The Multilingual Forest – Investigating High-quality Parallel Corpus Development

Yvonne Adesam

The Multilingual Forest

Investigating High-quality Parallel Corpus Development

Yvonne Adesam

ISBN 978-91-7447-536-4

Printed in Sweden by Universitetsservice AB, Stockholm 2012
Distributor: Department of Linguistics, Stockholm University

Abstract

This thesis explores the development of parallel treebanks, collections of language data consisting of texts and their translations, with syntactic annotation and alignment, linking words, phrases, and sentences to show translation equivalence. We describe the semi-manual annotation of the SMULTRON parallel treebank, consisting of 1,000 sentences in English, German and Swedish. This description is the starting point for answering the first of two questions in this thesis.

- What issues need to be considered to achieve a high-quality, consistent, parallel treebank?

The units of annotation and the choice of annotation schemes are crucial for quality, and some automated processing is necessary to increase the size. Automatic quality checks and evaluation are essential, but manual quality control is still needed to achieve high quality.

Additionally, we explore improving the automatically created annotation for one language, using information available from the annotation of the other languages. This leads us to the second of the two questions in this thesis.

- Can we improve automatic annotation by projecting information available in the other languages?

Experiments with automatic alignment, which is projected from two language pairs, L1–L2 and L1–L3, onto the third pair, L2–L3, show an improvement in precision, in particular if the projected alignment is intersected with the system alignment. We also construct a test collection for experiments on annotation projection to resolve prepositional phrase attachment ambiguities. While majority vote projection improves the annotation, compared to the basic automatic annotation, using linguistic clues to correct the annotation before majority vote projection is even better, although more laborious. However, some structural errors cannot be corrected by projection at all, as different languages have different wording, and thus different structures.

Sammanfattning

I denna doktorsavhandling utforskas skapandet av parallella trädbanker. Dessa är språkliga data som består av texter och deras översättningar, som har märkts upp med syntaktisk information samt länkar mellan ord, fraser och meningar som motsvarar varandra i översättningarna. Vi beskriver den delvis manuella uppmärkningen av den parallella trädbanken SMULTRON, med 1.000 engelska, tyska och svenska meningar. Denna beskrivning är utgångspunkt för att besvara den första av två frågor i avhandlingen.

- Vilka frågor måste beaktas för att skapa en högkvalitativ parallell trädbank?

De enheter som märks upp samt valet av uppmärkningssystemet är viktiga för kvaliteten, och en viss andel automatisk bearbetning är nödvändig för att utöka storleken. Automatiska kvalitetskontroller och automatisk utvärdering är av vikt, men viss manuell granskning är nödvändig för att uppnå hög kvalitet.

Vidare utforskar vi att använda information som finns i uppmärkningen, för att förbättra den automatiskt skapade uppmärkningen för ett annat språk. Detta leder oss till den andra av de två frågorna i avhandlingen.

- Kan vi förbättra automatisk uppmärkning genom att överföra information som finns i de andra språken?

Experimenten visar att automatisk länkning som överförs från två språkpar, L1–L2 och L1–L3, till det tredje språkparet, L2–L3, får förbättrad precision, framför allt för skärningspunkten mellan den överförda länkningen och den automatiska länkningen. Vi skapar även en testsamling för experiment med överföring av uppmärkning för att lösa upp strukturella flertydigheter hos prepositionsfraser. Överföring enligt majoritetsprincipen förbättrar uppmärkningen, jämfört med den grundläggande automatiska uppmärkningen, men att använda språkliga ledtrådar för att korrigera uppmärkningen innan majoritetsöverföring är ännu bättre, om än mer arbetskrävande. Vissa felaktiga strukturer kan dock inte korrigeras med hjälp av överföring, eftersom de olika språken använder olika formuleringar, och därmed har olika strukturer.

Acknowledgements

A great number of people have helped and supported me during my years as a PhD student. First, I would like to thank my supervisors Martin Volk and Joakim Nivre. Martin's enthusiasm sparked my love for computational linguistics in general, and parallel treebanks in particular. Without his encouragement, I probably would not have pursued a PhD. Joakim has been invaluable in helping me understand the statistical and formal foundations, and has always been there to answer questions and offer insightful advice.

Thanks to the Department of Linguistics at Stockholm University, for supporting me, even when I wasn't physically in Stockholm any more. And thanks to GSLT, the Swedish graduate school of language technology, for funding my PhD education, and offering an environment in which to thrive and meet other students and researchers. Thanks also to the Department of Philosophy, Linguistics and Theory of Science (FLoV) at the University of Gothenburg, and especially to Robin Cooper and Jan Lif, for kindly allowing me to breathe the scientific air at their department.

I also want to thank my colleagues and fellow PhD students at the Department of Linguistics in Stockholm, FLoV, Språkbanken and the Centre for Language Technology (CLT) in Gothenburg, and GSLT, for numerous discussions over coffee, lunch, or dinner, as well as end-game pep talks.

Thanks also to the sysadmins I've come to rely on at different points in time, and to the enthusiastic developers who have created the TreeAligner, without which the alignment work would have been almost impossible, and to all the annotators for the work they have put into continuing the annotation I once started, to build what eventually became the SMULTRON parallel treebank. Thanks to Markus Dickinson, Martha Dís Brandt, Gerlof Bouma, Martin Kaså, Kristina Nilsson Björkenstam, Johan Bondefelt, and Robert Adesam for reading and commenting on parts of the thesis.

Finally, thanks to my family and friends. A special thank you to my sister Miriam, for always believing in me. And to Robert and Alexander, for life, in love.

Contents

List of Figures

List of Tables

1. Introduction

Language is one of the fundamental means of human communication. Without language, all our lives would appear very different. To study language is thus to study a part of what defines us as humans.

Large collections of language data, called corpora, are one means of studying language. Corpora have become invaluable, not just for language study, but also for a wide range of applications, which are aimed at modeling natural language in some way. Adding mark-up, annotation, to corpora has further improved their usefulness. In this thesis, we focus on corpora consisting of written texts, and explore a particular type of corpora, called parallel treebanks. Parallel treebanks consist of translated texts in two or more languages, which have been syntactically annotated and where corresponding parts are linked through alignment. They are used for a wide range of applications, such as machine translation and translation studies, and teaching computers how to annotate syntactic structure or alignment.

1.1 Thesis Objectives

Most corpus annotation is a time-consuming and labour-intensive task. Parallel treebanks contain several levels of annotation and are thus especially costly to produce. Additionally, for the treebank to be useful, large data sets are often required, with millions of words (see, e.g., Nivre et al., 2005).

To get the size required, we can collect any parallel text, and use automatic methods to annotate the data. There are numerous taggers, parsers, and aligners available off-the-shelf or for training on already annotated data. While this may seem easy in theory, in practice, there are a number of obstacles. First, tools and/or annotated data are generally only available for a number of the largest languages in the world. Second, automatic annotation (as well as human annotation) induces errors, and for the data to be useful for many applications we still need to check and manually correct the annotation.

The main theme explored in this thesis is how to create a parallel treebank of good quality. This is done in two separate ways. Part I gives an in-depth description of how the SMULTRON parallel treebank was created, consisting of two types of texts in German, English, and Swedish. We also explore how

to assure quality in a parallel treebank, which is a crucial issue for usability. Starting from this description, we will discuss a more general first question.

Question 1 *What issues need to be considered to achieve a high-quality, consistent, parallel treebank?*

While much of the work described in Part I was done manually or semi-automatically, Part II explores ways of automatically building or increasing a parallel treebank, in particular, methods of projecting knowledge from multiple languages, known as multilingual annotation projection (see Section 6.2). We focus on improving automatically created annotation. The second, more specific, question is thus the following.

Question 2 *Can we improve automatic annotation by projecting information available in the other languages?*

A recent PhD thesis (Haulrich, 2012) also explores using cross-lingual information to create better parallel treebanks. It differs from this thesis in that he only deals with automatic creation of dependency structures, whereas we focus on constituency structures and are more generally concerned with high-quality annotation, which includes both manual and automatic processing at various annotation levels.

1.2 Thesis Outline

The current chapter introduces the task of this thesis, and the research questions. Chapter 2 continues by defining parallel treebanks and what we consider to be important aspects of quality. The thesis is then divided into two parts.

Part I consists of Chapters 3, 4, and 5. They describe the creation of the SMULTRON parallel treebank, and explore how to build a high-quality parallel treebank. This ties together a number of previous publications, giving a complete description of the whole process. Additionally, this defines the parallel treebank that will be used as a gold standard in later experiments. Chapter 3 describes how the monolingual treebanks were created, from part-of-speech tagging to parsing, In Chapter 4, we explore how the alignment was added to tie the treebanks together. Finally, in Chapter 5, we discuss automatic quality checking of annotation.

Part II of this thesis, consisting of Chapters 6, 7, 8, and 9, deals with experiments on multilingual annotation projection. First, we automatically create alignment for the parallel treebank and explore projecting alignment across language pairs. Second, we investigate the structural improvements we can make by projecting knowledge between the languages, across the alignment.

For the purpose of this thesis, we will focus on a sub-area of syntax, which is generally problematic in automatic parsing, namely PP-attachment. This is the question of correctly attaching a prepositional phrase, or for dependency parsing, an argument headed by a preposition, in the tree structure. We explore projection of information available in one, two, or three languages, as well as different ways of selecting what information to project.

Parts of this thesis elaborate work by the author that has been published elsewhere, and will be referenced in the text.[1] Other parts contain recent, unpublished work. In general, the work that Part I is based on is collaborative work, while most of the experiments described in Part II were carried out by the author.

[1]The author of this thesis has changed her name from Yvonne Samuelsson to Yvonne Adesam, which are thus the same author.

2. Defining Parallel Treebanks

In this thesis, we will explore parallel treebanking. Thus, we need to properly define what a parallel treebank is. We will discuss what characterizes corpora and treebanks, and also give a more formal definition of parallel treebanks.

2.1 From Corpora to Parallel Treebanks

In this section we will define what parallel treebanks are by looking at a spectrum of text collections. They range from plain text corpora to parallel corpora, and from syntactically annotated corpora, treebanks, to parallel treebanks.

2.1.1 Corpora and Parallel Corpora

A *corpus* is "a body of naturally-occurring (authentic) language data which can be used as a basis for linguistic research" (Leech and Eyes, 1997, p. 1).[1] One of the first large corpora was what is now generally referred to as the Brown corpus, the Brown University Standard Corpus of Present-Day American English (Francis and Kucera, 1964, Kucera and Francis, 1967), a one million word corpus compiled in the 1960s.

We define a *parallel corpus* as a collection of naturally-occurring language data consisting of texts and their translations. There has, however, been some term confusion, as genre-based text collections in different languages also have been called parallel corpora (see the introductory chapter in Véronis, 2000). We will distinguish between parallel corpora and comparable corpora, where comparable corpora do not contain actual translations, but are collections of texts in multiple languages within the same genre, about the same topics. For further discussions see also, e.g., McEnery and Wilson (2001), Tiedemann (2003) and Merkel (1999b).

Parallel corpora, and in particular aligned corpora, are used in a variety of linguistic fields. They have become essential in the development of, e.g., machine translation (MT) systems, and offer useful information for contrastive

[1]The data used to build a corpus may range from spoken data, over transcribed speech, to written text. In this thesis, we will focus on the latter.

linguistics, translation studies and translator's aids, foreign language pedagogy, and cross-lingual information retrieval and word-sense disambiguation (see, e.g., Botley et al., 2000, Leech and Eyes, 1997, McEnery and Wilson, 2001, Véronis, 2000, and references therein). An important field when creating parallel corpora is research about alignment and in particular automatic alignment. Alignment defines links between the translations, showing which part of a text in one language corresponds to which part in another language. These links connect words and strings of words in the texts by different methods (see Chapter 6.1).

2.1.2 Treebanks and Parallel Treebanks

Some corpora contain annotation (sometimes specifically called *annotated corpora*). This annotation often consists of part-of-speech tagging ('word classes') and possibly morphological information. In many situations, however, more annotation is necessary to explore and exploit the corpus. We consider a *treebank* to be a special type of annotated corpus, a collection of sentences, which have been grammatically tagged and syntactically annotated, "a linguistically annotated corpus that includes some grammatical analysis beyond the part-of-speech level" (Nivre, 2008, Nivre et al., 2005). Each sentence is thus mapped to a graph, which represents its syntactic structure. Some examples of treebanks are the Swedish Talbanken (Teleman, 1974), the Lancaster Parsed Corpus (Garside et al., 1992) and the Penn treebank (Marcus et al., 1993).

The annotation of part-of-speech (PoS) tags is often done automatically, while syntactic parsing, due to lower accuracy rates, usually entails some manual labour. The term *treebank* is sometimes used interchangeably with the term *parsed corpus*. We define a treebank as a corpus with manually checked syntactic annotation (manually or automatically annotated), while a parsed corpus is an automatically constructed corpus, which has not been manually checked.

The graphs in a treebank are often represented as trees, with a top node (generally called a sentence or root node) and tokens (words) as leaf-nodes (terminal nodes). One question in creating a treebank is of course which grammatical theory to follow. The most common grammars in treebanking are phrase structure (constituent, or bracketing) grammars and dependency (predicate-argument, or functional) grammars, and hybrids thereof. Several treebanks also contain semantic annotation, often together with syntactic annotation.

Phrase structure grammars look at the constituent structure of the sentence, creating a hierarchic tree. Parsing can be full, which is a detailed analysis of the sentence structure, or skeletal, which depicts a simplified constituent structure with fewer syntactic constituent types, where the internal structure of some constituents is ignored. Dependency grammars focus on the relation between

words, representing syntactic dependencies between words directly. They look at the grammatical functions of the words and mark the dependencies between them, meaning that the verb and its valency often creates the sentence structure. There are good arguments for both approaches (see, e.g., Cyrus and Feddes, 2004, Gildea, 2004, Johansson and Nugues, 2008).

Parallel treebanks are treebanks over parallel corpora, i.e., the 'same' text in two or more languages. As in the case of parallel corpora, these multilingual texts are translations, where one text might be the source text and the other texts are translations thereof, or where all texts are translations from a text outside the corpus. In addition to the monolingual annotation, they usually contain alignment information.

Treebanks have become a necessary resource for a wide area of linguistic research and natural language processing. For example, they are important tools in teaching computers tagging and parsing (see, e.g., Bod, 1993, Charniak, 1996, Collins, 1999). The field of parallel treebanks has just recently evolved into a research field, but they have shown to be instrumental for, e.g., syntax-based machine translation (e.g., Lavie et al., 2008, Tinsley, 2010, Tinsley et al., 2007a), and as training or evaluation corpora for word and phrase alignment (e.g., Tiedemann, 2010).

In recent years, there have been a number of initiatives in building parallel treebanks (see also Abeillé, 2003, Nivre et al., 2005). One of the early large parallel treebank projects was the Prague Czech-English Dependency Treebank (Čmejrek et al., 2003). It was built for the specific purpose of machine translation, and consists of parts of the Penn Treebank and their translation into Czech. Version 1 contains just over 20,000 sentences.[1] The translators were asked to translate each English sentence into one Czech sentence and to stay as close to the original as possible. The texts were then annotated with tectogrammatical dependency trees (the tectogrammatical structure being the underlying deep structure).

Croco (Hansen-Schirra et al., 2006) is a one million word constituent structure English-German parallel treebank for translation studies. The English-Swedish parallel treebank LinES (Ahrenberg, 2007) is a dependency structure treebank aimed at the study of variation in translation, and the English-French HomeCentre treebank (Hearne and Way, 2006) is a hand-crafted parallel treebank consisting of sentences from a Xerox printer manual, annotated with context-free phrase structure representations. The Korean-English treebank[2] was created for language training in a military setting and is annotated with constituent bracketing. Additional projects include experiments with parallel treebanking for the widely differing languages Quechua and Spanish (Rios

[1] See http://ufal.mff.cuni.cz/pcedt/.

[2] See http://ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2002T26.

et al., 2009), and a number of parallel parsed corpora (automatically parsed and aligned), e.g., the Dutch-English Europarl by Tiedemann and Kotzé (2009a).

A few projects apart from the SMULTRON project, which will be described in Part I of this thesis, have dealt with multilingual parallel treebanks. For example, the Nordic Treebank Network (NTN)[1] syntactically annotated the first chapter of Jostein Gaarder's novel Sofies verden (Gaarder, 1991) and its translations into a number of (mostly Nordic) languages, creating the Sofie treebank with a wide range of different annotation styles.[2] The Copenhagen Dependency Treebanks (Buch-Kromann and Korzen, 2010) consist of parallel treebanks for Danish, English, German, Italian, and Spanish. The parallel treebanks have multiple levels of annotation (e.g., discourse and anaphora) in addition to dependency structures. The English-Swedish-Turkish parallel treebank created at Uppsala University (Megyesi et al., 2010) is annotated with dependency structures and used in teaching and linguistic research to study the structural differences between the languages. The Japanese-English-Chinese treebank created in Kyoto (Uchimoto et al., 2004) contains newspaper text (part of it from the Penn treebank), which has been closely translated.

Next, we will define more formally what a parallel treebank is. This is followed by a short discussion about well-formedness, consistency, and soundness as important aspects of *quality*, a central concept in this thesis.

2.2 Formal Definition of Parallel Treebanks

Let us define formally what a parallel treebank is. The following definition of syntax graphs draws on König and Lezius (2003) and Hall (2008). A syntax graph consists of nodes V, edges E and a set of functions F mapping nodes and edges to labels L. The nodes are divided into two disjoint subsets, consisting of terminal nodes and possibly non-terminal nodes. In our case, the treebank consists of phrase structure trees, requiring at least one non-terminal per graph.

Definition 1 *A* syntax graph *for a sentence* $x = (w_1, ..., w_n)$ *is a triple* $G = (V, E, F)$*, where*

- $V = (V_T \cup V_{NT})$ *is a set of nodes, consisting of two disjoint subsets,*
  - $V_T = \{v_1, ..., v_n\}$*, the non-empty set of terminal nodes, one node for each token* w_i *in x,*
  - $V_{NT} = \{v_{n+1}, ..., v_{n+m}\}(m \geq 0)$*, the possibly empty set of non-terminal nodes,*

[1] See http://w3.msi.vxu.se/~nivre/research/nt.html.

[2] See http://hf.uio.no/iln/om/organisasjon/tekstlab/prosjekter/arkiv/sofie.html.

- $E \subseteq V \times V$ *is a set of edges,*
- $F = \{f_1, f_2, f_3\}$ *is a set of functions* $f_i : D_i \rightarrow L_i$*, where* $D_i \in \{V_T, V_{NT}, E\}$ *and* L_i *is a set of labels.*

A treebank is a collection of syntax graphs, where every graph corresponds to a sentence (or partial sentence) of a text or texts.

Definition 2 *A* treebank *is a set* $B = \{(x_1, G_1), ..., (x_n, G_n)\}$*, where every* x_i *is a natural language sentence and* G_i *is a syntax graph for* x_i*.*

An alignment A connects sentences of one treebank with sentences of another treebank. Generally, it is assumed that these treebanks contain texts that are translations of each other in two different languages. An alignment link is a triple (v_1, v_2, s), where v_1 is a node in B_1, v_2 is a node in B_2, and s is an alignment type, e.g., exact or approximate (see Chapter 4). It is possible to use one type of alignment link, or to use finer distinctions.

Definition 3 *Given two treebanks* $B_1 = \{(x_1, G_1^1), ..., (x_n, G_n^1)\}$ *and* $B_2 = \{(y_1, G_1^2), ..., (y_m, G_m^2)\}$*, we let* $V(B)$ *denote the set of all nodes occurring in some syntax graph* G_j*, so that* $(x_j, G_j) \in B$*. Let S be the set of possible alignment types. An* alignment *A between the two treebanks* B_1 *and* B_2 *is a relation* $A \subseteq V(B_1) \times V(B_2) \times S$*.*

A parallel treebank consists of two treebanks (in different languages where the texts are translations of each other) and the alignment connecting them, telling us what parts are translationally equivalent.

Definition 4 *A* parallel treebank *is a triple* $P = (B_1, B_2, A)$*, consisting of two treebanks and the alignment connecting them.*

2.3 Quality in Parallel Treebanks

The main question explored in this thesis is how to create a high-quality parallel treebank. We thus need to specify what we mean by quality, before turning to methods for achieving it. In particular, there are three aspects of quality that are crucial for this thesis, and they are well-formedness, consistency, and soundness.

A treebank that is well-formed is complete in the sense that each token and each non-terminal node is part of a sentence-spanning tree, and has a label. If a certain type of token should be excluded from the annotation, e.g., punctuation symbols or HTML-tags, all of them should be excluded. Well-formedness is

closely related to consistency. A consistent treebank is consistently annotated, which means, e.g., that the same token sequence (or part-of-speech sequence or constituent sequence) is annotated in the same way across the treebank, given the same context. It also means that sentence splitting and tokenization is uniform, following a fixed set of rules. Well-formedness and consistency are important issues when using a parallel treebank, whether as training data or for linguistic querying. They determine to what degree we can trust the information that is encoded in the parallel treebank.

Soundness is an important aspect of quality, which should motivate all decisions behind the compilation and annotation of the parallel treebank. This may mean that the parallel treebank conforms to a linguistic theory. At the very least, it should conform to sound linguistic principles. Related to this is also the question of what kind of information we encode in the annotation of a parallel treebank, and how this information can be extracted or accessed. The problem is that there is a trade-off between linguistic soundness, and simplifications made to create distinct categories, as well as simplifications for speed of (human) annotation.

Quality is an issue that has been somewhat ignored in the area of computational linguistics. Often a resource, e.g., the Penn treebank, is used as the truth, without regard to what is in the data, or to issues that could have been done better if not pursuing getting good scores when evaluating against this particular resource. Or, as put by Levin, "[t]he people who say they love data the most seem to be the most afraid of looking at it." (Levin, 2011, p. 14) It is of utmost importance to look at the data, and not just the global accuracy scores.

This concludes the introductory chapters, leading us to Part I of the thesis. There, we describe our own efforts in building the tri-lingual parallel treebank SMULTRON.

Part I

Creating the SMULTRON Parallel Treebank

3. Grammatical Annotation in SMULTRON

In this chapter, we describe the grammatical annotation of the SMULTRON parallel treebank. The purpose of this is twofold. First, this is the starting point for our discussion about which issues have to be considered when building a high-quality parallel treebank. Second, it describes the parallel treebank that is used as a gold standard for the experiments on augmenting automatic annotation, which we will turn to in Part II of this thesis.

SMULTRON, the Stockholm MULtilingual TReebank, is a multilingual parallel treebank.[1] Version 1 (Gustafson-Čapková et al., 2007) contains two different types of text, with around 1,000 sentences per language, in English, German and Swedish. The latest version, Volk et al. (2010), contains about 1,500 sentences for English, German and Swedish, and 500 of these also in Spanish. There is also a German-French part with about 1,000 sentences. In this thesis, however, SMULTRON refers to version 1, unless otherwise stated.

One part of SMULTRON contains the first two chapters of Jostein Gaarder's novel Sophie's World. The Norwegian original (Gaarder, 1991) has been translated into English (Gaarder, 1995), German (Gaarder, 1994) and Swedish (Gaarder, 1993). There are around 530 sentences in each language (there is some variation between the different language versions), with an average of about 14 tokens per sentence. The number of sentences and tokens can be found in Appendix A.

The Sophie text was initially chosen by initiative from the Nordic Treebank Network (NTN).[2] The NTN syntactically annotated the first chapter of the book in the Nordic languages. This text has been translated into a vast number of languages and it includes interesting linguistic properties such as direct speech. The fictional Sophie text is also interesting, in that many of the widely used corpora contain non-fiction texts, e.g., newspaper texts as in the Penn treebank (Marcus et al., 1993), or parliament proceedings as in Europarl (Koehn, 2002).

[1]We gratefully acknowledge financial support for the SMULTRON project by Granholms stiftelse, Rausings stiftelse and the University of Zurich.

[2]See http://w3.msi.vxu.se/~nivre/research/nt.html.

As we expanded the SMULTRON treebank, however, we wanted a different text type than the novel, and decided on economy texts, which were available in Swedish, English and German. The second part of the parallel treebank thus contains economy texts from three different sources. The first is a press release about the quarterly report (Q2 2005) from the multinational company ABB, the second is the Rainforest Alliance's Banana Certification Program and the third a part of the SEB bank's annual report 2004 (Report of the directors: Corporate Governance). There are about 490 sentences per language, with an average of around 22 tokens per sentence. The number of sentences and tokens can be found in Appendix A. These economy texts, besides having more tokens per sentence, generally are more complex and differ more in number of sentences and average number of tokens between the languages (e.g., five sentences in the English version have more than 100 tokens).

A parallel treebank, as well as any corpus, may be collected for a number of different reasons and these reasons are usually reflected more or less transparently in the text collection and the annotation. Generally, the topic of the text is of less importance when building a parallel treebank, except for from a genre perspective. Several corpora, e.g., the Brown corpus (Francis and Kucera, 1964) and the Stockholm-Umeå corpus (SUC, Ejerhed et al., 1992), consist of different text types, to try to be balanced or representative. However, a corpus can never be a complete representation of all of language, not even of written language (see, e.g., Francis, 2007, Zanettin, 2007). We cannot include all text ever written, for storage or accessibility reasons. Additionally, what is considered 'typical' language or text changes in different social settings and over time (compare with some more recent initiatives, e.g., Borin et al. (2010) of building diachronic corpora). Thus, although a carefully compiled corpus can be representative, it is by necessity a partial representation of language.

Parallel corpora contain translations. The nature of these translations may vary. For instance, the corpus can contain a text and its translations, or, as is the case for the Sophie part of SMULTRON, all texts can be translations of a text not included. Also, the translations can be produced by a translator in a freer sense, trying to capture the essence of the original text (which generally applies to novels and poems) or as stricter translations for a particular purpose other than carrying the information, as was done, e.g., for the Prague Czech-English Dependency Treebank (Čmejrek et al., 2004).

Apart from the obvious differences between languages, there may be traces of the original language in a translation, often called translationese (see, e.g., Baker, 1993, Gellerstam, 2007). This means that the language of a translation differs from the language of an original text, and would imply that we need to look at the texts of a parallel corpus strictly in terms of source and target texts. However, according to, e.g., Nord (1997), any translation is the translators

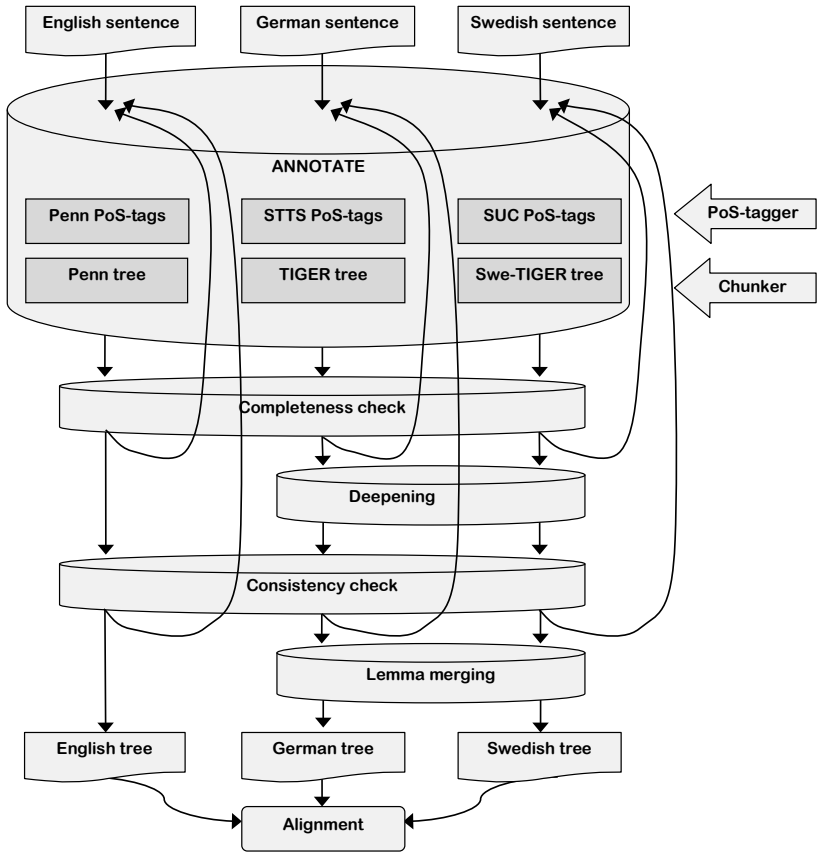


Figure 3.1: The process of creating the SMULTRON parallel treebank.

interpretation of the original text. "No one can claim to have *the* source text at their disposal to transform it into *the* target text" (Nord, 1997, p. 119). The offer of information in a source text is simply the starting point for the offer of information that is formulated in the target text (Nord, 1997). This view suggests that we see both source and target texts as texts in their own right. The information about the origin of a translation is, however, important knowledge to encode in the meta-data of a parallel corpus. It may help us understand why some constructions appear in a text.

The SMULTRON parallel treebank is created in several steps. An overview of the process is shown in Figure 3.1. After some initial pre-processing (including, if necessary, scanning and OCR, followed by sentence splitting and

tokenization), the texts are part-of-speech (PoS) tagged, as is described in Section 3.1.1. The German and Swedish texts are lemmatized, as described in Section 3.1.2, information that is later added to the PoS-tagged and parsed sentences. The texts of the three languages are then parsed, as described in Section 3.2. In addition, the German and Swedish trees are deepened, as described in Section 3.2.2. Finally, the three treebanks are joined pairwise by alignment, described in Chapter 4. The quality of the annotation is checked during the process, as described in Chapter 5. The completeness check makes sure that the tagged and parsed data is well-formed, so that no words or phrases are left unannotated. This is required before applying further automatic tools. Consistency checks are then carried out for all layers of annotation, although this has been done automatically only for PoS-tags and the syntactic structure, as shown in Figure 3.1.

3.1 Word-level Annotation

3.1.1 Part-of-speech Tagging

Texts that are incorporated into a corpus require preprocessing, starting with tokenization, where generally one graph word is considered as one token. Punctuation symbols, not counting stops in abbreviations and German ordinal numbers, are considered separate tokens. For English, genitive *'s* and negation *'t* are considered separate words from the token they are clitically attached to.

Additionally, the texts are split into sentences. Sentence splitting mainly follows sentence final punctuation, with a few exceptions. For instance, *" Bow-wow ! Bow-wow ! "* is considered one interjection, despite of the intermediate exclamation mark after the first *bow-wow*. There are, however, some cases where sentences have been split differently between the languages. One Swedish sentence, for example, consists of the interjection *"Äsch!*, while the next sentence starts with *Det är bara du som är så van vid världen*. The German sentence corresponding to the second Swedish sentence starts with *"Pah! Du hast dich nur so gut in der Welt eingelebt*. The English translation does not contain anything corresponding to the first Swedish sentence (or the beginning of the German sentence), *You've just grown so used to the world*. Sentence splitting should be uniform between the languages, for the parallel treebank to be consistent (as defined in Section 2.3). We thus consider such inconsistencies errors, which should be taken care of for future releases of the SMULTRON parallel treebank.

While tokenization and sentence splitting are necessary for annotation, we need to be aware of what this does to the text. Texts are split into sentences and tokens, and we shift focus from the text as a whole, its structure of chapters and

paragraphs, to smaller parts of the text, generally no larger than the sentence. This is especially apparent when we explore parallel treebanks in visualization tools, where often only one sentence or sentence pair is visible on the screen at the same time. While this is natural, as the trees for a whole text simply require too much space to fit on a screen, we need to remember that a larger context is necessary to fully understand a text. For several areas of linguistic research a larger context than the sentence is crucial, e.g., for anaphora resolution. Preferably, the removed formatting should be substituted, for example by XML-tags, to retain the information. This was not done in SMULTRON, where such information is only partially and indirectly present through some HTML-tags for, e.g., headings in the economy part of the parallel treebank.

After the pre-processing step, the texts were annotated in a number of layers. The treebanks for all three languages were separately syntactically annotated (tagged and parsed) with the help of the treebank editor ANNOTATE, developed at the University of Saarbrücken.[1] It includes a statistical PoS-tagger and chunker for German by Thorsten Brants.

The first layer of annotation is determining the PoS-tag of each token. We (semi-automatically) annotated the German sentences with PoS-tags following the STTS (Stuttgart-Tübingen TagSet, Thielen et al., 1999). The STTS is a tagset for annotating German text corpora developed by the Institut für maschinelle Sprachverarbeitung of the University of Stuttgart and the Seminar für Sprachwissenschaft of the University of Tübingen. It has been used for the NEGRA[2] and TIGER[3] corpora, German newspaper text corpora with 355,000 and 900,000 words, respectively. The English sentences were tagged using the Penn Treebank PoS-tagset (Marcus et al., 1993), developed for the Penn treebank project, which includes the 1 million word Wall Street Journal material. The Swedish treebank is annotated with an adapted version of the SUC (Ejerhed et al., 1992) tagset. SUC is a 1 million word representative Swedish corpus, annotated with manually checked PoS-tags, morphological tags, lemmas and name classes. We have only used the SUC base tags, instead of the full morphological forms of the tags. To account for some of the information loss, we added a subdivision of the PoS-tags for verbs, incorporating tense and finiteness. We thus distinguish between VBFIN (finite), VBIMP (imperative), VBINF (infinitive), and VBSUP (supine). This makes the Swedish tagset more similar to the German and English tagsets.

The PoS classification is a surrogate representation for the function of each word in a sentence, based on morphological, syntactic and semantic criteria. While it is easy to show that words can be divided into classes with different

[1] See http://coli.uni-sb.de/sfb378/negra-corpus/annotate.html.

[2] See http://coli.uni-saarland.de/projects/sfb378/negra-corpus/.

[3] See http://ims.uni-stuttgart.de/projekte/TIGER/TIGERCorpus.

behaviour, it is not always easy to decide what class a certain word belongs to. For example, some words are truly ambiguous, even given the context. In such cases, one solution for annotation is to have multiple tags for one token. Generally, however, as in the case of SMULTRON, only one tag is assigned to each token. The annotators thus have to disambiguate all words.

There is no general agreement about a classification that works for all languages, or even what set of criteria such a classification would be based on. Petrov et al. (2012) propose a universal PoS-tagset containing 12 categories. They also provide a mapping from the tagsets used in 25 treebanks. While this facilitates, e.g., evaluation of parsers, the tagset may be too coarse for more detailed linguistic analysis of various languages. Different languages have different lexical categories, and different needs for the representation and even within one language there may be different representations, depending on whether the representation focuses on syntax, semantics, or pragmatics. First, we need to agree on the item (word) to tag. This can range from looking at the full form (including morphological information) of each word to the lemma (word class information without morphology) of each word. The decision here defines how rich a tagset is.

Secondly, it is a question of the size of the tagset. By comparing the tagsets for the three languages in SMULTRON (which are considered standards for the respective language), we already see some differences. The English Penn tagset has 48 tags, the German STTS-tagset 57 and the Swedish SUC tagset only 29 (counting the base tags, including the three added tags for verbs).[1] For some tags, the classification is simply more fine grained for one of the tagsets. While the German and Swedish tagsets distinguish between nouns and proper nouns (names), the English tagset also distinguishes between singular (or mass) and plural. While the German set has three tags for punctuation (comma, sentence final and other), the English set has seven different tags and the Swedish set only one. The Penn tagset contains PoS-tags that are specific for newspaper text and especially texts about economics (the Wall Street Journal). Thus, some of these tags, e.g., # and $, are unnecessary for other types of text. For Swedish and German there is only one tag for non-word symbols.

For some types of tags, the difference in size of the tagset is the result of language differences, rather than classification granularity. The Penn tagset presupposes a tokenization where the possessive ending is stripped from nouns, giving it a tag of its own. The Swedish tagset has one tag for prepositions, labelling 1,961 words in SMULTRON, while the English tagset has one tag for prepositions and subordinating conjunctions, labelling altogether 2,337 words.

[1]Not all of these tags are used in SMULTRON, for any of the tagsets.

For German, however, there are several tags, differentiating between postpositions, left- and right-side prepositions and prepositions clitically joined with an article, labelling 1,668 words. Additionally, there are two German tags for subjunctions, depending on whether the following verb is finite or non-finite.

The issue of pronouns and other function words is not easy to compare between the three tagsets in SMULTRON. English and Swedish have similar tags, distinguishing between personal and possessive pronouns and four different kinds of interrogatives (determiners, pronouns, possessive pronouns and adverbs). English also has a tag for predeterminers. German has 16 different pronouns and interrogatives. Swedish has a separate tag for ordinal numbers, while German and English view them as adjectives.

The Penn tagset contains some tags, such as the TO and IN tags, which are used for certain tokens, ignoring a linguistically based classification. For example, the TO tag is used for all instances of the word *to*, whether it is a preposition or an infinitive marker. Such a classification allows for faster processing, and is sometimes easily resolveable (e.g., a *to* followed by a noun phrase versus followed by a verb phrase). However, one should be careful when adding classes to a classification, which do not contribute any information.

The differences we have discussed are visible from the comparison of these three tagsets, for three languages that are similar. Comparing them to other tagsets would show more divergences. The different decisions affect the quality of a treebank. A finite set of tags is efficient to handle and to predict. However, a very large tagset (e.g., one tag for each word type) would not be informative, and a rich, more fine-grained, tagset may render more tagging errors. Using morphological information in the tagset for a morphologically rich language is informative, but increases the size of the tagset and hence the computational difficulty. On the other hand, a very small tagset (e.g., only one) would be easy to predict, but would also be uninformative. A coarser tagset limits our perception of the corpus, since we can only see what we can search for and what we can search for is in part dependent on the tagset.

3.1.2 Lemmatization and Morphological Annotation

For German and Swedish, we added lemmas and morphological information, in addition to the PoS-tags. For the lemmas, the goal was to give each token exactly one lemma. For Swedish, we used the morphology system Swetwol (Karlsson, 1992) and for German, the German version Gertwol. Word boundaries are marked by #. Gertwol marks suffixes with ~.

There are a number of tokens, which do not get a lemma. XML tags are left unannotated, as well as punctuation symbols (which are, however, tagged). Numbers that consist only of digits (and related symbols) do not receive a

lemma either, e.g., *30'261*, and *+4.3*, and neither do Roman numbers, e.g., *IV*.

If Gertwol or Swetwol provides more than one lemma for a given token (and its PoS), then the most appropriate lemma is chosen in the given context. For German, the selection of nouns, adjectives and verbs, in particular the disambiguation between different possible segmentations, is done automatically, as specified in Volk (1999). One example is the word *Zweifelsfall* ('doubt, case of doubt'), with the possible segmentations *Zwei#fels#fall* ('two rock fall', unlikely) and *Zweifel\s#fall* ('doubt case', correct). The disambiguation between multiple lemmas for other word classes for German is done manually. For example, perfect participle forms of verbs that can belong to two different verbs need to be manually checked, e.g., *gehört*, which could be a form of *hören* ('hear') or *gehören* ('belong').

In the case of multiple lemmas for a given token for Swedish, the most appropriate lemma is chosen manually. We also add information to the Swetwol lemma by marking the gap 's' (sometimes called interfix) with '\s'. For example, the correct lemmatization of *arbetsförhållanden* ('working conditions') is *arbet\s#förhållande* (*work condition'), not *arbetsför#hållande* ('able-bodied keeping') or *arbetsför#hål#land* ('able-bodied hole land').

If Gertwol or Swetwol does not provide a lemma, then the lemma is chosen by the human annotator. If the word is a proper name (German PoS NE, Swedish PM), the lemma is identical with the word form, except for compound names, e.g., geographical names, or names in genitive, in which case the genitive suffix -s will be removed. For example, *Amerikas* is lemmatized *Amerika*, *ABB:s* is lemmatized *ABB*, and *Nordamerika* ('North america') is lemmatized *Nord#amerika*. For foreign words (German PoS FM, Swedish UO), the lemma is identical with the word form, unless it is an English word in plural. In that case the suffix -s is removed, lemmatizing, e.g., *Technologies* as *technology*.

Foreign words are additionally subclassified by a label specifying the language. The label is the two-character ISO language code. For example *Board* is labelled *Board EN*, and *Crédit* is labelled *Crédit FR*. Foreign words that are established enough in a language to be considered loan words, and thus are no longer PoS-tagged as foreign words, do not get a language label. A foreign word that is part of a compound (including hyphenated compounds) with a German or Swedish word, receives the PoS-tag of that word, and a combined language label reflecting the origin of the parts. For example, the German *Building-Systems-Geschäft* ('building systems business') is tagged NN, with the language label EN-EN-DE.

If the token is an abbreviation, then the full word is taken as the basis for lemmatization. For example, German *Mio.* is lemmatized as *Million* ('million'), Swedish *kl* is lemmatized *klocka* ('clock', in time expressions), and % is lemmatized in German as *Prozent*, and in Swedish as *procent* ('percent').

Acronyms, however, are not spelled out, e.g., *USA* is lemmatized as *USA*.

In two cases the lemmatization in SMULTRON deviates from the Gertwol/Swetwol suggestions. First, if the token is an elliptical compound, then the full compound is taken as the basis for lemmatization. For example, the German expression *Energie- und Automationstechnik* ('energy and automation technology') is lemmatized as *Energ~ie#techn~ik und Automat~ion\s#techn~ik*. Similarly, the Swedish multi-word unit *lågspänningsbrytare och -omkopplare* ('low-voltage breakers and switches') is lemmatized *låg#spänning\s#brytare och låg#spänning\s#omkopplare*. The second point of deviation concerns the lemmatization of determiners. For Swedish, we follow the SUC conventions. For German, if the token is a determiner or an attributive pronoun, we choose the determiner in the correct gender as the lemma.

Each noun (PoS NN) gets a label specifying its grammatical gender. German names (PoS NE) also receive such a label, while this is not yet done for Swedish names (PoS PM). German nouns and names are feminine, masculine, neuter, or none (e.g., family names have no gender). Swedish nouns are either uter, neuter, or none. Foreign words (German PoS FM, Swedish UO) and nouns that occur only in plural (pluralia tanta) do not get a gender label.

There are at least two problematic areas. First, many German nouns derived from verbs have a homograph noun with a different gender. They must be manually inspected and corrected. Some examples are *das Packen* ('the (act of) packing') vs. *der Packen* ('the bundle'), and *das Rätseln* ('racking one's brains') vs. *den Rätseln* ('the riddles'). Secondly, nouns that have different genders in different readings need to be manually checked. Some examples are *der Moment* ('the moment') vs. *das Moment* ('the circumstance') and *der Flur* ('the hallway') vs. *die Flur* ('farmland, meadow').

Misspelled words are not corrected, as a principle of faithfulness to the original text. For lemmatization, however, the (imagined) corrected word is taken as the basis. For example, the German *abgeschaft* (correctly *abgeschafft*, 'abolished'), gets the lemma *ab|schaff~en*, and the Swedish *avfallshateringsregler* (correctly *avfallshanteringsregler*, 'waste management rules') is lemmatized *avfalls#hantering\s#regel*.

3.2 Phrase-level Annotation

The next layer of information, on phrase and sentence level, is the syntactic annotation. This annotation tries to make the structure of a sentence explicit. Different formalisms have different views of the structure. One possible view is that words are combined into larger building blocks, such as phrases, which have different functions in a sentence. Another view is that each word relates

to, or is dependent upon, another word in the sentence, and that these dependencies are defined by their function.

3.2.1 Parsing

After having PoS-tagged the SMULTRON texts, we semi-automatically parsed them in ANNOTATE. German and English have seen large treebank projects that have acted as a standard for treebanking in these languages. Thus, the English sentences were parsed according to the Penn Treebank grammar (Bies et al., 1995). For the German sentences, we followed the NEGRA/TIGER annotation scheme (Brants et al., 2002, Skut et al., 1997). While the two annotation schemes differ in several respects, differences which will be discussed in Section 3.2.3, both prescribe phrase structure trees, where tokens are terminal and phrases are non-terminal nodes. Nodes are connected to each other through edges, which can have function labels. The connected nodes form a tree structure, from the terminal nodes to a single root node for each sentence.

There has been an early history of treebanking in Sweden, dating back to the 1970s (see, e.g., Nivre, 2002). The old annotation schemes were difficult for automatic processing, as in the case of Talbanken (annotated with the MAMBA scheme, Teleman, 1974),[1] or too coarse-grained, as in the case of Syntag (Järborg, 1986). Without a suitable Swedish treebank to train a Swedish parser, we developed our own treebanking guidelines for Swedish inspired by the German guidelines.

Initially, we mapped the Swedish PoS-tags in the Swedish sentences to the corresponding German tags. Since the German chunker works on these tags, it then suggested constituents for the Swedish sentences, assuming they were German sentences. After the annotation process, we converted the PoS-labels back to the Swedish labels. A small experiment, where the children were manually selected, shows that the German chunker suggests 89% correct phrase labels and 93% correct edge labels for Swedish. These experiments and the resulting time gain were reported in Volk and Samuelsson (2004). The final guidelines for Swedish parsing contained a number of adaptations, compared to the German guidelines, to account for linguistic differences between German and Swedish (see Section 3.2.3).

After finishing the monolingual treebanks in ANNOTATE, we converted the trees into TIGER-XML (Mengel and Lezius, 2000), an interface format which

[1] Talbanken has recently been cleaned and converted to a dependency treebank by Joakim Nivre and his group, see http://w3.msi.vxu.se/~nivre/research/talbanken.html. A version with constituent-like structures is also part of the Swedish treebank, see http://stp.ling.uu.se/~nivre/swedish_treebank/.

can be created and used by the treebank tool TIGERSearch.[1] This will be further described in Chapter 4.

3.2.2 Building Treebanks with Automatic Node Insertion

The TIGER annotation guidelines give a rather flat phrase structure tree.[2] For example, they have no unary phrases, no 'unnecessary' NPs (noun phrases) within PPs (prepositional phrases) and no finite VPs (verb phrases). This facilitates annotation for the human annotator by fewer annotation decisions, and a better overview of the trees, speeding up the annotation process.

