give the added benefit of document standardisation, and can also make translation easier. Controlled languages are usually designed for specific applications, where exacting standards of writing must be met. Attempo Controlled English (ACE) (Fuchs and Schwitter 1996), intended for writing requirements, imposes restrictions on grammar and style but requires users to populate the lexicon. CREWS (Achour 1998) is a controlled language for authoring RE scenarios. This is a more restrictive formalism than ACE, prescribing semantic aspects of language as well as syntactic aspects.

Osborne and MacNish (1996) discuss the drawbacks of controlled languages, claiming that they reduce the *habitability* of language. They may be no less irritating to use than a formal language, and they may force naïve users to choose phrasing they do not entirely understand. Also, controlled languages generally omit non-linguistic formats preferable for representing elements such as algorithms and structural relationships. Kamsties (2001) finds no RE usage of them in medium- or large-scale industrial projects as at 2001.

Style Guides, Lexicons and Other Informal Guidelines

Style guides are reference works that help authors avoid ambiguities by making them aware of pitfalls in the language used. They include, for instance, the guidelines suggested by Kovitz (1999). Common style recommendations focus on the use of ambiguous quantifiers, modal verbs, passive voice and anaphora. However, Sawyer, Rayson, and Cosh (2005) claim that such rules will always be broken by requirements writers. This does not give confidence in the adoption of style guides and other aids requiring voluntary application. Additionally, Easterbrook et al. (1998) report that such informal techniques do not provide the assurance to RE practitioners that formal techniques can.

Glossaries and lexicons of domain-specific terminology are sometimes recommended, as for example by Hillelsohn (2004), as ways of avoiding ambiguity. However, it is generally assumed that such reference works will be of limited size. They would therefore only be useful for solving the most narrowly-defined RE-specific ambiguities. Such lexicons cannot begin to contain all possible combinations of words. They therefore will not adequately cover more generic linguistic ambiguities, or even unforeseen RE-specific ones. For this reason they will always be of only limited use.

Other, less formal, recommendations for appropriate style have been suggested. For instance, Kovitz (2002) suggests simply adding redundant (i.e. repeated) references to context in informal natural language requirements. The readers will then be able to disambiguate each ambiguous expression more easily. This might constitute an annoying style of writing. However, Kovitz asserts that it is an inexpensive, acceptable and broadly-applicable method of avoiding ambiguity.

2.3.3 Detecting Ambiguity in RE

Ambiguity can be *detected* in requirements by using fit criteria, test cases and inspection techniques. Of these, inspection techniques offer the most established way of detecting ambiguity. We survey inspection technique approaches that consider ambiguity, and then discuss in detail one of these that tackles it with more thoroughness.

Fit Criteria and Test Cases

A *fit criterion* adds the conditions that must be true for a requirement to be successfully implemented, often adding quantification where none exists before (Robertson and Robertson 1999). *Test* (or *use*) *cases* can be used in conjunction with this. They look at input and output states and help refine requirements by making inferred aspects more concrete (Kamsties 2001).

However, Kamsties (2001) observes that fit criteria and test cases generally only reduce vagueness and generality. These concepts are similar to ambiguity, but arise from lack of specificity. Ambiguity, on the other hand, tends to arise from the possibility of more fundamentally differing meanings. Fit criteria and test cases will therefore not be sufficient for analysing ambiguity in requirements.

Inspection Techniques

Inspection techniques are known to be effective and efficient ways of reducing errors of various kinds in requirements (Shull, Rus, and Basili 2000). They can be used to search for ambiguities in completed requirements documents. They have, however, more often been used to detect inconsistency and incompleteness in requirements (Zowghi, Gervasi, and McRae 2001; Porter, Jr, and Basili 1995). Those that do address ambiguity use checklists or other sets of manually applied heuristics (Rupp 2000; Gause and Weinberg 1989; Freedman and Weinberg 2000). These checklists contain questions that will reveal some ambiguities often encountered in requirements. They often draw attention to lexical, referential and discourse ambiguities. However, such techniques tend to be incomplete and time-consuming solutions to the problem. Kamsties, Berry, and Paech (2001) observe that most inspection techniques addressing ambiguity merely ask the question "is the requirement ambiguous?". This is true even for well-developed scenario-based inspection approaches, both of the defect-based reading (Porter, Jr, and Basili 1995) and perspective-based reading (Shull, Rus, and Basili 2000) varieties. The questions posed in such approaches bring ambiguity to the readers' attention. They may not, however, make the readers aware of the extent to which misinterpretations might occur.

Kamsties, Berry, and Paech (2001) present a study aimed at detecting ambiguity in requirements more thoroughly than those discussed above. They use a hybrid approach based on an inspection technique. They augment incomplete ambiguity checklists with questions about ambiguity obtained by building a formal (or semi-formal) model. The latter questions depend on the modelling language used. However, the model need not be built. There need only be awareness of the ambiguity-exposing questions that building it would generate. Kamsties et al. apply their technique to some complex requirements in which many types of ambiguity were identified by hand. They achieve higher detection rates than when simpler methods are used, and this improvement is deemed to be statistically significant. This is despite their detection rate being no more than 25%. Interestingly for us, they recognise that not all ambiguities need to be detected. However, there is no certainty that they detect those that most need to be detected. They conclude that their technique successfully raises awareness of ambiguity in RE, and successfully combines generic and model-specific approaches.

All the inspection approaches discussed here attempt detection of ambiguities. They do not seek to determine whether or not these ambiguities are dangerous and need to be addressed. They also will not be as thorough as the most well-developed ambiguity avoidance techniques such as controlled languages and formal methods. In their defence, most researchers advocating these techniques appear to appreciate this and also the impossibility of achieving total awareness of ambiguity.

2.4 Evaluating Dangerousness of Ambiguity

Other researchers have remarked on the fact that some ambiguities are more likely than others to lead to misunderstandings. Others consider quantifying this characteristic. Some of these go further, suggesting thresholds for deciding when this characteristic is sufficiently in evidence to be noteworthy. All of these researchers have contributed to the starting point for our own approach.

Firstly we discuss RE research, then theoretical NLP research that considers distinguishing ambiguity by taking account of how dangerous it might be. Some previous NLP approaches with a more practical orientation are then introduced. Lastly we look briefly at research into quantification of the dangerousness of ambiguity in fields of research outside of NLP.

2.4.1 Dangerousness of Ambiguity in RE

Kamsties, Berry, and Paech (2001) survey the ways in which NLP techniques can be used to detect ambiguity in requirements. They point out that NLP techniques "raise more ambiguities than are perceived by humans". These extraneous ambiguities will not lead to misunderstandings because people are generally agreed upon a single interpretation of them.

Other RE researchers recognise that some ambiguities are more dangerous than others. We discuss observations they have made about the potential problem of not acknowledging the existence of ambiguities in requirements. We then look at attempts which have been made at quantifying ambiguity in requirements.

Acknowledgement of Ambiguity in RE

Not acknowledging the presence of ambiguities in requirements has been recognised as a serious problem by some researchers, e.g. (Gause and Weinberg 1989; Mullery 19; Berry and Kamsties 2005). However, the idea has not been pursued or investigated empirically in the RE literature. It has generally been thought that ambiguity is best solved by disambiguation. The assumption is that, by performing disambiguation as comprehensively as possible, the problem of unacknowledgement will also disappear.

Kamsties (2001) is one researcher who has investigated the problem of unacknowledgement of ambiguity in RE. He conducted a case study on the use of different formal and semi-formal requirements specification techniques. There were several aims of this exercise. The one of interest to us concerns measuring the extent to which ambiguities, and other “defects”, are resolved unconsciously. The participants in the task were required to develop requirements models using each of the specification techniques. The numbers of ambiguities — and conflicts and incompletenesses — discovered by the development processes were noted. Also noted were the numbers of these defects that were unconsciously removed — i.e. disambiguated, in the case of ambiguity. Kamsties found that 20% of the known ambiguities were misinterpreted. This represents unacknowledgement of ambiguity: the participants of the test assigned different readings to the experts who disambiguated the original requirements. Misinterpretation of the known incompletenesses was only 4%. This indicates that ambiguity is a much more dangerous source of misunderstanding.

Kamsties’ case study goes further. He is interested in the development of requirements and the many forms in which they are modelled. To this end, he is concerned with the repercussions of not acknowledging ambiguity. When developing the requirements models, 57% of the known ambiguities were correctly disambiguated but not acknowledged to be ambiguities. This process contributes to the correctness of the models, but the original specifications still contain ambiguities which might cause misunderstandings in the future. This might also be classed as unacknowledged ambiguity. However, the ambiguities concerned are more realistically classed as innocuous, as they are interpreted in the same way by the participants. Kamsties concludes, however, that this type of ambiguity may become a serious threat in the future.

Ambiguity Quantification in RE

The most notable attempt at quantifying ambiguity for the purposes of requirements analysis is made by Mich (2001). She presents ambiguity measures that address two types of ambiguity: lexical ambiguity and “phrase and sentence” ambiguity. For the former, firstly, a function of the number of possible meanings of any given word is used.

Each meaning is weighted according to how frequently it is found, creating a *weighted semantic ambiguity* function. Secondly, a function of the number of syntactic roles that a word can have is used. Each role is weighted according to how frequently it is used, creating a *weighted syntactic ambiguity* function. For the latter ambiguity type, the weighted semantic ambiguity function is also used, for all words in the sentence or phrase. Secondly, a function of the possible parsing trees of the sentence or phrase is used. Each parsing trees is weighted with a "penalty", which represents the effort required to make the parse. Mich claims that this parsing effort equates to the effort required to interpret the sentence. The numbers of possible meanings of words are obtained from WordNet. The numbers of possible syntactic roles of a word are obtained from a large semantic net incorporated in the LOLITA system (Morgan et al. 1996). This system also provides the penalties for each parse tree.

This method of measuring ambiguity using generic resources is appealing, as little work needs to be done to assemble the data. Unfortunately, however, no results appear to be available from this study concerning its overall accuracy. The dataset is tiny, and it therefore cannot be judged how successful the approach might be. Also, the approach relies on probabilistic analysis of language usage, and not directly on human perception, which we believe to be the key to evaluating ambiguity.

2.4.2 Theoretical NLP Approaches to Dangerousness of Ambiguity

Here we introduce theoretical ideas from the field of linguistics which seek to evaluate ambiguities according to how dangerous they might be. Poesio's notion of *perceived ambiguity* provides an interesting and relevant philosophical distinction between different realisations of ambiguities. Van Deemter's Principle of Idiosyncratic Interpretation sheds some light on why some ambiguities are not acknowledged. Van Deemter and van Rooy introduce, respectively, the notions of *vicious ambiguity* and of what makes a sentence *truly ambiguous*. These last two ideas are largely the result of work carried out in parallel with our own.

Perceived Ambiguity

Poesio's (1996) notion of *perceived ambiguity* is based on the idea that humans can "entertain more than one interpretation at a time, and they may not be able to choose between them". Perceived ambiguity is also experienced in cases where multiple readings are *intended* to coexist happily, for example in the case of puns and for rhetorical effect.

Poesio distinguishes perceived ambiguity from *semantic ambiguity*. The latter is concerned with "the interpretation that the grammar assigns to a sentence", the former with "the process by which interpretations are generated". Poesio makes the further distinction that "semantic ambiguity has to express the truth-conditional properties of an expression", but that perceived ambiguity is realised via reasoning processes consisting of "defeasible inferences that are not supported by the semantics of ambiguous expressions". Poesio's philosophical distinction between language-inherent and human disambiguating factors is important for us: the latter are what actually cause an ambiguity to be misunderstood and therefore nocuous.

Principle of Idiosyncratic Interpretation

van Deemter (1998) introduces a Principle of Idiosyncratic Interpretation. Much ambiguity is resolved by consideration of context, but several factors, including the reader's degree of competence in the language used, can affect understanding. An aspect of van Deemter's Principle of Idiosyncratic Interpretation is that, in any given context occurrence, different human interpreters may be unaware of each other's interpretations. Especially in technical domains, common sense may not always come to the rescue. Some readers may have *niche interpretations* of particular linguistic constructions, i.e. interpretations which are situation-specific. Especially if they are not native speakers of the language used, the common-sense interpretation may not occur to them. Conversely, those less well versed in a domain may not know the niche interpretations and may assume the common-sense interpretation is the only one possible. This affects perceptions of ambiguity.

Vicious Ambiguity

van Deemter (2004) uses the term *vicious ambiguity* to refer to an ambiguity which has no distinct interpretation that is strongly preferred over other interpretations. He uses a threshold to quantify "strongly". An ambiguity is said to be viciously ambiguous *simpliciter*[4] if it has no interpretation that is strongly preferred over *all* other interpretations. Less unequivocally, any given interpretation can be viciously ambiguous with respect to any other given interpretation.

Viciousness is determined using probabilities taken from corpus data. A form (i.e. a surface realisation) F is said to be viciously ambiguous with respect to a content (i.e. a meaning) C, taking into account a threshold t such that the following is true:

$$t \cdot p(C|F) < p(C'|F)$$

The form F is said to be viciously ambiguous simpliciter if it is viciously ambiguous with respect to all *C'*. (Van Deemter adds the stipulation that contents C and C' must be "sufficiently different that the choice matters". However, he does not fully explore the fact that such an assertion is very hard to make.) The notion of vicious ambiguity is then used for determining *superoptimality*, by taking into account the ambiguity of a contents with respect to form in addition to the reverse situation just discussed.

[4] i.e. absolutely ambiguous, without qualification

van Rooy (2004) defines a notion of *true ambiguity*: "a sentence can be truly ambiguous only if there are at least two interpretations of this sentence that are optimally relevant". Like vicious ambiguity, it evaluates ambiguities according to how they might be differently interpreted. However, it tends by its nature to be a much stricter definition. Also, optimal relevance is hard to evaluate practically.

Van Rooy considers *relevance* to be a measure of the *utility* value of any given interpretation, based on Sperber and Wilson's (1982) relevance principle. This principle (into which the substitutions *author* for *speaker* and *reader* for *hearer* can be made) states that: "The speaker tries to express the proposition which is the most relevant one possible to the hearer" (van Rooy 2004). Relevance, as defined for the purposes of this principle, depends on two factors: the processing effort needed to come to this interpretation, and the number of contextual implications that the interpretation gives rise to. Van Rooy's interpretation of Sperber and Wilson's prevalence of contextual implications factor is that it is a measure of utility. The aim of the participants in a communication is to maximise utility, and several assumptions are bound up in this idea. It is assumed that the author intends the sentence to have the interpretation that has the highest utility value for the reader. Conversely, the reader selects the interpretation of the sentence which he/she judges to be the most relevant, and assumes that this is the one intended by the author.

Both the aforementioned assumptions about how author and reader arrive at the same interpretations are at odds with Poesio and van Deemter's claims: authors and readers cannot always be relied upon to behave consistently with one another. Such is the strictness of van Rooy's notion that a reader needs to be torn between choosing from two equally likely interpretations of an ambiguity before it is considered truly ambiguous. This excludes situations which are less clear-cut but where misinterpretations arise, for instance due to differences in language competence between author and reader or other contextual factors.

2.4.3 Practical NLP Approaches to Dangerousness of Ambiguity

We introduce here some examples of research that differ from the more purely linguistic notions discussed previously: they are substantiated with some empirical application or are designed with an empirical application in mind.

Essential Ambiguity

Mich's (2001) NLP approach to ambiguity in RE — see Section 2.4.1 — introduces the idea of *essential ambiguity.* This occurs when a sentence has incurred equal penalties from the LOLITA parser (Morgan et al. 1996) on two or more parse trees. In addition to this, the penalties must be below a certain limit for an essential ambiguity to occur: penalties above this limit supposedly indicate a structural problem in the text, perhaps resulting from "missing or repeated parts of speech". The fact that she sets this limit as high as 1000 suggests that it would be quite rare to find parses with the same number of penalties, or that sentences incurring few penalties might be unrealistically likely to be classed as essential ambiguities. Unfortunately there appears to be no empirical work to show how prevalent her essential ambiguities are, or how structurally deficient, rather than ambiguous, high-penalty sentences are.

Spurious Ambiguity

Park and Cho (2000) discuss the idea of *spurious ambiguity* to account for readings of ambiguities with "irrelevant" syntactic structures generated by their combinatory categorial grammar on Korean. They say that the remaining structures contribute to *structural ambiguities.* They claim that the distinction is specific to the type of grammar they are using. Discriminating between these two forms of ambiguity is just one of the techniques they use in their empirical work, and they claim to reduce the number of structures generated by 72.1%. Unfortunately there is no clear indication of the contribution to this performance by the spurious ambiguity detection module, or of the module's reliability.

This concept, whereby a parse forest is thinned so that one (or several) parse trees representing preferred readings can be brought into focus, is also used with other systems without the dismissed ambiguities receiving a charactising name. For instance, hybrid natural language generation systems use corpus information to perform the thinning function by generating ranking parse trees in order of statistical likelihood and then dismissing the least likely (Langkilde 2000).

Semantic Indeterminacy

Lauer (1995) uses the concept of *semantic indeterminacy* when tackling the binary decision problem of disambiguating the bracketings in noun compounds of the form *noun1 noun2 noun3*. He uses this term to refer to situations where the possible bracketings of such ambiguities "cannot be distinguished in the context". This would in theory be of interest to us, as it might imply that such ambiguities are particularly liable to be misunderstood. However, Lauer erroneously attributes this name for the concept to Hindle and Rooth (1993), but appears to be using the *systematic ambiguity* notion referred to in that paper. In that case he is referring to a type of ambiguity that gives multiple readings all of which have approximately the same semantic content and therefore do not lead to misunderstandings. This is discussed below.

Systematic Ambiguity

Hindle and Rooth (1993) introduce the concept of *systematic ambiguity* when attempting disambiguation of PP attachments. An attachment is systematically ambiguous when "given our understanding of the semantics, situations which make the interpretation of one of the attachments true always (or at least usually) also validate the interpretation of the other attachment". This concept is exemplified in the following sentences, which give examples of systematic locative ambiguity and systematic benefactive ambiguity respectively:

I am going to visit some friends in this town

We are arranging a birthday party for John

In the former, the *visit* event is located in the same place as the *friends*; in the latter, John benefits from the *arranging* as well as the *party.* It therefore doesn't matter where the prepositional phrase is attached.

2.4.4 Ambiguity Quantification in Other Fields

Information retrieval is a field closely related to NLP, where ambiguity is also of considerable interest. For instance, Cronen-Townsend and Croft (2002) develop a *clarity score* in order to estimate the lack of ambiguity of queries with respect to the documents they are intended to query. This measures the relative entropy between the query language model and the language model of the corresponding documents. The language models are based simply on unigram distributions. Cronen-Townsend and Croft seek to measure ambiguity without resolving it, as we do; they do not explicitly discuss imposing a threshold on their clarity score, above which a query is declared to be unambiguous, but this could be implemented.

2.5 Summary

In this chapter we have introduced several types of previous research as background to our work presented in this thesis. Firstly, using coordination ambiguity as an example, we have presented the traditional way of dealing with ambiguity, which is to eliminate it, followed by a more general discussion of the less frequently chosen option of preserving it. We then surveyed the ways in which RE researchers have approached ambiguity. In the following section we discussed the ideas of other researchers that have something in common with our own approach. This included reference to unimplemented theories and ideas in RE and NLP, and to NLP projects which had included some empirical investigation. We conclude that much previous research motivates the need for a model of ambiguity that evaluates how likely it is to lead to misunderstandings. However, none of this related research describes and implements empirically a suitable model that accounts for the human perceptions that cause this.

Chapter 3

Model of Ambiguity

In this chapter we motivate and describe our model for discriminating ambiguities which may lead to misunderstandings from those which will not. The former are *nocuous* ambiguities; the later are *innocuous* ambiguities. We explain how the former, which is the type that concerns us, can be acknowledged and/or unacknowledged. A multi-layer representation is used to describe the distinction that we make, and each layer is discussed in turn. Our model of ambiguity uses human perception as a basis for three important criteria: determining preferred readings of ambiguities, determining the dividing line between nocuous and innocuous ambiguities, and allowing disambiguation by hand.

We argue that our approach to ambiguity, represented by our model, is novel and well-motivated. Our approach shares some characteristics with others in the literature, but is distinctive in the several ways. Firstly, and most vitally, it incorporates the notion that some ambiguities are too likely to be interpreted in different ways: assigning one reading to them (disambiguation) is therefore unwise. Secondly, it takes a pragmatic approach by determining that some ambiguities need not be addressed as their interpretation is obvious. Thirdly, it allows for a flexible dividing line between ambiguities likely to be interpreted differently and those that need not be addressed. Fourthly, it uses human perception as it's criteria for judging ambiguity.

3.1 Introduction

We are concerned with finding nocuous ambiguities: those which may lead to misunderstandings and which may therefore prove costly. We are not in a position to quantify the cost of such mistakes in the real world, as they are rarely traced back to individual passages of text. Therefore we concern ourselves with finding all ambiguities that may lead to misunderstandings. These are ambiguities that are interpreted in different ways: in other words, they are given more than one reading by people. This is distinct from ambiguities that *can* be given more than one reading in theory, but where in practice people always assign a particular interpretation. The latter type may have more than one syntactic structure, but only one structure allows for a commonly used or meaningful interpretation.

Importantly, our approach uses only human perception to identify ambiguity. Our use of human perception to judge ambiguity means that we are able to determine whether humans recognise that an ambiguity is present: we term this situation *acknowledged ambiguity*. We can also determine when different judges have alternative interpretations of an ambiguity yet do not acknowledge this: we term this situation *unacknowledged ambiguity*. These factors are important when deciding if an ambiguity is likely to lead to misunderstandings, and is therefore nocuous.

We introduce a *Model of Ambiguity* that represents our way of looking at ambiguity and classifying it according to our stated aims. This model has a multi-tiered structure, shown in the diagram in Figure 3-1. The tiers are *syntactic structure*, *interpretation* (or *nocuousness*) and *acknowledgement*. What proportion of sentences fall into either of the options on any given layer is dependent on the strictness of the criteria used for

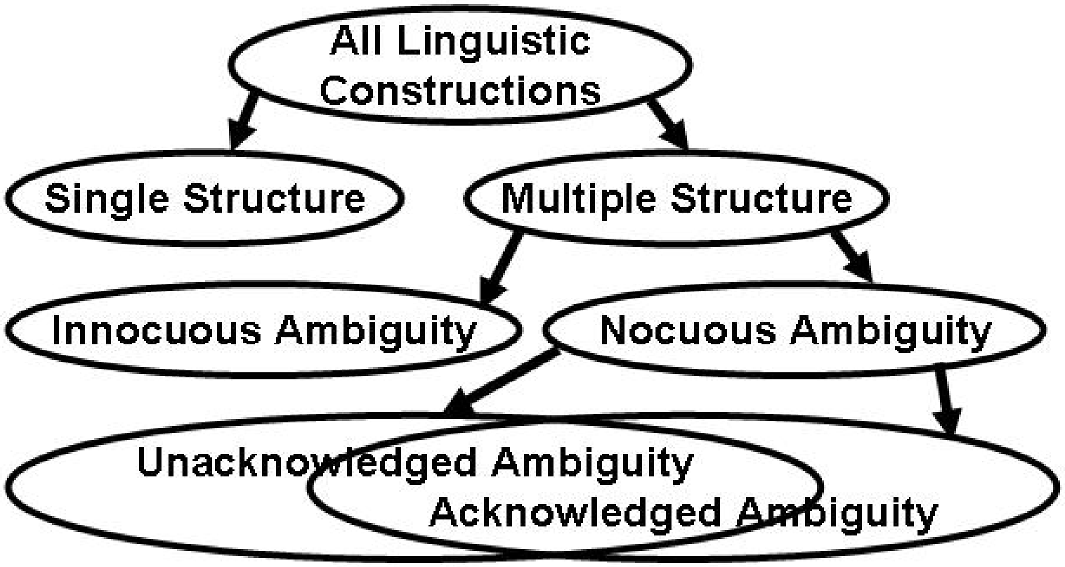


Figure 3-1: Multi-tier ambiguity representation

distinguishing the options. The dividing line between what is nocuous ambiguity and what is innocuous ambiguity is naturally a fundamental concern for us: we make special provision for this by allowing for a flexible intolerance to ambiguity that can be adjusted depending on how serious the implications of misunderstandings are thought to be. The tiers of this model are discussed in the following sections.

3.2 Single and Multiple Structure

The *structure* of text, as compared to its *semantics*, is a much more prescriptive indication of the number of interpretations that are possible. Typically, in the literature on syntactic disambiguation, choosing a single syntactic structure for a sentence amounts to choosing a particular reading; the number of structures that can exist are finite and are specified by clearly defined rules of syntax. In line with this tradition, and for the purposes of this thesis, we will not consider passages of text as ambiguous if they only have one syntactic structure. We refer to these as having *single structure*, and assume, for our purposes, that they have only one possible reading. We concentrate our search for nocuous ambiguities instead on examples where more than one structure is possible, which we refer to as having *multiple structure*.

3.3 Acknowledgement and Unacknowledgement

Here we present the two ways in which we consider an ambiguity to be nocuous. People can acknowledge it as being ambiguous, or it can go unacknowledged, and both scenarios can be applicable to some degree for any ambiguity. We discuss these before introducing nocuous ambiguity, as they are what determines whether an ambiguity is nocuous.

After introducing unacknowledged and acknowledged ambiguity, we present theories from psycholinguistics that support this distinction. These theories explain how both types of ambiguity may occur, and also, as a counter-argument to our own, how they may be mitigated by human thought processes and become less of a problem. Then we explain how the ambiguity models of other researchers, discussed in Section 2.4.2, do not fulfill the same function as our own. Lastly we present some arguments from our chosen application domain that support the significance of our research.

3.3.1 Unacknowledged Ambiguity

An ambiguity can go unnoticed. When various people think there is only one reading of an ambiguity, and yet the readings they ascribe are different from one another, this is *unacknowledged ambiguity*. If a suitably large number of people assign different readings to an ambiguity, that ambiguity can be said to be generally an unacknowledged ambiguity. How large the “suitably large number of people” should be is determined by how tolerant we are of unacknowledged ambiguity.

Unacknowledgement is the worst scenario for any nocuous ambiguity, as the ambiguous passage will not get rewritten and may cause misunderstandings at any time. For this reason, it is the worst scenario for any ambiguity in text used for instruction, such as requirements. An unacknowledged ambiguity in a requirement may be carried forward into later stages of the software life-cycle. A system may then be built incorrectly, and correcting it or rewriting it may be very costly.

We present here an alarming example of unacknowledged ambiguity to show its potential dangers. Misunderstandings arising from mismatch of language patterns have been the cause of air traffic accidents and near-accidents (Jones 2003). These include the worst disaster in aviation history, at Tenerife in 1977, when a Dutch pilot spoke the following sentence:

> *We are now at takeoff*

He is thought to have translated *at* to an equivalent Dutch word, and then interpreted in a way that was likely in Dutch but not in English. The air traffic controller made a different interpretation. Unfortunately, one referred to the process of taking off and the other to a position on the runway. It is likely that neither the air traffic controller nor the pilot had acknowledged that different interpretations of the sentence were possible.

3.3.2 Acknowledged Ambiguity

We say that when somebody realises that a given linguistic construction can be interpreted in more than one way, that construction is an *acknowledged ambiguity* for that person. When a suitably large number of people acknowledge this ambiguity, it can be said to be generally an acknowledged ambiguity. The number of people needed for this to be the case is determined by how tolerant we are of acknowledged ambiguity. Acknowledged ambiguity may seem much less dangerous than unacknowledged ambiguity. However, for the reasons given below, we consider that it contributes to nocuous ambiguity.

Acknowledged ambiguity is not necessarily a problem for those who acknowledge it.

They can try to obtain the intended reading by asking the author. If that is not possible, they can canvass other people's opinions and hopefully obtain a clear verdict on what the preferred reading is. However, access to authors is often not possible, and an infinite number of opinions would be needed to be certain of the preferred reading. Because different readings are possible for an acknowledged ambiguity, it could easily, and unpredictably, be an unacknowledged ambiguity for another hitherto unknown reader.

Additionally, a reader of a linguistic construction can acknowledge an ambiguity and assume that it was put there intentionally by the author. The reader may then assume that freedom is being given about how the construction should be interpreted. But if the ambiguity was unintentional, this can then lead to a misunderstanding.

3.3.3 Supporting Psycholinguistic Theories

Here we present some research from the psycholinguistics community which remarks upon the acknowledgement and unacknowledgement of ambiguity. We do not model or implement any of the ideas introduced here. Investigating our model of ambiguity using a psycholinguistics approach would require a different type of thesis to the one presented here.

Psycholinguists have for some time been interested in how humans process ambiguity. Kess and Hoppe (1981) state that it is the unacknowledged aspect of ambiguous sentences which makes such process analysis noteworthy. They go on to present the debate about whether we generally process all possible readings of an ambiguity, and choose the one that then seems the most appropriate, or whether we only compute one reading for any given ambiguous sentence. The former scenario suggests that acknowledged ambiguity is the norm: consideration is made of the various possible readings of an ambiguity, and the reader is able to interrupt the interpretation process to acknowledge that the ambiguity might be too ambiguous to be reliably given a single interpretation. The latter scenario could quickly result in unacknowledged ambiguity, with each reader fixing on one interpretation — context could play a large part in focusing readers' decisions on just one reading, but context can also be interpreted variously.

The potential for unacknowledged ambiguity when readers compute only one reading for a sentence may be mitigated by certain psychological factors. Hobbs (1983) presents an argument that people do not necessarily disambiguate sentences fully but are still able to reach an informed interpretation of them. According to this way of thinking, people are able to hold several different interpretations of an utterance simultaneously in their minds, and still understand what was meant by the sentence. They will therefore only unconsciously acknowledge an ambiguity.

Poesio (1996) suggests that humans make unconscious use of an underspecified language to process highly ambiguous sentences successfully. This may account for the effect that Hobbs is describing. The existence of such a facility would be a contributing factor to reduction of acknowledged ambiguity, as well as of unacknowledged ambiguity, but there would be inherent dangers of bias and subjectivity as with any unconscious process. Processing effort, often found by measuring processing speed, has frequently been used to measure ambiguity (Kess and Hoppe 1981).

In a task involving completing sentence fragments, MacKay (1966) found that people were inclined to take more time completing ambiguous ones than unambiguous ones, even though they were unaware of the ambiguous status of the fragments during the trial. Such studies support Poesio's notion, and would be worth pursuing in future work to determine how agreement between participants — and therefore the level of unacknowledged ambiguity — correlates with processing speed.

3.3.4 NLP Perspectives

Here we discuss the models of ambiguity, introduced in Section 2.4.2, put forward by other NLP researchers. They provide interesting comparisons with our research, and some motivation for it. However, none of them models ambiguity in the way we consider to be most appropriate or provides a method of implementation.

Poesio's (1996) notion of *perceived ambiguity* is similar to our notion of acknowledged ambiguity. The fact that people are unable to choose a reading will make them stop and acknowledge the ambiguity. The notion of *semantic ambiguity* that he posits opposite to perceived ambiguity contains all unacknowledged ambiguity. However, it must contain all acknowledged ambiguity as well, as “most, if not all, sentences are semantically ambiguous”. The distinction he makes is therefore not implementable in a way that is useful to us.

Van Deemter's (1998) *Principle of Idiosyncratic Interpretation* describes unacknowledged ambiguity, though not a method for quantifying it. His discussion of this concept gives some reasons for how unacknowledged ambiguity can come into being. His notion of *vicious ambiguity* (van Deemter 2004) is intended for determining the status of ambiguities on an individual basis. It is not designed to compare ambiguities. It is empirical, since it uses corpus data, but is not based directly on human perceptions. Van Deemter offers no empirical test of either these ideas.

Van Rooy's (2004) notion of what makes a sentence *truly ambiguous* is of interest to us as it involves consideration of whether an ambiguity in the sentence is acknowledged or not. However, he makes assumptions that speakers and hearers have knowledge of each others' understanding of language. This leads to the conclusion that unacknowledged ambiguity will tend not to occur. This is contrary to our hypothesis, and to the ideas of Poesio and van Deemter, also discussed in Section 2.4.2.

3.3.5 RE Perspectives

From a requirements engineering perspective — though this will also apply in other domains — Berry, Kamsties, and Krieger (2003) suggest that communication is more reliable when it is between people who have similar language abilities. Therefore, it is more reliable between two people who are not well versed in the rules of the language being used, than it is between one who knows the rules well and one who does not. The reasoning is that, in the former case, both author and reader will tend to misuse the language in similar and commonly accepted ways, and the latter is therefore more likely to understand the intent of the former. This implies that ambiguities will tend to be correctly categorised as either innocuous or nocuous for people with an equally inadequate grasp of the language, even if the reasons for such categorisation are false. Unacknowledged ambiguity will thereby be reduced in this scenario. This reasoning may be valid if people write and interpret the language in ways that are more commonsensical than the actual rules of the language. However, we are not wholly convinced by the hypothesis. People with different first languages may well have different language patterns that they apply erroneously when writing or reading documents. This will result in differing readings of ambiguities, and unacknowledged ambiguity will result.

3.4 Nocuous and Innocuous Ambiguity

Here we describe how we classify ambiguities as being either nocuous or innocuous. This includes introduction of the theoretical linguistics distinction between *performance* and *linguistic* data, which bears similarity to ours. We then present perspectives from other researchers that motive our research. These are discussions of the approaches to ambiguity introduced in Section 2.4.2 and Section 2.4.3.

Nocuous ambiguities are those which are given more than one reading; innocuous ambiguities are those which are given only one reading. To reiterate, nocuous ambiguity can be acknowledged and/or unacknowledged ambiguity: it may cause misunderstandings because we cannot be sure that all people will read it in the same way. Innocuous ambiguities will not cause misunderstandings as they are generally read in the same way. Referring back to Figure 3-1, both nocuous and innocuous ambiguities are *multiple structure* ambiguities. In the case of innocuous ambiguities, only one structure allows for a reading that is commonly used. In the case of nocuous ambiguities, both structures are likely. Judging the nocuousness of an ambiguity is a simple binary decision: if it is not nocuous it is innocuous.

3.4.1 Theoretical Linguistics Perspective

From a theoretical linguistics perspective, the difference between the multiple reading aspect of our model and the multiple structure aspect can be regarded as the difference between dealing with *performance* data as opposed to *linguistic* data (Abney 1996b). It is suggested that performance accounts for whatever the grammar does not account for (Abney 1996b): the former is concerned with language processing and perception rather than language structure. This is a useful distinction for us as we base our distinction between nocuous and innocuous ambiguity on human perception, rather than on prescriptive rules.

3.4.2 Perspectives from Other Researchers

Poesio (1996) approaches the idea of needing to distinguish between ambiguities which are dangerous and those which are not. He states that "in general, all systems that engage in conversations with their users need to be able to recognise an ambiguity, to ask for clarifications when necessary rather than guess one possible interpretation, and to make their own output ambiguous". The "when necessary" in this statement could refer to situations when ambiguities are nocuous, concurring with our belief that people should be notified of these ambiguities, and that they should not be automatically assigned an interpretation.

Of all previous research, van Deemter's (2004) notion of vicious ambiguity is the closest to our notion of nocuous ambiguity. However, although his notion is empirical — it is determined using corpus data — he offers no empirical test of it. Also, it is intended for determining the status of ambiguities on an individual basis. This means that, while van Deemter is determining whether one surface form is significantly preferable over others, he is not providing validation of that significance by comparing it with other ambiguities. We, on the other hand, use human perception as our metric to determine nocuousness, and investigate empirically whether corpus data can be used to predict this successfully. Also, we evaluate how nocuous ambiguities are compared with other ambiguities, providing a realistic measure of their relative danger.

Spurious ambiguities, as defined by Park and Cho (2000), would be innocuous as they are "semantically unambiguous". The remaining *structural ambiguities* may be nocuous, but this is in no way certain. Such a classification is therefore not a substitute for the one we present here.

Systematic ambiguity, as used by Hindle and Rooth (1993), will tend to result in disagreement about which structure is preferred. This is because all structures give the same interpretation. Such ambiguities are therefore nocuous according to our model. However, no misunderstanding will result and so their nocuousness is trivial. This is an interesting situation, but as it does not lead to misunderstandings it is not of concern to us in this thesis.

