Universität Zürich
Computerlinguistik

Spanish Expansion of a Parallel Treebank

Spanische Erweiterung einer Parallelen Baumbank

Lizentiatsarbeit der Philosophischen Fakultät der Universität Zürich

Referent: Prof. Dr. M. Volk

Verfasserin:
Anne Göhring
Dienerstrasse 81
8004 Zürich

November 14, 2009

Abstract

Parallel treebanks are an important ressource for improving and evaluating machine translation systems. To integrate Spanish into SMULTRON, a manually revised English-German-Swedish parallel treebank, adequate annotation schemes and guidelines are evaluated. A multilingual parallel treebank based on user manuals has been built. Quality control and process optimisation have been the central issues in building this new treebank. The expansion of SMULTRON has proven the practical validity of the annotation schemes.

Zusammenfassung

Parallele Baumbanken sind eine wichtige Ressource für die maschinelle Übersetzung. Die Integration von Spanisch in SMULTRON, eine manuell revidierte parallele Baumbank, erfordert ein passendes Annotationsschema und entsprechende Annotationsrichtlinien. Im Rahmen dieser Arbeit wurde eine multilinguale parallele Baumbank erstellt, die SMULTRON um eine weitere Sprache und die Textsorte der Bedienungsanleitungen erweitert. Qualitäts- und Optimierungs-Überlegungen begleiten den Aufbauprozess der neuen Baumbank. Die erfolgte Erweiterung zeigt die praktische Gültigkeit der angewandten Annotationsschemata.

Acknowledgements

I want to express my gratitude to Prof. Dr. Martin Volk for his support, advices and enthusiasm, in the course of my master thesis project. Thanks to him I discovered the fascinating world of parallel treebanks.

I want to thank the annotators for their excellent work. I learned much on practical and theoretical linguistics from the team discussions we had about the best way(s) to annotate the corpora.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

SMULTRON is the name of the existing parallel treebank developed at Stockholm University that is to be expanded with Spanish texts; the acronym stands for "**S**tockholm **MUL**tilingual **TR**eebank" and *smultron* is also the Swedish word for the wild strawberry. The field of this project has only a remote relation to Nature's biological or geographical products[1]; the natural environment in which it is embedded is the multilingual world we are living in, the human language and our capacity to reflect on it, to model it, and to process the natural language products with the help of computers.

The motivation to expand an existing treebank with Spanish lies equally in my curiosity for the Spanish language as well as my interest for participating in a promising computational linguistics field. Searching for a convenient title I tried some online machine translation services and tested possible title variants, their translations and translation's translations. Just to have a first glimpse, here are some results:

> DE : Spanische Erweiterung einer parallelen Baumbank
> DE → EN : Spanish Extension of a parallel treebank
> EN ↔ DE : Spanisch Verlängerung eines parallelen Baumbank
> (from google.com/translate_t)
>
> DE → EN : Spanish expansion of a parallel tree bench
> DE → FR : Elargissement espagnol d'un banc d'arbre parallèle
> (from linguatec.de)

It broadly confirmed the idea the general public has about automatic translations: it doesn't work well, or at the best –like some self-ironical computational linguists think– it is a great source of entertainment! This was reason enough for me, if any needed, to try to contribute –modestly but with much enthusiasm and without

[1] the wild strawberry or the garden of the English countryside where Sampson (2003: 23) sets the "opening days of [his] career as a producer of natural-language parse trees"

losing good humour– to the construction of future machine translation.

1.2 Project Definition

The first idea for this project was to gather the information and tools necessary to build a Spanish treebank. At the same time, we[2] would build parallel treebanks in three other languages already present in SMULTRON: English, German and Swedish. Starting from the paper multilingual text versions up to the fully annotated parallel treebanks, the task was to first select the corpus, annotation schemes, guidelines, and tools; then put the pieces together for processing the textual data, and, parallel to the annotation process, write Spanish annotation guidelines and adapt them after consulting the student annotators we contracted to do the job; finally, check the annotation consistency, improve iteratively the quality of both the data and the guidelines, and document the process and the resulting product. The immediate goal was to annotate more sentences from another text type –user manuals– and another language: Spanish.

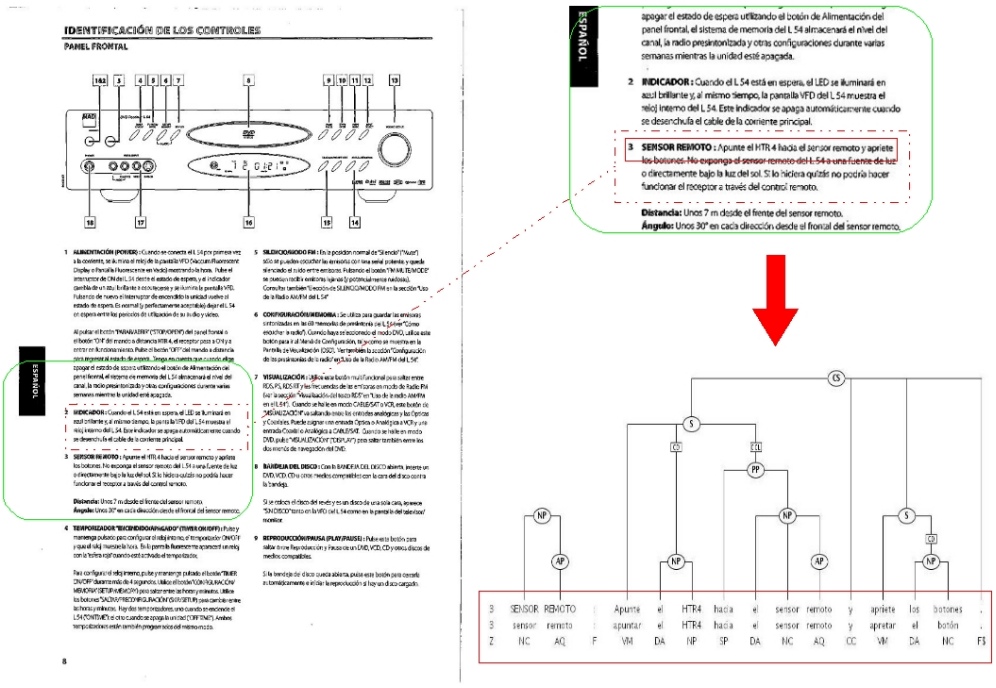


Figure 1: From a Spanish DVD user manual page to an annotated sentence

[2]The pronoun *we* is either used to imply the reader or, like in this case, when at least two persons involved in this project are meant, and not as a "plural majestatis" for the author alone; in this case the pronoun is simply *I* and implies only the author's own responsibility

Why did we choose user manuals? We looked for a third kind of text that would be 1) different from the first ones, that consist of economy and fiction, almost essay texts; and 2) available in at least all four languages we wanted to analyse: English, Swedish, German, and Spanish. This genre and domain diversification allows testing the annotation tools, schemes and guidelines on a better balanced resource.
The intermediate goal was to provide a test data set for the automatic alignment software developed during the same period by a fellow student (see master thesis of Sandra Roth (2009)). The final goal was to have fully aligned high quality parallel treebanks in four languages to be offered to the research community to investigate various subjects such as multilinguality, translation equivalence, contrastive studies, and the like. The long-term goal is to contribute to the development of better machine translation systems.

Three leading research questions guided my work:

1. What are the challenges in integrating a new language into an existing parallel treebank?
2. How can we evaluate the quality of monolingual treebanks and parallel treebank alignments?
3. How can we streamline and optimise the process of building parallel treebanks?

These questions accompany us in the exploration of parallel treebanks and shall be answered at the end of this thesis.

1.3 Thesis Structure

In this first chapter I have exposed the personal motivation, the project outline and the central research questions. Chapter 2 introduces general notions of treebanks, lists some treebank examples, and explains their possible uses. Chapter 3 details the building steps of the Spanish treebank. Chapter 4 discusses general aspects of aligning and describes a first experiment with automatic alignment. Chapter 5 focuses on project statistics, total and specific Spanish treebank statistics, quantitative and qualitative linguistic results. The sixth and last chapter sums up the project progress and the state currently reached, comments on the main experiments and experiences made during the project and the personal insights gained through it, points out the planned extensions, and sketches possible future applications.

2 Parallel Treebanks

This chapter begins with the definition of treebanks, presents annotation principles, gives a brief historical overview of monolingual and multilingual treebanks, lists some well-known examples, and introduces the notion of parallel treebank. A second section presents in more details the two treebanks at the origin of this project: SMULTRON and AnCora-Es. The last section sketches the possible uses of treebanks: in applications like machine translation or in more theoretical fields like contrastive linguistic studies.

2.1 Treebanks

What is a treebank[1]? The corresponding entry in the *Glossary of Corpus Linguistics* (Baker et al. 2006) says:

treebank A corpus that has been grammatically annotated in order to identify and label different constituent structures or phrases. Because of the system of labelling, such structures are sometimes referred to as 'trees'.

This definition does not explicitly state that treebanks have to follow a certain annotation formalism. Nevertheless, it contains –in my opinion– a certain bias: it suggests a preference for constituency over dependency annotation. The decision between these two or more annotation formalisms is often merely pragmatic: there are some existing resources (data and tools) available to the group as well as a certain amount of past expertise working with these resources. Sometimes –but rarely, I guess– it mirrors an almost religious conviction about the rightness of one or the other linguistic theory. In some cases, where the treebank aims to apply a certain theory, the annotation choice corresponds to the underlying theory. But without this theoretical objective, the arguments for and against one or another annotation can be reduced to practical considerations and personal preferences. As Abeillé (2003)

[1] According to Sampson (2003: 24) "the term 'treebank', now in standard international use in this sense [structurally analysed sample of natural language], was first coined by Geoffrey Leech" in 1983.

points out, the argument in favour of the constituency annotation is its greater readability compared to dependency annotation because it keeps the word order. On the negative side, constituency is less flexible than dependency annotation, and tends to introduce many intermediate levels, increasing thus the structural complexity. To take advantage of the constituency and dependency approaches, some treebanks combine both aspects of syntactic category and function in a hybrid representation model. All our treebanks are built upon this hybrid model, following the examples of the Penn Treebank II, Negra, and Tiger treebanks (see section 2.2).

Orthogonal to the choice of the annotation type, the treebanks differ in the levels of annotation and in the details at each level. Most treebanks typically include, next to the original words and other tokens, part-of-speech tags, some lemma form of the words, syntactical units (chunks, flat or recursive phrases), some grammatical functions (subject, predicate, complement, etc.), and morphological information. But before introducing the different sorts of treebanks, let us look at the concept of annotation.

2.1.1 Annotation

Annotation describes different types of information about a certain dataset; this information is broadly divided in content and context information. Context information includes information external to the dataset itself, e.g. the source, the date of record or publication, the authors or other actors, etc. Content annotation is the explicit description of information already present –more or less implicitly– in the raw material. However, it is important to keep in mind that annotation implies a certain degree of interpretation. This is certainly true for linguistic annotation. As comparison, the annotation of text structure (titles, paragraphs) is less a matter of interpretation, though it still requires careful attention. Finally, annotation depends on the medium: there is no prosodic information to annotate in a written text; gestic information is only present in visual media!

Table 1[2] shows the possible linguistic annotation layers for texts[3], from the most basic type of linguistic corpus annotation, the part-of-speech tagging (POS tagging, also known as grammatical tagging or morphosyntactic annotation), to further forms of annotation including syntactic parsing, semantic disambiguation à la WordNet (WN) or semantic role annotation in the style of FrameNet (FN).

[2]inspired from (Lemnitzer and Zinsmeister 2006: 64)

[3]In the following, *annotation* will refer to *linguistic annotation of written text* unless otherwise specified.

Linguistic level	Annotation	Example
Morphosyntax	part of speech	POS tags
Morphology	flexion lemmatisation	morph tags (POS tags) lemmas
Syntax	phrase constituency phrase dependency sentence order	phrase categories dependency labels topological fields
Semantics	named entities word senses roles, frames	POS tags WN synset, POS tags role tags, FN frames
Pragmatics	coreference discourse	anaphor links argument chains

Table 1: Text linguistic annotation layers

Note that part-of-speech tagging is also a form of semantic annotation, identifying named entities through appropriate proper name tag and some word senses.[4] In general, POS tags encode distinct criteria apart from the classic part-of-speech categories, typically a selection of the following features:

- surface and syntactic distribution: position and function
- morphological features: gender, number, person, mode, etc.
- semantic feature: named entities (NE) and NE types

The degree of annotation details may also differ from one treebank to the other, e.g. the depth of syntactical analysis using shallow or deep parsing. The granularity of the tagsets and the depth of annotation depend on several factors: the underlying linguistic theory, the intended uses, the compatibility to existing projects, the tools at hand coupled with the available human resources and expertise.

Leech's seven maxims of annotation Some general annotation principles were formulated by Leech (1993) in his article "Corpus annotation schemes":[5]

1. It should be possible to remove the annotation from an annotated corpus in

[4]POS tags partly disambiguate the word senses of homographs, i.e. only if they have different parts-of-speech.

[5]The formulation of these maxims is copied from the lecture notes of M. Dickinson (Department of Linguistics, Indiana University, Autumn 2008): `http://jones.ling.indiana.edu/~mdickinson/08/700/slides/annotation-2x3.pdf` (last visit: October 6, 2009).

order to revert to the raw corpus.