The downside is that the trees do not have the same level of detail for similar phrases. For example, an NP that consists of only one child is not marked as an NP, while an added determiner results in an explicitly annotated NP. Similarly, an NP that is part of a PP is not marked, while the same NP outside a PP is annotated as an NP. These restrictions also have practical consequences: If certain phrases (e.g., NPs within PPs) are not explicitly marked, they can only indirectly be searched for in corpus studies.

In addition to the linguistic drawbacks of the flat syntax trees, they are also problematic for phrase alignment in a parallel treebank. We align subsentential units (such as phrases and clauses), and the alignment focuses on meaning, rather than sentence structure. Sentences can thus have alignment on a higher level of the tree (for instance if the S-node carries the same meaning in both languages), without necessarily having alignment on lower levels (for instance an NP without correspondence). With 'deeper' trees, we can annotate more fine-grained correspondences between languages. The more detailed the sentence structure is, the more expressive is the alignment. (Alignment is discussed in detail in Chapter 4.)

We first annotated the German and Swedish sentences semi-automatically, in the flat manner. The syntax trees were then automatically deepened through our program, which inserts phrase nodes that can be derived from the structure. We only insert unambiguous nodes, so that no errors are introduced. Examples of such an unambiguous node are an AP (adjective phrase) for a single adjective in an NP, and an NP in flat PPs. Figure 3.2 shows an example tree before and after the automatic insertion of an adjective phrase and a noun phrase.

The program contains two sets of rules; rules for the insertion of unary phrases and rules for handling complex phrases. The first set of rules inserts adjective phrases (APs), adverbial phrases (AVPs), noun phrases (NPs) and

[1] See also http://ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/doc/html/TigerXML.html.

[2] This section has been previously published in Samuelsson and Volk (2004).

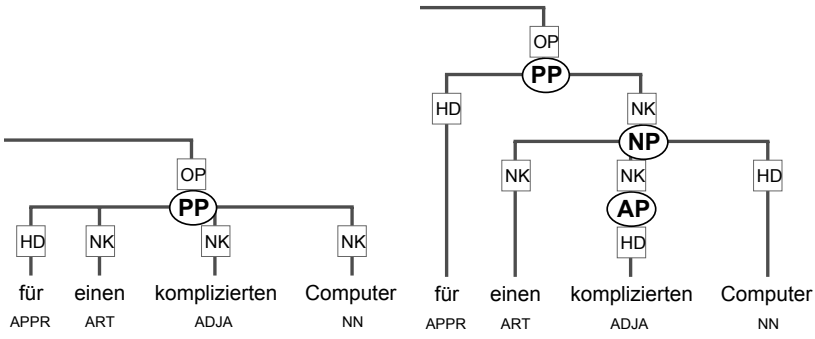


Figure 3.2: Part of a German tree, annotated in the flat manner (to the left) and after automatic node insertion (to the right).

verb phrases (VPs). The rules for unary branching NP nodes are listed in Table 3.1.

The first rule of Table 3.1 states that if we have an NP with an immediate AG (genitive attribute), APP (apposition) or GL (prenominal genitive) child, then this child is annotated as NP. For example, we insert an NP for *Sophies* (AG) in the NP *Sophies Mutter* ('Sophie's mother') and for *Sherekan* (APP) in the NP *die Katze Sherekan* ('the cat Sherekan').

The second rule states that if there is an S or CNP (coordinated noun phrase) with a direct PN (multi-word proper noun) child, then this child is annotated as NP. This means, e.g., that the PN-phrase *Claude Débussy* in the PP *von Claude Débussy* ('of/from Claude Débussy') gets an additional NP node.

The third rule states that an S, VP or CNP with a nominal (noun, pronoun or the like) child is annotated as NP. This ensures that, e.g., a subject or object like *Menschen* ('people'), *Sophie*, or *Sie* ('she') is attached to a noun phrase, which in turn is attached to the sentence or verb phrase.

The second set of rules, for complex phrases, binds several words together into one phrase. For example, there is a rule stating that if there is a PP (prepositional phrase), we insert an NP as a parent to all the children of the PP, except for the preposition. If, however, there is only an AP (e.g., *seit längerem*, 'for a long time'), AVP (e.g., *bis morgen*, 'until tomorrow') or CNP (e.g., *zwischen Zweigen und Blättern*, 'between branches and leaves') together with the preposition in the PP, we do not insert the NP. This binds the parts of an NP inside the PP together (like in Figure 3.2). Finally, the program checks that every NP

1	if NP with a direct genitive or apposition child, then insert an NP: (NP) →[AG|APP|GL]→ NE ⟹ (NP) →[AG|APP|GL]→ (NP) →[HD]→ NE
2	if S or CNP with a direct PN child, then insert an NP: (S|CNP) →[X]→ (PN) ⟹ (S|CNP) →[X]→ (NP) →[HD]→ (PN)
3	if S,VP or CNP with a direct noun or pronoun child, then insert an NP: (S|VP|CNP) →[X]→ (NN|NE|PPER|PDS|PRF|PPOSS|PIS|PRELS|PWS|TRUNC) ⟹ (S|VP|CNP) →[X]→ (NP) →[HD]→ (NN|NE|PPER|PDS|PRF|PPOSS|PIS|PRELS|PWS|TRUNC)

Table 3.1: Rules for insertion of unary branching NP nodes, of the format 'flat structure' ⟹ 'deep structure', where phrases are in parentheses, edge labels in brackets, and X stands for any label.

has a head.

These node insertion rules should all be reliable, because they are unambiguous, as long as the manual part of the annotation is correct according to the guidelines. However, there are some known problems. Cardinal numbers are not handled since they can be of different types, e.g., adjective-like in *25 Computer* ('24 computers') and noun-like in *im Jahre 2000* ('in the year 2000'). As they are ambiguous, they cannot easily be handled automatically. Adverbs in adjective phrases are still not handled (they should have their own AVP) and there are several STTS-tags that do not have their own phrase label, e.g., PTKNEG for the negation particle. Several of them could probably be made into adverbial phrases (AVPs).

We have not deepened the English structures, although they contain some nodes with flat annotation. For example, the Penn guidelines advocate flat structures within the core NP. This means that an NP consisting of a determiner, possibly one or more adjectives, and one or more nouns would profit from internal groupings, such as APs and noun groups.

The Swedish trees are automatically deepened in the same manner as the German trees. Since Swedish and German are similar languages, and have similar annotation schemes, there are only minor differences between the insertion programs. For example, pre-nominal genitives in German are always

assumed to be proper names (NE) but in Swedish they could also be regular nouns (NN), e.g., *vid världens ände* (literally 'at the world's end'). Additionally, a Swedish PP can consist of a preposition and a sentence or verb phrase, which is not possible in German. For example, a phrase like

(3.1) (göra) min av att svara

(make) as if to answer, literally 'make an expression of answering'

should contain the structure (PP av (VP att svara)). This means that we have to add the possibility of having an S or VP in the PP for Swedish.

Some problems with the node insertion for Swedish resulted from errors in the mapping from STTS-tags back into SUC-tags. For instance, the STTS-tag KOKOM (comparison particle, without sentence) is translated into PR (preposition). According to the German annotation guidelines the KOKOM is part of the NP, while a preposition should have a PP parent phrase. Since the translation of the STTS-tags back into SUC-tags is done after the node insertion, this creates erroneous tree structures. A check in the SUC corpus showed that the correct translation of KOKOM should have been KN (conjunction). One example of such an erroneous preposition is *mer (NP (PR än) en maskin)* ('more than a machine'). These errors can easily be found and should be changed in a future version of SMULTRON.

The successful automatic node insertion (see Samuelsson and Volk, 2004, for an evaluation) raised the question whether the flat annotation according to the TIGER guidelines could be made even flatter. In other words, is it possible and feasible to define a minimal set of human annotation decisions with a maximum number of automatically inserted nodes and labels? A minimal form for the manual annotation, where the rest of the nodes are automatically inserted later, could save a lot of time. The deepening of course still has to be totally safe, i.e., unambiguous.

The problem is defining a minimal form that is still linguistically plausible. If the form is too minimalistic and skeletal, it will be hard for the human annotator to still maintain an overview and to see what is correct in the annotation. The advantages of the flat annotation should not be renounced.

It is a complex task to determine what nodes are always unambiguous. There are mainly two difficulties. The first one is that a node has to be manually inserted (i.e., a manual decision has to be made) if any of its edges are ambiguous. This means for instance that we cannot automatically insert a missing top node S (sentence), since the distinction between subject and object is often ambiguous (at least for the computer).

The second difficulty in finding the unambiguous nodes lies in language differences. One example of this is the node VZ (*zu*-marked infinitive), which in German unambiguously groups the infinitive marker and the infinitive verb.

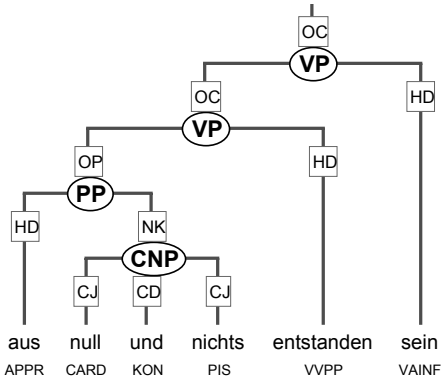


Figure 3.3: Example of a nested VP that could be automatically inserted.

The Swedish equivalent, marked with *att* (see Section 3.2.3, p. 52), is ambiguous, just like the English equivalent *to*.

One example of phrases that could be left out in the manual annotation for both German and Swedish (to be automatically inserted afterwards) is nested VPs in verb chains. According to the NEGRA guidelines we need nested VPs for every verb in a chain (see Figure 3.3). But it would facilitate the manual annotation if all verbs in such a chain could be entered into one VP (i.e., as sister nodes on the same level) in the flat tree structure, and the nesting could be done automatically afterwards. This nesting is unambiguous in both German and Swedish since the order of the verbs within the verb group is fixed (with few exceptions like the German *Oberfeldumstellung* (auxiliary inversion), which need to be handled manually).

3.2.3 Differences between the Annotation Schemes

As we will see in Chapter 4, the use of different annotation schemes for different languages is sometimes problematic for the alignment. However, we want the monolingual treebanks to be standalone, in addition to being used together in the parallel treebank, and therefore compatible with existing treebanks.

As mentioned, The Penn and TIGER formats both describe phrase structure, or constituent, trees. However, both formats also allow for some level of functional annotation, by having non-terminal node labels representing phrase categories and edge labels representing functions. While the TIGER annotation requires functional information on all edges, edge labels are sparse in the Penn

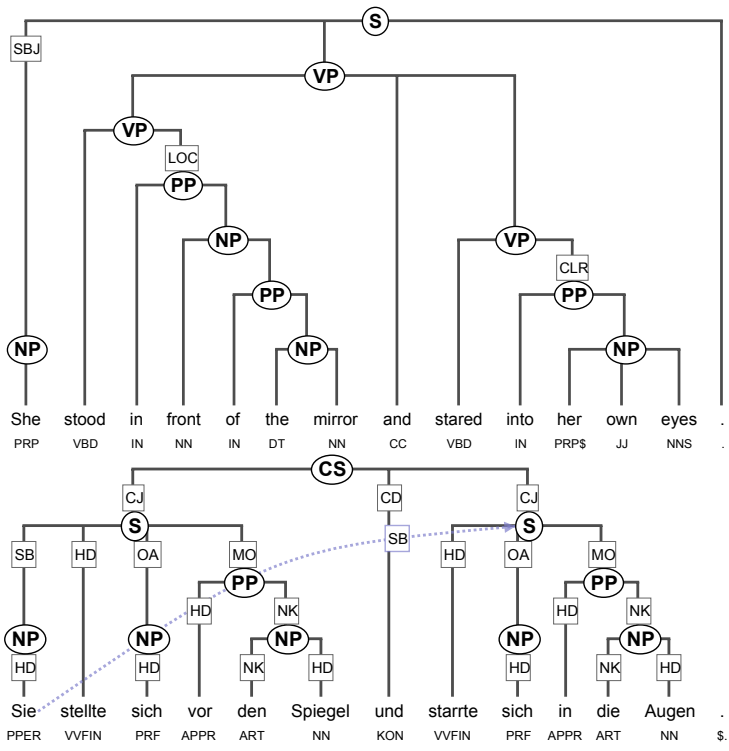


Figure 3.4: An English and a German sentence with coordination, which has a special phrase label in the German annotation.

annotation, only used for specific, possibly ambiguous functions like subject, predicate, and different types of adjuncts (temporal, local, manner, etc.).

One of the more important differences between the Penn and the TIGER formats is the treatment of discontinuity in a sentence. While the Penn treebank annotation introduces symbols for empty tokens (traces) and secondary edges for the representation of such phenomena, the TIGER annotation uses crossing edges for discontinuous units.

The number of different phrase categories is about the same for both annotation schemes (26 for TIGER and 25 for Penn), and while some of the categories, like VP and NP, are similar between the two annotation formats there are many differences in the interpretation of the different categories. TIGER has specific labels for coordinated phrases, while coordination is handled in the ordinary categories in the Penn annotation. One example of this can be

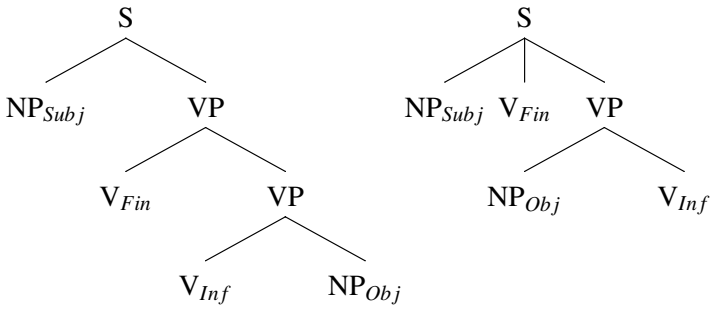


Figure 3.5: Structural differences at sentence level between the English Penn treebank (on the left) and the German TIGER treebank (on the right).

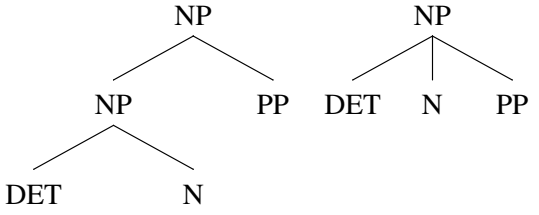


Figure 3.6: Structural differences for noun phrases containing prepositional phrases between the English Penn treebank (on the left) and the German TIGER treebank (on the right).

seen in Figure 3.4. The two coordinated VP phrases are attached to a VP in the annotation of the English sentence, while the two coordinated S phrases in the annotation of the German sentence are attached to a coordination CS phrase.

The TIGER sentence is shallower than the Penn sentence. One reason for this is the difference in handling verbs and verb phrases, shown in Figure 3.5. The Penn sentence (S) directly dominates the subject and a VP, which in turn contains the finite verb. The TIGER sentence, on the other hand, directly dominates the subject and the finite verb, reserving VPs for non-finite verbs.

As can be seen in Figure 3.6, the TIGER format is also shallower when it comes to noun phrases. While the TIGER NP dominates determiner, noun and modifying PP directly, the Penn NP first splits into an NP and a PP, to have the determiner and noun in an NP of its own.

Even though the German and Swedish treebanks were parsed using basically the same annotation scheme, the Swedish guidelines have been adapted to account for particular Swedish linguistic issues. Such adaptations require

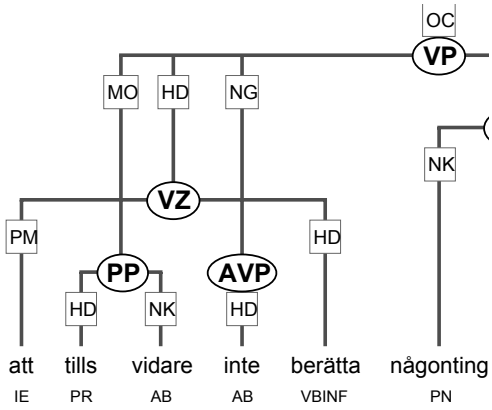


Figure 3.7: A Swedish VZ-phrase with multiple phrases between the non-finite verb *berätta* ('tell') and the infinitive marker *att* ('to').

careful investigation. One example is using the function labels DO (direct object) and IO (indirect object) instead of the German OA (accusative object) and DA (dative). German has four different cases, while Swedish only has two (nominative and genitive).

Some differences between Swedish and German do not affect the annotation scheme, just the annotation guidelines. As was mentioned on p. 48, Swedish prepositional phrases are more flexible regarding the object of the preposition, and allow for a non-finite verb phrase or a subordinate clause to follow the preposition.

In some cases, however, using the German guidelines as basis for the Swedish guidelines resulted in problems, where ignoring German particularities would have been a simpler solution. One such example is the VZ-phrase (zu-marked infinitive). In German the non-finite verb and the infinitive marker are always treated as one unit, which cannot be separated. Therefore, it makes sense to mark them as one phrase in the German sentence. In English they are more loosely connected, and a typical example of the split infinitive is the Star Trek opening line *to boldly go...* The same applies to Swedish. In Figure 3.7 we see a discontinuous VZ-phrase, with a prepositional phrase and an adverbial phrase wedged between the infinitive marker and the non-finite verb. Swedish infinitives are annotated this way in SMULTRON, but it would have been more suitable not to use VZ-phrases for Swedish. They could, however, easily be located and re-attached to the VP immediately dominating the VZ.

Even though the ANNOTATE tool mostly suggests the correct phrase and edge labels for Swedish, there are still a number of difficult cases for the annotator to decide. One example is the difference in Swedish between prepositions and verb particles. In spoken language, prepositions are not stressed, while verb particles are. In some cases the word is a preposition, not a verb particle, but it still is closer to the verb and therefore behaves 'strangely'. One example is the following relative clause.

(3.2) (hår) som varken gelé eller spray bet på

((hair) that neither gel nor spray would work on)

In this relative clause the pronoun *som* is put in the beginning and is thus separated from the stranded preposition *på*. The annotator has to establish that this is indeed a preposition and then decide whether the pronoun should be in the prepositional phrase (with crossing branches) or not.

3.3 Summary

In this chapter, we have described how the monolingual treebanks of the SMULTRON parallel treebank were created. The treebanks consist of texts with about 1,000 sentences per language, from two different genres, in English, German and Swedish. We started by exploring the word-level annotations, such as PoS-tags, morphological annotation, and lemmas. We then continued with phrase and sentence-level annotations, such as the syntactic structure.

The English part of SMULTRON is annotated with the widely used Penn tagset and Penn treebank annotation, and the German part with the STTS tagset and TIGER annotation guidelines. While the SUC tagset is considered a standard for Swedish (of which we used a slightly adapted version), there was no proven standard for Swedish syntax trees. Instead we used an adapted version of the German tree guidelines. This turned out to work well in some respects, but would need more adaptation in others. Overall, this choice of syntactic annotation schemes resulted in stand-alone monolingual treebanks, compatible with existing treebanks for each language.

We have pointed to some problematic areas for achieving high quality in treebanking, and discussed possible ways to automate part of the annotation process. A number of issues have surfaced, which are crucial to quality, such as the choice of data and annotation schemes, representation formats, automated processing, and quality checks. Some of these issues will be addressed again, especially quality checking, in Chapter 5. Next, however, we will investigate another level of annotation, the alignment, which turns the monolingual treebanks into a parallel treebank.

4. Alignment in SMULTRON

When working with translations in multiple languages, we would like to see what the relation between these translations is. As stated by Tiedemann (2003), the correspondences between the parts in source and target texts are one of the most important features of texts and their translations. Establishing translation correspondences is a difficult task, generally called alignment. It is usually performed on the paragraph level, sentence level and word level. From a statistical point of view, alignment shows which words (or strings of words) co-occur across parallel texts. From a linguistic point of view, it shows which part of a sentence in one language is equivalent in meaning to which part of a corresponding sentence in another language.

In this chapter, we describe the alignment of the SMULTRON parallel treebank. Again, such a description allows us to discuss what issues could arise when building a high-quality parallel treebank. Additionally, the alignment of SMULTRON will be used as a gold standard for the experiments on automatic alignment in Part II of the thesis.

As described in Chapter 3, we created the monolingual treebanks in Annotate, and then exported them in TIGER-XML format (Mengel and Lezius, 2000). TIGER-XML is a powerful database-oriented representation for graph structures, including syntax trees with labels for non-terminals, edge labels, multiple features on the word level and crossing edges.[1] It is typically used with constituent symbols on the non-terminal nodes and functional information on the edge labels, which enables a combination of constituent structure and dependency structure information. In a TIGER-XML graph each terminal node (token) and each non-terminal node (syntactic constituent) has a unique identifier (prefixed with the sentence number). Terminals are numbered from 1 to n and non-terminals from 500 to m (under the plausible assumption that sentences never have more than 499 tokens).

Figure 4.1 shows part of an example sentence in TIGER-XML, where node 500 in sentence 5 is of the category prepositional phrase (PP). The phrase consists of word number 4, which is the preposition (APPR) *über* ('over, about') with the edge label head (HD), and of node 503, which is marked as noun

[1] See http://ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/doc/html/TigerXML.html.

```
<s id="s5">
<graph root="s5_502">
<terminals>
  <t id="s5_1" word="Sie" pos="PPER" />
  <t id="s5_2" word="hatten" pos="VAFIN" />
  <t id="s5_3" word="sich" pos="PRF" />
  <t id="s5_4" word="über" pos="APPR" />
  <t id="s5_5" word="Roboter" pos="NN" />
  <t id="s5_6" word="unterhalten" pos="VVPP" />
  <t id="s5_7" word="." pos="$." />
</terminals>
<nonterminals>
  <nt id="s5_500" cat="PP">
    <edge label="HD" idref="s5_4" />
    <edge label="NK" idref="s5_503" />
  </nt>
  <nt id="s5_503" cat="NP">
    <edge label="HD" idref="s5_5" />
  </nt>
[...]
</nonterminals>
</graph>
</s>
```

Figure 4.1: An example of a parse tree represented by TIGER-XML.

phrase (NP) and has the function noun kernel (NK).

The unique node identifiers can be used for the phrase alignment across parallel trees, or more precisely across trees in corresponding translation units. We decided to also use an XML representation for storing the alignment (see Volk et al., 2006). The alignment file, as shown by the example in Figure 4.2, contains the names of the treebank files with identifiers assigned to them. By convention, we use the language code as treebank identifier. Each alignment is stored within an align tag, utilizing the treebank identifier and the node identifier to uniquely refer to a word or phrase. Figure 4.2 shows the alignment of the non-terminal node 506 in sentence 20 of the German treebank to the non-terminal node 503 of sentence 21 in the English treebank.

The alignment of SMULTRON was annotated manually between sentences, phrases and words over parallel trees. For the alignment we developed a tool called the Stockholm TreeAligner (Lundborg et al., 2007, Volk et al., 2006,

```
<treebanks>
 <treebank id="de" language="de_DE"
  filename="SMULTRON_DE_Sophies_World.xml"/>
 <treebank id="en" language="en_US"
  filename="SMULTRON_EN_Sophies_World.xml"/>
</treebanks>
<alignments>
 <align type="good">
  <node treebank_id="de" node_id="s20_506"/>
  <node treebank_id="en" node_id="s21_503"/>
 </align>
</alignments>
```

Figure 4.2: An example of alignment represented by TreeAligner-XML.

2007), a graphical user interface to insert (or correct) alignments between pairs of syntax trees, which displays two trees and allows the user to draw alignment lines by clicking on phrases and words. It can handle different types of alignment (e.g., exact and approximate), marked by different colours. Additionally, we can search for aligned structures in the TreeAligner, which has the expressive power of a treebank search tool like TIGERSearch. The TreeAligner is available for download with the source code.[1]

Our work focuses on word and phrase alignment. Phrase alignment encompasses the alignment from simple noun phrases and prepositional phrases all the way to complex clauses. We consider phrase alignment to be alignment between linguistically motivated phrases, in contrast to work in statistical machine translation, where phrase alignment is defined as the alignment between arbitrary word sequences (n-grams). Alignment can thus be regarded as an additional layer of information on top of the syntactic structure. We believe that such linguistically motivated phrase alignments provide useful phrase pairs for example-based machine translation, and provides interesting insights for translation science and cross-language comparisons. Phrase alignments are particularly useful for annotating correspondences of idiomatic or metaphoric language use.

Figure 4.3 shows an example with two trees. Alignment is represented as lines between words and phrases. We see that, e.g., the English word *enigmas* corresponds to the Swedish word *gåtor* and that the English NP dominating *A lot of age-old enigmas* corresponds to the Swedish NP *Många gamla gåtor*.

[1] See http://kitt.cl.uzh.ch/kitt/treealigner/.

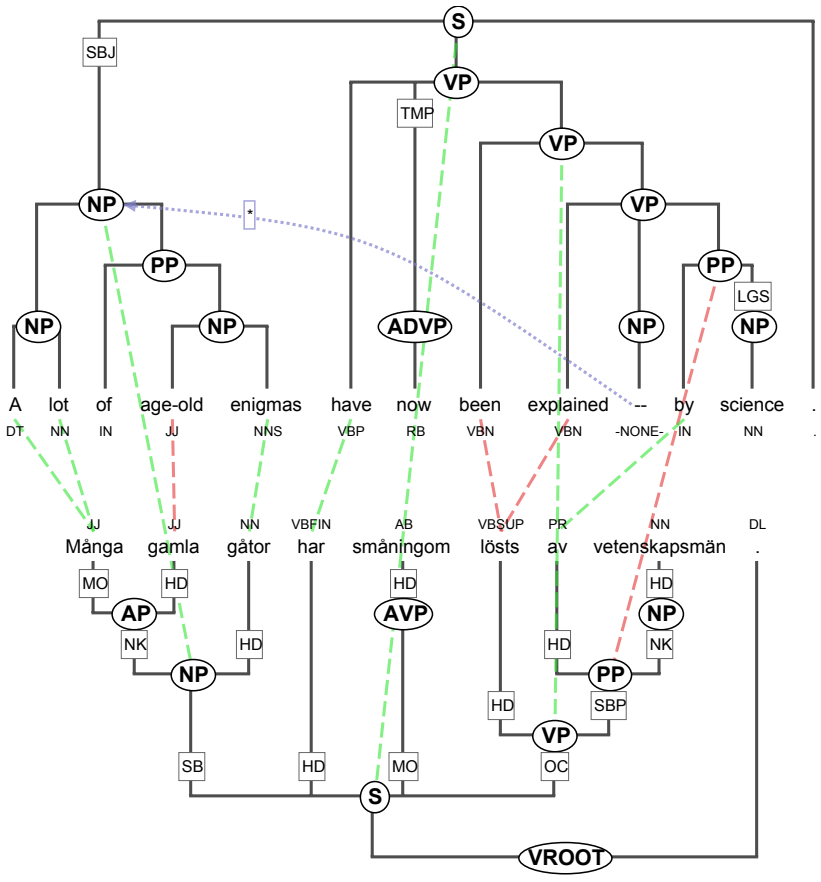


Figure 4.3: An English and a Swedish tree with word and phrase alignment.

4.1 Alignment Guidelines

Detailed annotation guidelines are crucial for annotation consistency. Several such guidelines are available for alignment, e.g., for the Blinker (Melamed, 1998a), ARCADE (Véronis, 1998), and PLUG (Merkel, 1999a) projects, as well as numerous other projects (Kruijff-Korbayová et al., 2006, Lambert et al., 2005, Macken, 2010). Most of these deal with word alignment.

The SMULTRON alignment guidelines describe word and phrase alignment between annotated syntax trees. They were developed by the author and have previously been discussed by Samuelsson and Volk (2006, 2007a,b), and Volk

et al. (2008). The guidelines consist of general principles and concrete rules.

The principle most central to the SMULTRON alignment is to align words and phrases that represent the same meaning. The goal of the alignment is to show translation equivalence. However, the alignment should not be used to show parallelism in the translations, i.e., that two phrases were probably triggered by the same phrase in the original. Words and phrases shall be aligned only if they, or the tokens that they span, are general enough to serve as translation units outside the current sentence context, e.g., in a machine translation system. The focal point is thus meaning rather than syntactic structure or grammatical forms.

For example, in the English part of our Sophie's World treebank the phrase *their astonishment at the world* appears, which we have decided not to align with the German phrase *die Verwunderung über das Leben* ('the astonishment at life'), although these two phrases were certainly triggered by the same phrase in the Norwegian original, and both have a similar function in the two corresponding sentences. These two phrases seen in isolation are too far apart in meaning to license their re-use. The same applies to cases where one language uses a pronoun, and the other a full noun.

The two other most important principles for alignment are more technical. One principle tells our annotators to align as many words and phrases as possible, ensuring that the alignment is comprehensive. The other principle is to align as close as possible to the tokens. In case of doubt, the alignment should go to the phrasal node that is closest to the terminals. For example, our German treebank guidelines require a multi-word proper noun to be grouped in a PN phrase, which is dominated by a noun phrase, *(NP (PN Sofie Amundsen))*, whereas the English guidelines only require the NP node, *(NP Sophie Amundsen)*. When we align the two names, principle 3 tells us to draw the alignment line between the German PN node and the English NP node since the German PN node is closer to the tokens than the German NP node.

Alignment showing more specific information is better than alignment showing more general information. In Example 4.1 this means that the English NP *nothing* should be aligned to the Swedish NP *ingenting* rather than the coordinated NP (CNP) *noll och ingenting*.

(4.1) **SV:** ha blivit till av (CNP noll och (NP ingenting))
(have become from zero and nothing)
EN: have come from (NP nothing)

Often, we are confronted with phrases that are not exact but approximate translation correspondences. Consider the English phrase *more than a piece of hardware* and the German *mehr als eine Maschine* ('more than a machine').

The pair does not represent the closest possible translation, but a possible 'second-best' translation in many contexts. We therefore differentiate between two types of alignment, displayed by different colours in our alignment tool. Phrases and words representing exactly the same meaning are aligned as exact translation correspondences, like in Example 4.2.

(4.2) **DE:** (NP den mänskliga hjärnan)

EN: (NP the human brain)

If they represent approximately the same meaning, they are aligned as fuzzy translation correspondences, like in Example 4.3, because of the pronoun *her*.

(4.3) **DE:** (PP auf dem Heimweg von der Schule)

(on the home-way from the school)

EN: (PP on her way home from school)

If an acronym is to be aligned with a spelled-out term, it is always an approximate alignment. For example, in the economy reports in SMULTRON, the English acronym *PT* stands for *Power Technology* and is aligned to the German *Energietechnik* as a fuzzy correspondence. Proper names are aligned as exact alignments, even if they are spelled differently across languages, e.g., *Sofie* vs. *Sophie*.

The alignment guidelines created for the SMULTRON project allow many-to-many alignments on sentence level. Thus, nodes from one sentence can be aligned to nodes from several sentences. At word and phrase level, however, we do not allow for many-to-many alignments, for simplicity and clarity reasons. Many-to-many alignments can mislead the annotator to align too many words. It is also problematic for visualization of the alignment in the TreeAligner, as too many alignment links tend to make the parallel trees messy and difficult to interpret. Most importantly, however, the alignment gets more complex. Wu (2010) states that because the complexity of word alignment rises for non-monotonic alignment (compared to alignment that preserves word order between the languages) it is usually assumed that word alignment cannot be many-to-many. One of the problems is that the matching process is very complex, when using the data in, e.g., an EBMT system. As an example, consider the (allowed) one-to-many alignment of Example 4.4.

(4.4) **SV:** fruktträden

EN: the fruit trees

Such correspondences make it difficult to only allow one-to-one alignments. However, based on such an example in, e.g., an EBMT system, finding *the*

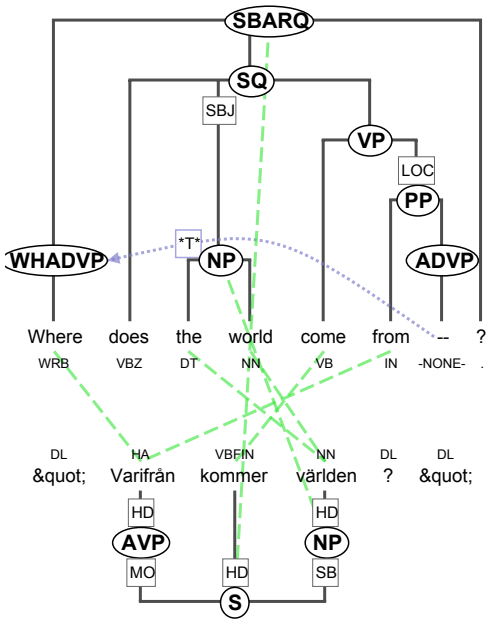


Figure 4.4: The one-to-many alignment of *Where from* and *Varifrån* is not implicitly present in the structural alignment.

would give us *fruktträden*. We then would have to check the rest of the words aligned to *fruktträden* and only use the match if *the* is followed by *fruit trees*. With many-to-many alignments, for example all words of the Swedish phrase *de stora fruktträden* aligned to all words of the English phrase *the large fruit trees*, we would need to check the context of both languages. Thus, we allow for one-to-many alignments, but assume that the equivalence of many-to-many alignments can generally be shown by alignments at a higher level in the tree.

Zhechev (2009) points out that, contrary to SMULTRON, their system only allows one-to-one links. One of the main reasons given is that there is no consensus in the field as to whether one-to-many links should be treated in a conjunctive or in a disjunctive manner, increasing the development complexity for systems using parallel treebanks, as they need to support both types of treatment. Additionally, Zhechev states that the information encoded by conjunctive one-to-many links is implicitly present in structural alignment.

While we assume that this applies to many-to-many alignment, we are aware that this is not always the case. If, for example, a phrase in one language contains additional information, it rules out the necessary alignment link above

the un-annotated many-to-many equivalence. This means that the equivalence is missing, leading to information loss.

There are also a number of problems with only allowing one-to-one alignments. For example, sentence correspondence is not always one-to-one, and this information will not be implicitly present in structural alignment. Additionally, the assumption will disregard the fact that languages treat some issues differently, and what is one word or phrase in one language may be split up into several non-consecutive words or phrases in another. One example of this can be seen in Figure 4.4. While *the world* and *världen* would not require one-to-many alignment, as this information is present in the alignment of the NPs, the alignment of *Where from* and *Varifrån* could not be shown by one-to-one alignment higher up in the tree.

The one-to-many alignment option is not used if some part of a sentence is repeated. Consider, e.g., the repeated subject in a coordinated sentence like in Example 4.5, where the German pronoun *sie* is realized as *they* twice in the English sentence.

(4.5) **DE:** (S Beim Supermarkt hatten (NP sie) sich getrennt)
(At the supermarket had they themselves parted)
EN: (S When (NP they) got to the supermarket (NP they) went their separate ways .)

If a node in one language is aligned to several nodes in the other language, these links are all of the same type, either exact or fuzzy.

Alignments are different in nature from most other annotation, as the annotation does not introduce abstract categories such as, e.g., PoS. It relies upon defining translation units with equivalent meanings, which is not a trivial task. Alignment is thus in some ways as difficult as translation itself. Problematic cases, some due to the translator's freedom, can be found on all levels (word, phrase, clause and sentence). In these cases it is important to remember the main goal, that the aligned phrases should be equivalent in meaning outside of the sentence context.

Many open questions regarding the distinction between exact and fuzzy alignment persist. Is *at this time of the year* an exact or a fuzzy translation correspondence of *zu dieser Jahreszeit*, where a literal translation would be *in this season*? Examples like this make us wonder how useful the distinction between exact and approximate translation correspondence really is.

Sometimes the distinction between fuzzy alignment and no alignment is also problematic. Fuzzy alignment is used when there is additional information in the phrase in one language, while the general content is still the same. However, if vital information is missing in one language, there should be no

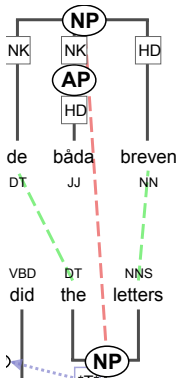


Figure 4.5: An additional adjective in an NP is acceptable for fuzzy alignments.

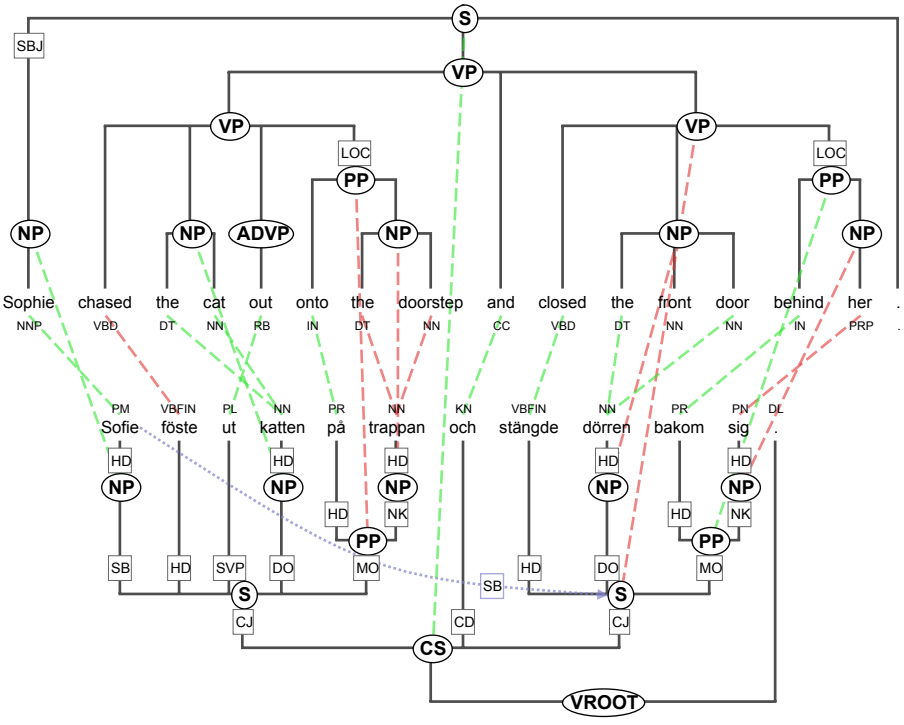


Figure 4.6: Structural differences between the annotation schemes leave the leftmost English VP and Swedish S in the coordinations unaligned.

alignment between the phrases. One example of this distinction is seen in Figure 4.5. An additional adjective is not considered vital information and thus we have fuzzy alignment between the NPs.

Figure 4.6, in contrast, shows two sentences, the Swedish containing two coordinated sentences (one without subject) and the English containing two coordinated verb phrases. The Swedish coordinated sentence is aligned to the English S, and the second Swedish S is aligned to the second English inner VP. The first Swedish S however cannot be aligned to the first English VP, since the Swedish phrase contains the subject. This problem is frequent in our treebank, due to differences in the annotation schemes between the languages. While we have chosen different annotation schemes for the different languages, to make them compatible with existing treebanks, using a common annotation scheme for all languages (see, e.g. Dyvik et al., 2009) avoids such problems.

Prepositions are another problematic area, as they vary from being semantically well defined to semantically vague or empty. We had decided that typical correspondences between prepositions are to be marked as exact while others are approximate. The prepositions in the English phrase *in the mailbox* and the German phrase *in dem Briefkasten* are clearly a case of exact equivalence. But what of the English phrase *a bowl of cat food* and German *eine Schale mit Katzenfutter* (literally 'a bowl with cat food')? To facilitate for the annotators, we decided that if the NPs are exact correspondences, then the prepositions are considered to be as well.

An additional complication is that some prepositions and articles in German can be contracted, e.g., *an dem* becomes *am*, *in das* becomes *ins*. If the contraction equals a preposition and article in the other language, these should be aligned as exact equivalence, e.g., *at the end* and *am Ende*. However, when aligning with Swedish, where the definite article is a suffix, this is problematic. For, e.g., German *beim Supermarkt* and Swedish *vid storköpet* (at the supermarket) we have decided to draw fuzzy alignment between the prepositions.

4.2 Inter-annotator Agreement

An issue to explore in our quest for high-quality parallel treebanks is how reliable the manual annotation is. How comprehensive are the annotation guidelines and are there questions not addressed? Here we will discuss these questions regarding the alignment. The first question can partly be answered through quality checks of the alignment (see Section 5.4). In an attempt to answer both questions, however, and to improve the annotation guidelines, we carried out two evaluations of inter-annotator agreement. In the first evaluation three annotators independently aligned 100 sentence pairs, while a number of undergraduate students aligned 20 sentence pairs in the second evaluation.

Evaluation of inter-annotator agreement is not necessarily a measure of quality for the annotation of the corpus. Buch-Kromann (2010) discusses several ways of adapting the annotation schemes, which would increase the agreement but not the quality. Simplifying the annotation, at the cost of information and linguistic plausibility, clearly does not improve the soundness of the parallel treebank, which we consider a vital part of quality (Section 2.3). However, the assessment of inter-annotator agreement for a corpus project is an important and helpful tool, to guide the development of the annotation schemes.

Several other researchers have previously reported on inter-annotator agreement experiments for alignment. Unfortunately, the metrics used have differed across evaluations and thus the results are not comparable. In Section 4.2.1 we will have a look at several previous evaluations. There we also define the metrics that will be used in Sections 4.2.2 and 4.2.3, where we report on the two evaluations carried out for the SMULTRON data.

4.2.1 Agreement Evaluations and Metrics

First, let us explore how to evaluate inter-annotator agreement (IAA). This is also relevant to evaluation of alignment, which will be discussed in Section 6.1.4. A number of previous evaluations of inter-annotator agreement have used precision, recall and F-score, initially used in information retrieval, to compare the annotations. Precision is the fraction of found instances that are correct, while recall is the fraction of correct instances that are found. The scores are calculated according to the following definitions.

Precision $\frac{\text{found correct items}}{\text{found items}}$

Recall $\frac{\text{found correct items}}{\text{correct items}}$

The F-score, also called F_1-score or balanced F-score, combines precision and recall, giving them equal weight. The F_2-score weights recall higher than precision, while the $F_{0.5}$-score weights precision higher than recall. (The β-value in the following equation is set to 1, 2, or 0.5, accordingly.)

F_β-score $(1+\beta^2) \times \frac{\text{precision} \times \text{recall}}{\beta^2 \times \text{precision} + \text{recall}}$

For example, Dorr et al. (2002) explore translation divergences, structural differences between languages, and transform the sentence structure in one language into a structure more similar to the other language, to help in the alignment task. In two experiments, one for English-Spanish and one for English-Arabic, the annotations of four annotators were evaluated pairwise, with an

average f-score of 0.79 for the untransformed English-Spanish and 0.70 for the untransformed English-Arabic. Li et al. (2010) had two annotators carry out Chinese-English word alignment, with a precision between 0.90 and 0.95 and recall between 0.91 and 0.96.

A similar but complementary metric is used by Bojar and Prokopová (2006). Two annotators carried out Czech-English word alignment, and then the mismatch was calculated, alignments present in only one of the annotations. The annotations not agreed upon by the two annotators were divided by the total number of annotations by the annotators. The inter-annotator mismatch was 18%, and 9% when ignoring the alignment type.

Several similar metrics have been used, which we can call alignment error rate (AER) variants. AER was first described by Och and Ney (2000a), to measure the quality of an alignment. The manual alignment has sure and possible alignments and the AER is calculated as follows, where A is the alignment created by the system, S is the set of sure manual (gold standard) alignments, and P is the set of possible manual alignments.

AER $1 - \frac{A \cap S + A \cap P}{A + S}$

This metric was also used by Lambert et al. (2005), who created guidelines for a word alignment evaluation scheme, and had three annotators manually align a Spanish-English reference corpus. The AER metric was used for comparing automatically computed alignments to the manually aligned corpus, but also for evaluation of inter-annotator agreement. The alignments of each annotator were used as reference to calculate the AER of the other annotators, whose alignments were considered as computed alignments. The scores were between 8 and 14 (seemingly calculated as percent).

Kruijff-Korbayová et al. (2006) created guidelines for Czech-English word alignment and evaluated annotations based on these guidelines. They had three sets of annotations for the same data, where two sets were annotated by the same annotator months apart, and the third set was annotated by a second annotator. For the evaluation, they calculated an agreement measure in the following way, where A1 and A2 are the two sets of annotations to compare.

AGR $2 \times \frac{A1 \cap A2}{A1 + A2}$

The intra-annotator agreement was about 95%, and the interannotator agreement about 93%.

The same metric has been used by, e.g., Graca et al. (2008) and Mareček (2009). Graca et al. (2008) had four annotators manually align data for six language pairs (between Portuguese, English, French and Spanish), with two annotations of each language pair. Inter-annotator agreement was calculated

only for word pairs aligned by at least one of the annotators, "[s]ince this is a structured problem one cannot measure inter-annotator agreement by counting how many times the annotators agree for each possible decision link, since this value would be highly optimistic and overwhelmed by agreement on unaligned links". Thus, following Kruijff-Korbayová et al. (2006), the agreement for an intermediary evaluation was calculated as 84%, and in a final evaluation 92%. Mareček (2009) discussed the differences between traditional word alignment and deep surface alignment. For training and evaluation purposes, he created a corpus of 2,500 sentences, which was aligned by two different annotators. Distinguishing between three possible types of alignments, the agreement was 83%. Disregarding the types, only distinguishing between aligned/unaligned word pairs, the agreement was 90%.

Davis (2002) defines a Word Alignment Agreement (WAA) score for evaluation of automatic alignment, where every alignment link receives a weight based on the number of words in the source and target sentence. This metric is also used for IAA by, e.g., Macken (2007, 2010). Macken created an English-Dutch word alignment gold standard. For the inter-annotator agreement experiment, three annotators aligned part of the corpus and each of these three annotations were compared to the author's alignments. Phrase-to-phrase alignments are converted into word-to-word alignments and then the weight of each word alignment is calculated as the number of source words and the number of target words divided by 2. The WAA score is calculated as the agreed weight divided by the total weight. The agreement was between 0.84 and 0.94.

Since the alignment task is more comparable to semantic annotation than to PoS-tagging or parsing, we may also look at other tasks in semantic corpus annotation, e.g., the figures for inter-annotator agreement in frame-semantic annotation as found in the German SALSA project (Burchardt et al., 2006). They report the percentage of labelled exact match, where agreement is 85% on frames and 86% on roles when comparing two annotations.