3.5 Using Human Judgements as Criteria

Our model of ambiguity relies on human decision-making, instead of any computational approach, in key ways. Firstly, we use human judgement to evaluate ambiguities. Secondly, we allow human intervention to specify how frequently an ambiguity must have multiple readings in order for it to be considered nocuous. This gives control over the sensitivity of discrimination — the intolerance to ambiguity — and makes this discrimination adaptive to different situations. Thirdly, we assign to humans the task of disambiguating the nocuous ambiguities. This aspect does not form part of this dissertation, as our concern is to identify the ambiguities liable to cause misunderstandings. However, we provide motivation and justification for taking this approach.

3.5.1 Judging Ambiguity

We place human perception of ambiguity at the heart of our model: it is our criterion for judging the ambiguity of any given linguistic construction. We take the view that ambiguity is always a product of the meanings that people assign to language (Wasow, Perfors, and Beaver 2003), and so is a fluid and subjective phenomenon. Our reliance on human perception as our criterion ensures that our evaluation of ambiguity is pragmatic. Abney (1996b) reminds us that the non-structural aspects of language include a good deal that is not computationally tractable. In line with this, our policy is to obtain human judgements about ambiguity without eliciting the complicated reasons for these judgements.

We ask human judges for their interpretation of a linguistic construction. How we carry this out is explained in Section 5.3. Many interpretations for the same reading indicate that that reading is strongly preferred. Alternatively, if the judges assign different readings to a construction, this indicates a degree of unacknowledged ambiguity. We also ask the judges whether or not they believe that the construction is ambiguous. Such judgements indicate a degree of acknowledged ambiguity. Whether the degrees of unacknowledged and acknowledged ambiguity are sufficient to indicate that the ambiguity is nocuous depends on our intolerance to unacknowledged and acknowledged ambiguity. This is discussed in the next section.

3.5.2 Ambiguity Intolerance

Clearly, a key issue is to decide where the dividing line between nocuous and innocuous ambiguity should lie. This matters in two respects. Firstly, from the perspective of validating our model, we have choices to make about the distribution of judgements that will lead to an expression being deemed a nocuous or innocuous ambiguity. We will investigate three different ways of deciding which ranges of judgements associated with an expression will be associated with expressions being classified as nocuous or innocuous. These three methods will be described in Section 5.4. Secondly, from the perspective of an application detecting nocuous ambiguities, it is important to realise that different application areas have different tolerance levels to ambiguity. Also, for experimental purposes, we want to investigate how our heuristics perform at different tolerance levels. This we will implement in the shape of ambiguity thresholds, also described in Section 5.4. To our knowledge, no previous research with the same purpose as ours has ever been implemented such an idea.

There is a trade off between detecting as much ambiguity as possible, and minimising the effort used in this process. We discuss below some possible scenarios which demonstrate why this should not be a fixed trade off for all applications, and which therefore motivate the need for a flexible tolerance to ambiguity.

High Intolerance

Many text communication situations might have a high intolerance to ambiguity, requiring that ambiguities be passed as innocuous only if it is almost absolutely certain that they will not be misinterpreted. Examples of this are medical notes, instructions for precision instruments and other documents describing safety-critical systems. Problems arising from ambiguity are exacerbated when the language skills of the authors or readers of the documents are in doubt (Berry, Kamsties, and Krieger 2003), which also motivates the need for a high tolerance to ambiguity.

Low Intolerance

At the other extreme, a lower intolerance to ambiguity would be appropriate for documents which concern less critical matters and which are written and read by people who are all proficient in the language used. Ambiguity in these documents would not be such a danger. Firstly, any resultant misunderstandings would not have such grave consequences, due to the subject matter of the text; secondly, they are written and read by people who share the same knowledge and experience of the language. Abney (1996b) reminds us that NLP ambiguity detection techniques can alert the user to many more ambiguities than are actually perceived by humans, and similar remarks have been made about ambiguity detection in an RE context (Kamsties, Berry, and Paech 2001). This demonstrates that low ambiguity intolerance can be more appropriate than might at first be appreciated.

3.5.3 Human Disambiguation

Many NLP researchers have worked to resolve ambiguities on behalf of users who wish to understand, generate or translate text. However, few have considered that the most advantageous approach to the problem is to let the computer and the human participants do what each one does best. Computers are good at recall and people are good at precision (Kilgarriff 2003b). We therefore let computers find the nocuous ambiguities, and let humans decide how they should be interpreted (and rewritten).

Such a hybrid approach has been used as an adjunct to automatic disambiguation (Yamaguchi et al. 1998), and frequently in the field of machine translation (Mitamura 1999; Blanchon, Loken-Kim, and Morimoto 1995; Boitet and Tomokiyo 1996). In speech translation, it has often been considered appropriate to leave humans to disambiguate the residual ambiguities that automated processes *cannot* disambiguate, under the assumption that humans can do this correctly (Seligman 1997). This may be true for most cases, but it does not necessarily use human intervention appropriately. Borderline cases may be incorrectly disambiguated by the automated process, some humans are unreliable judges, and a lack of adjustability of the automated disambiguation process can make it unsuitable for some applications. In contrast, we ask humans to disambiguate ambiguities which have a stated probability of being misunderstood, instead of simply those which we cannot disambiguate. We therefore believe that our approach offers greater sensitivity to human error and the flexibility needed for a wide range of applications.

3.6 Summary

In this chapter the concepts that form the basis of our approach to ambiguity have been discussed. The ambiguity model, which represents the conceptual architecture of our approach, has been introduced and explained. We have discussed each layer of this model. These layers relate to the structure, interpretation and acknowledgement of ambiguity. We have motivated and discussed our reliance on human perception as the criterion for judging ambiguity, and the ways in which we use human intervention in some aspects of our approach. As part of this, we have introduced our key idea of an adjustable intolerance to ambiguity that distinguishes nocuous from innocuous ambiguity. The work of other researchers has been referenced where it is relevant to our discussions. In particular, we have discussed how our model of ambiguity is different from those of other researchers, and we have argued it is more appropriate for a range of applications.

Chapter 4

Coordination Ambiguity

In this chapter we first make a detailed analysis of the type of ambiguity we use as our test case. This involves discussing coordination ambiguity generally and then focusing on the manifestation of it that we use as our test case. Secondly, we discuss the ways in which this type of ambiguity can become nocuous or innocuous. Some of these ways are generic to all types of ambiguity, while others are specific to coordination ambiguity.

4.1 Our Test Case Ambiguity

Here we discuss coordination ambiguity and how it is manifested. We begin by introducing the general characteristics of this type of ambiguity and the terminology we use to describe it. Then we present the criteria we use to define the sub-type of coordination ambiguity that we use as our test case. This is followed by a discussion of our reasons for choosing coordination ambiguity. We then describe coordination ambiguity more fully in terms of lexical, semantic, syntactic and pragmatic factors. The discussions of these factors explain more fully the scope and manifestation of coordination ambiguity. It also indicates the aspects of it we deal with in this thesis and those that we do not.

4.1.1 Introduction to Coordination Ambiguity

Coordination ambiguities are structural (i.e. syntactic) ambiguities in that alternative readings result from the different ways that sequences of words can be grammatically structured. Coordination ambiguity can occur whenever coordinating conjunctions are used. Examples of these are *and*, *or*, *as well as*, etc, though other methods of coordinating are also possible in English. We use the term *coordinating conjunction* in preference to others, such as *coordinators* (Quirk et al. 1985) and *connectives* (Langendoen 1998), which are preferred by some specialists but are not in such common usage. In English, many types of linguistic units can be coordinated: words, phrases, clauses, sentences, and even sub-lexical morphemes. We use the widely used term *conjuncts*, e.g. (Reibel and Schane 1969), to refer to these linguistic units. This is in preference to the rarer albeit more specific *conjoins*, favoured by grammarians (Quirk et al. 1985).

4.1.2 Our Test Case Criteria

This section is partly a reiteration of the introduction to coordination ambiguity given in 1.3. We explain here the criteria that we use to select comparable examples of coordination ambiguity. Coordinations can be highly combinatorial in English, allowing for many possible syntactic structures. We therefore limit our study to detection of nocuous ambiguity arising from one specific type of coordination. For our test case ambiguity we consider only one coordination at a time. We therefore never consider more than two conjuncts for any example of coordination ambiguity. Our test case ambiguity also contains only one *external modifier*. This may apply to both conjuncts or just to the one to which it is adjacent, and may appear before or after the conjuncts.

For instance, let us consider the following phrase:

Assumptions and dependencies that are of importance

Assumptions and *dependencies* are the conjuncts, *and* is the coordinating conjunction, and *that are of importance* is the external modifier. The type of ambiguity that concerns us, to the exclusion of all others, concerns whether *that are of importance* attaches to *Assumptions and dependencies* or just to *dependencies*. We refer to the former case as *coordination-first*, and to the latter case as *coordination-last*.

4.1.3 Motivation

Structural ambiguities are known to be often more difficult and time consuming to process than other ambiguities (MacKay 1966), and therefore worthy of attention. Coordination ambiguity is one of the three major sources of structural ambiguity, together with prepositional phrase attachment and noun compounding (Nakov and Hearst 2005), but it has received less attention than these other two types in the NLP literature. This is despite of recognition that coordinations are known to be a "pernicious source of structural ambiguity in English" (Resnik 1999), and that parsing them is a very hard task (Kilgarriff 2003a).

To our knowledge there has been no previous empirical research on coordination ambiguity which pursues the same goals we do, but some examples from other studies give further motivation for our choice of test case. For instance, in one of Hirschberg and Litman's (1993) well known experiments, disambiguating cue phrases between discourse or sentential signifiers, conjunctions were found to be considerably more ambiguous than "nonconjunctions". They report 86.3% agreement compared with 97.2% agreement, respectively, between their two judges. Although their results are from one single source, a transcribed keynote address, they contain a comparatively large amounts of data — nearly half of their 953 cue phrases are coordinating conjunctions, indicating the prevalence of coordination ambiguity, at least in this type of corpus.

From an RE perspective, the presence of coordinations has been cited as a clear indication of potential ambiguity (Sawyer, Rayson, and Cosh 2005). In a discussion of guidelines for ambiguous writing style, Kamsties (2001) lists the coordinating conjunctions *and* and *or* as two words particularly guilty of causing ambiguity. However, to our knowledge, coordination ambiguity has not been subjected to systematic analysis by any other RE researchers. As a side effect of our study into nocuous and innocuous ambiguity we can therefore add to the RE literature by investigating it fully and indicating the situations in which it most dangerous.

Coordination has been recognised as a potential source of problems in fields outside the sphere of computing where ambiguity is a key issue. In the legal sector, for example, verdicts in murder cases have been known to hinge on the interpretation of coordinations (Solan 1993).

Using coordination ambiguity as a test case also gives several benefits for analysis in that it gives opportunities for investigating ideas about similarity and parallelism. We wish to make a contribution to the literature by extending the notion, suggested by Okumura and Muraki (1994), that parallel factors in conjuncts, particularly in English, are cues for preferred readings of coordination ambiguities.

4.1.4 Lexical Aspect

The most widely used coordinating conjunctions that have the most versatility at coordinating different types of conjuncts are the *central coordinators* (Quirk et al. 1985) *and* and *or*. The other very common coordinating conjunction, *but*, denotes contrast and is not quite a central coordinator (Quirk et al. 1985). Because of this difference, we do look at coordinations using *but*, considering them not to produce ambiguities that are compatible with those indicated by central coordinators. We do not consider phrasal coordinators, such as *as well as*, having approximately the same effect as central coordinating conjunctions: these are not common enough to warrant the effort required to capture examples of each different phrasing.

Together, *and* and *or* account for approximately 3% of the words in the British National Corpus[1] (BNC). We confine our investigations to *and*, *or* and (by extension) *and/or*.

4.1.5 Syntactic Aspect

Conjuncts of all syntactic types can be coordinated in English (Okumura and Muraki 1994). The external modifier can also be a word or phrase of almost any type, and it can appear before or after the coordination. We explain this further using the example from our dataset introduced earlier:

> *Assumptions and dependencies that are of importance*

The external modifier *that are of importance* applies either to both the *assumptions* and the *dependencies* or to just the *dependencies*. We refer to the former case as *coordination-first*, and to the latter as *coordination-last* because of the order in which the words are connected[2].

We concentrate only on coordinations of this type, where exactly two syntactic structures are possible. However, a third possibility can also be considered with constructions such as this. In this interpretation, usually known as *all-way* coordination (Rus, Moldovan, and Bolohan 2002; Nakov and Hearst 2005), both the conjuncts and any modifying or attaching words are considered to be a unit which means something distinct. This is demonstrated in the following examples from Rus, Moldovan, and Bolohan (2002) and Nakov and Hearst (2005), respectively:

[1] http://www.natcorp.ox.ac.uk

[2] Other terminology can be used, e.g. *low attachment* and *high attachment*, depending on where the coordinated phrase furthest from the modifier attaches in the parse tree (Goldberg 1999); and *ellipsis* and *no ellipsis*, depending on whether the modifier has been elided from the phrase it might form with the conjunct it is furthest from (Nakov and Hearst 2005). Coordinations, at least when in the form of *named entities*, can be classed as *simple* and *coordinated* depending on the status of the conjunct with the elided or not elided attachment (Rus, Moldovan, and Bolohan 2002). However, we feel that our terminology is better suited to our task.

Health and Human Services

Securities and Exchange Commission

We appreciate that such terms taken in their entirety refer to something specific. Nevertheless, they can still be broken down structurally. The *Commission* deals with both *Securities* and *Exchange*, and so it is given a coordination-first reading. The fact that such a term is a commonly used idiom in fact makes this an innocuous ambiguity because it is nearly always read in this way. Rather than consider all-way coordination to be a separate interpretation of such an example, we allow for the fact that it is always read coordination-first to indicate that it is innocuous. We therefore dismiss all-way interpretations from our analysis.

4.1.6 Semantic Aspect

Coordination tends to link things of equal rank and importance (Quirk et al. 1985). Also, intuition and experience tell us it is more common to join semantically similar things in syntactic relationships (Jurafsky and Martin 2000). Munn (1993) expands on these ideas of semantic parallelism, and claims that coordinations are more similar to plurals than was thought by linguistics up to that time. This suggests that coordination-first readings will often be possible, as the processing of the coordination will take place early. The external modifier will then apply equally to both conjuncts. Kilgarriff (2003a) develops this idea by suggesting that a coordination first reading will be more likely if the conjuncts are distributionally similar, and argues that distributional similarity can be used as a substitute for semantic similarity in such analysis. This is exemplified by the following examples:

Old boots and shoes

Old boots and apples

Boots and *shoes* are more similar than *boots* and *apples* are. Coordination of the former pair is therefore more likely to be processed before the modifier takes effect than coordination of the latter pair is.

Of course, the semantics of the modifier, when seen in relation to the semantics of the conjuncts, will have considerable effect on how a coordination is interpreted. These relationships can be captured using lexical affinity models of various kinds (Terra 2004), or with simpler distributional analysis. e.g.(Rus, Moldovan, and Bolohan 2002; Nakov and Hearst 2005).

The central coordinating conjunctions *and* and *or* have a special property whereby one can take on the meaning of the other. NLP researchers looking at coordination have recognised that this to be an issue (Agarwal and Boggess 1992) (Nakov and Hearst 2005). Also, when negations are present together with coordinating conjunctions, application of De Morgan's laws will swap the meanings of *and* and *or.* Such switches of meaning may influence whether an ambiguity is nocuous or innocuous. However, as our policy is not to capture the overall meaning of the coordinations we analyse, we do not expect such phenomena to be accounted for in our studies.

4.1.7 Pragmatic Aspects

Pragmatics, in relation to our work, covers the many aspects of context that affect how an ambiguity is read and whether it is nocuous or innocuous. Detailed contextual knowledge is not available for our purposes. For instance, we cannot analyse all the text surrounding an ambiguity, and we cannot determine the cultural background of the author and intended readers of the text. These aspects are discussed in greater depth in Section 4.2.

Type of Coordinate Compound	Example
Full Syndetons	*Blood and sweat and tears of joy*
Partial Syndetons	*Blood, sweat and tears of joy*
Asyndetons	*Blood, sweat, tears of joy*

Table 4.1: Classification of Multiple Coordinations

4.1.8 Multiple Coordination

To reiterate, our test case ambiguities only include one coordination, realised using a central coordinating conjunction. But, single coordinations are contained within multiple coordinations: the latter can be seen as nested constructions of the former (Langendoen 1998). It will be useful to us if we can obtain, from multiple coordinations, single coordinations conforming to our test case criteria. This will only be acceptable, however, if they represent a semantic subset of the original multiple coordinations. We discuss here the types of multiple coordination we can use in this way. We also describe the other types which are not suitable.

Multiple (or *chained*) coordinations occur, with different types of surface realisation. These can be classified as shown in Table 4.1, following Langendoen (1995). We consider that only *full syndetons*, incorporating a central coordinating conjunction, can produce coordinations conforming to our test case criteria. This is because *partial syndetons* and *asyndetons* use *listing commas*. These punctuation marks are commonly considered to be replacable by central coordinating conjunctions (Trask 1997). However, their *usage* differs from that of conjunctions in full syndetons (Quirk et al. 1985). They tend to form a list which is more of a discrete entity than one represented by a full syndeton. It is therefore less likely that sub-parts of that list can be modified individually. Therefore we do not consider it appropriate to derive single coordinations from *partial syndetons* or *asyndetons*.

We consider *blood of joy*, *sweat of joy* and *tears of joy* to be possible semantic entities resulting from the full syndeton in Table 4.1. When describing our implementation in Section 5.2.2, we explain how we reduce chained coordinations to single coordinations to account for these situations.

4.2 Factors Influencing Nocuousness

Here we present a classification of the reasons why ambiguities are interpreted in different ways, leading to them being classed as nocuous or innocuous when such interpretations are not made universally. This analysis in some cases highlights the problem of distinguishing nocuous from innocuous ambiguities, further motivating the need for an adjustable ambiguity threshold. In other cases it indicates how the problem is minimised by certain linguistic factors. We discuss firstly the difficulty of distinguishing nocuous from innocuous ambiguities, then how we have arrived at our method of classification. Then we present the classification itself.

Ambiguities may be nocuous for a multitude of reasons, which are not always easy to ascertain. Many of these are to do with elusive differences in meaning, and personal and unpredictable differences between author and reader. On the other hand, it can be easy to ascertain that ambiguities are innocuous when semantics and other factors often make only one reading likely. Therefore, the simplest way to locate nocuous ambiguities, or at least a subset of the ambiguities in a text that contains all the nocuous ones, is simply to say that they are what remains when all ambiguities judged to be innocuous have been removed. This is the approach we take in most of this section: most of the factors we describe make ambiguities innocuous, while later we look at some contextual factors from the standpoint that they make ambiguities nocuous.

4.2.1 Introduction to the Classification of Factors Influencing Nocuousness

Taking a sentence containing an ambiguity as the unit under consideration, the ambiguity may be found to be innocuous based on what is contained within that sentence, and/or based on reference to its context. The contents of the sentence — leaving aside the structure, which has been dealt with in Section 3.2 — can be said to comprise the semantics of the individual words and the semantics of the phrases contained within the sentence. The latter, where they are not a product of the former, can be said to be a question of *idiom* (Davidson 1996). The context of the sentence, loosely termed the *pragmatics*, comprises the text around the sentence and factors specific to the individual readers and writers. These latter factors, which we call *reader-specific factors*, include language ability, the historical time at which the text is written or read, and factors which are sociological or psychological in essence.

Background knowledge is the knowledge presupposed by the text, and it can apply to either the contents or the context of a sentence. We will consider only its effect on the contents, as considering its contextual effect would involve repetition and the context is largely unknown to us anyway in our language model. Background knowledge can be said to be linguistic (*word-knowledge*) or extra-linguistic (*world-knowledge*) (Navarretta 1994), though the two types are interrelated. We say that the former comprises knowledge about a word and the roles it can fulfill, while the latter is knowledge about how it tends to be used in the real world.

Another way of subdivided Background knowledge is into common-sense knowledge and domain-specific knowledge (Navarretta 1994). While one type is not necessarily more useful than the other for distinguishing between nocuous and innocuous ambiguities, using one where the other is expected can indeed result in costly misunderstandings

(Cushing 1994). For instance, a serious accident at John Wayne Orange County Airport in California occurred in 1981 when the word *hold* was misinterpreted in the following dialogue:

> Captain: *Can we hold, ask him if we can — hold*
>
> Air Traffic Control: *Air Cal* nine thirty one *if you can just go ahead and hold —.*

The air traffic controller was using *hold* according to domain-specific technical aviation parlance, meaning “stop what you are now doing and thus to go around in a landing situation”. The captain however had momentary lapsed into everyday American English usage, where *hold* means “continue what you are now doing and thus to land” (Cushing 1994). Situations such as this, where speakers or writers slip from one accepted pattern of linguistic usage into another, is known as *code switching*.

Based on the discussions above, we group the factors that can make an ambiguity innocuous or nocuous into the, sometimes interrelated, categorisations which we present below. These categorisations are grouped together according to the standard ambiguity classification of whether they are broadly semantic, syntactic, structural or pragmatic. We add a prosody classification to this. We illustrate our discussion with examples of the type of ambiguity that we use as our test case. We wish to provide a categorisation of coordination ambiguity that is parallel to, though more thorough than, that provided by Hindle and Rooth (1993) for prepositional phrase attachment ambiguity. Some of the aspects of nocuousness that we present are specific to our chosen type of ambiguity. By concentrating on this one type we do not wish to suggest that we are covering the factors that influence the nocuousness of all types of ambiguity, but that by covering the factors that relate to one type the scale of the subject can be appreciated.

4.2.2 Semantic Factors

Here we discuss the semantic factors that influence whether an ambiguity is nocuous. In our classification these include Word-Knowledge, World-Knowledge, Idiom, Semantic Parallelism, and a special case that applies to coordinations — Non-Coordination.

Word-Knowledge

Knowledge about individual words, leading to innocuous ambiguity, might take the form of semantic certainty that the pairing of one word and another word which potentially modifies it could not possibly denote anything in the real world (Jurafsky and Martin 2000). We can explain this using the phrase:

> *The mounted horsemen and footsoldiers*

It can be argued that *footsoldiers* cannot, by their very nature, be mounted, and the phrase *mounted footsoldiers* would be an oxymoron. Because that interpretation is not acceptable, at least not in an exacting environment such as RE, the ambiguity is innocuous.

Repetition of words and phrases is a clear indication of innocuous ambiguity. This can be demonstrated using the sentence:

> *We stock timber and timber products*

The coordination-first interpretation contains a coordination of identical words. That such constructions give clearly preferred readings was considered significant enough by Rus, Moldovan, and Bolohan (2002) for them to implement a heuristic capturing this.

World-Knowledge

The chief reason that most ambiguities are innocuous is that readers (and authors) supplement the semantics contained in the ambiguous construction with their knowledge of the world (Ioannidis and Lashkari 1994). Because world knowledge is to a great extent common, this process is approximately the same for all those involved, and the same conclusions are therefore reached about which reading is obviously preferred over all others. The other readings might be highly unlikely, or semantically impossible, given common sense knowledge of the world.

This concept can be demonstrated using the sentence:

Today I bought cutlery and towels for the bathroom

World-knowledge tells us that cutlery generally have no place in the bathroom, so *for the bathroom* only applies to *towels* and a coordination-last reading is obviously the preferred one.

The presence of *pleonasms* has a similar effect on the understanding of a sentence as does the repetition of words. An interpretation that repeats knowledge will be dispreferred as it is wasteful of human time and effort. It is therefore an indication of innocuous ambiguity, though not as clear an indication as repetition as it requires slightly more mental processing. This can be demonstrated using the sentence:

We carried away the immobilised wounded and corpses

Applying *immobilised* to *corpses* would be pointless and so this is a dispreferred reading. Even if both readings are entertained however, pleonastic expressions do not, by their very nature, tend to cause misinterpretations because they are merely reiterating knowledge.

Idiom

Some readings of ambiguous passages of text may be perfectly acceptable, given knowledge of the meaning and usage of the individual words contained within them, but are innocuous due to unidiomatic combination. This can be demonstrated using the sentence:

Please supply us with your phone numbers and addresses

It is highly likely that *phone* only applies to *numbers*. *Phone addresses* could in theory refer to, for instance, international dialing codes or IP addresses. But scanning the Internet for this phrase reveals that it is not (comparatively) a common idiomatic expression for anything in the real world. The original phrase containing the coordination is therefore an innocuous ambiguity.

Idiom tends to be a weaker criterion than word- or world-knowledge, which are more purely derived from semantics. Changing usage of language, particularly in a rapidly evolving field such as telecommunications, could easily make *phone addresses* an acceptable phrase at some time in the future.

Idiom can also play a positive part in establishing one out of a selection of realistic alternatives as the most likely one. This results in innocuous ambiguity if it is sufficiently likely. For instance, let us consider the sentence:

We are looking at research and development costs

Research and development is such a well-known phrase that most people would prefer the coordination-first reading.

More extreme and clear-cut examples of this phenomenon occur when a rare or unlikely word is used: such words may be used in very few contexts. This is exemplified by the sentence:

I will do it with all my might and main

The word *main*, meaning *strength*, has virtually no modern usage except idiomatically in coordination with *might*. (This usage, whereby one word in a coordination is not found on its own, is known as *Siamese twins* (Fowler and Gowers 1965)). Therefore, *my* applies to both *might* and *main*, and the ambiguity is innocuous.

Non-Coordination

A special case of the idiomatic usage criterion occurs when constructions containing coordinating conjunctions are not in fact true coordinations. These can be detected because they violate Ross's (1967) *Coordinate Structure Constraint*, which states that:

> *In a coordinate structure, no conjunct may be moved, nor may any element contained in a conjunct be moved out of that conjunct.*

Certain constructions exist which include coordinating conjunctions but contravene the clauses of the coordinate structure constraint. Examples include the following two types of sentence:

> *She has gone and ruined her dress now*
>
> *I've got to try and find that screw which is very small*

These form grammatical sentences when the coordinate structure constraint is contravened, implying that they are not true coordinations:

> *Which dress has she gone and ruined now?*
>
> *That screw which I've got to try and find is very small*

It can be said that the word *gone* in the first sentence is being used as an adverb, whereas *try and find* in the second sentence is a type of *hendiadys* (Fowler and Gowers 1965) and really means *try to find*. Although, syntactically, there is a choice about where to attach the modifiers *now* and *that screw which is very small*, semantically this is not the case. The supposed conjuncts must be treated as a semantic unit, a coordination first reading is therefore preferred, and the ambiguity is innocuous.

4.2.3 Syntactic and Structural Factors

Although we have addressed the structure of our ambiguities in our single and multiple structure phase, extracting only those from our corpus with a specific number of possible structures, ambiguities can be innocuous due to certain of these structures being dispreferred.

Several theories, which we will term *syntactic preference principles*, have been proposed to explain why certain readings of a coordination might be preferred based purely on syntactic factors. These are particularly attractive strategies as they do not require any investigation in the complex and fine-grained semantic and discourse domains (Hindle and Rooth 1993). Two of these theories are of particular relevance to us: the *late closure strategy* and the *minimal attachment*. Frazier and Fodor (1978) make the point that the psycholinguistics literature presents abundant evidence that humans use such syntax-only approaches to make attachment decisions before the semantics have been considered. If one of the syntactic preference principles can be said to apply conclusively to a coordination, — the semantics do not provide contrary disambiguating evidence — then that coordination will be innocuous.

We first discuss the two syntactic preference principles mentioned above, preceded by the generally applicable *immediacy principle* and followed by an evaluation. Then we identify two other structural criteria for dispreferred readings: differing subcategorisation and syntactic parallelism.

Immediacy Principle

The immediacy principle (Just and Carpenter 1980) is a general principle which states that people decide where a word should fit in a parse tree immediately upon encountering it. It accounts for *garden path* sentences (Carroll 1999) such as the following:

The horse raced past the barn fell

We tend to interpret this sentence incorrectly because we try to process each word as we encounter it without looking ahead at the remainder of the sentence. Because such garden path sentences are very clear examples for most readers, we use this example, and the different ways in which it can processed, to explain processing theories in the following subsections.

Late Closure Strategy

The late closure strategy states that "incoming items are preferentially analysed as a constitute of the phrase or clause currently being processed" (Frazier 1985). This means that, when a reader is processing a sentence, new words are more readily considered to be part of the phrase currently being processed as opposed to being part of a new phrase that needs to be constructed.

Clear evidence that humans do use the late closure strategy, at least some of the time, is provided by garden paths. The sentence introduced previously can be used to explain this:

The horse raced past the barn fell

We try initially to incorporate *fell* into the phrase currently being processed, *the barn*, which we assume to be the direct object of the sentence. However, we then realise that in fact *fell* is the main verb of the sentence and that all the words preceding it must be reprocessed as the subject.

The similar *last antecedent rule* is a principle sometimes invoked in courts of law. It simply states that a post-modifying clause only applies to the last antecedent (Solan 1993). This could be interpreted as the last (and nearest) conjunct in the case of coordinations.

Minimal Attachment

Minimal attachment is a principle for syntactic processing proposed by Frazier and Fodor (1978). It states that, when constructing parse trees, the number of nodes, and therefore the branching that links them, is to be minimised. Therefore, avoiding constructing any new nodes is key, and high attachment of any phrase or lexical item into the phrasal representation of a sentence is favoured. Frazier and Fodor claim that this principle is very generally applicable and accounts for several specific strategies that inform attachment decisions.

Garden path sentences can also be used to show the minimal attachment principle in operation:

The horse raced past the barn fell

Backtracking is required, as with the late closure strategy, but for different reasons. In this case, the minimal attachment readily allows attachment of *fell* to the topmost node once *raced past the barn* is established to attach to *horse.* However, the principle dictates that*raced* "is more readily interpreted as an active intransitive verb in the main clause than as the passive participle of a transitive verb in a relative clause modifying *horse*" (Frazier and Fodor 1978). This is because of the reduced branching in the former analysis, caused by high attachment of the verb *raced* and avoidance of creating a complex new node for the subject in the latter analysis.

It should noted that the minimal attachment principle, as a parsing technique, was originally promoted on the grounds that it was memory efficient as well as predictive. But Frazier (1978) has reported "experimental evidence for the operation of minimal attachment in a variety of different constructions" — reported by Frazier and Fodor (1978).

Summary and Significance of Syntactic Preference Factors

The significance of these simple, syntax only methods has been demonstrated in several domains where correct interpretation of language is crucial. For instance, *syntactic misdirections*, such as may be caused by inappropriate application of the late closure strategy or minimal attachment, are recognised as a problem in air traffic communications (Cushing 1994). In the legal domain these heuristics can conflict and cause confusion even when applied by learnéd people. The outcomes of court cases have hinged upon whether the judge decides that one of these heuristics should be applied. This sometimes this can result in "unintuative" judgements (Solan 1993), suggesting that a better understanding and synthesis of syntax-only methods would be beneficial. It can be seen that these principles can easily conflict with each other. Experiments, for example by Taraban and McClelland (1988), have shown that such syntax-only approaches are not a good way of predicting preferred readings of ambiguities.

Differing Subcategorisation

Some structures are awkward due to the fact that the coordination is of words which are dissimilar in the roles they play, even though they have the same basic part of speech. For instance, coordinations of words with different subcategorisations can sound unlikely, meaning that alternative readings which don't require different subcategorisations of those words are preferred. The following sentence demonstrates this:

> *Type and save the data*

The verb *type* may be either transitive or intransitive, depending on the reading. The exhortation to *type* sounds unlikely, with no object, when contrasted with the *save* operation which has *the data* in that role. This point can probably be best appreciated by realising how rarely coordinations of verbs with different subcategorisations, with the coordination-last reading necessarily preferred, become idiomatic. Some following sentences are examples that do exist:

Go and fetch your coat

Run and catch the bus

Syntactic Parallelism

Structural (or *syntactic*) *parallelism* is known to have an effect on how sentences are interpreted, and it has been cited as a particular aspect of coordinations that has a bearing on their interpretation (Okumura and Muraki 1994) (Dubey, Sturt, and Keller 2005).

Frazier, Munn, and Clifton (2000) perform some experiments that test whether structural parallelism has the significance that is predicted. One experiment investigates the role of syntactic parallelism in the processing of conjuncts with and without parallel internal structure. Four variations of sentences with a simple binary coordination were used, like the following:

a) *William made friends with a talkative salesman and a foreign executive while waiting at the airport*

b) *William made friends with a salesman and a foreign executive while waiting at the airport*

c) *William made friends with a talkative salesman and an executive from France while waiting at the airport*

d) *William made friends with a salesman and an executive from France while waiting at the airport*

The overall conclusion of this experiment was that it is easier to read conjuncts that have the same internal structure than to read ones with different internal structures. Looking at the sentences above, it takes less time to process those of type a) than those of type b), and those of type d) than those of type c). This can be taken as an indication that the latter cases are more nocuous than the former cases: either more readings are being considered or they are being considered more seriously. However, the results are not conclusive, as syntactic parallelism was also shown to reduce the processing time of non-ambiguous sentences. Additionally, and perhaps surprisingly, little difference in assignment of modifiers — *from France* in the above examples — was noted between the conjuncts that were syntactically parallel and those that were not.

4.2.4 Pragmatic Factors

Several factors drawn from the context of an ambiguity can have a considerable effect on whether it is nocuous or innocuous. Berry, Kamsties, and Krieger (2003) note that, in a requirements context, readers can be so unaware of correct language usage that "a sentence can be unambiguous from a linguistic point of view, but be ambiguous from a pragmatic point of view". We identify three groupings which are useful to us in research: textual context, reader-specific factors and re-specific context.

Textual Context

Of course, context can play a major role in making an ambiguity innocuous. Indeed, some researchers, for example (King et al. 2000), seem to make the assumption that all humans need is sufficient context in order to disambiguate language correctly. However, others explicitly state that this is not realistic. For example, van Deemter's (1998) Principle of Idiosyncratic Interpretation stresses that ambiguity is not guaranteed to disappear even when all linguistic and non-linguistic context is taken into account. This can be seen most clearly in short and isolated statements, such as the phrase "lifetime guarantee" found on many commercial products. This phrase has caused confusion about whether the guarantee is for the lifetime of the product or the lifetime of the user (Wasow, Perfors, and Beaver 2003), and the nature of the product and the environment in which it is sold is unlikely to provide conclusive disambiguating information. Extrapolating from this, it is not possible to say for certain that any amount of context will give absolute immunity against ambiguity. We therefore adhere to van Deemter's (1998) Principle of Idiosyncratic Interpretation, and take the view that there may always be differences of opinion regarding the interpretation of any ambiguity. With regard to context, every ambiguity is therefore potentially nocuous. Also, from a practical point of view, consideration of large amounts of context is very difficult to accomplish computationally, and even humans can forget relevant information they have been told.

Reader-Specific Factors

Some factors that influence whether ambiguities are nocuous or nocuous are part of the context of a sentence containing an ambiguity but not part of the text that surrounds it. These include the language abilities and mindsets of the authors and readers of the texts, and the historical times at which the text is written and read. The impact of these factors, which are generally concerned with differences between author and reader, has been discussed in Section 3.3 as they are greatly responsible for non-acknowledgement of ambiguity.

Insufficient proficiency in the language used to write text can easily cause either author or reader to make a serious misjudgement, showing evidence of nocuous ambiguity. Indeed, even difference in dialects of the same language can cause author and reader to misunderstand one another. Let us consider the following example:

Stop while the red light is flashing

This could, in theory at least, be misunderstood by speakers of some dialects of English in which *while* means *until*.

Cushing (1994) summarises the discussions in linguistics literature about the importance of individual cognitive factors and social interactive factors in communication. The former include mental models of the world, belief systems and so on, and all the judgements and expectations that flow from these. The latter are concerned with interaction between communicators; they include aspects such as conventions of language usage, official protocols and the relative status of the communicators. Appreciation by author and reader of the need to make allowances for individual cognitive factors, and to harmonise social interactive factors, will reduce the danger of nocuous ambiguity. Cushing points out that the two different types of factor should also match each other if serious consequences are to be avoided.

