Chapter 1

Introduction

1.1 Ambiguity in Language

In the field of computational linguistics, researchers are mainly concerned with the computational processing of natural language. A number of results have already been obtained, ranging from concrete and applicable systems able to understand or produce language to theoretical descriptions of the underlying algorithms.

However, a number of important research problems have not been solved. A particular challenge for computational linguistics pertaining to all levels of language is ambiguity. Most people are quite unaware of how vague and ambiguous human languages really are, and they are disappointed when computers are hardly able to understand language and linguistic communication the way humans do. Ambiguity means that a word or sentence can be interpreted in more than one way, has more than one meaning. It should not be confused with vagueness, in which a word or phrase has only one meaning whose boundaries are not sharply defined. Mostly ambiguity does not pose a problem for humans and is therefore not perceived as such. The only exception where ambiguity is actively employed are jokes and puns. For a computer, however, ambiguity is one of the main problems encountered in the analysis and generation of natural languages.

We can distinguish various kinds of ambiguity. A word can be ambiguous with regard to its internal structure (morphological ambiguity). Compounds are a typical source of morphological ambiguity. Two Dutch examples are massagebed, which can be analyzed as massage-bed (massage bed) or massagedeb (mass prayer), and computertaalkunde, with the two analyses computertaalkunde (computer linguistics) and computertaal-kunde (programming language knowledge).
This kind of ambiguity can also be observed more implicitly, such as for example with the English verb form *look*: It can either be the infinitive, first or second person singular or plural, but as soon as the word immediately preceding *look* is known, the ambiguity can be resolved in most cases (e.g. *to look* is the infinitive, *I look* is first person singular, etc.). Look can also be ambiguous with regard to its syntactic class, its so called part-of-speech. In the sentence “We look at her” *look* is a verb whereas in “She gave him a warning look” it is a noun.

Another kind of syntactic ambiguity can be found at sentence level. A classic example is so called PP attachment ambiguity: The sentence “The man saw the girl with the telescope” is ambiguous with respect to whether the man had the telescope and was using it to see the girl or whether the girl was carrying the telescope. In contrast, the sentence “The man saw the girl with the ice cream” is not ambiguous for the human reader (we know that ice cream cannot be used to see), while it presents the same difficulty as the telescope sentence for the computer to resolve. Pragmatics can also lead to ambiguity, as e.g. with the interpretation of pronouns. Consider for example the two utterances in (1):

(1) Mary’s mother is a gardener. John likes her.

The pronoun *her* in the second sentence can either refer to Mary or to her mother. The preferred (and congruent) reading would be that John likes Mary’s mother, but once more there is potential ambiguity that needs to be resolved.

At word level again, *lexical* semantic ambiguity occurs when a single word is associated with multiple senses. We will be focusing on this type of ambiguity in the present thesis. To illustrate the problem of lexical ambiguity, consider the noun *party*. It can refer to (at least) 4 different things:

- an organization to gain political power (*political party*)
- a band of people associated temporarily in some activity (*search party/*party of three*)
- a group of people gathered together for pleasure (*birthday party*)
- a person involved in legal proceedings (*third party rights*)

1 An exception occurs if the preceding word is the personal pronoun *you* which can either be singular or plural and which, in addition, can also be used as a direct object instead of as the subject. In those cases, more context and more information has to be taken into account to achieve disambiguation.
1.1. Ambiguity in Language

Without any further information, a list of possible senses like the one above is the best we can do to decide what *party* refers to. One could also argue that all these meanings are related and could be subsumed in a more general sense of *party*, namely “group of people” (but for many other words no such general sense can be found).

However, for various applications, such as information retrieval queries or machine translation, it is important to be able to distinguish between the different senses of the word *party*. In order to correctly translate an English sentence containing *party* to Dutch for example, we first have to know which meaning of *party* is intended in English and then find the best translation equivalent in the given context in Dutch. The preferred translation for *birthday party* would be *(verjaardags)feestje*, whereas for *political party* it would be *partij*—two words with quite distinct meanings. Also, when we formulate an Internet query, there is usually one specific meaning we intend and we only want to retrieve documents or links relevant for that particular meaning. So, if, for instance, we are looking for information on a political party, we are not interested in documents on search parties that have been conducted or legal issues. For this reason, it is crucial to be able to distinguish the various senses of a word.

Now let us consider the meaning of *party* in the following sentence:

(2) The guests left John’s party right away.

It is quite clear to the human reader that the only possible reading here is the ‘social gathering for pleasure’. It is interesting to note that most people are not even aware of the potential ambiguity contained in this sentence. Humans are so skilled at resolving potential ambiguities that they do not realize that they are doing it. There has been research on how people resolve ambiguities (see Small et al. (1988) for a collection of articles from a psycholinguistic and neurolinguistic point of view), but since we (still) do not exactly know how lexical ambiguity resolution is done by humans, it is even more difficult to teach a computer to achieve the same thing. Especially if more than one ambiguous word is present in a sentence, the number of potential interpretations of the sentence “explodes”: the number of interpretations is the product of all possible meanings of the words. Assume that only *left* and *party* are ambiguous in the example sentence, and that they both have 4 senses. This brings the number of possible interpretations to 16. Imagine what happens if there are more senses to take into account as illustrated in figure 1.1 (on page 4) or if the sentence gets longer.