As is clear from this sample of inter-annotator agreement evaluations, there is no consensus regarding what evaluation metric to use. Carletta (1996) advocated using the Kappa statistic for evaluating agreement on classification tasks in computational linguistics, which has since emerged as the metric commonly used. Unfortunately, there are a number of different ways of calculating these chance-corrected metrics and due to term confusion (see, e.g., Artstein and Poesio, 2008, giving an excellent overview) it is not even certain that Kappa values are really comparable across data sets.

Cohen's Kappa is popular (used, e.g., by Kruijff-Korbayová et al., 2006, Macken, 2010), but, as discussed in Artstein and Poesio (2008) is limited in that all disagreements are treated equally. Burchardt et al. (2006) also reject Kappa, since it requires that a single label is chosen from a globally fixed

	Total	W-to-w		P-to-p		W-to-p		Exact	
A1	1,196	690	(58%)	495	(41%)	11	(1%)	949	(79%)
A2	1,555	952	(61%)	600	(39%)	3	(<1%)	1,071	(69%)
A3	1,553	943	(61%)	609	(39%)	1	(<1%)	1,134	(73%)

Table 4.1: Number and type of alignment link (percentage of the total number of alignments by each annotator) for experiment A.

pool for each annotated instance, which does not apply for their annotation of semantic frames. Additionally, as mentioned by Graca et al. (2008), it is not entirely clear how to handle unaligned items. Adding all unaligned word pairs means overwhelming agreement on unaligned links, but not adding them means not having a fixed number of instances to evaluate.

For the evaluations of agreement in Sections 4.2.2 and 4.2.3 we will use a precision-like evaluation as well as Krippendorff's α, following the recommendations of Artstein and Poesio (2008). This metric can handle multiple coders, which we have in our experiments. Additionally, it allows for different magnitudes of disagreement, meaning that we can treat, e.g., a difference between exact alignment and no alignment as worse than a difference between fuzzy alignment and no alignment. One of the problems, however, is the difficulty of interpreting the results.

4.2.2 Evaluating the Alignment Task

In the first evaluation, here called experiment A and originally published in Samuelsson and Volk (2006), three people (the original annotator and the two authors of the paper) independently of each other aligned one hundred sentence pairs of the English-Swedish Sophie treebanks. This experiment was conducted early in the creation of SMULTRON, to evaluate and compare the agreement, or consistency, between the annotators. From this evaluation, we estimated the general level of difficulty of the alignment task.

For experiment A, the number and type of alignments by each annotator is shown in Table 4.1. Here, w-to-w means word alignment, p-to-p means phrase alignment, while w-to-p means alignments that connect a word in one language to a phrase in the other language. While the latter is possible, we prefer to align words to words and phrases to phrases, thus separating the alignment of the textual layer from the alignment of an annotation layer.

Annotators 2 and 3 have approximately the same amount of alignment

	A
Union	4,304
% of possible	0.04%
Intersection	1,045
% of union	58%
% word align	59%
α (2 categories)	0.83
Type agreement	882
% of intersect	84%
% word align	62%
α (3 categories)	0.80
Type disagreement	163
% word align	44%

Table 4.2: Annotator agreement for the annotators of experiment A.

links while annotator 1 has been more restrictive, both on the word and phrase levels. For all three annotators, about 60% of the links are word alignments, and about 70-80% of the links are exact links (in contrast to fuzzy links).

In Table 4.2 we see the agreement when comparing the annotations. The intersection is the number of alignments where all annotators agree that these two words or phrases should be aligned. This set can be divided into the alignments where all annotators agree upon the alignment type (full agreement), and the alignments where they do not agree upon the type (partial agreement). The possible alignments are calculated as all alignments possible through the whole corpus, i.e., all words and phrases in one language times all words and phrases in the other language. This means that we could possibly align words of the first sentence in the corpus with words of the last sentence, which is unlikely. Thus the number of unaligned items agreed on is extremely high. However, calculating α by restricting the number of possible alignments by only counting words and phrases in aligned sentences does not make much of a difference, with only a minor drop in α. The α-score for two categories means that we only look at the distinction between aligned and unaligned items, with a distance of one between the categories. The α-score for tree categories means that we distinguish between unaligned, exact aligned and fuzzy aligned items, with a distance of 0.5 between unaligned and fuzzy aligned, and between fuzzy aligned and exact aligned, and a distance of 1 between unaligned and exact

aligned items.

The α-score for agreement on aligning words and phrases for experiment A seems quite high, at close to 80%, while the agreement calculated as the intersection is only 58% of the union. This indicates that the α-score is overly positive, due to the large number of unaligned items agreed upon. However, it is interesting that the figures for full agreement (the intersection with type agreement) are so high, 84% of the intersection. This is also visible from α, which is only slightly lower for three categories than for two. When there is agreement on which words/phrases to align, there is often no problem deciding whether it should be exact or fuzzy alignment. There seems to be a core of alignments that are obvious and uncontroversial.

Let us look at the distribution of phrases and words in the alignments agreed upon by all annotators. Overall, there are more word alignments in the intersection than phrase alignments. The ratio between word and phrase alignments roughly follows the ratio when looking at the individual alignments in Table 4.1 (around 60% word alignments). However, we see that the annotators agree more on the type for word alignments, and disagree more on the type for phrase alignments. This is not surprising, considering that a phrase often contains more information than one word. Therefore, it is more difficult to set the boundary for when something is an exact translation rather than an approximate translation.

In addition to looking at the agreement in the annotation, it is interesting to investigate the alignments found in only one of the annotations. Annotator A1 had only a few annotations that none of the others agreed upon. Considering that this annotator had fewer alignments, this indicates that A1 has chosen to do safer alignments, leaving more difficult cases unaligned. Most alignments only done by annotator A1 are errors, like aligning the Swedish NP containing *Kaptenssvängen* to the English NP containing *Captain's*, when it should be the larger English NP *Captain's Bend*. Additionally, A1 has the largest number of word-to-phrase alignments, for example the Swedish word *Sofie* aligned to the English noun phrase (NP) containing the word *Sophie*, which should be word-to-word alignment. Thus, the guidelines need to stress that word-to-phrase alignments should be avoided.

Most of the alignments only annotated by annotator A2 show that this annotator has aligned parts that are similar rather than equivalent in the translations. The annotator has word aligned the adjectives in *very distant* and *ganska avlägsen* ('rather distant'), and the verbs in the phrases *asked the question* and *ställt frågan* (literally 'posed the question'). Thus, the guidelines need to emphasize the main goal of the alignment, that the aligned parts should be equivalent outside the current sentence context.

Annotator A3 has several times aligned a sequence of words in one lan-

guage to a sequence in the other language by many-to-many alignment, e.g., between *was like* and *påminde om* ('reminded of'). As discussed on p. 60, we only allow one-to-many alignment on word and phrase level.

This first evaluation of inter-annotator agreement aided us in the early stages of alignment annotation. It showed us which points were problematic, issues that one annotator might not even be aware of as problematic, and helped us to agree on the guiding principles. This information was utilized to develop our alignment guidelines, which were then applied (and in some instances changed and expanded) during the whole annotation process. Next, we will look at inter-annotator agreement to evaluate the alignment guidelines.

4.2.3 Evaluating the Alignment Guidelines

In a second evaluation, experiment B, twelve advanced undergraduate students in Switzerland were given twenty sentence pairs. This was originally published as Volk et al. (2008). By comparing their alignments to our gold standard and to each other, we gained valuable insights into the difficulty of the alignment task and the quality of our guidelines. The sentences were taken from the German-English treebank, half from the Sophie part, half from the economy part. We made sure that there was one 1:2 sentence alignment in the sample. The students did not have access to the gold standard alignment. We demonstrated the alignment tool and introduced the general alignment principles to the students. The students then received a copy of the alignment guidelines, and were asked to do the alignments independently of each other and to the best of their knowledge according to the guidelines.

The student alignments varied much in numbers of alignments. In the Sophie part they ranged from 125 alignments to a mere 47 alignments (exact alignments and fuzzy alignments taken together). In the economy part the variation was between 259 and 62 alignments. The student with the lowest numbers turned out to work as a translator and used a very strict criterion of translation equivalence rather than translation correspondence. Three other students at the end of the list were not native speakers of either German or English. We therefore excluded these 4 students from the comparison.

For experiment B, the number and type of alignments by each of the remaining annotators can be seen in Tables 4.3 and 4.4. As mentioned, the variation in number of alignments is large, and even larger for the Sophie part than the economy part. The Sophie part generally has fewer alignments than the economy part. Between 60 and 80% of the links in the Sophie part are word alignments, compared to between 65 and 85% in the economy part. This is thus slightly higher than the 60% of experiment A. 55 to 80% of the links in the Sophie part are of the exact type, meaning that the upper bound is closer

	Total	W-to-w		P-to-p		W-to-p		Exact	
B1	120	71	(59%)	49	(41%)	0	(0%)	76	(63%)
B2	119	81	(68%)	32	(27%)	6	(5%)	80	(67%)
B3	89	59	(66%)	30	(34%)	0	(0%)	67	(75%)
B4	98	77	(79%)	21	(21%)	0	(0%)	80	(82%)
B5	107	68	(64%)	39	(36%)	0	(0%)	71	(66%)
B6	120	80	(67%)	40	(33%)	0	(0%)	82	(68%)
B7	68	51	(75%)	17	(25%)	0	(0%)	37	(54%)
B8	125	81	(65%)	44	(35%)	0	(0%)	95	(76%)
Gold	136	78	(57%)	58	(43%)	0	(0%)	121	(89%)

Table 4.3: Number and type of alignment link (percentage of the total number of alignments by each annotator) for the Sophie part of experiment B.

	Total	W-to-w		P-to-p		W-to-p		Exact	
B1	211	136	(64%)	75	(36%)	0	(0%)	126	(60%)
B2	253	172	(68%)	71	(28%)	10	(4%)	225	(89%)
B3	209	152	(73%)	57	(27%)	0	(0%)	125	(60%)
B4	189	144	(76%)	45	(24%)	0	(0%)	146	(77%)
B5	219	164	(75%)	55	(25%)	0	(0%)	193	(88%)
B6	182	152	(84%)	30	(16%)	0	(0%)	141	(77%)
B7	191	143	(75%)	48	(25%)	0	(0%)	142	(74%)
B8	259	183	(71%)	76	(29%)	0	(0%)	187	(72%)
Gold	249	178	(71%)	71	(29%)	0	(0%)	221	(89%)

Table 4.4: Number and type of alignment links (percentage of the total number of alignments by each annotator) for the economy part of experiment B.

	B Sophie	B Economy
Union	846	1,713
% of possible	1.78%	0.98%
Intersection	36	93
% of union	4%	5%
% word align	94%	96%
α (2 categories)	0.71	0.77
Type agreement	24	55
% of intersect	67%	59%
% word align	96%	96%
α (3 categories)	0.66	0.74
Type disagreement	12	38
% word	92%	95%

Table 4.5: Annotator agreement for the annotators of experiment B.

to the rates for experiment A. For the economy part, there are between 60 and 90% exact alignments.

In Table 4.5 we see the agreement when comparing the annotations. Again, the intersection is the number of alignments where all annotators agree that these two words or phrases should be aligned. This set can be divided into the alignments where all annotators agree upon the alignment type, and the alignments where they do not agree upon the type. The α-score is calculated the same way as for experiment A.

The intersection compared to the union was low for experiment A (Table 4.2), but is much lower for experiment B, only 4 and 5%. This is reflected (though not as dramatically) in the α-score, especially for the Sophie part of experiment B. Thus, we concluded that the alignment guidelines needed refinement. However, it should be pointed out that the annotators of experiment A were experts, while the annotators of experiment B were inexperienced.

While the full agreement (type agreement) was a large part of the alignments agreed upon for experiment A (around 90%), these figures are only around 60-65% for experiment B. This ratio is lower for the economy part than the Sophie part, while the drop in α between two and three categories is slightly larger for the Sophie part. The results, despite the inconsistency, suggest that the students of experiment B disagree about the alignments more, compared to experiment A, and also that when they agree, they disagree more

about the type.

The alignments agreed upon by all annotators almost entirely consist of word alignments. The mere 4 and 6% (for the Sophie and economy parts, respectively) phrase alignments do not mirror the amount of phrase alignments made by the annotators (21-41% for the Sophie part, 16-36% for the economy part, see Tables 4.3 and 4.4). As stated earlier, phrase alignments are more difficult, because there is often more information in a phrase than in a single word, and more possible variation between the languages. One way of helping the annotators is adding examples for phrase alignments in the guidelines.

Comparing the eight students' annotations to the gold standard, there was between 48 and 81% overlap with the gold standard for the Sophie part, and between 66 and 89% overlap with the gold standard for the economy part. This can be regarded as their recall values, if we assume that the gold standard represents the correct alignments. Additionally, the students had between 2 and 22 own alignments in the Sophie part and between 12 and 55 own alignments in the economy part. When adding the gold annotations to the students' annotations, the α-score increases slightly. The question is what kind of alignments the students have missed, and which additional own alignments they have suggested (alignments that are not in the gold standard).

Looking at the intersection of the students' alignments (ignoring the difference between exact and fuzzy alignments), about 50% of the alignments done by the student with the smallest number of alignments is shared by the other students. All of the alignments in the intersection of the students' annotations are in our gold standard file. This, again, indicates that there is a core of alignments that are obvious and uncontroversial. Most of them are word alignments.

Comparing the union of the students' alignments to the gold standard pointed to some weaknesses of the guidelines. For example, one alignment in the gold standard that was missed by all students concerns the alignment of a German pronoun, *wenn **sie** die Hand ausstreckte* ('when she extended her hand') to an empty token in English (*herself __ shaking hands*). Our guidelines recommend to align such cases as fuzzy alignments, but of course it is difficult to determine that the empty token really corresponds to the German word.

Other discrepancies concern differences in grammatical form, e.g., a German definite singular noun phrase (*die Hand*) aligned to an English plural noun phrase (*Hands*) in the gold standard but missed by all students. Finally, there are a few cases where obvious noun phrase correspondences were simply overlooked by all students (*sich - herself*) although the tokens themselves were aligned. Such cases should be handled by an automated process in the alignment tool, and will be discussed further in Section 5.4.

As in experiment A, some students had aligned words to phrases. For

example, one student had aligned the word *natürlich* to the phrase *of course* instead of to the word sequence *of course*. Another discrepancy occurred when the students aligned a German verb group with a single verb form in English (e.g., *ist zurückzuführen* vs. *reflecting*). We have decided to only align the full verb to the full verb (independently of the inflection). This means that we align only *zurückzuführen* to *reflecting* in this example.

Uncertainty on how to deal with different grammatical forms led to the most discrepancies. Should we align the definite NP *die Umsätze* with the indefinite NP *revenues*, since it is much more common to drop the article in an English plural NP than in German? Generally, at word level, only the nouns are aligned, while at phrase level the NPs are aligned. Should we align a German genitive NP with an of-PP in English (*der beiden Divisionen* vs. *of the two divisions*)? A search through the parallel German-English treebanks of SMULTRON for a German NP dominating an NP with the edge label AG (genitive attribute) and an English PP dominating an NP shows that this has not been consistently handled. There are 82 cases where the NP has been aligned to the PP and 45 cases where the NP has been aligned to the NP in the economy part, and 3 NP-PP cases and 19 NP-NP cases in the Sophie part. This issue definitely needs to be resolved and discussed in the guidelines.

In an alignment project like this, with the large number of phrases and words, annotators easily miss alignments that should be annotated. This could be solved by forcing the annotator to mark phrases without correspondence as unaligned. Such an approach was proposed, e.g., in the Blinker project (Melamed, 1998b). However, the vast amount of extra time it would take to mark everything without correspondence, both on word and on phrase level, made this approach unmanageable.

4.3 Summary

In this chapter, we have described the alignment of the SMULTRON parallel treebank, which shows translation equivalence by linking texts in two languages. A number of issues, which are important for alignment quality, have been discussed, such as what units to align, how to handle correspondences between multiple words, and different types or grades of alignment.

We have discussed some of the difficulties in creating cross-language word and phrase alignments. Two experiments evaluating inter-annotator agreement were conducted during the annotation process, which have given us insights regarding problematic areas for alignment and helped to identify weaknesses, issues that need to be resolved in our gold standard alignment and points that need further description in our alignment guidelines. These evaluations showed that alignment is not an easy task, but that when there was agreement

about aligning a pair of words or phrases, there was often also agreement about the type of alignment, exact or approximate. Additionally, there was more agreement on word than phrase alignment, indicating that word alignment is easier.

In the last two chapters we have described the creation of the SMULTRON parallel treebank. We have tried to detail the considerations and decisions during the creation of the high-quality annotation in a parallel treebank, as well as discuss problematic or inconsistent areas in the annotation. The parallel treebank thus outlined will be used as the gold standard in the experiments described in Part II of this thesis. There, we will explore automatic ways of expanding and improving a parallel treebank. First, however, we will discuss automatic methods for quality checking of parallel treebanks.

5. Quality Checking of Parallel Treebanks

Consistency and well-formedness (see Section 2.3) are important characteristics of corpus annotation, essential in creating a high-quality treebank. Previous research has shown that the presence of errors in linguistic annotation creates various problems, for both computational and theoretical linguistic uses of corpora. The problems range from unreliable training and evaluation of natural language processing technology (e.g., Hogan, 2007, Padro and Marquez, 1998), to low precision and recall of queries for already rare linguistic phenomena (e.g., Meurers and Müller, 2008). Even a small number of errors can have a significant impact on the uses of linguistic annotation. For example, Habash et al. (2007) compare Arabic parsing systems and find that the one that initially had better results was actually worse when treebank errors were accounted for. Considering that linguistic annotation such as PoS or syntactic annotation is often used as input for further layers of annotation or processing, even a small set of errors can significantly affect a higher layer. For example, the errors in syntactic rules identified in Dickinson and Meurers (2005b) are often attributable to erroneous PoS annotation.

Both automatic systems and human annotators produce errors. Multiple annotators may make different decisions. Even one and the same annotator can make different decisions, due to varying difficulty of the annotation. All annotation variation does not come from errors in the annotation, some pointing to ambiguities, and some stemming from annotator bias for the more difficult annotation cases (see, e.g., Beigman and Beigman Klebanov, 2009). One solution is to remove these instances, for example, potentially unfavourable sentence pairs when training a statistical MT system, to avoid incorrect word alignments (Okita, 2009). However, this removes all relevant data from those sentences and does not help alignment evaluation.

Thus, when creating a parallel treebank, quality checks need to be carried out along the way. Each step in the creation process needs to be inspected before the next step is carried out. An all manual check is laborious and time-consuming and it is easy to get blind to errors. If the treebank is large, it is almost an unmanageable task to check every sentence. What we need then is some kind of help, telling us where there seems to be a problem or a potential

error in the treebank. This lets the human annotation checker focus on the problem areas.

Before we can detect errors, we need to discuss what types of errors there are, since we can assume that different types of errors have different properties. Blaheta (2002) defines an error taxonomy consisting of three types. Detectable errors can be automatically detected and corrected, generally a result of the annotator choosing a different (but still reasonable) way of annotating a certain phenomenon, compared to the annotation guidelines. Fixable errors are usually harder to detect, since they are not fully systematic. These errors can occur, e.g., when the annotator misunderstands the guidelines. When found, however, they are easily corrected, at least by a human annotator. Systematic inconsistencies, finally, are problems not covered by the guidelines or not described precise enough. These are said to be hard to detect and to correct. Ule and Simov (2004) add violations of the annotation guidelines and violations of language principles not covered by the annotation guidelines, stating that this classification is orthogonal to, but not independent of, the classification proposed by Blaheta.

These classifications is not entirely clear, or unproblematic. First, we need to distinguish between the detection of the instances of a problem, and the detection of a problem in the first place. Blaheta (2002) does not discuss how to detect, e.g., divergences from the annotation guidelines. Second, the annotation guidelines cannot describe everything that could occur in natural language, and thus the distinction between what is, or is not, mentioned in the guidelines is not always relevant for a classification of errors.

We propose an alternate approach to error classification, where all annotation is divided into annotation with variation and annotation without variation, across a corpus (or possibly corpora). The unit for which annotation varies (or not) may differ, depending on the type of annotation, but could be the PoS-tag for a given word, or the span or label of a phrase for a given sequence of words or PoS-tags, throughout the corpus. We can further divide annotation with variation into two subclasses, since the variation may stem from ambiguities, e.g., a word can have different PoS-tags depending on the context, or from errors. Annotation without variation, finally, may be correct, or contain systematic errors.

Automatic checks of well-formedness and consistency are essential tools to ensure quality for the parallel treebank. In the following we will give a short overview of how to ensure well-formedness in a treebank (Section 5.1). This is followed by an overview of work done on consistency checking for PoS tagging (Section 5.2) and parsing (Section 5.3). Finally, we conclude with our own work on consistency checking for alignment (Section 5.4).

5.1 Well-formedness in a Treebank

The first level of quality checks for a treebank is to see that the annotation is well-formed. This means that we need to check, e.g., that each token and each non-terminal node has a label, and is attached to the tree (see also Section 2.3). This is generally easy to check and should ideally be part of the annotation tool. Checking that a treebank is well-formed requires a program that runs through the treebank file or files searching for sentences that do not fulfil the definition of a tree (see Section 2.2). This means that one (or several) of the following is found.

- A token without label (or with multiple labels)
- A token not connected to the tree
- A sentence with more than one node without parent node

If a match is found the program needs to output an error message, preferably containing information about the sentence number and node containing the possible error.

The first point means that all tokens should have a label. The type of the label depends on the treebank. Most would require a PoS label. Some could also require morphological labels and/or lemmas. The second point deals with identifying words that are left out in the syntactic annotation. This point needs to be refined, depending on the annotation scheme. Different treebanks may take different positions on, e.g., whether special tokens like punctuation symbols should be part of the tree. For example, the Penn Treebank guidelines (Bies et al., 1995) require punctuation marks to be part of the tree whereas the German TIGER guidelines (Brants et al., 2002, Skut et al., 1997) leave them unattached. In the latter case, the program needs to disregard them (and give an error message if they are connected to the tree). Other information, for example XML tags, can be treated differently in different treebank annotation guidelines. Additionally, there are problematic cases, e.g., grammatical errors due to superfluous words. One German sentence in SMULTRON contains a separated verb prefix twice. We decided that even such 'superfluous' tokens need to be included in the tree, although they get a special function label.

The third point handles the fact that a sentence should not have more than one root node. If there is more than one top node, one of them is likely to be an unattached node. One additional point might be looking for a sentence with a single root node that does not have a top node label. This check would find errors in the root node label, which should ideally be the S (sentence) node. However, a tree with a different node label as top label does not necessarily

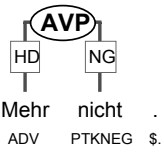


Figure 5.1: A German sentence consisting of an incomplete clause.

mean that there is an error. A treebank often contains incomplete clauses like headings or other fragments, as in Figure 5.1.

While well-formedness is easy to check for items where each instance (e.g., token) should be annotated, it is more difficult for items where no annotation is also information. This is, for example, the case for alignment in SMULTRON, and will be discussed in Section 5.4.

5.2 Consistency Checking for PoS-tagging

PoS-tagging can be handled automatically and fast with good results today, e.g., with statistical or rule-based methods, usually with an accuracy of over 95%, depending on language and tagset. Charniak (1997) pointed out that assigning the most common tag to each known word and the tag *proper noun* to all other words will approach 90% accuracy for English, since many words are unambiguous. However, even an accuracy of 95% means that if the average sentence is about 20 words long there is a tagging error in every sentence. If we add automatic (or semi-automatic) parsing on top of the PoS-tagging, it will induce errors that are propagated through the tree structure.

Consistency checking over treebanks is more complicated than checking well-formedness. Annotation consistency has turned into an interesting issue in recent years, as more annotated corpora have been developed. One approach to consistency checking for PoS tagging is to use one (van Halteren, 2000) or several (Loftsson, 2009) automatic taggers and compare the manual annotation to the automatic annotation to find discrepancies. Another approach is the variation n-gram method, developed by Dickinson and Meurers (2003a). It examines variation, which can be either due to ambiguity or tagging errors, and detects errors by exploring the context. A word appearing in the same context, defined as an n-gram, should be consistently tagged. This method will be discussed further in Sections 5.3 and 5.4.

5.3 Consistency Checking for Parsing

Consistency checking for parsing ensures that all trees of the treebank have been syntactically annotated in the same manner. To be able to correct annotation errors we want to automatically detect inconsistencies and automatically determine which tree structures vary because of ambiguities and which are errors. An early approach to consistency checking was proposed by Brants and Skut (1998). They worked on automation of treebank annotation through bootstrapping, using the system annotations as training material to improve their tools iteratively. Their approach to consistency checking entails having more than one annotation for each sentence, comparing these different annotations to find inconsistencies. However, multiple annotations of all data is not always an option when creating large corpora. Additionally, this method focuses on comparison on the sentence level, which may not detect more or less systematic problems throughout the corpus.

Ule and Simov (2004) present a method for detecting annotation errors in a treebank, assuming that errors are 'unexpected small tree fragments' and arguing that unexpected events occurring rarely in a corpus might be errors. They apply machine learning to automatically produce a list of error candidates. The method looks at non-terminal nodes in a treebank and compares the distribution of each node's children with its productions (i.e., children) when appearing under a certain parent node. Each iteration of the algorithm looks at a single type of focus node that has the most unexpected distribution of productions in a certain context and the distributions are compared using the χ^2 metric. The list of error candidates needs to be judged by a human annotator, to see if they are true errors. This method will, however, not detect frequent errors, meaning that systematic inconsistencies will be almost impossible to detect automatically. It will also mark rare phenomena as potential errors.

Dickinson and Meurers have proposed a number of different approaches to checking treebanks for consistency violations, including the previously mentioned variation *n*-gram method for error detection in PoS-tagging. Expansions of this method have been proposed to detect bracketing and labelling errors in syntactic annotation (e.g., Dickinson and Meurers, 2003b, 2005a). The approach is based on detecting strings that occur multiple times in the corpus with varying annotation, the so-called variation nuclei. Every variation detected in the annotation of a nucleus is classified as either an annotation error or as a genuine ambiguity. The basic heuristic for detecting annotation errors requires one word of recurring context on each side of the nucleus. The nucleus with its repeated surrounding context is referred to as a variation *n*-gram. Using only the immediately surrounding words as context is sufficient for detecting errors in grammatical annotation with high precision (Dickinson, 2008).

Parent category	Function label	Child category	Frequency
VP	accusative-object	cat="CNP"	6
VP	accusative-object	cat="NP"	96
VP	accusative-object	cat="PN"	1
VP	accusative-object	pos="ART"	1

Table 5.1: Consistency checking of functional triples, where rare children may point to errors. The determiner (ART) as accusative object is an error.

As reported in Samuelsson and Volk (2007a), we have used one of their early methods (Dickinson, 2006, Dickinson and Meurers, 2005b), which is based on the insight that the children in a local tree restrict the labels possible for the parent. For example, if the sequence of *Determiner - Noun* is mostly annotated as a noun phrase, then the few other occurrences are possible annotation errors. The method extracts all phrase structure rules from the treebank and sorts and counts them by their right-hand sides. This is useful as it reduces the manual labour.

The German and Swedish parts of SMULTRON have function labels for all edges. However, the method did not handle functional labels. We therefore generalized the method and developed our own approach to consistency checking of functional triples consisting of *function label - parent node - child node*. For example, we counted all different function labels of noun phrases that are children of an S node. According to the TIGER guidelines such noun phrases can only be of certain functions. Other function labels indicate a possible annotation error. The function triples were checked manually. For example, our German economy treebank, with around 7100 nodes, resulted in 277 different function triples. The method could thus be improved by formulating the constraints over functional triples explicitly, to check them automatically.

Table 5.1, an excerpt from our German treebank, shows that we have annotated 96 noun phrases as accusative object in a verb phrase. We have also annotated 6 coordinated noun phrases (CNP), one proper noun (PN) and one determiner (ART) as accusative objects. The latter turned out to be an annotation error. A relative pronoun was erroneously tagged as a determiner (they are homographs in German). By changing the PoS tag to relative pronoun and inserting the appropriate noun phrase node, the error was corrected. The approach was not properly evaluated, but we found it helpful for locating errors.

Some researchers advocate finding annotation errors by finding violations of the annotation guidelines. One such consistency check for, e.g., the German

treebank is checking coordinated phrases, which are marked by a different label than non-coordinated phrases. They should only consist of the nodes to be coordinated and possibly conjunctions. Thus, a CVP may dominate VPs and conjunctions. As we find other children, for example verbs, we can assume that these are errors. However, while finding such errors is not difficult, finding the problematic area from the guidelines is. To automatically detect such issues we need to encode the guidelines as explicit rules, which is usually not a simple, straightforward task. This approach is mainly feasible if the annotation is grammar-driven (see, e.g. Rosén and De Smedt, 2007, Rosén et al., 2006).

5.4 Consistency Checking for Alignment

In addition to the completeness check and the consistency check for tagging and parsing, we need to check the alignment. High-quality alignments, i.e., alignments that are free from errors, are crucial to parallel corpora. However, as for other annotation, even gold standard annotation can contain annotation errors. These can come from any step in the mark-up process, such as automatic (pre-)processes, human annotation, or human post-editing.

There have been few, if any, attempts to develop general approaches to error detection for aligned corpora. During the creation of SMULTRON, several methods for consistency checking of the alignments were employed. We checked whether the aligned source and target token sequences differ in length (calculated as number of characters). Large length differences might point to erroneous alignments. We also checked, for all aligned single tokens and all aligned token sequences (phrases), whether they are aligned with the same alignment labels (i.e., with the predicate 'exact' or 'fuzzy') to the same corresponding tokens. For example, the pair consisting of the English *and* and the German *samt* ('together with') had 20 fuzzy matches and one erroneous exact match. The labelling variations had to be checked manually.

One limitation of this method is that it uses no grammatical abstraction, e.g., PoS-tags. For example, the English-Swedish alignment pair <*higher*, *ökade*> has acceptable variation between 2 fuzzy tags and 3 exact tags. The uses of *ökade* ('increased') are quite different in the different instances, however. Even though *higher* is always JJR (comparative adjective), *ökade* can be VBFIN (finite verb), as in <*orders were higher*, *orderingången ökade*> or PC (participle), as in <*higher material costs*, *ökade materialkostnader*>.

Additionally, we examined those cases where different types of phrases are aligned across the languages (e.g., when an adjective phrase in one language is aligned with a prepositional phrase in the other). The consistency checks are done manually over an extracted table of the aligned token sequences (with their phrase labels). This allows us to sort the token sequences according to

different criteria and to abstract away from the dense forest of syntactic information and alignment lines in the TreeAligner (see p. 56).

Recent versions of the TreeAligner also contain a module for consistency checks that are computed during annotation. It checks a number of structural constraints, which are applied regardless of language or corpus, as they express certain invalid subgraphs. One structural constraint, which has proven useful to the annotators and helped them spot simple alignment mistakes, is branch link locality. It demands that if two phrases $<p_1, p_2>$ are aligned, any node dominating p_1 can only be aligned to a node dominating p_2. Additionally, the consistency module measures association strength between collocates, looking, e.g., at the word-word or PoS-PoS combinations, pinpointing anomalous translation equivalents. While potentially effective,[1] these methods do not address the use of alignments in context, i.e., when we might expect to see a rare translation. Additionally, such methods are limited, in that they do not, e.g., handle missing alignments. In the rest of this section we propose two methods, which address these limitations and can be used to complement this work. The two methods have been developed in cooperation with Markus Dickinson, and much of the following has been previously published as Dickinson and Samuelsson (2010).

5.4.1 Variation N-gram Method for Alignment Quality Checking

As a starting point for an error detection method for alignment in parallel corpora, we used the variation *n*-gram approach for syntactic annotation (Dickinson and Meurers, 2003b, 2005a). The variation *n*-gram method relies upon setting up a one-to-one mapping between a (potentially discontinuous) string and a label. To adapt the variation *n*-gram method and determine whether strings in a corpus are consistently aligned, we must 1) define the units of data we expect to be consistently annotated, and 2) define which information effectively identifies the erroneous cases.

Units of Data

Alignment is a relationship between words in a source language and words in a target language. Even phrase-to-phrase alignments contain a string-to-string mapping, from the yield of the source phrase to the yield of the target phrase. Following the variation *n*-gram method, we can thus define the units of data, i.e., the variation nuclei, as strings.

Since alignments are string-to-string mappings, the natural choice for the label is the target language string. Although alignments are (bi-directional)

[1] The methods have not been properly evaluated yet.

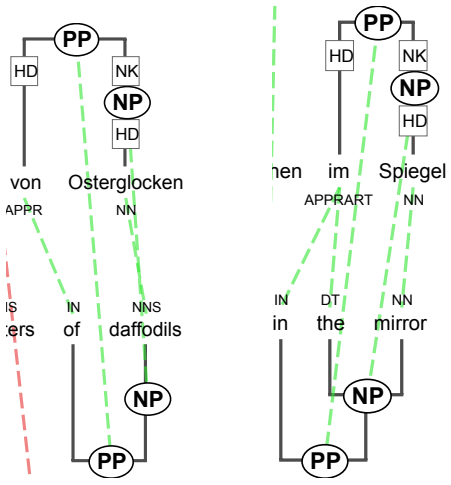


Figure 5.2: Word and phrase alignments span the same string on the left (the word *daffodils* vs. the NP *daffodils*), but not on the right (the word *mirror* vs. the NP *the mirror*).

relations between two languages, we break the problem down into two different source-to-target mappings. With a German-English aligned corpus, for example, we look for the consistency of aligning German words to their English counterparts and separately examine the consistency of aligning English words with their German 'labels.' Additionally, because a translated word can be used in different parts of a sentence, including the beginning, we normalize all target language string labels into lower-case, thus preventing variation between, e.g., *the* and *The*.

This general idea will work for any aligned corpus, but there is an additional question of how to handle alignments that operate on the phrase level. Although alignment is a string-to-string mapping, complications may arise if a phrase node spans over only one word, which has two distinct mappings, one as a word and one as a phrase, which may or may not result in the same yield. Figure 5.2 illustrates this. On the left side, *Osterglocken* is aligned to *daffodils* at the word level, and the same string is aligned on the phrase level (NP to NP). In contrast, on the right side, the word *Spiegel* is aligned to the word *mirror*, while at phrase level, *Spiegel* (NP) is aligned to *the mirror* (NP). We do not want unnecessary variation between a word-level string (e.g., *mirror*) and a phrase-level string (e.g., *the mirror*).

Thus, we further split the error detection method into one on the word level and one on the phrase level. Phrases potentially behave differently from

words, and as we will see, these levels have different requirements for handling strings that have been left unaligned, so this split allows for cleaner processing. By splitting the problem first into different source-to-target mappings and then into words and phrases, we conveniently do not have to change the underlying way of finding consistency through the variation *n*-gram method.

For phrase alignment, there is an intermediary layer of syntactic annotation. Thus, we must ensure that we properly account for syntactic annotation that violates our assumptions. In particular, unary branches present a potential difficulty, in that a single string could have more than one label, violating the assumption that the string-to-label mapping is a function. For example, in Penn Treebank-style annotation, an NP node can dominate a QP (quantifier phrase) node via a unary branch. Thus, an annotator could (likely erroneously) assign different alignments to each phrasal node, one for the NP and one for the QP, resulting in different target labels.

We handle all the (source) unary branch alignments as a conjunction of possibilities, ordered from top to bottom. Just as the syntactic structure can be relabelled as NP/QP (Dickinson and Meurers, 2003b), we can relabel a string as, e.g., *the man*/*man*. If different unary nodes result in the same string (*the man*/*the man*), we combine them (*the man*). Note that unary branches are unproblematic in the target language since they result in the same string, i.e., are still one label.

Since we have defined the problem as a mapping between source strings and their target string labels, this naturally handles any one-to-many, or many-to-many, alignment. For example, if *Gärten* maps to *the gardens*, we do not see the alignment as two separate links for *the* and *gardens*, but *the* and *gardens* is considered one string. Likewise, in the opposite direction, *the gardens* maps as a unit to *Gärten*, even if discontinuous.

Error detection for syntactic annotation finds inconsistencies in constituent labelling, e.g., NP vs. QP, and inconsistencies in bracketing, e.g., NIL (non-bracketed) vs. NP. Similarly, we can distinguish between inconsistency in labelling (different translations) and inconsistency in alignment (aligned/unaligned). Detecting inconsistency in alignment deals with the completeness of the annotation, by using the label NIL for unaligned strings.

We use the method from Dickinson and Meurers (2005a) to generate NILs, but using NIL for unaligned strings is too coarse-grained for phrase-level alignment. A string mapping to NIL can be a phrase that has no alignment, or it may not be annotated as a phrase and thus could not possibly be aligned as one unit. Therefore, we create NIL-C as a new label, which indicates a constituent with no alignment, differing from NIL strings that do not even form a phrase. For example, on the left side of Figure 5.3, the string *someone* is aligned to *jemanden* on both the word and the phrase level. On the right side of Figure 5.3, the

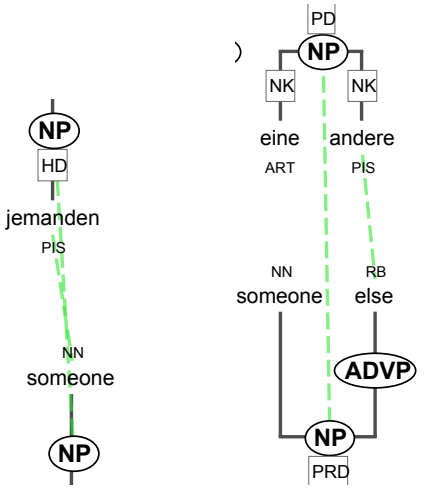


Figure 5.3: The English word *someone* aligned as a phrase on the left, but not a phrase by itself on the right.

string *someone* by itself does not constitute a phrase (even though the alignment in this instance is correct) and is labelled NIL. If there were instances of *someone* as an NP with no alignment, this would be labelled NIL-C. Distinguishing between these types of NIL seems to be useful for inconsistency detection, as we primarily expect consistency for items annotated as a phrase. However, by using the non-constituent NIL label we do not have to rely on potentially erroneous phrase annotation. We could thus find alignment errors even though there are also errors in the syntactic annotation.

Aligned corpora often specify additional information about each alignment, as was done, e.g., by Och and Ney (2003), distinguishing between sure (unambiguous) and possible (ambiguous) alignment. In SMULTRON, as discussed in Section 4.1, an exact alignment means that the strings are considered direct translation equivalents outside the current sentence context, while a fuzzy alignment is not as strict a translation equivalent.

Generally, details about the alignment can be used to control which variations are worth examining further and which are not. Since exact alignments are the ones we expect to consistently align with the same string across the corpus, we attach information about the alignment type to each corpus position. This can be used to filter out variations involving, e.g., fuzzy alignments.

When multiple alignments form a single variation nucleus, there could be different types of alignment for each link, e.g., *dog* exact-aligning and *the*

fuzzy-aligning with *Hund*. We did not observe this, but one can easily allow for a mixed type (exact-fuzzy).

The Algorithm

In a pre-processing step, we extract the necessary information from the TIGER-XML (Mengel and Lezius, 2000) treebank files and the TreeAligner-XML (Volk et al., 2006) alignment files, through XSLT-processing. The algorithm then splits the data into the four appropriate units, two SL-to-TL mappings (SL=source language, TL=target language) each for the word-level and phrase-level alignments. For each of the four sets of alignments, perform the following:

1. Map each string in SL with an alignment to a label
   - Label = <(lower-cased) TL translation, exact|fuzzy|exact-fuzzy>
   - Join constituent phrases that are unary branches, giving them a single, normalized label representing all target strings.
2. Generate NIL alignments for string tokens that occur in SL, but have no alignment to TL, using the method described in Dickinson and Meurers (2005a).
   - Constituent phrases with no alignment are given the special label, NIL-C.
3. Find SL strings that have variation in labelling.
4. Filter the variations from step 3, based on the likelihood of being an error, to distinguish between errors and acceptable variations.

The algorithm essentially works by first extracting the string-to-label mappings from the alignment data. The variation *n*-gram method then proceeds in a straightforward manner (steps 2-4). As words and phrases have acceptable variants for translation, the described method will lead to detecting acceptable variations. We use several heuristics to filter the set of variations.

Filtering the Variations

The variation *n*-gram method has generally relied upon immediate lexical context around the variation nucleus, in order to sort errors from ambiguities (see, e.g., Dickinson, 2008). For alignment, we tested three different notions of context. Matching the variation *n*-gram method, we first employed a filter identifying those nuclei that share the 'shortest' identical context, i.e., one word

of context on every side of the nucleus. Secondly, we relaxed this to require only one word of context, on either the left or the right side of the nucleus. Finally, we ignored the context in the source language and relied only on other filters. As an example, consider the nucleus *come* in the sentence *Where does the world come from*. The first notion of context requires *world come from* to recur, the second either *world come* or *come from*, and the third only requires that the nucleus itself recur (*come*).

As discussed previously, the label NIL-C refers to syntactic constituents that are unaligned, while NIL refers to non-constituent strings without alignment. A string that varies between NIL and NIL-C, then, is not really varying in its alignment, i.e., it is always unaligned. We thus removed cases varying only between NIL and NIL-C.

Translation is an open-ended task, which means that there can be different translations in a corpus. In contrast, for PoS and syntax, there is generally a clear-cut annotation, given a particular annotation scheme. Thus, it is not clear how useful the surrounding context is for translation tasks, given the wide range of possible translations for the same context. Further, requiring identical context around source words is very strict, leading to sparse data problems, and it ignores alignment-specific information, like information in the target language and information about the links between the languages.

Instead, one could explore properties related to the target language and the alignment itself. Information about what possibilities exist in the target translation can provide better information as to what constitutes an error. We want to filter out cases where there is variation in alignment stemming from multiple translation possibilities.

We implemented a target language filter, which keeps only the variations where the target words are present in the same sentence. If word x is sometimes aligned to y_1 and sometimes to y_2, and word y_2 occurs in at least one sentence where y_1 is the chosen target, then we keep the variation. If y_1 and y_2 do not occur in any of the same sentences, we remove the variation: given the translations, there is no possibility of having the same alignment.

This works for both regular labels and NIL labels, given sentence alignments. Sentence alignments in SMULTRON are not given directly, but can be deduced from the set of word alignments. For NIL labels, the check is in only one direction: the aligned sentence, which the NIL string's sentence corresponds to, must contain the target string used as the label elsewhere in the corpus. For instance, the word *All* aligns once with *alle* and twice with NIL. We check the two NIL cases to see whether one of them contains *alle*.

A final filter relies on alignment type information. Namely, the fuzzy label already indicates that the alignment is not perfect, i.e., not necessarily applicable in other contexts. For example, the English word *dead* fuzzy-aligns with

	Word-level	Phrase-level
all	540	251
oneword	340	182
shortest	96	21
all-TL	194	140
oneword-TL	130	94
shortest-TL	30	16

Table 5.2: Number of variations found for word and phrase-level, using three different notions of context, and with or without the target language (TL) filter.

the German *verschwunden* ('gone, missing'), which is not a perfect translation, but the best in its context. In another part of the corpus, *dead* exact-aligns with *leblosen* ('lifeless'). While this is variation between *verschwunden* and *leblosen*, the presence of the fuzzy label alerts us to the fact that it should (or at least could) vary with another word. The alignment type filter removes cases varying between one exact label and one or several fuzzy labels.

Evaluation

Evaluation was done for English to German on half of SMULTRON (the part taken from the novel *Sophie's World*), see Appendix A for numbers of words and alignments. The number of variations found for the different settings are given in Table 5.2. As mentioned, we filter NIL/NIL-C and fuzzy/exact variations. In the following, we experiment with the target language (*TL*) filter, and three different contextual definitions: no context, i.e., all variations (*all*); one word of context on either the left *or* the right (*oneword*); and one word of context on both the left *and* the right, i.e., the shortest possible surrounding context (*shortest*). The filters reduce the number of variations, as can be seen in Table 5.2, with a dramatic loss for the shortest contexts.

A main question concerns the impact of the filtering conditions on error detection. To gauge this, we randomly selected 50 (*all*) variations for the word level and 50 for the phrase level. For each level, the 50 variations corresponded to around 400 corpus instances, with an average of 2.7 labels per variation. The variations were checked manually to see which were true variations and which were errors.

We report the effect of different filters on precision and recall (multiplied by 100) in Table 5.3. Recall is calculated with respect to the *all* condition.

	Cases	Errors	Prec	Rec
Word-level				
all	50	17	34.0	100.0
oneword	33	12	36.4	70.6
shortest	8	2	25.0	11.8
all-TL	20	11	55.0	64.7
oneword-TL	15	6	40.0	35.3
shortest-TL	2	1	50.0	5.9
Phrase-level				
all	50	15	30.0	100.0
oneword	33	8	24.2	53.3
shortest	4	1	25.0	6.7
all-TL	27	12	44.4	80.0
oneword-TL	14	7	50.0	46.7
shortest-TL	3	1	33.3	6.7

Table 5.3: Evaluation of the filtering heuristics to remove acceptable variations, for word and phrase-level, using three different notions of context, and with or without the target language (TL) filter.

Adding too much lexical context in the source language (i.e., the *shortest* conditions) results in too low a recall to be practically effective. Using one word of context on either side has higher recall, but the precision is no better than using no source language context at all. What seems to be most effective is to only use the target language filter (*all-TL*). Here, we find higher precision, higher than any source language filter, and the recall is respectable.

Errors are categorized as a) an added link (i.e., a link that should not be there), b) a missing link, or c) an alignment type error (i.e., fuzzy or exact). The number of token errors can be seen in Table 5.4. It should be noted that an error can be of several types at the same time.

In Figure 5.4 the nucleus is *came*. The German word *stammte* is aligned to *came from*, meaning that *came* has a NIL alignment in isolation. This example is categorized as an added link error, because the link to *from* is wrong (*woher* should preferably be aligned to *where from*). Additionally, *stammte* ('originated or descended') is only an approximate translation of *came*, which should be marked as fuzzy instead of exact, and is thus categorized as a link type error.