2. It should be possible to extract the annotations by themselves from the text.
3. The annotation scheme should be based on guidelines which are available to the end user.
4. It should be made clear how and by whom the annotation was carried out.
5. The end user should be made aware that the corpus annotation is not infallible, but simply a potentially useful tool.
6. Annotation schemes should be based as far as possible on widely agreed and theory-neutral principles.
7. No annotation scheme has the a priori right to be considered as a standard. Standards emerge through practical consensus.

These principles are meant to "maximise the usability and interchangeability of annotated corpora", although McEnery and Wilson (2001: 34) note that "there is scope for considerable variation in what distinction are made within the annotations which are applied". We will see the specific annotation characteristics of each treebank as they are presented in section 2.2 and in chapter 3 about our Spanish DVD treebank.

There are two more notes left about annotations. First, the annotation issues I talked about in this section concerne mainly linguistic annotations; as already mentioned above, there are other annotation types that also capture textual and extra-textual information like text structure and identification features. Some of the seven principles presented above apply to annotations in general, though I think they more specifically aim the linguistic annotations.

As a final note to this section, the definition of the treebank does not imply the way it is annotated, if the underlying corpus gets *automatically* or *manually* analysed; the stricter definition I follow here is that of Martin Volk, who states that a treebank must be at least manually checked –and probably corrected– for there are no automatic methods yet that would ensure the high quality we strive for.

2.1.2 Towards Treebanks: From Plain to Annotated Corpora

The first examples of the so-called "first generation corpora" date from the early 1960s and include the now famous **Brown** Corpus of American English and the Lancaster-Oslo/Bergen (**LOB**) corpus of British English. These corpora were further developed and transformed in the late 1980s into treebanks: the Brown Corpus

as a part of the Penn Treebank and two treebanks derived from LOB corpus subsamples.

A brief historical background of corpora will lead us to the history of treebanks. McEnery and Wilson (2001: 202–208) list in an appendix the corpora and treebanks mentioned in the textbook. The corpora appear under categories according to the finiteness (stable vs. monitor[6] corpus), the channel (mixed, spoken, written), the time aspect and period covered (diachronic or synchronic, historical or current period), the multilinguality. Here are a few chosen examples from these –not exclusive but overlapping– categories:

The **Bank of English** is a *monitor* corpus launched in 1991 by Collins and the University of Birmingham. According to Collins' website[7], "the corpus contains 524 million words and it continues to grow with the constant addition of new material". It is composed of contemporary spoken and written British and American English texts, is POS tagged and parsed using the Helsinki English constraint grammar; however, on the Collins WordbanksOnline English corpus sampler, only the part-of-speech can be searched online[8].

ARCHER[9] stands for "A Representative Corpus of *Historical* English Registers" and contains British and American English texts written from 1650 to 1990; it is POS tagged. First constructed by Douglas Biber and Edward Finegan in the 1990s, it has been further developed, and the latest version is being coordinated from Manchester University.

An early example of a big monolingual corpus is the British National Corpus (**BNC**)[10] with its approximately 100 million words of *spoken and written* British English; it is POS tagged and a subset of 1 million is skeleton[11] parsed.

The **London-Lund Corpus** consists of half a million words of *spoken* British English collected in the 1960s-70s and prosodically annotated.

As mentioned above, the **Brown Corpus** and the Lancaster-Oslo/Bergen (**LOB**) Corpus both contain one million words of *written* American resp. British English texts dating from 1961. The **Lancaster Parser Corpus** and the **Lancaster-**

[6] McEnery and Wilson (2001: 199) define *monitor corpus* as "a growing, non-finite collection of texts, of primary use in lexicography".

[7] `http://www.collins.co.uk/books.aspx?group=153`

[8] `http://www.collins.co.uk/Corpus/CorpusSearch.aspx`

[9] `http://www.llc.manchester.ac.uk/research/projects/archer/`

[10] `http://www.natcorp.ox.ac.uk/`

[11] McEnery and Wilson (2001: 199) define *skeleton parsing* as "a form of grammatical analysis which represents only a basic subset of the grammatical relationships within a sentence".

Leeds Treebank were built on subsamples of the LOB Corpus, which is on his part only POS tagged. The **SUSANNE** (Surface and underlying structural analysis of natural English) Corpus, developed by Geoffrey Sampson and first released in 1992, is a POS tagged, parsed and lemmatised subsample of the Brown Corpus.

The last category includes *multilingual* corpora to which belong the **Canadian Hansard Corpus** on French-English parliament proceedings or the **CRATER** corpus. This latter trilingual parallel corpus of French, English and Spanish on texts from the International Telecommunication Union (ITU) contains 1 million words in each corpus and is lemmatised, POS-tagged, and sentence-aligned. I mention CRATER here because this corpus plays an important role in the annotation of our Spanish corpus, more precisely as the training corpus used by the POS tagger (see section 3.1.5).

All the monolingual examples presented so far are corpora of English, being Indian, Australian, British, American or New Zealand English. The first corpora in other languages are more difficult to trace. I searched for the first representative examples of corpora and treebanks in the four languages we are interested in. To this end I consulted Wikipedia to check what had already been written in this online encyclopedia on the subject. There are 29 languages for which one or more treebanks[12] are listed, including old languages, like Latin and Ancient Greek, and more "exotic" languages like Turkish, Semitic languages like Arabic and Hebrew, Asian languages like Chinese, Korean, or Thai, besides the usual European languages like the four present in our multilingual treebank project.[13]

2.1.3 Monolingual Treebanks

In this section I will first show examples of monolingual treebanks in all four languages included in our parallel treebank. To complete the overview I will briefly present an example of a dependency treebank. A last side note to semantic annotation will end the section.

English

A famous representative is the **Penn Treebank**[14], built at the University of Pennsylvania and based on the Brown corpus among other material. It has been devel-

[12]Note that I have not verified if the listed treebanks all satisfy our strict definition of treebank.
[13]`http://en.wikipedia.org/wiki/Treebank` (last visit: April 7, 2009)
[14]`http://www.cis.upenn.edu/~treebank/`

oped in three phases, following evolving annotation schemes –called Treebank I and II bracketing styles– and growing in size (see Table 2).

Phase	I	II	III
Period	1989-1992	1993-1995	1996-2000
Sources	Brown corpus	+ WallStreet Journal	+ Switchboard
	Dow Jones newswire		
	Energy Dept abstracts		
	MUC, ATIS, IBM manual		
Mio of words	3	>1	
Bracketing style	I	II	II
Annotation elements	POS, CAT	+FUNC, +null, +traces	

Table 2: Development phases of the Penn Treebank

The example sentence "The safety and operating instructions should be retained for future reference ." taken from the English DVD treebank illustrates the Penn Treebank bracketing II style:

```
( (S
     (NP-SBJ-1 (DT The) (NN safety) (CC and) (NN operating) (NNS instructions) )
     (VP (MD should)
       (VP (VB be)
         (VP (VBN retained)
           (NP (-NONE- *-1) )
           (PP-CLR (IN for)
              (NP (JJ future) (NN reference) )))))
     (. .) ))
```

At the top of the structure, the so-called "root" node **S** dominates two non-terminal nodes, a nominal phrase **NP** and a verb phrase **VP**, as well as one terminal node, a punctuation token with its POS label (both represented by a period). The indentations represent the tree structure levels; in this example, the maximum depth is 6: `S-VP-VP-VP-PP-NP`. The movement of the patient object of the verb *retain* to the subject position in the passive sentence is annotated: a null element is inserted `(NP (-NONE- *-1) )` and a reference number (here: `1`) connects both elements `(NP-SBJ-1 ... )`.

German

As a descendant of the Penn Treebank, but with notable innovations, the **TIGER Treebank** is, along with the **NEGRA** corpus, one of the most important tree-

banks for German. It contains 50,000 sentences from the Frankfurter Rundschau, annotated in a PennTreebank-II-like manner but using another tagset (STTS) and following other guidelines (NEGRA/TIGER), e.g. building "flatter" trees and allowing crossing branches instead of inserting null elements and traces. This innovating feature implies a more complex representation model that in turn adds up to a more complex structure handling: from the mathematically pure tree to the more general directed acyclic graph (DAG). The other conceptual revolution is the possibility to define secondary edges, i.e. to add another annotation layer on top of the constituent structure. To express this major complexity with greater flexibility, another format is used as shown in the following TIGER example sentence and the corresponding tree[15] in Figure 2.

```
<s id="s1376">
 <graph root="s1376_VROOT" discontinuous="true">
  <terminals>
    <t id="s1376_1" word="Auf" lemma="auf" pos="APPR" morph="--" />
    <t id="s1376_2" word="Armeeseite" lemma="Armeeseite" pos="NN" morph="Dat.Sg.Fem" />
    <t id="s1376_3" word="seien" lemma="sein" pos="VAFIN" morph="3.Pl.Pres.Subj">
        <secedge label="HD" idref="s1376_508" />
    </t>
    <t id="s1376_4" word="35" lemma="35" pos="CARD" morph="--" />
    <t id="s1376_5" word="Soldaten" lemma="Soldat" pos="NN" morph="Nom.Pl.Masc" />
    <t id="s1376_6" word="getötet" lemma="töten" pos="VVPP" morph="Psp" />
    <t id="s1376_7" word="und" lemma="und" pos="KON" morph="--" />
    <t id="s1376_8" word="mindestens" lemma="mindestens" pos="ADV" morph="--" />
    <t id="s1376_9" word="200" lemma="200" pos="CARD" morph="--" />
    <t id="s1376_10" word="weitere" lemma="weit" pos="ADJA" morph="Comp.Nom.Pl.Masc" />
    <t id="s1376_11" word="verwundet" lemma="verwunden" pos="VVPP" morph="Psp" />
    <t id="s1376_12" word="worden" lemma="werden" pos="VAPP" morph="Psp">
        <secedge label="HD" idref="s1376_507" />
    </t>
    <t id="s1376_13" word="." lemma="--" pos="$." morph="--" />
  </terminals>
  <nonterminals>
    <nt id="s1376_500" cat="PP">
      <edge label="AC" idref="s1376_1" />
      <edge label="NK" idref="s1376_2" />
      <secedge label="MO" idref="s1376_503" />
    </nt>
    ...
```

[15] In general, the term "tree" is used even if the structures are strictly speaking DAGs and not trees anymore; I will follow this convention in the present text.

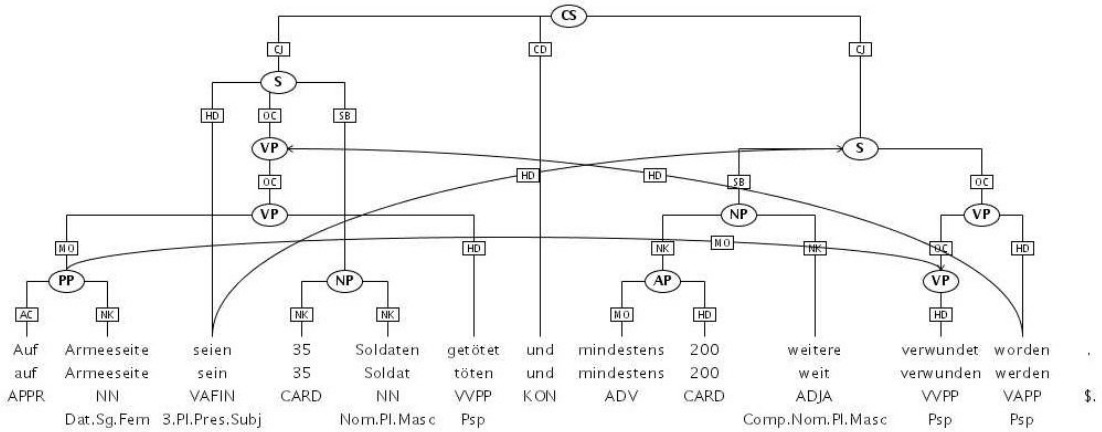


Figure 2: TIGER example tree with crossing and secondary edges

In TIGER-XML the sentences `<s id="s1376">`, terminal `<t id="s1376_1">` and non-terminal nodes `<nt id="s1376_500">` receive an identifier. Labeled secondary edges can be defined between a terminal node and a non-terminal node, as the arrow from the token *seien* to the constituent node **S** with label *HD* shows in Figure 2. In TIGER-XML the same secondary edge is represented as follows:

```
<t id="s1376_3" word="seien" lemma="sein" pos="VAFIN" morph="3.Pl.Pres.Subj">
    <secedge label="HD" idref="s1376_508" />
</t>
```

If secondary edges typically appear with coordination, the crossing edges in this example are due to topicalisation. The presence of crossing edges is denoted as attribute `discontinuous` in the `graph` element:

```
<graph root="s1376_VROOT" discontinuous="true">
```

The latest version (2.1) also includes morphological and lemma information: the attributes `morph` resp. `lemma` in the terminal elements `<t>`. Since the German treebank in SMULTRON, and hence in our expansion project, is inspired by the TIGER treebank, further details will follow in section 2.2.1.