4.2.5 Prosody

Prosody is an aspect of language that is more usually studied with regard to poetry and to speech. However, in the sense that it refers to the rhythm of language, it also has repercussions for written text. Rhythms in text are often the direct result of punctuation, but they can also occur where no punctuation is present but where readers feel that a natural pause is appropriate.

Schepman and Rodway (2000) examine the effect of prosodic boundaries on coordination disambiguation, using a similar test case to ours. Although their work is on spoken language, aspects of prosody such as numbers of syllables and words can also affect the way written language is interpreted due to the human capacity of processing only a limited number of linguistic units at one time. Punctuation may also be regarded as an aspect of prosody, but our test case criteria allow for no punctuation marks in our examples of coordination.

4.3 Summary

In this chapter we have discussed coordination ambiguity in detail and then given many reasons for why it can be nocuous or innocuous. We have discussed coordination ambiguity in general terms, and have provided motivation for choosing this type of ambiguity. We then focused upon the specific manifestation of it that we use as our test case. This discussion looked at lexical, syntactic, semantic, pragmatic and combinatorial aspects. We then set out many reasons why coordination ambiguity can be nocuous or innocuous, based on semantic, syntactic, pragmatic and prosodic criteria.

Chapter 5

Implementation

In this chapter we describe how we implement our ideas about ambiguity analysis and classification. This involves describing the *life cycle*, from our perspective, of the ambiguities that we consider. We locate them in appropriate texts, ensure that they conform to our experimental criteria, discover whether and how humans disambiguate them, determine from this whether they are nocuous or innocuous, and then try to predict this automatically using heuristics.

5.1 Building a Corpus of Requirements

To our knowledge, there are no existing corpora of requirements documents that we could use to analyse ambiguity in requirements. Taking our examples of ambiguity from an existing generic corpus would not be suitable for our purposes. We wish to look at ambiguities that are representative of those found in actual requirements, and that might cause actual problems. We therefore use a corpus from actual requirements documents.

We explain here the advantages we obtained from building this corpus, and how the documents that constitute it were selected. Then we describe how we mark up the words in the corpus with their parts of speech. This is necessary in order to ensure that any ambiguities we locate conform to our test case criteria. Then we present our corpus. Lastly we discuss some positive and negative aspects of our corpus and the process used to build it.

5.1.1 Selection of Documents

The requirements documents in our corpus are obtained from RE practitioners and from the public domain. Size and content matter are the most important criteria for determining whether we include a requirements document in our corpus. Also, we prefer not to select documents which predominantly use tables and other structured formats. These tend to contain isolated words and short, sub-sentential linguistic constructions. Our test case ambiguity, on the other hand, is generally found in running text.

Size

Including very small documents in our corpus would not be suitable. These are more likely than others to have been written as part of a trivial (or informal) specification exercise. On the other hand, we do not choose overly large documents. These would bias the corpus in favour of the terminology and writing style that they contain. Within these intuitively determined limits, we select documents of varying sizes.

Content

We wish the documents in our corpus to be representative of a variety of application domains. This will avoid bias towards any highly specialised usage of terminology only used in certain domains. Domain is a vital factor when considering RE-specific ambiguity in requirements (Kamsties, Berry, and Paech 2001). Context has a great influence on these, and domain-specific aspects of ambiguities are revealed in context. These aspects provide vital disambiguating information that is specific to what is being described. On the other hand, our test case ambiguity is more purely linguistic, rather than RE-specific. Therefore, although the examples of it we capture will use language common to RE, they will not necessarily require overly large consideration of domain-specific context. We therefore use RE as *our* application domain, and treat all application domains described by requirements homogeneously.

Criteria Not Used

In the process of selecting documents, we do not consider the quality of the writing to be a criterion. We wish our corpus to be a reflection of actual writing used in requirements for non-trivial exercises. We therefore require that it includes all the ambiguities and errors that these requirements contain.

We do not discriminate regarding the geographical or cultural origins of the documents we consider. We prefer to capture the actual unbiased diversity of these factors. Organisations engaged in writing requirements in English may be located in any country. Also, within an organisation, the stages of the software life-cycle are sometimes distributed between groups in different locations. Requirements may therefore be written by people with different cultural background to the other stakeholders. This diversity in the requirements process may lead to ambiguity, and we wish to observe that.

5.1.2 Building and Tagging the Corpus

To build our corpus we add requirements documents to it iteratively. Each one of these must be prepared to conform to the corpus format, then each word in it must be tagged with its part of speech. The former task is achieved using text manipulation programs that we have developed. These ensure that the text is properly tokenised and that each line of text is a sentence. Sub-sentential passages contained in titles, table entries, bullet points, etc, are also retained. We refer to all such passages of text as “sentences”.

Document	Application Domain	System Described	No. of Words	No. of Sentences	% of corpus (by sentence)
#1	HCI	Graphics system	1,739	137	5.7
#2	HCI	Appointments scheduler	1,860	100	4.1
#3	Healthcare	Information system	1,583	196	8.1
#4	Software engineering	Software compiler	6,492	497	20.5
#5	Governmental	Voting software	5,805	547	22.6
#6	Telecommunications	Mobile networking	7,036	374	15.5
#7	Software engineering	Requirements validation	8,564	568	23.5
		Total	33,079	2419	100
		Average	4,725	345.6	14.3

Table 5.1: Characteristics of the texts in our corpus

When a new document is added to the corpus, each word is tagged with its part of speech. We use Brill's (1992) rule-based tagger for this. Prior to each addition, the tagger is trained on the tagged text already in the corpus. Each tagging exercise creates new rules in the tagger, giving inferences about how unseen text should be tagged. After each addition, we check manually that the words have been tagged correctly. This ensures both that our corpus is accurately tagged and that the tagger has accurate training data for the next iteration. By training the tagger on our corpus, we achieve greater accuracy at tagging terms found commonly in requirements. This saves time in the long term.

5.1.3 The Corpus

Table 5.1 shows the characteristics of the texts that make up our corpus. For each text, these characteristics include its size in terms of words and sentences, the percentage of the corpus that it comprises, the type of system it specifies, and the application domain in which this type of system is from.

The documents in our corpus describe the design of (and therefore contain requirements for) a wide range of systems for use in various industries. This heterogeneity gives us a broad perspective on the use of language in requirements documents. It goes some way to ensuring that our techniques do not become overfitted to a specific domain. The sizes of the documents in our corpus are varied, but fulfill the criteria that we stated in Section 5.1.1. None of them represent trivial requirements specification exercises, and none overly bias the corpus with their terminology.

5.1.4 Advantages and Disadvantages of Our Corpus

In addition to being specific to RE, our custom-made corpus gives several other advantages. These include coverage of up-to-date terminology and writing which demonstrates a realistic range of language proficiencies. These points are important as we wish to ensure that the techniques we develop are suitable for typical authors working in RE today. These advantages are significant for all types of writing, but are more than usually so for RE. Requirements describe projects from software and systems engineering, and these industries have a more fluid and rapidly evolving terminology than many others. Also, language proficiency is an issue as these industries often employ a multi-national workforce, sometimes working in different countries.

Set against these advantages is the fact that building such a corpus is a fairly laborious and time-consuming task. This is mainly because all the words in the corpus need to be tokenised and accurately tagged with their parts of speech. This is done to ensure that we only locate ambiguities conforming to our test case criteria. Also, it allows for detailed analysis of the types of ambiguity that we locate. However, we use the tagging process to our advantage by training our tagger specifically on our corpus. This results in greater accuracy and analytical possibilities. The extra time spent developing the corpus is deemed to be worthwhile considering the sensitivity to ambiguity required in our domain.

5.2 Obtaining Examples of Our Test Case Ambiguity

In this section we explain how we obtain examples of our test case ambiguity from our corpus of requirements. (A new file is created of these, with all words still tagged with their parts of speech. This forms the basis of our dataset.) We first explain here our need to focus on a narrowly-defined type of ambiguity as our test case. We then introduce how we obtain examples of this type.

We wish to have as much data as possible to use on one narrowly defined problem. This maximises the reliability of our results by focusing our body of data on the same issues. It also simplifies the task of our judges — the people we ask to give their perceptions about our examples of ambiguity — who can repeatedly apply themselves to the same judging task. We therefore want our data to present ambiguities with a uniform number of possible syntactic interpretations. We choose to consider only ambiguities with exactly two possible syntactic interpretations.

To reiterate, we wish to use examples of coordination ambiguity of the form:

> *[Assumptions and [dependencies]]* <u>*that are of importance*</u>

Each example must contain just one coordination, and there must be just one modifying element whose attachment is ambiguous. From here onwards the examples we present have the same typographical notation that we used to help the judges understand the task. The modifiers are underlined. The two possible phrases to which a modifier applies — representing either coordination-first or coordination-last interpretations — are indicated with square brackets.

There are several test case criteria that we use to ensure that the examples we obtain conform to this pattern. The first of these is simply that the coordinating conjunctions *and*, *or* or *and/or* are used. We must then ensure that our examples have more than one alternative *structure* — i.e. parse tree — as opposed to more than one alternative *reading*. We describe in detail the set of criteria that we have developed in order to perform this discrimination. Then we explain how we deal with examples that have too many alternative structures for our purposes. Lastly, we describe the *flexible chunker* we have developed to help locate the coordinations we use as our examples.

5.2.1 Eliminating Single Structure Cases

We discard any test case examples we have located in our corpus that have only one possible structure. The criteria we have developed to do this are explained below. Examples of these criteria, and explanations of those examples, are presented in Table 5.2. We established these criteria to distinguish between the structure and interpretation levels of our model of ambiguity. Not all of them were required when we processed the sentences located in our corpus.

Some of the criteria described here are near the borderline between distinguishing ambiguities on the grounds of structure and on the grounds of perception. Indeed, some of the structural criteria we present here have been used as disambiguation heuristics, for instance by Nakov and Hearst (2005) as discussed in 2.1.2. Employing such criteria as heuristics might increase our ability at locating innocuous ambiguities, as they would locate ambiguities that are nearly always given one particular reading. However, our concern is nocuous ambiguity, so we prefer to ignore as many such ambiguities as possible on the grounds that their structure is obvious. In this way, the ambiguities that we consider contain a higher concentration of nocuous ambiguities. Because of this, we may sacrifice performance when we come to distinguishing nocuous from innocuous ambiguity using heuristics. However, it enables us to focus more on the differences of human perception that make an ambiguity nocuous.

No.	Reason for Exclusion	Example	Explanation
1	One of the conjuncts can't stand alone syntactically	[hot or [cold]] water can be used	Only 1 syntactic reading: coordination-first.
2	No external modifying element	[Dogs and [bats]] carry rabies	Coordination-last readings result in coordination of dissimilar types of phrase (& possible ungrammaticality): must be coordination-first.
3	Premodification, and 2nd conjunct beginning with determiner	I ate green [[beans] and the sausages]	A modifier cannot premodify a determiner (except if the former is a *predeterminer*, e.g. all, both, half): coordination-last.
4	Premodification, and 2nd conjunct begins with pronoun	I like tall [[women] and her over there]	A modifier cannot usually premodify a pronoun: coordination-last.
5	Coordinated nouns have different number and followed by present tense verb	I wear [boots and [a raincoat]] that are waterproof	Subject of 3rd person plural verb must be plural: can only be coordination-first.
6	Bracketings, and other types of parenthetical punctuation indicating an interruption	I like green [[beans] (and sausages)]	Such punctuation devices signify asides, which are not affected by external modifiers: coordination-last (Borderline cases exist however).
7	Commas indicating end of a phrase or list item	Trade, [[unions] and disputes] were discussed	A supposed modifier shown instead to be a discrete item: must be coordination-first; (Other comma usages less clear though).
8	Acronyms that abbreviate preceding words	natural language processing [[(NLP)] and synthesis]	Words before conjunction clearly constitute a semantic unit: can only be coordination-last.
9	Capitalisation of words preceding conjunction	Adaptive Frequency [[Modelling] and transmission]	Modifier is clearly part of a name and therefore part of a semantic unit: must be coordination-last.
10a	Hyphens indicating compound words	[Self-motivation and [-orientation]] workshop	Hyphen prefixing 2nd conjunct indicates it forms a compound with 1st part of 1st conjunct: must be coordination-first.
10b	”	[bottle- or [breast-feeding]] of babies	Hyphen suffixing 1st conjunct indicates that it forms a compound with 1st part of 1st conjunct: must be coordination-first.
11	Phrases with head words with dissimilar parts of speech are coordinated	Dnyanesh is most famously [[a bowler] and very quick]	A modifier can be syntactically adjacent to words of different parts of speech, but the similarity of these words cannot easily be compared.
12	A coordinated head word is a company or proprietary name	We evaluated [Acme Widgets and [Jones Tappets]] products	Must be kept anonymous. Substituting proper names with dummy words would introduce false data and skew the results with repeated use.

Table 5.2: Criteria Used to Eliminate Coordinations from the Dataset

Syntactic Criteria

Criteria 1 to 5 are essentially syntactic in nature, and their validity can be demonstrated by looking at a well-developed set of grammar rules. These criteria are, with very few exceptions, clear indicators of single structure cases.

Punctuation Criteria

Punctuation plays a significant part in determining the structure of written language. The two criteria we present here cover the most common ways in which punctuation can indicate single structure in our sentences. Full stops, colons and bullet points all signify the *end* of sentences in our corpus, so we do not require criteria to take account of them.

Criterion 6 covers several forms of parenthetical punctuation: dashes, parentheses and bracketings of all types. These generally indicate that an *aside* is being made. They tend to be discrete semantic units and so modification does not tend to cross the boundary represented by a parenthetical symbol. This is shown in the following sentence:

The car was [washed — and [polished]] vigorously — before the race

In such cases, parenthetical punctuation demands coordination-last interpretations, indicating single structure. However, counter examples exist, particularly when the modifier is outside the parenthetical insertion. The following sentences indicate that coordination-first interpretations are perfectly possible:

The green [[hills] — and valleys —] of my native land

He had the reputation of being a true [[gentleman] (and scholar)]

Criterion 7 reflects the fact that commas can often, but not always, be clear indicators of syntactic structure. We follow Trask's (1997) classification of comma usage into four categories. Some indicate single structure coordination ambiguity, while others are not so clear.

- *Listing commas* list items. This can be demonstrated by adapting the example from Table 4.1 on page 72:

  Blood, sweat and tears

  This usage signifies that a supposed modifier, in this case *Blood*, is in fact another conjunct. Criterion 2 is thereby infringed, signifying single structure.

- *Joining commas* join sentences or clauses together. They must be followed by either *and*, *or*, *but*, *yet*, or *while*, such as in the following sentence:

  I said goodbye, and the train departed.

  Therefore there tends to be no external modifier, and criterion 2 is infringed. Counter examples are possible, for instance when a sentence adverb is a modifier:

  <u>Allegedly</u> [[the story is true] , and it cost him his career.]

  Except in these exceptions, joining commas signify single structure.

- *Gapping commas* indicate elision of certain words; for instance the word *was* in:

  Gavaskar was <u>probably</u> [[the greatest Indian batsman] and Chandrasekhar, the worst.]

  The (uncommon) gapping comma generally appears to enforce a coordination-first reading. We consider therefore that gapping commas signify single structure.

- *Bracketing commas* signify asides and other parenthetical insertions. They behave similarly to the punctuation considered in criterion 6. To understand this, bracketing commas can be substituted for the parenthetical symbols in our discussion of that criterion. Similarly, bracketing commas sometimes demand a coordination-last interpretation, signifying single structure, but by no means always.

Punctuation, taken as a whole, is a useful, though not foolproof, indicator of single structure.

Orthographic Criteria

Orthography is concerned with spelling, punctuation, capitalisation and the symbols that can be used in a language. Our orthographic criteria are concerned with capitalisation, and with miscellaneous non-alphabetic symbols which clarify understanding but which are not actually classed as punctuation marks. Our capitalisation criteria, 8 and 9, can be understood by reference to the concept of *named entities*. These are semantic units representing organisations, people, places, and so forth, which often involve capitalisation. Aside from these, most symbols do not affect interpretation of coordination ambiguities. We believe that hyphens operating as infixes are the only miscellaneous symbols which require our attention. This usage is covered by criterion 10.

Criterion 8 is based on the convention that bracketed and entirely capitalised "words" are generally acronyms which refer to an immediately preceding named entity. If the post-modifying "modifier" is the acronym and the named entity it abbreviates is the conjunct immediately preceding it, they will be structurally conjoined to each other. The only possible structure is the one that gives a coordination-last reading. This is illustrated in the example from Table 4.1, where *natural language processing* naturally attaches only to *NLP*:

We perform <u>natural language processing</u> [[(NLP)] and synthesis]

Criterion 9 is based on the convention that words with initial capital letters generally refer to named entities. If both the "modifier" and the conjunct nearest to it have initial capital letters, then together they usually form a named entity. (Capitalisation in titles and at the beginning of sentences can, of course, confuse the issue.) This is indicated by the example from Table 4.1:

> *…accomplished using Adaptive Frequency [[Modelling] and transmission]*

Criterion 10 covers cases where hyphens are used to indicate compounded words. Coordinations can be made of such words. Two conjuncts may be hyphenated compound words, with the common element elided from one of them. This is shown in the example from Table 4.1, where *feeding* is the common element but it has been elided from the first conjunct:

> *Is [bottle- or [breast-feeding]] of babies preferable?*

The lexical unit *bottle-* cannot stand alone. Because of the (otherwise unexplainable) *dangling* hyphen, only a coordination-first interpretation is possible, indicating single structure.

Pragmatic Criteria

We reluctantly use some criteria to eliminate coordinations from our dataset for pragmatic reasons, rather than because they signify single structure coordinations. Because our heuristics largely use notions of similarity to form their predictions, it must be possible to make comparisons between the important words in the coordinations.

Criterion 11 eliminates coordinations of phrases which have head words with dissimilar parts of speech. Such coordinations are possible with some parts of speech. These constructions are never particularly common, but the example from Table 5.2 shows they can both exist and be ambiguous:

> *Dnyanesh is most famously [[a bowler] and very quick in the field]*

We are forced to implement this as, for instance, a noun and an adjective like bowler and quick cannot be directly compared. Indirect, for example semantic, comparison may be made, but that is beyond the scope of this thesis.

Criterion 12 exists because we cannot ask for people's perceptions about ambiguities involving proprietary names: some of the requirements documents in our corpus were given to us in confidence. We could replace these names with generic nouns, as is sometimes done in information retrieval. But usage of generic nouns in this way has at least two unwelcome effects. Firstly, choosing a noun to represent the proprietary name introduces one person's judgement, which may be unreliable. Secondly, repeated use of the same generic nouns begin to make an unrealistically large contribution to the data, which will then bias the results.

5.2.2 Accounting for Multiple Coordination Ambiguity

We use only examples of coordination ambiguity with two possible structures. The majority of multiple structure coordination ambiguities that we locate in our corpus have only two possible structures. However, it is sometimes possible to obtain examples of coordination ambiguity with two structures from those with more than two. This involves duplicating the original sentence and simplifying each new version to reduce the number of possible structures. This can only be done, however, when no important disambiguating information is lost from the original. There are two types of multiple coordination ambiguity from which we can obtain examples conforming to our test case criteria. These are due to either multiple modifiers or multiple coordinators.

Multiple Modifiers

When more than one modifier is present together with the coordination, this can result in a coordination with more possible structures than we require. Let us consider the phrase:

Design Review and Checking System

Design could apply to both *Review* and *Checking System* (which might be procedures implemented by an organisation), and *System* can be modified by *Review* and *Checking* (which might be its functions). (Note that in this example while it is required to include the modifier *Checking* in the former scenario, as this word is nested within the coordination, we can omit *Design* when considering the latter scenario. This is because *Design* is a modifier that occurs outside the coordination, it is not vital for understanding the phrase, and it might cause the judges some confusion.)

Our solution to this scenario is to include multiple variations of the sentence in our dataset. The following two sentences are created from the original one to represent the two alternative coordination ambiguities:

Design [[Review] and Checking System]

[Review and [Checking]] System

Note that the coordination-last readings of these phrases are actually the same reading: both indicate that *Design Review* and *Checking System* are discrete semantic units. The original phrase has three possible syntactic structures which can now be analysed as two instances of two structures.

Multiple Coordinators

We can use instances of multiple coordination where it is realised by *full syndetons*, as explained in Section 4.1.8. An external modifier must also be present, such that none of our criteria for eliminating single structure cases are infringed. We use the example from Table 4.1 on page 72 to illustrate our way of using multiple coordinations:

Sweat and blood and tears of joy

Ideally, we would like to discover which of the three conjuncts — *Sweat*, *blood* and *tears* — can be modified with *of joy*. So we create the following two phrases from the original:

> *blood and tears of joy*
>
> *Sweat and tears of joy*

The former of these is simply a semantic possibility in the original. The latter captures, in the form of our test case ambiguity, the possibility of the furthest conjunct from the modifier being modified.

5.2.3 Flexible Chunker

We need to accurately locate examples of coordination ambiguity that conform to our test case criteria in our corpus. This can be slow and painstaking if done by hand. We therefore designed and developed our own text manipulation software to assist with this task. We term this a *flexible chunker*: it has the output of a chunker, but the flexibility of a non-deterministic parser.

Firstly we explain what chunking is and why it might be suitable for our work. Secondly we discuss some other implementations of chunking for the purpose of analysing coordinations. Then we explain the workings of our chunker, and conclude by evaluating its performance and its potential. We believe that our software goes considerably beyond the scope and specification of the earlier implementations of the idea. However, our implementation revealed some of the difficulties inherent in the task we were addressing. This meant that our software was not able to fully realise our aims. For this reason we describe some aspects of our chunker in theoretical terms only.

Chunking

Chunkers group words of running text into *chunks*. These can be noun phrases, verb phrases and other such coarse-grained units. Although originally intended to capture prosodic structure (Abney 1995), chunkers have been increasingly used as a preprocessing step for parsing (Sang and Buchholz 2000). Traditionally, chunkers have not coped well with coordinations, and generally avoid processing them (Abney 1996a) (Sang and Buchholz 2000). However, to address coordination ambiguity, the conjuncts of any coordination must be discerned somehow. A parser must of course make choices at these junctures. However, it would be useful to avoid the complications of full parsing when focusing only on one linguistic construction. A chunker might therefore provide a quicker and simpler solution. Also, coordinations have the special characteristic of parallelism between the phrases that are its conjuncts (Okumura and Muraki 1994). A chunker has the advantage that it readily assembles phrases, and can be built so that it looks for this parallelism.

Previous Use of Chunkers for Coordinations

Despite the aforementioned reluctance of many chunker designers to address coordinations, we are not the first to realise that it might be useful. Goldberg (1999) uses a “simple chunker” to obtain training data containing coordination ambiguities. This operates on tagged text, and uses “two small regular expressions” to return the head words of noun and quantifier phrases. However, it is not explained what these expressions are nor how they achieve this. It is also not clear how efficient the chunker is at this task. Agarwal and Boggess (1992) use a custom-built “semi-parser” to prepare their text for coordination analysis. This may be a chunker in all but name. This includes conjunct identification and prepositional phrase attachment amongst its capabilities. However, Agarwal and Boggess do not describe the capabilities or coverage of their software.

Our Chunker

Our chunker is designed to operate on the text of our corpus, tagged using Brill's (1992) tagset. As with other chunkers it groups words together into phrases. However, rather than being purely deterministic, it can output several alternative sequences of chunks. Each sequence represents an alternative reading of the coordination ambiguity being addressed. The output, taken as a whole, is analogous to a forest of parse trees produced by a parser with some underspecification in its grammar. It is this aspect which makes the chunker flexible: it allows for the many different possible combinations of words that can be coordinated. This facility helps us identify examples of coordination having only one syntactic structure, which can then be eliminated from our dataset. (It can also, in theory at least, identify coordination ambiguities having too many syntactic structures.) It simultaneously ensures that the words we are considering as the triggers for coordination ambiguity — *and*, *or* and *and/or* — are present in the sentences.

To process sentences containing coordinations, our chunker has both basic and advanced features. The former conglomerate words into basic phrases. The latter produce the different possible syntactic structures, and their constituent phrases, of the sentences using these basic phrases. Both sets of features mainly use the part of speech tags of words to decide how the phrases should be formed. The chunker appends new tags to all the phrases it creates. These new tags constitute a small tagset of our own formulation, as is common practice with other chunkers (Abney 1996a) (Sang and Buchholz 2000).

Conglomeration of words into basic phrases starts by locating the most likely head word of each phrase. This is done by using a prioritised order of word types that are likely to be heads. The process of finding suitable head words is iterative: candidate heads are selected in order of decreasing priority. We then look for pre-modifying and post-modifying words that can be joined to each head word. When it is considered whether a word can modify a head word, it is judged according to a *bag of tags* criterion. This means that it must have a tag from the set of tags representing the types of word that can modify the head word. (A set of these tags is maintained for each word type that can be a head word.) For example, an adjective can pre-modify a noun, an adverb can pre- or post-modify a verb or pre-modify an adjective, a possessive inflection can post-modify a noun phrase, and so on. This procedure is also performed iteratively. Each phrase is built up around the head word until no more suitable pre-modifying and post-modifying words can be found. The conglomeration process halts when all words are accounted for.

The advanced features of our chunker iteratively conglomerate the phrases formed by the basic features into ever larger units. This requires some basic functions akin to parsing. These include finding verb phrases for subject, finding objects for verb phrases, applying prepositional phrases to verb phrases and noun phrases, compounding nouns, and so on. This is where coordination also becomes an issue. The goal of this exercise is to generate all the possible structures for any given coordination. This demonstrates the number of permutations of conjuncts and attendant modifying phrases. Coordinations with only one syntactic structure can then be discarded and those with exactly two structures can be included in the dataset. Those with more than two can be discarded or evolved into several sentences containing just two structures. A side effect of this process is the capturing of the head words of the conjuncts and modifying phrases. This information can be used to categorise and analyse the coordination alternatives.

Performance and Evaluation

Our flexible chunker does not output perfect results. The work with the advanced features in particular proved to be more difficult than anticipated. We therefore simply use the chunker to weed out some sentences which are not of interest to us. We do not use the advanced features to their full extent. This ensures that we do not exclude sentences that we wish to include in our dataset. This is at the expense of including some sentences that fail the criteria and which we have to eliminate later by hand. We checked that using the chunker in this way gives the results we anticipated. We chunked a section of the corpus and then carefully examined it by hand.

The basic aspects of our chunker perform well. The prioritised head word selection and bag of tags techniques are simple but effective. Our chunker appears to be unusual (for one operating on English) in that it allows post- as well as pre-modification of head words (Sang and Buchholz 2000). We believe that this is a useful and viable approach. Developing this software has also given us insights into the fecundity of coordinations as a source of ambiguity in real life contexts.

5.3 Obtaining Human Judgements about Ambiguity

Here we explain how we use ambiguity questionnaires to obtain human judgements on the examples of coordination ambiguity we have taken from our corpus. This involves simplifying the sentences where appropriate. We then present them in the form of questionnaires to a group of human judges. After describing these procedures, we explain in detail the characteristics of this group of judges. This detail is important: our model of ambiguity requires that we look closely at the formation of human perceptions.

5.3.1 Preparing Example Sentences

It is important that judging the coordination ambiguity examples is not too onerous a task. The judges are unpaid volunteers with little available time. They should be able to complete the questionnaires without more effort than is essential for treating the task seriously. We therefore endeavour to present the examples of ambiguity in a form which fulfills this wish.

If the sentences are very long, we simplify or elide parts of them which are clearly not essential to their interpretation. We omit preambles and trailing clauses for the same reasons. This procedure is also used for hiding proprietary names, where these are not head words and therefore not semantically required for our analysis. Such alterations are indicated graphically using conventional typographic notation: substitution is represented by parentheses, elision by a series of dots. The first lines in the following examples show the text as found in the corpus, the second ones show how it is presented in the questionnaires:

> *1(a): In order to plan the possible upscaling of the system, it will be made highly scalable by making it easy to switch from a 2-tier client/server solution to a 3-tier distributed solution with the use of a middleware like [Proprietary Name] Transaction Server to take care of business rules and transactions*
>
> *1(b): Take care of business [[rules] and transactions]*
>
> *2(a): The [Proprietary Name] will be implemented and executed on the 32-bit [Proprietary Name] platform*
>
> *2(b): (It) will be [implemented and [executed]] on the ...... platform*

These representations of the original sentences, resulting from the aforementioned procedures, are those used in the ambiguity questionnaires.

5.3.2 The Questionnaires

The sentences, prepared as described earlier, are presented in the form of questionnaires to our group of judges. The judges are asked to give their opinion about how each ambiguity should be read. We formulated the instructions for this task during the course of our pilot study. For any given example, such as:

(It) will be [implemented and [executed]] <u>*on the ...... platform*</u>

the judges are asked which of the following three options they think applies to the coordination:

- It is interpreted coordination-first
- It is interpreted coordination-last
- It is ambiguous so that it might lead to misunderstanding

In the last case, the coordination is then classed as an *acknowledged ambiguity* for that judge. We refer to the first two cases as the *non-ambiguous judgements*. Allowing the third "ambiguous" choice is used in other ambiguity studies, for instance by Hirschberg and Litman (1993) when determining whether coordinating conjunctions act as discourse or sentential cue phrases. The full set of instructions that we present to the judges is given, verbatim, in Appendix A.

We made sure that the judges were not given other information of any of the following types which might influence their opinions:

- The judgements and comments of other judges about any particular ambiguity
- What we, the researchers, thought the judgements should be
- The overall popularity of each of the three judgement options

In addition, we gave no indication of the context from the original text of the examples.

5.3.3 Selecting the Judges

The people who judge our ambiguities must be suitably familiar with the problem of ambiguity in English. Also, familiarity with the terminology used in requirements will be an advantage. We must use a sufficient number of them to give confidence in the patterns of perception that we capture in our data. We do not use the same judges for each batch of ambiguities that we present in our questionnaires. However, we do use the same number of them each time. They are taken from a pool of judges that conform to our criteria.

Profile of The Judges

The judges are all working in academia, and have specialist knowledge of computing and other systems-related activities. As a group they are therefore well qualified to judge ambiguity in requirements. (Additionally, some have expertise in requirements engineering and some are also trained linguists.) Their ages range from twenties to forties, their academic status ranges from postgraduate to professor, and they originate from various countries. All are fluent speakers of English, though some are non-native speakers. They therefore bring skills to the task that in line with those of many requirements engineers (and other stakeholders who read requirements).

Sample Size

The *sample size* for this task is the number of people who act as our judges. We need to choose a suitable number. RE is sensitive to ambiguity, and so using many judges is desirable to capture nuances of human perception (Berry, Kamsties, and Krieger 2003). However it is also expensive in terms of effort, so we aim to reach a compromise.

One problem that can be encountered in any data collection process is *noise*. In terms of human judgements, this is constituted by judgements resulting from judges displaying lack of due care and attention. We term these *rogue* judgements. They are not the same as genuine differences in opinion, which result in the nocuous ambiguity that we wish to model. Rogue judgements can have a particularly large influence when intolerance of ambiguity is high. In that situation, one judgement that does not agree with the majority viewpoint might classify a clearly innocuous ambiguity as marginally nocuous. If this judgement was not a genuine and considered perception of that ambiguity, then it has a unwarranted influence on the classification of the ambiguity. Choosing a sample size large enough to minimise the effect of noise is a recognised technique (Keren 1992), and one which we follow.

We look for a sample size that establishes an acceptable basic level of confidence in our data. This basic level of confidence is the lower-bound achieved if the judges give informed — i.e. better than random — judgements. We find this lower-bound using the standard statistical method of postulating, and trying to reject, a *null hypothesis* (Umarji 1962). This hypothesis says that there is no substantial evidence that the judgements are informed; the *alternative hypothesis* says that there is. We wish to determine the possibility that agreement between the majority of the judges is due to uninformed decision making. For us, rejecting the null hypothesis would demonstrate that this possibility is acceptably small. We wish to determine how many judges we should use, given that a number of them are likely to give rogue judgements, so we can reject the null hypothesis. Achieving this for any combination of a given number of judges will prove that we have a basic level of confidence for that sample size.

For this particular exercise we wish to analyse only the non-ambiguous judgements, i.e. judgements of *coordination-first* and *coordination-last*. This is because the third choice, acknowledgement of ambiguity, can be considered as akin to a *don't know* option. It is therefore an answer to a different question than the coordination-first or coordination-last question (Fowler 2001). Removing the "ambiguity" option reduces the problem to one of binary decision making.

Our confidence that the judgements are informed is the probability that the null hypothesis is rejected. In other words it is the probability of proving the null hypothesis subtracted from 1. Let N be the number of experts giving judgements and M be the number of them giving a rogue judgement. The probability of proving the null hypothesis is the probability of finding M or fewer rogue judgements from any of the N judges. Using standard probability theory, our confidence that the judgements are informed is therefore calculated to be:

$$1 - \frac{1}{2^N} \sum_{m=0}^{M} \binom{N}{m}$$

$\frac{1}{2^N}$ is the product of all the (no better than random) 50% success rates for all the judges. The binomial coefficient $\binom{N}{m}$ is the number of ways to choose exactly m out of N, and these are summed for all possible values of m to account for all possible numbers of rogue judgements. The final figure is subtracted from 1 to give the likelihood of rejecting the null hypothesis. The binomial coefficients are calculated as:

$$\binom{N}{m} = \frac{N!}{m!(N-m)!}$$

Table 5.3 shows confidence levels for sample sizes from one to twenty. The numbers of these judges giving rogue judgements range from zero to eight. Initial investigations carried out in a pilot study[1] suggested that approximately a third of all judgements would be an acknowledgement of ambiguity. Therefore, to account for these, we wish to choose from Table 5.3 a sample size that is one and a half times the sample size that best fits our purpose. In our pilot study there were very seldom more than two judges at any time who appeared to give a rogue judgement, so we let $M = 2$. As is common with

[1] This asked judges similar questions about PP-attachment ambiguities using several surveys of different sizes. It is true that PP-attachment ambiguities are not necessarily perceived as being equally ambiguous as coordination ambiguities. However, we are unaware of any comparable coordination ambiguity study that we could use to give this information.

No. of	Number of Rogue Judgements								
Experts	0	1	2	3	4	5	6	7	8
1	50.00	0.00	N/A	N/A	N/A	N/A	N/A	N/A	N/A
2	75.00	25.00	0.00	N/A	N/A	N/A	N/A	N/A	N/A
3	87.50	50.00	12.50	0.00	N/A	N/A	N/A	N/A	N/A
4	93.75	68.75	31.25	6.25	0.00	N/A	N/A	N/A	N/A
5	96.88	81.25	50.00	18.75	3.13	0.00	N/A	N/A	N/A
6	98.44	89.06	65.63	34.38	10.94	1.56	0.00	N/A	N/A
7	99.22	93.75	77.34	50.00	22.66	6.25	0.78	0.00	N/A
8	99.61	96.48	85.55	63.67	36.33	14.45	3.52	0.39	0.00
9	99.80	98.05	91.02	74.61	50.00	25.39	8.98	1.95	0.20
10	99.90	98.93	94.53	82.81	62.30	37.70	17.19	5.47	1.07
11	99.95	99.41	96.73	88.67	72.56	50.00	27.44	11.33	3.27
12	99.98	99.68	98.07	92.70	80.62	61.28	38.72	19.38	7.30
13	99.99	99.83	98.88	95.39	86.66	70.95	50.00	29.05	13.34
14	99.99	99.91	99.35	97.13	91.02	78.80	60.47	39.53	21.20
15	100.00	99.95	99.63	98.24	94.08	84.91	69.64	50.00	30.36
16	100.00	99.97	99.79	98.94	96.16	89.49	77.28	59.82	40.18
17	100.00	99.99	99.88	99.36	97.55	92.83	83.38	68.55	50.00
18	100.00	99.99	99.93	99.62	98.46	95.19	88.11	75.97	59.27
19	100.00	100.00	99.96	99.78	99.04	96.82	91.65	82.04	67.62
20	100.00	100.00	99.98	99.87	99.41	97.93	94.23	86.84	74.83

Table 5.3: Confidence levels with varying numbers of rogue judgements

such exercises we use a confidence level of 95% (Umarji 1962). Table 5.3 shows that a minimum of eleven experts would provide this lower-bound level of confidence. We then add in the projected acknowledged ambiguity judgements, i.e. we multiply by one and a half. Seventeen is then the minimum suitable number of judges, so this is the sample size that we use.