Of the 50 word-level *all* variations, 17 pointed to errors (i.e., precision is

	Total	Added	Missing	Type
Word-level				
all	28	6	21	2
oneword	20	5	14	2
shortest	3	0	3	0
all-TL	19	6	13	1
oneword-TL	10	5	5	1
shortest-TL	2	0	2	0
Phrase-level				
all	31	2	22	8
oneword	15	1	13	2
shortest	1	0	1	0
all-TL	27	2	19	7
oneword-TL	14	1	12	2
shortest-TL	1	0	1	0

Table 5.4: Found errors (tokens) for different contexts, with and without target language (TL) filtering, according to error type.

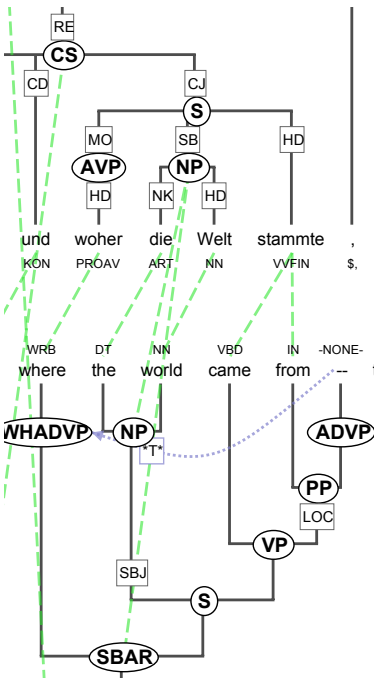


Figure 5.4: The English word *came* is involved in an added link error and a link type error.

34 for this sample), with a total of 28 erroneous links: 21 missing links, 6 added links, and 2 type errors. For the phrase level, 15 of the 50 variations pointed to errors (precision is 30), for a total of 31 erroneous links: 22 missing links, 2 added links, and 8 type errors.

Some observations can be made. First, there are slightly more type errors and less added link errors at the phrase level than at the word level, which could point to a difficulty for annotators to agree about consistency of translations for phrases. If the annotators are unsure about these decisions, they tend to leave the phrases unaligned, while word alignments occasionally receive too many links. Second, using either one word of context or the target language filter (but not both) seems to preserve the most errors, with the target language filter being preferable for phrases.

To evaluate recall more properly, we conducted a small evaluation of 20 randomly chosen sentence pairs. Sentence pairs with less than 5 tokens in one of the sentences were ignored. The same data as above (the English-German Sophie part of SMULTRON) was used, but the system was run for both directions (both English-German and German-English) and for both the word level and the phrase level, using the target language filter (*all-TL*), which achieved the best results in the previous evaluation (Table 5.3). These 20 sentence pairs (there were 21 sentences in each language, since one German sentence was aligned to two English sentences, and one English to two German) contained a total of 287 German and 318 English tokens, and 179 German and 278 English phrases. The 20 sentence pairs contained a total of 22 errors.

There were 378 variating items for all the evaluated sentences, counting both word and phrase level and both language directions. Counting items present in both language directions only once results in 292 items. Of these, 25 pointed to 14 errors (10 on word level and 4 on phrase level), while 8 errors (2 on word level and 6 on phrase level) were missed. Thus, recall is 64 (83 on word level and 40 on phrase level). A check of the variations without the target language filter (*all*) shows that one of the missed word-level errors and two of the phrase level errors are present there. Thus 5 errors are not found at all by the method. Of these, the word level error is a missing link between the words *liegen* and *lay*. The word *liegen* occurs three times in the whole corpus (including this occurrence) and is unaligned all three times, while *lay* only occurs this once. Thus, data sparseness is a problem for the model.

The program does not find variations locally, for individual trees, but globally, for the whole corpus. A marked variation thus points to possible errors involving this word somewhere in the corpus. For example, the German word *Du* ('You') is marked as a variation, 39 times with exact alignment to *you*, and twice being unaligned. Here, we can assume that the alignment between *Du* and *you* is correct, and instead inspect the two unaligned cases. However, since

all 39 occurrences still appear in the output of the program, it does not make sense to calculate precision when evaluating based on a particular sentence.

An advantage of the target language filter is its ability to handle lexical variations. One example of this is the English phrase *a dog*, which varies between German *einem Hund* (dative singular), *einen Hund* (accusative singular) and *Hunde* (accusative plural). Similar to the lower-case labels, one could map strings to canonical forms (e.g., lemmata). However, the target language filter naturally eliminates such unwanted variation, without any language-specific information, because the other forms do not appear across sentences.

Several of the variations that the target language filter incorrectly removes would, once the error is fixed, still have variation. As an example, consider *cat*, which varies between *Katze* (5 tokens) and NIL (2 tokens). In one of the NIL cases, the word should be fuzzy-aligned with the German *Tigerkatze*. The variation points out the error, but there would still be variation (between *Katze*, *Tigerkatze*, and NIL) after correction. This shows the limitation of the heuristic in identifying the required non-exact alignments. For NIL cases, one might thus consider relaxing the target language filter, by identifying sentences or phrases that are paraphrases (see, e.g., Okita, 2009) and filtering out only those NIL examples.

Another error which the filter misses is the variation nucleus *heard*, which varies between *gehört* (2 tokens) and *hören* (1 token). In this case, one of the instances of <*heard*, *gehört*> should be <*heard*, *gehört hatte*>. Note that here the erroneous case is not variation-based at all; it is a problem with the label *gehört*. This illustrates that the cases the target language filter misses are largely not issues for a variation-based method to solve. We need another method to detect more translation possibilities.

As an example of a problem for phrases, consider the variation for the nucleus *end* with 5 instances of NIL and 1 of *ein Ende*. In one NIL instance, the proper alignment should be <*the end*, *Ende*>, with a longer source string. Since the target label is *Ende* and not *ein Ende*, the filter removes this variation. One might explore more fuzzily matching NIL strings, so that *Ende* matches with *ein Ende*. We explore a different method for phrases next, which deals with some of these NIL cases.

5.4.2 Quality Checking of Phrase Alignment Based on Word Links

Although the string-based variation method of detecting errors works for any type of aligned corpus, it is limited in the types of errors it can detect. There may be ways to generalize the variation *n*-gram method, as it has been generalized to increase recall in other domains (see, e.g., Dickinson, 2008), but this does not exploit properties inherent to aligned treebanks. We pursue a comple-

mentary approach, as this can fill in some gaps a string-based method cannot deal with (see Loftsson, 2009).

Using the existing word alignments, we can search for missing or erroneous phrase alignments. If the words dominated by a phrase are aligned, the phrases generally should be, too. We take the yield of a constituent in one side of a corpus, find the word alignments of this yield, and use these alignments to predict a phrasal alignment for the constituent. This is similar to the branch link locality of the TreeAligner, and has been called the crossing constraint (Wu, 1997) or the well-formedness constraint (e.g., Lavie et al., 2008, Tinsley et al., 2007b). The difference is that we use it as a prediction, rather than a restriction, of alignment.

For example, consider the English VP *choose her own friends* in Example 5.1. Most of the words are aligned to words within *Ihre Freunde vielleicht wählen* ('possibly choose her friends'), with no alignment to words outside of this German VP. We thus want to predict that the phrases be aligned.

(5.1) a. [$_{VP}$ choose$_1$ her$_2$ own friends$_3$]

b. [$_{VP}$ Ihre$_2$ Freunde$_3$ vielleicht wählen$_1$]

The Algorithm

The algorithm works as follows:

1. For every phrasal node s in the source treebank:
   (a) Predict a target phrase node t to align with, where t could be non-alignment (NIL):
      i. Obtain the yield (i.e., child nodes) of the phrase node s: s_1, ... s_n.
      ii. Obtain the alignments for each child node s_i, resulting in a set of child nodes in the target language (t_1, ... t_m).
      iii. Store every parent node t' covering all the target child nodes, i.e., all $<s,t'>$ pairs.
   (b) If a predicted alignment ($<s,t'>$) is not in the set of actual alignments ($<s,t>$), add it to the set of potential alignments, $A_{S\mapsto T}$.
      i. For nodes that are predicted to have non-alignment (but are actually aligned), output them to a separate file.
2. Perform step 1 with the source and target languages reversed, thereby generating both $A_{S\mapsto T}$ and $A_{T\mapsto S}$.

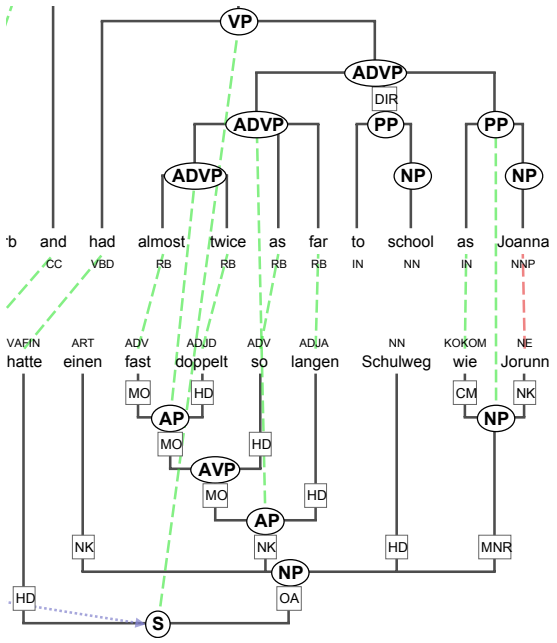


Figure 5.5: Since most words of the English ADVP *almost twice as far to school at Joanna* are aligned to words in the German NP *einen fast doppelt so langen Schulweg wie Jorunn*, the phrases should be aligned.

3. Intersect $A_{S \mapsto T}$ and $A_{T \mapsto S}$, to obtain the set of currently unaligned predicted phrasal alignments.

The main idea in Step 1a, where we predict possible targets to align with the source language node, is to find the children of a source node and their alignments and then obtain the target nodes that have all of these aligned nodes as children. Such a target node is a plausible candidate for alignment.

Consider Figure 5.5. Within the 8-word English ADVP (*almost twice as far to school at Joanna*), six words are aligned to words in the corresponding German sentence, all under the same NP. It does not matter that some words are unaligned; the English ADVP and the German NP cover basically the same set of words, suggesting that the phrases should be aligned, as is the case here.

The prediction of an aligned node in Step 1a allows for multiple possibilities: in Step 1(a)iii, we only check that a parent node t' covers all the target children, disregarding extra children, since translations can contain extra words. In general, many such dominating nodes exist, and most are poor

candidates for alignment of the node in question. This is the reason for the bidirectional check in Steps 2 and 3.

For example, in Example 5.2, we correctly predict alignment between the NP immediately dominating *you* in English and the NP immediately dominating *man* in German.

(5.2) a. But it 's just as impossible to realize [$_S$ [$_{NP}$ **you**$_1$] have to die without thinking how incredibly amazing it is to be alive] .

b. [$_S$ Und es ist genauso unmöglich , darüber nachzudenken , dass [$_{NP}$ **man**$_1$] sterben muss , ohne zugleich daran zu denken , wie phantastisch das Leben ist .]

From the word alignment, we generate a list of parent nodes of *man* as potential alignments for the *you* NP. Two of these (six) German nodes are shown. In the other direction, there are eight nodes containing *you*; two of these English nodes are shown. These are the predicted alignment nodes for the NP dominating *man*. In either direction, this overgenerates, while the intersection only contains alignment between the lowest NPs.

We initially tried to reduce the set of potential nodes in Step 1(a)iii, by choosing nodes closest in length. This heuristic, however, can eliminate candidates that are correct, including the target node that the source node already aligns with, thus resulting in too many false positives.

The method is intuitive and generally effective, but certain predictions are less likely to be errors. In Figure 5.6, the sentence pair is a complete rephrasing, and <*her*, *ihr*> is the only word alignment. The method only requires that all words of a SL phrasal node be aligned with the words of the TL node. Here, there are several such nodes. Thus, the English PP *on her*, the VP *had just been dumped on her*, and the two VPs in between are predicted as possible alignments of the German VP *ihr einfach in die Wiege gelegt worden* or its immediate VP child: they all have *her* and *ihr* aligned, and no contradicting alignments. Sparse word alignments lead to multiple possible phrase alignments. We do not evaluate cases with more than one predicted source or target phrase here.

Additionally, we mark those cases in the output where we predict alignment, yet either the source or the target node is already aligned, as these cases are less likely to be errors. For instance, in Example 5.3, there is (correct) existing alignment between the English S *live here* and the German S *die wir hier wohnen*, yet we incorrectly predict alignment with a larger English constituent, the SBAR *who live here*. However, such cases are rare (11 out of 453 cases).

(5.3) a. We [$_{SBAR}$ who [$_{S_1}$ live here]] are microscopic insects ...

b. Wir , [$_{S_1}$ die wir hier wohnen] , sind das wimmelnde Gewürm ...

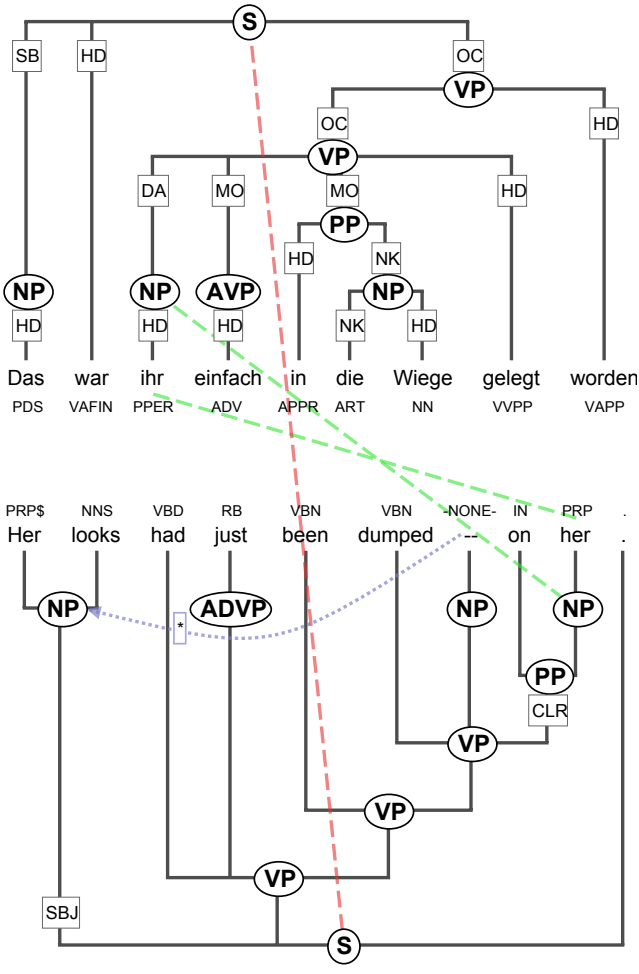


Figure 5.6: A sentence pair with few word alignments is problematic for predicting phrase alignments.

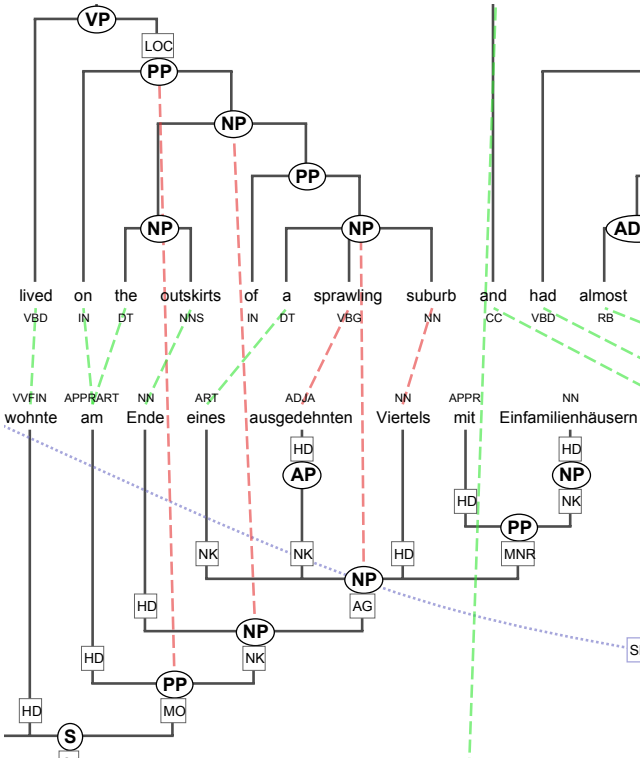


Figure 5.7: The English NP *the outskirts of a sprawling suburb* is predicted to have no alignment, as the determiner *the* is aligned to *am*, which is outside the corresponding German NP.

If in Step 1(a)iii, no target parent (t') exists, but there is alignment in the corpus, then in Step 1(b)i, we output predicted non-alignment. In Figure 5.7, for instance, the English NP *the outskirts of a sprawling suburb* is (incorrectly) predicted to have no alignment, although most words align to words within the same German NP. This prediction arises because *the* aligns to *am* outside of the German NP, due to *am* being a contraction of the preposition *an* ('on') and the article *dem* ('the'). The method for predicting phrase alignments relies upon words being within the constituent. We thus conclude that: 1) the cases in Step 1(b)i are unlikely to be errors, and 2) there are types of alignments that we simply will not find, a problem also for automatic alignment based on similar assumptions (e.g., Zhechev and Way, 2008). If the NPs in Figure 5.7 were unaligned, we would not predict alignment between them.

	Items	Correct	Precision
Total	55	25	45.5
Length difference 0	21	18	85.7
Length difference 1	16	3	18.8
Length difference >1	18	4	22.2

Table 5.5: Evaluation of the phrase alignments suggested based on word alignments, broken down into length differences (number of tokens) between the source and target phrases.

Evaluation

The method returns 318 cases, apart from 135 cases with multiple source/target phrases and 104 predicted non-alignments. To evaluate, we sampled 55 of the 318 flagged phrases and found that 25 should have been aligned as suggested (a precision of 45.5). The results are shown in Table 5.5.

21 of the phrases have zero difference in length between source and target (calculated as number of tokens), while 34 have differences of up to 9 tokens. Of the phrases with zero-length difference, 18 should have been aligned (a precision of 85.7), while only 7 with length differences should have been aligned (a precision of 20.6). This is in line with previous findings that length differences can help predict alignment (see, e.g., Gale and Church, 1993). Interestingly enough, of the 16 suggested phrase pairs with a length difference of only 1, 13 are incorrect. About half of all phrase pairs that should be aligned should be exact, regardless of the length difference.

The method is good at predicting the alignment of one-word phrases, e.g., pronouns, as the *you-man* case in Example 5.2. Of the 11 suggested alignments where both source and target have a length of 1, all were correct suggestions. This is not surprising, since all words under the phrases are (trivially) aligned. Such a check is, however, extremely helpful (see p. 74).

Although shorter phrases with small length differences have a higher probability of being correct suggestions, we do not want to filter out items based on phrase length. There are outliers that are correct suggestions, e.g., a phrase pair with lengths of 15 and 13 (a difference of 2) or a pair of 31 and 36 (a difference of 5). It is worth noting that checking the suggestions took very little time, and maybe checking some bad suggestions is preferable to missing some good suggestions.

As with the previous method, we carried out an additional evaluation, to

assess recall more properly. The same set of 20 sentence pairs was checked. As the consistency check is bi-directional, reversing source and target languages gives the same set of alignment suggestions. There were 10 suggestions for the set of 20 sentence pairs, also counting cases with multiple source/target phrases. As previously stated, the sample contained a total of 22 true errors, of which 10 were at phrase level. 4 of the 10 suggestions made by the phrase link predictor pointed to errors, while 6 errors were missed by this method, rendering a recall of 40.

Of the 8 errors missed by the variation n-gram method with the target language filter, 6 were at phrase level, and 3 of these 6 were found by the phrase link prediction method. This means that the two methods for consistency checking of alignment are only partially complementary. A few additional errors would have been caught when not using the target language filter (the *all* condition) for the previous method. In total, one error at word level and two at phrase level (of the 22 errors in the sample), could not be found by either method. Thus, there is still room for improvement, both in finding errors and in reducing the false positives.

5.5 Summary

In this chapter, we have discussed automatic methods to detect annotation errors in a parallel treebank, thus trying to improve the work of the two previous chapters. As both machines and humans produce annotation errors, such quality checks are essential in producing a high-quality parallel treebank. We have proposed an error classification where annotation is divided into annotation with variation, either ambiguous or erroneous, and annotation without variation, either correct or systematically erroneous.

Additionally, we have examined a number of approaches to quality checks, exploring annotation with variation. They help us ensure that the annotation is well-formed and consistent, which is crucial for quality. While several of the methods discussed are effective and can be used to suggest revisions for incorrect annotation, they still do not give a final, entirely automatic, solution. The suggestions are, however, helpful and possibly essential for guiding manual quality control.

This chapter concludes Part I of the thesis, which has described our endeavour to build a high-quality parallel treebank. In Part II we will explore automatic ways of expanding and improving a parallel treebank, using SMULTRON as a gold standard.

Part II

Boosting Quality through Annotation Projection

6. Automatic Alignment and Annotation Projection

Before we turn to the experiments on improving the automatic annotation of a parallel treebank, some background is needed. Alignment is crucial for annotation projection, and Section 6.1 deals with automatic methods for alignment at sentence, phrase, and word level. This is followed, in Section 6.2, by an overview of previous research in the area of annotation projection.

6.1 Automatic Alignment Methods

As was described in Chapter 4, alignment shows us which part of a text (or tree) in one language corresponds to which part in another language. Alignment can be defined at numerous different levels, ranging from document, paragraph, and sentence, to phrase, word, or character level. In the following, we will focus on sentence, phrase, and word alignment. Alignment is useful for a wide variety of applications, such as machine translation, multilingual lexicon extraction and domain-specific terminology extraction, word sense disambiguation, and annotation projection, just to name a few.

Automatic alignment allows us to to align larger data sets in a fast and efficient way. Most alignment systems are developed for machine translation purposes. Both sentence and word alignment has to handle possible one-to-many alignments, i.e., sometimes a sentence in one language is translated as two or three sentences in the other language. In other respects, sentence alignment and word alignment are fundamentally different. While sentence order is often (but not always!) preserved in a translation, such a monotonicity assumption is not possible for word alignment, which needs to allow for word order differences and thus for crossing alignments.

A visualization tool is crucial when annotating, correcting, or exploring alignments. An overview of a number of tools that allow manipulation of word and phrase alignments can be found in, e.g., Samuelsson and Volk (2007a).

This section will give a short overview of automatic alignment methods.[1]

[1] Additional, recent, overviews can be found, e.g., in Wu (2010) and Tiedemann (2011).

It serves as a background for our experiments on automatic alignment, which will be explored in Chapter 8. We will also discuss a variety of metrics for evaluation of alignment.

6.1.1 Automatic Sentence Alignment

There are well-established algorithms for aligning sentences (which represent translation correspondences) across parallel corpora. The models are usually based on statistics, linguistic knowledge, or both. Statistical methods are better for large corpora, while linguistic methods are better for small corpora (see, e.g., McEnery and Wilson, 2001, p. 151f). The algorithms are based on features such as sentence length, word correspondences (as taken from bilingual dictionaries, or automatically found cognates), and distance factors. Such sentence alignment algorithms have even found their way into commercial translation tools as a means to fill translation memories with previously translated texts (Groß, 1998).

Alignment methods based on sentence length rely on the fact that a long sentence is usually translated into a long sentence. The approach was first described by Gale and Church (1991) and Brown et al. (1991), and is further described and evaluated in Gale and Church (1993). The two methods are similar, in that dynamic programming is used to determine sentence mapping across paragraphs, based on sentence length. This alignment is monotonic, but allows for sentences to be unaligned, or linked to one or more sentences in the other language. However, while Brown et al. (1991) calculate sentence length in number of tokens, Gale and Church (1991) count number of characters and find that character based models outperform word based models, at least for similar languages like English, German and French.

Lexical alignment methods, which use word correspondences, were introduced by Kay and Röscheisen (1993).[1] They used partial alignment at word level as anchor points for a maximum likelihood sentence alignment, which in turn iteratively improved the word alignment. While the algorithm is computationally expensive, their initial evaluation achieved good results with an accuracy of over 96%.

Simard et al. (1992) proposed using cognates as anchors for sentence alignment. Cognates are pairs of tokens that share some property, for example that they are both punctuation characters, contain digits, or contain the same letter sequences. The cognate-based alignment did not outperform the length-based alignment and only gave a small improvement when combined with the length-based algorithm. It seems that while there is a difference in number of cognates

[1] First published as Kay and Röscheisen (1988).

between sentences that are translations and random sentence pairs, it is still difficult to distinguish the two categories due to a large variance. Additionally, as pointed out by, e.g., Wu (2010), the method cannot handle very dissimilar languages, for example when no alphabetic matching is possible, as in English and Chinese.

An additional approach, Bleualign (Sennrich and Volk, 2010, 2011), involves using machine translations and the Bleu metric (Papineni et al., 2001) for evaluation of machine translation. Bleu is used as a similarity score to identify alignment anchor points, which are then improved by BLEU-based and length-based heuristics. This method has shown to be particularly effective for texts that are otherwise difficult to align.

In addition to the methods mentioned here, a number of variants and combinations of sentence alignment methods have been proposed, including using multiple languages, see Simard (1999), and also Section 6.2. Generally, sentence alignment is carried out with high accuracy. Length-based methods are simple and fast, while linguistically informed methods are more robust. Length-based methods, however, assume that the translation units occur in the same order in the parallel texts, and fail to capture variations that are more than direct pairwise permutations, which can be matched as one block. In the SMULTRON texts, we found a stretch of sentences where one translator had decided to render several of the sentences in a different order (see p. 137). Such variations cannot be handled correctly with length-based sentence alignment algorithms.

6.1.2 Automatic String-to-String Alignment

Word and phrase alignment goes beyond sentence alignment in that it captures sub-sentential correspondences: Which part of a sentence in one language corresponds to which part of a translated sentence in a parallel text? It is, however, not easy to define what a word is (compare with Section 7.2.1). Considering multi-word tokens and, e.g., the different compounding dynamics in languages like English and German (or Swedish) it is obvious that one-to-n token correspondences must be applied. Often a German or Swedish compound noun corresponds to a complex noun phrase in English. For example, in the SMULTRON corpus, the German word *Zertifizierungsverfahren* corresponds to the English three word unit *process of certification*.

Sometimes meaning correspondences hold between larger units, without equivalences on the word level. For example, the co-ordinated German phrase *die Papierindustrie und der Bausektor* corresponds as a whole to *the paper and construction sector*, but we would not want to align *Papierindustrie* to *paper* alone. We capture such correspondences by syntactic phrase alignments.

Syntactic phrase alignment, which is essentially tree-to-tree alignment, will be discussed in Section 6.1.3. In the current section we will discuss string-to-string alignment, which includes word alignment, character alignment and what the statistical machine translation community calls phrase alignment. These phrases are in effect sequences of words of length n appearing together. We will, however, use the word *phrase* in the linguistic sense, as the syntactic level between word and sentence (even though a phrase can consist of only one word or of a whole sentence), and use the term n-gram for a string of words.

Word alignment algorithms usually require sentence-aligned corpora. The main difference between sentence and word alignment is that we can assume that the translation units occur in a similar order at sentence level. This is not the case for word alignment, unless the languages are similar (if at all).

Tiedemann (2003) distinguishes between two broad types of word alignment approaches. He calls them estimation approaches and association approaches. They are also called statistical models and heuristic models by, e.g., Och and Ney (2003). Estimation approaches use probabilities estimated from parallel corpora, inspired from statistical machine translation. Association approaches use string similarity measures, word order heuristics, or co-occurrence measures to see if a cross-language word pair co-occurs more often than could be expected from chance.

In statistical machine translation, translation is modeled by the probabilistic distribution $P(t|s)$, which is the probability of a target string t given a source string s. The most well-known approach to probability estimation for word alignment is the IBM noisy channel model, also called IBM models 1-5, described in Brown et al. (1993). The five unsupervised generative models create a pipeline, where each model is initialized from the parameters of the previous model, finding a local maximum through an EM-algorithm training procedure on a parallel corpus.

Model 1 only uses lexical translation probabilities, while model 2 takes into account the length of the two strings and word order, i.e., the probability of an alignment given the position of the source and target words in their sentence. Model 3 also uses fertility, the number of target words that are generated by each source word, and distortion, the target position given the source position. Model 4 contains a relative distortion model, looking at the positions of other target words aligned to each source word, to account for the tendency of phrases, or chunks of words, to move as units. It also uses word classes, by clustering the vocabulary, to model general structural differences between the languages. Model 5, finally, is similar to model 4, but tries to find a place for target tokens with undefined position, a problem in the previous models.

The models are computationally expensive, and require large amounts of parallel text. Additionally, the alignment is asymmetric, meaning that results

vary depending on the alignment direction. The IBM models have numerous variations and extensions, but it is difficult to add linguistic knowledge to the algorithms. Recently, researchers have shown that a small amount of aligned data improves word alignment (Callison-Burch et al., 2004) and supervised learning techniques have gained interest. The supervised alignment system Lingua-Align (Tiedemann, 2010) will be discussed further in Section 8.5.

6.1.3 Automatic Tree-to-Tree Alignment

With annotation added to a text corpus, it becomes possible to also align these annotations and not just the text itself. Here, we will focus on alignment of the syntactic structure, but one could also imagine alignment for, e.g., morphological or semantic annotation. Tree-to-tree alignment consists of links between non-terminal nodes in source and target tree structures.

As with word alignment, sentence alignment is a prerequisite for tree-to-tree alignment. While one problem for string-to-string alignment is knowing how to define a word, this is generally not a problem for tree-to-tree alignment, where tokenization has been taken care of during the process of annotating the tree structures. Wu (2010) lists a number of issues, which are particular to tree-to-tree alignment. For example, monolingual grammars are rarely robust enough, especially for low-resource languages. There is also often a mismatch between the grammars of different languages (see also the problems for manual alignment described in Chapter 4), and the alignment heuristics may not be able to select the correct link from several possible links.

The idea of aligning constituents was used early on for the 'machine translation by analogy principle' (Nagao, 1984), which is now known as example-based machine translation (EBMT, see Carl and Way, 2003, for a thorough overview of the field). Tree-to-tree alignments are also used for, e.g., machine learning of syntactic transfer patterns for rule-based machine translation.

Many approaches use the crossing constraint (see Wu, 1997, and Section 5.4.2) to ensure that alignments between subtrees do not cross. Most approaches use association features to compute alignment costs, including, e.g., lexical matching between constituents, calculated from bilingual or probabilistic lexicons (e.g., Zhechev and Way, 2008), using word alignments as constraints (e.g., Lavie et al., 2008), and node label matching (e.g., Groves et al., 2004). Tiedemann and Kotzé (2009b) describe a model to combine weighted feature functions, used in the Lingua-Align system (Tiedemann, 2010), which will be discussed further in Section 8.5.

Another approach to tree-to-tree alignment is biparsing, where both the source and the target of a sentence aligned parallel corpus are parsed together. This is done through a generative model called a transduction grammar, and

creates a common, linked, structure for the two texts. One of the main models is Inversion-transduction grammars, or ITGs (see Wu, 1997), which restrict the computational complexity of the task. An even more restrictive model is linear transduction grammars (Saers, 2011), which is less expressive, but more computationally tractable.

6.1.4 Alignment Evaluation

As we saw in Section 4.2.1, evaluation of inter-annotator agreement is not a simple or solved issue. The same is essentially the case for evaluation of alignment. The created alignment needs to be compared to something that is seen as the correct alignment. As pointed out by, e.g., Ahrenberg et al. (2000), this reference data can be either created before hand (thus applying the alignment algorithm to already annotated data), or the output can be evaluated by experts after the alignment process. This distinction is also made by Davis (2002), who, however, discerns between human evaluation and automatic measures. This division is slightly misleading, as both anterior and posterior evaluation requires human intervention. While posterior evaluation requires humans to look at the system output, and thus requires that new data sets are inspected after each system run, we assume human involvement also in creating the gold standard reference data.

The reference alignment, or gold standard, is often an aligned parallel corpus. In the following, we will focus on metrics to compare system output from an aligner to a gold standard corpus. An important issue, however, is that systems evaluated against different gold standards cannot easily be compared, as differences between the gold standards (such as text properties, text length and number of alignments, or different alignment schemes) influence the results. The evaluation metric needs to handle missing or added alignments, different types of alignment, or alignment with confidence-levels, and multi-word tokens, including one-to-many, and possibly many-to-many alignments.

Precision, recall, and the combined f-measure (defined in Section 4.2.1) are some of the most commonly used measures for evaluation of alignment. A problem with these measures (and others) is that word alignments can be partially correct, as multi-word units have multiple links of which one, several or all may be correct (see, e.g., Tiedemann, 2003). While Davis (2002) argues that a word can only be aligned once, it is a matter of perspective if we score what we may call alignments (where word X is aligned to words Y_1 and Y_2) or alignment links (where word X is aligned to word Y_1 and word X is also aligned to word Y_2). We would most likely not think that a link between Swedish *huset* ('house-the') and English *house* is entirely wrong, just because the gold standard has alignment between *huset* and *the* + *house*. On

the other hand, aligning *huset* to *the* would intuitively be incorrect, but distinguishing between the single alignment to *house* and the single alignment to *the* is difficult using automatic evaluation methods.

Och and Ney (2000a) proposed the Alignment Error Rate (AER, described in Section 4.2.1). While this metric is similar to the F-measure, Fraser and Marcu (2007) showed that it does not correlate to machine translation quality, measured with the (highly disputed) BLEU-score (Papineni et al., 2001). They argue that this is because the AER does not penalize unbalanced precision and recall, which makes it possible to achieve a good AER score by favouring precision, e.g., by only making a few high-probability links.

Ahrenberg et al. (1999) defined the PLUG-measures, which categorize alignments as correct, partially correct, incorrect, or missing. Precision and recall were calculated, with partially correct alignments receiving a weight of 0.5. Ahrenberg et al. (2000) and Tiedemann (2003) then proposed the PWA-measures, which incorporate the number of source and target tokens for partially correct alignments, allowing for a more diverse view on partial correctness. However, as discussed in Ahrenberg et al. (2000), the method cannot handle partial alignments in many-to-many alignments correctly. If the gold standard contains, e.g., alignment between *dangerous goods* and the Swedish *farligt gods*, the system could achieve perfect scores with the incorrect alignment between *dangerous* and *gods*, and between *goods* and *farligt* (example from Ahrenberg et al., 2000). Additionally, as pointed out by Davis (2002), allowing for overlap between the number of incorrect alignments and the number of missing alignments (the same word may be counted twice, once as part of a missing alignment and once as part of an incorrect alignment) gives an incorrectly spread probability mass when calculating recall.

Word Alignment Agreement (Davis, 2002, also described in Section 4.2.1) gives every link a weight based on source and target tokens. Davis advocates aligning every word, as in the Blinker project (Melamed, 1998b), linking them either to words in the other language, or to a null-token, thus marking them as unaligned. As was mentioned in Section 4.2.3, this is unmanageable when manually aligning larger data sets, especially when aligning both words and phrases (constituents). Most automatic aligners do not align every word either. It would, however, be possible, for evaluation purposes, to automatically extract these null-alignments, by assuming that an unaligned word or phrase is in fact null-aligned. This means that we do not differentiate between missing alignments and null-alignments (even though some of the latter may in fact be of the former type).

A final metric, which needs to be mentioned here, is the relative error reduction. When comparing two methods to each other, the scores for comparing each method to the gold standard only tell half the story. For example, if

a gold standard contains 10 alignments, and system 1 correctly finds four of them, making two errors, and system 2 finds five of them, making one error, the F_1-score is 0.5 for system 1 and 0.62 for system 2. We then find a method for improving the output of the two systems, resulting in system 1 correctly finding 8 alignments, still making two errors, and system 2 finding 9, still making one error. The F_1-score is now 0.8 for system 1 and 0.9 for system 2. It seems that system 2 always finds one more of the correct alignments, and outputs one less of the incorrect alignments. Comparing the F_1-scores, however, the difference between the improved versions of the two systems decrease, as we get closer to the upper bound.

We therefore calculate the proportional reduction in error (PLE), according to the following formula, where S1 is the annotations by system 1 and S2 the annotations by system 2.

Error reduction $\frac{\text{S1 errors} - \text{S2 errors}}{\text{S1 errors}} \times 100$

The initial version of system 2 shows an error reduction of 25% compared to the initial version of system 1. For the improved versions, however, the error reduction is 50%. While a comparison of F_1-scores for two systems shows smaller differences the better the systems are, the relative error reduction does not.

6.2 Previous Annotation Projection Research

In the following chapters we will explore annotation projection to improve the automatic annotation of a parallel treebank. This section contains an overview of which annotation projection is, and what approaches have been explored in previous research involving annotation projection.

Annotation projection means transferring annotation from one language onto another language. Annotation projection has gained vast interest in the past few years and for a large number of areas in computational linguistics, such as PoS-tagging and morphology, syntactic annotation, semantic annotation, and alignment. Most research has focused on bilingual corpora, but a number of projects are dealing with multilingual data. As the access to multilingual data increases, we can expect more work in this area.

The term was introduced by Yarowsky and Ngai (2001) and Yarowsky et al. (2001), but we can find researchers trying out these ideas earlier than that. Alignment naturally deals with multiple languages, and some of the early attempts at annotation projection dealt with projection of alignment. Simard (1999) carried out multilingual experiments on alignment and first computed bilingual sentence alignments for three languages, to find the most similar pair, and then aligned the sentences of the remaining language with that pair. The

results were as good as or slightly better than using only the bilingual system. Borin (1999, 2000) carried out multilingual experiments on word alignment, what he called pivot alignment, where he computed bilingual alignments for three language pairs and then added links from the indirect path L1-L2-L3 to the alignment of L1-L3. A first small experiment gave an improvement for some languages, and additional experiments confirmed the results. Generally, using pivot languages increased recall without lowering precision.

A number of aspects are important to annotation projection, one of the most important being how information is projected across languages. Many researchers have used raw, or direct, projection from one language to another. Yarowsky and Ngai (2001), Yarowsky et al. (2001) carried out bilingual experiments to induce a PoS-tagger, an NP-bracketer, a named entity tagger, and a morphological analyzer. They annotated the English part of their corpus and directly projected the annotation to the target language, over the word alignment. The result was used as training data for a new system. They found that alignment accuracy was important for the results, which were, however, encouraging. A number of researchers have also applied raw projection to semantic annotation, with positive results, e.g., Bentivogli et al. (2004), Bentivogli and Pianta (2005), Johansson and Nugues (2005).

Hwa et al. (2002a,b) worked on projecting dependency structures from English to Chinese and formulated the direct correspondence assumption (DCA). It states that for two parallel sentences, the syntactic relationships in one language directly map to the syntactic relationships in the other. To evaluate the DCA they developed the direct projection algorithm.

- If there is one-to-one alignment for a source head and its modifier, project the dependency analysis across word alignments to the target words.
- If a source head or modifier is unaligned, while its head or modifier is uniquely aligned, create an empty target word and project the dependency analysis to the aligned and the empty target word.
- If there is a one-to-many relationship, create a new empty word as parent of the target words and align the source word to it.
- If there is a many-to-one relationship, remove alignments from source words, except for the source head.

The evaluation showed that although the algorithm works well, there is no direct mapping between the syntactic dependencies of the source and the target.

Padó (2007), experimenting with projection of FrameNet frames, tried to identify frame parallelism distributionally by observing properties of the translation and its context. He found that while lexical-semantic annotation shows a

high degree of cross-lingual parallelism (more than 80%), syntactic annotation shows a low degree of parallelism (less than 40%).

Possibly as a result of this non-parallelism for syntax, Hwa et al. (2002a,b) found noise reduction in the projected treebank to be a challenge. Additional experiments by Hwa et al. (2005) showed that a small set of linguistically informed, language independent rules to modify the projected structures improved results. They also found that training a parser on projected structures was almost as good as a commercial parser, and better if poor quality trees were pruned. The parser trained on projected trees had the same accuracy regardless of the number of training sentences, while a parser trained on human annotation outperformed the induced parser when there were more than 2500 training sentences.

Bouma et al. (2008) worked on multilingual argument identification and raw consensus projection, using the result as data for machine learning. They found that consensus projection was better than single source projection for induction of target language classifiers, but that a small set of manually annotated data was better for induction of target language classifiers than consensus projection. Fossum and Abney (2005), in experiments on bilingual and multilingual PoS-tagging, found minor differences between selecting the majority tag (the tag picked by most taggers) and selecting the tag through linear combination, where they calculated the linear combination of the vectors over possible tags from each tagger. Riloff et al. (2002), building a bilingual information extraction (IE) system for English and French, found that they received the best precision by applying machine learning on the original source text, projecting the annotations to the target text and then using the projected target data to train a new system. The best recall was achieved by applying machine learning on the machine translated source text and then using raw projection onto the target text.

Spreyer (2011) explored, among other things, training a parser on projected dependency tree fragments. She found that parsers trained on tree fragments can perform as well as parsers trained on trees. This allows for projecting the knowledge with the highest probability, to be used as partial training data.

An alternative to annotation projection was explored by Kuhn and Jellinghaus (2006). They manually annotated a small section (about 100 sentences) of a large multilingual corpus with flat bracketing and phrase alignments. The set was then used as seed for bootstrapping phrase correspondence patterns for the whole corpus. Based on statistical word alignment they trained monolingual parsers over the consensus representation. Their initial experiments seemed promising, as the results improved with the bootstrapping approach.

A second important aspect of annotation projection is the number of languages used as source languages, where it seems that the more languages to

project from, the better the results (see, e.g., Borin, 1999, 2000, Fossum and Abney, 2005, Yarowsky and Ngai, 2001, Yarowsky et al., 2001). Results also indicate that the relationship between the source language or languages and the target language is of importance for the results (see, e.g., Borin, 1999, 2000, Mitkov and Barbu, 2004), although Fossum and Abney (2005) found that multiple source languages improved the induced PoS-tagger, possibly regardless of the language relation.

An third point is that annotation projection can be used for different reasons. Snyder and Barzilay (2008), working on morphological segmentation, list a number of possible scenarios for the transfer of knowledge.

- Indirect supervision, without supervised target data, but with annotated resources for source languages (classical projection)
- Augmentation, where supervised target data is available but can be supplemented by unsupervised data in additional languages
- Shared supervision, where supervised data is available for both languages

While this list is not exhaustive, it is clear that the amount of information projected differs if we want to augment target annotation, focusing on problem areas, or induce the whole annotation from the beginning. Additionally, as pointed out by, e.g., Spreyer (2007) and Bouma et al. (2008), training a labeller on the projected annotations helps us to abstract away from the parallel corpus. The projected annotation should not be the final step. Different researchers have, however, found different levels for how much manually annotated training data is needed to outperform such a labeller.

In our experiments, we will use annotation projection to improve the automatically created annotation, thus using it for shared supervision, or possibly augmentation. In Chapter 8, we explore projection for alignment, and in Chapter 9 for syntactic attachment.

6.3 Summary

This chapter has presented overviews of several areas of research, starting with automatic alignment methods, continuing with metrics for evaluation of alignment, and finally annotation projection. We have given an overview of methods for automatic alignment at sentence and word level, as well as phrase level in annotated constituency trees. We will return to some of these methods, for the experiments with several alignment systems to automatically create alignment for the SMULTRON data.

Annotation projection, transferring annotation between languages, is an area of research that is growing. A variety of annotation types have been projected from one or multiple languages onto another language. Important issues are projection method, the number of source languages, and the relationship between source and target languages. While raw majority vote projection has been the most common method, it may be too blunt a tool, due to, e.g., structural differences between languages, and differences in translation choice by translators, as well as possible annotation errors in the source annotation. Incorporating some linguistic knowledge or training an annotation system on the projected information seems to improve results.

Following this overview, in Chapter 7 we describe the data preparation necessary to carry out further experiments. We then turn to our experiments on automatic alignment in Chapter 8, which include experiments on improving the automatic alignment of the aligners through annotation projection. In Chapter 9, finally, we try to improve the syntactic attachment decisions of the parsers through annotation projection. We will evaluate both majority vote projection, as well as linguistically grounded clues to select the structure to project.

7. Automatically Tagging and Parsing SMULTRON

In the following chapters, we will investigate multilingual annotation projection. To be able to evaluate the impact of projection properly, we need data that has been automatically annotated, and a gold standard to compare the projected annotations to. We will use SMULTRON as our gold standard. This chapter describes how we prepared the automatically annotated data for the experiments. The texts of SMULTRON are automatically tagged and parsed (Sections 7.1 and 7.2) and the automatically annotated data is then compared to the SMULTRON gold standard (Section 7.3).

7.1 Tagging and Parsing English and German

For languages like English and German there are resources available and thus a number of taggers and parsers are obtainable. We used the pre-trained versions of the Stanford parser for both English and German. The English factored parser (Klein and Manning, 2003) is a widely used lexicalized probabilistic parser, trained on the Penn treebank (Marcus et al., 1993). The German lexicalized probabilistic parser (Rafferty and Manning, 2008) is trained on the NEGRA corpus (Skut et al., 1997).

We had to make a number of simplifying assumptions for the automatic annotation. For example, the parse evaluation tools require that the texts have the same sentence boundaries and each sentence the same number of tokens. To achieve this, the English, German, and Swedish parsers were fed the text in such a way, that the sentence boundaries were already given. In addition, the economy part of SMULTRON contains some HTML to mark, e.g., headings. Since this was problematic for the parsers to handle, all HTML was removed from the texts, for all three languages.

The Stanford parser accepts texts where the sentence boundaries are predefined, marked by newline in the input data. This way the sentence boundaries were preserved, compared to SMULTRON. Unfortunately, the parser could not handle one English sentence, from the economy part of the treebank, which contains 152 words. This English sentence is therefore left out from the gold

standard in the evaluation. The output of the parsers consists of tree structures in Penn bracketing format, which was transformed into TIGER-XML through TIGERSearch.