Swedish

One of the first treebanks, if not the first, is the Swedish **Talbanken**[16], a corpus of written and spoken Swedish from the 1970s, manually annotated with lexical and syntactic information. Talbanken05 is a new version of the original Talbanken, renamed Talbanken76 to avoid confusion. Talbanken76 contains about 300,000 words annotated in the MAMBA standard. Talbanken76 has been converted into three different annotation schemes: flat phrase, deep phrase and dependency structure; see Nivre et al. (2006) for an overview of the conversion. The respective treebank versions are delivered together with the original treebank as a whole under the name of Talbanken05.[17]

The **Swedish Treebank** (STB)[18] is in a certain way the follower of Talbanken. It is the "result of the harmonization of the linguistic information in two existing Swedish language resources", namely the Talbanken and the Stockholm Umeå Corpus (SUC), this latter resource being a morphosyntactically annotated corpus and not a treebank. Researchers from the University of Uppsala and Växjö have merged and reannotated both resources. STB is now distributed by Språkbanken at the Göteborg University in TIGER-XML format and is thus ready to use for linguistic investigations with TIGERSearch. It is difficult to guess the exact size of the STB; the project homepage says that "Version 1.0 of the Swedish Treebank includes all of SUC but only the professional prose section of Talbanken". The Talbanken subcorpus of professional prose *and* high school essays consists of 200,000 tokens of written text. On the other hand, SUC contains 1.2 million tokens, so that the overall (intended) size will be well above one million. The status of the manual revision indicates that it has been completed for the morphological annotation of both resources; for the syntactic annotation it reports only a partial check in Talbanken but none in SUC. In this respect, the size of the current 'real' treebank part is less than 200,000 tokens.

Spanish

The **UAM Spanish Treebank** developed at the end of the 1990s at the Autonomous University of Madrid[19] is the first syntactically annotated corpus for

[16] The reference papers/works about Talbanken (Talbankens skriftspråkskonkordans and Talbankens talspråkskonkordans, by Jan Einarsson, Lund University: Department of Scandinavian Languages) date from 1976.

[17] `http://w3.msi.vxu.se/~nivre/research/Talbanken05.html`

[18] `http://spraakbanken.gu.se/stb/eng/`

[19] `http://www.lllf.uam.es/ESP/proyectos/UAMTreebank.html`

Spanish (see Moreno et al. (2000)) and still one of the few Spanish treebanks. As of September 1999, it consisted of 1'500 manually annotated sentences extracted from *El País Digital*, a newspaper online edition, and from *Compra Maestra*, a consumer association magazine. The corpus amounts to 22'695 words in total, resulting in an average sentence length of 15.13 words. The first 500 sentences are manually selected according to some difficulty criteria to "attack the problematic issues from the beginning". The rest is randomly selected "to avoid human bias in the selection". The project also developed annotation tools and guidelines. The texts are preprocessed: the tokenisation splits off postclitics from infinitives and participles, splits portmanteau words *al* and *del* into prepositions and articles, and combines multi-word units into single tokens. The sentences are automatically pre-tagged and then manually annotated according to the annotation guidelines. A statistical POS tagger developed in-house is used for preprocessing; after manual post-editing, a chunker recursively identifies ADJP, ADVP, NP, PP, and VP phrases. The trees are represented as nested parenthesis structures with category and feature information corresponding to the word or phrase level. They follow a simple, vertical, indented format after the Penn Treebank model (see Figure 3).

```
"Una persecución policial en Valencia deja cuatro muertos y un herido grave\
"

 A police pursuit in Valencia leaves four corpses and one badly wounded.

(S
 (NP SUBJ FEM SG P3
     (ART "<Una>" "un" INDEF FEM SG)
     (N "<persecución>" "persecución" FEM SG)
     (ADJP FEM SG
      (ADJ "<policial>" "policial" FEM SG))
     (PP EN LOCATIVE
         (PREP "<en>" "en")
         (NP
          (N "<Valencia>" "Valencia" PROPER))))
 (VP TENSED PRES IND SG P3
     (V "<deja>" "dejar" TENSED PRES IND SG P3)
     (NP OBJ1 COORDINATED
         (NP OBJ1
             (QP
              (Q "<cuatro>" "cuatro" PL))
             (N "<muertos>" "muerto" MASC PL))
         (C "<y>" "y" COORDINATING)
         (NP OBJ1
             (ART "<un>" "un" INDEF MASC SG)
             (N "<herido>" "herido" MASC SG)
             (ADJP MASC SG
              (ADJ "<grave>" "grave" MASC SG)))))
 (PUNCT "." PERIOD))
```

Figure 3: UAM Spanish Treebank example sentence in indented parenthesised format

The features encode morphological, syntactic and even semantic information because, as the project developers explain, agreement and grammatical functions are necessary for Spanish to induce rules from the treebank. The trees reflect the surface syntax; only null subjects and elisions due to coordination are added and accord-

ingly annotated. As described in the UAM Spanish Treebank guidelines (Moreno et al. (1999)) and summarised in Table 3, there are 15 POS tags with 0 to 14 possible features, these features having from 1 to 17 values (the different punctuation symbols). Some feature implications are defined for verbs and pronouns.

Part of speech	POS tag	nb of features	comment
Adjectives	ADJ	9	
Adverbs	ADV	5	
Articles	ART	3	
Auxiliaries	AUX	14	
Conjunctions	C	6	
Demonstratives	DEM	3	
Maths operators	MATHS-OP	0	
Nouns	N	11	
Possessives	POSS	4	
Prepositions	PREP	0	
Pronouns	P	10	
Punctuation marks	PUNCT	1	(17 symbols)
Quantifiers	Q	4	
"Se" marks	SE-MARK	1	(passive or impersonal)
Verbs	V	14	

Table 3: UAM Spanish Treebank POS tags (with number of features)

The developers of the UAM Spanish Treebank use three 'debugging' tools: a graphical annotation tool, a feature checker and a rule generator. Annotating with the graphical tree-drawer CLIG[20] helps them check the correctness of the resulting structures by visualising them (Figure 4). A feature checker enforces the features implications they have previously defined. And a phrase structure rule generator serves to detect inconsistencies. Finally, the coherence and quality of the analysis is checked manually.

This relates to our second research question: how do we evaluate the quality of monolingual treebanks? Apart from a graphic annotation tool, I have used and developed other quality checks for the Spanish treebank. These are explained in section 3.3. As an idea for future projects, a POS checker for closed class words could be integrated into the graphical annotation interface to enforce the correct POS tags, or at least warn the annotator of inconsistencies.

[20]CLIG (Computational Linguistic Interactive Grapher) was developed by Karsten Konrad at the University of Saarbrücken.

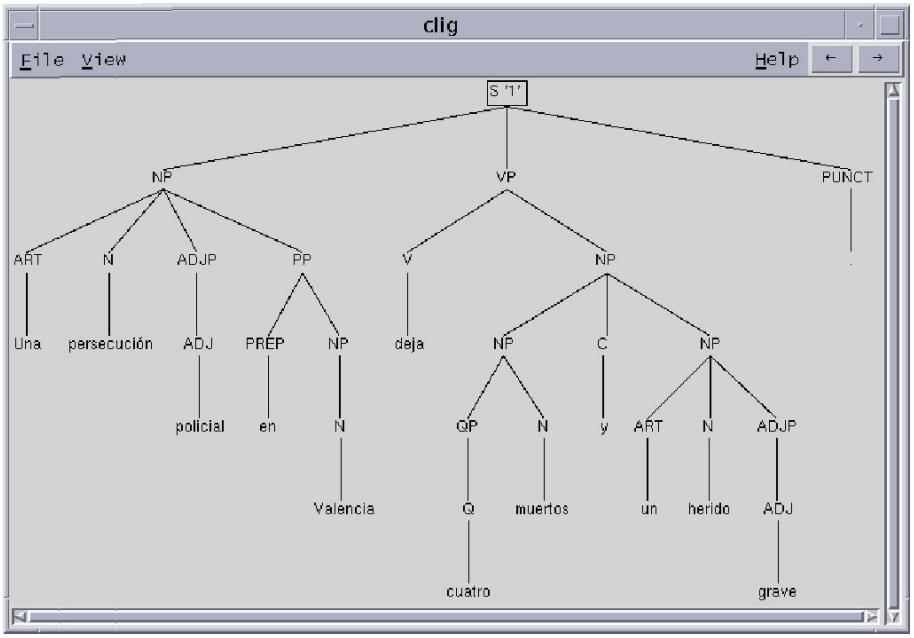


Figure 4: UAM Spanish Treebank example sentence in CLIG

The developers of the UAM Spanish Treebank used the annotated corpus to train a statistical parser; the parses produced for a separate test set of sentences achieved a recall of 73.6% for a precision of 74.1% compared to the gold standard from the treebank (correct bracket and constituent label).

Another Spanish treebank developed since the beginning of 2000 in Barcelona is **AnCora-Es** to which a separate section is dedicated (see 2.2.2). The Spanish annotation scheme followed in the AnCora project inspired the Spanish treebank built in our project. Note that, analog to the POS tags used in the UAM Spanish Treebank (Table 3), the POS tagset of AnCora-Es also includes subcategorisation features (Table 20 in Annex A).

Other Treebanks: Dependency and Semantic Annotation

The next description shall exemplify a different approach to syntactic analysis and representation: the dependency treebank. The **Prague Dependency Treebank** (PDT) was developed in two phases (1996-2000 and 2000-2004) along the dependency grammar approach. The first version contains a monolingual Czech treebank of 1.8 million morphologically and syntactically annotated words from news texts as well as an aligned parallel Czech-English corpus of about 1 million words for each language. Interestingly, the syntactic annotation (the analytic or middle layer in

their terminology) uses "dependency treebank with a level conceptually close to the syntactic annotation used in the Penn Treebank".[21] The PDT 2.0 released in 2006 contains a data subset annotated at the so-called tectogrammatical or highest layer: semantic annotation including coreference relations.

Apropos semantic annotation, there are further developments of existing treebanks that add semantic information, e.g. predicate-argument or frame structures, on top of them. The **PropBank** project[22] adds predicate-argument relations to the syntactic trees of the Penn Treebank. The Propbank annotation consists of three tasks: consistent argument labeling across the different surface realisations, annotation of verb modifiers of time, location, manner and others, and coreference resolution to find antecedents of ellipsed arguments. Current developments at the University of Colorado include an English-Chinese parallel Treebank/Propbank, "propbanking other genres" as well as linking PropBank with the lexical resources VerbNet, FrameNet and WordNet like described in the SemLink project[23].

For German, the **SALSA** project[24] semantically annotated 20,000 sentences from the Tiger corpus with frames; the set of frames based on the FrameNet model has been extended to better fit German, and to describe the new situations encountered in the texts.

The Spanish version of FrameNet (**SFN**)[25] is developed at the Autonomous University of Barcelona in collaboration with the FrameNet project at Berkeley, based on a corpus of various text sources and genres for a total of 350 million words. The goal of this project is to create lexical resources for Spanish with valence and word sense information extracted from "human approved and automatic annotated example sentences".

To sum up, there are many projects dealing with data annotation, building new resources or adding annotations to existing ones. The recent treebank developments show that we are moving away from an Anglo-centered monoculture: English has lost part of its preponderance for the benefit of other majority languages and even less influential, resource low languages. Nonetheless, there is still much work to do, above all regarding the development of parallel treebanks. In his review of Abeillé's book on treebanks, Nivre (2004: 375) symptomatically reveals that "the

[21] Quoted from the LDC catalog `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2001T10`

[22] `http://verbs.colorado.edu/~mpalmer/projects/ace.html`

[23] `http://verbs.colorado.edu/semlink/`

[24] The **Sa**arbrücken **L**exical **S**emantics **A**cquisition Project: `http://www.coli.uni-saarland.de/projects/salsa/`

[25] `http://gemini.uab.es/SFN`

book does not contain any contributions dealing with parallel treebanks, i.e. syntactically annotated parallel corpora, which are even more directly relevant for machine translation."

The first research question derives from the above statements: What are the challenges in expanding a parallel treebank by integrating Spanish, a majority language still under-represented among the existing treebanks? Can we analyse the Spanish syntax with the same formalism and tools already used for the annotation of the other treebanks in our project? Are there Spanish schemes compatible to the SMULTRON guidelines? And once built, how easy can we align the Spanish treebank to the other treebanks in the expanded SMULTRON?

2.1.4 Parallel Treebanks

A parallel treebank is built upon a parallel corpus. What various corpus linguistics experts label as a parallel corpus is a question of terminology, but they certainly agree on one fact: it involves more than one language. A parallel corpus is either a bilingual or a multilingual corpus, where the monolingual parts are:

1. translations of each other (resp. source texts and translations)
2. designed using the same sampling frame
3. a combination of these (translations and somehow similar texts)

As shown in Table 4, the different terms *translation*, *parallel*, and *comparable* are all used to describe these types of corpora, but do not always designate the same type. Johansson for example refers to them interchangeably as *parallel* corpora. McEnery and Wilson (2001) employ *parallel* for the first and *comparable* or *translation* for the second type. Still others switch these terms and talk about *translation* corpora in the first case and *parallel* in the second.

	Johansson	McEnery and Wilson	others	HERE
1. translations	parallel	parallel	translation	parallel
2. similar texts	parallel	comparable/translation	parallel	comparable

Table 4: Corpus type definitions: parallel, comparable or translation corpora

McEnery and Wilson (2001: 198) simply define a parallel corpus as "a corpus which contains the same texts in more than one language". The reason why comparable corpora are also called translation corpora is due to the insight that they are better suited for professional translators or linguists to study texts in their original (L1)

languages and so avoid the translation effect, the lack of "naturalness". Hereafter, we understand and use both the concepts of parallel corpora as being exact translated texts, and comparable corpora as a collection of texts from the same genre, treating a common matter within a certain domain.