5.4 Distinguishing Nocuous from Innocuous Ambiguity

As a result of our ambiguity questionnaires, we have seventeen judgements on each ambiguity in our dataset; these judgements can be either "coordination-first", "coordination-last" or "ambiguous". This situation is represented spatially in Figure 5-1: it equates to a three-dimensional vector space diagram. The three sides of the triangle represent the three types of judgement, which can be given in varying degrees for any given ambiguity. The point where the dotted lines within the triangle intersect represents a point in the

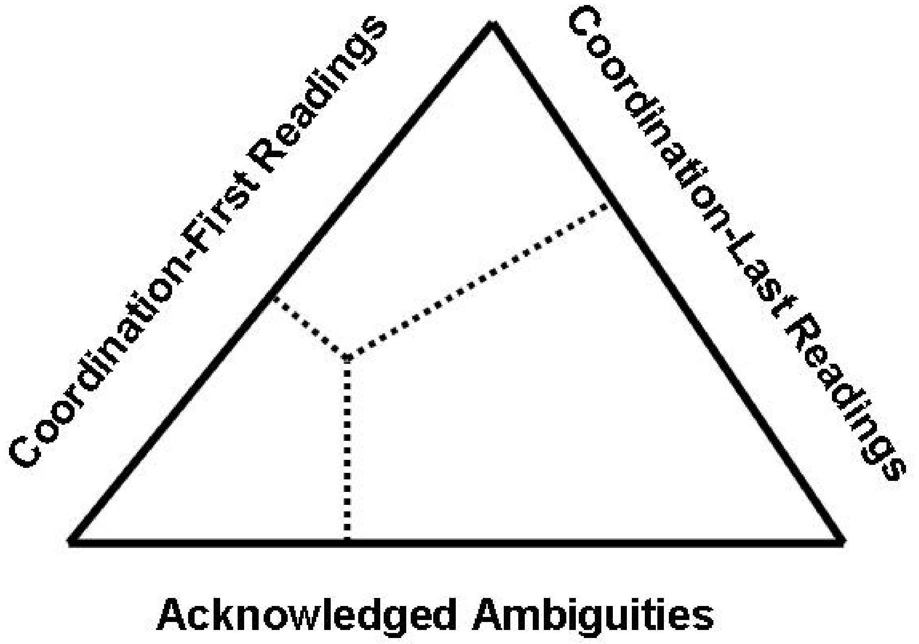


Figure 5-1: Spatial Representation of the Judgements given on an Ambiguity

space where any given ambiguity in our dataset lies. The lengths of each dotted line from the intersection to the solid line boundary of the triangle represents the proportion of judgements represented by that boundary. These lengths can be in any proportion relative to each other, but will always equal the same number.

Now we need to determine, from this situation, whether any given ambiguity is nocuous or innocuous. There are several ways in which this can be done. The use of *ambiguity thresholds* is important in two of our methods, so we begin by introducing these thresholds. We then explain the three methods that we use in this research to distinguish ambiguities that may be dangerous from those which are not.

We present a *weighted* method of distinguishing nocuous from innocuous ambiguity. We have used this successfully when implementing our model in a tool for RE practitioners (Chantree, Nuseibeh, de Roeck, and Willis 2006). In this method, unacknowledged ambiguity is generally considered to be more important than acknowledgement ambiguity as a component of nocuous ambiguity. This is done as unacknowledgement of ambiguity is ultimately what makes ambiguity nocuous in our model. No separately controllable threshold is allowed. However, the method of determining nocuousness does represent a threshold in its own right.

We then present a *flexible* method of distinguishing nocuous from innocuous ambiguity. Here, unacknowledgement and acknowledgement of ambiguity are considered to be of equal importance. Whether or not they result in nocuous ambiguity is determined by use of ambiguity thresholds. This implements our desire to have a flexible tolerance to ambiguity. For example, this flexibility allows a high ambiguity threshold to be set for applications in safety-critical domains. Conversely, less critical applications might require a lower ambiguity threshold.

Thirdly, we analyse solely *unacknowledged* ambiguities. These may be more immediately dangerous than other, acknowledged, nocuous ambiguities: they have actually been found to be understood differently by different people. We wish to determine whether unacknowledged ambiguity can be characterised successfully by a model, and how this compares to our model of nocuous ambiguity.

There are other ways of analysing ambiguity that accord with our stated aims. The flexible method implements the core premise of nocuous ambiguity, that the phenomenon of having multiple readings is the defining criterion. Unacknowledged and acknowledged ambiguity are therefore considered to be of equal importance. Similarly, the method focusing solely on unacknowledged ambiguities is a pure representation of that phenomenon. We would not therefore wish to omit analysis of our data by either of these methods. The weighted method, however, is open to discussion. The criteria for what makes an ambiguity acknowledged or unacknowledged represent thresholds. To some extent these thresholds are intuitive, making an independent estimate of the level of danger inherent in each of the two phenomena that make an ambiguity nocuous. They have, however, been approved as suitable for purpose by colleagues working in requirements engineering.

5.4.1 Ambiguity Thresholds

We use ambiguity thresholds to reflect our notion of a degree of tolerance to ambiguity, for the following reasons. Firstly, not all ambiguities can be disambiguated reliably. Secondly, we intend that in any system implementing our method, users can be given control over the degree of tolerance to nocuous ambiguity. As such, the ambiguity threshold represents the minimum certainty required for an ambiguity to be considered innocuous. The higher the ambiguity threshold is, the more intolerant we are of ambiguity. We explain below how different ambiguity thresholds are set and used. The effects they have on the classification of an ambiguity are explained for each of the methods of distinguishing ambiguity that we use.

Users of any system incorporating our approach to ambiguity must decide which level of ambiguity intolerance is appropriate to the task at hand. They would then set the ambiguity threshold to a level representing that intolerance. All ambiguities found in their texts would then be judged by that criterion. Alternatively, a particular ambiguity threshold could be implemented as a policy decision by an organisation, or an individual author creating text. For instance, high intolerance to ambiguity would be required when creating documents describing a safety-critical system. An organisation would then implement a high ambiguity threshold to avoid any disastrous misunderstandings. Alternatively, an individual might have limited time to write informal specifications — which perhaps will be thoroughly discussed before they are acted upon. Intolerance of ambiguity would in this case be lower. A lower ambiguity threshold would then be set, ensuring that the documents were created quickly.

Ambiguity thresholds are important in our empirical studies. We test our heuristics against a range of them, thereby evaluating their performance at different tolerances to ambiguity. This we will explain in Chapter 6.

Our ambiguity threshold determines whether an ambiguity it is nocuous or innocuous: it is a binary decision. Other classification schemes are possible. For instance, Oberlander and Nowson (2006) demonstrate how classifications representing differing levels of certainty can be used in sentiment analysis. They discuss the advantages of various ways of classifying the strength of textual indicators for any given sentiment. These classifications include simply "high" and "low", having a intermediate "medium" classification, and interposing "relatively high" and "relatively low" between these. We could use any of these schemes, substituting "nocuous" and "innocuous" for "high" and "low". However, our use of an ambiguity threshold replaces the flexibility of having the choice of one of these schemes. The threshold allows the classification of nocuous ambiguity to be as "relative" as required. All that is required for our approach is that the threshold is set judiciously and that that decision is respected. Oberlander and Nowson state that their consideration of increasingly fine-grained classification schemes is simply intended to gradually approximate *continuous rating*. We start from the position of continuous rating, giving us greater flexibility.

5.4.2 Weighted Method

This method states that a coordination ambiguity is nocuous if it is an acknowledged ambiguity or if it is an unacknowledged ambiguity. However, these two components are calculated in a way that tends to give more importance to the latter. A coordination ambiguity is said to be *acknowledged* if (and only if) it is judged ambiguous at least as often as it is judged coordination-first *and* at least as often as it is judged coordination-last. A coordination ambiguity is said to be *unacknowledged* if (and only if) it contains above average unacknowledged ambiguity. This average can be calculated over any number of

For sentences $(1 \ldots i \ldots n)$

CF_i = No. of Coordination First Judgements on Sentence i

CL_i = No. of Coordination Last Judgements on Sentence i

QUA_i = Quotient Unacknowledged Ambiguity for Sentence i

$$= \frac{min(CF_i, CL_i)}{(CF_i + CL_i)} \qquad (= 0 \quad \text{if} \quad CF_i + CL_i = 0)$$

A_i = No. of Ambiguous Judgements on Sentence i

AA_i: Sentence i is judged to contain Acknowledged Ambiguity

IFF $(A_i \geq CF_i)$ AND $(A_i \geq CL_i)$

For sentences $(1 \ldots j \ldots n)$

UA_i: Sentence i is judged to contain Unacknowledged Ambiguity

IFF $\left(QUA_i > \frac{\sum_{j=1}^{n} QUA_j}{n}\right)$

Sentence i is Judged to contain Nocuous Ambiguity

IFF AA_i OR UA_i

Table 5.4: Weighted Method for Determining Nocuous Ambiguity

ambiguities that one wishes to survey. Unacknowledged ambiguity is calculated solely on the numbers of non-ambiguous judgements, i.e. coordination-first and coordination-last; the acknowledged judgements are all ignored. These criteria for what makes an ambiguity classed as acknowledged or unacknowledged by this method represent fixed thresholds.

Determining nocuous ambiguity using the Weighted Method is explained mathematically in Table 5.4. Worked examples are given in Section 6.2 to demonstrate the use of the weighted method.

5.4.3 Flexible Method

When looking at judgements, the *consensus judgement* can be said to be the majority non-ambiguous judgement: in our case, coordination-first or coordination-last. All judgements that dissent from this consensus viewpoint can then be conflated together. Therefore, judging a coordination to be ambiguous is considered equally important as judging it to have the minority non-ambiguous reading. The former represent the acknowledged ambiguity component; the latter represent the unacknowledged ambiguity component. This reduces the three possibilities to two, either agreeing with the consensus judgement or disagreeing with it, as shown in Figure 5-2. Whether there are enough of the latter to

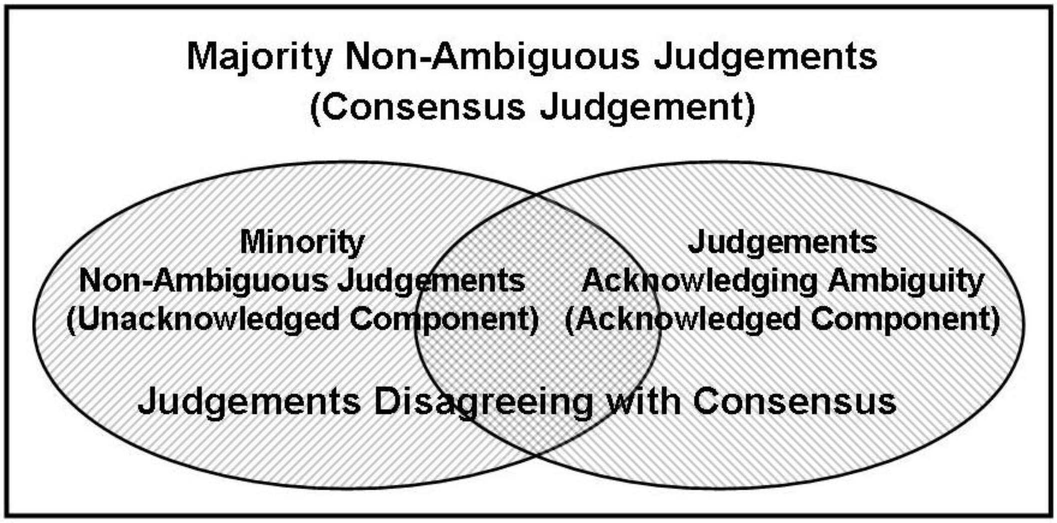


Figure 5-2: Assessment of Judgements using the Flexible Method

classify an ambiguity as nocuous is determined by the ambiguity threshold. The *certainty* of the consensus judgement (or *percentage agreement* (Gale, Church, and Yarowsky 1992)) is simply the percentage of the judgements that agree with it.

We demonstrate here the effect of ambiguity thresholds when used in the Flexible Method. Let us say that a coordination has been judged by 65% of the judges to be coordination-first and by the remainder to be either coordination-last or ambiguous. 65% is therefore the certainty. Then, if the ambiguity threshold is set at 60%, the consensus judgement for that coordination will be considered to be coordination-first. It will therefore be classified as innocuous. If, on the other hand, the ambiguity threshold is set at 70%, the coordination will be classified as nocuous.

5.4.4 Purely Unacknowledged Method

In this method, ambiguities are determined to be dangerous purely in terms of unacknowledgement. All judgements acknowledging ambiguity are simply ignored. Only the minority non-ambiguous judgements — the unacknowledged ambiguity component of the dissenting judgements — count towards categorising an ambiguity as dangerous. This is

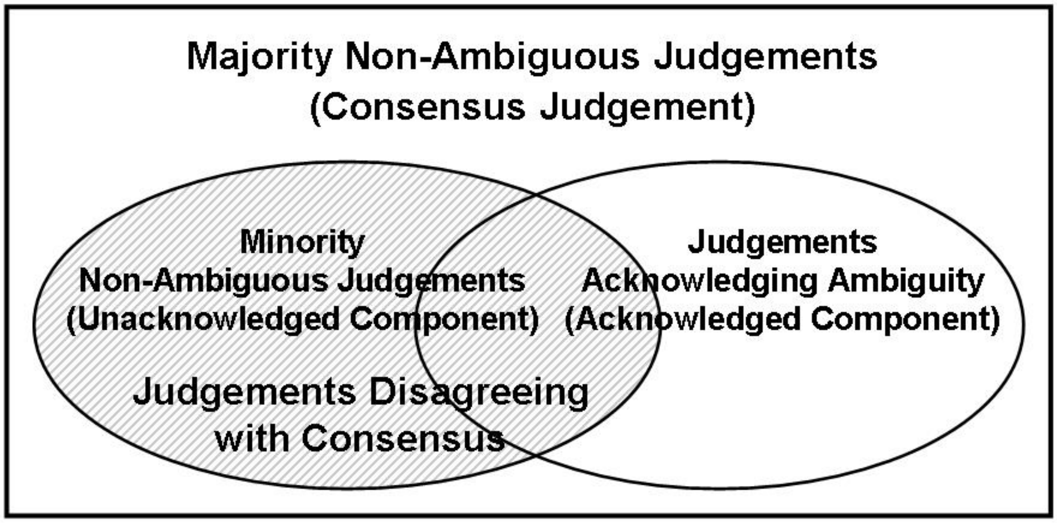


Figure 5-3: Assessment of Judgements using the Unacknowledged Method

shown in Figure 5-3. The degree of unacknowledgement is the number of minority non-ambiguous judgements divided by all the non-ambiguous judgements. We do not use the term *nocuous* here, as we have stated that acknowledged ambiguity is an important factor in that definition. Instead we talk of distinguishing innocuous from *unacknowledged* ambiguity. We use ambiguity thresholds, as per the Flexible Method. These thresholds represent different intolerances to unacknowledgement, and determine whether an ambiguity is judged to be innocuous or unacknowledged.

We demonstrate here the effect of ambiguity thresholds when used to categorise levels of unacknowledged ambiguity. Let us say that a coordination has been judged by 65% of the judges to be coordination-first, by 20% of them to be coordination-last and by 15% to be ambiguous. The degree of unacknowledgement is therefore $20/85 = 23.5\%$, and the certainty is 76.5%. If the ambiguity threshold is set at 70%, the consensus judgement for that coordination will be considered to be coordination-first. It will therefore be classified as innocuous. If, on the other hand, the ambiguity threshold is set at 80%, the coordination will be classified as an unacknowledged ambiguity.

5.5 Tools

Here we present the tools we use to create many of our predictions about ambiguity. The heuristics we have developed to make these predictions are presented in Section 5.6. The aspects of the tools which we present here are used generically by several of these heuristics. The tools are a generic corpus and a statistical package that extracts word distribution information from that corpus. The package we use is the Sketch Engine (Kilgarriff et al. 2004); the generic corpus we use is the BNC. We introduce these tools, after discussing some motivation for using such an approach.

Using word statistical distribution information obtained from a corpus to predict preferred readings of structural ambiguities has become a recognised technique in recent years. For instance, McLauchlan (2004) and Calvo, Gelbukh, and Kilgarriff (2005) use word distribution, presented via a thesaurus, to predict prepositional phrase attachment. Collocations and n-grams are frequently used to predict preferred readings of many types of structural ambiguities, for instance by Nakov and Hearst (2005) and Rus, Moldovan, and Bolohan (2002). In his adaption of bidirectional optimality theory, van Deemter (2004) incorporates corpus-based probabilities to judge which ambiguities are vicious ambiguities.

Our use of statistical corpus-based NLP techniques is in line with some current thinking in RE. For instance, Sawyer, Rayson, and Cosh (2005) believe that such techniques have now reached a level of maturity to be useful in RE support tools for *early phase RE*. Their work recognises how critical ambiguity can be in the early stages of formulating requirements. They advocate several uses of statistical distribution information for addressing this problem.

5.5.1 Sketch Engine

The Sketch Engine is a software package which produces statistical information about the distribution of words in a corpus. It accepts input of lemmatised nouns, verbs and adjectives. We use two of the key facilities offered by Sketch Engine. The first of these is a word sketch facility giving information about the frequency with which words are found collocated with each other. The second is a thesaurus giving distributional similarity between words.

Word Sketch Facility

The word sketch facility generates lists of word collocations. Each collocation corresponds to a syntactic relationship, such as modification by adjective or prepositional phrase. Relationships at a syntactically higher level are also represented, such as those between verbs and their subjects and objects. Sketch Engine finds the correct collocations for a word by use of grammatical patterns (Kilgarriff et al. 2004). This is aided by working on a tagged and annotated corpus. Head words of conjuncts can therefore be found with more certainty than by looking at an arbitrary window of text around a word.

When a lemmatised word is input, a *sketch* of that word is generated. This sketch consists simply of the lists of word collocations. The *salience* score is given for each word in each list. Salience is calculated from the frequency which the listed word is found collocated in the corpus with the word that was input, and the overall frequencies of both words in the corpus: it is "estimated as the product of Mutual Information and log frequency" (Kilgarriff and Tugwell 2001). This salience statistic is well-founded and avoids the common bias towards overly rare words (Kilgarriff et al. 2004). As an example, let us consider a generated list of nouns that are the objects of the verb *kick*. The words *ball*, *penalty*, *heel* and *habit* are the words with the highest frequency of collocations in this syntactic relationship. *Ball* is found 108 times as the object of *kick* in the BNC, and has a salience of 40.79. *Penalty* is found 63 times, but has a salience only slightly lower at 37.68.

Parameters for minimum frequency, minimum salience and maximum number of words in a list can be entered. We use a minimum frequency of one and a minimum salience of zero throughout, to ensure we get results even for unusual words.

Thesaurus

The Sketch Engine's thesaurus measures similarity between any pair of words according to the number of corpus contexts they share. In this regard it is a distributional thesaurus in the tradition of Sparck-Jones (1986) and Grefenstette (1994). The *distributional similarity* that such a thesaurus measures has many useful properties for NLP research. It can be calculated quickly on any corpus that has been tagged appropriately. This means that it is easily used for texts from any domain. Kilgarriff (2003a) points out the advantages that it has over semantic similarity. Thesauruses claiming to capture the latter concept often fail to agree with one another, due to the well-known difficulty of analysing *meaning.* They also often introduce unwanted ambiguity by way of their hierarchical structures or their clustering of words together. Distributional thesauruses, on the other hand, avoid such issues by not requiring these error-prone conceptualisations. A recent study at disambiguating attachment ambiguities indicates that distributional thesauruses perform at least as well semantic thesauruses (Calvo, Gelbukh, and Kilgarriff 2005).

The Sketch Engine's thesaurus calculates distributional similarity in the following manner. The corpus is parsed, and all triples comprising a grammatical relation and two collocates (eg ⟨*object, drink, wine*⟩ or ⟨*modifier, wine, red*⟩), are identified. Contexts are shared where the relation and one collocate remain the same. So, ⟨*object, drink, wine*⟩ and ⟨*object, drink, beer*⟩ count towards the similarity between *wine* and *beer.* Shared collocates are weighted according to the product of their *mutual information*, and the similarity score is the sum of these weights across all shared collocates. This similarity score is the one developed by Lin (1998), based on dependency triples.

5.5.2 British National Corpus

The BNC is a modern, largely text-based, corpus containing over 100 million words of British English, containing more than 700,000 distinct words (Sawyer, Rayson, and Cosh 2005). It is *generic*, collated from a variety of sources, including some that share specialist terminology with our chosen domain. These two factors make information obtained from the corpus more *robust*, in that a large amount of linguistic variation is accounted for. The BNC is also *synchronic*: the non-fiction documents in the BNC are from a defined period, 1975 – 1995. Coverage from this time period means that a higher proportion of computing documents are represented than would be true for coverage from an earlier period.

It theory, the fact that the BNC is mainly British English might present a few problems. It is to be expected that most requirements documents are written using American spelling. However, the spelling differences between British and American English are minor and predictable. Our policy is to look up both spellings of any word in the Sketch Engine, and use the version of English that generates the highest scores.

In theory it would be possible to use a specialised corpus of requirements or of documents relating to computer science and systems engineering. These might provide better coverage of the specialised words and expressions on which we require data. However, such a corpus would need to be very large to avoid sparseness even with common non-specialised words. It would also need to be accurately tagged so that statistical data can be obtained from it. We are not aware of any sufficiently large and publicly available corpora conforming to these criteria.

5.6 The Heuristics

Here we present the heuristics that we use to predict the human judgements about ambiguity that we have gathered. We then briefly discuss other candidate heuristics that were not part of our empirical study. Details of the performance of each heuristic we use in our empirical study will be given in Chapter 6.

Most of our successful heuristics use word distribution information obtained from a generic corpus, the BNC, using the Sketch Engine. Most of the other heuristics base their predictions on information contained within the tagged text itself. When using the Sketch Engine, our heuristics use rankings rather than scores, wherever this is appropriate. In similar work, other researchers have considered rankings can be more appropriate. For instance, McLauchlan (2004) has found this when using thesauruses to disambiguate prepositional phrase attachment ambiguities.

5.6.1 Coordination Matching

This heuristic makes predictions about the preferred interpretations of coordinations by looking for the incidence of those coordinations in a generic corpus.

Hypothesis

Our hypothesis here is that if a coordination in our dataset is found commonly in language, then that coordination is likely to be a syntactic unit and a coordination-first reading is therefore likely. We determine whether it is common in language by finding out if it occurs a significant number of times in a generic corpus. The following sentence can be used to illustrate this:

I always eat fish and chips on Friday

The coordination *fish and chips* occurs commonly in English. We hypothesise that the coordination process will take place first, and the attachment of *on Friday* will happen later. If the phrase *fish and chips* is found frequently enough, the ambiguity will be innocuous.

It is generally not productive to look in a corpus for the exact sequence of words, as sparseness will quickly become a problem. Therefore it is preferable to consider just the head words of the conjuncts in our coordination examples. We look in the corpus for all coordinations of these, ignoring any modifying words that might be interpolated.

Implementation

Sketch Engine's word sketch facility provides lists of head words of phrases that are coordinated with *and* or *or*. Using these lists, we search the BNC for the coordinations in our dataset. For each coordination, each head word is looked up in turn. The ranking of the second head word with the first head word may not be the same as the ranking of the first head word with the second head word. This is partly because different words simply have different numbers of other words that they are found coordinated with. It is also partly because of factors affecting the salience scores. There can be a significant difference between the two rankings. The following sentence illustrates this:

The whole kit and kaboodle

Kaboodle is ranked 19th of all the words that are coordinated with *kit*, but *kit* is the top ranked (and only) word coordinated with *kaboodle*. (The salience scores are 10.42 and 9.59 respectively). We want to represent the fact that expressions such as *kit and kaboodle* are firmly established, and we do not wish to discriminate against unusual words. Therefore we choose the higher of the two rankings to be the result of the coordination match heuristic.

The word sketch lists do not distinguish between head words coordinated with *and* and those coordinated with *or*: they lump them all together. (*And/or* is not considered at all). We can test whether the coordinating conjunction used is predictive of whether an ambiguity is nocuous or innocuous. If this is found not to be the case, then not discriminating coordinations based on the coordinating conjunction they use may be acceptable for our task. This is discussed in Section 5.6.8, as one of our unsuccessful (lexical) heuristics.

5.6.2 Distributional Similarity

This heuristic predicts the preferred interpretation of coordination ambiguities by looking at the distributional similarity between the head words of the conjuncts of those coordinations.

Hypothesis

As explained in 5.5.1, distributional similarity is a measure of the similarity of words based on shared context. We measure this between the head words of the conjuncts of the coordinations in our dataset. We hypothesise that if the two head words display strong distributional similarity, then the conjuncts are likely to be a syntactic unit. This hypothesis, based on the idea that thesaurally close words are often found in conjunction, has been suggested by Kilgarriff (2003a)[2]. In such a case, a coordination-first reading is preferred: the conjuncts form a syntactic unit, so the process of coordinating them occurs before the attachment of the modifier. If the preference is indicated strongly enough, the ambiguity will be innocuous. For example, let us consider the two following phrases:

old [[*boots*] *and shoes*]

[2] Kurohashi and Nagao (1992) and Resnik (1999) have also remarked upon this. However, they prefer to consider semantic similarity rather than distributional similarity as the appropriate measure.

old [[boots] and apples].

The former more naturally has a coordination-first reading and the latter a coordination-last reading. This may be understood experimentally by considering situations in which the modifier would be applied to both conjuncts simultaneously. These can be imagined for the first phrase but not for the second. Distributional similarity measures reflect this fact by determining that *boots* and *shoes*, but not *boots* and *apples*, occur in the same contexts.

In addition to the advantages of distributional similarity already mentioned, it may have special suitability for analysing coordinations. For instance, words having opposite meaning, such as *good* and *bad*, are frequently coordinated:

This system has very [[good] and bad] aspects

It seems likely that opposites such as these will often be found in the same contexts. For this example, it will be the same things that receive qualitative modifications *good* and *bad*. A distributional thesaurus will therefore indicate that such words have similarity. A semantic thesaurus will, however, not indicate this. If any coordinations of opposites are found to be read coordination-first, then distributional similarity may be the better similarity measure for our purposes. There are words in our dataset, like *input* and *output*, which have some opposite aspects. These are very strongly perceived to be read coordination-first by our judges.

Implementation

For each coordination, the lemmatised head words of the conjuncts are looked up in turn in the Sketch Engine's thesaurus. The thesaurus generates a ranked list of the words with which the entered word has distributional similarity. If the other head word is found in that list, its ranking is noted. For any given coordination, we may obtain different results depending upon which head word we look up. This is similar to the situation with the Coordination Matching heuristic, and for the same reasons we use the higher of the two rankings as the result of the heuristic.

5.6.3 Collocation Frequency

This heuristic predicts the preferred interpretations of coordination ambiguities by using a generic corpus to determine how frequently each of the head words of conjuncts are modified by the modifier. This heuristic differs from the other successful heuristics based on word distribution information in that it predicts coordination-last readings.

Hypothesis

We hypothesise that if a modifier is much more frequently collocated in the corpus with the coordinated head word that it is nearest to, than it is with the further head word, then it is more likely to form a syntactic unit with only the nearest head word. This implies that a coordination-last reading is the most likely. This can be demonstrated using the following example from our dataset:

[Reliability and [security]] <u>considerations</u>

If *security considerations* are found much more often in the corpus than *reliability considerations*, then we say that a coordination-last reading is indicated. If this indication is strong enough, the ambiguity is innocuous.

Note, however, that the possibility of finding collocations for many examples of coordination ambiguity will be small due to combinatorial factors. This is illustrated in the following example from our dataset:

Facilitate the [scheduling and [performing]] <u>of works</u>

No collocations are found. It could be that there is low likelihood of ever finding *works* associated with either *scheduling* or *performing*. However, it is probably more likely because the preposition of must be considered as well. The possibility of finding the phrase of works as a modifying collocation is much less likely.

Nakov and Hearst (2005) and Rus, Moldovan, and Bolohan (2002) had some success using collocation frequencies in tasks similar to ours. However, they have a much more narrowly defined dataset containing only noun coordinations. From an RE perspective, Maarek and Berry (1989) have used *lexical affinities* to find abstractions in requirements documents. The way they use lexical affinities is very similar to our use of collocation frequencies. They do, however, permit a longer distance between words with lexical affinity than we do between words with collocation frequency.

Implementation

Using the word sketch facility's collocation lists, we find the frequencies in the BNC with which the modifier in each sentence is collocated with each of the conjuncts' head words. There are collocation lists for most relationships that a word can have with a modifier. These include situations where the modifier is an adjective, noun, particle, predicate, possessive, adverb, the head word of all types of prepositional phrases, and others.

Although the modifier is often adjacent to the conjunct head word, there can be other words interposed between the two. The Sketch Engine's grammatical patterns should account for this, and enable a non-adjacent modifier to be considered as a collocation. For any word sketch, the Sketch Engine provides a concordance of the actual sentences it has located containing the input word. Upon inspection of this concordance, non-adjacent modifiers are found with encouraging regularity.

5.6.4 Morphology

This heuristic attempts to capture the morphology of the head words of the conjuncts of coordination ambiguities. It uses this to predict how these ambiguities should be interpreted.

Hypothesis

The inflectional morphology of English consists largely of suffixes. These include *-ed* to indicate past tense, *-ing* to indicate a progressive action, *-s* to indicate a plural, and so on. The derivational morphology of English is more complex, but suffixes are also very common. For instance, *-ation* creates a noun from a verb, *-able* creates an adjective from a verb and *-ise* (or *-ize*) makes a verb from a noun.

For these reasons it makes sense to compare *trailing* characters of the head words of conjuncts, as a first step to exploring how morphology might predict preferred readings of coordinations. If the two head words have the same specified number of trailing characters, then they are likely to be morphologically similar.

We hypothesise that if the head words of conjuncts end with the same characters then a coordination-first reading is preferred. This is because similarities in *inflection* will be captured. Words that are similar in this way will be more likely to form a syntactic unit. This theory follows on from Okumura and Muraki's (1994) notion of using *syntactic parallelism* as a method of disambiguating coordinations. The idea can be seen in the following sentence taken from our dataset:

It will be [implemented and [executed]] <u>*on the ...... platform*</u>

Intuitively, if the morphology of the head word furthest from the modifier (*implemented*) were different, for example if it were *implementable*, then a coordination-first interpretation would be a less natural.

The morphology of English is simple compared to that of many other languages. The results of this heuristic might therefore be dramatically different depending on the language used. However, the principle of syntactic parallelism might be considered applicable to all human language.

Implementation

To implement our morphology hypothesis, we simply attempt to match an equal number of trailing characters from the head words of the conjuncts. A minimum of one character is used, which may capture morphology such as plural endings, though it will more likely capture noise. A maximum of 6 letters is used, since the possibility of capturing English morphology becomes vanishingly small with consideration of any greater number.

5.6.5 Phrase Length Difference

This heuristic captures the lengths of the conjuncts of a coordination, in terms of numbers of words. It attempts to predict preferred readings of coordination ambiguities by comparing these.

Hypothesis

Schepman and Rodway (2000) examine the effect of *prosodic boundaries* on coordination disambiguation, using a similar test case to ours. Although their work is on spoken language, aspects of prosody such as numbers of syllables and words can also affect the way written language is interpreted. This might be due to the human capacity of processing only a limited number of linguistic units at one time. Also in support of this is the idea that sentences are read in *chunks*. The initial purpose of chunking was to model prosodic structure (Abney 1995). It aims to give a representation of how words are actually grouped together in a reader's mind. It is therefore a suitable model for thinking about the natural boundaries that occur in language.

We apply these ideas to the lengths of conjuncts in coordinations. Coordinating conjunctions, broadly speaking, yoke things of equal "status" (Jurafsky and Martin 2000). We hypothesise that when processing the structure of the sentence, readers will be more inclined to consider chunks of the same length to be equal in status. We hypothesise that if the conjuncts of a coordination are equal in length, then a coordination-first reading will be the most likely. Conversely, increasing disparity between their lengths will indicate increasing likelihood of a coordination-last reading.

Implementation

To test this hypothesis we simply count and compare the numbers of words in the conjuncts. No account is made of numbers of syllables or characters. (If English used compounding more frequently, resulting in more diverse word lengths, then this might be an preferable strategy.) Also, no account is made of which conjunct is the longer. The result returned by the heuristic is the number of words of the longer conjunct minus the number of words of the shorter one.

5.6.6 Noun Number Agreement

This heuristic uses number agreement between nouns to predict how coordination ambiguities should be interpreted. It requires that the head words of conjuncts are nouns. Its coverage overlaps with the morphology heuristic, as noun number is usually indicated by suffixes, but it is more specific. It is an investigation into Resnik's (1999) claims about the importance of number agreement when determining how noun coordinations should be read. Coordinations of nouns generally make up the great majority of coordinations, and this is also the case for our dataset.

Hypothesis

Resnik devises heuristics for disambiguating constructions that are coordinations of nouns and also have only nouns as their modifier — in other words of the form *noun1 coord noun2 noun3*. He can therefore compare noun number between all three of these nouns. This allows the use of rules that clearly indicate preferred bracketings, as demonstrated in the following examples:

> *Several [business and university] groups*
>
> *Several businesses and [university] groups*

We do not have such a narrowly defined dataset. We have sufficient noun coordinations to warrant a noun number heuristic, but not sufficient of these have a noun modifier to warrant implementation of Resnik's heuristics. Also, Resnik's examples appear to contain elements that would contravene our single structure criteria.

We therefore look simply at number agreement between the head words of the conjuncts of a coordination. A noun coordination can be of two singular nouns, two plurals, or one of each. The first two situations represent noun number agreement, the third situation represents lack of agreement. Adapting Resnik's example, the following sentence indicates why noun number agreement of conjuncts alone may be a weaker indicator of preferred readings:

> *We are targeting [businesses and [universities]] in this town*

However, number agreement is still a measure of similarity. As with distributional similarity, we wish to discover if this is a useful predictor of coordination-first readings — and if lack of it predicts coordination-last readings. If such predictions are found to be sufficiently strong for an ambiguity, then this will indicate that the ambiguity is innocuous.

Implementation

To test whether noun number plays a part in the interpretation of coordination ambiguities, we manually determine whether the head word of each conjunct of nouns is single or plural. When evaluating this heuristic, we consider its performance to be only on the sentences in the dataset that are noun coordinations.

5.6.7 Mass/Count

This heuristic captures whether coordinated nouns are mass or count, and uses this information to determine how the coordination containing them should be read. It's coverage overlaps to some extent with the Noun Number heuristic — and to a lesser extent with the Morphology heuristic. This is because true mass nouns are always singular, and that fact will generally be captured by those heuristics. However, they will not capture many differences between singular nouns, such as whether they are mass or count.

Hypothesis

The distinction between mass and count is one obvious type of noun subcategorisation in English. Coordinations of words with different subcategorisations can sound incongruous or unlikely, implying that alternative readings which assign identical subcategorisations to those words are generally preferred. Plenty of counterexamples exist, but the mass/count distinction may be a particularly significant indication of preferred readings of ambiguities, as shown in following sentence:

> *I drink [water and [a pint]] <u>at lunchtime</u>*

We hypothesise that coordinations of nouns that are either definitely both mass or definitely both count — a situation we term *unequivocal agreement* — will be more likely to be read coordination-first. Where there is the possibility that one or both of the nouns can be either mass or count — a situation we term *equivocal agreement* — we predict that the indication will be weaker. If either type of agreement is found to make sufficiently strong prediction for an ambiguity, then this will indicate that the ambiguity is innocuous.

Implementation

We determine manually whether each head word of a noun coordination is mass, count or either. When evaluating this heuristic, we consider its performance only on the sentences in the dataset that are noun coordinations.

5.6.8 Other Candidate Heuristics

We evaluated several other heuristics to see if they could predict coordination-first or coordination-last interpretations, and therefore innocuous ambiguity. These looked reasonable in theory, but most did not demonstrate predictive power in our experiments. The semantic similarity heuristic is the exception, in that we replaced it with another heuristic.