7.2 Tagging and Parsing Swedish

As mentioned in Section 3.2, no suitable Swedish treebank existed for training a Swedish parser when we started constructing SMULTRON. More recently, the Svannotate parser for Swedish has become available, which creates phrase structures with functional edge labels, as in the Swedish Treebank.[1] Svannotate is a pipeline consisting of a tokenizer and sentence splitter (not used for this data), a tagging model for the Hunpos tagger (Halácsy et al., 2007) trained (Megyesi, 2009) on the SUC corpus (Ejerhed et al., 1992), and a parsing model for the MaltParser (Nivre et al., 2007) trained on the cleaned and converted Talbanken corpus (Teleman, 1974).[2] The texts were fed to the tagger and parser with the same sentence splitting as SMULTRON.

Initially, however, we used a cruder parsing method. We applied a tagger and chunker for Swedish, ran the tagged text through a dependency parser, and finally combined the chunks with the dependencies. To generate the PoS-tagged and shallow parsed output, we used the Granska TextAnalysator (GTA) (Knutsson et al., 2003).[3] The texts with only the PoS-tags were converted into Malt-TAB[4] and the texts with tags and chunks were converted into TIGER-XML (Mengel and Lezius, 2000). The GTA parse consists of flat chunks. We therefore inserted one additional phrase, with the label *S* (sentence), for each sentence. All chunks were attached to this phrase, and we thus had a sentence spanning tree.

The Malt-TAB files, containing the tagged texts, were run through the MaltParser pre-trained on Talbanken[5] and the output was converted into TIGER-XML with the help of MaltConverter.[6] Finally, the GTA chunks and Malt dependencies were combined into one structure. This was done in the following way.

[1] See http://stp.ling.uu.se/~nivre/swedish_treebank/.
[2] See http://w3.msi.vxu.se/~nivre/research/talbanken.html.
[3] A simple web interface is available at http://skrutten.nada.kth.se/grim/form.html.
[4] See http://w3.msi.vxu.se/~nivre/research/MaltXML.html.
[5] See http://stp.ling.uu.se/~nivre/step.html.
[6] See http://w3.msi.vxu.se/~nivre/research/MaltXML.html.

1. For each GTA phrase, if one word dominates all the other words in the phrase, according to the Malt parse,
   - give that word the edge label 'HD' (head),
   - give the rest of the words edge labels according to the Malt parse's relation labels.
2. If there is no one word dominating the other words in the phrase, give all words the edge label '–'.
3. For each GTA phrase, replace the attachment to the dummy S with attachment to the head's parent phrase.
4. For unattached nodes:
   - unattached words (according to the Malt parse) are attached to their GTA phrase with the edge label '–',
   - unattached phrases are attached to the root with the edge label '–'.
5. For circular dependencies, ignore the intermediary dependency and attach the grandchild to the grandparent.

The two final points handle unattached items and conflicts between the GTA and the MaltParser output. While unattached words mainly appear when the parser does not know how to attach them, some are a product of the process of combining the chunks and the dependencies. The MaltParser sometimes assigns several head words to a sentence, which it solves by using a dummy root node to join these multiple heads under one top node. In the conversion from Malt-TAB to TIGER-XML only one head is chosen, by default the first. To handle the resulting unattached words, the combination program checks if the root node has any children. If not, we select the word which has the most children as the root node.

Circular dependencies appear when there is a conflict between the phrases and the dependencies. One example is the (partial) sentence (with token numbers as subscripts) in Example 7.1.

(7.1) Finns$_1$ det$_2$ då$_3$ något$_4$ som$_5$ alla$_6$ borde$_7$ [vara . . .]

Is there something that everyone should [be . . .]

According to the shallow GTA parse, as shown in Figure 7.1 (where we ignore words 2 and 3 for clarity), words 4-7 are children of the same phrase (504). The MaltParser dependencies, shown in Figure 7.2, mark words 5 and 6 as dependents of word 7. Word 4 is a dependent of word 1, which in turn is a

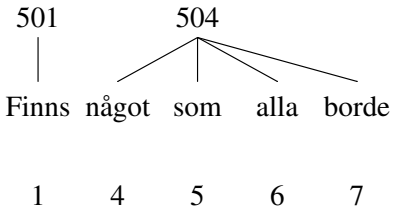


Figure 7.1: An example of a GTA structure.

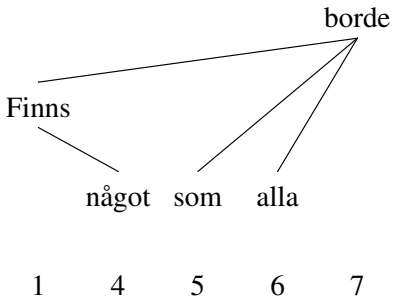


Figure 7.2: An example of a MaltParser structure.

dependent of word 7. Thus, combining them results in a tree where 504 is dominated by phrase 501, which is dominated by phrase 504. We solved this problem by looping upwards for phrases. If a grandparent is the same as the current phrase we do not save the dependency between the current phrase and the parent. In the example, word 1 is dominated by phrase 501, and phrase 501 is a sibling of words 4-7, all children of phrase 504. We thus let the phrase information take precedence over the dependency information in such cases.

The tagset used by the GTA is slightly different from the SMULTRON tagset, although both are altered versions of the SUC tagset. For the evaluation, the GTA tags were converted into the SMULTRON tags. This conversion is rather straightforward, since the SMULTRON tagset is slightly smaller.

One problem, however, is that both GTA and Svannotate have a different word tokenization than the SMULTRON treebank. This needs to be handled, for the number of tokens to be the same between the automatically parsed data and the gold standard in the evaluation. These problems are discussed in Section 7.2.1.

7.2.1 Re-tokenization of Swedish

Tokenization is an important issue in corpus work, which unfortunately is often treated as a solved and unproblematic subject. However, many problems arise when combining different tools, or comparing corpora where different tools have been used. He and Kayaalp (2006), for example, compare 13 tokenizers for English and get 13 different results.

For the current work, one of the main problems was that the evaluation methods rely on the number of sentences and the number of tokens per sen-

tence being equal for system and gold data. One additional reason for wanting to harmonize tokenization between SMULTRON and the automatically parsed data is that we would like tokenization for a particular language to be as similar as possible between different corpora. This is helpful when building new or larger corpora for a particular language (see, e.g., Nivre and Megyesi, 2007).

For all three languages, there were initially differences in sentence boundaries between the gold standard SMULTRON treebank and the automatically parsed data. To solve this problem, as mentioned, the sentences were fed to the parser one by one, according to the sentence boundaries of SMULTRON. This was unproblematic for the English and German data, as was the subsequent tokenization.

The Svannotate parser, however, did not fully accept the sentence splitting of the input. In one case this was understandable, due to sentence final punctuation used within a sentence unit (*Vov-vov! Vov-vov!*). While this is handled as one unit in SMULTRON, it could just as well be seen as two distinct sentences, as Svannotate did. In other cases, Svannotate changed the sentence boundary when there was a colon, which the parser apparently viewed as sentence final. In three instances, the punctuation of an abbreviation was tokenized separately, for *Inc .*, *U.S .*, and the middle initial of a name. This led the parser to initiate a new sentence after the punctuation. The GTA also had problems recognizing the punctuation of *Inc.* as part of the abbreviation, but did not initiate a new sentence. For the Svannotate data, these sentence were joined again, without regard for the resulting structure. The top nodes, each contained in a ROOT node, were simply attached to the same ROOT node.

At word level, the English and German versions of the Stanford parser apply the same rules for tokenization as we have done in the SMULTRON treebanks. This means that they generally see a graph word as one token, splitting multi word units into several tokens. Additionally, the English genitive *'s* is separated from its noun. For Swedish, the SUC corpus (Ejerhed et al., 1992), is generally viewed as a standard for Swedish annotation. They have a rather pragmatic view of tokenization. "The guiding principle for tokenizing should be to retain as close a correspondence as possible between a graph word, a sequence of characters surrounded by spaces, and a token." (Ejerhed et al., 1992, p. 3). We have mostly followed the SUC tokenization. The Svannotate parser, which allows for tokenized data as input, only had minor differences. The GTA tagger, however, is based on a different approach to tokenization. For example, it treats many multi-word units as one token, thus ignoring a level of analysis. Below, we will discuss a number of the problematic cases concerning tokenization, and how they were handled.

To get the same number of tokens for the gold standard and automatically parsed Swedish data, the parser output was post-edited, joining some tokens

and splitting others. For the Sophie part of the data, we changed 4 sentence boundaries and 1 token in the Svannotate version, and 29 tokens (14 different types) in the GTA-Malt version. The most tokenization differences were found in the economy part, where 4 sentence boundaries and 14 tokens (13 types) were changed in the Svannotate version, and 173 tokens (102 types) in the GTA-Malt version. Additionally, three tokens were changed in the economy part of the gold standard treebank.

Elliptical Compounds

In a coordination between compounds, where a part of these words is identical, this part does not need to be repeated, but is omitted and replaced by a hyphen. This omission is possible for the former part, as in *problemformuleringar och -lösningar* ('problem formulations and solutions'), as well as for the latter part of compounds, as in *kapital-, risk- och finansieringsfrågor* ('capital, risk, and financing issues').

The first type, where the first part of a second compound is omitted, only appears once in the Swedish SMULTRON data, consisting of the expression *lågspänningsbrytare och -omkopplare* ('low-voltage breakers and switches'). Both GTA and Svannotate have split the hyphen from the noun *omkopplare*. The GTA has given the hyphen a DL-tag (delimiter) and made it the only child of a mid-phrase (minor delimiter), while the noun has the NN-tag and is the child, but not the only one, of an NP. Svannotate has made the noun the single child of an NP, which in turn is the child, together with the MID-tagged hyphen, of an NP containing the coordination. The re-tokenizer joins the two tokens, using the PoS-tag and attachment of the noun.

The second type of elliptical compounds, where a latter element is omitted, is common in the SMULTRON data, especially in the economy texts. These cases have been consistently treated as multi-word tokens by the GTA, which does not give a complete linguistic analysis. For example, the first element may have a different PoS than the whole multi-word unit, as in *höger-/JJ och/KN vänsterhalva/NN* ('right and left half') or *vad-/HP och/KN hur-frågorna/NN* ('the what and how questions'). In the SUC corpus, there are 30 different tag combinations for the first and second compounds in a coordination where the latter part of the first compound has been omitted. In accordance with SMULTRON, and SUC, we want to analyze the finer parts of these noun phrases.

Elliptical compounds are retokenized and receive the tag originally given to the whole expression (generally NN) for all elements except the conjunction. There may be instances in the data where the two (or more) coordinated items should not have the same tag, but there is no way of knowing from the data we have. A more elaborate solution would be to re-employ a tagger.

Splitting Tokens

A number of other multi-word expressions have been problematic. They can be categorized as either multi-word functional expressions, which with a few exceptions are tokenized as multi-word token adverbs by GTA, or expressions containing numbers. The Sophie data almost exclusively contains the former, while all types of tokenization differences appear in the economy data.

Most multi-word functional expressions are tokenized and tagged as one token by GTA, which requires that they are split and retagged to adhere to the SMULTRON (as well as SUC and Svannotate) tokenization. Automatically splitting one token into several is not a major problem. The new tokens are given new id numbers, requiring a renumbering of the rest of the sentence. The new tokens then receive new PoS-tags and the phrases of the sentence are processed, to point to the right tokens. For example, the expression *framför allt* ('first and foremost', literally 'in front of all') has been tagged as two tokens, with a preposition-pronoun tag sequence in both SUC and SMULTRON, as well as by Svannotate. The GTA has analyzed the whole expression as one token, tagged as adverb. The re-tokenizer splits the GTA token into two, with the tags preposition and pronoun. Similar problems occur for expressions like *bland annat* ('amongst other things'), *med andra ord* ('in other words'), *till exempel* ('for example'), and *från och med* ('from', literally 'from and with').

The Swedish expression *till och med* can have two different meanings, once translated into English as *until* and once as *even*. In the first case, the words (except the conjunction) should be tagged as prepositions, in the other as adverbs, according to SUC. Interestingly enough, the first reading is found in the economy texts (three occurrences), while the other reading is found in the Sophie text (one occurrence). Since the prepositional reading is the more common one, and since we would probably first consider the words *till* ('to') and *med* ('with') as prepositions, if taken out of context, this tagging was chosen for the re-tokenization program. This also meant that it could be handled in the same way as *från och med*, which only has one reading.

The expression *och/eller* ('and/or') is the only case where the SMULTRON tokenization deviates from the SUC tokenization (it only occurs once in the SMULTRON data). Both SUC, Svannotate and GTA see it as three tokens, tagged as two conjunctions separated by a delimiter. This is an issue that should be changed for future versions of SMULTRON, and we thus changed the gold standard for the evaluation.

Some expressions with numbers also required splitting, such as dates, where SMULTRON and SUC tag them, e.g., as *den/DT 1/RG januari/NN 2001/RG* (where RG is cardinal number). GTA sees them as multi-word tokens, e.g., *den 1 januari 2001/RG*. Additionally, time expressions were problematic. Swedish

time expressions are often accompanied by the word *klockan* ('clock'), possibly abbreviated (*kl.*). GTA keeps time expressions like these as one token, e.g., *kl 10:00*, tagged as a cardinal number. SMULTRON analyzes them as an adverb followed by a cardinal. SUC has a similar view, but tags the abbreviated form as an adverb and the unabbreviated form as a noun. The GTA tokens were split and retagged according to the SMULTRON analysis.

Joining Tokens

While splitting tokens is mostly straightforward, it is more problematic to join tokens, especially since these tokens already have a phrase structure built on top of them. In most cases, for GTA, this problem was avoided by re-tokenization of the GTA data, followed by re-running the MaltParser, thus only having to take care of the PoS-tag of the new token and the attachment to the immediate parent phrase.

Luckily, only a few expressions required that tokens were joined. For example, there is one instance of *c/o* in SMULTRON, analyzed as one token, tagged as a preposition. Both the GTA and Svannotate have split the expression into three tokens. Svannotate has analyzed it as a proper noun-minor delimiter-proper noun sequence, all children of a larger NP. They were joined into one token, keeping the attachment to the NP. GTA has viewed it as a noun-delimiter-noun sequence, where each of the three tokens belong to different phrases. MaltParser has analyzed the tokens as a chain of dependencies. The three parts were joined into one token, with the tag PR. The token is then attached to the highest of the original phrases.

Percentages are generally expressed with a cardinal number followed by the string *procent* or the percentage sign, as two (or more) distinct words. While SUC and Svannotate treat the two words as separate tokens, tagging them as a cardinal and a noun, GTA sees the whole expression as one token, a cardinal. Unfortunately, in SMULTRON, in the two instances of a number followed by the percentage sign, the number and the sign are considered one token, a cardinal number. The same analysis appears in the GTA tagged data. Even though one would like to analyze *100%* in a way similar to *100 procent* (a cardinal and a noun), the fact is that they were one graph word in the original text. Thus, in violation of the SUC data, the Svannotate tokenization was changed, joining the number and the sign and tagging the new token as a cardinal. One additional issue is the fact that the GTA, for some reason, has split *procentig* into two tokens, 'procent' and 'ig'. Thus we have '100 procent/RG' and 'ig/NN'. These are joined, and *procentig* given the tag JJ.

One final problem is telephone numbers, which consist of a sequence of digits sometimes separated by blanks or apostrophes. In the economy texts in

SMULTRON, some telephone numbers are preceded by a '+', indicating outgoing country code. In SMULTRON this is considered part of the telephone number, while GTA has treated it as a separate token. The same applies to a few instances of a code that is to be dialed on a telephone, consisting of three digits and the '#'-sign. One could argue for both treatments, but in accordance with the SMULTRON analysis, the whole sequence is seen as one token. Unfortunately, one instance of a telephone number in SMULTRON has been tokenized the wrong way, and then mistagged. Part of the number, tagged as a cardinal number, has been separated from the rest, tagged as a noun. This should be corrected in a future version of SMULTRON and we thus changed this in the gold standard for the evaluation.

7.3 Parse Evaluation

Above, we have described the automatic annotation of the SMULTRON texts. We automatically tagged and parsed the texts as a preparation for experiments on information transfer from other languages. While this projection of annotation will be evaluated on a test collection, rather than the whole corpus (see Chapter 9), we still need to evaluate the over-all correctness of the parser output.

Parsers can be evaluated along several dimensions (Kakkonen, 2007): preciseness (which we will call accuracy), coverage, robustness, efficiency, and subtlety (the level of detail in the annotation). Here, we are mainly interested in accuracy, specifically how close the automatically produced parse structure is to the gold standard.

A number of different evaluation methods for parsers have been devised during the years (see, e.g., Carroll et al., 1998). Many of these methods, just like the parsers they were used to evaluate, have been developed for phrase structures of the Penn Treebank type. A script evaluating pure constituent structure, operating on Penn bracketing format files, only considers the span of the phrase and the phrase label. Edge labels are not handled by the Penn format. However, we want to evaluate phrase structure trees with functional edge labels and crossing branches. We will briefly discuss some of the most widely used evaluation methods: Parseval, Leaf-Ancestor and dependency-based evaluation. This is followed by an evaluation of the automatic annotation of SMULTRON.

7.3.1 Evaluation Methods for Parsing

The Parseval metric (Black et al., 1991) calculates precision and recall (defined in Section 4.2.1) over constituents. Labelled Parseval sees a constituent as

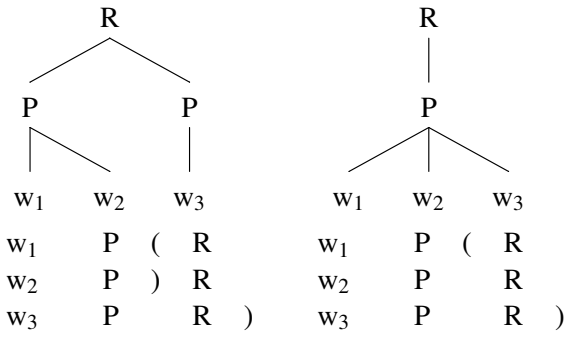


Figure 7.3: Example gold standard and automatically parsed trees and their Leaf-Ancestor lineages.

correct if a constituent in the gold standard dominates the same sequence of terminals (has the same bracketing) with the same labels (PoS and syntactic labels). In general, the Parseval-scripts are used for English Penn Treebank annotation types. Language specific items like auxiliaries, *not*, pre-infinitival *to*, and possessive endings, as well as word-external punctuation, are removed from the fully-parsed sentence. Additionally, empty brackets and unary nodes are removed, and then the result is compared to "a similarly reduced version of the Penn Treebank parse of the same sentence" (Black et al., 1991). While the metric has been criticised, see e.g., Carroll et al. (1998), Rehbein and van Genabith (2007), it is still widely used. We use a script provided by Johan Hall, for Parseval evaluation on TIGER trees with crossing branches.

The Leaf-ancestor metric (Sampson, 2000, Sampson and Babarczy, 2003) assigns a score to every word in a test sentence by comparing the lineage (the sequence of non-terminals from a word up to the root node) of the word in the parser output tree to the lineage of the same word in the gold tree, using a Levenshtein or edit-distance. To distinguish between the lineages of different phrase structures, Sampson and Babarczy (2003) add markers to the left-most and right-most child of a branching node in the lineage. These left-most '(' and right-most ')' markers are inserted once for each terminal, at the top-most node that the child is the left-most or right-most child of. This is only done for nodes that have multiple children, i.e., not for unary nodes. Figure 7.3 shows two example trees and their lineages, for a three word sentence. The difference between the two trees is where the markers are inserted.

The Leaf-Ancestor metric was not developed for trees containing crossing branches. In Figure 7.4, we see two trees, which the Leaf-Ancestor evaluation

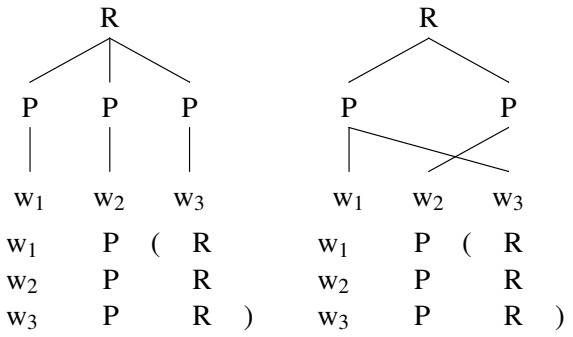
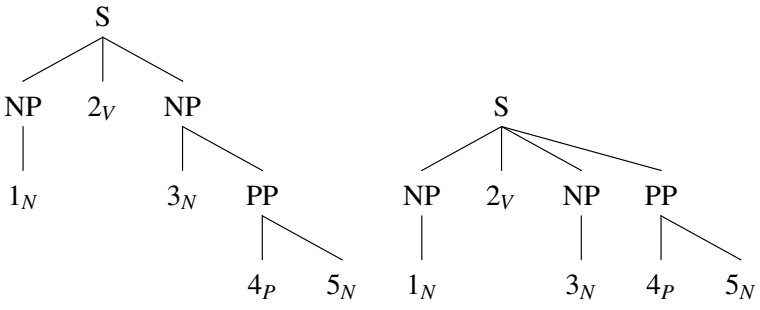


Figure 7.4: Example trees and their Leaf-Ancestor lineages, showing that the Leaf-Ancestor metric cannot properly handle trees with crossing branches.

considers to be identical. Researchers using the Leaf-Ancestor metric on treebanks containing crossing branches generally converting the trees by resolving the crossing branches (e.g., Rehbein and van Genabith, 2007).

We created a partial solution to circumvent this problem. We simply add markers for all left-most and right-most children, and even for unary nodes. While this lets us discern between most trees with crossing branches, the main problem is that it gives, e.g., attachment errors too much weight. An example of this can be seen in Figure 7.5, where the sentence score for the sentence with noun attachment (on the left), compared to the sentence with verb attachment (on the right) is 0.7 $(1-3/(5+5))$ for the classical version of the Leaf-Ancestor metric. The same example gets a score of 0.6 $(1-4/(5+5))$ using the version of the metric that is adapted to handle crossing branches.

Dependency evaluation for constituent structures has been advocated by a number of researchers (e.g., Lin, 1998, Rehbein and van Genabith, 2007). The method relies upon automatically extracting dependency relations from the constituent structure. There have been several attempts at automatically converting phrase structures into dependency structures (e.g., Johansson and Nugues, 2007). This allows for using evaluation metrics for dependency structures, calculating the labelled attachment score (LAS, the percentage of words with the correct head and dependency label), the unlabelled attachment score (UAS, the percentage of words assigned the correct head), and the labelled accuracy score (LAcc, the percentage of words with the correct dependency label). The major drawback of this evaluation method is that it entails converting the trees into a different format. In part, the evaluation may thus reflect the conversion process, rather than the parser accuracy.



Classical Leaf-Ancestor

1_N	NP	(	S		1_N	NP	(	S	0	
2_V	S				2_V	S			0	
3_N	(	NP	S		3_N	NP	S		1	
4_P	(	PP	NP	S	4_P	(	PP	S	1	
5_N	PP	NP	S	)	5_N	PP	S	)	1	
									0.7	

Adapted Leaf-Ancestor

1_N	(	NP	(	S		1_N	(	NP	(	S	0	
2_V	S					2_V	S				0	
3_N	(	NP	S			3_N	(	NP	)	S	1	
4_P	(	PP	NP	S		4_P	(	PP	S		1	
5_N	PP	)	NP	)	S)	5_N	PP	)	S	)	2	
											0.6	

Figure 7.5: The Leaf-Ancestor lineages for a prepositional phrase with noun (left) and verb (right) attachment, for the classical Leaf-Ancestor metric and the adapted version. Attachment errors are punished harder by the adapted metric.

7.3.2 Adapting the Data for Evaluation

As discussed in Section 7.2.1, we had to edit sentence splitting and tokenization. The resulting versions of the automatically parsed SMULTRON texts are considered the basic versions in the evaluation. Next, while the structures of the German and Swedish SMULTRON treebanks have been deepened (see Section 3.2.2), the automatic parses have not. Thus, we also evaluate the automatic parses against a flat version of the SMULTRON trees. Finally, the annotation schemes of the Swedish automatic parses differ from the one used for the SMULTRON annotation. We therefore mapped the labels in the automatic parses to the gold standard labels.

No PoS-tags were mapped for the GTA-Malt trees. For the Svannotate trees, we changed the PP tag of prepositions into PR, and mapped the three tags for delimiters (sentence inner, sentence final, and parentheses) to one delimiter tag. For phrase and edge labels, however, several of the mappings are problematic, as there is no one-to-one correspondence. For example, there are a number of GTA-Malt NP phrase labels, such as NPjj, which are mostly translated into NPs, but sometimes correspond to more than one label. The label NNjj is used, e.g., for ordinal numbers and phrases consisting of a determiner followed by an adjective. While the latter is treated as an NP in SMULTRON, the former is labelled AP. The Svannotate HD (Head) phrase label is used for a wide variety of unary nodes, such as verbs, adverbs, and even nouns. Some of the problematic labels are converted to the closest matching label, while widely varying labels are left untranslated.

Additionally, there are many phrases in the GTA-Malt tree with lower-case labels. These are mostly unary nodes, where the phrase label equals the PoS-label of their child. One possibility, which we have not explored in the evaluation, would be to remove all phrases with lower-case labels. At the moment, they are left untranslated.

To conclude, we have four different combinations to evaluate for Swedish: a basic version of the automatically parsed data and one with the mapped labels, which are compared to a flat and a deepened version of the gold standard. We call them basic, adapted, flat, and flat adapted. For German, we have two combinations, the basic version of the automatically parsed data, compared to a flat and a deepened version of the gold standard. For English we only evaluate a basic version.

7.3.3 Evaluation of PoS-tagging

First, we evaluate the accuracy of PoS-tagging. The results can be found in Table 7.1. Accuracy is calculated as the percentage of the number of correct tags divided by the number of tokens. We do not report PoS accuracy for the flat

		DE	EN	SVG	SVS
Sophie	basic	94.7	93.7	96.0	56.7
	adapted	-	-	-	77.1
Economy	basic	89.8	93.7	94.6	61.7
	adapted	-	-	-	86.3

Table 7.1: Accuracy for the automatic PoS-tagging for German, English, and Swedish (where SVG is GTA-Malt, and SVS is Svannotate), compared to the SMULTRON gold standard.

tree versions, since the difference between a flat and a deepened tree does not affect the word level. As we can see the adaptation of PoS-labels gave a major improvement for the Swedish Svannotate parse. However, we can also see that the transformation of the Svannotate tags is far from perfect. The accuracy for the Svannotate PoS-tags increases from 55-60 for the unadapted labels, to 75-85 for the adapted tags, compared to around 95 for the other taggers. While the accuracy for the English Sophie data is similar to the economy data, the accuracy for the economy data is worse for both the German and the Swedish GTA-Malt parses, while better for the Swedish Svannotate parse. The Swedish GTA-Malt parse has the highest PoS accuracies, around 95-96.

Error analysis for the PoS-tags shows that one of the main sources of errors for the German automatic parse is punctuation, and in particular parentheses. There are 6 parentheses (3 opening and 3 closing) in the Sophie part, and 162 in the economy part, which have been mostly tagged as cardinals, but range from verbs to foreign material and prepositions. Correcting the tags for only the parentheses would mean an improvement in accuracy for the economy part from 89.8 to 91.3. Unfortunately, the mistagged parentheses have in some cases led to other tagging errors in the sentence.

One of the main problems with the Svannotate parse is verbs. The parser has one tag for all verbs, while they are categorized as finite, non-finite, imperative or supine in the Swedish part of SMULTRON. Disregarding this difference, counting all words that have been tagged as verbs and are some type of verb in the gold standard as correct, accuracy goes from 77.1 to 95.1 for the Sophie part, and from 86.3 to 96.0 for the economy part. Then only 1-2% of the verbs have been mistagged.

The German parser has mistagged 4-5% of the verbs, but less than 2% with non-verbal tags. The English parser has mistagged 6-7% of the verbs, less than 2% of the verbs in the Sophie part with non-verbal tags, but close to 5% of the

verbs in the economy part with non-verbal tags. The Swedish GTA-Malt parse contains the least errors for verbs, around 1% for the economy part and only 0.3% (4 out of 1340 verbs) for the Sophie part. Two of the mistagged Sophie verbs are supine verbs, which have been tagged as adjectives, one of which is an adjective that has been mistagged in the gold standard (*vem kortet var adresserat till*, 'who the card was addressed to').

Another problem, relevant for the economy part of the German and Swedish treebanks, is the large amount of foreign (English) words used in the texts. For German, 233 tokens have been correctly (according to the gold standard) assigned the PoS-tag FM, while 361 have been mistagged, mostly as nouns or names. The Swedish Svannotate parser has correctly (again, according to the gold standard) assigned the tag UO to 11 tokens, which leaves 136 (i.e., all) GTA-Malt and 125 Svannotate tokens, mostly mistagged as names. The tag is not entirely unproblematic, as it is sometimes hard to decide what is a foreign word, and what has been incorporated into the language, as well as what is a name. For example, consider the noun phrase *företagets amerikanska dotterföretag Combustion Engineering (CE) och ABB Lummus Global Inc.* ('the company's american daughter companies CE and ABB LG Inc.'). In the gold standard, all tokens of the latter company's name (including *Inc.*) are tagged as names, while the tokens of the former company's name have all been tagged as foreign words (apart from the abbreviation in parentheses).

7.3.4 Evaluation of Parse Trees

Next, we evaluate the syntactic structures created by the parsers. We report the Leaf-Ancestor score, as well as unlabelled and labelled precision, recall, and F_1 (multiplied by 100) of the Parseval evaluation. For the Swedish parses, we additionally report scores for edge labels. The German and English automatic parses do not have edge labels.

Evaluation of the German trees is shown in Table 7.2. We achieve the highest scores when comparing the automatic parse to the flat version of the SMULTRON gold standard, with an average leaf-ancestor score close to 85, and an average unlabelled F_1-score of around 50. The difference between the basic comparison and the comparison to the flat gold standard comes from a major increase in recall. This is expected, as the automatic German parse is closer to the undeepened version of the gold standard. There is only a small decrease between the unlabelled and the labelled scores, which means that most of the correctly found phrases also have the correct label. The parser seems robust across genres, as the scores for the Sophie and the economy data are similar.

For English, shown in Table 7.3, the average Leaf-Ancestor score is just below 90, and the average unlabelled F_1-score is around 80. The decrease

Metric	Sophie's World		Economy Texts	
	Basic	Flat	Basic	Flat
Leaf-Ancestor phrase	76.2	**84.4**	76.2	**82.8**
Unlabelled precision	**45.3**	42.3	**46.1**	45.5
Unlabelled recall	37.9	**60.8**	37.3	**55.8**
Unlabelled F_1	41.2	**49.9**	41.2	**50.1**
Labelled phrase precision	**43.4**	40.6	**43.4**	42.4
Labelled phrase recall	36.3	**58.4**	35.1	**52.0**
Labelled phrase F_1	39.6	**47.9**	38.8	**46.7**

Table 7.2: Evaluation of the automatic German parses compared to the SMULTRON gold standard.

Metric	Sophie's World	Economy Texts
Leaf-Ancestor phrase	89.1	88.6
Unlabelled precision	78.7	75.7
Unlabelled recall	85.5	80.3
Unlabelled F_1	82.0	77.9
Labelled phrase precision	75.6	73.8
Labelled phrase recall	82.2	78.3
Labelled phrase F_1	78.8	75.9

Table 7.3: Evaluation of the automatic English parses compared to the SMULTRON gold standard.

Metric	GTA-Malt			Svannotate		
	B	A	FA	B	A	FA
Leaf-Ancestor phrase	21.6	**61.2**	58.3	**74.5**	**74.5**	62.7
Leaf-Ancestor edge	32.4	**48.9**	43.2	31.4	**58.1**	44.8
Unlabelled precision	**40.0**	**40.0**	18.9	**53.8**	**53.8**	21.1
Unlabelled recall	**38.0**	**38.0**	32.5	**64.2**	**64.2**	45.6
Unlabelled F_1	39.0	39.0	23.9	58.5	58.5	28.8
Labelled phrase precision	8.3	**31.9**	13.9	**51.6**	**51.6**	19.2
Labelled phrase recall	7.9	**30.4**	23.9	**61.6**	**61.6**	41.6
Labelled phrase F_1	8.1	31.1	17.6	56.2	56.2	26.3
Labelled edge precision	0.1	**23.1**	9.5	1.9	**28.7**	5.8
Labelled edge recall	0.1	**21.9**	16.3	2.2	**34.3**	12.6
Labelled edge F_1	0.1	22.5	12.0	2.0	31.3	8.0

Table 7.4: Evaluation of the Sophie's World texts, comparing the automatic Swedish parses to the SMULTRON gold standard. (The evaluated data sets are the basic parse (*B*), the parse with adapted labels (*A*), and the parse with adapted labels compared to the flat gold parse (*FA*).)

between the unlabelled and the labelled scores is again small, but the difference between the Sophie and the economy part is larger than for the German parses, with the Sophie data achieving the higher scores.

Evaluation of the Swedish automatic parses, finally, is shown in Tables 7.4 and 7.5. The Svannotate parser creates structures that are more similar to the gold standard SMULTRON trees, than does the combined GTA-Malt parser. The versions of the Swedish automatic parses with edited labels score higher than the basic versions. (For this reason, the tables ignore the basic versions of the automatic parses compared to the flat gold standard trees.) The edited Swedish GTA-Malt parse achieves an average Leaf-Ancestor score of around 60-65 for phrases and 45-50 for edges, and a labelled F_1 of 25-30 for phrases and 15-20 for edges. The edited Svannotate parse has an average Leaf-Ancestor score of around 75 for phrases and close to 60 for edges, and a labelled F_1-score of 50-55 for phrases and 30-35 for edges. For the GTA-Malt parse, the adaptation of labels gives better results for phrase labels, while the Svannotate parse differs from the basic version only in the scores for edge labels. The structures created by both parsers, and by the Svannotate parser in particular, seem to be more similar to the deepened versions of the gold standard. The Svannotate parser is more robust across genres, while the GTA-Malt

Metric	GTA-Malt			Svannotate		
	B	A	FA	B	A	FA
Leaf-Ancestor phrase	24.8	60.7	**64.54**	**73.5**	**73.5**	64.2
Leaf-Ancestor edge	26.3	**45.6**	44.7	30.4	**58.0**	47.2
Unlabelled precision	**35.8**	**35.8**	26.9	**53.0**	**53.0**	26.4
Unlabelled recall	25.2	25.2	**31.8**	**57.2**	**57.2**	47.7
Unlabelled F_1	29.6	29.6	29.2	55.0	55.0	34.0
Labelled phrase precision	14.3	**28.8**	21.8	**47.7**	**47.7**	20.6
Labelled phrase recall	10.1	20.3	**25.7**	**51.5**	**51.5**	37.3
Labelled phrase F_1	11.9	**23.8**	23.6	**49.5**	**49.5**	26.6
Labelled edge precision	1.4	**18.0**	11.3	7.3	**32.6**	9.3
Labelled edge recall	1.0	12.7	**13.4**	7.9	**35.2**	16.8
Labelled edge F_1	1.2	**14.8**	12.3	7.6	**33.9**	12.0

Table 7.5: Evaluation of the economy texts, comparing the automatic Swedish parses to the SMULTRON gold standard. (The evaluated data sets are the basic parse (*B*), the parse with adapted labels (*A*), and the parse with adapted labels compared to the flat gold parse (*FA*).)

parser achieves much better scores for the Sophie part of the treebank.

By exploring the annotation, we will discuss the main differences between the automatic parses and the SMULTRON gold standard. The English Stanford parse is, on a general level, the automatic parse most similar to the gold standard. The difference is that in the SMULTRON trees some edges have edge labels, while there are none in the automatic parse. Additionally, the English SMULTRON trees contain traces (empty tokens) and secondary edges. These have, however, already been removed for the comparison, as the added tokens result in sentences of different length between the gold standard and automatically parsed trees. The automatic parse trees contain some bracketing errors, where words have been erroneously excluded or included in a phrase or clause, and some attachment errors, where a correctly identified phrase or clause is incorrectly attached in the tree. There are few labelling errors, mostly affecting short phrases, such as an adverb erroneously tagged as particle, which thus receives the phrase label PRT instead of ADVP.

There are three main differences between the German Stanford parse trees and the German SMULTRON trees. First, the SMULTRON trees have edge labels on all edges, while the Stanford trees have none. Second, the SMULTRON trees contain crossing branches, which the Stanford trees do not. Finally, the

punctuation symbols are included in the tree by the Stanford parser, while they are left out of the SMULTRON tree, only attached to the virtual root node. Apart from these general differences, the errors are mostly attachment errors, followed by bracketing errors (which is more common in the economy part). There are only few labelling errors.

The two Swedish automatic parses naturally differ most from the gold standard trees, as they have different annotation schemes. One scheme is not more correct than the other, but they focus on different things. Thus, the evaluation does not assess the correctness of the automatic parses, but the closeness of the automatic parses to the gold standard. Although adapting the labels improves the scores, it is impossible to say how much the evaluation scores are affected by differences between the annotations and how much by annotation errors.

While the SMULTRON edge labels mostly contain functional information, the Svannotate edge labels are a mixture of form and function, such as subject and object, and subdivision of verbs and punctuation symbols. The GTA-Malt structure is often flatter than the flat gold standard trees as, e.g., NPs within PPs (not even complex NPs such as coordinated NPs) are not annotated in the GTA-Malt tree. The Svannotate trees, however, are more similar to the deepened version of the gold standard, with detailed annotation of, e.g., NPs within PPs and unary phrases. Punctuation symbols are attached to both the GTA-Malt and the Svannotate tree, which is not the case in the gold standard.

Another major difference between the Swedish automatic parses and the gold standard is how verbs and verb phrases are treated. The Svannotate verbs do not have different PoS-labels for different types of verbs, but the difference between finite and non-finite verbs is shown on the edge label. The VZ label, which connects the infinitive marker to the non-finite verb (discussed on p. 52), only appears in the SMULTRON trees. The Svannotate parse, however, has one VP for the infinitive marker, dominating another VP containing the verb. The GTA-Malt tree does not make the same distinction between an S and a VP as in the SMULTRON or Svannotate tree, where the S is headed by a finite verb and the VP by a non-finite verb. Instead, the GTA-Malt clause may directly dominate several verbs, which are distinguished by the Malt edge labels (dependencies). This affects the experiments in Chapter 9, where we explore annotation projection to improve PP-attachment, since it makes it more difficult to determine to which verb a PP is attached.

7.4 Summary

In this chapter, we have described how the SMULTRON texts were automatically tagged and parsed for later experiments. Sentence boundaries and tokenization were given to the tools, as the evaluation tools require a fixed set of

tokens and sentences. For English and German we used pre-trained versions of the Stanford lexicalized probabilistic parser. For Swedish we used two different parsers. We combined the output of the GTA tagger and chunker for Swedish with the dependency parses of a pre-trained model for the MaltParser. Additionally, we used the Svannotate parser, which consists of a pipeline of pre-trained models for the Hunpos tagger and the MaltParser (constituency structures). The Swedish parses required adaptation of sentence boundaries and tokenization, as well as PoS-tags, phrase labels and edge labels.

The different taggers achieved accuracies around 94-95%, apart from the Svannotate tagger, which was below 90%. The GTA-Malt tagger received the highest scores. The error analysis, however, showed that taking systematic differences between the annotation schemes into account, all systems achieved accuracies around 95%.

The English version of the Stanford parser created the structures most similar to the gold standard, while the Swedish GTA-Malt structures were the most dissimilar, even with adaptations of the labels. Tagging and parsing of the English texts did not differ much between the Sophie and the economy part of the treebank, suggesting that this is a robust tagger and parser. The scores for the other languages varied between the two text types.

We will not use the complete automatic parses in the remainder of this thesis. In Chapter 8 we use the gold standard trees for the alignment experiments, as we need a gold standard alignment to which we compare the automatically created alignment. For the experiments with prepositional phrase attachment in Chapter 9 we will extract a test set from the gold standard and automatic parse trees. The evaluation of the parse trees reported in the current chapter is, however, important to assess the general correctness of the automatic parses.

8. Automatically Aligning SMULTRON

Alignment, and alignment quality, is important for annotation projection. Without alignment, we do not know which annotation to transfer or where to transfer it. In this chapter, we explore automatically creating the alignment by several methods, both for word alignment and for phrase alignment. Additionally, we will explore a method of annotation projection to improve the automatically created annotation. Sentence alignment is generally a pre-requisite for sub-sentential alignments, but since there are well-established algorithms for aligning sentences, we use the gold standard sentence alignment as a basis for our experiments with word and phrase alignment.

Using the gold standard sentence alignment also alleviates an issue particular to the German economy treebank. For one part of the text, the translator had decided to render the sentences in a different order. As we can see from Figure 8.1, two stretches of the text are placed earlier in the text for German, while they are closer to the end for English and Swedish. The text is still coherent, the paragraphs have just been re-ordered. This also shows that not even sentence alignment can always be assumed to be sequential.

The English economy part of SMULTRON contains one sentence with 152 tokens, which has been recurrently problematic during the creation of, and the experiments with, the parallel treebank. For example, this sentence was difficult to align manually, since the annotators could not get an overview of the syntactic structure of the sentence. Additionally, the Stanford parser could not handle it (see Section 7.1). Due to these difficulties, the sentence is ignored during all evaluation of the alignment.

Several of the alignment methods are directional, creating alignment from a source language to a target language. These methods thus exhibit different results for the two possible directions of a language pair. It is important to point out that the alignment of the parallel treebank, however, is not directional.

We have evaluated the automatic alignment by comparing it to the SMULTRON alignment. To measure the similarity between the automatic alignment and the gold standard, we have used precision, recall and the combined F_1-measure (Section 4.2.1). The scores have been multiplied by 100.

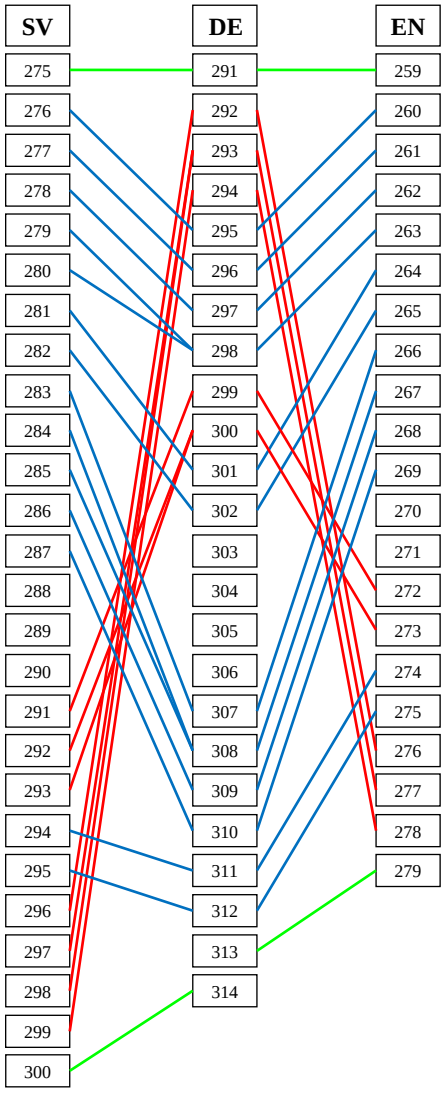


Figure 8.1: Sentence alignments in the economy part of the SMULTRON parallel treebank, where the German text has a different sentence order.

8.1 GIZA++ and Moses

GIZA++ (Och and Ney, 2000b, 2003)[1] is probably the best known and most widely used system for computing word alignments. It builds on the program GIZA, which is a part of the EGYPT toolkit for statistical machine translation.[2] GIZA++ is based on IBM models 1-5, described in Brown et al. (1993) (see also Section 6.1.2), with an implementation of the Hidden Markov Model (HMM) alignment algorithm. A detailed description of the GIZA++ software can be found in Och and Ney (2003).

We have run GIZA++ through the Moses software (Koehn et al., 2007), which uses the GIZA++ word alignments as a step in creating word alignments for the Moses n-gram alignments.[3] Moses combines the GIZA++ word alignment through the grow-diag-final heuristic, which uses the intersection of the two GIZA++ alignments and then adds alignment points. These additional points are created by aligning the immediate neighbours of already aligned words, if these neighbours are aligned in at least one of the two GIZA++ alignment files.

Results

The results of the GIZA++ alignment with the Moses grow-diag-final heuristic are listed in Table 8.1. We ran Moses in both language directions for all language pairs. While Table 8.1 only shows the best results for each language pair, there are minor differences depending on the direction, between 0.25 and 1.25 points in F_1-score. Moses uses the GIZA++ alignments in both language directions, and the results should thus be the same for running Moses in both directions. For example, both the German-English and the English-German runs of Moses start from the intersection of the German-English and English-German GIZA++ alignment. Therefore, we can assume that the differences between the language directions come from differences in the additional Moses heuristic.

While recall is quite respectable, around 75-80 except for the Swedish-German economy texts, precision is low, generally below 50. It should be noted, however, that 500 sentences is considered a too small data set for GIZA++ and Moses. Statistical training modules need large sets of training data to extract reliable statistics.

In an attempt to improve the alignment of the about 500 sentences in each of the two halves of SMULTRON (the Sophie and economy parts, which were

[1] See also http://code.google.com/p/giza-pp/.

[2] See http://clsp.jhu.edu/ws99/projects/mt/.

[3] See also http://statmt.org/moses/.

Data set	Languages	Links	Precision	Recall	F_1
Sophie	DE-EN	8,699	42.1	76.5	54.3
	SV-EN	8,508	44.1	80.4	57.0
	DE-SV	8,387	46.6	79.0	58.6
Econ	DE-EN	12,551	46.9	75.4	57.8
	SV-EN	10,716	59.5	82.7	69.2
	DE-SV	12,342	37.3	67.8	48.1

Table 8.1: Results for word alignment with GIZA++ and the grow-diag-final heuristic of Moses.

run separately), we added data for training. More economy texts were added to the economy part, increasing the number of tokens 4 times. The rest of the novel Sophie's world was added to the Sophie part, which increased the number of tokens 26 times. As expected, precision and recall were improved. The alignment for the economy part was only slightly improved, as the increase in extra data was small. For the Sophie part, the average precision went from 44 to 49, and the average recall from 78 to 86.