What makes parallel corpora valuable is the explicit alignment between corresponding parts in the different languages. The levels on which the texts get aligned depend on the particular parallel corpus. Usually, there is some alignment on the text structure level, on sentence and/or on sub-sentential level: whole text; text divisions like heading, body, appendices; smaller text subdivisions like chapters, sections, and paragraphs; sentences, and within them, phrases and words. Hence, our revised definition of parallel corpora reads as follows:

parallel corpus
: Collection of translated texts aligned on the sentence level.

Note that also comparable corpora can be aligned, 'topic-aligned', i.e. similar topics, news about the same events, etc., are marked as corresponding text units based on automatic detection –preferably automatic, since comparable corpora tend to be very large. Chapter 4 is dedicated to the alignment of our treebank and also addresses general alignment issues.

Before we take a closer look at the parallel treebank **SMULTRON** and the monolingual Spanish treebank **AnCora-Es**, here is the short list of parallel treebanks that include at least two of the four languages: English, German, Spanish, and Swedish.

- EN-SV : LinES
- EN-DE : CroCo; (FuSe)[26]
- DE-SV : (none)
- ES-{DE|EN|SV} : (none)

At the time of writing, the only parallel example with Spanish is not even a treebank but an Arabic-English-Spanish parallel corpus annotated with named entities developed at the Autonomous University of Madrid.[27]

[26]The FuSe project is often cited as example of an English-German treebank, but it is unclear if the planned **fu**nctional and **se**mantic annotated treebank has ever been built.

[27]browsable at `http://elvira.lllf.uam.es/ESP/proyectos/espa-arabe/Es-EnAlign.html`

LinES is a English-Swedish parallel treebank developed at Linköping University on the basis of the Linköping Translation Corpus (LTC).[28] LinES contains 1'200 sentences[29] in each language, randomly selected from the LTC, parsed with the Machinese Syntax parsers from Connexor Oy[30], postprocessed –missing morphosyntactic information is added and some annotation is converted to the desired scheme– and manually reviewed. The morphosyntactic annotation uses the same general POS categories for both languages, with some fine-grained morphological variations. The syntactic annotation follows a dependency scheme where every token depends on a single head token (except the sentence head itself); the resulting structure is projective.[31] Words are annotated with word stem, part-of-speech, some morphosyntactic properties, dependency relation to a head word, and position of the head. The sentence pairs are aligned at word level. The sentence alignments come from the LTC; the word alignments are manually reviewed using the interactive system I*Link[32]. Multiple alignments are allowed. Two XML files contain the monolingual annotated texts; the word aligment information is saved in a separate link file. The link file has two formats: XML-format or a short five-tuple format.

Ahrenberg (2007) notes that LinES, in contrast to SMULTRON, aligns as many sentence segments as possible, regardless of the translation equivalence, but the two resources serve different overall goals: contrastive studies versus machine translation.

CroCo The CroCo Corpus[33] is a bidirectional (English-German and German-English) "parallel treebank for translation studies and practice".[34] Each subcorpus is further divided in eight registers to allow register-specific studies. The corpus is annotated with morphosyntactic information (part-of-speech, morphology, lemmas) as well as with syntactic phrase categories and functions. The texts are automatically tokenised and POS tagged using Brants' statistical TnT tagger (see page 44). MPro, a rule-based morphology tool developed in Saarbrücken[35], automatically provides morphological information as well as first phrase structure analysis. The grammat-

[28] see Ahrenberg (2007)

[29] LinES consists of two subcorpora (MS Access Help texts and a novel from Saul Bellow) containing 600 sentence pairs each.

[30] `http://www.connexor.com/`

[31] A projective analysis in both dependency and constituency structures does not allow discontinuous phrases, i.e. crossing branches.

[32] `http://www.ida.liu.se/~nlplab/ILink/`

[33] `http://fr46.uni-saarland.de/croco/`

[34] See Hansen-Schirra et al. (2006) and the abstract to the talk "The CroCo Corpus: towards a parallel treebank for translation studies and practice" held by Silvia Hansen-Schirra in September 2006 at the International Symposium on Parallel Treebanks in Stockholm

[35] see white paper http://www.iai.uni-sb.de/docs/mmpro.pdf

ical functions are added manually while checking the phrase chunks output from the automatic analysis. The texts are aligned on multiple levels: the word and sentence alignments are performed automatically with the statistical alignment tool GIZA++[36] and Win-Align (integrated within Translator's Workbench by Trados), respectively; the clauses are aligned according to their semantic contents; this alignment is done manually with the help of MMAX II[37], a multilevel annotation tool allowing arbitrarily many annotation levels and relations within and between these levels through stand-off annotation files. The corpus is said to comprise the (targeted) overall size of 1 million words, but it is unclear to me if the manual revision of the annotation and alignment has been completed for the whole corpus. Hansen-Schirra (2008: 27–28) only mentions that beyond "automatic annotation, syntactic functions are *currently* annotated manually with the help of MMAX II" (my emphasis).

2.2 Parallel Treebank meets Spanish Treebank

This is the starting point of my project and the challenge of integrating Spanish into the SMULTRON parallel treebank. This section begins with a description of SMULTRON since it was given at the start as the parallel treebank to expand. In the second part I present AnCora-Es, the Spanish side of a comparable multilingual treebank that inspired the annotation model we used to build the Spanish treebank. The selection of AnCora-Es as an annotation scheme for our Spanish treebank is my answer to one challenge of the first research question: to find –or define– an adequate scheme for both the Spanish linguistic characteristics and the SMULTRON specificities. At the end of the section, I explain the reasons behind my choice.

2.2.1 SMULTRON

SMULTRON[38] is the English-German-Swedish parallel treebank developed by the Computational Linguistics Group at Stockholm University. It includes texts of two different genres (fiction and economy texts) translated in the three aforementioned languages. It contains about 500 sentences for each text genre, accounting thus for a total of 1,000 POS-tagged sentences that have been annotated with phrase struc-

[36] `http://code.google.com/p/giza-pp/`

[37] http://mmax.eml-research.de

[38] The current SMULTRON Version 1.1 is freely distributed after registration at: `http://www.cl.uzh.ch/kitt/smultron/`.

tures and checked for completeness, correctness, and consistency. The monolingual treebanks have been built separately, following different annotation guidelines, but with regard to the final alignment of parallel parts. These alignments have been carried out on three levels: sentences, linguistic motivated phrases, and words. The first release is available since January 2008 and consists of 6 monolingual treebank XML files –2 subcorpora in 3 languages– and 6 pairwise alignment XML files.

The **sources** for the fiction part are the first two chapters of the novel *Sophie's World* translated from the Norwegian original into English, German, and Swedish. The economy texts come from three different sources:

- Press release ABB quarterly report Q2 2005
- The Rainforest Alliance's Banana Certification Program
- SEB annual report 2004

Before describing the language specific guidelines and the tools used at each annotation stage, I introduce the general figures of SMULTRON's subcorpora to help situate them within the spectrum of parallel treebanks, and compare them at the end of our project with the statistics of the new parts (see Table 18 on page 71).

	nb of sentences			avg nb of tokens/sent		
	EN	DE	SV	EN	DE	SV
Sophie's World	528	529	536	14.83	14.02	13.79
ABB Q2 2005	199	227	204	25.85	21.24	21.56
SEB report 2004	193	197	195	23.05	20.94	17.00
Rainforest Certification Program	80	86	93	23.50	22.16	18.61
Subtotal economy texts	473	510	492	24.58	21.28	19.20

Table 5: Number and average length of sentences in SMULTRON sources

With about 1,000 trees per language, SMULTRON is tiny in comparison to long-established English, German, and Swedish monolingual treebanks, but its size is acceptable among parallel treebanks and impressive for a manually revised trilingual parallel treebank.

The annotation **tagsets** and **guidelines** used in SMULTRON up to the syntactic analysis level come from well-known previous treebank projects, except for the Swedish treebanking and the alignment guidelines that have been developed as part of the project itself. Table 6 shows the annotation schemes followed for the different languages. Let me step through the annotation stages to explain the differences and similarities between the SMULTRON monolingual treebanks.

	POS tagging	lemmatisation	parsing	deepening	alignment
English	PennTreebank	-	PennTB II	-	language independent
German	STTS	GerTWOL	TIGER	node	guidelines developed
Swedish	SUC	SweTWOL	SWE-TIGER	insertion	for SMULTRON

Table 6: SMULTRON language dependent tagsets, schemes, guidelines

Tagging To begin with the basic annotation steps, there are three part-of-speech tagsets. The **Penn Treebank tagset** amounts to 36 part-of-speech tags, punctuation symbols not included.[39] The Stuttgart-Tübingen Tagset (**STTS**) developed at the Institute for Natural Language Processing (IMS, Stuttgart) and at the Linguistics Departmennt at the University of Tübingen consists of 54 part-of-speech tags, 48 of which encode true word subclasses distributed over eleven main word categories; the 6 tags left are for punctuation marks, foreign words, truncated compound parts, and non-words. The original **SUC tagset** contains 22 two-letters tags. In SMULTRON, the Swedish is slightly changed: there is an additional tag for special symbols (non-words) and the verb is subcategorised in four tags (with tense/mood information); on the other hand, the three delimiters –not included in the basic 22 tags– are labeled with a single tag. Obviously, the sizes of the tagsets differ (EN:36 ; DE:54; SV:26), but the general word categories they encode are similar, distinguished only through subcategorisation including morphological features (finite vs. non-finite) or syntactic functions (attributive vs. predicative). The tagsets' scale makes them comparably manageable for both automatic and manual tagging. The tagsets follow the recommendations of divisibility and, to a certain extent, of readability or mnemotechnics; the applied schemes follow Leech's annotation maxims of usability, separability, complete documentation, neutrality, and are de facto standards.

Two POS taggers are used in SMULTRON: the **TreeTagger** as standalone tagging tool to automatically preprocess the data and the **TnT** tagger integrated in Annotate (see Parsing/Chunking below) to interactively set and correct the part-of-speech labels during the treebank annotation. These taggers are described in details in Section 3.1.5.

[39] see "Part-of-Speech Tagging Guidelines for the Penn Treebank Project", 3rd revision, Beatrice Santorini, June 1990, available at `ftp://ftp.cis.upenn.edu/pub/treebank/doc/tagguide.ps.gz`

Lemmatisation The automatic morphosyntactic processing for English is concluded after assigning POS tags. Only the German and Swedish treebanks contain lemma information. Both lemmatisation processes use a two level morphological analyser called **GERTWOL** and **SWETWOL**[40]. These morphological analysers use a lexicon of about 85,000, resp. 75,000 base-forms and provide all the possible readings for a word they recognise, finding it in the lexicon and analysing it by derivation and composition mechanisms. Disambiguation is twofold: a local context-free disambiguation based on the principles of compound elimination and derivative elimination (see Karlsson (1992)); a context-dependent disambiguation based on constraint grammar rules and probabilistic rules extracted from corpora. These disambiguation steps are applied sequentially, first the local, then the context-sensitive, and finally the heuristics disambiguation. Some lexical ambiguities are resolved thanks to the already tagged part-of-speech, others as described above. The remaining ambiguous cases, often due to many possible word segmentations, are treated differently in German and in Swedish: the selection of the right base word is automatically done for German nouns, adjectives and verbs as explained in Volk (1999); in Swedish the decision is left to the human annotator. Also, if GERTWOL, respectively SWETWOL, cannot analyse the morphology of an unknown word, the lemma information is left empty and set later manually. Apart from XML tags, punctuation marks and numbers, every token gets a lemma.

Some more details concerning the lemmatisation as described in the guidelines[41]: elliptical compounds are completed, abbreviations are spelled out, but acronyms are left as is. Misspelled words are only corrected in the lemma form. Proper names and foreign words are left unchanged, only without –if given– the German and Swedish genitive suffix -s or the English plural mark, respectively. Additionally, the provenance of foreign words is labeled with the two-character ISO language code or a combination of codes in the case of mixed compound words: EN, FR, LA, SV, EN-DE, and even EN-EN-DE. In German, the type of named entities is set: GEO, ORG, PERS, WEB and MISC. In summary, the lemmas are set by the morphology analysers, corrected and completed manually, some corrections deviating from the suggestions, following other guidelines instead (e.g. the SUC guidelines for the Swedish determiners).

[40] For more information on GERTWOL and SWETWOL, see the technical documents "GERTWOL: German Morphological Analyser" and "SWECG: Constraint Grammar Parser of Swedish" at `http://www2.lingsoft.fi/doc/` and `http://www2.lingsoft.fi/doc/swecg/intro/overview.html`

[41] `http://www.ling.su.se/dali/research/smultron/{DE|SV}_Lemmatisation_Guidelines.htm`

Parsing/Chunking The semi-automatic syntactic annotation is done in **Annotate** (see section 3.2.3). The pretagged data is loaded into the tool, manually corrected where required and further annotated. This process is interactive since the annotator decides when the tool should provide new suggestions that he/she can accept, complete, partially correct or reject entirely. As shown in Table 6, the tool and the annotator have to follow three different models. However, the German and Swedish schemes are very similar given that **SWE-TIGER** is an adapted version of the original German **TIGER**.[42]

After the manual syntactic annotation, and before the alignment, the flat German and Swedish tree structures are automatically **deepened**, i.e. nodes are inserted. The enriched annotation corresponds to a closer analysis of the German and Swedish linguistic structures. The deepened trees also provide more alignment possibilities.[43]

The following trees of the same sentence annotated in English (Figure 5), German (Figure 6) and Swedish (Figure 7) illustrate the differences and similarities in parsing between the Penn Treebank II bracketing style and the 'deepened' TIGER schemes.

