Chapter 1

Introduction

1.1 Ambiguity in language

In understanding language, humans have to deal with large amounts of ambiguity. If we restrict our attention to the syntax in texts, then we may focus on ambiguity in two forms. The first is lexical ambiguity, the second is structural ambiguity. Lexical ambiguity arises when one word can have several meanings. Structural ambiguity arises when parts of a sentence can be syntactically combined in more than one way. Humans resolve most ambiguity, of both types, without even being consciously aware of alternatives. If ambiguity remains of which we are aware, knowledge about the world is used in combination with what is known about the linguistic context of the ambiguity to arrive at the most likely analysis.

The sentence in example (1) contains potential lexical ambiguity in the word bank. This word has several meanings as a noun, including that of the financial institute and that of the shore of a river. In addition, it may be the first person form of the verb to bank, which also has several meanings. In the example, the syntactic context of the word (such as the preceding determiner) tells us that it is likely to be a noun, while the word money suggests that out of the possible meanings of the noun bank, the financial institute sense is the most plausible one.

(1) I bring my money to the bank.

Both the choice between financial bank and river bank, and that between verb and noun, are instances of lexical ambiguity. However, while the choice between verb and noun would result in different syntactic analyses, the institute/river ambiguity is purely semantic as both meanings concern nouns.

In example (2) the two different bracketings of a sentence show a case
of structural ambiguity. The choice is between wearing a white coat being combined with the vet or with the dog. Knowledge about vets and dogs will tell humans that the vet is most likely to be wearing the coat.

(2) a. The vet operates on [the dog]$_{OBJ}$ wearing a white coat.
   b. The vet operates on [the dog wearing a white coat]$_{OBJ}$.

Out of the different types of ambiguity, purely semantic ambiguity such as the difference between a financial bank and a river bank, will not be addressed in this research (this is the problem of word sense disambiguation as addressed for example in [40]). In the next section, the problem posed by ambiguity to computational parsers is discussed. The parser that we have in mind is one that would work with the difference between the syntactic categories of nouns and verbs, but not with semantic differences between nouns that share a word form.

1.2 Natural language parsing by computer

While humans may resolve most ambiguity without consciously considering all alternatives, the ambiguity in natural language is all the more a problem to wide-coverage parsers of natural language implemented on computers. A grammar describing all possible syntactic constructions in a given language, together with a lexicon listing the possible syntactic categories of the words in that language, will on average lead to a very large number of possible analyses for a given sentence if no form of disambiguation is applied.

While the ideas described in this work apply to wide-coverage parsers in general, the particular system to which these ideas are applied is the Alpino wide-coverage parser for Dutch [15]. Although special care is taken in this parser to reduce ambiguity, as described in chapter 4, the problem is still a significant one.

In addition to ambiguity, the complexity of wide-coverage parsers is in itself a cause of inefficiency. Aiming at covering as much of a language as possible, the grammars contained in these systems are large and complex and typically do not allow for efficient processing. It would be preferable to apply the knowledge in such complex systems in a more efficient way.

The goal of the work described in this thesis is to improve parsing performance through reduction of ambiguity, and to do so efficiently. In the approach taken, a model of the parser is created, aimed not at replacing the parser in its entirety, but at supporting the parser by performing ambiguity reduction at an early stage of parsing. In order to represent the parser accurately and at the same time allow for efficient processing, the parser is
approximated in a finite-state model. Practical experiments will be carried out to find an answer to the following question: “How can a finite-state approximation of a wide-coverage natural language parser be constructed and used in order to improve parsing performance?”

1.3 Overview of the dissertation

In chapter 2, the finite-state automaton is defined. The ideas of competence and performance in human language processing are used to illustrate the role finite-state processes may have in language processing. Several methods of approximating a grammar of greater than finite-state power with a finite-state model are presented.

In chapter 3, it is shown how a stochastic finite-state model can be derived from a wide-coverage parser through the method known as inference. In the following chapters, a number of different models constructed using this technique will be used in the task of reducing lexical and structural ambiguity in parsing.

The problem of lexical ambiguity is addressed in chapter 4. This chapter is based on a previous publication [70]. It is shown how a stochastic finite-state model inferred from annotated data produced by a wide-coverage parser is used to increase the speed of the parser, while also increasing its accuracy. The model is a POS tagging model and is used in a tagger that will also be used in chapters 5 and 6.

In chapter 5 the POS tagging model created in chapter 4 is extended with specific syntactic information in order to reduce the tagging error rate. Aiming at the most frequent error made in tagging Dutch, a decrease in tagging error rate is observed.

In chapter 6, ideas from chapters 4 and 5 are combined to create a model of syntactic structure on the level of chunks. The approach described in chapter 5 is used to add information on noun phrase chunks to the model. This model is then used in a setup that aims at increasing the efficiency of the parser by decreasing structural ambiguity. It is shown that the current implementation of this idea does not result in an improvement in parsing performance, and a number of suggestions are made on how this approach could be improved upon in future work.

Finally, chapter 7 will present conclusions, answering the above research question based on the results described in the preceding chapters.
Chapter 1. Introduction