More training data, with millions of tokens rather than 200,000, would most likely give more improvement. What is clear, however, is that the alignment method focuses on aligning as many words as possible, based on co-occurrence, in contrast to the alignment of SMULTRON, which tries to find translation equivalence. This is also visible in the low precision, compared to the much higher recall.

8.2 Berkeley Aligner

The GIZA++ word alignment was hard to improve (see, e.g., the minor improvements achieved by Och and Ney, 2003, for Model 6), which inspired several groups to begin working on other approaches to alignment. One of these groups was the Natural Language Processing Group at Berkeley University of California. The Berkeley Aligner (DeNero and Klein, 2007, Liang et al., 2006) is a word aligner built in Java, for statistical machine translation.[1]

The aligner builds on the observation that the intersection of the predictions from two directional models gives better alignments than each of the two

[1]See also http://code.google.com/p/berkeleyaligner/ and http://sourceforge.net/projects/berkeleyaligner/.

Data set	Languages	Links	Precision	Recall	F_1
Sophie	DE-EN	5,477	63.3	72.3	67.5
Sophie-large		5,452	**73.0**	**83.0**	**77.7**
Sophie	EN-SV	5,741	63.4	77.9	69.9
Sophie-large		5,638	**69.1**	**83.4**	**75.6**
Sophie	SV-DE	5,545	65.8	73.8	69.6
Sophie-large		5,442	**74.5**	**81.9**	**78.0**
Economy	DE-EN	7,529	65.5	63.2	64.3
Econ-large		7,537	**68.6**	**66.2**	**67.4**
Economy	EN-SV	8,178	72.4	76.8	74.6
Econ-large		8,150	**74.0**	**78.2**	**76.0**
Economy	SV-DE	7,459	57.1	62.7	59.8
Econ-large		7,525	**59.9**	**66.4**	**63.0**

Table 8.2: Results for word alignment with the Berkeley Aligner, using the SMULTRON texts, as well as the larger data sets.

models by themselves. The aligner contains a function that incorporates data likelihood and a measure of agreement between models, and this function is maximized by an EM-like algorithm. This method encourages agreement during training through joint training, which is then followed by intersecting the produced alignments for further agreement. Initial tests (Liang et al., 2006), where two simple sequence-based HMM models were jointly trained, reduced AER (Section 4.2.1) by 29% over test-time intersected GIZA++ model 4 alignments.

Results

The Berkeley Aligner is available for both supervised and unsupervised alignment, and for plain-text alignment as well as with a syntax-aware distortion component. The syntactic alignment model requires the trees in a bracketed format, which means that SMULTRON-trees with crossing edges cannot directly be used as input. For the experiments here we have used the unsupervised aligner (v2.1), without syntax.

We ran the Berkeley Aligner with the basic settings, with joint training of two HMM models (one in each direction) and two iterations per model. Increasing the number of iterations did not improve the results. As for the GIZA++ alignment, we added data for training. The aligner was thus run on

four different data sets for all three language pairs. Reversing the source and target languages is pointless, due to the inherent bi-directionality of the model. The results can be seen in Table 8.2. More data improves the alignment, visible from the difference between the SMULTRON data and the larger data set, and from the difference between the Sophie data (26 times more training data) and the economy data (4 times more training data).

Let us compare these results to the results of the GIZA++/Moses alignment reported in Section 8.1. The average precision, for all language pairs and all text types on the smaller data set, for that alignment was 46 and the average recall 77. The Berkeley Aligner, with the smaller data set, achieves an average precision of 65 and an average recall of 71. For the same data set, the Berkeley Aligner thus gets a slightly lower, but still respectable, recall, and a major improvement in precision. Additionally, when adding more data to the Sophie part of SMULTRON, the average precision jumps to 72, and recall to 82, for the Berkeley Aligner, with an average precision of 49 and a recall of 86 for the GIZA++/Moses alignment.

8.3 HeaDAligner

When we started building SMULTRON, there were not many phrase alignment systems available. We thus explored linking phrases based on automatic word alignment (see Samuelsson, 2004). The initial system used the word links from a specific aligner, and aligned phrases dominating words with a number of particular edge labels. They were HD (head, for simple phrases), CD (coordinating conjunction, for coordinated phrases), PH (place holder) and PNC (proper noun component, for multi-word proper nouns), which were considered to be the main parts of a phrase, and present in most phrases of the German and Swedish treebanks of SMULTRON.

This aligner was tailored for the specific annotation of the German and Swedish parts of SMULTRON, and could not handle annotation schemes with other edge labels, or without edge labels. This is problematic, e.g., for the English treebank of SMULTRON, which only contains a few edge labels. Additionally, we wanted to make the aligner work on any external word alignment. We now have rewritten and adapted that method for new phrase alignment experiments, calling it the HeaDAligner.

As the new aligner still works on edge labels, we created a pre-processor, which inserts HD-labels into the treebanks. This program is specific to the annotation scheme, but can be adapted to work on any annotation of choice. If a phrase has one and only one HD label, this one is kept, else it adds an HD label to one word, depending on what type of phrase it is. The HeaDAligner then examines every word pair from the word alignment and checks the edge

Data set	Languages	Filter	Links	Precision	Recall	F_1
Sophie	DE-EN	-	3,704	**43.9**	**57.6**	**49.8**
		+	3,647	43.8	56.7	49.4
Sophie	EN-SV	-	3,961	47.7	**60.2**	**53.2**
		+	3,913	**47.7**	59.5	53.0
Sophie	SV-DE	-	3,964	66.9	**79.0**	**72.5**
		+	3,914	**67.1**	78.3	72.3
Economy	DE-EN	-	3,910	30.6	**36.4**	33.2
		+	3,845	**31.1**	**36.4**	**33.5**
Economy	EN-SV	-	4,894	44.7	52.4	48.2
		+	4,825	**45.4**	**52.5**	**48.7**
Economy	SV-DE	-	4,936	50.6	**57.7**	53.9
		+	4,836	**51.3**	57.3	**54.1**

Table 8.3: Results for phrase alignment with the HeaDAligner, based on Berkeley Aligner word alignment trained on the larger data sets, with and without filtering many-to-many alignments.

labels of these words in the TIGER-XML treebank files. If both labels are HD, the phrases directly dominating these words are aligned.

The previous version of the HeaDAligner, as mentioned in Samuelsson (2004), had different rules for one-to-one and for many-to-many phrase alignment. The current version instead has a filter, which tries to find a common ancestor for multiple alignments. If phrase $L1_x$ gets aligned to $L2_y$ and $L2_z$, we check if the parent node of $L2_y$ is also the parent node of $L2_z$. If they have the same parent, we align $L1_x$ to this phrase instead, else we fall back and keep the multiple alignments.

Results

The HeaDAligner was run on the word alignment created by the Berkeley Aligner (joint training with two iterations on the larger data sets, see Section 8.2). The evaluation results of the phrase alignment are shown in Table 8.3. We have aligned all three language pairs for the two treebank parts, with and without the filter for many-to-many alignments. As we can see, the results vary much between the different language pairs and text types, with German-Swedish alignment of the Sophie part getting the best results, with precision around 65 and recall at almost 80. The German-Swedish alignment

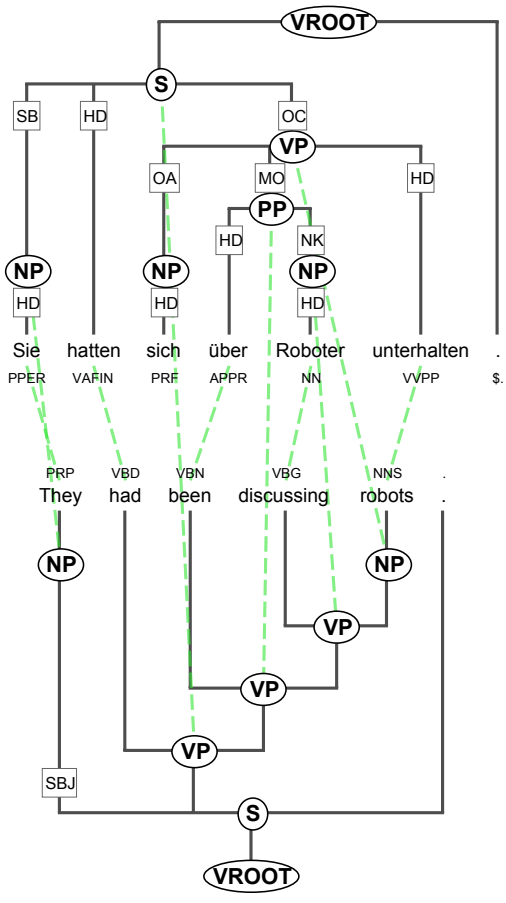


Figure 8.2: A sentence with word alignment by the Berkeley Aligner and phrase alignment by the HeaDAligner.

is most likely better because the tree structures are more similar.

The filter for many-to-many alignments seems to slightly improve precision, but not recall. We thus probably remove some alignments where one of the multiple links was correct. It is also interesting that not using the filter seems to improve the F_1-score for the Sophie treebanks, while using the filter does improve the score for the economy treebanks. The differences between using the filter or not are, however, small.

In Figure 8.2 we see an example sentence aligned by the HeaDAligner. *Sie* and *They* were wordaligned by the Berkeley Aligner and are head within their

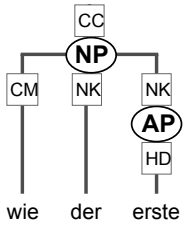


Figure 8.3: A phrase, where the head of the phrase is another non-terminal (here the AP), cannot be aligned by the HeaDAligner.

phrase, and the two NPs dominating these words thus, correctly, get aligned. The German words *Roboter unterhalten* have been misaligned at word level to *discussing robots*, resulting in erroneous alignment at phrase level (the NPs should be aligned, and the German VP *sich über Roboter unterhalten* should be aligned to the English VP *been discussing robots*). The alignment between *hatten* and *had* is problematic, since although the word alignment is correct, the difference between the German and the English tree structure results in the English VP (without subject) being aligned with the German S (with subject).

Additional experiments were conducted, using the GIZA++/Moses word alignment and the SMULTRON gold standard word alignment as a basis for the HeaDAligner phrase alignment. The alignment based on the GIZA++/Moses word alignment was much worse, while using the gold standard alignment to produce the filtered phrase alignment received an average precision of 59 and recall of 65, compared to the average precision of 48 and recall of 57 for the filtered phrase alignment based on the Berkeley alignment.

A number of phrases cannot be aligned, because they do not have a child with a HD label. For example, since the HeaDAligner bases its decisions on the word alignment, non-terminals that only dominate non-terminals, or where the head of the phrase is a non-terminal, cannot be aligned. In Figure 8.3 the AP is the head of the NP *wie der erste* ('like the first').

For the German Sophie treebank, 29 of 4,839 non-terminals did not get a head, for English 60 of 7,020, and for Swedish 23 of 5,351. For the economy treebanks, 367 of 7,139 German, 160 of 8,765 English, and 86 of 7,090 Swedish phrases do not have a head. For the English treebanks, some phrases are missing a head because we have chosen not to make empty tokens (traces) head, even if they are the only child of their phrase. It is unclear why there is such a large amount of phrases without head for the German economy treebank, but one possible explanation is that it contains a large portion of complex

NPs, where a top NP contains a determiner and a non-terminal.

Another problem with phrase alignment that is based on the head of the phrase is, for instance, Swedish sentences without a main verb. The auxiliary verb *ha* ('have') can be omitted from a subordinate clause when the verb phrase is in perfect or past perfect tense, from any kind of clause when it is a non-finite part of a verb chain and from main clauses that contain *kanske* ('maybe'). Since the S has no HD in these cases, the alignment program will not find anything to align.

It is clear that the results depend on several factors here. It is important to put some effort into the program that inserts head-labels in the treebanks. Phrases without a head or with the wrong head will get bad alignment or no alignment at all. Additionally, the quality of the word alignment is crucial.

8.4 The Linguistic Alignment Filter

After the HeaDAligner, we experimented with automatically creating phrase alignment based on the n-gram alignment of freely available statistical machine translation systems. We call the system the Linguistic Alignment Filter (LAF), and it was previously described by Samuelsson and Volk (2007b), where we reported experiments for English and Swedish, using the phrase tables generated by Pharaoh and Thot. Here, we have repeated the experiments for all of the SMULTRON data, using Moses (Koehn et al., 2007), which has since superseded Pharaoh.

Moses is a decoder for statistical machine translation (SMT), but is available together with scripts for training the SMT system, including scripts for extracting n-grams from word-aligned sentences. The word alignments, described in Section 8.1, are used to estimate a maximum likelihood lexical translation table. Pairs of n-grams are extracted that are consistent with the word alignment, meaning that consecutive words in one language all have alignment points connecting them to consecutive words in the other language. The end result is the translation model, the so called phrase-table, which contains a set of multi-word unit correspondences in the form of a source language n-gram, a target language n-gram, and conditional probabilities, including the phrase translation probability for source|target.

Statistical machine translation tools like Moses align items at a level above the basic word level and are therefore generally called phrase-based. However, systems working on unannotated parallel texts do not use syntactic information, and these are thus not phrases in a linguistic sense. As discussed in Section 6.1.2, we call these n-grams. We will, however, continue calling the translation model of the SMT system phrase-table. Additionally, when referring to Moses we mean the scripts for extracting phrase-tables and not the

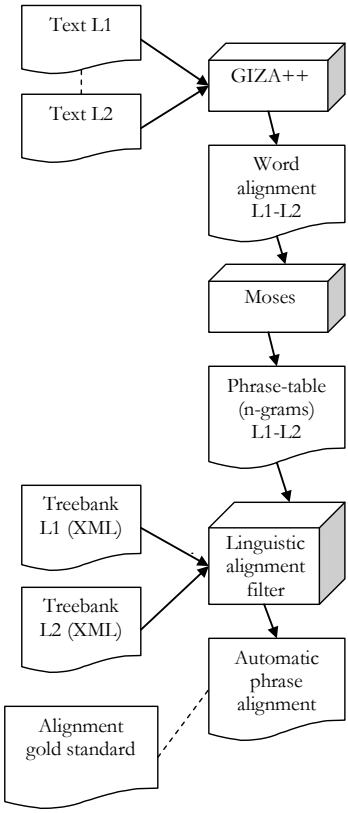


Figure 8.4: The Linguistic Alignment Filter: Basic experimental set-up.

decoder, unless otherwise stated.

Our experiments have two phases. As can be seen in Figure 8.4, we first ran the Moses training module to create the n-gram alignment. The output of each run of the SMT training module is a phrase-table, which in the second phase is fed to the LAF program. The LAF needs the treebank file for both language 1 (L1) and language 2 (L2), the phrase-table for L1-L2 and a file containing sentence alignment. The LAF goes through the treebank file of one language and for each sentence extracts every phrase of the tree as a string containing the words that it spans. It then compares this string to the n-gram entries of the phrase-table. If there is a match between the string containing the phrase and the string containing the n-gram, the node number is stored together with the corresponding (aligned) n-gram (L2) and the probability. The number of

possible links from the phrase-table is thus reduced to only the ones where the L1 n-gram equals a syntactic phrase. Then the second treebank is processed. Every node of every tree is again extracted as a string of words and this string is compared to the already stored L2 n-gram strings. If there is a match, the node identifier is stored. This reduces the number of alignment links further, only leaving links where both the L1 and the L2 strings equal syntactic phrases.

The phrase-table is type-based, which means that we do not know which token instances to align. If the n-gram *she* is aligned to the n-gram *hon* in the phrase-table, every *she* in the English treebank would be aligned to every *hon* in the Swedish treebank. To avoid this, we used a separate XML-file with sentence alignment to restrict the alignment links created. Phrase links are only allowed within aligned sentences. For these experiments, we extracted the sentence alignment from the gold standard alignment, but it could of course be automatically constructed with an automatic sentence aligner (which is needed for running the statistical machine translation system anyway) for a live system, without gold standard data.

The phrase-table contains punctuation symbols, which are also part of the tree in the English annotation format. Punctuation symbols are, however, not included in the tree structure in the Swedish or German annotation format. It would have been possible to attach them to the gold standard trees, but we decided to remove them from the n-grams. For concatenating the string of words of a syntactic phrase, a token not containing any alphanumerics was just not added to the string. Removing all punctuation symbols from the phrase-table string is not as simple, since we do not want to remove, e.g., the apostrophe of *haven't*. The (rather simple) solution was to remove all punctuation symbols following or followed by a blank. Then, for tokenization, a blank is inserted before the remaining apostrophes, unless the apostrophe was surrounded by n and t, where the blank is inserted before the n. This might still induce some errors (e.g., the tokenization of *can't* should be *can - 't* and not *ca - n't*), but it will handle most cases.

We experimented with adding data to get more training material, as described in Section 8.1. Additionally, the maximal length of the n-gram can be adjusted, and source and target languages can be switched, which allows for evaluation along several dimensions.

Results

The results for basic phrase alignment with the LAF can be seen in Table 8.4, with a general precision of around 70 and recall of around 40. The phrase tables are created with only the SMULTRON data (the data to be aligned), for both language directions of a language pair. The n-gram length defines the longest

Data set	Languages	Links	Precision	Recall	F_1
Sophie	DE-EN	1,536	71.0	38.7	50.1
	EN-DE	1,591	68.6	38.7	49.5
Sophie	EN-SV	1,780	74.0	42.0	53.6
	SV-EN	1,735	75.5	41.7	53.7
Sophie	SV-DE	1,773	75.1	39.7	51.9
	DE-SV	1,736	75.6	39.1	51.6
Economy	DE-EN	1,403	66.5	28.4	39.8
	EN-DE	1,405	65.1	27.8	39.0
Economy	EN-SV	2,218	78.4	41.7	54.4
	SV-EN	2,243	79.1	42.5	55.3
Economy	SV-DE	1,384	71.1	22.7	34.5
	DE-SV	1,420	70.2	23.0	34.7

Table 8.4: Results for basic phrase alignment with the Linguistic Alignment Filter, based on phrase-tables with a maximal n-gram length of 10 words, trained on only the SMULTRON texts.

sequence of words in the phrase-table. It is set to a maximal n-gram length of 10 here, meaning that the phrase-table contains n-grams of lengths 1 to 10. Treebank phrases consisting of more than 10 words thus cannot be matched by the LAF. Increasing the length of the n-grams will match more phrases in the treebanks (at the cost of processing time). Phrases in the treebanks consist of anything from one word up to a whole sentence (the root node). The maximal number of tokens per sentence for SMULTRON is below 60 for the Sophie part (46, 57, and 48, for German, English, and Swedish respectively), and below 110 for the economy part (86, 96, and 109, for German, English, and Swedish respectively), not counting the English 152 token sentence. The average sentence length is 14-15 tokens for the Sophie part and 19-24 tokens for the economy part.

We conducted experiments to see how much the n-gram length affects the results, by increasing the maximal n-gram length in steps of 10. Longer n-grams also give larger phrase-tables. This is not a computational issue with only 500 sentences, but will be with a larger corpus. As expected, precision and recall increase with the length of the n-grams, until no additional n-grams can be matched to phrases. For the Sophie part, precision and recall did not increase above n-grams of length 30, peaking at an average precision of 74 and an average recall of 43, compared to an average precision of 73 and a recall of

40 for a maximal length of 10.

The results when switching source and target language are similar, since Moses combines the uni-directional word alignment. We also merged the two phrase alignment files. With a maximal n-gram length of 10, precision for the intersection of the Sophie part reached 76 for German-English, and 81 for English-Swedish and Swedish-German, with a recall below 40.

As was mentioned in Sections 8.1 and 8.2, 500 sentences is not enough data for the statistical model. An early experiment, mentioned in Samuelsson and Volk (2007b), where we doubled the amount of data by adding text from Europarl (Koehn, 2002), resulted in a major drop in precision, while recall improved. Adding more text of the same type for the Sophie part (increasing the number of tokens 26 times), at a maximal n-gram length of 10, we again found that recall increased, from an average 40 to 47, while precision dropped, from an average 73 to 65. It is unclear why extra data is detrimental to the Sophie part. One possible explanation is that OCR errors in the Sophie extra data damaged results. Another possibility is that the Sophie text is heterogeneous, so that the extra data caused the probabilities for certain strings to plummet. This needs to be explored further in the future.

There are a number of problems with the LAF, besides the advantage of being able to create phrase alignment without anything more than a parallel text. First, frequent short n-grams are problematic. At the moment there is alignment between all instances of a word that appears multiple times in a sentence (which is sometimes the case for, e.g., pronouns). This is difficult to handle through statistical alignment.

Second, we cannot easily handle crossing edges. The German and Swedish treebanks of SMULTRON allow for crossing edges. Since this gives us discontinuous constituents, these phrases cannot be matched to any n-grams, as long as n-grams are defined as adjacent words.

On a more general level, the n-gram alignment itself is problematic for alignment of linguistic phrases. The n-gram alignment is based on the assumption that all words of the n-grams have been aligned. This is a too hard restriction for syntactic phrase alignment, where unaligned words may be acceptable. This is one reason for the low recall of the LAF.

8.5 Lingua-Align

Lingua-Align (Tiedemann, 2010) is a set of modules to be used as a toolbox for supervised tree alignment. The classifier makes local binary decisions, based on local, contextual and history features. This is possible by ignoring the crossing constraint (Wu, 1997) (see Section 5.4.2), otherwise used by many tree-to-tree alignment approaches (see, e.g., Zhechev, 2009).

The system has previously shown good results when training on 100 sentence pairs of the English-Swedish Sophie part of the SMULTRON parallel treebank (Tiedemann, 2010, Tiedemann and Kotzé, 2009a,b). It can be used with any standard classifier that supports real-valued features, but currently comes with a log-linear model and a maximum entropy classifier. The Lingua-Align system handles several treebank and alignment formats, including TIGER-XML for treebanks and the Stockholm TreeAligner-XML for alignment, as used for SMULTRON.

Word alignments, such as created by GIZA++ or Moses, can be used for extracting lexical alignment features. The system can align both words (strings) and phrases (trees), and it is possible to force the system not to mix the two. Lingua-Align automatically divides alignments into good and fuzzy alignments. Alignments with a probability below 0.5 get the label 'fuzzy' and alignments with a higher probability receive the label 'good'.

Results

We experimented with different features, aligning words and phrases, but not words to phrases. To get additional lexical clues, we used the word alignment created for the experiments in Sections 8.1 and 8.4. We divided the SMULTRON data into training and test sets, for both 2-fold and 5-fold cross validation. Unfortunately, 5-fold cross validation gave poor results for some parts.

Additionally, for some language combinations and training sections, all probabilities are either 0 or 1 during training, which is detrimental for the alignment results. The runs affected vary with the features used, and a variety of different features were tested, but we could not find the optimal feature set to get results for all data sets. It is not clear whether this is a problem with the training data or the system. It may be the case that while Tiedemann and Kotzé (2009a) found that around 100 sentences was enough training material, this only holds for certain data sets, while being too little data, with too few links, for some other data sets. A full evaluation of features is needed, which is not within the scope here. See Tiedemann (2010), Tiedemann and Kotzé (2009b) and especially Tiedemann and Kotzé (2009a) for discussions and evaluations of features. The economy texts were more problematic than the Sophie texts. As a consequence, we only report results for the Sophie part.

In Table 8.5 we report the results for 2-fold cross validation using a greedy alignment procedure and a well-formedness constraint. At word level, precision is around 60, and recall just over 70, while precision at phrase level is around 70-75, and recall close to 80. Precision is thus lower than the 80 reported for the Swedish-English Sophie part in Tiedemann and Kotzé (2009b) and Tiedemann (2010).

Languages	Level	Links	Precision	Recall	F_1
DE-EN	word	2,729	63.0	71.7	67.09
	phrase	1,631	68.5	79.3	73.5
EN-SV	word	2,750	61.0	71.7	65.9
	phrase	1,718	72.7	79.5	76.0
SV-DE	word	2,977	58.7	70.5	64.0
	phrase	1,723	77.4	79.5	78.3

Table 8.5: Average results for 2-fold cross validation using LinguaAlign, for word and phrase alignment of the Sophie part, compared to the SMULTRON gold standard.

8.6 Alignment Projection

The final method for automatic alignment, first briefly mentioned in Samuelsson and Volk (2007a), draws on the alignment of the other methods, and is a form of alignment projection. The alignment between languages L1 and L2 is derived from the alignment L1–L3 and L2–L3.

We have found nine cases where transferring alignment to words or phrases in L1 and L2 is desirable or possible. The simplest case, shown in Figure 8.5, concerns one-to-one mappings for all three languages. This applies to, e.g., the German PP *von Osterglocken*, which is aligned to the English PP *of daffodils*. The English PP is also aligned to the Swedish PP *av påskliljor*. We conclude that the German PP and the Swedish PP should be aligned. In a second case, shown in Figure 8.6, two words or phrases in the pivot language L3 are linked to one word or phrase in L1 and one in L2. The two English tokens *oil tanker*, e.g., are aligned to the German word *Öltanker* and the Swedish word *oljetanker*. The German word should thus be aligned to the Swedish word.

Next, in what we call case 3 and 4 in Figure 8.7, one word or phrase in L3 is aligned to one word or phrase in L2, but two (or multiple) words or phrases in L1. (We see this as two cases since either L1 or L2 can have the two words or phrases.) This also applies to the example with the *oil tanker* above, but looking at, e.g., English as L1, Swedish as L2, and German as L3. We can still conclude that the English *oil tanker* should be aligned to the Swedish *oljetanker*. Similarly, for cases 5 and 6 in Figure 8.8, if two words in language L3 have a one-to-one correspondence to two words in language L1, and both L3-words are aligned to one word in L2, then we can align the two L1-words to the L2-word.

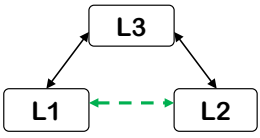


Figure 8.5: Alignment transfer case 1

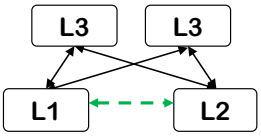


Figure 8.6: Alignment transfer case 2

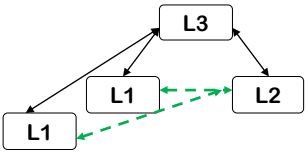


Figure 8.7: Alignment transfer case 3 and 4

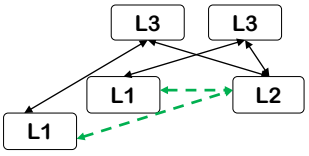


Figure 8.8: Alignment transfer case 5 and 6

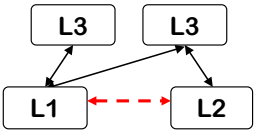


Figure 8.9: Alignment transfer case 7 and 8

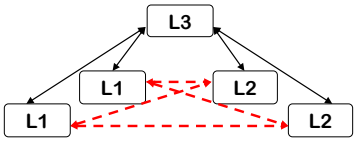


Figure 8.10: Alignment transfer case 9

Cases 7 and 8, in Figure 8.9, and case 9, in Figure 8.10, are more difficult to handle. An example of case 7 or 8 is shown in Figure 8.11. For English-German *in the* is aligned to the contracted preposition and article *im*, while for English-Swedish *in* is aligned to *i*. We thus cannot safely conclude that *im* should be aligned to *i* (the gold standard has them fuzzy-aligned). The example in Figure 8.11 has additional complications, as the same scenario also applies to the alignment of the nouns. Projecting the alignment leads to multiple links between German *im Garten* and Swedish *i trädgården*, which is not allowed in the gold standard. Case 9, similarly, would result in many-to-many alignments, as one pivot language item is aligned to two items in each of the other languages. We cannot distinguishing separate links, e.g., which L1 word should be aligned to which L2 word, from the information we have.

The annotation is projected by a program, which extracts the alignment between languages L1 and L2 from the alignment files of L1–L3 and L2–L3. To remove the problematic cases 7 through 9, many-to-many alignments are ignored for the output.

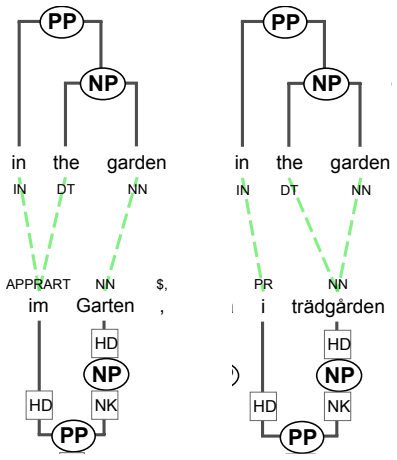


Figure 8.11: An example of case 7/8 of alignment projection, where the alignment is difficult to project to the language pair German-Swedish.

Results

First, we explore projecting the gold standard alignment. Table 8.6 shows the results, when filtering many-to-many alignments, for word and phrase level. Unsurprisingly, without filtering, recall tends to be slightly higher, while precision is lower. When applying the filter, precision is around 90, with phrase alignment for the the German-English economy part a clear exception. Recall varies more, from around 55 to 80, and phrase alignment for the economy part receives lower scores.

Projecting gold standard alignment shows that the six cases described contain the information we need for high precision projection of alignment. However, we also need to evaluate what happens when we use non-perfect, automatically created, alignment as a base for projection. These results can be found in Tables 8.7 and 8.8. The numbers are averaged over all languages. To facilitate comparison, we also show the average precision and recall scores for each system without projection, as reported previously in this chapter.

The GIZA++-system is the GIZA++/Moses word alignment, trained on the smaller data set, while the Berkeley alignment is trained on the larger data sets. The LAF-system refers to the version without a pivot language, using phrase tables with a maximum n-gram length of 10 trained on the smaller data sets. Projection of the HeaDAligner alignment is based on the version using the Berkeley word alignment on the larger data sets, with filtering. As

Data set	Languages	Level	Links	Precision	Recall	F_1
Sophie	DE-EN	word	3,700	90.8%	70.1%	79.1%
		phrase	2,472	87.1%	76.4%	81.4%
	EN-SV	word	4,058	90.2%	78.4%	83.9%
		phrase	2,391	90.1%	68.6%	77.9%
	SV-DE	word	3,809	94.4%	72.7%	82.2%
		phrase	2,261	95.3%	64.2%	76.7%
Economy	DE-EN	word	5,651	89.3%	64.7%	75.1%
		phrase	3,117	75.7%	71.8%	73.7%
	EN-SV	word	5,933	90.4%	69.5%	78.6%
		phrase	2,709	87.3%	56.7%	68.7%
	SV-DE	word	5,965	91.1%	80.0%	85.2%
		phrase	2,557	92.5%	54.6%	68.7%

Table 8.6: Results for alignment projection of the gold standard alignments.

		GIZA++		Berkeley		Lingua-Align	
Data set	Method	Prec	Rec	Prec	Rec	Prec	Rec
Sophie	Sys	44.3	**78.6**	72.2	**82.8**	60.9	**71.3**
	Proj	27.7	70.3	65.7	69.4	57.9	54.8
	Int	**57.2**	61.7	**83.9**	65.6	**81.9**	48.2
Econ	Sys	47.9	**75.3**	67.5	**70.3**	-	-
	Proj	29.5	64.1	57.6	55.1	-	-
	Int	**64.1**	55.0	**79.5**	48.6	-	-

Table 8.7: Average precision and recall for word alignment, showing the system alignments, the projected system alignments, and the intersection of the two.

		HeaDAligner		LAF		Lingua-Align	
Data set	Method	Prec	Rec	Prec	Rec	Prec	Rec
Sophie	System	52.9	64.8	73.3	**40.0**	72.9	**79.4**
	Projected	49.9	52.1	67.5	26.2	72.9	59.3
	Intersection	**83.9**	**65.6**	**80.9**	22.7	**85.2**	54.7
Economy	System	42.6	**48.7**	71.7	**31.0**	-	-
	Projected	38.7	33.0	66.1	15.3	-	-
	Intersection	**79.5**	48.6	**77.8**	12.0	-	-

Table 8.8: Average precision and recall for phrase alignment, showing the system alignments, the projected system alignments, and the intersection of the two.

in Section 8.5, we only report scores for the Sophie part of the treebank for Lingua-Align.

When we project the automatic alignment we lose some precision, and have a major drop in recall, like when projecting the gold standard alignment. Our goal is high precision alignment, if necessary at the cost of recall. Alignment projection is a good start if we have alignment for two language pairs and none for the third pair. The question is if we can improve the alignment even further, with regard to precision. Experiments where we intersected the projected alignment with the system output show that this is possible.

The intersected alignment, also shown in Tables 8.7 and 8.8, receives low recall, worse than the system alignment and the projected system alignment, except for the phrase alignment created by the HeaDAligner. Precision for the intersected phrase alignment, however, outperforms both the original system alignment and the projected system alignment. The main surprise is the HeaDAligner phrase alignment, which receives the lowest scores when evaluation the basic system alignment. Intersecting the HeaDAligner alignment with the projected system alignment results in a major improvement in precision, without reducing recall.

The error reduction (see p. 112) shows bad results. However, the error reduction is based on both the number of errors (i.e., the precision) and the number of missing alignments (i.e., the recall). Since we focus on improving precision here, the error reduction scores are thus misleading.

Error analysis for all of the automatic alignment is difficult, due to the combinatorial power of the five systems, three languages pairs, and two text types. However, we manually inspected the English-Swedish Sophie data, comparing the gold standard alignment to the intersection of the system alignment and the projected alignment, for all six systems (separating the Lingua-Align word alignment from the phrase alignment).

Some differences are striking. For example, at word level, the GIZA++ alignment has many links that are not in the gold standard, while the Berkeley and Lingua-Align aligners more often miss alignments that are in the gold standard. The GIZA++ alignment also has numerous many-to-many alignments. One example is shown in Figure 8.12. These many-to-many alignments are the result of too little training data for the aligner. While the other two systems leave unseen words unaligned, GIZA++ aligns unseen, low-probability, words to each other. Over-all, the Berkeley and Lingua-Align alignments are similar, with Lingua-Align missing more alignments.

At phrase level, the HeaDAligner has numerous alignments that are not in the gold standard, but where the gold standard contains alignment between phrases just above or below in the tree. In Figure 8.13, we see several examples of such phrase alignments. The automatic alignment is close, but not correct.

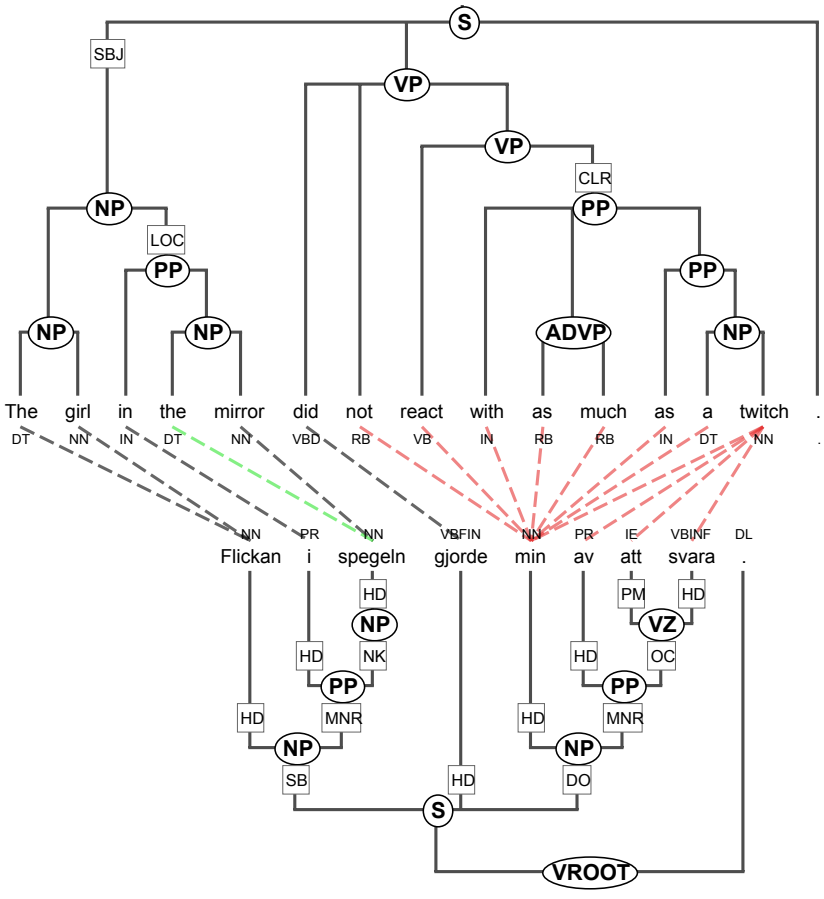


Figure 8.12: A sentence with alignment by GIZA++ intersected with the projected system alignment, compared to the gold standard alignment. Black links appear in both the automatic and the gold standard alignment, while green links are missed and red links added by the automatic alignment.

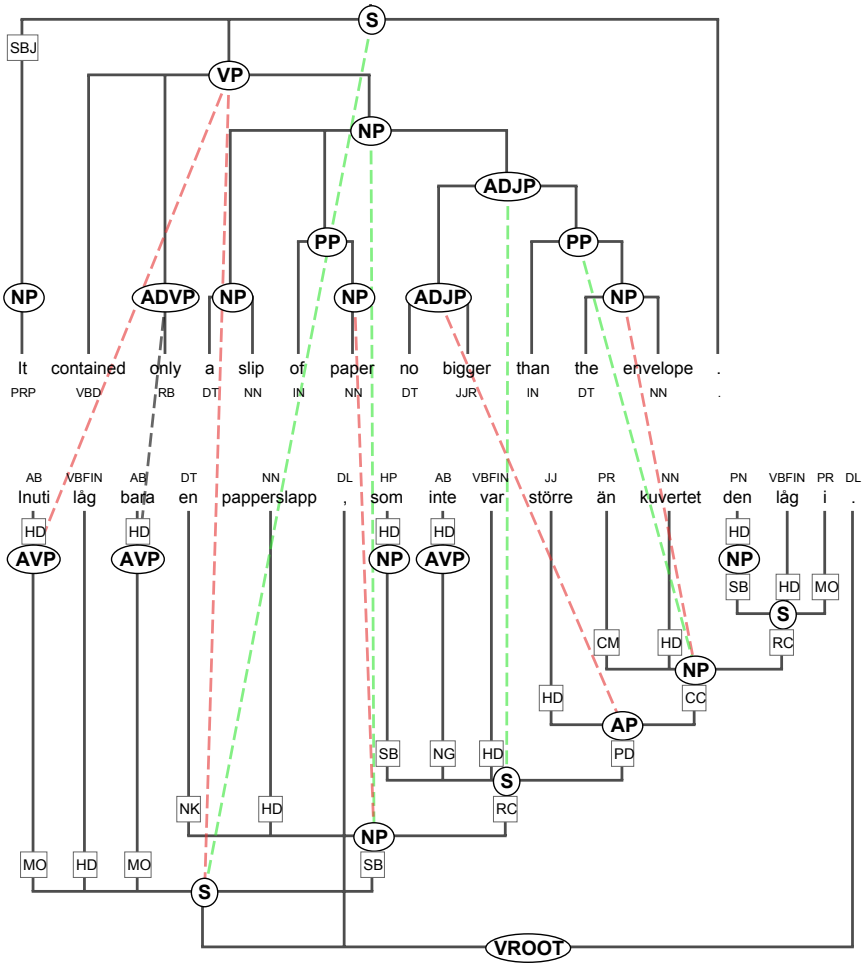


Figure 8.13: A sentence with alignment by the HeaDAligner (based on the Berkeley word alignment) intersected with the projected system alignment, compared to the gold standard alignment. Black links appear in both the automatic and the gold standard alignment, while green links are missed and red links are added by the automatic alignment.

While the HeaDAligner contains many alignment links not in the gold standard alignment, this is more rare for the alignments created by the Linguistic Alignment Filter and Lingua-Align. Their alignments are similar to each other, with the LAF missing more of the gold standard alignment links.

8.7 Summary

In this chapter, we have used a variety of different alignment methods to create word and phrase alignments. We compared these alignments to the gold standard alignment of SMULTRON. The final method described used the output of the other systems to project alignment from two language pairs onto the third pair.

The best word alignment was created by the Berkeley aligner, with a precision close to 85 for the projected alignment intersected with the system alignment, and a recall of over 80 for the pure system alignment. The best phrase alignment was created by Lingua-Align, which, however, did not produce output for all texts and language pairs. The HeaDAligner produced the worst results initially, but got a major improvement in precision without reducing recall, when intersecting the system alignments with the projected alignment.

Some of the alignment methods focus on precision and some on recall. In particular, pure statistical methods tend to achieve better recall values, as they try to align all words. One of the most interesting findings is that we can boost precision by intersecting the projected alignment with the system alignment. If we can improve precision, pick the best alignments, we can provide high-quality alignment for, e.g., machine learning, or for subsequent tasks like syntactic annotation projection.

It is important to remember that accuracy scores from comparing system alignments to a gold standard are relative to that gold standard. For example, our gold standard does not allow many-to-many alignments, but one-to-many alignments, and we do not align punctuation symbols. In contrast to the adaptation of tagging and parsing in Chapter 7, the alignment methods evaluated here have not been adapted to produce alignments similar to the alignments in the gold standard.

In the following, when we turn to syntactic annotation projection, we will extract a test set from the gold standard and automatically parsed SMULTRON data. There, the gold standard alignment is used to select the test set. We will assume perfect alignment to be able to fully explore the projection of syntactic annotation.

9. Annotation Projection to Improve PP-attachment

As we have seen in Chapter 7, the automatically produced trees contain errors. We would like to improve these structures before human quality checks. In a multilingual treebank, we can profit from the information available in the other languages, by exploring annotation projection. For this thesis, we will focus on a sub-area of parsing that is problematic, namely PP-attachment. This is the question of attaching a prepositional phrase, or for dependency parsing, an argument headed by a preposition, correctly in the tree structure. The restriction allows us to explore a defined problem, and investigate it in more detail. The goal is thus to improve attachment decisions in the automatically parsed data, described in Chapter 7, compared to the gold standard data.

To define the problem of PP-attachment, let us look at a classical example, Example 9.1, showing a true attachment ambiguity.

(9.1) She saw the man with the telescope.

The question is whether she saw with the telescope (verbal attachment) or the man had a telescope (nominal attachment). This can be problematic, e.g., for a machine translation system, where the two readings should give two different translations for many languages. Many such ambiguities are unproblematic for humans, since we have world knowledge helping us discern between different readings. A parser, however, generally does not. There are clues that the parser can use to make a decision. For example, some prepositions are more common with one type of attachment. Additionally, there are cases where a possibly erroneous attachment does not matter, because there is still only one possible reading.

(9.2) The GCC held 29 meetings during 2004.

In Example 9.2, it does not matter if we analyze the sentence so that something was *held during 2004* or so that there were *29 meetings during 2004*. There is no difference in content. However, it is still problematic for the parser to decide which cases are truly ambiguous (and should therefore be passed on to a human for decision) and which are not.

There have been many approaches to monolingual ambiguity resolution for PP-attachment, and some also involve multiple languages, such as Fossum and Knight (2008), Schwartz et al. (2003), and Huang et al. (2009). Most of the methods achieve good results, but require large corpora to find clues and features to resolve the ambiguity. For many languages, the training data required is not available. Preferably, we use resource-rich languages to improve PP-attachment for a resource-poor language. Most previous multilingual approaches have disambiguated by using a language where PPs (or their equivalents) are syntactically unambiguous. In contrast, we try to extract the necessary information from closely related languages, where the PP-attachment is potentially ambiguous and incorrect in all languages.

In Chapter 8 we explored automatic creation of alignment at word and phrase level. While we would use automatic alignment in a live situation, we will use the gold standard alignment to extract a test collection for the experiments in this chapter. In the projection experiments we assume perfect word and phrase alignment for the prepositional phrases we investigate. We thus avoid problems that are not related to the syntactic projection, but stem from erroneous or missing alignment.

First, we will describe the extraction of the PP test collection. Second, we will use this test collection to explore and evaluate projection. We project the attachment decisions using a majority vote approach, as well as by exploring linguistic clues.

9.1 Extracting PPs for a Test Collection

For exploration and evaluation of multilingual syntactic projection over PP-attachment, we need a test collection. Initially, we select all prepositional phrases in the parallel treebanks, and then reduce this set, step by step, to arrive at a selection consisting of linguistically interesting examples.

We are interested in PPs that are linked to phrases in the other languages, for projection to be possible. The gold standard alignment information is used to filter the set of PPs and extract phrase triplets. These triplets thus consist of three phrases, one in each language of the parallel treebank, which are aligned to each other, and where at least one is a PP. In Example 9.3 we see such a triplet, where the German phrase has the label NP, while the English has the label PP and the Swedish the label S, and the phrases are aligned for all three language pairs.

(9.3) **DE:** (NP *wie das am Briefmarkensammeln*)
EN: (PP *like - - collecting stamps*)
SV: (S *som när man samlar frimärken*)

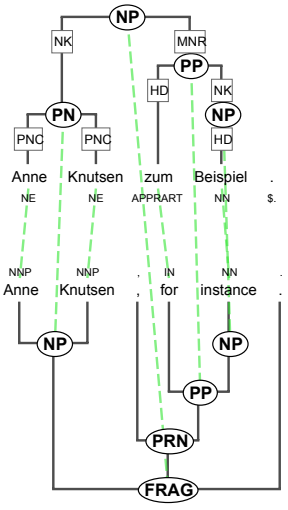


Figure 9.1: The prepositional phrase *for instance* cannot be ambiguously attached in this minor sentence.

A few simplifications are made when extracting the set of triplets. First, we require alignment between at least two of the language pairs. This means that we do not require alignment for all three language pairs, to avoid losing data due to missing alignments (missed by the annotators). Second, we only pick triplets where we have one-to-one alignments. If DE_x is aligned to EN_y and EN_z, we do not know where SV_n belongs, if there is no alignment between EN and SV for this instance. While there are not many phrases with one-to-many links in the SMULTRON parallel treebank, we ignore them to simplify the selection process.