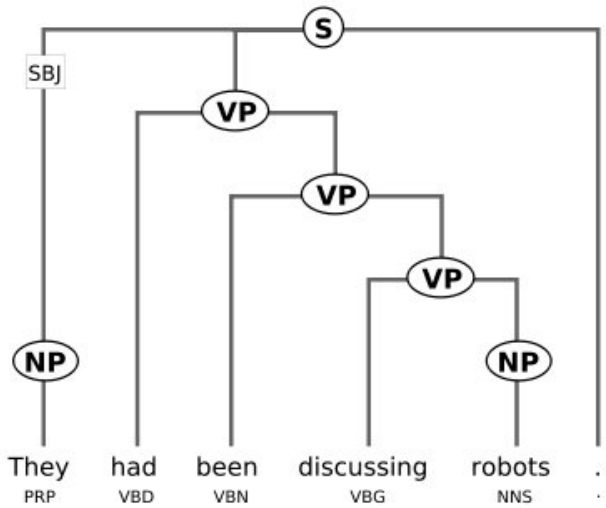


Figure 5: SMULTRON English (Sophie: sentence 5)

Both schemes have about the same constituents only with other labels; additionally, TIGER has specific labels for coordination phrases. The differences lie in the use and grouping of these constituent nodes.[44] PennTB II introduces empty elements

[42]The SWE-TIGER documentation is written in Swedish, a language I unfortunately do not understand.

[43]The node insertion is described in Samuelsson and Volk (2004).

[44]Before deepening, the English annotation along PennTB II shows deeper structures for propositional and nominal phrases than the German and Swedish equivalents following the 'undeep-

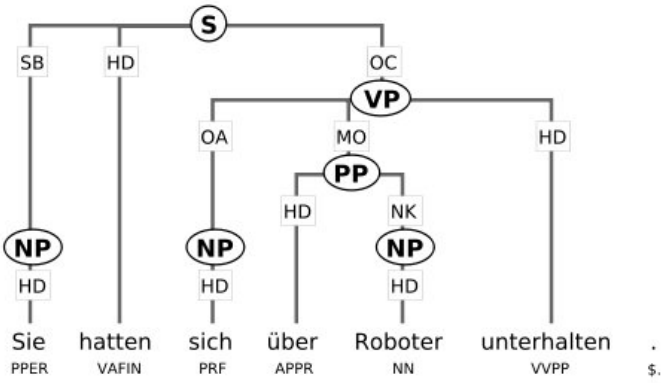


Figure 6: SMULTRON German (Sophie: sentence 5)

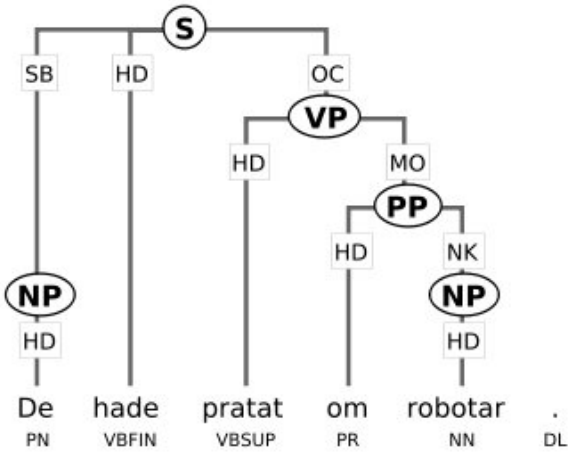


Figure 7: SMULTRON Swedish (Sophie: sentence 5)

and traces, whereas TIGER allows crossing branches (not the case in the illustrations chosen for practical space reasons). The punctuation symbols are attached to the tree structures in PennTB II but not in TIGER. Another difference is the use of the verb phrase constituent **VP**: in TIGER it only dominates non-finite verbs whereas PennTB II produces 'cascaded verb chains', as shown in Figure 5 for *has been discussing*: `VP(has VP(been VP(discussing)))`.

Figure 8 and Figure 9 –copied from the SMULTRON project site– show the process of building the monolingual treebanks and aligning them.

ened' TIGER guidelines, where there is no unary node (a constituent with a single child node).

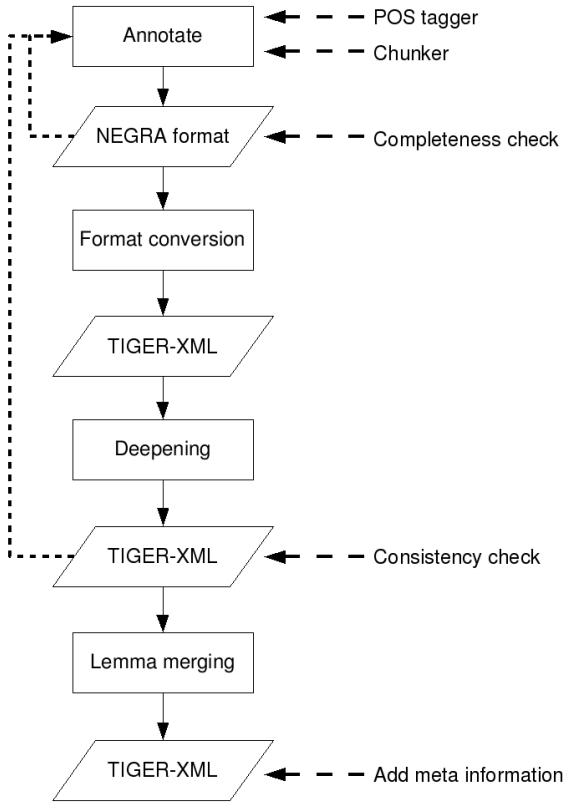


Figure 8: SMULTRON's monolingual treebank building process

Checks and Conversions At the end of the semi-automatic annotation, separate scripts check the data **completeness** and **consistency** (see 3.3.2). For this latter check, the relations are automatically extracted, categorised and quantified. Then the annotator can take a closer look at deviating or rare relations and correct them if appropriate. These checks are repeated until a satisfying annotation correctness and consistency is achieved. An intermediate step is necessary to convert the **NEGRA format** output by Annotate to the **TIGER-XML format** expected by the alignment tool. In SMULTRON, this conversion was carried out loading the data in NEGRA output format into TIGERRegistry and letting a built-in procedure convert it to TIGER-XML format. There is now a separate program available that eliminates manual interventions, avoiding thus to interrupt the pipelined data processing (see 3.2.4.2).

Alignment The alignment tool Stockholm TreeAligner (**STA**) has been developed in the course of the SMULTRON project. It has been further developed since then

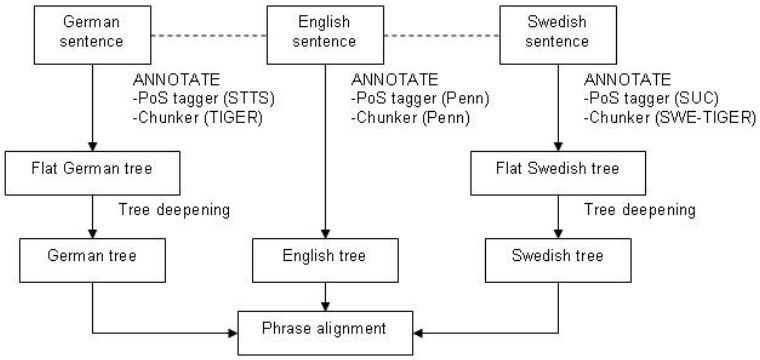


Figure 9: SMULTRON's alignment process

and is currently maintained, enhanced and distributed under a simplified name: **TreeAligner** (see 3.2.3). The alignments are liable to general, language independent guidelines. Translational equivalent units are aligned on and between the word/token, sub-sentential phrase and sentence levels. An important principle for the annotators is to try to align the maximum number of corresponding parts, using *exact* or so-called *fuzzy* alignment lines. Another rule states that the aligned elements should be as near to the word level as possible. As a final remark about the guidelines, multiple alignment to words is allowed.

The developers of SMULTRON experimented an interesting automatic alignment strategy, a kind of transitive alignment by 'triangulation': given three languages L1, L2 and L3, and two aligned language pairs, say L1-L2 and L1-L3; find the parallel alignments and apply them to the unaligned language pair L2-L3. This aligment transfer should not depend on the structural similarity between the languages as it makes use of the translational equivalence of the alignments in SMULTRON. It was tested on the two manually aligned language pairs English-German and English-Swedish to compute the alignments in the third language pair German-Swedish. The results are encouraging: corrections are necessary but this automatic transfer significantly speeds up the whole alignment process (see (Samuelsson and Volk 2007a) and (Samuelsson 2007)).

2.2.2 AnCora

AnCora[45] is a project at the Centre de Llenguatge i Computatió (CLiC) from the University of Barcelona that develops in collaboration with TALP[46] a "multilevel annotated corpora for Catalan and Spanish" (Taulé et al. 2008). These **AnCora-Ca** and **AnCora-Es** corpora are built upon corpora from previous projects (**3LB** and **CESS-ECE**). Each corpus currently contains half a million words annotated at different linguistic levels. They are the largest multilayer annotated corpora freely available for these languages at the time of writing. The corpora include texts from various sources, mostly newspapers and news agencies; the Spanish corpus also contains a subset of the balanced **LexEsp** corpus.[47] About 40% of each corpus stem from the Catalan and Spanish versions of *El Periódico* and collect the same news. As far as I know, the similarity implies only a broad (factual) correspondency of news items and not a narrow (linguistic) translation equivalence. The corpora are thus in our terminology comparable but not parallel.

Table 7 summarises the corpus figures of AnCora-Es.

	3LB-Cast	**AnCora-Es**
Size	100,000	500,000
Sources	EFE (25%) Lexesp (75%)	EFE (45%) Lexesp (40%) El Periódico (15%)
POS tagging	automatic	automatic (MACO)
POS validation	manual	-
Chunking	automatic	automatic (TACAT)
Syntax	manual	manual
Thematic Roles	-	semi-automatic
Noun senses	-	manual

Table 7: AnCora-Es corpus figures

In the following, the **AnCora** treebank will refer to the Spanish corpus AnCora-Es, unless otherwise specified.

The reasons why I chose AnCora as a base for the Spanish annotation tagsets and guidelines are manifold. The project is lead by an active research group that has

[45] see (Martí et al. 2007); `http://clic.ub.edu/ancora/`

[46] TALP is the Software Department of the Catalonian Polytechnical University.

[47] LexEsp contains over 5.5 million words; there is a search engine at `http://www.lsi.upc.es/~nlp/tools/corpus-es.php`, but until now none of my queries have returned any result.

experience in multilingual corpora and multilayer annotation.[48]

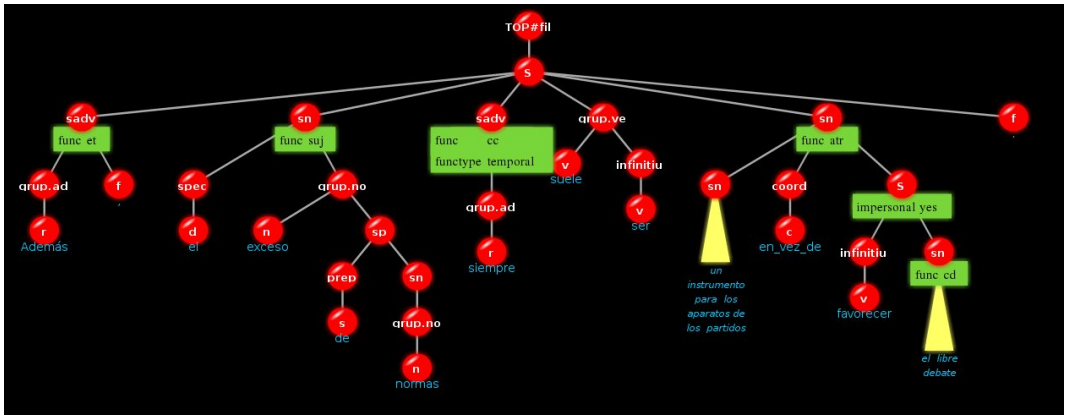


Figure 10: AnCora online tree example

Besides their freely available treebanks, there is a useful –though somehow restricted– query interface coupled with a nicely configurable visualisation online tool (see Figure 10). The annotation layers are clearly defined, based on well-founded linguistic analyses, and richly documented. The group built two Spanish parsers using the 3LB and the AnCora treebanks. Unfortunately, these Spanish parsers are not openly available. Nonetheless, in my opinion, they represent a good usability proof of the resources developed so far. Further annotation layers encoding coreferences and argument structures of nominal predicates are/were planned (2008).

In AnCora, the annotation is made layer after layer, mixing automatic, semi-automatic and manual processing, with quality control checks between each stage. As shown in Table 7, the morphological tagging and chunking are done automatically with MACO, a morphological analyser integrating a POS tagger, and with the TACAT chunker, respectively. The next annotation step is performed manually on the chunks and yields a deeper syntactic analysis. The syntactic annotation is done in two phases, first defining the syntactic constituents, then adding the function labels. AnCora has based the syntactic annotation on theoretical and methodological principles formulated after consulting the most important corpora for different languages.[49] The basic assumptions are the following:

- implicit vs. explicit information: only elliptical subjects are added (leaving the remaining elliptical constituents for a later version)
- constituency vs. dependency annotation: constituency framework is adopted

[48] M.A. Martí from the CLiC kindly sent me all the requested information about the guidelines.