Semantic Similarity

The only other heuristic with which we achieved limited success in preliminary studies (Chantree, Nuseibeh, de Roeck, and Willis 2005) used a simple metric of semantic similarity. This was based on the closeness of the coordinated head words to their lowest common ancestor in hierarchies of WordNet hypernyms. Similar use of semantic similarity has been used with some success for disambiguating coordinations (Resnik 1999). We achieved weak prediction of coordination-first interpretations. Kilgarriff (2003a) claims that distributional similarity is for many NLP tasks preferable to, and substitutable for, semantic similarity. We have therefore preferred to develop our heuristic based on the former notion in preference to, and in place of, a heuristic based on the latter notion.

Orthography

Several aspects of word appearance suggest themselves as indicators of preferred readings of coordination ambiguities. The orthography heuristic that held most promise for us was based on named entity recognition. We hypothesised that if the two head words of the conjuncts begin with a capital letter, but the modifier does not, then a coordination-first reading is preferred. This is because capitalisation of the first letters of a group of words indicates they they are a named entity and therefore form a syntactic unit. (Note that this would not be the same as the second orthographic criterion discussed in Section 5.2.1. The situation is less clear-cut: it is always possible that each conjunct is a named entity and the uncapitalised modifier attaches to just one of them.)

This hypothesis is demonstrated in the following phrase from our dataset:

> *more* <u>*sophisticated*</u> *[[Capacity] and Coverage]*

If the head word furthest from the modifier does not begin with a capital letter, then the sentence is not so clearly interpreted coordination-first. However, there were not sufficient examples such as this in our dataset to test whether this heuristic is an effective predictor.

Lexical Heuristics

We tested several lexically-based heuristics, but none appeared to have significant power at predicting preferred readings of coordination ambiguities. In our dataset, neither the part of speech of the modifier nor the part of speech of the head words indicated preferred interpretations.

The same was true of the actual word used as a coordinating conjunction: *and* and *or* displayed similar prediction characteristics. (This has significance for the validity of our Coordination Matching heuristic. When generating the lists used by this heuristic, Sketch Engine does not distinguish between coordinations using *and* and those using *or*.

Heuristic	Designation in Dataset		
Results	innocuous	nocuous	total
predict	True Positives (TP)	False Positives (FP)	(W)
¬ predict	False Negatives (FN)	True Negatives (TN)	(X)
total	(Y)	(Z)	(T)

Table 5.5: Contingency matrix for deriving evaluation measures

An *and* in the dataset may therefore be mismatched with an *or* in the lists, and vice versa. However, the similar prediction characteristics we have found between *and* and *or* implies that this may not be a problem. Which coordinating conjunction is used in a sentence may have little effect on the results of our Coordination Matching heuristic.)

5.7 Evaluation

Here we present the measures and statistics that we use to evaluate the performance and validity of our techniques. Our gold standard in all cases is human perception of ambiguity. First we present the statistics which we use to evaluate individual heuristics. These, along with the use of cut-offs, demonstrate how innocuous ambiguity can be optimally determined. We then introduce ROC curves, which we use to test the overall validity of each heuristic as a diagnostic test. Then we present the methods we use to combine our heuristics' predictions. Firstly we use a simple disjunction method that demonstrates transparently how effective the heuristics are in combination. Secondly, we use a more advanced statistical model using all the heuristics' data simultaneously. The advantages and disadvantages of alternative methods are also discussed.

5.7.1 Performance Measures for the Individual Heuristics

We evaluate and optimise the performance of our individual heuristics using measures of precision, recall, fallout and accuracy. We combine precision and recall using an f-measure evaluation. These calculations are commonly used in information retrieval and statistical NLP (Manning and Schütze 1999). We use these measures when implementing the Weighted Method. We use the Weighted Method to demonstrate the individual performance of each heuristic and how this can be maximised. We also use recall and fallout to create ROC (*receiver operating characteristic*) curves. The ROC curves give an overall appreciation of how effective the heuristics are as diagnostic tests of innocuous ambiguity. In doing this they also test the validity of the hypotheses on which the heuristics are founded. We use *accuracy* when implementing the Flexible Method and the Purely Unacknowledged Method.

All the measures discussed here are ultimately derived using the contingency matrix in Table 5.5. The individual heuristics are evaluated on their ability to predict *innocuous ambiguities*. This is because the heuristics naturally predict innocuous ambiguities: they attempt to predict that an ambiguity has a single reading. We wish to determine how successful they are at this. Of the measures we present here, accuracy is the only one that indicates more than ability at predicting innocuous ambiguity: it also factors in ability at predicting nocuous ambiguity. For this reason it is not used to evaluate the individual heuristics. We use it, however, to evaluate them in combination, as this demonstrates the overall efficacy of our technique.

Basic Measures

Precision for a heuristic is the proportion of its predictions of innocuous ambiguity that are correct. Recall for a heuristic is the proportion of innocuous ambiguities that it predicts. Fallout for a heuristic is the proportion of nocuous ambiguities that it wrongly predicts to be innocuous. Accuracy for a heuristic is the proportion of all the predictions it makes (both innocuous and nocuous) that are correct. In other words, using the abbreviations used in Table 5.5:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} = \frac{TP}{W}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} = \frac{TP}{Y}$$

$$\text{Fallout} = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}} = \frac{FP}{Z}$$

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Number of Examples}} = \frac{TP + TN}{T}$$

Precision is much more important to us than recall: we wish each heuristic to indicate reliably how any given coordination should be read. We do not wish it to make a lot of suggestions which may be questionable, thereby inspiring doubt. Also, the heuristics' predictive powers can be combined by using the heuristics as a suite of techniques. Good recall may thereby be achieved if the heuristics have complementary coverage of the data. Many coordinations will then be disambiguated with good precision. Fallout is useful for evaluating *collateral damage*: in our case, coordinations that are judged to be innocuous when in fact they are nocuous. This is the most dangerous scenario, and wish to avoid this as far as possible. We use fallout, together with recall, to create ROC curves.

F-Measure

However, high precision with trivial recall would render our techniques rather pointless. We therefore employ measures to combine the two, but with precision prioritised. We use the commonly-used weighted *f-measure* statistic (van Rijsbergen 1979) to combine precision and recall:

$$\text{F-Measure} = \frac{(1+\beta^2) * \text{Precision} * \text{Recall}}{\beta^2 * \text{Precision} + \text{Recall}}$$

A weighting of $\beta = 1$ will give equal weight to precision and recall; a weighting of $\beta = 0.5$ is commonly used, for example (Ishioka 2003), to ensure that true positives are not obtained at the expense of also obtaining too many false positives. We use a weighting of $\beta = 0.25$, even more strongly in favour of precision. In line with our policy of achieving high precision and complementary recall, we aim to maximise the f-measure for all our heuristics.

Cut-Offs

We employ a cut-off when evaluating each individual heuristic in order to maximise its performance. Results up to and including the cut-off are considered to predict innocuous ambiguity; those beyond the cut-off are considered to predict nocuous ambiguity. This means that innocuous ambiguity is predicted by results either no smaller or no greater than the cut-off, depending on the heuristic. The cut-offs are found experimentally for each heuristic. The optimal cut-off chosen for any heuristic is the one that optimises the weighted f-measure statistic.

ROC Curves

ROC curves are the evaluation measure of choice in some fields of engineering (Manning and Schütze 1999). They are also commonly used for evaluating diagnostics in medicine (Zhou, McClish, and Obuchowski 2002). They measure how good a diagnostic test is, representing the trade-off between the sensitivity (represented by recall) and the specificity (represented by fallout) of that test. A good diagnostic test is one that has small fallout and high recall rates across a range of cut-off values. A bad diagnostic test is one where the only cut-offs that give low fallout cannot produce high recall, and vice versa.

We measure how effective a heuristic is as a diagnostic of innocuous ambiguity by measuring the area under the ROC curve. Fallout is plotted on the x-axis and recall on the y-axis. The larger this area, the better the heuristic is as a diagnostic. The baseline of a ROC curve graph is a straight line representing equal trade-off between sensitivity and specificity — running from the bottom left corner to the top right corner. The baseline of a ROC curve graph is therefore always 50%. The closer the ROC curve is to the baseline, the worse the heuristic is as a diagnostic of innocuous ambiguity — little better than flipping a coin.

We produce ROC curves using *all* the cut-offs that we use for evaluating our heuristics. The area under each curve therefore measures the *overall* effectiveness of a heuristic at predicting innocuous ambiguity, without any attempt at tuning it in order to maximise performance. This measurement also validates (or not) the hypothesis underlying the heuristic. It is for this reason that we use ROC curves to demonstrate prediction of *innocuous ambiguity*. The hypotheses underlying the heuristics assert that a certain reading of an ambiguity is the preferred one if certain criteria are fulfilled. The heuristics naturally predict innocuous ambiguity as a result of this. We therefore use measurements of recall and fallout based only on prediction of innocuous ambiguity. (Using *accuracy*, factoring in prediction of nocuous ambiguity, would make the validation of the hypotheses less clear.)

These are the most important conclusions we take from our ROC graphs, but they are not the whole story. A heuristic may still be useful, even though its ROC curve demonstrates indifferent power as a diagnostic test. ROC curves also demonstrate the variability of a heuristic's performance around the baseline. By using a heuristic optimally, we hope it will perform at the point where its ROC curve is highest above the baseline. Such optimisation is achieved using the cut-offs, or the regression modelling introduced in the next section.

5.7.2 Statistics to Combine Heuristics

The predictive powers of our heuristics can be combined in various ways. These include machine learning, straightforward combinatorial approaches and statistical models. We first explain briefly our reasons for not using the first of these in this research, then we describe the two techniques we have used: a simple disjunction approach and logistic regression.

Machine learning algorithms can be used to combine predictions. This idea has attracted the attention of requirements engineers for a decade or so (Spanoudakis, d'Avila Garcez, and Zisman 2003). However, the number of features that we can muster — i.e. the number of heuristics — is much less than the minimum normally considered appropriate (Forman 2003). We experimented with machine learning algorithms, such as lazy learning (Daelemans et al. 2003) and decision tree (Quinlan 1992), but decided that they were not best suited to our task.

Combination Using Disjunction

In this method we simply say that the combined heuristics are considered to predict innocuous ambiguity if any one of the individual heuristics gives a positive result.This increases the recall of our combined heuristics, if they have complementary coverage. We hypothesise that it will be partly true that each heuristic measures something different. However, there is bound to be some overlap. For instance, the difference in the frequency ratios captured by the collocation frequency heuristic is to some extent a measure of distributional dissimilarity.

Note that we cannot use conjunction rather than disjunction to combine our heuristics. This would require saying instead that the combined heuristics predict innocuous ambiguity only if *all* of them give a positive result. This is because some heuristics predict coordination-first readings and others predict coordination-last readings, and they would be negating each other in such an arrangement.

Logistic Regression

The other method we use to combine and optimise our heuristics' predictive power is linear logistic regression. We use this method when implementing the Flexible Method, to determine our combined heuristics' power at discriminating between nocuous and innocuous ambiguity. We also use it similarly when evaluating unacknowledged ambiguity. Logistic regression is a statistic often offered by machine learning packages. It is the most suitable metric for us as it can be used to model a dependent variable that is dichotomous and independent variables that are of various types. The dependent variable in our case is the human judgement of nocuousness: the consensus of our judges opinions, taking the ambiguity threshold into account, as to whether an ambiguity is considered nocuous. The independent variables are the heuristics. Instead of using cut-off points we put the actual results returned by the heuristics into these variables. These variables can be continuous (Coordination Matching, Distributional Similarity and Collocation Frequency) or categorical (Morphology, Phrase-length, Mass/Count and Noun-Number) — the last named variable is also dichotomous.

The aim of this approach is to fit the linear logistic regression model to our data. The form of the model for our seven heuristics is:

$$\ln\left(\frac{p}{1-p}\right) = \alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \beta_5 X_5 + \beta_6 X_6 + \beta_7 X_7$$

p is the probability of there being a consensus of our human judges saying that the ambiguity is innocuous. The X_n values are the results returned by each of the heuristics (the independent variables). The β_n values are the coefficients that, along with the intersect α, balance the equation.

We utilise the LogitBoost algorithm (Friedman, Hastie, and Tibshirani 2000) to fit the model to the data. This algorithm operates iteratively, automatically selecting independent variables and only including those that improve the performance of the model on unseen cases. We use LogitBoost to find the maximum likelihood linear logistic model, using least squares regression and operating in a forward stage-wise manner (Landwehr, Hall, and Frank 2003).

This procedure is implemented using the WEKA machine learning algorithms developed at University of Waikato (http://www.cs.waikato.ac.nz/ml/weka/) (Witten and Frank 2005).

5.8 Avoiding Bias and Inappropriateness

We focus here on two ways in which we can ensure that the results we obtain are appropriate and not misleading. Firstly, it is important that we minimise any data-specific bias in our results. Succeeding at this will show that our heuristics can be applied to unseen data as well as to our dataset. Secondly, we also wish to ensure that we use appropriate ambiguity thresholds, when these are implemented.

We explain the cross-validation process we use to avoid bias and overfitting in our results. We discuss this mainly in relation to implementation using the Weighted Method. We use this method to demonstrate the performance of the individual heuristics and their underlying hypotheses. The other methods modelled using the WEKA machine learning package, have a less transparent process. Then we discuss the choice of appropriate ambiguity thresholds. This includes a discussion of how they might be used appropriately to optimise results.

5.8.1 Cross-Validation

We employ ten-fold cross-validation to ensure that any statistics obtained from our data are not unduly biased. This technique avoids the problem of generating statistics that are specific to a certain view of the dataset. This problem can result in *overfitting*, whereby the statistics are trained to perform well for a particular dataset but not for *unseen* data. Ten-fold cross-validation is recognised as an accurate and efficient method for datasets of sizes such as ours (Weiss and Kulikowski 1991). Our Weighted Method requires that we perform ten-fold cross-validation manually. We describe below how we perform this cross-validation, in conjunction with optimisation of the cut-offs. For the Flexible and Unacknowledged Methods, we use the ten-fold cross-validation facility offered by the WEKA machine learning package. This process is automatic, so we do not discuss it here.

When using the Weighted Method, our dataset is first randomly sorted. This removes bias caused by the order in which the sentences were collected. Then it is split into ten equal parts. Nine of the parts are concatenated to form the *training* set. This is used to find the optimal cut-off for each heuristic. These cut-offs are found experimentally. The heuristics are then run on the heldout tenth part, known as the *test* set, using those optimal cut-offs. This procedure is carried out ten times, using a different heldout part each time and concatenating the other nine parts to form a new training set. The performances on all the ten different heldout parts are then averaged to give the performances of the heuristics.

Some researchers prefer to use three datasets, in what the machine learning community often refers to as a *training and validation set* approach (Mitchell 1997). (It is also used in neural network methods (Ripley 1996); the terminology, however, can sometimes be used differently in different disciplines). In addition to training and test sets, a *validation* set is used. Like the training set, this is set of *seen* data. This extra set generally acts as a method of tuning the parameters that have been chosen as a result of using the training set. This technique adds another guard against overfitting and can be a way of increasing the reliability statistical data. In the work presented here, we could have presented analysis of our process of selecting heuristics, followed by analysis of how we optimised them. Training and validation sets could have been used respectively to do this — with the test set still providing the statistics on unseen data. However, we did not utilise a third dataset for two reasons. Firstly, we do not have a large dataset, and it is known that the validation set should be large enough to provide a suitable sample (Mitchell 1997). Secondly, by using the same set of data for both training and validation purposes, we can evaluate the interaction between the heuristics when they are used in combination. This would not be possible if different heuristics were used on the different sets.

5.8.2 Appropriate Ambiguity Thresholds

In deciding which combination of judgements make an ambiguity nocuous or nocuous, certain ambiguity thresholds should not be used, even if they appear to allow for good performance by the heuristics. For instance, with a threshold of 50%, no consensus among the judges is required for determining either nocuous or innocuous ambiguity. It seems to us that a minimum level of certainty should be higher than this. The upper limit of the ambiguity threshold is harder to judge. Using an ambiguity threshold of 80% will allow for three of our seventeen judges giving minority verdicts. This is arguably an acceptable accommodation of overly-cautious judgements and rogue judgements.

Implementing ambiguity thresholds as a policy decision, as recommended in Section 5.4.1, is not the only way of using them appropriately. They can be used to optimise performance. Users might prefer to set an ambiguity threshold that gives the best possible performance from our heuristics. This would then require that authors and readers are persuaded to check ambiguities with a diligence that bore this in mind. To optimise overall performance, a different ambiguity threshold can be set for each heuristic and for each iteration of the cross-validation exercise. Using this approach, we have achieved good performance from heuristics similar to those presented here (Chantree, Willis, Kilgarriff, and de Roeck 2006). However, this method does not best implement and test the model of ambiguity presented in this thesis. Therefore we do not present results based on this use of ambiguity thresholds.

5.9 Summary

In this chapter we have described how we implement the ideas contained in our language model. This has involved building a specialist corpus from which we can extract real-world ambiguities of the same type and structure for use as our test data. We have then described how judgements on these ambiguities were obtained by asking the opinions of a carefully selected group of people. We then described the different methods that we use to distinguish nocuous from innocuous ambiguities. This included the key concept of the ambiguity threshold, which establishes the dividing line between the two classifications. Next, we introduced the heuristics used to predict the judgements in our dataset. This involved describing the tools they use, the ways they are evaluated, and the methods used to ensure that their results are appropriate and unbiased.

Chapter 6

Empirical Study

This empirical study was conducted in order to test two main hypotheses. The first is that nocuous ambiguity exists to a significant extent when human judgements are considered. The second is that the distinction between nocuous and innocuous ambiguity can be predicted automatically using heuristics. We first describe the dataset that we created in order to test our hypotheses. Then we present the results of our heuristics on this dataset. The results, and our discussion of them, are presented for each of the three methods we use for determining nocuous ambiguity. The results of the Weighted Method are given first. This section is also used to present the general efficacy of each heuristic. This is followed by results using the Flexible Method, then those when only unacknowledged ambiguity is considered.

6.1 The Dataset

Our dataset is composed of sentences containing coordination ambiguities that conform to our test case criteria, together with judgements on them regarding their interpretation. Here we discuss the characteristics of the sentences we use from the corpus and the distribution of the judgements on these sentences.

Head Word	% of Total	Example from Questionnaires
Noun	85.5	Communication and performance requirements
Verb	13.8	Proceed to enter and verify the data
Adjective	0.7	It is very common and ubiquitous

Table 6.1: Breakdown of sentences by head word type (head words are underlined)

Modifier	% of Total	Example from Questionnaires
Noun	46.4	(It) targeted the project and election managers
Adjective	23.2	.... define architectural components and connectors
Prep	15.9	Facilitate the scheduling and performing of works
Verb	5.8	capacity and network resources required
Adverb	4.4	(It) might be automatically rejected or flagged
Relative Clause	2.2	Assumptions and dependencies that are of importance
Number	0.7	zero mean values and standard deviation
Other	1.4	increased by the lack of funding and local resources

Table 6.2: Breakdown of sentences by modifier type (modifiers are underlined)

6.1.1 The Ambiguity Questionnaires

We located a total of 639 sentences containing *and*, *or* or *and/or* in our RE corpus, some of which contained more than one instance of these words. From these we extracted 138 sentences. Each one of these contains one multiple structure coordination ambiguity conforming to our test case criteria, as set out in Section 4.1.2. A breakdown of these sentences by the part of speech of the head words of the conjuncts is given in Table 6.1. A breakdown by the part of speech of the external modifier is given in Table 6.2.

The sentences were divided up into four separate questionnaires, formulated as described in Section 5.3.2. These questionnaires were shown to seventeen judges, who were selected as described in Section 5.3.3. The 138 sentences are presented in Appendix B; the instructions are presented in Appendix A.

6.1.2 The Judgements

A breakdown of the judgements on the sentences is given in Table 6.3. Less than 1% of judgements were absent or spoiled, and these omissions were spread across sentences and

Judgement	No. of Judgements	Percentage of Total
Coordination Last	583	24.851
Coordination First	951	40.537
Ambiguous	791	33.717
(Missing or Spoiled)	21	0.895
Total	2346	100

Table 6.3: Breakdown of the judgements in the dataset

judges. We ignore these as they represent such a small percentage. We use totals of all the other judgements as the base figures for all our calculations. The average number of judgements for each sentence is 16.845.

6.2 Predictions Using Weighted Method

Here we present the results of our heuristics when using the Weighted Method to distinguish nocuous from innocuous ambiguity. We use this method to demonstrate the performance and characteristics of the individual heuristics. To reiterate, we use this method to demonstrate how our heuristics predict innocuous ambiguity. This is what they predict naturally. We therefore wish to see how they can do it most effectively, so we can judge the hypotheses that they implement. It is useful to use this method for this purpose, as it gives total control over the optimisation of the heuristics. We can therefore explain how and when they can achieve optimal performance. Also, this method is not complicated by the added variable of flexible ambiguity thresholds.

All the baselines used for this method are obtained by assuming that all ambiguities are innocuous. This gives a precision baseline of 54.3% and a f-measure baseline of 55.8%. The recall baseline is 100%, as all innocuous ambiguities have predicted. Only the precision baseline is plotted on the graphs for the individual heuristics: the f-measure baseline is very close to it, and plotting it would make the graphs hard to read.

For each heuristic, we present performance over a range of cut-offs. A cut-off is a ranking (or other measure) which is the highest (or lowest) point at which a heuristic is considered to predict innocuous ambiguity. The results for the individual heuristics are calculated prior to cross-validation. This is done because we wish to compare performance between different cut-offs, and view the prediction trends across these ranges. We cannot claim therefore that these results are necessarily valid for unseen data, but we hope to gain insight into the heuristics' behaviour in this way. We present worked examples of four heuristics found to be effective using the Weighted Method. These examples illustrate how the judgements on a sentence create acknowledged and unacknowledged ambiguity and how a heuristic predicts this. All calculations using the Weighted Method require knowledge of the average unacknowledged ambiguity over all the sentences in our dataset. This is calculated as described in Table 5.4 on page 118, and is 15.3%.

Then we provide ROC curves to show the overall effectiveness of each individual heuristic as a predictor. As explained in Section 5.7, the area under a ROC curve represents the power of a heuristic as a diagnostic test without any training. It is to be hoped that this area exceeds 50% of the graph, which is the baseline for all ROC curve graphs. In this exercise the area under the curve demonstrates a heuristic's power at predicting innocuous coordination ambiguity. It therefore tests the validity of the hypothesis underlying a heuristic. A ROC curve showing indifferent power as a diagnostic test does not necessarily prove that the heuristic is worthless, however. The variability of the heuristic's performance around a baseline is also significant. By using a heuristic optimally, we hope to use it at the point where its ROC curve is highest above the baseline.

We then combine the heuristics found to be effective using the Weighted Method. Results for these both before and after application of cross-validation are presented. This is followed by a discussion of these results.

6.2.1 Coordination Matching

Here we present the results achieved by the Coordination Matching heuristic on our dataset. This heuristic was introduced in Section 5.6.1. We present a worked example of its effectiveness when nocuous ambiguity has been determined using the Weighted Method. We then evaluate the heuristic and the hypothesis underlying it.

Example

We obtain the rankings of matching coordinations obtained from the BNC by Sketch Engine. For this heuristic, we evaluate cut-offs for these rankings in multiples of 5. We use the following sentence from our dataset to illustrate this heuristic:

> *Security and Privacy Requirements*

Of the 17 judges, 4 judged this sentence to be ambiguous, 12 judged it to be coordination-first and 1 judged it to be coordination-last. As the number of ambiguous judgements is not larger than the numbers of either of the other two types of judgement, this coordination is not an acknowledged ambiguity. The quotient unacknowledged ambiguity, calculated as described in Table 5.4 on page 118, is 1/13 = 7.7%. This is below the average unacknowledged ambiguity (15.3%), so this sentence is not an unacknowledged ambiguity. Neither of the criteria for being nocuous have been fulfilled, so this ambiguity is innocuous.

Privacy is the 19th highest match for *Security*, and *Security* is the 8th highest match for *Privacy*. Following our decision to not penalise coordinations involving unusual words (see Section 5.6.1), 8 is the result returned by this heuristic on this coordination. Our result is within all the ranking cut-offs except 5. The heuristic therefore yields a positive result for all the cut-offs except 5. For this requirement, this heuristic gives a true positive result for all our cut-offs above 5.

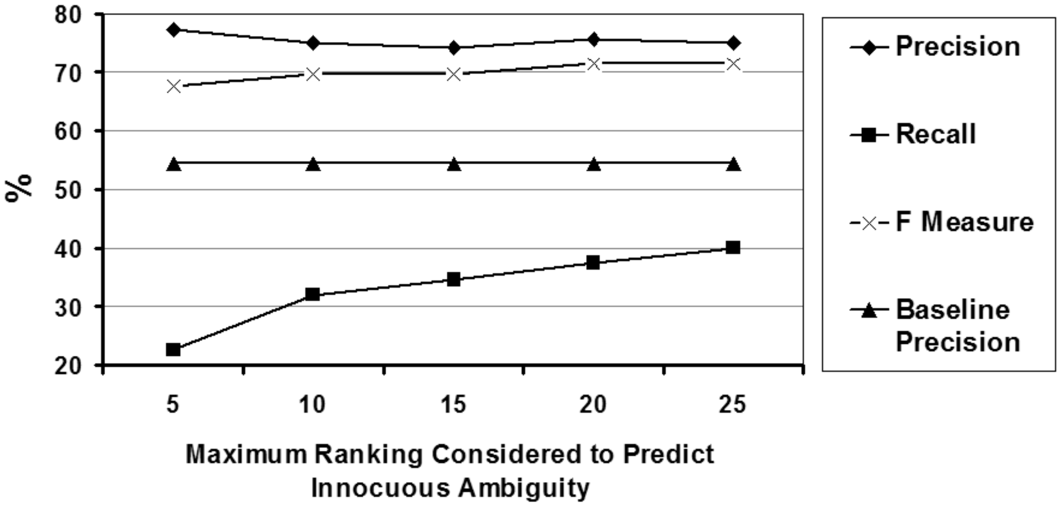


Figure 6-1: Coordination Matching Heuristic

Results

The results that we obtained using our Coordination Matching heuristic on our dataset are shown in Figure 6-1. As can be seen, precision in excess of 20 percentage points is achieved above the baseline of 54.3%. F-measure in excess of 15 percentage points above the f-measure baseline 55.8% and a maximum recall of 40% are achieved. Recall steadily increases as coordinations of increasingly lower rank from the BNC are considered to indicate innocuous ambiguity. However, precision decreases after more than the top 20 are considered. Precision is highest when only the top 5 ranked matches from the BNC indicate innocuous ambiguity. But it is very seldom that the head words of any coordination in our dataset are found so frequently as a coordination in the BNC. This is indicated by the low recall at that cut-off.

When implementing cross-validation, different cut-offs maximise the f-measure for different folds of the process. A maximum of 25 matches is optimal for 7 of the iterations and a maximum of 20 matches is optimal for the other 3.

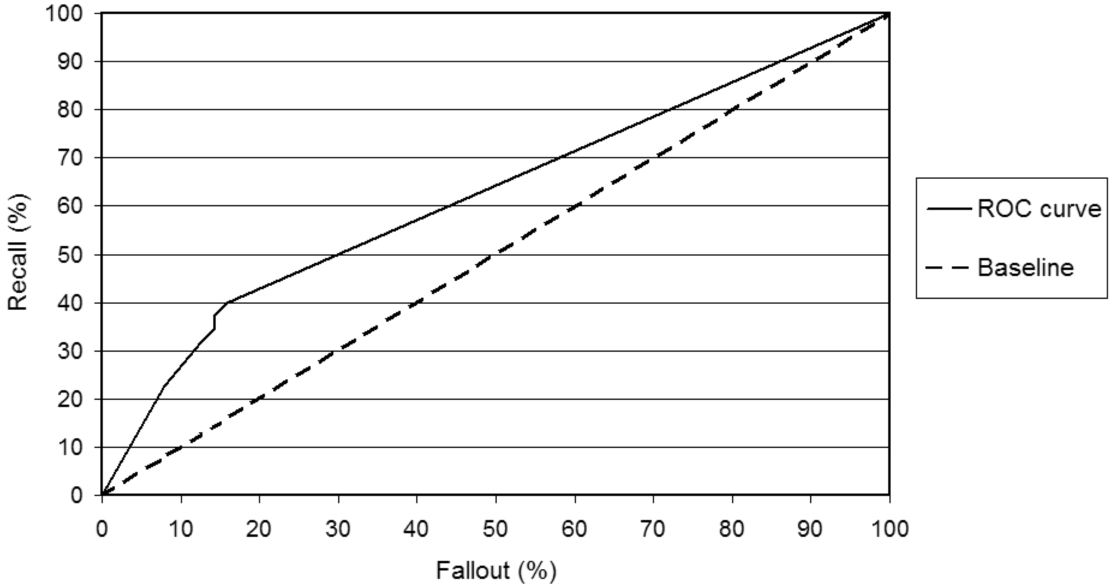


Figure 6-2: Coordination Matching heuristic ROC curve

Evaluation

The performance of this heuristic indicates that it is a useful and consistent predictor of innocuous ambiguity. If the f-measure was less strongly weighted in favour of precision, it might indicate better performance at higher cut-offs. However, the precision would be somewhat compromised. We require that precision be prioritised in order to inspire confidence in the heuristic.

The ROC curve for this heuristic is shown in Figure 6-2. The area under the ROC curve is 62.2%; the baseline is 50%. The predictive power of this heuristic is therefore 12.2 percentage points better than that of a test with no ability at predicting innocuous ambiguity. This goes some way to proving the hypothesis upon which the Coordination Matching heuristic is based. Namely, it demonstrates that if a coordination in our dataset is found a significant number of times in a generic corpus then it is more likely to be a syntactic unit. A coordination-first reading is therefore likely. If this likelihood is sufficiently strong, the ambiguity will be innocuous.

The ROC curve is always above the baseline. This indicates that the heuristic is consistently a good predictor of innocuous ambiguity over the cut-offs we have considered.

6.2.2 Distributional Similarity

Here we present the results achieved on our dataset by the Distributional Similarity heuristic. This heuristic was introduced in Section 5.6.2. We present a worked example of its effectiveness when nocuous ambiguity is determined using the Weighted Method. We then evaluate the heuristic and the hypothesis underlying it.

Example

We obtain the rankings of the distributional similarity of words from the Sketch Engine's thesaurus. Experimentation has shown us that the predictive power of distributional similarity decreases in a log-linear manner the more matches are considered. We therefore consider the *logs* of the numbers of matches obtained from Sketch Engine's thesaurus as cut-off points. We evaluate these cut-offs in multiples of 0.5. We use the following sentence from our dataset to illustrate this heuristic:

> *Definition of electrical characteristics and interfaces*

Of the 17 judges, 9 judged this sentence to be ambiguous, 4 judged it to be coordination-first and 4 judged it to be coordination-last. As 9 is greater than 4, this sentence is an acknowledged ambiguity. The percentage unacknowledged ambiguity is 4/8 = 50%. This is greater than the average unacknowledged ambiguity (15.3%), so the sentence is also an unacknowledged ambiguity. On both counts this ambiguity is nocuous.

No matches between *characteristic* and *interface* are found in the thesaurus. The Distributional Similarity heuristic therefore predicts that the coordination ambiguity in this sentence is never innocuous. This heuristic therefore predicts correctly that the

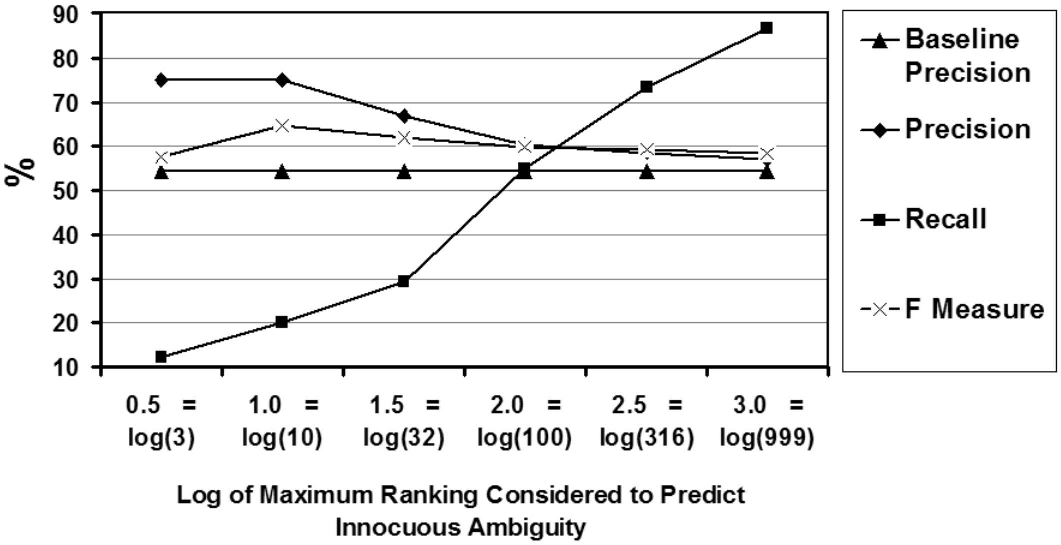


Figure 6-3: Distributional Similarity heuristic results at different cut-offs

coordination ambiguity in this sentence is not innocuous, regardless of what cut-off is used.

Results

The results we obtained using the Distributional Similarity heuristic on our dataset are shown in Figure 6-3. At its highest, precision in excess of 20 percentage points above the 54.3% baseline is achieved. This is when the distributional similarity between the two head words has to be in the top 10 ranked matches to be considered a positive result. An f-measure score of almost 9 percentage points above the 55.8% baseline is achieved at this cut-off. However, recall reaches only 20% here. Precision tails off markedly after more than the topmost rankings are considered, though some predictive power is still in evidence. Recall increases considerably when very low-ranked distributional similarity rankings are considered also indicative of innocuous ambiguity. The precision at these levels becomes unconvincing.

When implementing cross-validation, the f-measure is always maximised with a max-

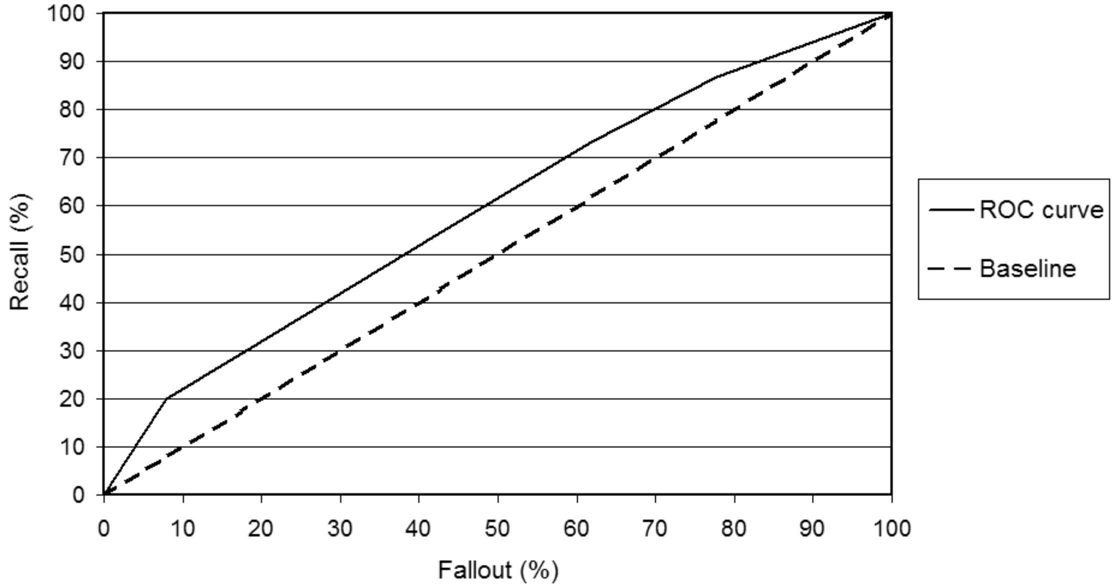


Figure 6-4: Distributional Similarity heuristic ROC curve

imum of 10 matches. This is therefore the cut-off we will use for all the iterations in the cross-validation exercise.