Many of these triplets contain PPs that are not interesting from an ambiguity point of view. There is, for example, no attachment ambiguity for the PP in the minor sentence *Anne Knutsen, for instance.* shown in Figure 9.1. The triplets thus need further filtering. To find possible attachment ambiguities, we are looking for a tuple consisting of a verb (V), a noun (N1), and a prepositional phrase, for example *found* (V) *envelope* (N1) *with her name on it* (PP). We require that a triplet contains a prepositional phrase that is part of such a tuple in at least one of the three languages.

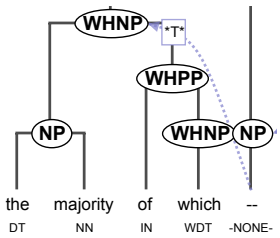


Figure 9.2: An example of a Wh-Prepositional Phrase (WHPP) that is attached to the noun it immediately follows.

A program thus checks the triplets in all three languages, and only keeps triplets where the preposition is immediately preceded by a noun, and preceded by a verb, without any other noun or verb intervening, in at least one of the languages. While prepositional phrases in Swedish and German can also be placed before the verb, we do not select such tuples for the test collection (it can be argued that N-P-V tuples tend to be attached to the noun and P-V-N tuples to the verb). They may, however, be present, because a selected PP in one language corresponds to such a PP in one of the other languages.

The prepositional phrase has the label PP in all three languages. For German and Swedish we also include CPP (coordinated prepositional phrase), where we picked the first prepositional phrase of the coordination. For English, we ignore WHPP (Wh-Prepositional Phrase). A WHPP that immediately follows a noun is generally attached to that noun phrase, as is the case in Figure 9.2.[1] Verbs are chosen regardless of tense or mood (although, e.g., a passive voice may affect the attachment for some PPs), and while N1 can be a noun or a proper noun (name), we do not select personal pronouns, which rarely have PP-attributes. This, again, only applies to the process of selecting, and personal pronouns may thus appear as N1 in the test collection.

In addition to the three gold standard parse trees, we have the automatically parsed data, which is one parse each for German and English (produced by the Stanford parser) and two parses for Swedish (produced by the GTA-Malt and the Svannotate parser). In the rest of this chapter, we will consider two variants of each triplet, once using the GTA-Malt parse for the Swedish part, and once using the Svannotate parse. Examples 9.4 and 9.5 show such triplets, where we see the tuple (one line for each language and parse tree), followed by the

[1]Of the 11 WHPP in SMULTRON, only three immediately follow a noun and only one of these has alignment that would make it a triplet.

attachment. Example 9.4 shows that there need not be a prepositional phrase in all three languages (here the German sentence has a relative clause), and both the English and the Swedish automatic parses have the wrong attachment. In Example 9.5, the GTA-Malt parse has the prepositional phrase attached to the noun, which gives an interesting reading (especially since the verb *stoppa* has multiple meanings ('put', 'fill', and 'mend').

(9.4) **DE gold:** (V fand) einen weiteren gelben (N1 Briefumschlag) (S auf dem ihr Name stand); N-attachment

DE Stanford: (V fand) einen weiteren gelben (N1 Briefumschlag) (S auf dem ihr Name stand); N-attachment

EN gold: (V found) one more brown (N1 envelope) (PP with her name on it); N-attachment

EN Stanford: (V found) one more brown (N1 envelope) (PP with her name); V-attachment

SV gold: (V hittade) ett nytt gult (N1 kuvert) (PP med sitt eget namn på); N-attachment

SV GTA-Malt: (V hittade) ett nytt gult (N1 kuvert) (PP med sitt eget namn på); V-attachment

SV Svannotate: (V hittade) ett nytt gult (N1 kuvert) (PP med sitt eget namn på); V-attachment

(9.5) **DE gold:** (V steckt) den (N1 Kopf) (PP in den Automotor); V-attachment

DE Stanford: (V steckt) den (N1 Kopf) (PP in den Automotor); V-attachment

EN gold: (V sticks) his (N1 head) (PP under the hood of the car); V-attachment

EN Stanford: (V sticks) his (N1 head) under the hood of the (PP car); V-attachment

SV gold: (V stoppar) (N1 huvudet) (PP i bilmotorn); V-attachment

SV GTA-Malt: (V stoppar) (N1 huvudet) (PP i bilmotorn); N-attachment

SV Svannotate: (V stoppar) (N1 huvudet) (PP i bilmotorn); V-attachment

In a first filtering attempt we only kept phrases where the yield of the gold standard phrase was the same as the yield of an automatically produced phrase.

This removed many phrases, which would have been interesting but, e.g., contained a modifier in the gold standard tree that was attached somewhere else by the automatic parser. Example 9.4 shows such a case for the English prepositional phrase, which has the words *with her name on it* in the gold standard tree, but only the words *with her name* in the automatically parsed tree. To avoid loosing data this way, we randomly selected 100 triplets, equally divided over the Sophie and the economy parts. We then determined the head of each phrase (which is often, but not always, a prepositional phrase) and checked the attachment of the phrases in both the gold and auto parses.

During manual evaluation, triplets were removed if the gold standard phrase has no corresponding phrase in the automatic parse. In Figure 9.3 we see how the prepositional phrase *för företaget* ('for the company') in the gold standard (bottom tree) does not have a correspondence in the automatic parse (top tree). The head word of the gold parse is thus required to be the head word of a phrase in the automatic parse. This may not be the case, due to differences in the annotation scheme, or parsing errors. Cases where the head word of the gold standard is the head word of a phrase in the automatic parse, even though the siblings and children of the head word differ, were kept in the collection. As a result, some phrases differ in the number of tokens they span between the gold standard and the automatic parse.

The final test collection thus contains 50 triplets from the Sophie part, and 50 triplets from the economy part. Each triplet consists of three phrases, one from each language, of which at least one is a possibly ambiguous prepositional phrase. Each triplet also has two variants, once connecting the German and English parses with the Swedish GTA-Malt parse, and once with the Svannotate parse. This is necessarily a small test collection. There are a number of issues that are non-trivial to solve for an automatic extraction process, meaning that every instance needs to be manually inspected. First, since we only require a prepositional phrase in one of the languages in a triplet, the corresponding phrase in the other two languages may be, e.g., an NP or a sentence (for example a relative clause). The type of the head word may thus vary. In spite of the varying labels across languages, we will include both the PPs and the corresponding non-PPs when we talk about P, the phrase for which attachment will be explored.

Second, finding the head word is crucial, since, as discussed, the parser may have grouped words differently than the gold standard parse. The gold standard phrase may thus correspond to a phrase in the automatic parse with more, or less, words (or none at all). Third, while one language has a V-N-P-tuple, the other languages may have another order, or even parts of the tuple missing (e.g., no preceding noun). Finally, the automatic parser may have given the head word, the verb V, or the noun N1 other PoS-tags than in the

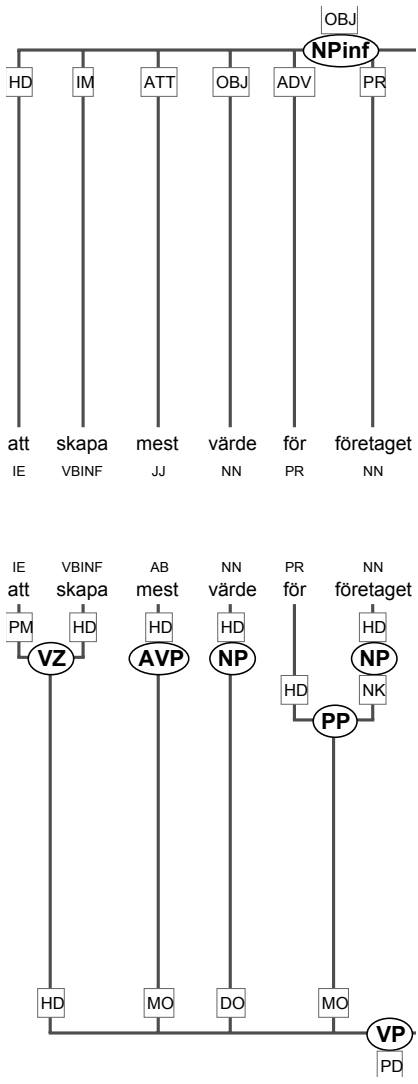


Figure 9.3: The prepositional phrase *för företaget* ('for the company') of the gold standard (bottom, partial tree) has no correspondence in the automatic parse (top).

Data set		N	V	N/V	Other	non-PP
DE	gold	36	53	10	1	9
	Stanford	42	55	-	3	9
EN	gold	40	45	12	3	5
	Stanford	41	58	-	1	8
SV	gold	40	44	13	3	6
	GTA-Malt	36	63	-	1	7
	Svannotate	32	67	-	1	7

Table 9.1: Attachments and non-prepositional phrases for the 100 tuples of the test collection. The PP can be attached to a N (noun), V (verb), N/V (either noun or verb, with no difference in meaning), or Other (adjective or adverb).

gold standard. The manual check of the 100 triplets requires inspecting 700 trees, as there are three gold standard trees and four automatically parsed trees for every triplet.

9.2 Attachment in the Test Collection

Let us begin by exploring the attachment that is our starting point. Table 9.1 shows the ratio of different attachments for P for the test collection. For some of the gold standard phrases, a different attachment decision does not give a different meaning. In a few instances the P is attached to an adjective or adverb. For about 5-10% of the triplets, the prepositional phrase P corresponds to a phrase with another label.

Volk (2006) computes the noun attachment rate, NAR, as the ratio of noun attachments for prepositional phrases immediately preceded by nouns. NAR is around 75% for the Wall Street Journal part of the English Penn treebank, 60% for the German ComputerZeitung, TIGER and NEGRA treebanks, and 66% for the Swedish Syntag and Talbanken treebanks. Our NAR values are much lower. The numbers are, however, not easily comparable. Volk does not include names immediately preceding the preposition, and in our case the N does not need to immediately precede the P for all three languages. Additionally, as pointed out by Volk, his N-P-tuples do not require the presence of a verb, which means that the P does not necessarily have ambiguous attachment.

A general approach to find possible errors for multilingual projection is to compare the attachment of all three languages, to see if they differ. The

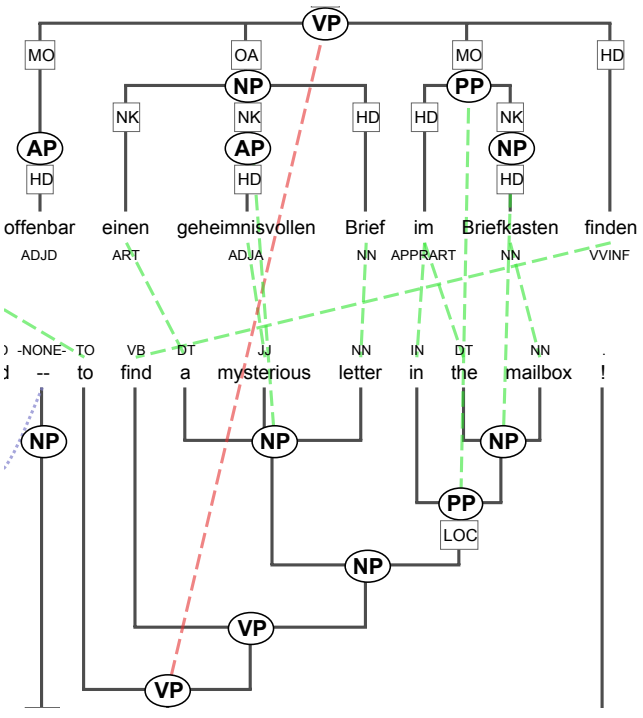


Figure 9.4: The different possible attachment decisions (N or V) for the phrase *in the mailbox* do not change the reading.

question, however, is if we can assume that differences in attachment point to an error, and if attachment to the same phrase type means that they are correct. Investigating the SMULTRON attachments, we found that all three languages agree about either noun or verb attachment in 71 of the 100 cases (37 for the Sophie and 34 for the economy data). Thus, just over one quarter of the cases have different manual attachment decisions in the different languages.

A closer inspection of these 29 triplets showed that four of them contain attachment errors. Of the remaining triplets 14 have different attachment due to differences in wording. In the remaining 11 triplets, a different attachment decision gives no difference in reading for one or more of the languages. For example, in Figure 9.4, the English phrase *in the mailbox*, and its German equivalent *im Briefkasten*, have different attachment. Switching the attachment decisions would not make a difference for the content of the sentences.

Table 9.2 shows the number of correct attachments in the automatically

Data	Correct
DE	76
EN	89
SV GTA-Malt	72
SV Svannotate	73
Triplets (with GTA-Malt)	50
Triplets (with Svannotate)	52

Table 9.2: Correct attachments, in percent, in the test collection, comparing the automatically parsed data to the gold standard.

parsed data. The correctness only applies to the type of attachment (N, V, adjective, or adverb), which means that the structural annotation still may contain errors. We have accounted for the attachment errors in the SMULTRON data, as well as for the cases where the attachment does not affect the reading. Of the 100 SMULTRON triplets, 6 contain errors (one German V-attachment, which should be attachment to an adjective, and three English and two Swedish V-attachments, which should be N-attachments). For 18 of the SMULTRON triplets the attachment does not matter for one or several of the languages, as there is only one possible reading.

There are two important points to make. First, while the differences between the gold standard and the corrected test collection seem to point to flaws in the SMULTRON treebank, we argue that such errors can be found in any gold standard data. Instead, it points to the importance of error analysis. Second, most differences between the SMULTRON data and the corrected test collection consist of cases where two different attachments are possible, and do not change the reading. A raw comparison for such a case points to an error, if the gold standard has one attachment and the automatic parser has chosen the other, although both are acceptable.

The numbers in Table 9.2 confirm that the automatic English parses are most similar to the gold standard, as discussed in Section 7.3, with a correct attachment choice in around 90% of the cases. The Swedish parses are most dissimilar, with just over 70% correct attachments. On average, around 50% of the triplets in the test collection are correct, meaning that all three automatic parses have the correct attachment.

In the following sections, we will explore improving the attachment rates for the 100 triplets of the test collection through annotation projection. There are several possible ways of projecting information in a tri-lingual parallel tree-

bank. We can investigate one language at a time and project information from one or both other languages, or we can look at the information present in all three languages and use the information that seems most likely from the pool. We will investigate these paths and compare the result to the corrected gold standard annotation, measured as a relative error reduction.

9.3 Majority Vote Projection

Majority vote projection means that we project information assuming that the most common decision is the correct one. Thus, if we project syntax from three parse trees, and at least two of these parses have the same structure, we assume that this is the correct structure to project. Several researchers dealing with multiple languages for annotation projection have used majority vote for deciding which structures to project (see, e.g. Bouma et al., 2008, Fossum and Abney, 2005, Kuhn and Jellinghaus, 2006).

We have projected the attachment decision from two languages onto a third for all three languages, as well as pooled the attachment of all three languages, to select the most common one. If there is no majority, no information is transferred between the languages. We project the preferred attachment for a PP (or its equivalent), without regards for position of the V and N1, or the alignment. Only if, e.g, a verb is missing in the automatic parse for one language (possibly due to mistagging), attachment to it cannot be projected to that language.

Table 9.3 contains the results of majority vote projection for the test collection. We report the results of projection for each of the languages individually, as well as the results for the whole triplets (where the attachment has to be correct for all three languages for the triplet to be counted as correct). The results contain the number of introduced errors, the number of correct attachments after majority vote projection, and the error reduction (see p. 112). We project from two languages to the third, as well as by pooling the information of the three languages.

Around 50 of the 100 triplets contain an erroneous attachment when comparing the automatically parsed data to the corrected gold standard data (see Table 9.2). Majority vote projection corrects 21-22 of these, meaning that the PP has the correct attachment in all three languages after projection.

For example, the German tuple *ging Sofie zum Briefkasten* (literally 'went Sophie to the mailbox'), has V-attachment, but received N-attachment in the automatic parse. The name *Sofie* is the subject, which has moved from its position preceding the verb in a main clause, to following the verb in this subordinate clause. In the English sentence, *Sophie* precedes the verb, and the tuple thus lacks an immediately preceding noun to the PP. In the Swedish sentence, the name is also placed between the verb and the PP, but the parsers

Data	New error	Correct	+/- %	New error	Correct	+/- %
	with SV GTA-Malt			with SV Svannotate		
Projection 2 → 1						
DE	6	82	25.0	4	83	29.2
EN	8	83	-54.5	8	84	-45.5
SV	3	82	35.7	1	83	37.0
Triplets	3	72	44.0	3	73	43.8
Projection pool of 3						
DE	6	82	25.0	4	84	33.3
EN	8	83	-54.5	8	84	-45.5
SV	3	82	35.7	1	83	37.0
Triplets	3	72	44.0	3	74	45.8

Table 9.3: Number of introduced errors, and the number of correct attachments, with error reduction (+/-) in percent, after majority vote projection for the test set.

have correctly attached the PP to the verb. As both the English and the Swedish parses have V-attachment, we project this to the German tree, correcting the parsers error.

While the method corrects some errors, it also introduces errors. For example, the English tuple *maintains contact with the auditors* has correctly received N-attachment for the PP by the parser. For the Swedish equivalent, *håller kontakt med revisorer*, the PP has erroneously been attached to the V. The German correspondence, *steht mit den Wirtschaftsprüfern in Verbindung* ('maintains contact with the auditors', literally 'stands with the auditors in connection'), is interesting. The additional PP *in Verbindung* (i.e., not the PP under inspection) is closely connected to the verb, and they form an ideomatic expression. While PPs are rarely right-attached to a following NP, one could argue for attaching the PP *mit X* to *in Verbindung* in this case. At the very least, the preferred attachment of the PP *mit X* depends on the placement of it. If it appears after the PP *in Verbindung*, it would without a doubt have N-attachment. We thus consider it a case that could have either V or N-attachment, and it has received V-attachment by the parser. The German and Swedish V-attachment is projected to English, resulting in not only keeping the error in the Swedish tree, but also introducing an error in the English tree.

Most errors are introduced in the English trees, which had the largest number of correct attachments to start with. Three errors are introduced in triplets

that had previously been correct (where all three languages had the correct attachment, without agreement on the attachment between the languages), while the rest are introduced into triplets that already had an error.

Despite these new errors, however, we see an overall error reduction of around 45%, going from about 50% correct attachments to more than 70%. The difference between projecting from two languages onto a third and using a pool of all three languages is negligible. However, there is a difference in the number of actual projections between the two approaches. For the pooling approach, we require that at least two (i.e. the majority) agree on the attachment, and use this attachment for all three languages. In only two of the 100 triplets, all three languages have different attachment. In such cases, no information can be selected for projection and therefore no change is made.

For example, one triplet has the English tuple *takes part in the discussions*, where the PP has N-attachment (correctly handled in the automatic parse). The German corresponding tuple, *an den Diskussionen teilnimmt* (without an immediately preceding N1 to the PP), has V-attachment, but the PP was attached to an adjective in the automatic parse. The Swedish corresponding tuple, *deltar i styrelsens diskussioner*, is also missing its immediately preceding N1, and has V-attachment (correctly handled by the parser). This means that there is one N-attachment, one V-attachment, and one attachment to an adjective, in the automatic parses.

While there are few cases like this for the pooling approach, we cannot select which information to project in around 40% of the cases when using the 2-to-1 approach. The two languages we project from need to agree about the information for projection to be possible, otherwise we keep the original attachment decision. This is thus consensus projection, rather than majority projection. As the pooling approach shows, more languages means a smaller risk of data sparseness, and therefore more reliable information.

The remaining question, which will be explored in Section 9.4, is if there are better methods for finding and correcting attachment errors than the simple majority vote projection. In particular, can we use linguistic information, available from the syntactic context, to correct errors and to select the structure to project?

9.4 Linguistically Informed Projection

Majority vote projection does not use any information about the specific PP and its context. The question is if there is information in the automatically created structures that can help us improve them automatically, even if we cannot trust them to be correct. In the following, we will explore a variety of clues about the attachment, available from the trees, such as linguistic knowledge

about the prepositional phrase, including the noun inside the PP, called N2, as well as the noun preceding the preposition, the N1. Some of the clues we will discuss are only helpful for determining the correct attachment within a language, while some are also helpful for selecting the attachment to project between languages.

We will not investigate clues that require external knowledge, such as statistics about prepositional bias, as they may be difficult to come by for low-resource languages. Such methods can, however, improve the resource-rich language(s), and give their structure more weight in the projection process. Another possibility, which we will not explore here, is using parser confidence scores, to improve the attachment decisions. For example, it would be possible to ignore attachment decisions with lower confidence scores, when selecting which structure to project.

9.4.1 Immediately Preceding N1

First, we will explore knowledge about the N1, the noun that the prepositional phrase may be attached to. Is the absence of an immediately preceding N1 helpful for the attachment decision when comparing the three languages? There are 52 triplets with at least one language missing an immediately preceding N1 to the P in the automatic parse. Of these, 24 have V-attachment for all three gold structures, 10 have possible V-attachment for all three gold structures (where the attachment thus does not change the reading), and 18 triplets have non-V-attachment in at least one language in the gold standard data. In one of these triplets, the automatically produced tree in one language is missing the N1, while another language is missing the V, due to mistagging. The absence of an immediately preceding N1 in one language thus does not help make the decision for the other languages.

Can we use the information to improve the attachment decision within a language? There are three cases in the gold standard, where the P has N-attachment while the N1 is not immediately preceding the P. In two of them, the P can have either V or N-attachment, without changing the reading. One of these instances is a German sentence where the N1 is separated from the P by a genitive NP within the NP, *die Produktion dieser Frucht* ('the production of this fruit'), where *Produktion* is N1, not *Frucht*. The German automatic parse has V-attachment for the PP. The second instance, also in a German sentence, is the expression *steht mit den Wirtschaftsprüfern in Verbindung* discussed on p. 172. This PP has V-attachment in the automatic German parse.

The third case is a Swedish sentence where N-attachment is required. The PP, containing a relative clause, has been moved to the end of the sentence, *ledamöter är närvarande av vilka...* ('members are present of which...'). The

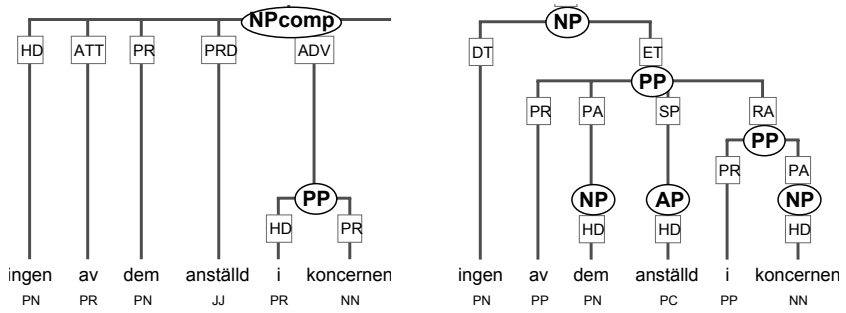


Figure 9.5: The Swedish PP *i koncernen* ('in the group') is attached to a noun for both Swedish automatic parses (GTA-Malt left and Svannotate right), but to the AP containing *anställd* ('employed') in the gold standard.

N1 *ledamöter* ('members') is thus separated from the PP, resulting in crossing branches. The GTA-Malt parse has V-attachment, while the PP is attached to the adjective phrase headed by *närvarande* ('present') by Svannotate. This case is difficult to correct automatically, as the long distance dependency requires correct anaphora resolution of the relative pronoun *vilka*.

In the automatically parsed trees, N-attachment although the N1 is not immediately preceding the PP appears only twice. In Figure 9.5, we see the Swedish PP *i koncernen* ('in the group'), which is attached to a larger NP in both automatic parses. This PP is attached to the AP headed by *anställd* ('employed') in the gold standard.

We change the attachment to V-attachment only if the P is attached to an N and there is no immediately preceding N1 to the P. Thus, we do not change instances where the P is attached to, e.g., an adjective. There are 9 errors in the 52 triplets where at least one language is missing an immediately preceding N1 to the P in the automatic parse. As we can see in Table 9.4, only one of these was corrected by checking for a missing immediately preceding N1 to the P. No errors were introduced. The remaining errors consisted mostly of the cases where the P was already attached to the V, but should have been attached to an adjective or adverb, similar to the example in Figure 9.5. While the method was not effective for this data set, the combination of N-attachment although the N is not immediately preceding the P is uncommon, both in the manual and the automatic annotation. This at least warrants flagging the attachment in the automatic parse as a possible problem, for a human to check.

Data	Items	Err	Corrected	New err	(+/-) %
DE	29	1	0	0	0.0
EN	21	1	0	0	0.0
SV GTA-Malt	20	5	0	0	0.0
SV Svannotate	22	4	1	0	25.0

Table 9.4: Using missing immediately preceding N1 of the P to correct the attachment in the test set, with number of affected items by the approach, and for these affected items the number of errors by the automatic parsers, corrected errors, introduced errors, and error reduction (+/-) in percent. The attachment is not projected.

9.4.2 Names as N1

Certain types of N1, e.g., if the N1 is a proper noun, tend not to be modified by a prepositional phrase. There are exceptions, such as when the PP corresponds to a relative clause, or Swedish PPs following a name. For example, the sentence *John pussade Lisa i kiosken* ('John kissed Lisa in the kiosk') has one reading where *i kiosken* specifies where the kissing took place, as well as one reading where *i kiosken* defines which *Lisa* we are talking about. The English equivalent, as well as the German (*John hat Lisa im Kiosk geküsst*), does not as naturally have the latter reading. In spite of these exceptions, there is still a higher probability that the PP will have V-attachment.

There are eight triplets in the test collection where at least one of the languages has a name for the N1 in the automatic parse. For seven of these, the gold annotation for all three languages has V-attachment. In only one of these seven, all four automatic parses also have V-attachment.

The case where not all gold parses have V-attachment contains different translation variants. The Swedish PP *i hela den tjocka katalogen* ('in the whole thick directory') has different attachments in the different parses, shown in Figure 9.6. The gold parse has V-attachment, while the GTA-Malt parse has N-attachment. Since the N1 of the GTA-Malt parse is correctly tagged as a name, we can apply a rule stating that a P, which is attached to an N1 that is name, needs to be re-attached to the verb. In the Svannotate parse, the P is attached to the verb, which is the mistagged name *Møller*. The N1, the second name *Knag*, has been mistagged as noun. While the PP appears to have the correct attachment, we would need to change several tags and phrase labels to get a correct tree. This problem will thus be left unhandled.

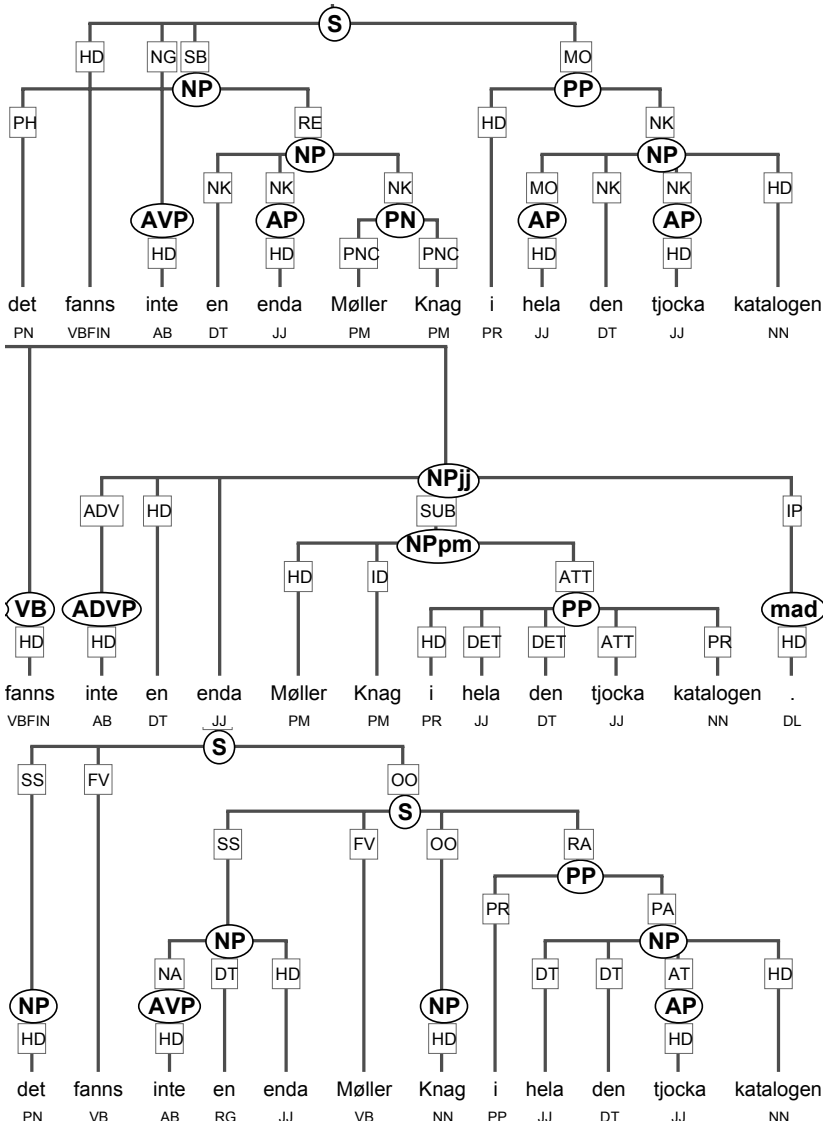


Figure 9.6: The Swedish PP *i hela den tjocka katalogen* ('in the whole thick directory') has different attachment decisions in the gold parse (top), automatic GTA-Malt parse (middle) and the Svannotate parse (bottom).

In the corresponding German sentence, the PP is fronted, without an immediately preceding noun. The German parser has, correctly, preferred V-attachment. The English PP is immediately preceded by the noun *nobody* (*There was nobody in the entire directory called Møller Knag*). The name is positioned after the verb and PP. While the gold standard has N-attachment, the automatic parse has V-attachment. Projecting V-attachment to all languages, based on the Swedish immediately preceding name, would thus correct the attachment of the Swedish GTA-Malt parse, and leave the English parse unchanged, but erroneous.

One English sentence contains the name *Marie* immediately followed by the prepositional phrase *to us*, followed by the verb *came*. The sentence has an uncommon, poetic, word order. The German sentence has a similar wording. The English and the German parsers have attached the PP to the N. For Swedish, the relative pronoun *som* ('who') intervenes between the name and the preposition. This indicates that what follows (including the preposition and the verb) is part of a relative clause, which has likely influenced the parsers decision to attach the PP to the verb. As both the German and the English parses have N-attachment, majority vote projection would incorrectly change the correct Swedish attachment. However, since the N1 is a name, we can instead conclude that the German and English parses have the incorrect attachment, leaving the Swedish attachment unchanged.

Another interesting triplet contains the English tuple *went to the mailbox*, where the N1 is missing. As can be seen in Figure 9.7, the German (and Swedish) subject, the name *Sophie*, appears between the verb and the preposition, due to the V2 rule and a fronted adverbial expression. While the Swedish parsers recognized the name, the German parser did not. Through the alignment, or by string similarity, we can conclude that the German noun, just like the Swedish N1 and the English subject placed just before the verb, should in fact be a name. As a result, the PP should be re-attached to the verb.

One of the English proper nouns, the word *President*, is problematic. Several nouns in the English part of the treebank, starting with an uppercase letter, such as *President*, *Board*, and *Non-core* (in *Non-core activities*) have been tagged as names. Applying the rule about N1 names does not change the attachment in this instance, since the English PP *on compliance matters* is already correctly attached to the verb *assist*. This annotation could, however, clearly be problematic. Nouns with proper noun usage are more likely, compared to real names, to be modified by a PP. The noun origin of this word is, however, unspecified in the current annotation. In these cases, a classification of names, as is available, e.g., in the SUC corpus, would be helpful.

Table 9.5 show the results of re-attaching the P to the V if the N1 is a name. We report numbers for the individual languages without projection,

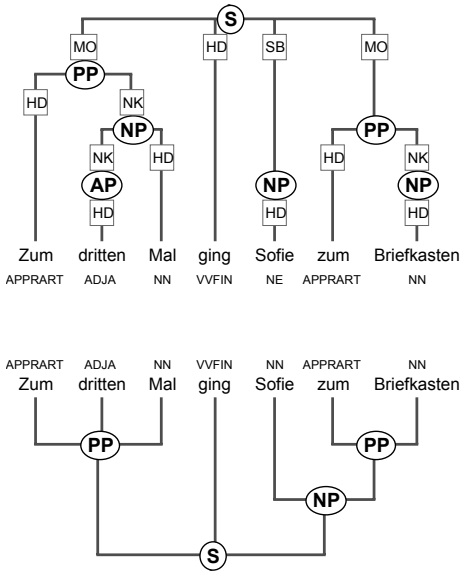


Figure 9.7: The N1 of the PP *zum Briefkasten* ('to the mailbox') in the German gold standard tree (top) is a name. The N1 of the automatic parse (bottom) has been mislabelled as a noun.

which show an error reduction of 100%. After projection, the 5-6 errors in the triplets are almost all corrected, meaning that the tuples of all three languages have the correct attachment.

The attachment can successfully be projected when one language has a name as N1. While the information about a missing immediately preceding N1 to the P does not transfer to other languages, it still can help us improve the attachment, and correct tagging errors, which in turn leads to changes in attachment. Additionally, making a linguistically informed decision in one or several languages can change the majority attachment for subsequent majority vote projection.

9.4.3 Non-Prepositional Phrases Equivalent to PPs

Since we only require the P to be a preposition in one of the three languages, it may be possible to extract information from non-PP phrase labels, to guide the attachment decision. There are 9 German, 8 English and 7 Swedish non-PPs in the automatic parses, which thus correspond to a PP in at least one of the other languages. In Figure 9.8, the Swedish PP *med sitt eget namn på* ('with her

Data	Items	Err	Corrected	New err	(+/-) %
DE	1	1	1	0	100.0
EN	5	1	1	0	100.0
SV GTA-Malt	6	1	1	0	100.0
SV Svannotate	6	2	2	0	100.0
Proj. (with GTA-Malt)	7	5	4	0	80.0
Proj. (with Svannotate)	7	6	6	0	100.0

Table 9.5: Correcting the attachment by not allowing N-attachment if the N1 is a name, with number of affected items by the approach, and for these affected items the number of errors by the automatic parsers, corrected errors, introduced errors, and error reduction (+/-) in percent. The attachment is not projected in the numbers for the individual languages. Projection is evaluated for whole triplets.

own name on it') has been erroneously attached to the verb *hittade* ('found') by the GTA-Malt parser. In the gold standard parse the PP is attached to the noun *kuvert* ('envelope'). As we can see from Figure 9.9, the English parser also made a mistake and attached the prepositional phrase *with her name* to the verb phrase. In the German tree, in Figure 9.10, the PP corresponds to a relative clause, *auf dem ihr Name stand*, attached to the noun phrase.

Majority vote projection within the pool of the three languages would select the Swedish and English V-attachment to project to German. However, the German phrase has the label S, and the left-most child of that S is a PP, containing a preposition followed by a PRELS, a substituting relative pronoun. As this is a relative clause, it should be attached to the noun. If we can conclude that the English and Swedish corresponding phrases are similar, e.g., because there is alignment between the three N1s, as well as between the prepositions and between the N2s, then we can use this information to re-attach the English and Swedish PPs.

The main problem with relative clauses is that they are not always unambiguously identifiable. Thus, such clues should only be used in cases where we have high confidence that the phrase is a relative clause. Only one German and one English phrase are clearly relative clauses. In the English case, the verb *appointed* is followed by the N1 *office*, which is immediately followed by the PP-correspondence *that lasts until the next Annual General* (SBAR, missing the final noun *Meeting* in the automatic parse). In this triplet, all four automatically parsed PPs (and their equivalents) already have N-attachment.

A PP that is equivalent to a German NP in genitive case contains additional

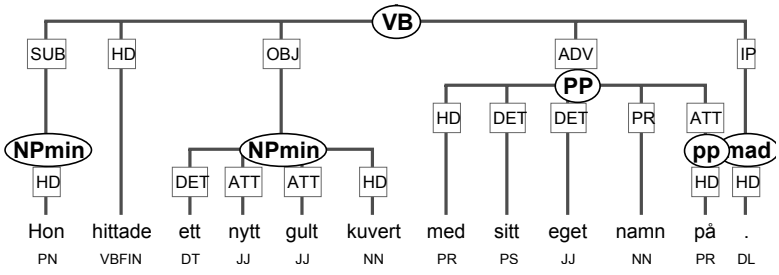


Figure 9.8: A Swedish sentence automatically annotated by the GTA-Malt parser, with erroneous attachment of the prepositional phrase *med sitt eget namn på* ('with her own name on it').

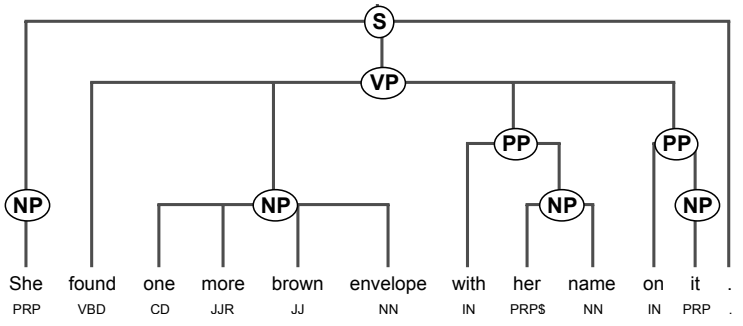


Figure 9.9: An English sentence automatically annotated by the Stanford parser, with erroneous attachment of the prepositional phrase *with her name*.

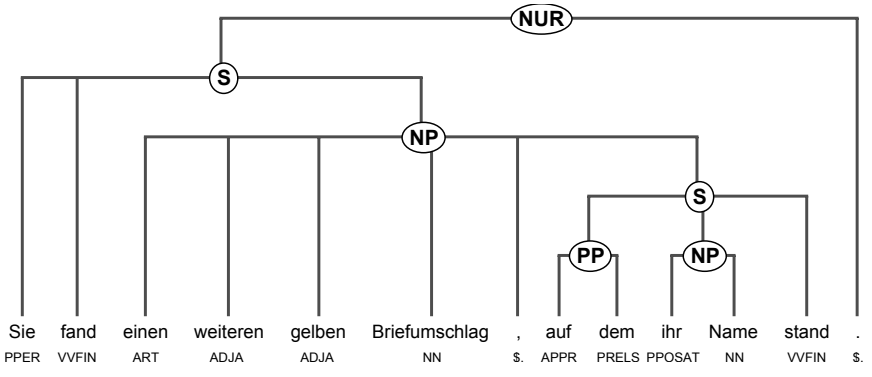


Figure 9.10: An automatically parsed German sentence with correct attachment of the relative clause *auf dem ihr Name stand* (literally 'on which her name stood').

Data	Items	Err	Corrected	New err	(+/-) %
Relative clause	2	1	1	0	100.0
DE NP genitive	5	4	2	0	50.0

Table 9.6: Correcting the attachment by investigating non-prepositional phrases corresponding to a PP, with number of affected items by the approach, and for these affected items the number of errors by the automatic parsers, corrected errors, introduced errors, and error reduction (+/-) in percent. Projection is evaluated for whole triplets.

information. The genitive noun phrase indicates that the preceding noun is being modified. There are five triplets where the PP corresponds to a German genitive NP. Case is sometimes difficult to determine automatically, as, e.g., the article *der* is used for a masculine noun in nominative, as well as for a feminine noun in dative or genitive, and for a plural noun in genitive. In four of the five triplets here, however, the article is *des*, which clearly indicates the genitive case. A fifth triplet contains the German NP *aller drei Ausschüsse* ('of all three committees'), which is also clearly genitive.

In one of these five triplets all gold and all automatic parses have N-attachment. In two cases, all gold parses have N-attachment, while one or both Swedish automatic parses have V-attachment. From the German genitive we can conclude that the Swedish PPs should be re-attached. In the remaining two cases, the German and English parses have N-attachment, while the Swedish gold PPs are attached to adjectives. Neither of the automatic Swedish parses have the correct attachment. However, there are no immediately preceding nouns in these Swedish sentences. We can conclude that this clue is only helpful if there is an immediately preceding noun in the other languages, in addition to having similar words in the other languages (which should be indicated by the alignment).

Differences between the annotation guidelines are problematic, in particular, the difference between the German and Swedish guidelines on the one side, and the English guidelines on the other side. In Figure 9.11 the PP *including the Chairman of the Committee* is attached to the VP, while it is actually a modifier of the NP. The corresponding German sentence has a different word order, with the PP following immediately after the N1. The Swedish sentence has the same word order as the English sentence, with crossing edges to attach the PP to the NP. Of the automatic Swedish parses, one has attached the PP to the verb phrase, and one to the adjective phrase corresponding to *present*.

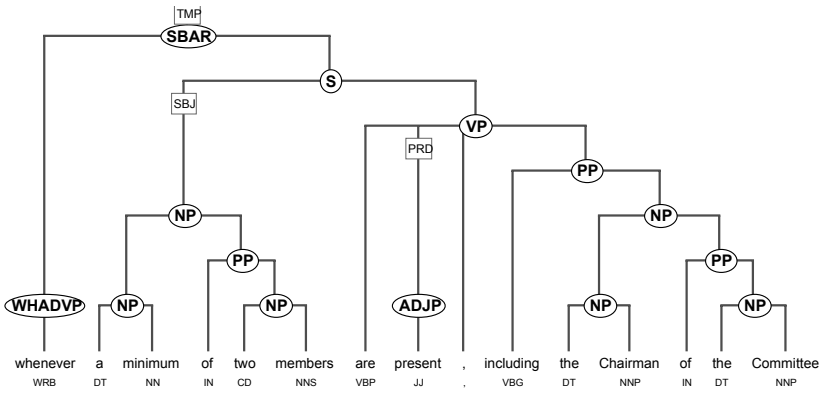


Figure 9.11: An English sentence where the PP *including the Chairman of the Committee* is attached to the VP, while really a part of the NP.

Although the English attachment seems erroneous, this is correct according to the guidelines. However, if we require the N1 to be immediately preceding, this English non-PP will not introduce an error.

9.4.4 Applying all Linguistic Clues

The automatic parsers found the correct attachment for around 90% of the English, and 75% of the German, Swedish GTA-Malt, and Swedish Svannotate tuples. In total, around 50% of the phrases were correctly attached for all three languages. Using a majority vote approach resulted in around 80-85% correct attachments for each of the languages, and close to 75% correct attachments for all three languages.

We applied all linguistic clues discussed above to the automatically parsed data. The information about a missing immediately preceding N1 was only used within a language, while the information about a name as N1, a P that is a relative clause, and a P that is a German genitive NP was also projected to the other languages. The result, shown in Table 9.7, is a correct attachment of almost 80% for German, 90% for English, and more than 75% for Swedish. The numbers for whole triplets are below 60%. The error reduction compared to the automatic parses is around 15%. However, compared to the majority vote experiments pooling the information of the three languages, we now have more attachment errors, except for English.

To gauge the effect of the pooling approach, we also applied the majority vote pooling approach to the data that had been improved by the linguistic

Data	Correct	A +/- %	M +/- %	Correct	A +/- %	M +/- %
	with SV GTA-Malt			with SV Svannotate		
Linguistic clues only						
DE	78	8.3	-22.2	78	8.3	-37.5
EN	91	18.2	47.1	91	18.2	43.8
SV	77	17.9	-27.8	77	14.8	-35.3
Triplets	55	10.0	-60.7	58	12.5	-61.5
Projection after linguistic clues						
DE	85	37.5	16.7	86	41.7	12.5
EN	86	-27.3	17.6	86	-27.3	12.5
SV	85	46.4	16.7	85	44.4	11.8
Triplets	75	50.0	10.7	76	50.0	7.7

Table 9.7: Number of correct attachments, when applying the linguistic clues only, and after subsequent majority vote projection, using the pooling approach. Error reduction (+/-) in percent is reported compared to the automatic parse (A) and majority vote pooling without using linguistic clues (M).

clues. These results, also in Table 9.7, show less variation between the languages than when using only the linguistic clues. While the majority vote again lowers the results for English, just as was reported in Section 9.3, we now get an improvement also compared to the previous majority vote experiments.

Using a small set of linguistic clues, extracted from the data, improves the structures before human quality checks. There are likely more linguistic clues to be extracted from the data. However, there is a problem with this approach. Manually collecting a set of rules requires knowledge about the languages, and time and effort to add rules to a system. Automatically collecting the knowledge requires large amounts of language data, which is not always at hand. In such cases, only checking the data manually may be easier, and the majority vote is a faster way to achieve an improvement of the automatically produced data.

9.5 Summary

In the final experimental chapter of this thesis, we have constructed a test collection for PP-attachment experiments. The test collection consists of 100

triplets, where each triplet contains equivalent phrases in the three languages, in the gold standard annotation. In at least one of the languages, the phrase fulfils the requirement of being a prepositional phrase, which is preceded by a verb, and immediately preceded by a noun. In addition to the three gold standard trees, the triplets also contain the four automatic parses, which is the annotation that we want to improve.

The test collection has been subjected to majority vote projection, as well as more linguistically informed correction and projection. We have shown that while majority vote projection improves the annotation, compared to the basic automatic annotation, using linguistic clues to correct the annotation before majority vote projection is even better, although more laborious. However, we have also seen that some errors cannot be corrected by projection at all, as some triplets have different wording in the different languages, and thus do not agree on the attachment. The process of finding attachment errors thus still needs to be refined. While the test collection is small, we have shown that automatic annotation can be improved, before manual, or semi-manual quality checking.

10. Conclusions

In this final chapter, we bring together the various aspects on high-quality parallel corpus development, which have been investigated in the thesis. We have explored developing parallel treebanks, collections of language data consisting of translated texts in two or more languages. These texts are syntactically annotated and corresponding parts are linked through alignment at word, phrase, and sentence level, showing translation equivalence. The development of a parallel treebank is a time-consuming and labour-intensive task.