[49] The cited corpus projects are: Susanne, Christine, Penn Treebank, Danish Dependency Treebank, Arboretum, Negra, Tiger, Tüba-D/Z, Floresta, BulTreeBank, Prague Dependency Treebank, and others for French, Turkish, Hungarian, Italian.

given that Spanish exhibits free constituent order rather than free word order within these constituents[50]

- arguments and adjuncts
- maintained surface word order
- theory-neutral (see Leech's maxims)

The previously mentioned treebank projects 3LB and the UAM Spanish Treebank would have been possible alternatives to AnCora. The former project has been integrated into AnCora and ceased to publish, or at least to update the project homepage, while the latter has taken a semantical approach and lately only follows this track. As stated before, I chose the AnCora Spanish corpus because the group appeared to be actively developing their treebanks, participating in conferences and competitions; the past, current and planned activities have since then confirmed this first impression. [51] As a final argument, the AnCora group has developed treebanks in different languages: Spanish, Catalan and Basque. This multilinguality aspect in the expertise of the group reinforced my choice, given the task of integrating Spanish into a multilingual treebank.

2.3 Applications of Treebanks

Treebanks are clearly not the end of the story. They represent important resources for the linguists, lexicographers, translators, first and second language teachers and students, as well as for us computational linguists. The first treebanks surged from the need for evidential language material with explicit linguistic information. With the upcoming of probabilistic approaches for tagging, parsing and machine translation, the applications in NLP mainly promoted the creation of more resources like larger treebanks. Traditionally, monolingual treebanks are used for training and evaluating parsers. They also opened new possibilities for linguistic researches and education.

McEnery and Xiao (2008: 18) comment that "parallel and comparable corpora have been a key focus of non-English corpus linguistics, largely because corpora of these two types are important resources for translation and contrastive studies". The principal aims of *parallel treebanks* also lies in these application and research fields: the

[50]The claim that dependency is more suitable for free word order languages does not hold anymore for a free constituent order language.

[51]CLiC participates in different competitions as organiser and resource provider; for example, its corpora and lexica are used in the following international competitions: CoNLL-2010, SemEval-2010 and ARE-2009. See `http://clic.ub.edu/es/competiciones` (last visit: April 14, 2009)

extra information added to the corpora through syntactic (possibly semantic) annotation opens new perspectives for constrastive linguistics with studies comparing whole structures; for machine translation, learning from these parallel structures. As a linguistic resource, a parallel treebank can be used to build a model on which a probabilistic system is based, to extract another resource (e.g. a bilingual lexicon), or serve as *gold standard* for evaluations. The following sections show briefly the use of parallel treebanks in both theory and application.

2.3.1 Contrastive Studies

Traditional corpus linguistic studies search for patterns of word combination, count word frequencies, register all the uses of a particular word or expression, analyse these results and elaborate theories as well as reference works on that empirical evidence. One motivation for annotating a corpus is that the implicit linguistic information it contains is made explicit and ready to use: the resulting treebank is "a repository of linguistic information" from which the linguists can directly profit. Now, what can we do with that extra information?

The types of linguistic questions is independent of the annotation scheme, for example whether dependency or constituency annotation, but its exact formulation and the level of detail in the results depend on the kind of information annotated. As an illustration, if we are interested in comparing relative clauses in two languages, then the expected minimum annotation information needed at least in one language is a clause category that explicitly label such relative clauses, or some other easy way to distinguish them from other clauses. How do different languages express the concept of time? Are temporal aspects covered by verbs and adverbs, or verb, adverb or prepositional phrases, or other phrases? Do languages use identical syntactic structures to convey equivalent semantic content? Is there a passive voice in both languages? How often is it used?

Note that an observer can visually grasp typological differences by merely looking at aligned trees from a certain distance. The tendency for trees to grow more on the right side in the so-called right-branching languages like English is striking when compared to more balanced-branching languages like German. Is Spanish as right-branching as English or does it more resemble German? Another characteristic visually noticeable through alignment is the word and constituent order: they differ by crossing alignment lines.

2.3.2 Machine Translation

Corpus-based machine translation has long been solely based on unsupervised large corpora like Europarl[52], but the new trends in MT include approaches combining probabilistic models and methods with the linguistic and translation information extracted from parallel treebanks. Parallel treebanks can be used in different tasks related to machine translation:

- as input for EBMT
- as training for translation model (PBSMT)
- as evaluation for alignments
- as training for transfer rules (hybrid: RBMT and SMT)

Like translation memory systems (TM), example-based machine translation (EBMT) relies on bilingual corpora, however EBMT uses the translation information expressed by sub-sentential alignments and not only the 'raw' sentence alignment. Phrase-based statistical machine translation (PBSMT) is a prominent model among the corpus-driven MT systems. Traditional PBSMT systems use unannotated parallel corpora. In an experiment, Tinsley et al. (2007) combined the data from two parallel corpora[53] with the data extracted from the parallel treebanks built upon them. They showed "that syntactically motivated word and phrase pairs extracted from an automatically built bilingual parallel treebank have a positive effect on translation scores in a baseline PBSMT system".

The current size of existing 'machine-built' parallel treebanks –automatically annotated and aligned on sentence and word levels– allows to train a statistical translation model. Additionally the language model needed in SMT can be trained on large monolingual corpora. Though SMULTRON is too small to compute a reliable statistical model on it, manually revised parallel treebanks play an important role as gold standard in parser and alignment evaluations. Indeed, the manual alignment makes SMULTRON a valuable resource for evaluation of word, sentence and sub-sentential alignments. Last but not least, as our parallel treebank contains sub-sentential alignments of sentence pairs, it could serve to build a transfer model based on these fine-grained translation equivalences.

[52]The Europarl corpus collects the European Parliament proceedings from 1996 to 2006 in 11 languages and contains up to 44 Mio words per language. See `http://www.statmt.org/europarl/`. The stated goal of Europarl is to provide aligned text for statistical machine translation (see Koehn (2005)).

[53]The sentences were selected from the Europarl corpus for a total of about 5,000 sentence pairs for English-Spanish and 10,000 for English-German.

3 Building the Spanish Treebank

Building a treebank comprises the compilation of a corpus plus the annotation thereof. According to the *Glossary of Corpus Linguistics* (Baker et al. 2006) the compilation of a corpus includes five stages: designing the corpus, planning a storage system, obtaining copyright permissions, collecting the texts and encoding them (e.g. using TEI standard). An additional stage is required to obtain a treebank unless we consider the linguistic annotation as pertaining to the general text encoding stage of corpus compilation. After a first definition phase, the central annotation activity or production phase –what Leech and Garside (1991) call "running a grammar factory"– is only one of the following 9 steps in the whole treebanking process[1]:

1. Define the purpose
2. Select a corpus
3. Choose annotation format
4. Choose annotation tool (tree editor)
5. Start the annotation (definition phase)
6. Select and adapt support tools
7. Run the grammar factory (production phase)
8. Check the annotation and make corrections
9. Distribute the treebank (with documentation)

In the following sections I will step through these compilation stages and describe the specific processing methods applied to build the Spanish part of the parallel treebank. The annotation of the English, German and Swedisch corpora runs mostly parallel to the Spanish annotation; I will mention the cases where the procedures differ. The building process description is divided in three sections: the first section is what I called preprocessing which comprises the first four steps and the beginning of the fifth step, pretagging the corpus; the second section partly covers the same decision steps but focuses on the treebank, i.e. on the syntactic annotation phases

[1]From Martin Volk's lecture notes on treebanks, Göteborg, May 2007

and tools; the third section documents the treebank postprocessing, i.e. checking and improving the annotation. A last section is dedicated to grammar induction, the possibility of extracting grammar rules from the treebank to check the consistency.

3.1 Preprocessing: Building a Tagged Corpus

3.1.1 Corpus Choice

To extend SMULTRON we decided to add not only a new language, Spanish, to our parallel treebank but also a new text genre. Typical examples of parallel texts are official documents of multilingual regions, e.g. proceedings of the European or Canadian Parliaments (Europarl resp. Hansards Corpus), reports of international organisations or companies, and fiction or non-fiction texts. SMULTRON already contains such economy texts and two chapters of a bestseller novel, so we could have chosen for example a collection of translated texts from the Swiss Government but they certainly do not contain any Swedish or Spanish as required by the predefined corpus language profile. Therefore we chose as source of our corpus a DVD user manual available in many European languages.[2] So the criteria were dictated, on one hand, by the existing parallel treebank SMULTRON: the new texts should be available at least in all four languages and their overall size should be approximately equal to that of each subcorpus, i.e. at least around 500 sentences. On the other hand, the project dimension put an upper limit to the corpus size as we could neither spend much money nor more time to manually annotate it. To build a multilingual treebank –carefully annotate the monolingual corpora, systematically check them, align the parallel treebanks, and check these alignments– is time consuming, as we shall see in the annotation evaluation in section 5.2.2, even if part of this process is done (semi-)automatically.

At this pre-digitalised stage, only a few qualitative corpus characteristics are available: the corpus consists of written texts from a new genre, user manuals, and in a new domain: consumer electronics; the manual contains about 25 to 30 pages per language; the text is technical both in language and presentation: it entails domain specific terms and formatting elements like enumerations; the overall text quality is good in the four language versions; the translation direction is unknown[3]. This

[2]NAD: L54 DVD Receiver Owner's Manual (in 8 languages). NAD Electronics International, a division of Lenbrook Industries Limited. 2007.

[3]There is no indication about the translation direction; is the corpus unidirectional or multidirectional? It is difficult to see which text is the source, if the translations come from one source text or from already translated texts, i.e. form a translation chain, etc.

latter point is not a critical issue as the general purpose of the corpus is to be used as a NLP resource and not for translation studies. Chapter 5 will present more linguistic insights as a result of the corpus selection, its consequences on the expanded treebank and its possible uses.

3.1.2 Scanning and OCR Correction

From the DVD manual booklet (as shown in Figure 1, chapter 1), we digitalised all the pages of the four language versions: the pages were scanned and the images transformed by an OCR (optical character recognition) program[4] into a text file with some formatting information and image fields. After correcting a first set of OCR uncertainties and errors within the program, we delivered the DVD manual text files together with the corresponding page images in PDF format to the annotators for a second OCR correction phase. According to the principle of faithfulness followed in SMULTRON, only the OCR errors are corrected, not the misspelled words: the typos stay in the text file as they are in the original printed version. Following the SMULTRON lemmatisation guidelines, misspelled words are only corrected in the lemma form; a word comment is also added at a later time (see 3.1.5 and 3.2.2). Finally, to ensure best quality, the corrected text was proofread by a "second set of eyes" (*Vier-Augen-Prinzip*).

The issue of typographical errors is double edged: On one side, the faithfulness is an understandable goal and a clear statement about the empirical nature of a corpus. On the other side: Do we want to build resources and systems upon corrupt words and sentences? Should the data not be as correct as possible to compute a good model? Or is it even better to have a bit of fuzziness that imitates the unperfect real world? This same question appears in the textbook of Lemnitzer and Zinsmeister (2006), but the authors leave it unanswered. The developers of the SUC corpus state that "clear typographical errors in the raw texts have been corrected in the annotated versions of the corpus texts, without recording the corrections"[5], however they do not explain why. Continuity motivated here our decision to stay faithful to the original texts, like SMULTRON before. Whatever solution we choose, we need some guidelines on how to treat not only typos but also special characters, symbols, titles, and captions. Some concrete questions that arose while preparing these guidelines were:

- What do we do with the repeating titles in page headers? Keep them or

[4] Abbyy FineReader version 7.0

[5] `http://www.ling.su.se/DaLi/suc/suc2.0_info.html`

eliminate all but the first occurrence?

- How do we handle images, figures and tables?
- How do we transcribe the special DVD player navigation functions, menus and buttons symbols like ▶"Next", ⏭"Skip Forward", ⏮"Skip Back", the copyright sign ©, the registered mark ®, etc.?
- Do we insert or eliminate blanks between numbers and measure units like in the sequence "12 V" resp. "12V"?

We decided to keep only the first occurence of each title, subtitle or heading, as we are not interested in the document structure. In this regard, we eliminated the page numbers, index table, figures and images, to obtain plain text with no formatting information. An exception was made for enumerations: the listed items keep their alphanumerical numbering and the bullet enumeration symbol is replaced by a list tag `<li>`. The need to transcribe special symbols emerged due to the encoding used in the annotation tool: Annotate is an old program that works only with ISO-8859-1 (Latin1) characters (see 5.2.1, p. 73). So I transcribed the DVD device symbols according to Table 8 and eliminated the few legal symbols.

Function	Symbol	Transcription
List item	•	
Next	▶	>
Skip/Preset Forward	⏭	>> |
Skip/Preset Back	⏮	| <<
Volume Up/Down	▲/▼	^/v

Table 8: Symbol transcription for lists and DVD functions

Note that not all these symbols have a single Unicode value, but it would be at least possible to combine codes to form the desired token. For example the symbol ⏮ can be represented as "a vertical bar followed by two left triangles" with the Unicode character string: U2503 U25C0 U25C0.[6]

The last point regarding measurement units pertains to general tokenisation issues covered in the next section.

[6]see the Unicode character code charts at `http://unicode.org/charts/`

3.1.3 Tokenisation

This section begins with some general linguistic and language specific considerations about tokenisation. In a second part, I will describe the practical side of the tokenisation procedures followed in this project.