Evaluation

These results indicate that this heuristic can be an effective predictor of innocuous ambiguity. This is when the distributional similarity between the head words is very high compared with the distributional similarities that they have with other words.

The ROC curve for this heuristic is shown in Figure 6-4. The area under the ROC curve is 59.4%. Again the baseline for the ROC curve is 50%, so the predictive power of this heuristic is 9.4 percentage points better than that of a test with no ability at predicting innocuous ambiguity. This indicates that, without training or tuning, distributional similarity is a weaker diagnostic test than coordination matching. It still, however, goes some way to proving the hypothesis upon which the Distributional Similarity heuristic is based, namely that if the conjuncts' head words display strong distributional similarity, then the conjuncts are likely to be a syntactic unit. A coordination-first reading is therefore likely. If this likelihood is sufficiently strong, the ambiguity will be innocuous.

The ROC curve is always above the baseline. This indicates that the heuristic is consistently a good predictor of innocuous ambiguity over the cut-offs we have considered.

6.2.3 Collocation Frequency

Here we present the results achieved by the Collocation Frequency heuristic on our dataset. This heuristic was introduced in Section 5.6.3. We present a worked example of its effectiveness when nocuous ambiguity has been determined using the Weighted Method. We then evaluate the heuristic and the hypothesis underlying it.

Example

We obtain the frequencies with which the modifiers are collocated with the head words using the Sketch Engine's word sketch facility. We then calculate the collocation frequency ratio from these, and evaluate cut-offs for this ratio in multiples of 5. The collocation frequency ratio is the frequency with which the modifier is found collocated with the head word it is nearest to divided by the frequency with which it is found collocated with the further head word. We use the following example from our dataset to illustrate this heuristic:

> *Project manager and designer*

Of the 17 judges, 5 acknowledged this coordination as being ambiguous, 4 judged it to be coordination-first and 8 judged it to be coordination-last. As 5 is less than 8, it is not an acknowledged ambiguity. However, 4/12 = 33%, which is greater than the average unacknowledged ambiguity (15.3%). So this coordination is a nocuous ambiguity.

Project has a collocation frequency of 29.55 with *manager* in the BNC, but it has no collocations there with *designer*. This heuristic therefore always predicts that the

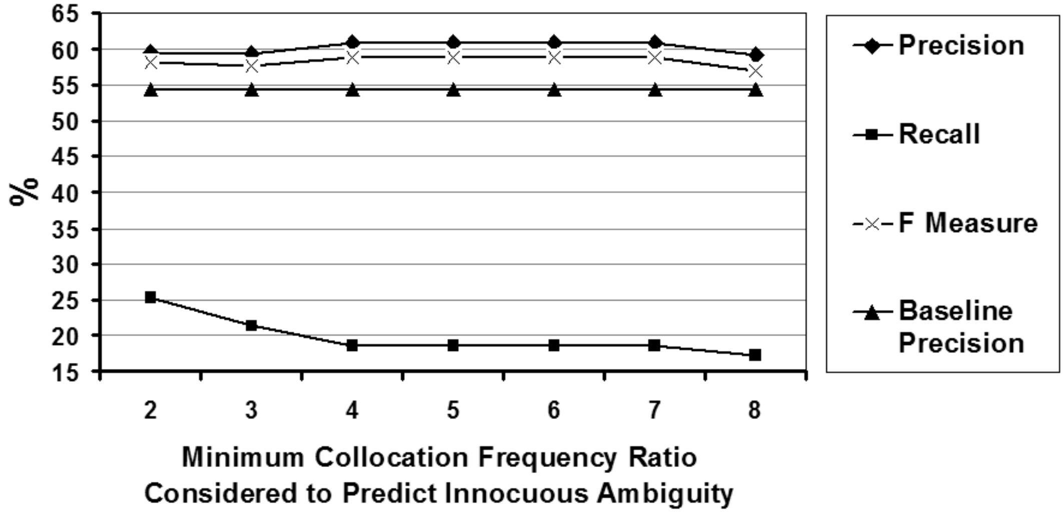


Figure 6-5: Collocation Frequency heuristic

coordination ambiguity in this sentence is innocuous. This is regardless of whatever cut-off is used. This heuristic therefore always falsely predicts that the coordination ambiguity in this sentence is innocuous.

Results

The results we obtained on our dataset using the Collocation Frequency heuristic are shown in Figure 6-5. As can be seen, the heuristic performs similarly for a range of the collocation frequency ratios shown here. The precision over this range is more than 6 percentage points above the 54.3% baseline. F-measure of 3 percentage points above the 55.8% baseline and recall of 18.7% are achieved here. At lower minimum ratios, the likelihood of finding the nearer head word collocated with the modifier is not many times more than the likelihood of finding the further head word collocated with it. Both head words are readily modified by the modifier. The precision here, at ratios 2 and 3, indicates that such low minimum ratios are less conclusive indicators of innocuous ambiguity. Recall, however, is considerably better here than at higher minimum ratios. At the other extreme, very high ratios capture cases where collocations between the further

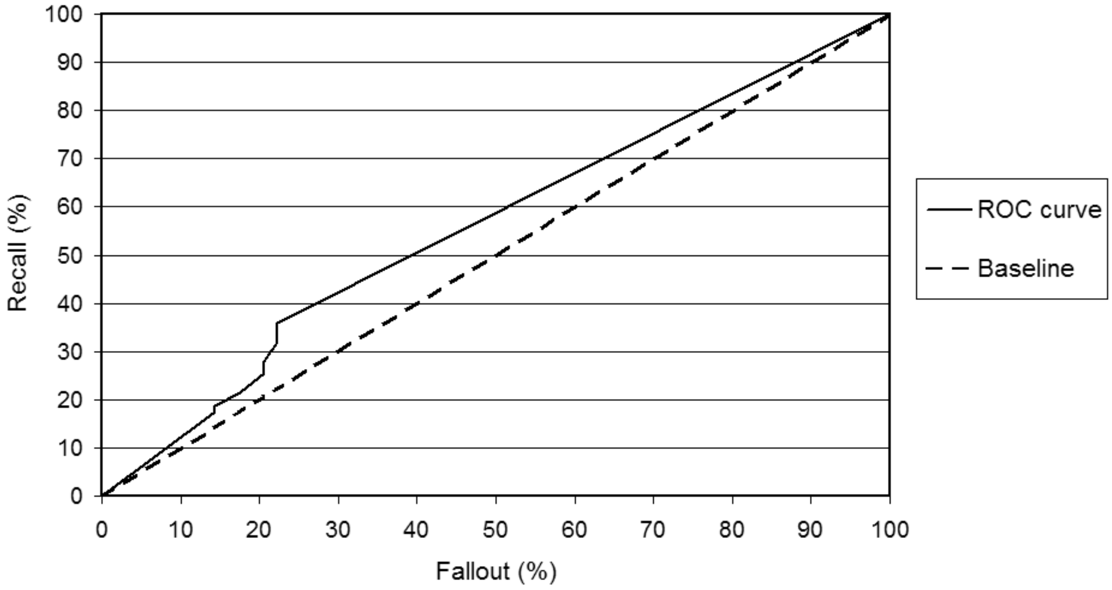


Figure 6-6: Collocation Frequency heuristic ROC curve

head word and the modifier are very infrequent or non-existent. Such cases are rare, and the poor precision here may be due to sparseness.

When implementing cross-validation, the f-measure is always maximised when a minimum ratio of between 4 and 7 (inclusive) is used. We use 4 as the cut-off for all the iterations of the cross-validation exercise.

Evaluation

The performance of this heuristic is modest. It is not so convincing a predictor of innocuous ambiguity. This may be partly due to the fact that collocation frequencies are not easily measurable for some types of coordination in our dataset. This has been explained in Section 5.6.3 on page 129. However, this is the only heuristic that successfully predicts coordination-last readings. It is therefore a vital contribution to our overall ability at predicting innocuous ambiguity.

The ROC curve for this heuristic is shown in Figure 6-6. The area under the ROC curve is 56.0%, the baseline being 50%. The predictive power of this heuristic is therefore

6 percentage points better than that of a test with no ability at predicting innocuous ambiguity. This indicates that, without training or tuning, collocation frequency is weaker than both coordination matching and distributional similarity as a diagnostic test for innocuous coordination ambiguity. However, this still indicates that the hypothesis upon which the Collocation Frequency heuristic is based has some validity. It demonstrates that if a modifier is much more frequently collocated in the corpus with the coordinated head word that it is nearest to, than it is with the further head word, then it is more likely to form a syntactic unit with only the nearest head word. This implies that a coordination-last reading is the most likely. If that likelihood is sufficiently strong, the ambiguity will be innocuous.

The ROC curve is always above the baseline. This indicates that the heuristic is consistently a good predictor, albeit not a strong one, of innocuous ambiguity over the cut-offs we have considered.

The performance of the Collocation Frequency heuristic raises a question concerning the potential overlap of our heuristics. The fact that recall is highest at very low minimum ratios, the cut-off of 2, has an important implication for us. At these low ratios, many pairs of head words for which the heuristic gives a positive result are collocated with the modifier in not greatly dissimilar frequencies. This represents a degree of distributional similarity between those headwords. This is because the modifier is an aspect of the *context* of the head words. At even lower ratios, even stronger distributional similarity would be captured. The metric, based on only one form of modification, is however much weaker than that used by a distributional thesaurus. Nevertheless, we would not consider using such low ratios for this heuristic. The heuristic would then be in danger of measuring something completely different.

6.2.4 Morphology

Here we present the results achieved by the Morphology heuristic on our dataset. This heuristic was introduced in Section 5.6.4. We present a worked example of its effectiveness when nocuous ambiguity has been determined using the Weighted Method. We then evaluate the heuristic and the hypothesis underlying it.

Example

We attempt to match the trailing characters of the conjuncts' head words in order to compare their morphology. As the cut-offs for this heuristic, we use integer values representing the numbers of trailing letters of the head words. We use the following example from our dataset to illustrate this heuristic:

> *It cannot function with the proper installation and configuration*

Of the 17 judges, 2 judged this sentence to be ambiguous, 13 judged it to be coordination-first, nobody judged it to be coordination-last and 2 people entered no response. As 13 is greater than 2, it is not an acknowledged ambiguity. Also, as $0/13 = 0$, it is not an unacknowledged ambiguity. On neither count is this ambiguity nocuous.

The trailing characters of *installation* match those of *configuration* up to a maximum of 5. The heuristic therefore predicts innocuous ambiguity for this sentence up to a cut-off of 5. The heuristic therefore correctly predicts that the coordination ambiguity in this sentence is innocuous, for all cut-offs up to 5.

Results

The results that we obtained using the Morphology heuristic on our dataset are shown in Figure 6-7. As can be seen, maximum precision is achieved at a cut-off of 5. Here it is more than 45 percentage points above the 54.3% baseline, though recall is only 2.7% here.

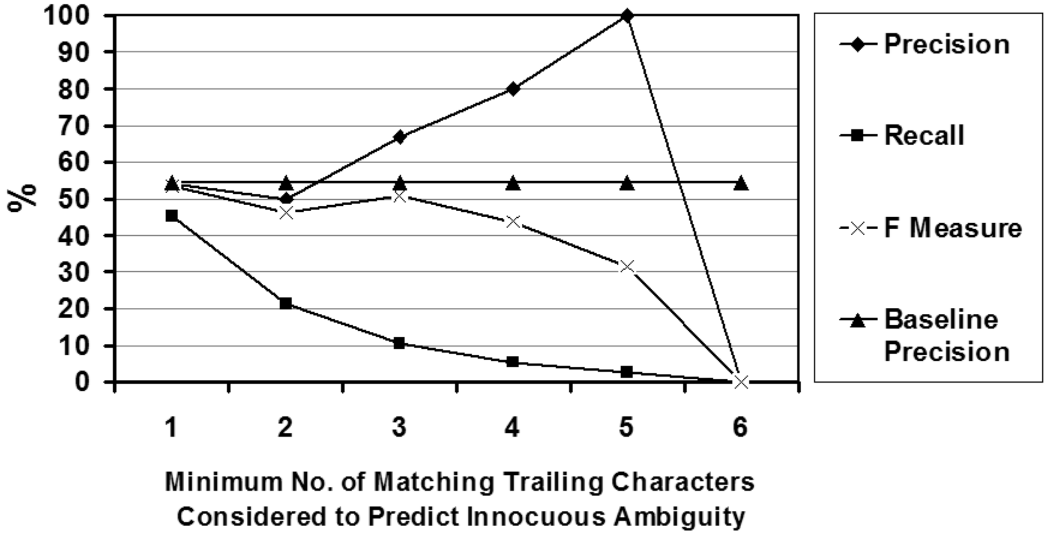


Figure 6-7: Morphology heuristic

More useful recall is achieved when only 1 or 2 matching trailing characters are considered indicative of innocuous ambiguity. However, precision is worse than the baseline here. F-measure, even using our weighting in favour of precision, at no point exceeds its 55.8% baseline. As anticipated, no similarities between trailing characters is captured when more than 5 are considered.

When implementing cross-validation, f-measure is maximised when 5 is the cut-off for 9 of the iterations in the cross validation exercise; 3 is the cut-off for the remaining one.

Evaluation

The performance of this heuristic shows that it has only limited power at predicting innocuous ambiguity. It can be a reliable indicator though. This happens when similar morphology has been captured with some certainty: when comparatively large numbers of trailing characters match. Data is sparse when this is the case, however. The fact that no predictions are made when still more trailing letters are considered confirms the assumptions made in our analysis of English morphology. The fact that the f-measure metric we use never exceeds the baseline is disappointing. However, this does not neces-

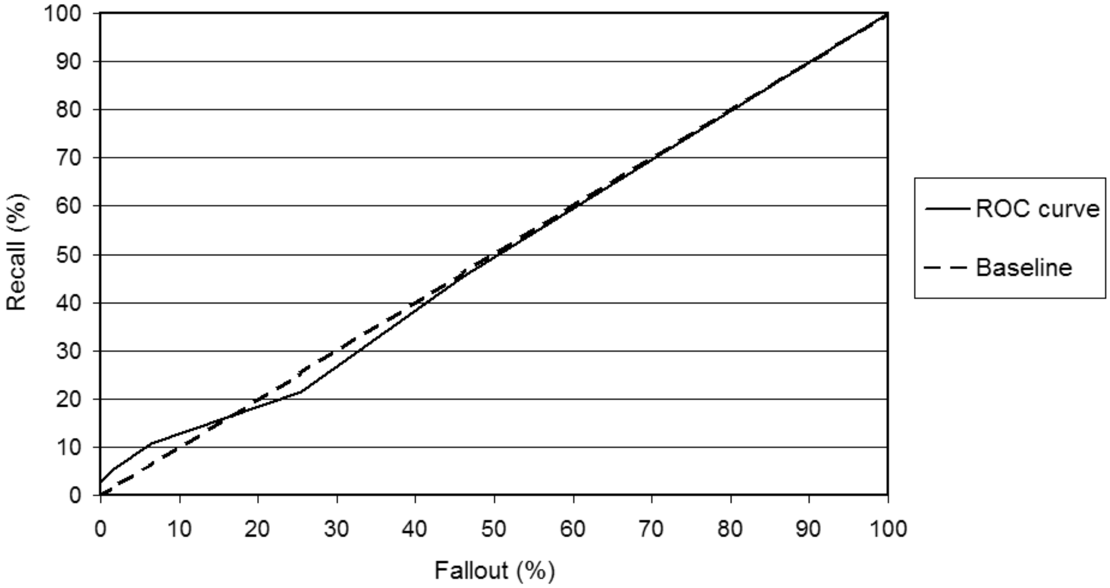


Figure 6-8: Morphology heuristic ROC curve

sarily prove that the heuristic is of no use to us. The high precision obtainable may be a useful contribution when this heuristic is used in combination with others.

The ROC curve for this heuristic is shown in Figure 6-8. The area under the ROC curve is 49.6%. This is actually slightly less than the 50% baseline. This indicates that, without training or tuning, morphology is not a valid diagnostic test for innocuous coordination ambiguity. We have not validated the hypothesis made for this heuristic. Namely, we have not proven that similarity between the morphology of the conjuncts' head words indicates that they form a syntactic unit. We can neither therefore infer that this indicates innocuous ambiguity.

However, part of the ROC curve is above the baseline. This confirms that the heuristic can be an effective predictor in some circumstances.

6.2.5 Phrase Length

Here we present the results achieved by the Phrase Length heuristic on our dataset. This heuristic was presented in Section 5.6.5. We evaluate its efficacy as a prediction metric and the hypothesis underlying it.

Results

In our dataset the difference in the lengths of the conjuncts ranges from 0 to 2. We do not consider 0, as this indicates the inverse of our hypothesis that differences in phrase length indicate innocuous ambiguity. There are therefore too few cut-off points to warrant a graph of the performance of this heuristic. At the cut-off of 2, there are two true positives and no false positives. This gives maximum precision but negligible recall and low f-measure. At the cut-off of 1, precision of 57.1% and recall of 21.3% are achieved. The precision is 2.8 percentage points above the baseline of 54.3%. The f-measure at this point is 52.0%. This is 3.8 percentage points *below* the baseline of 55.8%.

Evaluation

The performance of this heuristic gives no certainty that it can contribute to the predictive power of our combined heuristics, but might have some validity when lengths of conjuncts differ by more than a single word.

The ROC curve for this heuristic is shown in Figure 6-9. The area under the ROC curve is 49.2%, 0.8 percentage points below the 50% baseline. This indicates that, without training or tuning, phrase length difference is not a valid diagnostic test for innocuous ambiguity in coordinations. We have not validated the hypothesis made for this heuristic. We have not shown that numbers of words in a coordination's conjuncts affects how that coordination is read. We can therefore not infer that this indicates innocuous ambiguity.

However, part of the ROC curve is above the baseline. This indicates that this heuristic may be an effective predictor in some circumstances.

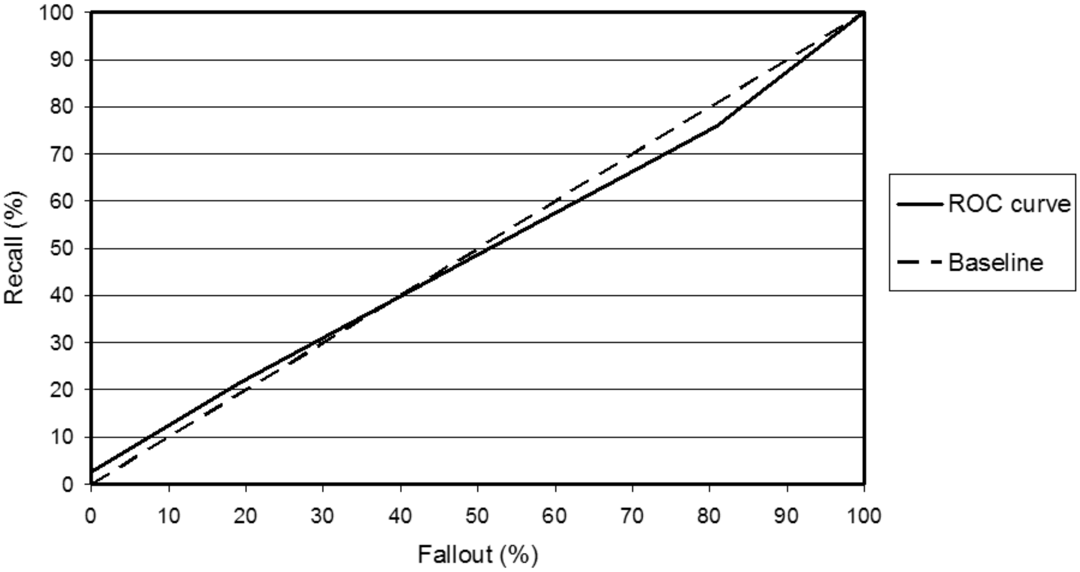


Figure 6-9: Phrase length heuristic ROC curve

6.2.6 Noun Number

Here we present the results achieved by the Noun Number heuristic on our dataset. This heuristic was presented in Section 5.6.6. We then evaluate its efficacy as a prediction metric and the hypothesis underlying it.

Results

We only have one cut-off point for this heuristic. This is agreement between the number of the nouns that are head words of the conjuncts. We do not therefore present a graph of this heuristic's performance. At the agreement cut-off, precision of 54.0%, recall of 85.7% is achieved. This precision is 0.3 percentage points *below* the baseline. F-measure of 55.2% is achieved, which is 0.6 percentage points *below* the baseline.

Evaluation

The performance of this heuristic gives no indication that it can predict innocuous ambiguity. The ROC curve for this heuristic is shown in Figure 6-10. The area under the

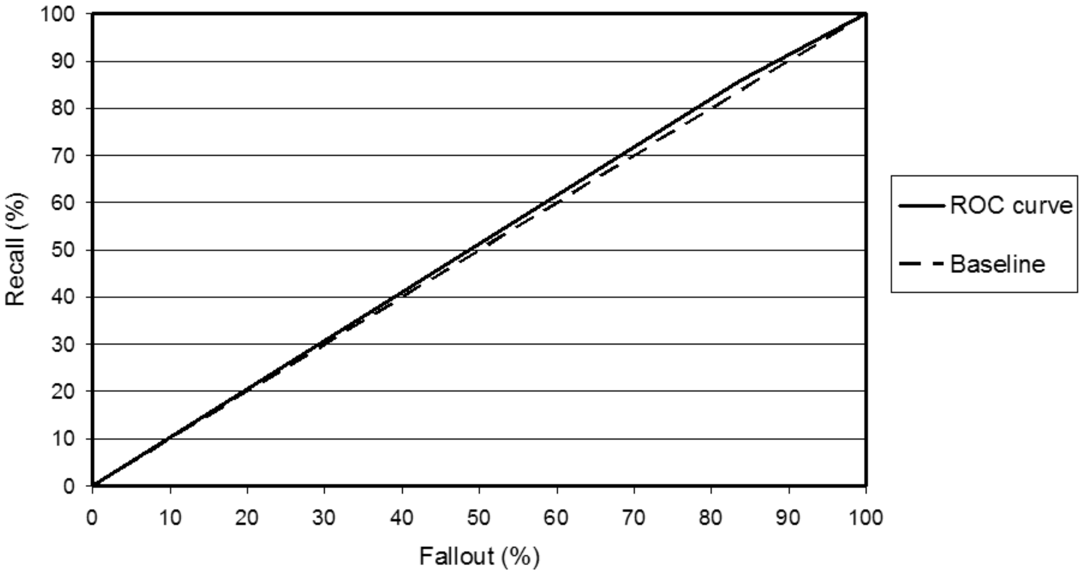


Figure 6-10: Noun number heuristic ROC curve

ROC curve is 51.0%. This gives only 1 percentage point over a test with no predictive power. This suggests that the noun number agreement we have captured is not a reliable diagnostic test for innocuous ambiguity in coordinations. We have not therefore been able to validate the hypothesis upon which this heuristic based. Namely, we have been unable to show that noun number agreement between the head words of conjuncts indicates that they should be read coordination first. We cannot say therefore that innocuous ambiguity can be predicted in this way. In the same way, have not been able to substantiate our interpretation of Resnik's (1999) claim that noun number is an “important” indicator of preferred readings of coordinations.

The ROC curve shows no significant deviation from the baseline. We cannot conclude that this heuristic can be trained to be a useful predictor.

6.2.7 Mass/Count

Here we present the results achieved by the Mass/Count heuristic on our dataset. This heuristic was presented in Section 5.6.7. We then evaluate its efficacy as a prediction

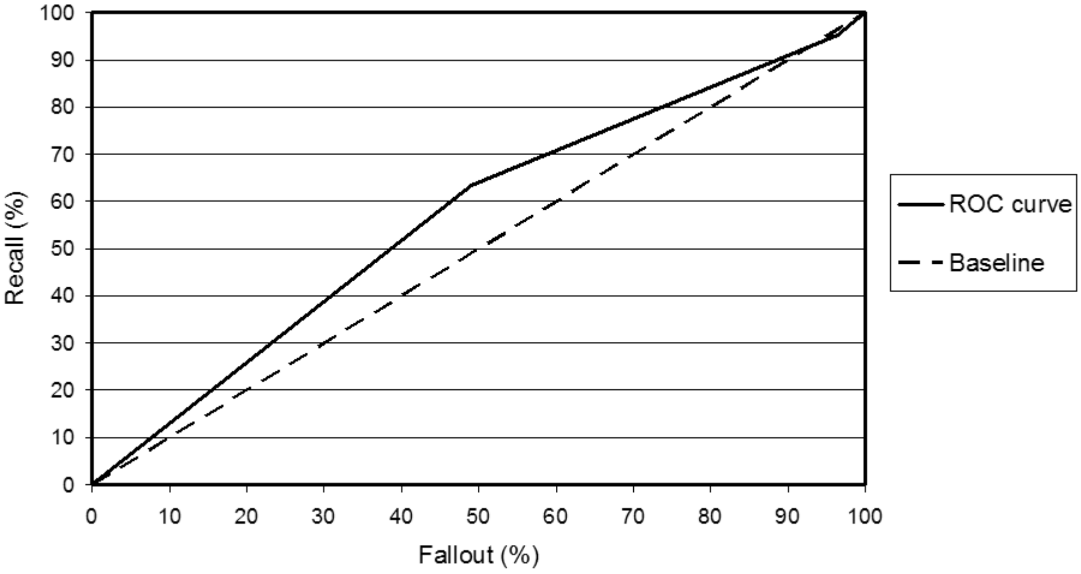


Figure 6-11: Mass/Count heuristic ROC curve

metric and the hypothesis underlying it.

Results

We only have two cut-off points for this heuristic: unequivocal agreement and equivocal agreement. The former refers to the situation where the head words of the conjuncts are either definitely both mass or definitely both count. The latter refers to the situation where there is the possibility that one or both of the head words can be either mass or count. As there are only two cut-off points, it is not instructive to give a graph of this heuristic's performance.

At the cut-off of unequivocal agreement, this heuristic has precision of 59.7% and recall of 63.5%. This precision is 5.4 percentage points above the baseline. F-measure of 59.9% is achieved. This is 4.1 percentage points above the f-measure baseline.

At the cut-off of equivocal agreement, this heuristic has precision of 53.1% and recall of 95.2%. This precision is 1.2 percentage points below the baseline. F-measure of 54.5% is achieved. This is 1.3 percentage points below the f-measure baseline.

Heuristic (CV = cross-validation exercise)	Recall (%)	Precision (%)	Precision Percentage Points over Baseline	F-Measure β=0.25 (%)	F-Measure Percentage Points over Baseline	ROC Curve Percentage Points over Baseline
Baselines	100	54.3	-	55.8	-	-
Coordination Matching	40.0	75.0	20.7	71.3	15.5	12.2
Distributional Similarity	20.0	75.0	20.7	64.6	8.8	9.4
Collocation Frequency	18.7	60.9	6.6	53.7	-2.1	6.0
Morphology	2.7	100	45.7	31.8	-24.0	-0.4
Phrase Length	21.3	57.1	2.8	52.0	-3.8	-0.8
Noun Number	85.7	54.0	-0.3	55.2	-0.6	1.0
Mass/Count	63.5	59.7	5.4	59.9	4.1	6.7
Combined Heuristics (pre CV)	58.7	71.0	16.7	70.1	14.3	
Combined Heuristics (post CV)	56.3	62.2	7.9	61.6	5.8	

Table 6.4: Performance of our Heuristics using Weighted Method

Evaluation

The performance of this heuristic shows that unequivocal agreement, with regard to nouns being mass or count, has some power at predicting innocuous coordination ambiguities.

The ROC curve for this heuristic is shown in Figure 6-11. The area under the ROC curve is 56.7%. This gives 6.7 percentage points over a test with no predictive power. This indicates that mass/count agreement for nouns is a valid diagnostic test, albeit not a particularly reliable one, for innocuous ambiguity in coordinations. However, this still indicates that the hypothesis upon which this heuristic is based has some validity. It demonstrates that if the head nouns of conjuncts are either both mass or both count, then they will more likely be read coordination first. If that likelihood is sufficiently strong, the ambiguity will be innocuous.

The ROC curve is always above the baseline. This indicates that the heuristic is consistently a good predictor, albeit not a strong one, of innocuous ambiguity.

6.2.8 Combined Heuristics Using Weighted Method

To combine our heuristics using the Weighted Method, we use a simple disjunction approach. This states says that if any of the heuristics predicts an ambiguity to be innocuous, then that ambiguity is innocuous. This is an appropriate way of combining the heuristics as we hypothesise that they all predict something different and we have endeavoured, using the Weighted Method, to maximise this predictive power. To use majority voting, whereby several of the heuristics must agree, would therefore be inappropriate. For the same reason, and to an even greater extent, it would be inappropriate to use a conjunctive approach, whereby the heuristics must be unanimous to give a prediction of innocuous ambiguity. For instance, the collocation frequency heuristic attempts prediction of coordination-last readings whereas the others predict coordination-first readings: if the former was successful, it would never agree with the others. The results of the combined heuristics are presented at the foot of Table 6.4, both before and after the cross-validation exercise.

The results of the individual heuristics are also presented in Table 6.4. These are given without having been subject to cross-validation, and at the cut-off that is most commonly used when maximising the combined heuristics' performance. The areas under ROC curves are also given in Table 6.4, to aid comparison of the heuristics in terms of the validity of their underlying hypotheses.

6.2.9 Discussion

As can be seen from Table 6.4, nearly all the individual heuristics can achieve higher precision than the baseline. In some cases they considerably exceed what a test with no ability at predicting innocuous ambiguity would achieve. The precision baseline is calculated by assuming that all the coordinations are innocuous, but this is an assumption that we in no way want to make. It would allow many nocuous ambiguities to pass unnoticed. We wish precision to exceed the baseline as much as possible. It is therefore appropriate that we have sought to maximise precision of the individual heuristics at the expense of low recall.

Combining the heuristics increases the recall considerably. This indicates that their coverage of innocuous ambiguities is to some extent complementary. However, the precision of the combined heuristics is less than the precision of most of the heuristics. This indicates that the intersection of the coverages of the individual heuristics contains true positives. In other words, some sentences are judged to be innocuous by more than one heuristic. This is only to be expected. For instance, the words in a coordination found commonly in BNC may also have strong distributional similarity. The potential overlap between collocation frequencies and distributional similarity has already been discussed in Section 6.2.3 on page 159. Such factors decrease the contribution of true positive predictions to the combined heuristics' predictive capability. However, the combined heuristics achieve an f-measure score of 70.1% before implementation of cross-validation. This is 14.3 percentage points above the baseline. We use the same weighting for this f-measure as for the individual heuristics. We believe that this performance is still acceptable. It is almost as good as the best heuristic but it represents much higher recall.

In the last row of Table 6.4 are the results of the combined heuristics after application of 10-fold cross-validation. During this exercise, $\frac{9}{10}$ of the dataset has been used as training data to determine optimal cut-offs. The heuristics are then run, using these cut-offs, on the remaining $\frac{1}{10}$, which is the test data. This process is iterated 10 times, so all data is used as test data. Because of this, heuristics are run on test data using cut-offs which are often sub-optimal for that data. This inevitably reduces performance. The f-measure drops to 61.6, which is 5.8 points above the baseline. The recall, however, is still much higher than for any of the individual heuristics. This performance shows that we have partly achieved what we set out to do: improving recall by combining our heuristics. However, the decreased precision makes the overall f-measure result disappointing.

For the performance figures obtained after the cross-validation exercise has been performed, we analyse the variance of the performance over the 10 folds. This is presented

	Recall (%)	Recall Standard Deviation	Precision (%)	Precision Standard Deviation	F-Measure $\beta = 0.25$ (%)	F-Measure Standard Deviation
Combined Heuristics after Cross-Validation	56.3	20.0	62.2	25.5	61.6	24.7

Table 6.5: Standard Deviations of Combined Heuristics after Cross-Validation

in Table 6.5. We use *standard deviations* to represent the variance of the performances in terms of percentages. The *means* from which the standard deviations deviate are the (averaged) performance figures, which are re-presented there. It can be seen that there is great variance between the folds, for recall and, even more markedly, for precision and f-measure. Considering precision, the interval between one standard deviation below the mean (36.7%) and one standard deviation above the mean (87.8%) is more than half the entire range of possible values. This represents a significant variance. It reflects the fact that the collections of lines contained in the folds can be very different from each other: the sub-datasets in the folds are highly heterogeneous. This is partly due to the small size of the dataset, resulting in small fold sizes. A less thorough cross-validation technique — using fewer folds — might give lower standard deviations, but might be less reliable as a technique for avoiding bias. Our 10-fold cross-validation randomizes the data so that a variation in a certain subset of the data (a fold) does not gain prominence in the overall analysis. Therefore, although the standard deviations are statistically significant, we do not consider this heterogeneity to be a problem.

The Weighted Method used here tends to weight unacknowledged ambiguities more heavily than acknowledged ambiguities. Nocuous ambiguity is therefore more likely to result from the former than from the latter. The method implements the idea that unacknowledged ambiguities are the ultimate cause of nocuous ambiguity. They are therefore more immediately dangerous than acknowledged ambiguities. The weighting we use means that each nocuous ambiguity has higher than average unacknowledged ambiguity. Such ambiguities can therefore be considered to be of immediate concern. They may result in misunderstandings, and therefore in incorrect implementation of requirements. We have published work using this method (Chantree, Nuseibeh, de Roeck, and Willis 2006), where it has received the approval of RE practitioners.

6.3 Predictions Using Flexible Method

Here we present the performance of our combined heuristics at predicting consensus human judgements obtained using the Flexible Method. To reiterate, this method treats unacknowledged and acknowledged ambiguity equally when determining whether an ambiguity is nocuous or innocuous. We implement ambiguity thresholds as a part of this process, which allows us to show the impact of heuristics at difference levels of intolerance to ambiguity.

We present the performance of the seven heuristics used only in combination. (The individual heuristics are introduced in Section 6.2. Their individual performances at a fixed threshold are discussed there as an demonstration of their effectiveness.) For all the results presented here, we fit our data to a logistic regression model. The WEKA machine learning package is used for this. We do not give worked examples as this software does not require user interaction or offer the necessary transparency. We use the software to determine performance in terms of the *accuracy* of predicting both innocuous *and* nocuous ambiguities, as explained in Section 5.7. This is done as an alternative to our approach so far with the Weighted Method in Section 6.2. The aim there was validate the heuristics, and the hypotheses that they are based on, by trying to predict innocuous ambiguity. The aim here is to show how nocuous ambiguity can be distinguished from innocuous ambiguity. The prediction of both types is therefore relevant.

We first introduce the baselines used for this exercise. Then we discuss the heuristics'

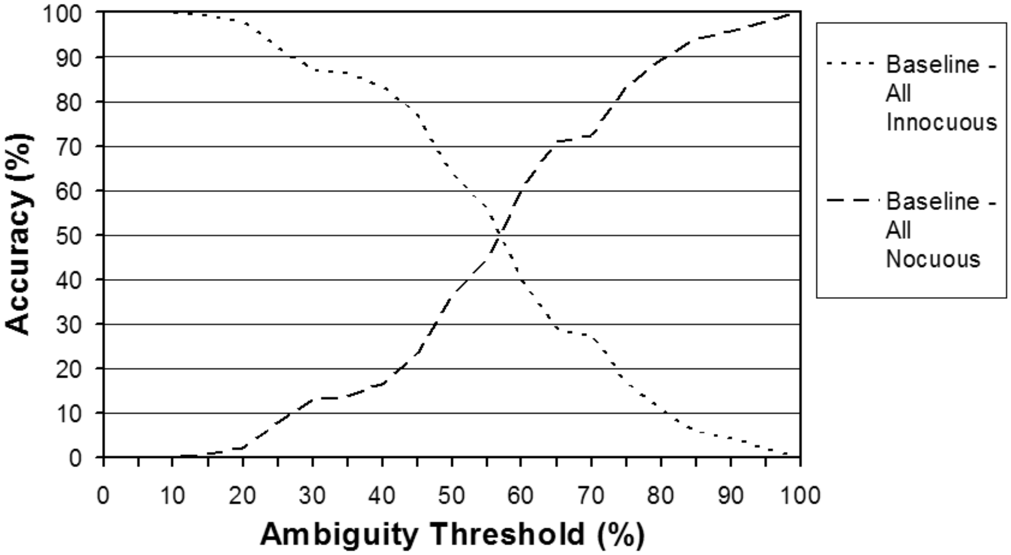


Figure 6-12: Baselines for Combined Heuristics' Accuracy at Discriminating Nocuous from Innocuous Ambiguity

performance in relation to these baselines. This is followed by a discussion of this exercise and of the implications of using ambiguity thresholds.