10.1 Developing a High-quality Parallel Treebank

We have described the semi-manual annotation of the SMULTRON parallel treebank. SMULTRON consists of texts with about 1,000 sentences from two different genres, in three languages: English, German and Swedish. We started by exploring word-level annotations, such as Part-of-Speech tags, morphological annotation, and lemmas. We continued with phrase- and sentence-level annotations, such as the syntactic structure, and concluded with the alignment. This description was a starting point for the first of two questions in this thesis:

- What issues need to be considered to achieve a high-quality, consistent, parallel treebank?

A number of issues have surfaced, which affect the well-formedness, consistency, and soundness of the parallel treebank, and are crucial for quality.

The choice of data is an important issue for the coverage and usability of a parallel treebank, but also for the quality of the annotation. The availability of tools varies between languages, and different tools are also more or less adapted to different types of texts. The performance of different automatic taggers, parsers, and aligners varied depending on the language, and most of them achieved better accuracy for the fiction part of the parallel treebank. A broad collection of texts, from different genres, allows for a multi-purpose corpus, but also requires awareness about the possibility of some tools performing worse for some parts.

The choice of annotation schemes is probably one of the most important issues for achieving high quality in building a parallel treebank. It is crucial

to have decided on rules for everything from sentence splitting and tokenization to annotation schemes for PoS-tagging, parsing, and alignment. These considerations help avoid unnecessary inconsistencies within or between languages. The granularity, or level of detail, of the annotation scheme is important, as there is a trade-off between detailed information and annotation difficulty, which affects quality. The choice of what units to annotate is also important. For example, the coarser level of annotation by the Granska tagger results in a loss of information, just as not allowing for many-to-many links in the alignment does. One also needs to be aware of the implications of these decisions for subsequent annotation. As we have seen, using different parsing schemes for different languages results in problems for the alignment. Overall, all such choices govern the level of complexity of the annotation task, and define what knowledge is encoded in the treebank, and thus what information can be extracted from the resulting data, e.g., through a query tool. They limit, in different ways, our perception of the world.

Methods for automated processing are essential for large-scale treebanking projects. The size of the parallel treebank is important for the usability, and influences what knowledge we can extract and what inferences we can make from the data. Automated processing may include automatic deepening or enhancement of the annotation, using more resource rich languages for bootstrapping, applying automatic taggers and parsers, and annotation projection. Automatic annotation is, however, not better than manual annotation, only possibly faster. In Section 2.1.2 we defined a treebank as manually checked, in contrast to the fully automatically produced parsed corpus. It is important to remember that while humans most likely make annotation errors, machines definitely do. The errors may just be of different types. Thus, some human intervention is necessary to achieve high quality, although low quality may be acceptable for some applications.

In addition to improving quality by increasing the size of the parallel treebank, automatic methods can be used to assess the correctness of the annotation, and to correct erroneous annotation. However, although using evaluation metrics is an easy way of evaluating annotation, it may also be misleading. First, the metrics themselves only show one or a few aspects of the annotation. They may also be biased, preferring some types of errors. Second, when using an annotation scheme that differs from the gold standard, the evaluation scores show the deviations from the gold standard annotation scheme, and not necessarily the errors in the automatic annotation. It is thus impossible to say how many errors there truly are. Third, gold standard annotation is not error-free, and as a result, the evaluation scores may point to errors that are not there. Additionally, there is no fully automated solution to quality checking. Both automatic evaluation and quality checking are thus helpful, but require

additional manual error analysis and error correction.

To summarize, we found that a number of choices are important for the quality of a parallel treebank, the most important being the choice of data, and the choice of annotation schemes, which includes the granularity of the units to annotate and of the annotation labels. Automated processing is necessary to reduce some of the manual work, to achieve a parallel treebank of sufficient size, as well as for quality checking and evaluation. However, while some processing can be automated, high quality still implies some manual work to detect true errors, and to resolve ambiguities.

10.2 Improving Automatic Annotation through Projection

As both manual and automatic annotation will introduce errors, quality checks are required after each step in the annotation. Some of the most effective methods discussed in Chapter 5 locate annotation variation, to help identify possible errors. They abstract away from the language specific issues in a parallel treebank. Annotation projection, explored in Chapters 8 and 9, can be used as a method to build or enlarge a parallel treebank. We have, however, essentially used it for quality checking, to improve automatic annotation. In a way, it is a multilingual method for finding variation. We have thus explored whether we can apply automatic tools, like taggers and parser, which we know to give erroneous output, and then use the information available from the annotation of the other languages to improve this output. This leads us to the second of the two questions explored in this thesis:

- Can we improve automatic annotation by projecting information available in the other languages?

We have shown that annotation projection is a feasible approach to reduce errors, both for syntactic annotation and for alignment. In contrast to many other approaches to annotation projection, we work with closely related languages.

In our experiments with automatic alignment, we used a variety of different alignment methods to create both word and phrase alignments. These system alignments were then projected from two language pairs, L1–L2 and L1–L3, onto the third pair, L2–L3. While the projected alignments had lower precision and recall than the original system alignments, this is still a good start if we have alignment for the two language pairs and none for the third pair. However, when we intersected the projected alignments with the system alignments, we improved precision by 10 to 20 points for word alignment, and around 10 points for phrase alignment, with the exception of the HeaDAligner, where precision was improved by 30-35 points. This is helpful as there are situations where we would prefer a few really reliable alignments to a larger number,

which contain more errors. The best alignment methods achieved a precision of over 80, and a recall of over 60, at least for the Sophie part of the data.

To explore a small area of syntactic annotation projection we constructed a test collection for prepositional phrase attachment experiments. The test collection consists of 100 triplets, where each triplet contains equivalent phrases in the three languages, taken from the SMULTRON annotation. In at least one of the languages, the phrase fulfils the requirement of being a prepositional phrase with ambiguous attachment, preceded by a verb, and immediately preceded by a noun. In addition to the three gold standard trees, the triplets also contain the automatic parses, which are the structures that we want to improve. The test collection has been subjected to majority vote projection, as well as more linguistically informed correction and projection.

Previous research has shown that important aspects for achieving high-quality annotations from the projection process are the quality of the alignment, the number of source languages, the relationship between source and target languages, and the projection method. Majority vote projections has been the most commonly used method.

We have shown that the majority vote approach gives an error reduction of around 45%, going from about 50% correct attachments in all three languages to more than 70% correct attachments. Using only the linguistic clues reduces the errors by about 10%, and applying the majority vote approach on top of this gives an error reduction of 50%, resulting in 75% correct attachments in all three languages. This holds for the three similar languages English, German, and Swedish, on a small test collection. Further experiments need to be carried out on a larger data set.

As expected, we found that using more languages gives more reliable information for projection. We can also conclude that while majority vote projection is an effective tool to improve the syntactic annotation, using linguistic clues to correct the annotation before applying majority vote projection is even better. There are, however, two important aspects to consider. On the one hand, sometimes a simpler approach with worse results may be acceptable, considering the effort of collecting or extracting linguistic information. On the other hand, the majority vote approach is a blunt tool, which resulted in an increase in errors by 45-55% (27% when applied to the data improved by the linguistic clues) for one of the languages, namely English, the language with the most correct attachments to start with. Finally, we found that there are remaining problems, as some structural errors cannot be corrected by projection at all. The different languages do not necessarily use the same words, or the same syntactic structure, to convey the same meaning.

The final point to take home is the importance of looking at the data. No annotation method, and no quality check, will render perfect annotations. No

gold standard is completely correct, and no annotation scheme shows the true meaning of a sentence. Corpus work always requires putting on glasses, which determine what we can see of the world, and it requires being aware of the shade or colour of these glasses. It is said that sometimes you can't see the forest for the trees. We can paraphrase this and say that with corpus linguistics, and in particular parallel treebanks, sometimes you can't see the trees for the forest. While it is not possible to look at everything in a large parallel treebank, or any large corpus, we can use automatic methods to guide us, to find these interesting issues we as computational linguists are, or should be, interested in.

10.3 Future Work

A number of areas need to be explored further. First, the experiments carried out in Chapter 9 are proof-of-concept, and we therefore need to develop and test quality checking through multilingual annotation projection on a larger scale. This includes using more data for testing, but also expanding the experiments to explore annotation projection for syntax in general, not just PP-attachment disambiguation.

Second, we have assumed perfect alignment for the PP in the experiments. The alignment of the surrounding words and phrases, as well as the alignment quality, is of great importance. Thus, we need to properly evaluate the impact of alignment quality for annotation projection.

Third, the choice of languages is of course a restriction on the set of inferences to be made. Previous research has shown that more diverging languages would give more useful information for some issues, while being too different to be of help for other issues. In this thesis, we have discussed issues particular to English, German, and Swedish. Some of the solutions are general enough to be transferable to other languages, while some of the considerations will look different for other languages. We would like to explore how the information available differs between closely related languages and more diverging languages.

Finally, some structural errors cannot be corrected by projection, meaning that the assumption that annotation variation between languages points to errors does not account for all cases. We thus need to explore additional methods of finding errors for annotation projection.

A. The SMULTRON Treebank in Numbers

		Sentences	Constituents	Tokens
German	Sophie	529	4,839	7,416
	Economy	518	7,139	10,987
English	Sophie	528	8,765	7,829
	Economy	473	7,020	11,626
Swedish	Sophie	536	7,090	7,394
	Economy	494	5,351	9,446

Table A.1: The size of the SMULTRON v1.0 parallel treebank, with numbers for the part taken from *Sophie's World*, and the part taken from economy texts.

		Word		Phrase		Total
		exact	fuzzy	exact	fuzzy	
DE-EN	Sophie	4,322	472	2,165	655	7,614
	Economy	6,901	901	2,649	637	11,091
EN-SV	Sophie	4,292	381	2,387	754	7,814
	Economy	7,139	573	3,819	354	11,885
SV-DE	Sophie	4,485	461	2,461	895	8,303
	Economy	5,464	1,329	3,194	1,135	11,125

Table A.2: The alignment of the SMULTRON v1.0 parallel treebank, with numbers for the part taken from *Sophie's World*, and the part taken from economy texts, for German-English, English-Swedish, and Swedish-German. (The few missing links, comparing word and phrase level to the total count, consist of spurious word-to-phrase links, all exact aligned.)

References

Abeillé, A., editor (2003). *Treebanks: Building and Using Parsed Corpora*. Kluwer Academic Publishers, Dordrecht. 29

Ahrenberg, L. (2007). LinES: An English-Swedish parallel treebank. In *Proceedings of the Nordic Conference on Computational Linguistics (Nodalida)*, Tartu, Estonia. 29

Ahrenberg, L., Merkel, M., Sågvall Hein, A., and Tiedemann, J. (2000). Evaluation of word alignment systems. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 1255–1261. 110, 111

Ahrenberg, L., Merkel, M., Sågvall Hein, A., and Tiedemann, J. (1999). Evaluation of LWA and UWA. Technical Report 15, Department of Linguistics, Uppsala University, Uppsala, Sweden. 111

Artstein, R. and Poesio, M. (2008). Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596. 67, 68

Baker, M. (1993). Corpus linguistics and translation studies: Implications and applications. In Baker, M., Francis, G., and Tognini-Bonelli, E., editors, *Text and Technology: In Honour of John Sinclair*. John Benjamins, Philadelphia and Amsterdam. 36

Beigman, E. and Beigman Klebanov, B. (2009). Learning with annotation noise. In *Proceedings of the Joint Conference of the Annual Meeting of the ACL and the Conference on Natural Language Processing of the AFNLP*, pages 280–287, Suntec, Singapore. 77

Bentivogli, L., Forner, P., and Pianta, E. (2004). Evaluating cross-language annotation transfer in the MultiSemCor corpus. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, page 364, Morristown, NJ, USA. Association for Computational Linguistics. 113

Bentivogli, L. and Pianta, E. (2005). Exploiting parallel texts in the creation of multilingual semantically annotated resources: the MultiSemCor corpus. *Natural Language Engineering*, 11(3):247–261. 113

Bies, A., Ferguson, M., Katz, K., and MacIntyre, R. (1995). Bracketing guidelines for treebank II style, Penn treebank project. Technical report, University of Pennsylvania. 44, 79

Black, E., Abney, S. P., Flickenger, D., Gdaniec, C., Grishman, R., Harrison, P., Hindle, D., Ingria, R., Jelinek, F., Klavans, J. L., Liberman, M., Marcus, M. P., Roukos, S., Santorini, B., and Strzalkowski, T. (1991). A procedure for quantitatively comparing the syntactic coverage of English grammars. In *Proceedings of the Workshop on Speech and Natural Language*, pages 306–311, Pacific Grove, CA, USA. 125, 126

Blaheta, D. (2002). Handling noisy training and testing data. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 111–116, Philadelphia. 78

Bod, R. (1993). Using an annotated corpus as a stochastic grammar. In *Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. 29

Bojar, O. and Prokopová, M. (2006). Czech-English word alignment. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 1236–1239. ELRA. 66

Borin, L. (1999). Pivot alignment. In *Proceedings of the Nordic Conference on Computational Linguistics (Nodalida)*, pages 41–48, Trondheim, Norway. Department of Linguistics, NTNU. 113, 115

Borin, L. (2000). You'll take the high road and I'll take the low road: using a third language to improve bilingual word alignment. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 97–103, Morristown, NJ, USA. Association for Computational Linguistics. 113, 115

Borin, L., Forsberg, M., and Kokkinakis, D. (2010). Diabase: Towards a diachronic BLARK in support of historical studies. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., and Tapias, D., editors, *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, Valletta, Malta. European Language Resources Association (ELRA). 36

Botley, S. P., McEnery, A. M., and Wilson, A., editors (2000). *Multilingual Corpora in Teaching and Research*. Rodopi, Amsterdam, Atlanta GA. 28

Bouma, G., Kuhn, J., Schrader, B., , and Spreyer, K. (2008). Parallel LFG grammars on parallel corpora: A base for practical triangulation. In Butt, M. and King, T. H., editors, *Proceedings of LFG08*, pages 169–189. CSLI Publications. 114, 115, 171

Brants, S., Dipper, S., Hansen, S., Lezius, W., and Smith, G. (2002). The TIGER treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories (TLT)*, pages 24–41, Sozopol, Bulgaria. 44, 79

Brants, T. and Skut, W. (1998). Automation of treebank annotation. In Powers, D. M. W., editor, *Proceedings of the Joint Conference on New Methods in Language Processing and Computational Natural Language Learning (NeMLaP-CoNLL)*, pages 49–57. Association for Computational Linguistics, Somerset, New Jersey. 81

Brown, P. F., Lai, J. C., and Mercer, R. L. (1991). Aligning sentences in parallel corpora. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 169–176, Berkeley, California, USA. Association for Computational Linguistics. 106

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311. 108, 139

Buch-Kromann, M. (2010). Open challenges in treebanking: some thoughts based on the Copenhagen dependency treebanks (invited paper). In *Proceedings of the Workshop on Annotation and Exploitation of Parallel Corpora*, Tartu, Estonia. 65

Buch-Kromann, M. and Korzen, I. (2010). The unified annotation of syntax and discourse in the Copenhagen dependency treebanks. In *Proceedings of the Workshop on Linguistic Annotation at ACL*, Uppsala, Sweden. 30

Burchardt, A., Erk, K., Frank, A., Kowalski, A., Padó, S., and Pinkal, M. (2006). The SALSA corpus: A German corpus resource for lexical semantics. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 969–974, Genoa, Italy. 67

Callison-Burch, C., Talbot, D., and Osborne, M. (2004). Statistical machine translation with word- and sentence-aligned parallel corpora. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 175–182, Barcelona, Spain. 109

Carl, M. and Way, A., editors (2003). *Recent advances in example-based machine translation*. Kluwer Academic Publishers, Dordrecht. 109

Carletta, J. (1996). Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22:249–254. 67

Carroll, J., Briscoe, T., and Sanfilippo, A. (1998). Parser evaluation: a survey and a new proposal. In *Proceedings of the First International Conference on Language Resources and Evaluation (LREC)*, pages 447–454, Granada, Spain. 125, 126

Charniak, E. (1996). Tree-bank grammars. In *Proceedings of the National Conference for the Advancement of Artificial Intelligence (AAAI)*. 29

Charniak, E. (1997). Statistical techniques for natural language parsing. *AI Magazine*, 18(4):33–44. 80

Čmejrek, M., Cuřín, J., and Havelka, J. (2003). Treebanks in machine translation. In *Proceedings of the Workshop on Treebanks and Linguistic Theories (TLT)*, pages 209–212, Växjö, Sweden. 29

Čmejrek, M., Cuřín, J., Havelka, J., Hajič, J., and Kuboň, V. (2004). Prague Czech-English dependency treebank: Syntactically annotated resources for machine translation. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, Lisbon, Portugal. 36

Collins, M. (1999). *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania. 29

Cyrus, L. and Feddes, H. (2004). A model for fine-grained alignment on multilingual texts. In *Proceedings of the Workshop on Multilingual Linguistic Resources at COLING*, Geneva, Switzerland. 29

Davis, P. C. (2002). *Stone soup translation: the linked automata model*. PhD thesis, Ohio State University. 67, 110, 111

DeNero, J. and Klein, D. (2007). Tailoring word alignments to syntactic machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 17–24, Prague, Czech Republic. Association for Computational Linguistics. 140

Dickinson, M. (2006). Rule equivalence for error detection. In *Proceedings of the Workshop on Treebanks and Linguistic Theories (TLT)*, Prague, Czech Republic. 82

Dickinson, M. (2008). Representations for category disambiguation. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 201–208, Manchester. 81, 88, 94

Dickinson, M. and Meurers, W. D. (2003a). Detecting errors in part-of-speech annotation. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 107–114, Budapest, Hungary. 80

Dickinson, M. and Meurers, W. D. (2003b). Detecting inconsistencies in treebanks. In *Proceedings of the Second Workshop on Treebanks and Linguistic Theories (TLT)*, pages 45–56, Växjö, Sweden. 81, 84, 86

Dickinson, M. and Meurers, W. D. (2005a). Detecting errors in discontinuous structural annotation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, Michigan, USA. The Association for Computer Linguistics. 81, 84, 86, 88

Dickinson, M. and Meurers, W. D. (2005b). Prune diseased branches to get healthy trees! How to find erroneous local trees in a treebank and why it matters. In *Proceedings of the Workshop on Treebanks and Linguistic Theories (TLT)*, Barcelona, Spain. 77, 82

Dickinson, M. and Samuelsson, Y. (2010). Consistency checking for treebank alignment. In *Proceedings of the Workshop on Linguistic Annotation (LAW)*, pages 38–46, Uppsala, Sweden. Association for Computational Linguistics. 84

Dorr, B. J., Pearl, L., Hwa, R., and Habash, N. (2002). Improved word-level alignment: Injecting knowledge about MT divergences. Technical Report LAMP-TR-082,CS-TR-4333,UMIACS-TR-2002-15, University of Maryland, College Park. 65

Dyvik, H., Meurer, P., Rosén, V., and De Smedt, K. (2009). Linguistically motivated parallel parsebanks. In Passarotti, M., Przepiórkowski, A., Raynaud, S., and Eynde, F. V., editors, *Proceedings of the Workshop on Treebanks and Linguistic Theories (TLT)*, pages 71–82, Milano, Italy. EDUCatt. 64

Ejerhed, E., Källgren, G., Wennstedt, O., and Åström, M. (1992). The linguistic annotation system of the Stockholm-Umeå corpus project - description and guidelines. Technical report, Department of Linguistics, Umeå University. 36, 39, 118, 121

Fossum, V. and Abney, S. P. (2005). Automatically inducing a part-of-speech tagger by projecting from multiple source languages across aligned corpora. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*, pages 862–873. 114, 115, 171

Fossum, V. and Knight, K. (2008). Using bilingual Chinese-English word alignments to resolve PP-attachment ambiguity in English. In *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA)*. 162

Francis, W. N. (2007). Problems of assembling and computerizing large corpora. In Teubert, W. and Krishnamurthy, R., editors, *Corpus Linguistics: Critical Concepts in Linguistics*, volume I, pages 285–298. Routledge, London & New York. First published in S. Johansson, ed. (1982) Computer Corpora in English Language Research, Bergen: Norwegian Computing Centre for the Humanities, p. 7-24. 36

Francis, W. N. and Kucera, H. (1964). *Brown Corpus Manual of Information*. Department of Linguistics, Brown University. Also available at http://khnt.hit.uib.no/icame/manuals/brown/. 27, 36

Fraser, A. and Marcu, D. (2007). Measuring word alignment quality for statistical machine translation. *Computational Linguistics*, 33:293–303. 111

Gaarder, J. (1991). *Sofies verden: Roman om filosofiens historie*. Aschehoug. 30, 35

Gaarder, J. (1993). *Sofies värld. Roman om filosofins historia*. Rabén och Sjögren. Translator Mona Eriksson. 35

Gaarder, J. (1994). *Sofies Welt. Roman über die Geschichte der Philosophie*. Carl Hanser Verlag. Translator Gabriele Haefs. 35

Gaarder, J. (1995). *Sophie's World. A Novel about the History of Philosophy*. Phoenix House/The Orion Publishing Group. Translator Paulette Møller. 35

Gale, W. A. and Church, K. W. (1991). A program for aligning sentences in bilingual corpora. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 177–184, Berkeley, California, USA. Association for Computational Linguistics. 106

Gale, W. A. and Church, K. W. (1993). A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102. 100, 106

Garside, R., Leech, G., and Váradi, T. (1992). Lancaster Parsed Corpus. A machine-readable syntactically-analysed corpus of 144,000 words. Available for distribution through ICAME, The Norwegian Computing Centre for the Humanities, Bergen. 28

Gellerstam, M. (2007). Fingerprints in translation. In Teubert, W. and Krishnamurthy, R., editors, *Corpus Linguistics: Critical Concepts in Linguistics*, volume IV, pages 352–365. Routledge, London & New York. First published in (2005) In and Out of English: For Better, for Worse?, Multilingual Matters, Toronto: University of Toronto Press, p. 201-213. 36

Gildea, D. (2004). Dependencies vs. constituents for tree-based alignment. In Lin, D. and Wu, D., editors, *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 214–221, Barcelona, Spain. Association for Computational Linguistics. 29

Graca, J., Pardal, J. P., Coheur, L., and Caseiro, D. (2008). Building a golden collection of parallel multi-language word alignment. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, Marrakech, Morocco. 66, 68

Groß, B. (1998). Vergleichende Untersuchung von Alignment-Tools. Saarbrücker Studien zu Sprachdatenverarbeitung und Übersetzen 15, Fachrichtung 8.6 - Angewandte Sprachwissenschaft sowie Übersetzen und Dolmetschen. 106

Groves, D., Hearne, M., and Way, A. (2004). Robust sub-sentential alignment of phrase structure trees. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, Geneva, Switzerland. 109

Gustafson-Čapková, S., Samuelsson, Y., and Volk, M. (2007). SMULTRON (version 1.0) - The Stockholm multilingual parallel treebank. http://www.ling.su.se/dali/research/smultron/index.htm. An English-German-Swedish Parallel Treebank with Sub-sentential Alignments. 35

Habash, N., Gabbard, R., Rambow, O., Kulick, S., and Marcus, M. (2007). Determining case in Arabic: Learning complex linguistic behavior requires complex linguistic features. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1084–1092. 77

Halácsy, P., Kornai, A., and Oravecz, C. (2007). HunPos: an open source trigram tagger. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 209–212, Morristown, NJ, USA. Association for Computational Linguistics. 118

Hall, J. (2008). *Transition-Based Natural Language Parsing with Dependency and Constituency Representations*. PhD thesis, Växjö University, School of Mathematics and Systems Engineering. 30

Hansen-Schirra, S., Neumann, S., and Vela, M. (2006). Multi-dimensional annotation and alignment in an English-German translation corpus. In *Proceedings of the Workshop on Multidimensional Markup in Natural Language Processing (NLPXML) at EACL*, pages 35–42, Trento, Italy. 29

Haulrich, M. (2012). *Data-Driven Bitext Dependency Parsing and Alignment*. PhD thesis, Copenhagen Business School. 24

He, Y. and Kayaalp, M. (2006). A comparison of 13 tokenizers on MEDLINE. Technical Report LHNCBC-TR-2006-003, The Lister Hill national center for biomedical communications, Bethesda, MD. 120

Hearne, M. and Way, A. (2006). Disambiguation strategies for data-oriented translation. In *Proceedings of the Conference of the European Association for Machine Translation (EAMT)*, pages 59–68, Oslo, Norway. 29

Hogan, D. (2007). Coordinate noun phrase disambiguation in a generative parsing model. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 680–687, Prague, Czech Republic. Association for Computational Linguistics. 77

Huang, L., Jiang, W., and Liu, Q. (2009). Bilingually-constrained (monolingual) shift-reduce parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1222–1231, Singapore. Association for Computational Linguistics. 162

Hwa, R., Resnik, P., and Weinberg, A. (2002a). Breaking the resource bottleneck for multilingual parsing. In *Proceedings of the Workshop on Linguistic Knowledge Acquisition and Representation: Bootstrapping Annotated Language Data at (LREC)*, Las Palmas, Canary Islands, Spain. 113, 114

Hwa, R., Resnik, P., Weinberg, A., Cabezas, C., and Kolak, O. (2005). Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, 11(3):311–325. 114

Hwa, R., Resnik, P., Weinberg, A., and Kolak, O. (2002b). Evaluating translational correspondence using annotation projection. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 392–399, Morristown, NJ, USA. Association for Computational Linguistics. 113, 114

Johansson, R. and Nugues, P. (2005). Using parallel corpora for cross-language projection of FrameNet annotation. In *Proceedings of the 1st ROMANCE FrameNet Workshop*, Cluj-Napoca, Romania. 113

Johansson, R. and Nugues, P. (2007). Extended constituent-to-dependency conversion for English. In *Proceedings of the Nordic Conference on Computational Linguistics (Nodalida)*, Tartu, Estonia. 127

Johansson, R. and Nugues, P. (2008). The effect of syntactic representation on semantic role labeling. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 393–400, Manchester, UK. Coling 2008 Organizing Committee. 29

Järborg, J. (1986). SynTag dokumentation. Manual för SynTaggning. Technical report, Department of Swedish, Göteborg University. 44

Kakkonen, T. (2007). *Framework and Resources for Natural Language Parser Evaluation*. PhD thesis, Department of Computer Science and Statistics, University of Joensuu, Finland. 125

Karlsson, F. (1992). SWETWOL: A comprehensive morphological analyser for Swedish. *Nordic Journal of Linguistics*, 15:1–45. Cambridge University Press. 41

Kay, M. and Röscheisen, M. (1988). Text-translation alignment. Technical Report P90-00143, Xerox Palo Alto Research Center. 106

Kay, M. and Röscheisen, M. (1993). Text-translation alignment. *Computational Linguistics*, 19:121–142. 106

Klein, D. and Manning, C. D. (2003). Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems 15 (NIPS)*, pages 3–10. MIT Press. 117

Knutsson, O., Bigert, J., and Kann, V. (2003). A robust shallow parser for Swedish. In *Proceedings of the Nordic Conference on Computational Linguistics (Nodalida)*, Reykjavik, Iceland. 118

Koehn, P. (2002). Europarl: A multilingual corpus for evaluation of machine translation. 35, 150

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics. 139, 146

König, E. and Lezius, W. (2003). The TIGER language - a description language for syntax graphs, formal definition. Technical report, IMS, Universität Stuttgart. 30

Kruijff-Korbayová, I., Chvátalová, K., and Postolache, O. (2006). Annotation guidelines for Czech-English word alignment. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*, pages 1256–1261. 58, 66, 67

Kucera, H. and Francis, W. N. (1967). *Computational Analysis of Present-day American English*. Brown University Press, Providence, RI. 27

Kuhn, J. and Jellinghaus, M. (2006). Multilingual parallel treebanking: a lean and flexible approach. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*, Genoa, Italy. 114, 171

Lambert, P., De Gispert, A., Banchs, R., and Mariño, J. (2005). Guidelines for word alignment evaluation and manual alignment. *Language Resources and Evaluation*, 39:267–285. 10.1007/s10579-005-4822-5. 58, 66

Lavie, A., Parlikar, A., and Ambati, V. (2008). Syntax-driven learning of sub-sentential translation equivalents and translation rules from parsed parallel corpora. In *Proceedings of the Workshop on Syntax and Structure in Statistical Translation (SSST2) at ACL*, pages 87–95, Columbus, OH. 29, 95, 109

Leech, G. and Eyes, E. (1997). Syntactic annotation: treebanks. In Garside, R., Leech, G., and McEnery, T., editors, *Corpus Annotation - Linguistic Information from Computer Text Corpora*, chapter 3, pages 34–52. Addison Wesley Longman Limited. 27, 28

Levin, L. (2011). Variety, idiosyncracy, and complexity in language and language technologies. *Linguistic Issues in Language Technology*, 6. 32

Li, X., Ge, N., Grimes, S., Strassel, S. M., and Maeda, K. (2010). Enriching word alignment with linguistic tags. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., and Tapias, D., editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC)*, Valletta, Malta. European Language Resources Association (ELRA). 66

Liang, P., Taskar, B., and Klein, D. (2006). Alignment by agreement. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, HLT-NAACL '06, pages 104–111, Stroudsburg, PA, USA. Association for Computational Linguistics. 140, 141

Lin, D. (1998). A dependency-based method for evaluating broad-coverage parsers. *Natural Language Engineering*, 4:97–114. 127

Loftsson, H. (2009). Correcting a POS-tagged corpus using three complementary methods. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 523–531, Athens, Greece. 80, 95

Lundborg, J., Marek, T., Mettler, M., and Volk, M. (2007). Using the Stockholm TreeAligner. In *Proceedings of the Workshop on Treebanks and Linguistic Theories (TLT)*, Bergen. 56

Macken, L. (2007). Analysis of translational correspondence in view of sub-sentential alignment. In *Proceedings of the Workshop on New Approaches to Machine Translation at METIS-II*, pages 97–105, Leuven, Belgium. 67

Macken, L. (2010). An annotation scheme and gold standard for Dutch-English word alignment. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., and Tapias, D., editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC)*, pages 3369–3374. European Language Resources Association (ELRA). 58, 67

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1993). Building a large annotated corpus of English: the Penn treebank. *Computational Linguistics*, 19(2):313–330. 28, 35, 39, 117

Mareček, D. (2009). Improving word alignment using alignment of deep structures. In *Proceedings of the 12th International Conference on Text, Speech and Dialogue (TSD2009)*, pages 56–63, Berlin, Heidelberg. Springer-Verlag. 66, 67

McEnery, T. and Wilson, A. (2001). *Corpus Linguistics - An Introduction*. Edinburgh University Press. 27, 28, 106

Megyesi, B. (2009). The open source tagger HunPoS for Swedish. In Jokinen, K. and Bick, E., editors, *Proceedings of the Nordic Conference on Computational Linguistics (Nodalida)*, volume 4 of *NEALT Proceedings Series*, pages 239–241, Odense, Denmark. 118

Megyesi, B., Dahlqvist, B., Csató, Éva. Á., and Nivre, J. (2010). The English-Swedish-Turkish parallel treebank. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., and Tapias, D., editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC)*, Valletta, Malta. European Language Resources Association (ELRA). 30

Melamed, I. D. (1998a). Annotation style guide for the Blinker project. 58

Melamed, I. D. (1998b). Manual annotation of translational equivalence: The Blinker project. Technical Report IRCS 98-07, Department of Computer and Information Science, University of Pennsylvania. 75, 111

Mengel, A. and Lezius, W. (2000). An XML-based encoding format for syntactically annotated corpora. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC)*, pages 121–126, Athens, Greece. 44, 55, 88, 118

Merkel, M. (1999a). Annotation style guide for the PLUG link annotator. Technical report, Linköping University, Linköping, Sweden. 58

Merkel, M. (1999b). *Understanding and Enhancing Translations by Parallel Text Processing*. PhD thesis, Linköping University. 27

Meurers, D. and Müller, S. (2008). Corpora and syntax (Article 44). In Lüdeling, A. and Kytö, M., editors, *Corpus Linguistics. An International Handbook*. Mouton de Gruyter. 77

Mitkov, R. and Barbu, C. (2004). Using bilingual corpora to improve pronoun resolution. *Languages in Contrast*, 4(2):201–211. 115

Nagao, M. (1984). A framework of a mechanical translation between Japanese and English by analogy principle. In *Proceedings of the international NATO symposium on Artificial and human intelligence*, pages 173–180, New York, NY, USA. Elsevier North-Holland, Inc. 109

Nivre, J. (2002). What kinds of trees grow in Swedish soil? A comparison of four annotation schemes for Swedish. In Hinrichs, E. and Simov, K., editors, *Proceedings of the First Workshop on Treebanks and Linguistic Theory (TLT)*, pages 123–138, Sozopol, Bulgaria. 44

Nivre, J. (2008). Treebanks (Article 13). In Lüdeling, A. and Kytö, M., editors, *Corpus Linguistics. An International Handbook*. Mouton de Gruyter. 28

Nivre, J., de Smedt, K., and Volk, M. (2005). Treebanking in Northern Europe: A white paper. In Holmboe, H., editor, *Nordisk Sprogteknologi. Årbog for Nordisk Sprogteknologisk Forskningsprogram 2000-2004*. Museum Tusculanums Forlag, Copenhagen. 23, 28, 29

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryiğit, G., Kübler, S., Marinov, S., and Marsi, E. (2007). MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 2(13):95–135. 118

Nivre, J. and Megyesi, B. (2007). Bootstrapping a Swedish treebank using cross-corpus harmonization and annotation projection. In *Proceedings of the Sixth Workshop on Treebanks and Linguistic Theories (TLT)*, pages 97–102. 121

Nord, C. (1997). *Translating as a Purposeful Activity. Functionalist Approaches Explained*. St Jerome, Manchester. 36, 37

Och, F. J. and Ney, H. (2000a). A comparison of alignment models for statistical machine translation. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 1086–1090, Morristown, NJ, USA. Association for Computational Linguistics. 66, 111

Och, F. J. and Ney, H. (2000b). Improved statistical alignment models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 440–447, Hongkong, China. 139

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51. 87, 108, 139, 140

Okita, T. (2009). Data cleaning for word alignment. In *Proceedings of the Student Research Workshop at ACL-IJCNLP*, pages 72–80, Suntec, Singapore. 77, 94

Padó, S. (2007). Translational equivalence and cross-lingual parallelism: The case of FrameNet frames. In *Proceedings of the Workshop on Building Frame Semantics Resources for Scandinavian and Baltic Languages at Nodalida*, Tartu, Estonia. 113

Padro, L. and Marquez, L. (1998). On the evaluation and comparison of taggers: the effect of noise in testing corpora. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the Conference on Computational Linguistics (ACL-COLING)*, pages 997–1002, San Francisco, California. 77

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2001). Bleu: a method for automatic evaluation of machine translation. Technical Report RC22176 (W0109-022), IBM Research Division. 107, 111

Petrov, S., Das, D., and McDonald, R. (2012). A universal part-of-speech tagset. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*. 40

Rafferty, A. N. and Manning, C. D. (2008). Parsing three German treebanks: Lexicalized and unlexicalized baselines. In *Proceedings of the Workshop on Parsing German (PaGe2008)*, pages 40–46, Morristown, NJ, USA. Association for Computational Linguistics. 117

Rehbein, I. and van Genabith, J. (2007). Evaluating evaluation measures. In Nivre, J., Kaalep, H.-J., Muischnek, K., and Koit, M., editors, *Proceedings of the Nordic Conference on Computational Linguistics (Nodalida)*, pages 372–379, Tartu, Estonia. 126, 127

Riloff, E., Schafer, C., and Yarowsky, D. (2002). Inducing information extraction systems for new languages via cross-language projection. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics. 114

Rios, A., Göhring, A., and Volk, M. (2009). A Quechua-Spanish parallel treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories (TLT)*, Groningen, the Netherlands. 29

Rosén, V. and De Smedt, K. (2007). Theoretically motivated treebank coverage. In Nivre, J., Kaalep, H.-J., Muischnek, K., and Koit, M., editors, *Proceedings of the Nordic Conference on Computational Linguistics (Nodalida)*, pages 152–159, Tartu. University of Tartu. 83

Rosén, V., De Smedt, K., and Meurer, P. (2006). Towards a toolkit linking treebanking to grammar development. In *Proceedings of the Workshop on Treebanks and Linguistic Theories (TLT)*, pages 55–66, Prague, Czech Republic. 83

Saers, M. (2011). *Translation as Linear Transduction: Models and Algorithms for Efficient Learning in Statistical Machine Translation*. PhD thesis, Uppsala University, Department of Linguistics and Philology. 110

Sampson, G. (2000). A proposal for improving the measurement of parse accuracy. *International Journal of Corpus Linguistics*, 5:53–68. 126

Sampson, G. and Babarczy, A. (2003). A test of the leaf-ancestor metric for parse accuracy. *Natural Language Engineering*, 9:365–380. 126

Samuelsson, Y. (2004). Parallel phrases - going automatic. Further experiments towards a German-Swedish parallel treebank. Master's thesis, Stockholm University. 142, 143

Samuelsson, Y. and Volk, M. (2004). Automatic node insertion for treebank deepening. In *Proceedings of the Workshop on Treebanks and Linguistic Theories (TLT)*, Tübingen, Germany. 45, 48

Samuelsson, Y. and Volk, M. (2006). Phrase alignment in parallel treebanks. In *Proceedings of the Workshop on Treebanks and Linguistic Theories (TLT)*, Prague, Czech Republik. 58, 68

Samuelsson, Y. and Volk, M. (2007a). Alignment tools for parallel treebanks. In *Proceedings of GLDV Frühjahrstagung 2007*, Tübingen, Germany. 58, 82, 105, 152

Samuelsson, Y. and Volk, M. (2007b). Automatic phrase alignment: Using statistical n-gram alignment for syntactic phrase alignment. In *Proceedings of the Workshop on Treebanks and Linguistic Theories (TLT)*, Bergen, Norway. 58, 146, 150

Schwartz, L., Aikawa, T., and Quirk, C. (2003). Disambiguation of English PP attachment using multilingual aligned data. In *Proceedings of Machine Translation Summit IX*. 162

Sennrich, R. and Volk, M. (2010). MT-based sentence alignment for OCR-generated parallel texts. In *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA)*, Denver, Colorado. 107

Sennrich, R. and Volk, M. (2011). Iterative, MT-based sentence alignment of parallel texts. In *Proceedings of the Nordic Conference on Computational Linguistics (Nodalida)*, Riga, Latvia. 107

Simard, M. (1999). Text-translation alignment: Three languages are better than two. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP-VLC)*, pages 2–11, University of Maryland, MD. 107, 112

Simard, M., Foster, G. F., and Isabelle, P. (1992). Using cognates to align sentences in bilingual corpora. In *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 67–81, Montreal, Canada. 106

Skut, W., Krenn, B., Brants, T., and Uszkoreit, H. (1997). An annotation scheme for free word order languages. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, pages 88–95, Washington, DC. 44, 79, 117

Snyder, B. and Barzilay, R. (2008). Cross-lingual propagation for morphological analysis. In *Proceedings of the National Conference for the Advancement of Artificial Intelligence (AAAI)*, pages 848–854. 115

Spreyer, K. (2007). Projecting temporal annotations across languages. Diploma thesis, Saarland University, Saarbrücken, Germany. 115

Spreyer, K. (2011). *Does it have to be trees? Data-driven dependency parsing with incomplete and noisy training data*. PhD thesis, University of Potsdam, Potsdam, Germany. 114

Teleman, U. (1974). *Manual for grammatisk beskrivning av talad och skriven svenska*. Studentlitteratur, Lund. 28, 44, 118

Thielen, C., Schiller, A., Teufel, S., and Stöckert, C. (1999). Guidelines für das Tagging deutscher Textkorpora mit STTS. Technical report, IMS, Stuttgart and SfS, Tübingen. 39

Tiedemann, J. (2003). *Recycling Translations - Extraction of Lexical Data from Parallel Corpora and their Application in Natural Language Processing*. PhD thesis, Uppsala university. 27, 55, 108, 110, 111

Tiedemann, J. (2010). Lingua-Align: An experimental toolbox for automatic tree-to-tree alignment. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, Valetta, Malta. 29, 109, 150, 151

Tiedemann, J. (2011). *Bitext Alignment*. Number 14 in Synthesis Lectures on Human Language Technologies, Graeme Hirst (ed). Morgan & Claypool Publishers. 105

Tiedemann, J. and Kotzé, G. (2009a). Building a large machine-aligned parallel treebank. In Passarotti, M., Przepiórkowski, A., Raynaud, S., and Eynde, F. V., editors, *Proceedings of the Workshop on Treebanks and Linguistic Theories (TLT)*, pages 197–208. EDUCatt, Milano/Italy. 30, 151

Tiedemann, J. and Kotzé, G. (2009b). A discriminative approach to tree alignment. In Ilisei, I., Pekar, V., and Bernardini, S., editors, *Proceedings of the Workshop on Natural Language Processing Methods and Corpora in Translation, Lexicography, and Language Learning at RANLP*, pages 33 – 39, Borovets, Bulgaria. Association for Computational Linguistics. 109, 151

Tinsley, J. (2010). *Resourcing Machine Translation with Parallel Treebanks*. PhD thesis, Dublin City University, School of Computing. 29

Tinsley, J., Hearne, M., and Way, A. (2007a). Exploiting parallel treebanks to improve phrase-based statistical machine translation. In *Proceedings of the Workshop on Treebanks and Linguistic Theories (TLT)*, pages 175–187, Bergen, Norway. 29

Tinsley, J., Zhechev, V., Hearne, M., and Way, A. (2007b). Robust language pair-independent sub-tree alignment. In Maegaard, B., editor, *Proceedings of Machine Translation Summit XI*, pages 467–474, Copenhagen, Denmark. European Association for Machine Translation. 95

Uchimoto, K., Zhang, Y., Sudo, K., Murata, M., Sekine, S., and Isahara, H. (2004). Multilingual aligned parallel treebank corpus reflecting contextual information and its applications. In *Proceedings of the Workshop on Multilingual Linguistic Ressources at COLING*, pages 63–70, Geneva, Switzerland. Association for Computational Linguistics. 30

Ule, T. and Simov, K. (2004). Unexpected productions may well be errors. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, Lissabon, Portugal. 78, 81

van Halteren, H. (2000). The detection of inconsistency in manually tagged text. In Abeillé, A., Brants, T., and Uszkoreit, H., editors, *Proceedings of the Workshop on Linguistically Interpreted Corpora (LINC2000)*, Luxembourg. 80

Véronis, J. (1998). ARCADE - tagging guidelines for word alignment. 58

Véronis, J. (2000). From the Rosetta stone to the information society. In Véronis, J., editor, *Parallel Text Processing - Alignment and Use of Translation Corpora*, chapter 1, pages 1–24. Kluwer Academic Publishers. 27, 28

Volk, M. (1999). Choosing the right lemma when analysing German nouns. In *Multilinguale Corpora: Codierung, Strukturierung, Analyse. 11. Jahrestagung der GLDV*, pages 304–310, Frankfurt. Enigma Corporation. 42

Volk, M. (2006). How bad is the problem of PP-attachment?: a comparison of English, German and Swedish. In *Proceedings of the Third ACL-SIGSEM Workshop on Prepositions*, Prepositions '06, pages 81–88, Stroudsburg, PA, USA. Association for Computational Linguistics. 168

Volk, M., Göhring, A., Marek, T., and Samuelsson, Y. (2010). SMULTRON (version 3.0) - the Stockholm multilingual parallel treebank. http://www.cl.uzh.ch/research/paralleltreebanks_en.html. An English-French-German-Spanish-Swedish parallel treebank with sub-sentential alignments. 35

Volk, M., Gustafson-Čapková, S., Lundborg, J., Marek, T., Samuelsson, Y., and Tidström, F. (2006). XML-based phrase alignment in parallel treebanks. In *Proceedings of the Workshop on Multi-dimensional Markup in Natural Language Processing at EACL*, Trento, Italy. 56, 88

Volk, M., Lundborg, J., and Mettler, M. (2007). A search tool for parallel treebanks. In *Proceedings of the Linguistic Annotation Workshop (LAW) at ACL*, pages 85–92, Prague, Czech Republic. Association for Computational Linguistics. 57

Volk, M., Marek, T., and Samuelsson, Y. (2008). Human judgements in parallel treebank alignment. In *Proceedings of the Workshop on Human Judgements in Computational Linguistics at COLING*, pages 51–57, Manchester, UK. Coling 2008 Organizing Committee. 58, 71

Volk, M. and Samuelsson, Y. (2004). Bootstrapping parallel treebanks. In *Proceedings of the International Workshop on linguistically Interpreted Corpora (LINC) at COLING*, pages 63–69, Geneva, Switzerland. 44

Wu, D. (1997). Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23:377–403. 95, 109, 110, 150

Wu, D. (2010). Alignment. In Indurkhya, N. and Damerau, F., editors, *Handbook of Natural Language Processing*, chapter 16, pages 367–408. CRC Press, second edition. 60, 105, 107, 109

Yarowsky, D. and Ngai, G. (2001). Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1–8, Morristown, NJ, USA. Association for Computational Linguistics. 112, 113, 115

Yarowsky, D., Ngai, G., and Wicentowski, R. (2001). Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the Human Language Technology Conference (HLT)*, pages 1–8, Morristown, NJ, USA. Association for Computational Linguistics. 112, 113, 115

Zanettin, F. (2007). Parallel corpora in translation studies. Issues in corpus design and analysis. In Teubert, W. and Krishnamurthy, R., editors, *Corpus Linguistics: Critical Concepts in Linguistics*, volume IV, pages 285–298. Routledge, London & New York. First published in M. Olohan, ed. (2000) Intercultural fault lines, Manchester: St Jerome Publishing, p. 105-118. 36

Zhechev, V. (2009). *Automatic Generation of Parallel Treebanks. An Efficient Unsupervised System*. PhD thesis, School of Computing, Dublin City University, Dublin, Ireland. 61, 150

Zhechev, V. and Way, A. (2008). Automatic generation of parallel treebanks. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 1105–1112, Manchester, UK. 99, 109