The most prominent way to determine the meaning of a word in a particular usage is to examine its context. The context can be seen as the words
Chapter 1. Introduction

The guests left John’s party right away. The dotted lines show all possible combinations of senses for all words, the black line indicates the correct path.

surrounding the ambiguous word, in this case party. A word such as guest might be a good cue for a particular sense of party. But words surrounding the ambiguous word is not the only kind of information that is available. Underneath the simple words lies information on whether a word in the context is a noun or a verb (its syntactic class), on whether that same word plays the role of subject or object, on the syntactic structure of the entire sentence, etc. All this information is certainly available to people in the process of disambiguation and a combination of all these different kinds of information together with general knowledge about the situation and the world is used to rule out improbable readings.

The main research question we will try to answer in the present thesis is which linguistic knowledge sources are most useful for word sense disambiguation, more specifically word sense disambiguation of Dutch. Therefore, the structure of the thesis is based on the various levels of linguistic information tested for word sense disambiguation, including morphology, information on the syntactic class of a particular ambiguous word, and the syntactic structure of the entire sentence containing an ambiguous word. Each source of linguistic knowledge is tested and evaluated individually in order to assess its value for word sense disambiguation. Finally, combinations of knowledge sources are investigated and evaluated.

The goal of our project was to develop a tool which is able to automatically determine the meaning of a particular ambiguous word in context, a so called word sense disambiguation system. In order to achieve this, we make use of the information contained in the context—similar to what humans do. So we use the words surrounding the ambiguous word, and additional underlying information, such as syntactic class and structure, to build a stat-
1.2 Overview

Chapter 2 contains an overview of word sense disambiguation, starting with an outline of the problem of word sense disambiguation and the difficulty of defining word senses. We then continue with an elaboration of the different approaches possible and the various information types used for sense disambiguation in computational linguistics. Next, a crucial, yet difficult issue in word sense disambiguation is addressed, namely the problem of evaluation. A description of the general approach adopted in this thesis concludes this chapter.

In chapter 3, preliminary experiments with pseudowords instead of real ambiguous words are reported on, investigating the importance of corpus size and frequency of context words. Furthermore, the equivalence between employing pseudowords or real ambiguous words to test word sense disambiguation algorithms is examined. The main conclusion is that the task of disambiguating pseudowords and real ambiguous words is not comparable.

The experimental setup used in the remainder of this thesis is introduced in chapter 4. We describe the classification algorithm and smoothing techniques as well as the corpus employed. A detailed explanation of the system and its implementation, as well as first results make up the rest of the chapter. These first results using only the context for disambiguation show that maximum entropy works well as a classification algorithm for word sense disambiguation when compared to the frequency baseline.

Chapter 5 presents a variation on the word sense disambiguation system introduced, the “lemma-based” approach. It tests the hypothesis that lemmas as bases for classifiers improve generalization and therefore accuracy. Comparing the lemma-based approach with the (traditional) word form approach on the Dutch SENSEVAL-2 data shows a significant improvement when lemmatization is used. Furthermore, the resulting word sense disambiguation system is smaller and more robust. We can conclude from this that the lemma-based approach is a better alternative than the word form-based approach. A detailed description and evaluation of a newly built stemmer/lemmatizer for Dutch (a necessary pre-processing tool for the lemma-based approach) are included, too.

Extending our word sense disambiguation system with information on part-of-speech and reporting on its impact on word sense disambiguation is the subject of chapter 6. We were especially interested in the importance
of the quality of the part-of-speech tagger used during pre-processing. We therefore compare the accuracy of our system including the part-of-speech of the ambiguous word generated by three different part-of-speech taggers. Two conclusions can be drawn from our results: first, that the most accurate tagger on a stand-alone task also outperforms the other taggers on the word sense disambiguation task, and second, that including information about the part-of-speech of the ambiguous word increases performance significantly. Including parts-of-speech of the context leads to an even bigger improvement of the disambiguation accuracy achieved.

The addition of deep linguistic knowledge, in the form of syntactic dependency relations, is discussed and evaluated in chapter 7. The results of our maximum entropy word sense disambiguation system including dependency relations are preceded by a detailed explanation of Alpino, the wide-coverage parser for Dutch used to annotate the data, as well as a description of the dependency relations employed. The results show that adding dependency relations to our statistical disambiguation system results in a significant increase in performance compared with all results presented earlier. The best results on the tuning data are achieved with a combination of features, including the part-of-speech of the ambiguous word, the context, and the dependency relations linked to the ambiguous word.

Chapter 8 presents the results on the Dutch Senseval-2 test data with the best model based on the tuning evaluation. First, we summarize our findings using the training data in a leave-one-out approach. Then, the results on the test data are presented. The first conclusion we reach is that the best model on the tuning data including syntactic information also works best on the test data. When applying the same model in a comparison between the word form-based approach and the lemma-based approach, we find that the lemma-based approach using dependency relations as features achieves the best overall performance of our system on the test data. In a last step, we compare our best model to another word sense disambiguation system which, to the best of our knowledge, has produced the best results for Dutch to date. Our system achieves significantly higher disambiguation accuracy than the other model which makes it state-of-the-art for Dutch word sense disambiguation. This is mainly due to the combination of the lemma-based approach and the integration of deep linguistic knowledge in the form of dependency relations.

We conclude in chapter 9 with some final remarks on the findings presented in the present thesis and thoughts on future work.