6.3.1 Baselines

Baselines (or *lower bounds*) are most relevant for classification tasks such as ours where the evaluation measure is accuracy (Manning and Schütze 1999). We use two baselines when evaluating the accuracy of our combined heuristics in this exercise, as shown in Figure 6-12. The baselines represent the performance of the simplest possible algorithms. These are the assignment of one of the two possible outcomes to all instances. One is found by determining the number of innocuous ambiguities correctly predicted if all ambiguities are considered innocuous. The other is found by determining the number of nocuous ambiguities correctly predicted if all ambiguities are considered nocuous. Both baselines are needed because we are measuring accuracy of classifying both nocuous and innocuous ambiguities. We target improvement against the higher of the two baselines at any given threshold.

It can be seen that the baselines tend towards 0% and 100% at the extreme right of Figure 6-12. At ambiguity thresholds approaching 100% (complete *intolerance*) there are a vanishingly small number of innocuous ambiguities. Let us suppose we are implementing a threshold of 90%. If only 2 of our 17 judges disagree with the majority non-ambiguous (i.e. coordination-first or coordination-last) opinion, then the ambiguity will still be classed as nocuous (15/17 = 88.2%). At the extreme left of the graph the baselines also tend towards 0% and 100%. Here there is a vanishingly small number of nocuous ambiguities, as *tolerance* to ambiguity tends towards being complete. Let us suppose we are implementing a threshold of 20%. Only 4 out of 17 judgements need to be in favour of one of the non-ambiguous opinions (coordination-first or coordination-last) for that ambiguity to be classed as innocuous (4/17 = 23.5%). Remember that acknowledged ambiguity judgements and minority non-ambiguous judgements are considered on a par. In the latter scenario, 3 could be minority non-ambiguous judgements and 10 could be acknowledged ambiguity judgements.

In the centre of Figure 6-12, the baselines cross each other when the accuracy of distinguishing nocuous from innocuous ambiguity is approximately 50%. This figure is the lowest baseline available. It is therefore the one we most wish to target, as distinguishing between nocuous and innocuous ambiguity is most difficult to achieve in this region. Indeed, prediction at very high or very low tolerance levels are easily accommodated by simpler methods, as reflected by the baselines there. The threshold giving the minimum baselines for our dataset is approximately 57%. At least 10 out of 17 judgements need to be in favour of the majority non-ambiguous opinion for an ambiguity to be classed innocuous (10/17 = 58.8%).

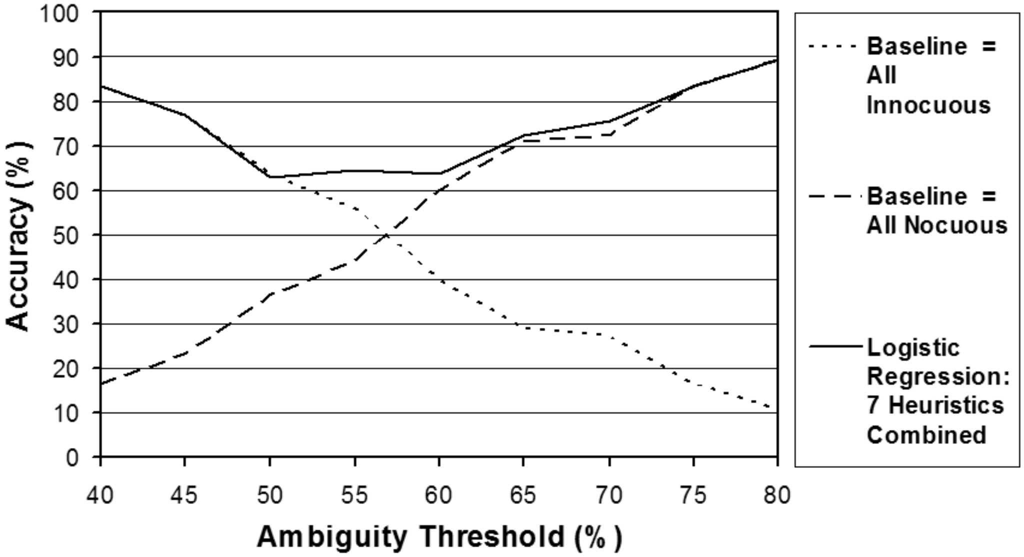


Figure 6-13: Combined Heuristics' Accuracy at Discriminating Nocuous from Innocuous Ambiguity using Logistic Regression

6.3.2 Performance of Combined Heuristics

The performance of the combined heuristics, along with the baselines, is presented in Figure 6-13. Only a range of thresholds covering where the former deviate from the latter is presented. The combined heuristics' performance is the same as the higher performing baseline at all ambiguity thresholds beyond this range. It is recognised that it can be hard to outperform high baselines (Manning and Schütze 1999). The last few percentage points are notoriously hard to achieve when performance at over 90%, for instance, is already achievable. This is the case with our baselines at extreme ambiguity thresholds. The combined heuristics' performance in relation to the baselines is therefore not so surprising at these thresholds.

However, Figure 6-13 shows that our combined heuristics can satisfactorily outperform the baselines in the middle region. They perform best between ambiguity thresholds of 50% and 60%, with their maximum improvement in performance where the baselines intersect. This can be expressed alternatively as 14 percentage points above the baselines or as 28% improvement on the baselines — the result presented in the abstract of this thesis . Between ambiguity thresholds of 60% and 75% the performance is less marked but still noteworthy. The average improvement in performance in this range can be expressed alternatively as 2.7 percentage points above the baseline, or 4.1% improvement on the baseline. Between the thresholds of 40% and 50%, no increase in performance above the baselines is witnessed.

6.3.3 Discussion

The performance of the combined heuristics between 50% and 60% shows they can have significant ability at distinguishing nocuous from innocuous ambiguities at a range of ambiguity thresholds appropriate for some tasks. We anticipate that this middle range of thresholds will in fact be useful for most tasks: the need to identify as many nocuous ambiguities as possible is balanced with the desire to avoid wasting effort by also identifying innocuous ones. At a threshold of 50%, it is true that ambiguities likely to lead to misunderstandings may be counted as innocuous. This may be the case if the number of judges concurring with the majority non-ambiguous verdict is only one more than the number who do not. However, in our experience, it is unlikely that all of the latter will concur with the minority non-ambiguous option. Instead, we find that the majority of these dissenters exercise caution. In our dataset, an average of only 18.7% judges assign the minority non-ambiguous judgement. The other 81.3% assign the "ambiguous" judgement. The average deviation from the former average is 14.3%. In only 4 cases do the numbers of judgements in favour of the minority non-ambiguous option exceed those in favour of the "ambiguous" option. This indicates that caution is generally exercised, with readers expressing any doubts they have by acknowledging ambiguity. Ambiguity thresholds of 50% and 60% are therefore more appropriate than they would be if they represented more of the (more dangerous) unacknowledged ambiguity.

The performance of the combined heuristics at thresholds higher than 60% is less conclusive. The improvements over the baselines show that the combined heuristics do have some ability at distinguishing nocuous from innocuous ambiguity at these levels. The fact that this is witnessed over a range of thresholds gives credence to this trend. However, it is harder to outperform the baselines here than at lower thresholds. We would wish the performance over the baselines to be higher at these thresholds. This wish is made stronger by that fact that thresholds in this range are appropriate for more safety-critical tasks. We desire high accuracy for such tasks to prove the worth of our techniques.

The combined heuristics show no ability at distinguishing nocuous from innocuous ambiguities at thresholds below 50%. However, this is not of major concern to us. It is unlikely that such thresholds would be used, as they can allow highly nocuous ambiguities to be passed as innocuous. At these low thresholds, significant unacknowledged ambiguity may be observed for ambiguities that are judged as innocuous. The aforementioned caution shown by dissenters mitigates against this somewhat, but this effect will be diluted at these lower thresholds. For example, an ambiguity judged by 8 people to be coordination-first, by 7 to be coordination-last and by 2 to be ambiguous has a certainty of 47.1%. It will therefore be considered innocuous if the ambiguity threshold is set at any percentage lower than this. This is despite the fact that it is an ambiguity that is unacknowledged by a lot of people, and so clearly may lead to misunderstandings.

Figure 6-14 shows the percentages of coordination-first and coordination-last interpretations at different ambiguity thresholds. These constitute innocuous ambiguity for our test case ambiguity. They are compared against the percentages of nocuous ambiguities, using the Flexible Method, at these thresholds. This figure indicates the difficulty of differentiating innocuous from nocuous ambiguity at the extremes of ambiguity tolerance, as there are very few instances of one type. At these extremes, any incorrect

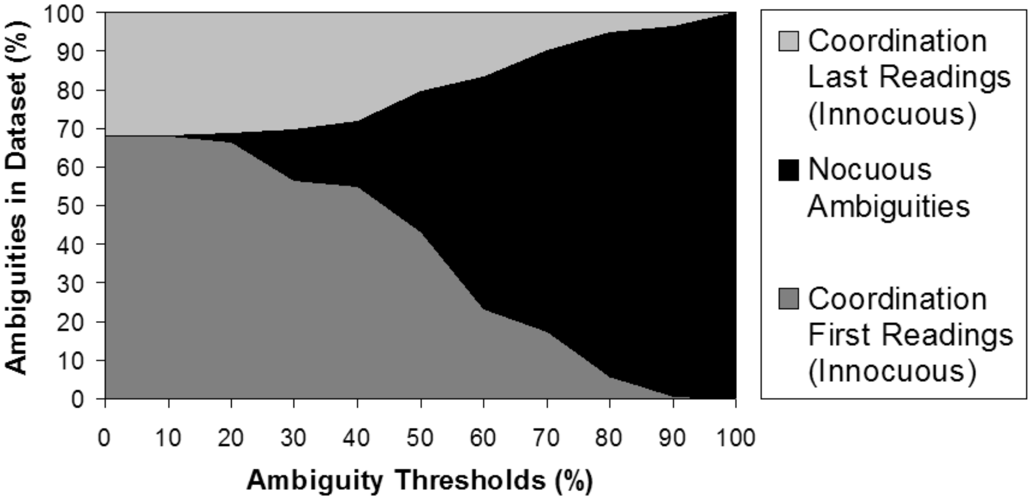


Figure 6-14: Proportions of Coordination-First and Coordination-Last Interpretations at Different Ambiguity Thresholds

predictions will have a large influence on accuracy in relation to the number of correct predictions (which can never be large). Figure 6-14 also demonstrates that the numbers of coordination-first and coordination-last interpretations are in a roughly equal ratio at any given ambiguity threshold. This indicates that ambiguities normally given either of these two types of interpretation are approximately equally susceptible to becoming nocuous with increased intolerance to ambiguity. The judgements dissenting from one interpretation (either acknowledging ambiguity or choosing the other non-ambiguous interpretation) become significant at an equally increasing rate.

In terms of our model of ambiguity, we wish to analyse specifically the relationship between the ambiguity threshold and the distribution of nocuous and innocuous ambiguities. Figure 6-15 is a simplification of Figure 6-14. It shows more clearly the relationship between the ambiguity threshold and the proportions of nocuous and innocuous ambiguities in our dataset (using the Flexible Method). In the middle of the graph, this relationship is approximately linear. As discussed previously, these are ambiguity thresholds in which we are most interested. For this range, then, the consensus about whether an am-

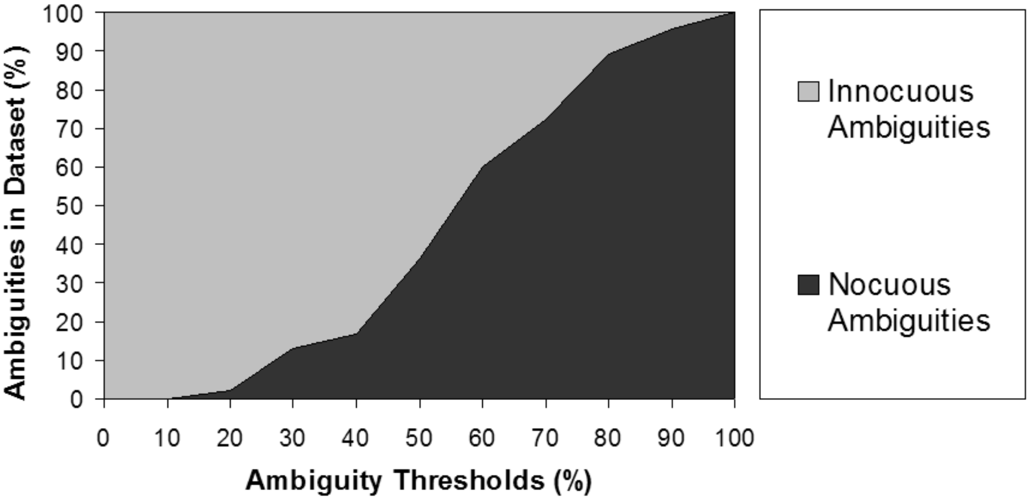


Figure 6-15: Proportions of Nocuous and Innocuous Ambiguities at Different Ambiguity Thresholds

biguity is nocuous varies approximately linearly with relation to ambiguity intolerance. This confirms the relationship, discussed above in regard to Figure 6-14, between nocuous ambiguities and those with a preferred interpretation.

This middle range of Figure 6-15 shows consistent increase of sensitivity to ambiguity among our judges. They have different levels of sensitivity (or intolerance), but these appear to be distributed along a continuum represented by this range. We may surmise from this that random (or *rogue*) judgements are not having a great effect over this range. (They will, however, have a greater effect at extreme ambiguity thresholds where judgements on one type are more sparse.) This says something about the ease with which our heuristics are able to be predict nocuous ambiguities using middle-ranging ambiguity thresholds. The data appears to be relatively trustworthy over this useful range. More trust can therefore be placed in the heuristics' performance here.

6.4 Unacknowledged Ambiguities

Here we present and discuss the performance of the seven combined heuristics when distinguishing *unacknowledged ambiguities* from innocuous ones. The former replace the nocuous ambiguities we have tried to find using the Weighted and Flexible Methods. Here we are trying to distinguish only those ambiguities that are of *immediate* concern, according to our judges. In this method, all acknowledged ambiguity judgements are ignored. Only minority non-ambiguous judgements count towards establishing unacknowledged ambiguity. We wish to determine what proportions of unacknowledged ambiguity are in evidence with different intolerances to ambiguity. We therefore experiment with a range of ambiguity thresholds, as with the Flexible Method in Section 6.3. Also, for the same reasons given in that section, we employ a logistic regression model and measure the accuracy of distinguishing the two types of ambiguity.

We first discuss the baselines used for this exercise. Then we present the heuristics' performance in relation to these baselines. (The individual heuristics are introduced in Section 6.2). This is followed by a discussion of this exercise and of the problems of predicting solely unacknowledged ambiguity.

6.4.1 Baselines

The two baselines for this exercise are shown in Figure 6-16. As when predicting nocuous ambiguity, they tend towards 0% and 100% at both extremes of the graph. However, their shape is decidedly different. At the right side of the graph unacknowledged ambiguity tends towards being non-existent. Here there are a vanishing small number of minority non-ambiguous judgements on the sentences but many sentences with very few of these judgements. In fact, at threshold as high as 90%, approximately half the ambiguities are shown to be innocuous. This represents the fact that many sentences have only a

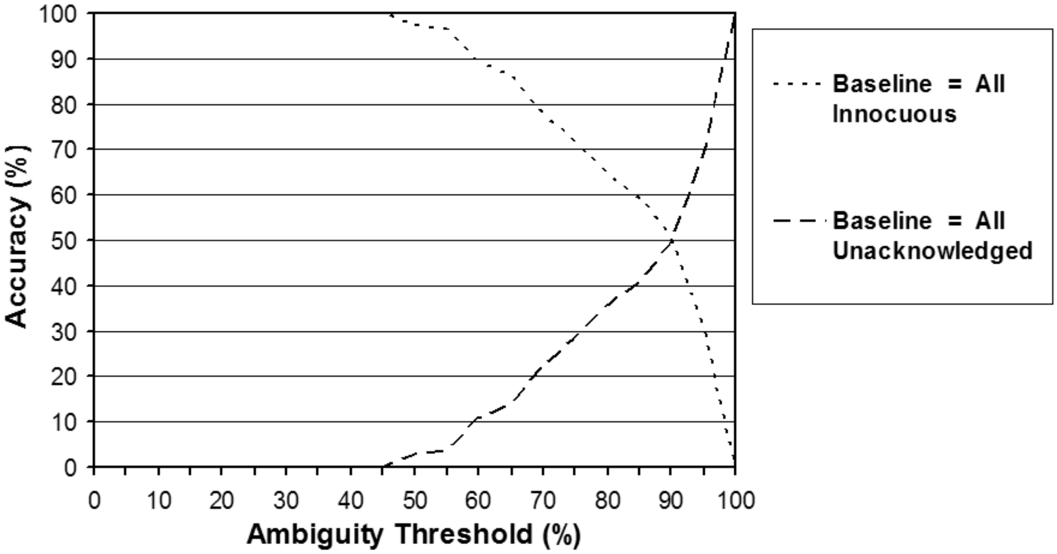


Figure 6-16: Baselines for Combined Heuristics' Accuracy at Discriminating Unacknowledged from Innocuous Ambiguity

single minority non-ambiguous judgement or none at all. On the left side of the graph, there are a very few unacknowledged ambiguities at a threshold of 50%. Here there are equal numbers of coordination-first and coordination-last readings on a sentence. At lower ambiguity thresholds, however, nocuousness cannot increase. This is because the number of minority judgements can never be greater than the number of majority judgements. The range of thresholds below 50% is therefore not of interest to us.

6.4.2 Performance of Combined Heuristics

The performance of the combined heuristics is presented in Figure 6-17, along with the baselines. This shows their accuracy at distinguishing unacknowledged from innocuous ambiguity using our logistic regression model. This is shown over the range of ambiguity thresholds that is meaningful for unacknowledged ambiguity. The only significant increase in accuracy of the heuristics above the baselines is at an ambiguity threshold of 90%. This is where the baselines intersect. The increase can be expressed alternatively as 10.2 percentage points above the baselines or as 20.1% improvement on the baselines. At no

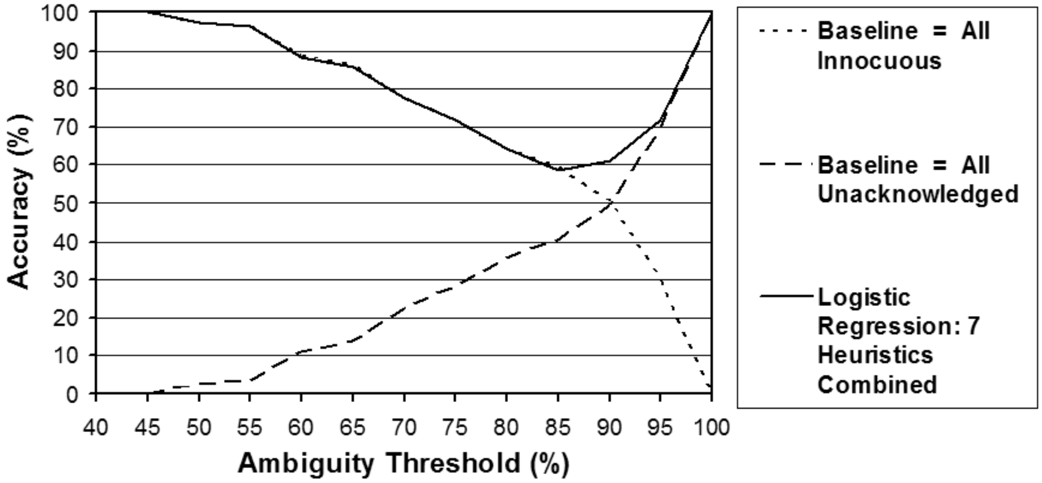


Figure 6-17: Combined Heuristics' Accuracy at Discriminating Unacknowledged from Innocuous Ambiguity using Logistic Regression

other threshold do the combined heuristics perform significantly better than the baselines, and they sometimes perform fractionally worse.

6.4.3 Discussion

The accuracy of the combined heuristics at a 90% threshold shows they can have significant ability at distinguishing unacknowledged from innocuous ambiguities. This threshold represents an intolerance to ambiguity which may be appropriate for some types of requirements. Such requirements include, for instance, those specifying safety-critical systems. These requirements can tolerate the possibility of only very occasional minority non-ambiguous judgements. However, the combined heuristics' lack of accuracy at other thresholds is not encouraging. In Section 6.3, we demonstrated that significant performance can be achieved over a low baseline. But that exercise also showed that the combined heuristics give some improvement over the baselines for a range of ambiguity thresholds. When looking solely at unacknowledged ambiguity, we are unable to improve on thresholds with baselines other than the lowest possible. We have not been able to predict unacknowledged ambiguity over a range of useful thresholds with any confidence.

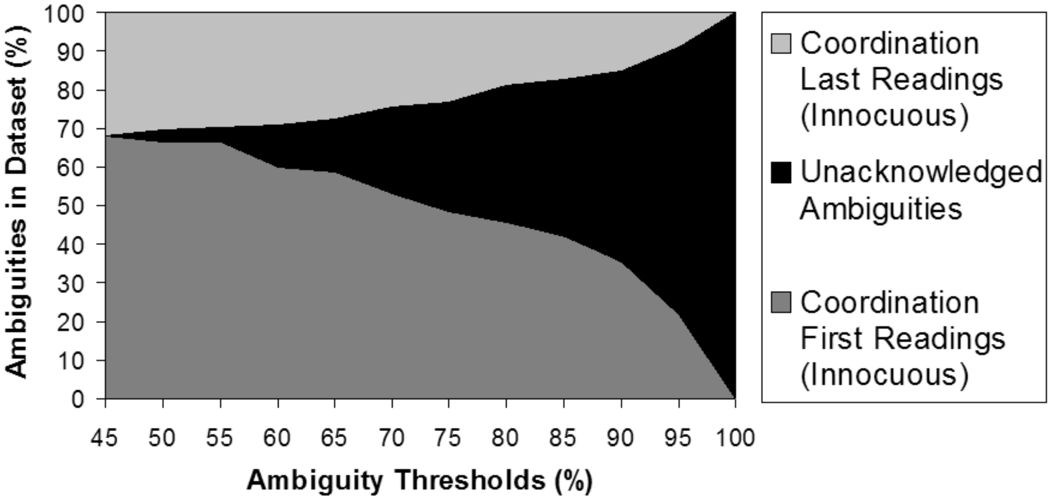


Figure 6-18: Proportions of Coordination-First and Coordination-Last Interpretations at Different Ambiguity Thresholds

Figure 6-18 shows the proportions of coordination-first and coordination-last interpretations at different ambiguity thresholds. Much of the observations made for Figure 6-14 on page 180 are also true for this figure. Similarities include the difficulties of differentiation at extreme ambiguity thresholds and the ratios of the two types of interpretation. However, the two graphs are somewhat differently shaped. This is more easily discussed by looking only at unacknowledged and innocuous ambiguities.

Figure 6-19 simplifies Figure 6-18 by plotting only unacknowledged and innocuous ambiguity. Compared to Figure 6-15 on page 181, this relationship has a more concave curve. This indicates that the number of unacknowledged ambiguities increases less rapidly than nocuous ambiguity when intolerance to ambiguity increases. This demonstrates that unacknowledged ambiguity tends not to be encountered as frequently as nocuous ambiguity. Unacknowledged ambiguity is therefore less in evidence than nocuous ambiguity regardless of one's intolerance to ambiguity. It also means that unacknowledged ambiguity is harder to characterise than nocuous ambiguity. The fewer unacknowledged ambiguities there are, signified by the judgements in our dataset, the more they are more subject

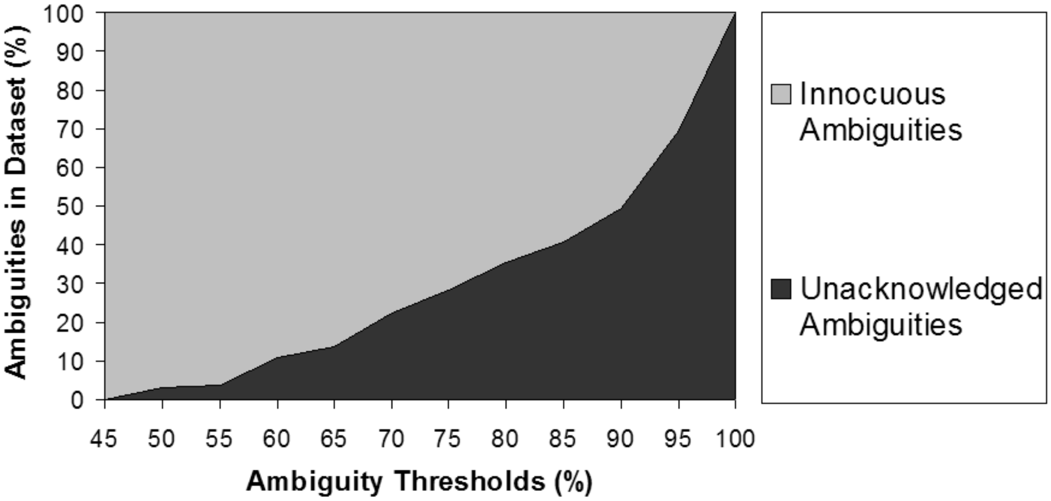


Figure 6-19: Proportions of Unacknowledged and Innocuous Ambiguities at Different Ambiguity Thresholds

to random errors by our judges. It can therefore be harder to predict unacknowledged ambiguities using our heuristics.

6.5 Summary

In this chapter we have discussed our dataset and the performance of our heuristics on this dataset. The former discussion involved characterising the corpus from which our dataset was derived. We then summarised the sentences presented to our judges and the judgements they gave on these sentences. Results were presented for each of our methods of distinguishing nocuous from innocuous ambiguity. During the presentation of the Weighted Method, each individual heuristic was discussed in terms of its implementation and its overall efficacy. Some worked examples were presented here to aid understanding of nocuous, unacknowledged and acknowledged ambiguity. Both the Flexible and Purely Unacknowledged Methods used our heuristics only in combination. These interpretation methods were implemented using ambiguity thresholds, to demonstrate our fundamental concern that intolerance to ambiguity should be flexible. The discussion of all these methods has revealed that nocuous ambiguity is easier to characterise and predict than unacknowledged ambiguity. The individual heuristics have very variable prediction capabilities, but work more effectively in combination. Nocuous ambiguity can be predicted at several ambiguity thresholds, including the one implicit in our weighted method, which are useful for RE practitioners.

Chapter 7

Conclusions

The main aims of this research have been to present and test a novel model that classifies ambiguities as being either nocuous or innocuous. Notification must be made of the former as they are likely to lead to misunderstandings, whereas the latter are likely to be easily interpreted and so no notification need take place. We have used human perception as the gold standard of how ambiguities are experienced, and thereby classified, and we have predicted this using a variety of heuristics.

From our use of human perception as a criterion for classification, we have seen that significant degrees of both acknowledged and unacknowledged ambiguity can be witnessed. We conclude that this should be of concern where misunderstandings resulting from language must be kept to a minimum, particularly in a domain such as RE where they can lead to costly mistakes. The extent of nocuous ambiguity resulting from the judgements we collected is affected somewhat by the fact that we were not able to provide the judges with much disambiguating context for the sentences. Almost certainly, more ambiguities have been judged as nocuous than would be the case if all the context was provided: our judges have more easily reached differing conclusions about the interpretation of the examples, or have more readily judged them to be ambiguous. However, even accounting for this, we believe that we have discovered sufficient differences of opinion to warrant our research. Also, a lower ambiguity threshold than might normally be preferred can be used to account for the increased numbers of nocuous ambiguities.

We draw various conclusions from our three methods of distinguishing ambiguities. Firstly, the Weighted Method of interpretation, allowing for no flexible intolerance to ambiguity, has shown that prioritising unacknowledged ambiguity is an effective and acceptable solution for real-world RE ambiguities. Using such a weighting captures the fact that unacknowledged ambiguity is a more immediate source of nocuousness than acknowledged ambiguity. Using this method we have also demonstrated the validity of the hypotheses underlying some of our heuristics. This is evidence that adds to the literature of disambiguation of coordinations.

Secondly, using the Flexible Method has shown that thresholds, representing a level of intolerance to ambiguity, can be used effectively in the classification of ambiguities as being either nocuous or innocuous. Significant improvements over simpler methods to discriminate between nocuous and innocuous ambiguity cannot be observed at all thresholds. However, the performance of our combined heuristics is considerably in excess of the baselines for some critical thresholds. It less conclusively exceeds the baselines for the other thresholds that we consider useful, but the performance there is still significant and encouraging.

Our third method is concerned only with unacknowledged ambiguities. They prove harder to classify than nocuous ambiguities (which may be acknowledged to a large extent). Comparing this method of interpretation with the others we have used, we believe that it is our definition of *nocuous* ambiguity that is more suitable for practical purposes.

The performance of our individual heuristics gives some valuable insights into how the data they utilise can be used to address the problem of ambiguity in text. A surprising number of coordinations and collocations in our specialised corpus were also found in the BNC, suggesting that generic language information, from a large enough source, can provide effective and readily-available solutions. It is also interesting that the morphology heuristic was effective at predicting preferred readings of ambiguities, at some cut-off points, and we believe that this is an aspect of language that might be further investigated as a disambiguation technique.

We conclude that our results are generalisable to a certain extent. Our use of a generic corpus (the BNC) means that the external linguistic data we use is applicable to a wide range of application domains. However, if a suitable specialised corpus were available to us, this might improve the performance of our heuristics by supplying data about some specialised words used in requirements. Coordination ambiguity is generic to language and so will be found in all types of documents. The process of coordination is similar across many languages, so the heuristics we use (or at least the hypotheses they are founded on) will also be applicable in these languages. However, our heuristics are specifically created for tackling coordination ambiguity, and others would have to be found for other types of ambiguity. Our dataset has necessarily been of restricted size. A larger number of examples would ensure that cross-validation would have a less detrimental effort of the heuristics' performance: the heuristics would be operating less sub-optimally on unseen data.

Chapter 8

Future Work

Here we present some ideas for future work that are logical extensions of the research presented in this thesis. We firstly discuss other aspects of coordination ambiguity that might be investigated, and other heuristics that might be tried to distinguish nocuous from innocuous coordination ambiguity. Secondly we look at how our approach might be applied to other types of ambiguity. Thirdly we look at an alternative way of validating the results achieved using our approach. Fourthly we discuss the idea of using our approach to ambiguity in the form of a *wizard*.

8.1 Further Analysis of Coordinations

Here we discuss an interesting aspect of coordination, the application of De Morgan's rules, that might merit future attention using the techniques that we have developed in our research. Then we discuss other heuristics we could evaluate for use in our analysis of coordination ambiguities.

8.1.1 De Morgan's rules

The semantic characteristics of coordinations change in the presence a negation and De Morgan's rules can be applied:

$$\neg(\ a\ and\ b\ )\ =\ \neg a\ or\ \neg b$$

$$\neg(\ a\ or\ b\ )\ =\ \neg a\ and\ \neg b$$

The coordinating conjunctions *and* and *or* swap meanings with each other. This can cause serious misunderstandings, and has been recognised as a potential problem in requirements (Kamsties, Berry, and Paech 2001). The outcome of important court cases have hinged upon whether these rules apply to instances of coordinations in legal statutes (Solan 1993).

The presence (or otherwise) of this effect would of course only be an issue with constructions which were read coordination-first. This is represented by the left-hand side of the equations above, where the negation is (or is part of) the external modifier. One of the main concerns in such cases is which words with a negative connotation are actual negations, thereby triggering De Morgan's rules. It would be instructive to collect examples of coordination ambiguity where such words are (or form part of) the modifier attaching to one or both of the conjuncts, and use judges to determine the preferred reading of the ambiguity. Where nocuousness was witnessed, it might indicate that the word with negative connotation was more likely to be an actual negation, because the potential for triggering De Morgan's rules was a cause of misunderstanding.

8.1.2 Possible Heuristics

Here we discuss two ideas for heuristics that could be tried in addition to the ones we have presented in this thesis.

Derivational Morphology

The simple technique that we have used in this thesis for capturing inflectional morphology encourages us to believe that morphology may be a fertile area of research. It would be interesting to determine whether more evidence can be found that similar morphology of the head words of conjuncts indicates that they tend to be read coordination-first. To this end, we would wish to consider more profoundly the derivational morphology of the words, as opposed to just comparing their trailing letters. This would entail a detailed analysis of the internal structure of the words, and utilisation of morphology classification schemes, particularly for nouns. Morphology is highly language-specific, so a separate treatment of it would be needed for any language that this approach was used on.

Semantic/Distributional Hybrid

A hybrid measure of similarity can be developed using distributional information, such as that supplied by Sketch Engine, and semantic information, such as that obtained using recent facilities available in WordNet. Such a hybrid arrangement has shown promise when used to predict predominant word senses (McCarthy et al. 2004).

8.2 Extension to Other Forms of Ambiguity

The main thrust of our research — introducing the key distinction between nocuous and innocuous ambiguity, and the fact that disambiguation is not always appropriate — is applicable to all forms of ambiguity. Some of the techniques that we use to predict human judgements about our test case coordination ambiguity appear quite specific to that type of ambiguity, due to the fact that we are testing types of linguistic similarity. But, as we explain below, many may have wider usage, most obviously when applied to other forms of structural ambiguity but also when applied to semantic and contextual ambiguities

8.2.1 Other Structural Ambiguities

Prepositional phrase attachment is the most obvious structural ambiguity to tackle next, because it is widespread and proven to cause misunderstandings. It also can be approached using collocation analysis. The Sketch Engine is able to supply highly suitable information for this. Also, its thesaurus could be used to determine similarity between the head noun in a prepositional phrase and the head nouns of each of the candidate phrases to which it might attach. It would be interesting to see, particularly for certain prepositions such as *with* and *by* which have a conjunctive aspect, whether preferred readings can be predicted using differences in these similarities.

8.2.2 Non-Structural Ambiguities

Many types of ambiguities which do not depend on differences in syntactic structure can be approached using the prediction techniques we have presented here. One interesting project would be to test whether similarity between an ambiguous word and any words of the same part of speech that occur in its context has any bearing upon how nocuous it is. A lack of similarity might indicate that an unusual meaning of the ambiguous word is intended, and therefore not only that the word has more than one meaning but also that one of the alternatives might be chosen by a reader less familiar with the language used.

8.3 Validation

It would be interesting to test our techniques against a widely-available benchmark dataset. The Penn Treebank might be suitable in this regard, as it is annotated and is in a mature stage of development. Firstly, it would be interesting to compare the diversity of our judges' interpretations with those of the coders who annotated the Treebank. Less account of the diversity of human perception is incorporated in the latter. We might be able to show that some Treebank annotations are too idiosyncratic, and may be minority judgements when compared to those of our judges. This would validate the notion in our ambiguity model that multiple reading ambiguities are dangerous. Then our heuristics could be run on Treebank data to determine whether they predict the structure there. This would validate the efficacy of the heuristics on generic data. This might produce an overall better performance than running them on a specialised dataset as we have done.

8.4 Wizard

Ultimately we would like to implement our approach in the form of a *wizard*, operating in conjunction with a word processor. This would be an animation that appears on a computer screen to inform authors of how nocuous an ambiguity is deemed to be. For this to be a worthwhile exercise, we would need to extend our techniques to cover, and prove them to be effective on, other types of ambiguity. Other proven techniques could be used to supplement the ones that we have described in this thesis, in order to widen our coverage of ambiguities. High reliability would be required to give people faith in the system.

Appendix A: Ambiguity Questionnaire Instructions

Surprise Prizes for those who complete this Disambiguation Task!!!