Tokenisation is ideally the identification of linguistic units (LU). Following Moreno et al. (2003) I will show how we handle complex constituents, especially what they call "asymmetric constituents". The double entry table 9 schematises the relation combinations between the "orthographic string" and the "lexemes", i.e. between the surface and a more abstract interpretation level.

		lexemes	
		1	m
ortho string	1	simple units	**amalgams**
	m	**multiword units**	complex abstract units

Table 9: Asymmetric constituents: amalgams and multiword units

In my opinion there is no clear-cut limit between asymmetric multiword units (MWU) and complex, more abstract units. Where do dates, proper names, idiomatic expressions, periphrases or lexicalisations belong to?

For our Spanish treebank, every token is taken separately and receives its own part-of-speech tag and lemma (see 3.1.5). We differ from the guidelines established for AnCora and Cast3LB in that we do not construct multiword tokens, neither for dates or proper names, nor for prepositional or conjunctional expressions. Here some examples to compare both practices: In a second annotation step, we group

expression	AnCora	DVD-ES	English
proper name	Reino_Unido	Reino Unido	UK
preposition	cerca_de	cerca de	near
conjunction	de_modo_que	de modo que	so that

Table 10: Spanish multiword expressions: AnCora vs. DVD-ES

these multiword expressions together under special nodes that look like but are not strictly speaking phrase constituents. This mixing of form and function is not linguistically motivated; there is a practical reason for this postponed annotation of complex expressions: they are hard to identify as they evolve quickly until they reach a first basic lexicalisation; in addition, this base still appears in numerous

variations, e.g. modified by adverbs. The goal to integrate the Spanish treebank into SMULTRON also supported me in that decision since no annotation scheme in SMULTRON allow multiword tokens.[7]

The tokenisation process can also break words into many tokens. It depends on the level of the units we want to process during the next steps, what we consider to be a "word" unit and how these "words" in each language are built. As concerns our four languages, the "word" units are generally separated by spaces (blanks, tabs, newlines) or punctuation signs. But the reality is more complicated than that. Apart from the well-known ambiguity problem we face in tokenising abbreviations (the dot belongs to the token and should not be interpreted as sentence end mark), there are also special characters inside words, i.e. not delimited by spaces, e.g. alternative verbs separated by a slash (open/close) that we want to split but also button names (OPEN/CLOSE) that we treat as single token proper nouns.

Tokenising is a straightforward task for most of the cases, but it gets tricky and hardly automatically error-free when the different text genres follow specific and sometimes contradictory conventions, requiring configurable rules. Manual correction was necessary to solve the conflicts.

Another question is how to handle amalgams: should we leave them as they are, as one token, or split their morphemes into separate tokens? In Spanish there are two types of amalgams: the so-called portmanteau words and postclitics. In the annotation of Cast3LB, Moreno et al. (2003) distinguish between the `"<orthographic string>"` and the `"lexical_unit"` as we do between the `word` and the `lemma` features of terminal tags. However we do not introduce null elements (the definite article "el" in the example below), i.e. the value of the feature `word` is always present.

```
(PP
  (PREP "<del>" "de")
  (NP
    (ART "el" DEF MASC SG)
    (N "<dentista>" "dentista" MASC SG)))
```

Regarding amalgams we follow the guidelines defined in SMULTRON for German and Swedish as well as AnCora's guidelines for Spanish: we treat them as one token. In contrast to this, in English the clitics are separated from the words they are joined to: the genitive marks *'s* and negations *n't* form a token on their own.

[7] An exception is made for abbreviations as explained in the Swedish Tokenisation guidelines: "Tokenization follows the principles of SUC. Words separated by whitespace or punctuation in the original text are considered separate tokens, as are punctuation marks. Exception is made for abbreviations containing punctuation and/or whitespace, which are kept together as one token with whitespace replaced by an underscore, e.g., t.ex. and t_ex."

In the following I will describe the tools used in this project and delineate the tokenisation procedure step by step. All the texts were sent through the same tokenisation process that had already been used in SMULTRON, first written for German and then adapted to Swedish texts. I included the specific Spanish reversed exclamation and question marks "¡" and "¿" into the list of punctuation symbols to be recognised as separate tokens.

I tokenised the Spanish text file with the help of the mentioned script[8] which works in nine steps, explained here as documented in the script itself:

1. tokenisation: recognise/normalise numbers, dates, quotation marks, sentence marks, and abbreviations; verticalise
2. named entity recognition: person and geographical names
3. tagger preprocessing: number recognition, special cases (Swiss spellings), "sentence end" marking for headers
4. POS tagging: TreeTagger with German parameters, without tokenisation, without lemmas
5. information merge: POS-tagged file with lemmas of named entities and known prepositions
6. sentence boundaries and tags corrections
7. sentence numbering
8. tag mapping: GerTWOL to STTS
9. rule-based chunking and output in NEGRA format

Apart from tokenisation, the script also includes tagging with lemmatisation, named entity recognition and some chunking. For the processing of the Spanish text, the tags, lemmas and phrase chunks added in the last two steps have been disregarded as they are meant for German. Only the tokens and the sentence boundaries are of interest to us at that point. In the next sections we will see how the Spanish tokens were further annotated.

3.1.4 Text Segmentation

Text segmentation at sentence boundaries presupposes defining first what is a sentence or at least where it ends. Here is our pragmatic approach with respect to the sentence boundaries: we set them between typeset paragraphs, i.e. headings and also

[8] see `GermAnno` script in Annex B

enumeration items, and after "major delimiters, defined as one of the punctuation marks ., ?, !, :, or combinations thereof" as stated in the SMULTRON guidelines. Some sentence boundaries were erroneously set so we had to correct them manually. Less obvious cases of sentence boundaries errors were fixed later while annotating. At this point a remark on process optimisation, the third research question, is necessary. Annotate, the annotation tool we use in the project (see 3.2.3), does not foresee the correction of sentence boundaries. As a consequence, a tedious manipulation is needed to change the boundaries and this error-prone operation can result, in the worst case, in the loss of annotated information. As long as we use the same annotation tool, it is important to check the correct sentence segmentation before starting to annotate.

3.1.5 POS-Tagging

Before assigning POS tags to the Spanish tokens I had to choose an appropriate tagset. I have already presented the reasons for selecting AnCora's annotation scheme in section 2.2.2. I will now outline the changes I made to adapt AnCora's POS tagset to our need. The adoption and adaptation of the syntactic annotation guidelines are the subjects of sections 3.2.1 and 3.2.2.

As a general tagging principle of the project, morphosyntactic ambiguities are not allowed: there are no *portmanteau tags*[9]. The guidelines associated to the tagsets for all our four languages foresee only one tag per token, that means they enforce a single interpretation. As a consequence, the annotators have to support the burden of selecting the correct interpretation; on the other hand, single labels are easier to process.

The original AnCora POS tags encode morphological as well as semantic information (e.g. proper name categories), but they do not distinguish between positions or syntactic functions.[10] From the original 280 morphosyntactic labels available in AnCora the reduced POS tagset in our project retains 35 labels as the result of a simplification. Table 20 (in the Annex A) shows the reduction factors for each POS type. The following examples illustrate the procedure in which the labels with different morphology features are mapped to a single label:

- adjectives, ordinal: gender, number
  {`ao0fp0, ao0fs0, ao0mp0, ao0ms0`} → `AO`

[9]Lemnitzer and Zinsmeister (2006: 70) comment that the BNC leaves not easily decidable cases of ambiguity unsolved, assigning them a portmanteau tag.

[10]see page 6 on the possible additional feature types/categories encoded in POS tags

- nouns, common: gender, number
  {`nc00000, nccn000, nccp000, nccs000, ncfn000, ncfp000, ncfs000, ncmn000, ncmp000, ncms000`} → `NC`
- verb, main: mode, tense, person, number, gender
  {`vmg0000, vmic1p0, vmic2s0, vmif3s0, vii1p0, vmip2s0, vmn0000, vmp00pf, vmsp3s0, ...`} → `VM`
  ...

This collapsing means an information loss, however I considered the efficiency gain expected for manual annotation with a smaller tagset more important than the momentary information deficit: a morphology analysis of the whole corpus is always possible at a later stage. The only question is if the annotation above the morphosyntactic layer would significantly profit from that extra information or if we can postpone the fine-grained morphological annotation without losing too much accuracy in the next processing steps. For the human annotator, this information is not necessary since he/she should be able to analyse the morphology by him/herself. For the semi-automatic syntactic analysis, it depends 1) whether the parsing model was trained on data containing morphological information, and 2) whether this information is indeed integrated into the model. The morphology information could be used for automatic alignment where a weighting algorithm would make use of the number or person features to discard suggestions that do not concord.

Manual vs. automatic tagging The POS tagging for Spanish has been carried out in distinct phases:

1) manual POS tagging with modified AnCora's tagset (mACT) in Annotate before syntactic annotation
2) automatic tagging
   2a) tagger selection
   2b) tagging with TreeTagger (TT)
   2c) mapping TT ⇒ mACT
3) manual check and correction in Annotate during syntactic annotation

Note that the German and Swedish corpora have also been pre-annotated with the TreeTagger as described in section 2.2.1. The respective parameter files correspond to target tagsets so that no mapping is needed. The English corpus has been semi-automatically annotated by the POS tagger integrated in Annotate, in interaction with the human annotator.

Tagger selection The reasons for choosing TreeTagger are best explained with an overview of the tagging tools available for Spanish and their characteristics. Either the tool is a statistical tagger trainable or, even better, already trained on a similar Spanish annotated corpus, a rule-based tagger explicitely written for Spanish, or a combination of both approaches. Other important criteria are the ease of use, the accuracy and the open access of the tools and resources. In this respect see also section 5.2.1. The POS tagger used by the AnCora group is integrated into the morphological analyser called MACO; this tool is not available. Searching on the internet for an appropriate tagger I found Stanford's page[11] on statistical NLP tools that lists freely downloadable or online usable transformation-based and memory-based taggers, and some commercial products, but many links are not valid anymore. A possible tool among the listed taggers is the commercial "Machinese Phrase Tagger" from Connexor[12], but I decided to work with open-source or freely available tools.

So the list of alternative tools I considered contained at the end only four candidates:

- TnT: Thorsten Brants Trigrams'n'Tags statistical POS tagger
- TreeTagger: a decision tree tagger
- Brill's hybrid tagger: rule-based and statistical
- MBT: a memory-based tagger based on TiMBL

MBT is a free software for generating memory-based taggers. MBT requires the installation of TiMBL on which it is based; another 'drawback' is the uncertain level of quality I could reach training it on little data and on a larger tagset than the "toy" experiment described in the MBT manual.[13]

Brill tagger is a hybrid tagger that combines rules with probabilities learned from annotated data. Brill (1995) described it as a "transformation-based error-driven learning" tagger. The homepage "μ-TBL" shows how to train a Brill tagger for Swedish with transformation-based learning; apart from a large training corpus, the training necessitates a set of hand-written rules. I have not investigated the question

[11] `http://www-nlp.stanford.edu/links/statnlp.html#Taggers`

[12] `http://www.connexor.eu/technology/machinese/demo/tagger/`

[13] To cite the MBT manual `http://ilk.uvt.nl/downloads/pub/papers/Mbt_3.1_Manual.pdf`, p. 4: "The data consists of only about 100,000 words, so the quality of taggers trained with this data will not be high. It is meant as a toy corpus. The tag set used consists of 10 broad-category POS tags."

of practicability and accuracy of this alternative because I estimated the investment disproportionate in relation to the gains for the overall project and its goal.

TnT is an "implementation of the Viterbi algorithm for second order Markov models [... that] comes with two language models, one for German, and one for English", but unfortunately no Spanish model. Even though Thorsten Brants claims that his statistical part-of-speech tagger is "trainable on different languages and virtually any tagset" and that "[a]dapting the tagger to a new language, new domain, or new tagset is very easy"[14], I did not explicitely try it at this stage because it is integrated in the second phase of the interactive annotation in Annotate (see 3.2.3).

TreeTagger is a probabilistic part of speech tagger (developed by H. Schmid, IMS, Stuttgart) which uses decision trees. Most of the probabilistic methods are based on first-order or second-order Markov models. The advantage of TreeTagger compared to the above methods is that it can better handle the sparse data problem. Problems especially arise when the frequencies are low, or even zero: in these cases, the tagger should be reliable and robust, i.e. exhibit a very small probability of error. To achieve this, TreeTagger uses a binary decision tree that automatically determines the appropriate kind and size (n-grams) of the context. In our case, where we have a small amount of training data available, this capacity to estimate reliable transitions probabilities is essential if we want to train a model, as my first intention was. After discovering the practical tagset mapping, the presence of a pre-trained model for Spanish became the main argument for choosing TreeTagger.

As with all machine learning systems, the TreeTagger is applied basically in two steps: learning and doing; in other words: training and effectively tagging. The TreeTagger system consists of two programs:

- `train-tree-tagger` is used to create a parameter file from a lexicon and a handtagged corpus
- `tree-tagger` expects a parameter file and a text file as arguments and annotates the text with part-of-speech tags.

Parameter files for many languages[15] are freely available from the net and, as mentioned above, a parameter file for Spanish is among them. The only problem I had to solve was to map its POS tags to ours: Indeed, the Spanish parameter file is

[14] `http://www.coli.uni-saarland.de/~thorsten/tnt/` (last visit: October 6, 2009)

[15] The TreeTagger project homepage lists parameter files for the following languages: English, German, Italian, Dutch, Spanish, Bulgarian, Russian, French and old French.

trained on the CRATER corpus and uses the Spanish lexicon of the CALLHOME corpus of the LDC.