Introduction

As you may (or may not) know, I am carrying out disambiguation tests on some real-life requirements texts. In this test I am looking at "coordination ambiguity". I would be very very grateful if you could help me by completing this small task, and returning it to me as soon as you are able. Here are the instructions, (preceded by an explanation of what coordination ambiguity is).

Coordination Ambiguity

Coordination ambiguity arises when the reader is unsure about the meaning of a sentence due to the presence of a coordinator. A coordinator, for our purposes here, is "AND", "OR" or "AND/OR". Uncertainty can arise if it is not clear whether the coordination of words (or phrases) should take priority over the effect of other words that surround those phrases. For instance, in the sentence:

"I like green beans and sausages"

the word "green" almost certainly applies only to the beans. So the coordination does not take priority. It coordinates the phrases to its left and right after the effect of the adjective "green" has taken place.

The Evaluation Task

In this study, we have decided to give you many different types of sentence. The modifier can be an adjective, as in the example above, or it might be an adverb, a prepositional phrase, a relative clause etc etc. However, you don't need to worry unduly about this, as the sentences are presented in the form:

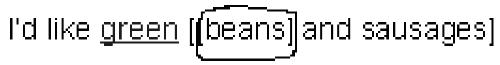
"I like <u>green</u> [[beans] and sausages]"

The modifier is underlined, and the two possible chunks of text to which it might be applied are represented by square bracketings.

You must decide whether or not you feel that a genuine coordination ambiguity exists: i.e. whether or not the sentence might be misunderstood because of that type of ambiguity. In the sentence above, because we are (reasonably) certain that one bracketing is the obvious one, we say that coordination ambiguity is not found.

Ignore any other types of ambiguity that you might encounter.

The lines of text are taken from actual documents which are often highly specialised and technical, so they can sound a little strange! Also, some of them are titles or asides, and so are not formed as proper sentences. An effort has been taken to reduce the original text to the simplest utterance that conveys all the meanings relevant to this disambiguation task. Words in round brackets represent my simplifications; ...... refers to an elision of irrelevant text.



I'd like green [[beans] and sausages]

Your Output

At the beginning of each line, please write an "A" if you feel that a coordination ambiguity is present. Otherwise, circle the bracketing which you feel is the correct one in each case, as shown in Figure 8.4.

Appendix B: Examples in Dataset

Survey 1

(It) manages ...... coordinate [[systems] and other related objects]

(It) must be configured with the proper [[item] and system components]

(It) cannot function with the proper [[installation] and configuration]

(It) shall display categorized [[instructions] and documentation]

The original [[meeting date] or location] may then need to be changed

(It is) useful in determining a best [[meeting date] and location]

Best [[meeting dates] and locations] should be determined

admission information, medication, equipments, daily [[record] and history]

build vehicles from mass-produced [[parts] and subassemblies]

(They are) assembled into ...... vehicle models for model-based [[design] and development]

It also lists applicable [[design constraints] and system attributes]

Create candidate assemblies from the architectural [[description] and component characterizations]

There are several [[assumptions] and dependencies]

The system will parse ...... scalar [[input] and output ports]

(It is) describing the size of vector-based [[inputs] and outputs]

For vector-based [[input] and output], the system will provide a complete characterization

The user may define architectural [[components] and connectors]

Non-Functional User's [[guide] and help documentation]

Signal [[units] and data transfer protocol] must match

Survey 2

Revamp the current ...... [[hardware] and software]

Take care of business [[rules] and transactions]

Involving ...... [election and/or [geopolitics]] entities

Keep track of any [modification or [access]] to the database

(It) might be automatically [[rejected] or flagged]

Initialization [[rules] or pre-conditions]

Termination [[rules] or post-conditions]

Activity like ...... [verification or [printing]] of voters' lists

(It might be) [rejected or [flagged]] for further processing

Display categorized [[instructions] and documentation]

Minimize the [time span and [human resources]] of regular inspection

Facilitate the [scheduling and [performing]] of works

Connectivity of all [control flow and [data flow signals]] between sub-models

This should lead to reduced [[cycle times] and costs] for automotive manufactures

Patients conditions are [inspected and [recorded]] automatically

[Constraints ...... and [dependencies]] that apply to the product

[Assumptions and [dependencies]] that are of importance

It is ...... very [[common] and ubiquitous]

(It) targeted the [project and [election]] managers

Contributing to the [recording and [accuracy]] of the data

Proceed to [enter and [verify]] the data

(He) has the ability to [generate and [print]] pre-defined reports

[Reliability and [security]] considerations

Risk greatly increased by the lack of [[funding] and local resources]

[Operating ...... and [performance]] requirements

Taken with respect to the [quantity and [type]] of data

Used for [verification and [clean-up]] purposes

(It) will be [implemented and [executed]] on the ...... platform

[Memory or [hard disk]] space

[Requirements or [data]] storing

[Communication and [performance]] requirements

[Assumptions and [dependencies]] that apply to the product

[Vandalism or [act]] of God

Survey 3

project aims to [develop and [demonstrate]] methods

the predicted [[coverage] and interference]

[mode and [hot spot]] location

frequency [[bands] and polarisations]

[network and [phone]] services

(It) is a [store and [forward]] function

functions for [[receiving] and transmitting]

[free-space propagation and [powerful field prediction]] tools

the commonly used [[Cell Identity] and Timing Advance]

[developed and [proposed]] to the industry

[free bands and [possible applications]] in other systems

Definition of electrical [[characteristics] and interfaces]

Mounting [[configurations] and hardware]

network [[monitoring] and dedicated field]

zero [[mean values] and standard deviation]

[processed and [stored]] in database

[capacity and [coverage]] relationship

[user and [Node B]] identifications

traffic [[conditions] and variations]

[capacity and [network resources]] required

[Capacity and [Coverage]] Planning

more sophisticated [[Capacity] and Coverage]

[power control and [Radio Resource Management]] parameters

[Planning and [Localization]] Information

network [[planning] and management]

[set and [control]] separately

the different [[importance] and use]

a typical modern [[XXX PC] or YYY workstation]

No need for [control or [supply]] cable

Without any [time or [frequency]] division

[coverage and [interference]] areas

[Cell Identity and [Timing Advance]] parameters

[monitoring and [dedicated field]] measurements

[Capacity and [Coverage]] planning procedure

[importance and [use]] of localisation information

Survey 4

Deficiencies in construction [[plans] and specifications]

[Project client and [project occupant]] representatives

[Interfaces and [qualification]] provisions will be described

Software [[Development] and Documentation]

System Design [[Analysis] and Description]

[Constructibility and [Design]] Reviews

Required [[States] and Modes]

Project [[name] and identification number]

Project [[manager] and designer]

The [start and [finish]] date

The specific set of [[plans] and specifications]

Depending on the [size and [scope]] of the project

The [plans and [specifications]] being reviewed

Identity of the author will be [captured and [included]] with each comment

Check [[boxes] and radio buttons] will be provided

The ability to [cut and [paste]] comments

Delete all associated [[comments] and evaluations]

Phone [[number] and e-mail address]

[Materials and [key]] words

Manual [[input] and selection]

(It) will be further [[evaluated] and included in XXX]

Native ...... [[system forms] and input widgets]

Transmitted ...... for [use and [evaluation]] by others

Descriptions will be [compiled and [included]] in XXX

The [costs and [benefits]] of using XXX

Improved [[design quality] and decreased construction]

Evaluation ...... will be [conducted and [documented]] under XXX

Data shall be [captured and [maintained]] in ...... files

[Security and [Privacy]] Requirements

Sets of approved [[lessons learned] and reference sources]

Access to ...... [Comment and [Evaluation]] Data

[Design and [Implementation]] Constraints

Small ...... [[firms] and construction offices]

[Precedence and [Criticality]] of Requirements

Design [[Review] and Checking System]

[Research and [Development]] Management Information System

Object-Oriented [[Modeling] and Design]

Users [[registration] and access]

[Plans and [specifications]] distributed for that review

Building [[materials] and key words]

[Input and [selection]] of options

(It) will be ...... [evaluated and [included]] in XXX

[Forms and [input]] widgets

Decreased [[construction] and operations cost]

The set of approved [[lessons learned] and existing reference sources]

[Review and [Checking]] System

Comments may not be individually [[modified] or deleted]

Using any ...... [evaluator or [lessons learned]] function

Delete [projects or [reviews]] within a project

Identify each participant as a project [[manager] and/or reviewer]

Operations [[cost] and time]

References

Abney, S. (1995). Chunks and dependencies: Bringing processing evidence to bear on syntax. In J. Cole, G. Green, and J. Morgan (Eds.), *In Computational Linguistics and the Foundations of Linguistic Theory*, pp. 145–164. CSLI.

Abney, S. (1996a). Chunk stylebook. Unpublished Paper, available at: http://www.vinartus.net/spa/96i.pdf (Date Accessed: 28th September 2006).

Abney, S. (1996b). Statistical methods and linguistics. In J. Klavans and P. Resnik (Eds.), *The Balancing Act: Combining Symbolic and Statistical Approaches to Language*, pp. 1–26. Cambridge, Massachusetts: The MIT Press.

Achour, C. B. (1998). Guiding scenario authoring. In G. Grosz (Ed.), *Proceedings of the 8th European Japanese Conference on Information Modelling and Knowledge Bases*, pp. 152–171.

Agarwal, R. and L. Boggess (1992). A simple but useful approach to conjunct identification. In *Proceedings of the 30th conference on Association for Computational Linguistics*, pp. 15–21. Association for Computational Linguistics.

Allen, J. (1995). *Natural language understanding* (2nd ed.). Redwood City, CA, USA: Benjamin-Cummings Publishing Co., Inc.

Bach, K. (1998). Ambiguity. In E. Craig and L. Floridi (Eds.), *Routledge Encyclopedia of Philosophy*. Routledge.

Bar-Hillel, Y. (1960). The present status of automatic translation of languages. *Advances in Computers 1*, 158–163.

Berry, D. M. and E. Kamsties (2005, Jan/Feb). The syntactically dangerous all and plural in specifications. *IEEE Software 22*(1), 55–57.

Berry, D. M., E. Kamsties, and M. M. Krieger (2003). From contract drafting to software specification: Linguistic sources of ambiguity. A Handbook.

Blanchon, H., K.-H. Loken-Kim, and T. Morimoto (1995). An interactive disambiguation module for English natural language utterances. In *Proceedings of NLPRS'95*, pp. 550–555.

Boehm, B. W. (1981). *Software Engineering Economics*. Englewood Cliffs, NJ, U.S.A.: Prentice-Hall.

Boehm, B. W. (1984). Verifying and validating software requirements and design specifications. *IEEE Software 1*(1), 75–88.

Boitet, C. and M. Tomokiyo (1996). Theory and practice of ambiguity labelling with a view to interactive disambiguation in text and speech mt. In *Proceedings of the 16th conference on Computational linguistics*, Morristown, NJ, USA, pp. 119–124. Association for Computational Linguistics.

Boyd, S., D. Zowghi, and A. Farroukh (2005). Measuring the expressiveness of a constrained natural language: An empirical study. In *Proceedings of the 13th IEEE International Conference on Requirements Engineering (RE'05)*, Washington, DC, USA, pp. 339–352. IEEE Computer Society.

Brill, E. (1992). A simple rule-based part of speech tagger. In *Proceedings of the third conference on Applied natural language processing*, pp. 152–155. Association for Computational Linguistics.

Caine, S. H. and E. K. Gordon (1975). Pdl – a tool for software design. In *AFIPS Conference Proceedings*, Volume 44, Montvale, NJ, pp. 271–276. National Computer Conference.

Calvo, H., A. Gelbukh, and A. Kilgarriff (2005). Distributional thesaurus vs. wordnet: A comparison of backoff techniques for unsupervised pp attachment. In *Proceedings of Sixth International Conference on Intelligent Text Processing and Computational Linguistics: CICLING05*, pp. 172–182.

Carroll, D. W. (1999). *Psychology of Language* (Third ed.). Toronto, Canada: Brooks/Cole Publishing Company.

Chantree, F. (2004). Ambiguity management in natural language generation. In M. Lee (Ed.), *In Proceedings of 7th Annual CLUK Research Colloquium*, Birmingham, U.K., pp. 23–28. University of Birmingham.

Chantree, F., A. Kilgarriff, A. de Roeck, and A. Willis (2005). Disambiguating coordinations using word distribution information. In G. Angelova, K. Bontcheva, R. Mitkov, N. Nicolov, and N. Nikolov (Eds.), *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, Borovets, Bulgaria, pp. 144–151.

Chantree, F., B. Nuseibeh, A. de Roeck, and A. Willis (2005). Nocuous ambiguities in requirements specifications. Technical Report 2005/03, Open University, Milton Keynes, U.K.

Chantree, F., B. Nuseibeh, A. de Roeck, and A. Willis (2006). Identifying nocuous ambiguities in requirements specifications. In M. Glinz and R. Lutz (Eds.), *Proceedings* *of the 14th IEEE International Requirements Engineering conference*, pp. 59–68. IEEE Computer Society.

Chantree, F., A. Willis, A. Kilgarriff, and A. de Roeck (2006). Detecting dangerous coordination ambiguities using word distribution. In N. Nicolov and R. Mitkov (Eds.), *book chapter to appear in Current Issues in Linguistic Theory: Recent Advances in Natural Language Processing.* Amsterdam, The Netherlands: John Benjamins Publishers.

Cronen-Townsend, S. and W. B. Croft (2002). Quantifying query ambiguity. In *Proceedings of Human Language Technology Conference*, pp. 94–98.

Cushing, S. (1994). *Fatal Words: Communication Clashes and Aircraft Crashes.* Chicago, U.S.A.: University of Chicago Press.

Daelemans, W., J. Zavrel, K. van der Sloot, and A. van den Bosch (2003). Timbl: Tilburg memory based learner (version 5.0) reference guide. Technical Report ILK 03-10, CNTS - Language Technology Group, University of Antwerp, Belgium.

Davidson, G. (1996). *Chambers Guide to Grammar and Usage.* Edinburgh, U.K.: Chambers.

Dubey, A., P. Sturt, and F. Keller (2005). Parallelism in coordination as an instance of syntactic priming: Evidence from corpus-based modeling. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing*, pp. 827–834.

Easterbrook, S., R. R. Lutz, R. Covington, J. Kelly, Y. Ampo, and D. Hamilton (1998). Experiences using lightweight formal methods for requirements modeling. *Software Engineering 24*(1), 4–14.

Emele, M. C. and M. Dorna (1998). Ambiguity preserving machine translation using packed representations. In *Proceedings of the 17th international conference on Computational linguistics*, Morristown, NJ, USA, pp. 365–371. Association for Computational Linguistics.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal Machine Learning Research 3*, 1289–1305.

Fowler, F. (2001). *Survey Research Methods* (Third ed.). Thousand Oaks, CA, U.S.A.: Sage Publications.

Fowler, H. W. and S. E. Gowers (1965). *A Dictionary of Modern English Usage* (Second ed.). Oxford, U.K.: Oxford University Press.

Frazier, L. (1978). *On Comprehending Sentences: Syntactic Parsing Strategies*. Ph. D. thesis, University of Massachusetts, Amherst, MA, U.S.A.

Frazier, L. (1985). Syntactic complexity. In D. Dowty, L. Karttunen, and A. Zwicky (Eds.), *Natural Language Processing; Psychological, Computational and Theoretical Perspectives*, pp. 129–189. Cambridge, United Kingdom: Cambridge University Press.

Frazier, L. and J. D. Fodor (1978). The sausage machine: A new two-stage parsing model. *Cognition 6*, 291–325.

Frazier, L., A. Munn, and C. Clifton (2000). Processing coordinate structures. *Journal of Psycholinguistic Research 29*(4), 343–370.

Freedman, D. P. and G. M. Weinberg (2000). *Handbook of Walkthroughs, Inspections, and Technical Reviews: Evaluating Programs, Projects, and Products*. New York, NY, USA: Dorset House Publishing Co., Inc.

Friedman, J., T. Hastie, and R. Tibshirani (2000). Additive logistic regression: a statistical view of boosting. *The Annals of Statistic 38*(2), 337–374.

Fuchs, N. and R. Schwitter (1996). Attempto Controlled English (ACE). In *Proceedings of the first international workshop on controlled language applications*.

Gale, W., K. W. Church, and D. Yarowsky (1992). Estimating upper and lower bounds on the performance of word-sense disambiguation programs. In *Proceedings of the 30th conference on Association for Computational Linguistics*, pp. 249–256. Association for Computational Linguistics.

Gause, D. C. (2000). User driven design – the luxury that has become a necessity. A Workshop in Full LifeCycle Requirements Management. At ICRE 2000, Tutorial T7, Schaumberg IL, USA.

Gause, D. C. and G. M. Weinberg (1989). *Exploring Requirements: Quality Before Design*. New York, NY, USA: Dorset House Publishing Co., Inc.

Gervasi, V. (2000). *Environment Support for Requirements Writing and Analysis*. Ph. D. thesis, Università degli Studi di Pisa.

Gervasi, V. and B. Nuseibeh (2000). Lightweight validation of natural language requirements: a case study. In *Proceedings of the 4th IEEE International Conference on Requirements Engineering*. IEEE Computer Society Press.

Gervasi, V. and B. Nuseibeh (2002). Lightweight validation of natural language requirements. *Software Practice and Experience 32*(2), 113–133.

Gervasi, V. and D. Zowghi (2005). Reasoning about inconsistencies in natural language requirements. *ACM Transactions on Software Engineering and Methodology (TOSEM) 14*(3), 277–330.

Gillon, B. S. (1990). Ambiguity, generality and indeterminacy: Test and definitions. *Synthese 85*, 391–416.

Gillon, B. S. (2003). Ambiguity, indeterminacy, deixis and vagueness: Evidence and theory. In S. Davis and B. S. Gillon (Eds.), *Semantics: A Reader*, pp. 157–187. Oxford University Press.

Goguen, J. A. (1994). Requirements engineering as the reconciliation of social and technical issues. In M. Jirotka and J. A. Goguen (Eds.), *Requirements engineering: social and technical issues*, pp. 165–199. San Diego, CA, USA: Academic Press Professional, Inc.

Goldberg, M. (1999). An unsupervised model for statistically determining coordinate phrase attachment. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, pp. 610–614. Association for Computational Linguistics.

Goldin, L. and D. M. Berry (1997, October). Abstfinder, a prototype natural language text abstraction finder for use in requirements elicitation. *Automated Software Engineering 4*(4), 375–412.

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

Grice, H. P. (1975). Logic and conversation. In Davidson and Harman (Eds.), *The Logic of Grammar*, pp. 64–75. Encino, CA, U.S.A.: Dickenson Publishing.

Hanks, K. S., J. C. Knight, and E. A. Strunk (2001, November). A linguistic analysis of requirements errors and its application. Technical Report CS-2001-30, University of Virginia, Department of Computer Science, Charlottesville, Virginia, U.S.A.

Heitmeyer, C. L., R. D. Jeffords, and B. G. Labaw (1996). Automated consistency checking of requirements specifications. *ACM Transactions on Software Engineering and Methodology 5*(3), 231–261.

Hillelsohn, M. J. (2004, April). Better communication through better requirements. *Crosstalk: The Journal of Defense Software Engineering 17*(4), 24–27.

Hindle, D. and M. Rooth (1993). Structural ambiguity and lexical relations. *Computational Linguistics 19*(1), 103–120.

Hirschberg, J. and D. Litman (1993). Empirical studies on the disambiguation of cue phrases. *Computational Linguistics 19*(3), 501–530.

Hobbs, J. R. (1983). An improper treatment of quantification in ordinary English. In *Proceedings of the Twenty-First Annual Meeting of the Association for Computational Linguistics*, pp. 57–63.

Ioannidis, Y. E. and Y. Lashkari (1994). Incomplete path expressions and their disambiguation. In *SIGMOD '94: Proceedings of the 1994 ACM SIGMOD international conference on Management of data*, New York, NY, USA, pp. 138–149. ACM Press.

Ishioka, T. (2003). Evaluation of criteria for information retrieval. In *Proceedings of IEEE/WIC International Conference on Web Intelligence (WI 2003)*, pp. 425–431.

Jackson, M. (1995). *Software requirements & specifications: a lexicon of practice, principles and prejudices.* New York, NY, USA: ACM Press/Addison-Wesley Publishing Co.

Jones, K. R. (2003). Miscommunication between pilots and air traffic control. *Language Problems and Language Planning 27*(3), 233–248.

Jurafsky, D. and J. H. Martin (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* Prentice-Hall.

Just, M. A. and P. A. Carpenter (1980). A theory of reading: From eye fixations to comprehension. *Psychological Review 87*, 329–354.

Kamsties, E. (2001). *Surfacing Ambiguity Natural Language Requirements.* Ph. D. thesis, Fraunhofer-Institute für Experimentelles Software Engineering, Kaiserslautern, Germany.

Kamsties, E., D. M. Berry, and B. Paech (2001). Detecting ambiguities in requirements documents using inspections. In M. Lawford and D. L. Parnas (Eds.), *Proceedings of the First Workshop on Inspection in Software Engineering (WISE'01)*, pp. 68–80.

Keren, G. (1992). Improving decisions and judgments: The desirable versus the feasible. In G. Wright and F. Bolger (Eds.), *Expertise and Decision Support*, pp. 25–46. New York: Plenum Press.

Kess, J. F. and R. A. Hoppe (1981). Ambiguity in psycholinguistics. *Pragmatics & Beyond II*(4), 1–123.

Kilgarriff, A. (2003a). Thesauruses for natural language processing. In C. Zong (Ed.), *Proceedings of NLP-KE*, Beijing, China, pp. 5–13.

Kilgarriff, A. (2003b). What computers can and cannot do for lexicography, or us precision, them recall. In *Proceedings of the 3rd Conference of the Asian Association for Lexicography (ASIALEX)*, Tokyo, Japan. Also published as ITRI University of Brighton technical report ITRI-03-16.

Kilgarriff, A., P. Rychly, P. Smrz, and D. Tugwell (2004). The sketch engine. In G. Williams and S. Vessier (Eds.), *Proceedings of the Eleventh European Association for Lexicography (EURALEX) International Congress*, pp. 105–116.

Kilgarriff, A. and D. Tugwell (2001). Word sketch: Extraction and display of significant collocations for lexicography. In *Proceedings of the Collocations Workshop, ACL 2001*, pp. 32–38. Association for Computational Linguistics.

King, T. H., S. Dipper, A. Frank, J. Kuhn, and J. Maxwell (2000). Ambiguity management in grammar writing. In E. Hinrichs, D. Meurers, and S. Wintner (Eds.), *Proceedings of the Workshop on Linguistic Theory and Grammar Implementation, ESSLLI*.

Knight, K. and I. Langkilde (2000). Preserving ambiguities in generation via automata intersection. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pp. 697–702. AAAI Press / The MIT Press.

Kovitz, B. L. (1999). *Practical Software Requirements: A Manual of Content & Style.* Greenwich, CT, USA: Manning Publications.

Kovitz, B. L. (2002). Ambiguity and what to do about it. In *Proceedings of the 10th IEEE Joint International Conference on Requirements Engineering*, pp. 213. IEEE Computer Society Press.

Kurohashi, S. and M. Nagao (1992). Dynamic programming method for analyzing conjunctive structures in japanese. In *Proceedings of the 14th conference on Computational linguistics*, pp. 170–176. Association for Computational Linguistics.

Landwehr, N., M. Hall, and E. Frank (2003). Logistic model trees. In N. Lavrac, D. Gamberger, L. Todorovski, and H. Blockeel (Eds.), *Proceedings of the 14th European Conference on Machine Learning*, pp. 241–252. Springer.

Langendoen, D. T. (1995). An analysis of coordinate compounding. Unpublished Paper, University of Arizona.

Langendoen, D. T. (1998). Limitations on embedding in coordinate structures. *Journal of Psycholinguistic Research 27*, 235–259.

Langkilde, I. (2000). Forest-based statistical sentence generation. In *Proceedings of NAACL'00*, Seattle, WA, pp. 170–177.

Lauer, M. (1995). Corpus statistics meet the noun compound: some empirical results. In *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*, Cambridge, Massachusetts, USA, pp. 47–54. Association for Computational Linguistics.

Lin, D. (1998). Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics*, pp. 768–774. Association for Computational Linguistics.

Maarek, Y. S. and D. M. Berry (1989). The use of lexical affinities in requirements extraction. In *IWSSD '89: Proceedings of the 5th international workshop on Software specification and design*, New York, NY, USA, pp. 196–202. ACM Press.

MacKay, D. G. (1966). To end ambiguous sentences. *Perception and Psychophysics 1*, 426–436.

Manning, C. and H. Schütze (1999, May). *Foundations of Statistical Natural Language Processing*. Cambridge, MA, U.S.A.: MIT Press.

McCarthy, D., R. Koeling, J. Weeds, and J. Carroll (2004). Finding predominant senses in untagged text. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pp. 280–287.

McLauchlan, M. (2004). Thesauruses for prepositional phrase attachment. In H. T. Ng and E. Riloff (Eds.), *Proceedings of Eight Conference on Natural Language Learning (CoNLL)*, pp. 73–80. Boston, MA, USA.

Mich, L. (2001). On the use of ambiguity measures in requirements analysis. In A. Moreno and R. van de Riet (Eds.), *NLDB'01: Proceedings of the 6th International Workshop on Applications of Natural Language to Information Systems*, pp. 143–152. GI.

Mich, L. and R. Garigliano (2000). Ambiguity measures in requirement engineering. In Y. Feng, D. Notkin, and M. Gaudel (Eds.), *In Proceedings of International Conference On Software Theory and Practice - ICS2000, 16th IFIP World Computer Congress*, Beijing, China, pp. 39–48. Publishing House of Electronics Industry.

Mitamura, T. (1999). Controlled language for multilingual machine translation. In *Proceedings of Machine Translation Summit VII*, Singapore.

Mitchell, T. M. (1997). *Machine Learning.* McGraw-Hill Higher Education.

Morgan, R., R. Garigliano, P. Callaghan, S. Poria, M. Smith, A. Urbanowicz, R. Collingham, M. Costantino, C. Cooper, and the LOLITA Group (1996). Description of the lolita system as used for muc-6. In *In Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pp. 71–85. Morgan-Kaufmann Publishers.

Mullery, G. (19). The perfect requirement myth. *Requirements Engineering 1*(2), 132–134.

Munn, A. (1993). *Topics in the Syntax and Semantics of Coordinate Structures.* Ph. D. thesis, University of Maryland, Maryland, U.S.A.

Nakov, P. and M. Hearst (2005). Using the web as an implicit training set: Application to structural ambiguity resolution. In *Proceedings of HLT-NAACL'05*, pp. 835–842.

Navarretta, C. (1994). Methodologies for knowledge acquisition from nl texts. In S. L.

Hansen and H. Wegener (Eds.), *Topics in Knowledge-based NLP systems*, pp. 7–17. Frederiksberg: Samfundslitteratur.

Nuseibeh, B. and S. Easterbrook (2000). Requirements engineering: a roadmap. In *ICSE '00: Proceedings of the Conference on The Future of Software Engineering*, New York, NY, USA, pp. 35–46. ACM Press.

Nuseibeh, B., S. Easterbrook, and A. Russo (2001). Making inconsistency respectable in software development. *The Journal of Systems and Software 58*(2), 171–180.

Oberlander, J. and S. Nowson (2006, July). Whose thumb is it anyway? classifying author personality from weblog text. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, Sydney, Australia, pp. 627–634. Association for Computational Linguistics.

och Dag, J. N., B. Regnell, V. Gervasi, and S. Brinkkemper (2005, Jan/Feb). A linguistic-engineering approach to large-scale requirements management. *IEEE Software 22*(1), 32–39.

Okumura, A. and K. Muraki (1994). Symmetric pattern matching analysis for English coordinate structures. In *Proceedings of the 4th Conference on Applied Natural Language Processing*, pp. 41–46. Association for Computational Linguistics.

Osborne, M. and C. K. MacNish (1996). Processing natural language software requirement specifications. In *Proceedings of the 2nd International Conference on Requirements Engineering (ICRE '96)*, Washington, DC, USA, pp. 229–236. IEEE Computer Society.

Park, J. C. and H. J. Cho (2000). Informed parsing for coordination with combinatory categorial grammar. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pp. 593–599.

Pinkal, M. (1995). *Logic and Lexicon.* Dordrecht, The Netherlands: Kluwer.

Poesio, M. (1996). Semantic ambiguity and perceived ambiguity. In K. van Deemter and S. Peters (Eds.), *Semantic Ambiguity and Underspecification*, pp. 159–201. Cambridge, England: Cambridge University Press.

Porter, A. A., L. G. V. Jr, and V. R. Basili (1995, June). Comparing detection methods for software requirements inspections: A replicated experiment. *IEEE Transactions on Software Engineering 21*(6), 563–575.

Quinlan, J. R. (1992). *C4.5: Programs for Machine Learning.* San Mateo: Morgan Kaufmann Publishers.

Quirk, R., S. Greenbaum, G. Leech, and J. Svartvik (1985). *A Comprehensive Grammar of the English Language.* New York: Longman.

Ratnaparkhi, A. (1998). Unsupervised statistical models for prepositional phrase attachment. In *Proceedings of the 17th International Conference on Computational Linguistics*, pp. 1079–1085.

Reibel, D. A. and S. A. Schane (1969). *Modern Studies in English.* Eaglewood Cliffs: Prentice-Hall.

Resnik, P. (1996). Selectional constraints: An information-theoretic model and its computational realization. *Cognition 61*, 127–159.

Resnik, P. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research 11*, 95–130.

Reyle, U. (1993). Dealing with ambiguities by underspecification: Construction, representation, and deduction. *Journal of Semantics 10*(2), 123–179.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks.* Cambridge, U.K.: Cambridge: University Press.

Robertson, S. and J. Robertson (1999). *Mastering the requirements process.* New York, NY, USA: ACM Press/Addison-Wesley Publishing Co.

Ross, J. R. (1967). *Constraints on Variables in Syntax.* Ph. D. thesis, MIT. Also Published as Infinite Syntax! Ablex, Norwood, NJ, U.S.A., 1986.

Rumbaugh, J., I. Jacobson, and G. Booch (1999). *The Unified Modeling Language reference manual.* Essex, UK: Addison-Wesley Longman Ltd.

Rupp, C. (2000). Linguistic methods of requirements engineering (nlp). In *Proceedings of European Software Process Improvement (EuroSPI 2000)*, Copenhagen, Denmark.

Rus, V., D. I. Moldovan, and O. Bolohan (2002). Bracketing compound nouns for logic form derivation. In *Proceedings of the FLAIRS 2002 Conference*, pp. 198–202.

Ryan, K. (1993). The role of natural language in requirements engineering. In *In Proceedings of the IEEE Int. Symposium on Requirements Engineering*, pp. 240–242. IEEE Computer Society Press.

Sang, E. F. T. K. and S. Buchholz (2000). Introduction to the conll shared task: Chunking. In *Proceedings of CoNLL-2000*, Lisbon, Portugal, pp. 127–132.

Sawyer, P., P. Rayson, and K. Cosh (2005). Shallow knowledge as an aid to deep understanding in early phase requirements engineering. *IEEE Transactions On Software Engineering 31*(11), 969–981.

Schepman, A. and P. Rodway (2000). Prosody and parsing in coordination structures. *The Quarterly Journal of Experimental Psychology: A 53*(2), 377–396.

Seligman, M. (1997). Six issues in speech translation. In S. K. et al. (Ed.), *Proceedings of Spoken Language Translation Workshop at (E)ACL-97*, pp. 83–89.

Shemtov, H. (1997). *Ambiguity Management in Natural Language Generation.* Ph. D. thesis, Stanford University, U.S.A.

Shull, F., I. Rus, and V. Basili (2000, July). How perspective-based reading can improve requirements inspections. *IEEE Computer 33*(7), 73–79.

Simpson, J. A., G. W. Davidson, and M. A. Seaton (1988). *Chambers Concise Dictionary.* Edinburgh, U.K.: W & R Chambers Ltd.

Sojka, P., I. Kopecek, and K. Pala (Eds.) (2004). *Proceedings of the Seventh International Conference on Text, Speech, Dialogue (TSD 2004).* Springer-Verlag in Lecture Notes in Artificial Intelligence (LNAI) subseries of LNCS series, Volume 3206.

Solan, L. M. (1993). *The Language of Judges.* Chicago, U.S.A.: University of Chicago Press.

Spanoudakis, G., A. d'Avila Garcez, and A. Zisman (2003). Revising rules to capture requirements traceability relations: A machine learning approach. In *Proceedings of 15th International Conference in Software Engineering and Knowledge Engineering (SEKE 2003)*, San Francisco, CA, USA, pp. 570–577.

Sparck-Jones, K. (1986). *Synonymy and semantic classification.* Edinburgh University Press.

Sperber, D. and D. Wilson (1982). Mutual knowledge and relevance in theories of comprehension. In N. Smith (Ed.), *Mutual Knowledge*, pp. 61–85. London: Academic Press.

Spivey, J. M. (1992). *The Z notation: a reference manual.* Hemel Hempstead, Hertfordshire, UK: Prentice Hall International (UK) Ltd.

Taraban, R. and J. L. McClelland (1988). Constituent attachment and thematic role assignment in sentence processing: influences of content-based expectations. *Journal of Memory and Language 27*, 597–632.

Terra, E. (2004). *Lexical Affinities and Language Applications.* Ph. D. thesis, School of Computer Science, University of Waterloo - Canada.

Trask, R. L. (1997). *The Penguin Guide to Punctuation.* Harmondsworth: Penguin.

Umarji, R. (1962). *Probability and Statistical Methods.* Bombay, India: Asia Publishing House.

van Deemter, K. (1998). Ambiguity and idiosyncratic interpretation. *Journal of Semantics 15*(1), 5–36.

van Deemter, K. (2004). Towards a probabilistic version of bidirectional ot syntax and semantics. *Journal of Semantics 21*(3), 251–280.

van Rijsbergen, C. J. (1979). *Information Retrieval.* London, U.K.: Butterworths.

van Rooy, R. (2004). Relevance and bidirectional ot. In R. Blutner and H. Zeevat (Eds.), *Optimality Theory and Pragmatics*, pp. 173–210. Basingstoke, Hampshire, U.K.: Palgrave/Macmillan.

Walton, D. (1996). *Fallacies Arising from Ambiguity.* Dordrecht, The Netherlands: Kluwer Academic Publishers.

Wasow, T., A. Perfors, and D. Beaver (2003). The puzzle of ambiguity. In O. Orgun and P. Sells (Eds.), *Morphology and the Web of Grammar: Essays in Memory of Steven G. Lapointe.* CSLI Publications.

Weiss, S. M. and C. A. Kulikowski (1991). *Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning, and expert systems.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Whitis, V. S. and W. N. Chiang (1981). A state machine development for call processing. In *IEEE Electro '81 Conference*, pp. 7/2/1–7/2/6. Computer Society Press of the Institute of Electrical and Electronic Engineers.

Witten, I. H. and E. Frank (2005). *Data Mining: Practical machine learning tools and techniques* (2nd ed.). San Francisco: Morgan Kaufmann.

Yamaguchi, M., T. Kojima, N. Inui, Y. Kotani, and H. Nisimura (1998). Combination of an automatic and an interactive disambiguation method. In *Proceedings of the 17th international conference on Computational linguistics*, pp. 1423–1427. Association for Computational Linguistics.

Zhou, X.-H., D. K. McClish, and N. A. Obuchowski (2002). *Statistical Methods in Diagnostic Medicine.* Wiley Series in Probability and Statistics. John Wiley and Sons.

Zowghi, D. and V. Gervasi (2002). The three Cs of requirements: Consistency, completeness, and correctness. In C. Salinesi, B. Regnell, and K. Pohl (Eds.), *Proceedings of 8th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ02)*, pp. 155–164. Essener Informatik Beiträge.

Zowghi, D., V. Gervasi, and A. McRae (2001). Using default reasoning to discover inconsistencies in natural language requirements. In *Proceedings of the 8th Asia-Pacific Software Engineering Conference*, Macau, China.

Zwicky, A. M. and J. M. Sadock (1975). Ambiguity tests and how to fail them. In J. Kimball (Ed.), *Syntax and Semantics*, Volume 4, pp. 1–36. New York: Academic Press.