Tagging with TreeTagger Before tagging, I first transformed the Spanish corpus in Negra output format into a file in 'TreeTagger input format'. I then used the TreeTagger to annotate the Spanish sentences with POS and lemma information. I converted the tags provided by the Spanish version of the TreeTagger, the CRATER tagset, to the desired tags according to a simple mapping procedure.[16] Finally, I merged the original corpus file with the converted tagged file. Later, the POS tags and lemmas were checked manually visualising the sentences in Annotate to correct the errors. To improve the treebank quality, I adapted a check script: it now provides an additional list of POS ambiguities, i.e. tokens with different POS tags. This list helps the annotator to spot possible inconsistencies.

Mapping TreeTagger CRATER tagset (TT) to mACT There are three different types of mapping between the source and the target tags:

- one-to-one
- one-to-many
- many-to-one

Table 11 illustrates these three possibilities. The last column shows the format of the mapping file needed by the conversion program `tagfixes`. Note that the new target tag is on the left, the constraint word in the middle, and the old source tag is on the right. The order in which the mappings appear is important: the source tag is mapped to the target tag of the last match. The star marks the absence of constraint ($*$ = any word).

As commented above for the simplification of AnCora's tagset, the mapping of 75 tags from TreeTagger onto the 35 mACT tags loses information. Note that the automatic tagging with "only" 75 tags can be easily mapped to the reduced modified tagset; inversely, mapping the automatic tags onto a four time larger set would have needed tedious manual intervention. In any case, some fine-grained distinctions could be recuperated or even refined, applying a morphology analyser if needed.

[16] The German and Swedish corpora are also tagged with the TreeTagger but no conversion is needed since the parameter files were trained on the same tagsets as ours (TIGER resp. SUC); only the German corpus contains lemma information; the English corpus has not been pre-tagged.

mapping type	source tag	target tag	constraint (word)	file
$1 \rightarrow 1$	ADV	RG	none (*)	`RG * ADV`
$1 \rightarrow m$	ART	DA	definite article (el)	`DA el ART`
	ART	DA	definite article (la)	`DA la ART`
	ART	DA	...	...
	ART	DI	indefinite article (un)	`DI un ART`
	ART	DI	...	...
$m \rightarrow 1$	CC	CC	none (*)	`CC * CC`
	CCAD	CC	none (*)	`CC * CCAD`
	CCNEG	CC	none (*)	`CC * CCNEG`

Table 11: Tagset Mapping: From TreeTagger (TT) to DVD-ES (mACT)

3.1.6 Named Entity Recognition

There is no named entity recognition (NER) in the annotation of the whole DVD corpus. Multitoken named entities are grouped under the special node category **MPN** and Spanish proper names are tagged as **NP** but not further classified as they are in the German part of SMULTRON for example. The NER included in the German preprocessing step in SMULTRON classifies person, organisation, geographical, web and miscellaneous named entities. The unusual named entities that appear in the DVD manuals –apart from the actual manufacturer's and player's brand names and few other company and product names– are specific to the user manual genre and, above all, to the domain of audiovisual devices. Using typical general NE lists, the recognition rate would have been low. I have not tested any methods based on context target words, yet after screening the corpus for some examples I think it is worth only in very specific cases according to the guidelines we established for NEs: *menu* and *button* could be the context tokens firing the recognition of the function names associated with them.

3.2 Treebank Annotation

At this stage, the German, Spanish and Swedish corpora are already pre-tagged, the English corpus is only tokenised and segmented at sentence boundaries. POS tagging is part of the treebanking process, but the distinctive processing step to obtain a treebank is parsing. But before starting with the syntactic annotation of

our corpora (step 5 of the treebanking process defined on page 34) we must return to the previous steps 3 and 4 to choose the annotation scheme and annotation tool. I will expose the selection and definition phase in the first three sections. The last sections will be dedicated to the production and revision phases.

3.2.1 Annotation Choice

I have already presented the reasons for selecting AnCora's annotation scheme in section 2.2.2. The question of constituency or dependency annotation is also evoked there. Let us briefly reformulate two arguments in favour of the dependency framework presented by experts. A theoretical counter-reason to choose constituency grammar is the claim from dependency grammar researchers that their formalism is able "to handle languages with relatively free word order" (Jurafsky and Martin 2009: 415). But we saw that Spanish has a free constituent order rather than a free word order.[17] (Ahrenberg 2007: 271) presenting LinES argues that "[d]ependency analysis has an advantage for parallel treebanks in that phrase alignment to a large extent is given for free from the word alignment". As I understand/In my opinion, this is only a structural phrase alignment and not the translation equivalent phrase alignment that we annotate. The hybrid annotation scheme we follow is the winning compromise for not deciding as it combines the advantages of both approaches.

Other relevant annotation issues are the degree of redundancy, recursion, internal structure of complex phrases, and details at each level. The next section addresses these general issues together with the annotation challenges specific to the DVD manuals. The presentation of the annotation tool is postponed after the guidelines to achieve argument continuity and a better reading flow.

3.2.2 Annotation Guidelines

The following language independent annotation strategies are the general principles defined in SMULTRON:

1. no redundancy: no redundant information is manually introduced to avoid unnecessary inconsistencies; the information is specified once in the relevant layer and is not reduplicated at other levels.

[17] Czech is said to have a more flexible word order than English; I don't know if this also holds in relation to Spanish: is word order more flexible in Czech than in Spanish? In any case, this fact partly explains the choice made for developping the well-known Prague Dependency Treebank, a resource that has also been used to train probabilistic dependency parsers.

2. complete disambiguation: even uncertain cases are unambiguously annotated, as there is no "portmanteau constituent node"; a comment at the sentence level explains the decision and the considered alternatives.

The second principle gives a practical "solution" to the general structural ambiguity problems of prepositional phrase attachment and scope of modifiers in coordinations. Other treebank annotation issues, e.g. the internal structure of complex nominal phrases (NP), depend on the language: German and Swedish NPs present a flat structure whereas English and Spanish tend to deeper right-branching NP structures. Elliptical phrases, e.g. elliptical coordinations, use so-called secondary edges to explicit the elided elements. Multiword units are grouped under non-terminal nodes specifically created for this purpose. Since the labels and guidelines for MWUs depend on the language, I will only document the Spanish version. Discontinuous units also receive different treatments: German, Swedish and Spanish make use of crossing edges; the English guidelines prohibit them and introduces null elements and traces instead.

The annotation issues specific to the DVD manual corpus are of textual and lexical more than syntactical nature. A technical manual contains other text elements (words, abbreviations, symbols) and structures (lists, headings) than a novel. It differs also from economy texts so that we had to decide on the concrete source material how to handle these new features. The decisions made during team discussions with our annotators apply to all languages as these are specific to the text genre and not limited to the Spanish part. The text particularities are discussed under tokenisation (3.1.3) and segmentation (3.1.4). The POS tags of special cases like the multiword units (MWU), foreign language expressions, function symbols, enumeration items and acronyms are documented in the separate guidelines.

Guidelines Adaptation: Part I In this section I explain why and how I adapted AnCora's constituent and function annotation.

	part of speech POS tag	constituent node label	function edge label
AnCora (original)	280	150	59
DVD-ES (modified)	**35**	**19**	**17**

Table 12: Spanish annotation sets: AnCora vs. DVD

AnCora has three levels of annotation: a morphological, a syntactic and a semantic level. The annotation of the DVD manuals focuses on the syntactic level; the

semantic level is left for future work. On the syntactic level AnCora annotates constituents and functions. As I did for the POS tagset on the morphosyntactic level, I reduced the number of constituent node labels from 150 to 19 (Table 12). The simplifications are due to the elimination of

- redundant information
  {`s.a.fs, s.a.fp, s.a.ms, s.a.mp`} → `AP`
- discontinuity marks
  `sn.1n, sn.1c` → `NP`
- other marks like verbless sentences or adjoined elements
- intermediate nodes
  `spec, grup.*` → ∅

In AnCora, morphosyntactic features are transmitted to the syntactic level, duplicating thereby the number of constituent labels; our Spanish annotation scheme DVD-ES only includes simple and coordinated phrase constituents as shown in Table 13.

label	constituent	coordinated label
S	sentence	CS
NP	noun phrase	CNP
PP	prepositional phrase	CPP
AP	adjectival phrase	CAP
AVP	adverb phrase	CAVP
	coordinated preposition	CAC
	coordinated complement	CCP
	coordinated unlike constituents	CO
MPN	multiword proper name	n.a.
NM	multiword numeral	n.a.
MTC	multitoken conjunction	n.a.
MTP	multitoken preposition	n.a.
SVC	support verb construction	n.a.
INC	inserted element	n.a.

Table 13: DVD-ES simple and coordinated node labels

The elements of discontinuous constituents are attached together as our scheme allows crossing edges. The verbless, adjoined and other subcategories created by adding suffixes to the main categories are simply ignored.[18] One of our main princi-

[18]This explicit information could be used to check the consistency of annotation; nonetheless this

ples is to keep annotation simple for the annotators. To facilitate and speed up their job they should annotate as flat as possible without losing information; in a second step, the structures will be automatically deepened to obtain the same tree as if following the AnCora guidelines. We thus discard some intermediate constituent nodes, typically nodes just under the phrases.

DVD-ES entails the phrase constituent categories present in the German TIGER scheme. Another change is needed to handle expressions that span multiple tokens. In AnCora the following multiword units are bound –with underscores– into a single token; in DVD-ES they are kept as separate tokens, tagged with their respective POS labels, and grouped under special "multitoken nodes":

- proper nouns **MPN**
- dates **NM**
- light verbs (aka support verb constructions) **SVC**
- conjunctional expressions **MTC**
- prepositional expressions **MTP**

Ideally these linguistic phenomena should be handled by a NER-like process that identifies tokens belonging to such entities and groups them under a labeled node.[19]

Concerning the syntactic functions, I decided to keep all function labels at the beginning; depending on the experiences made while annotating and the results extracted from a first set of sentences, some complex and unused labels could still be dropped in a second phase. As in AnCora, the function labels serve to tag only the edges under a sentence constituent **S**; they correspond to traditional syntactic functions like subject, object, attribute, as well as to discourse and modality elements.

Attachment policy as stated in the Spanish DVD guidelines
Root node. In general, we try to group as much constituent nodes together as we can. A root node S dominates all constituent nodes of a typical complete sentence, including the punctuation marks. Of course, not all "sentences" are complete sentences, but the root nodes are most of the time S, CS, NP or CNP.
Many root nodes. Depending on the text segmentation, we may have many root nodes in one segment ("sentence"); in the DVD manual corpus, these are enumerations consisting of short topic title preceded by an enumeration symbol and followed by a dash and a first sentence.

would imply an additional manual effort not necessarily worth the potential ease to automati-

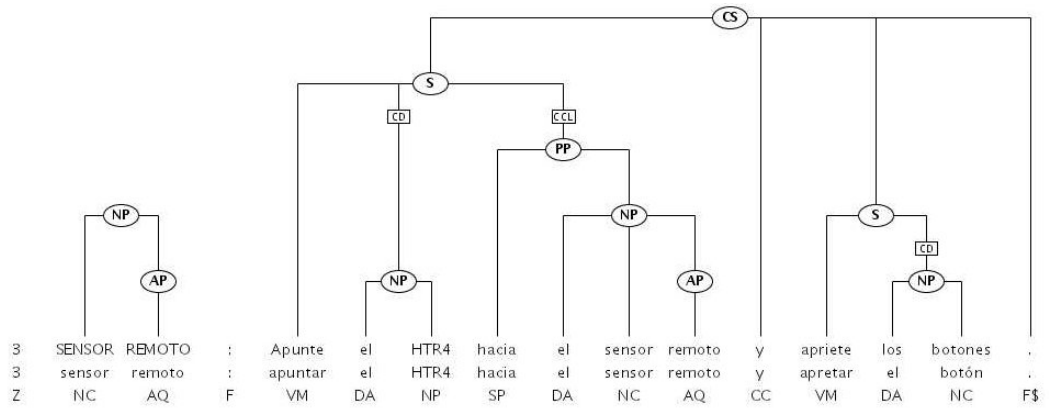


Figure 11: DVD-ES annotated sentence with many roots (s206)

Loose/unattached elements. Some terminals like enumeration symbols are not attached to any root node:

- : bullet tags
- –, :, and other separators
- a), b), c) . . . : small letters with closing bracket
- 1, 2, 3, . . . : numbers

Punctuation. Punctuation tokens are attached to the lowest possible node. In coordinated phrases, the comma is attached like the conjunction. Subordinated clauses, adverb phrases, and other textual element separated by a comma from the main clause dominate/attach this comma. The quotes around a section title or the name of a button or operating function are attached to the **MPN** node; there are other cases of quotes used as emphasis (that express a special meaning of a word): these quotes are attached to the corresponding dominating node of the quoted word.

A separate annotation guideline for every level (POS/lemma/morphology, constituents, functions/relations, alignment) together with the DVD-ES treebank aligned to the English counterpart will be available at the end of 2009.

cally control the treebank quality.

[19]Note that not every named entity consists of several tokens.

3.2.3 Syntactic Annotation Tools

Support tools In section 3.1.5 I have presented the POS tagging tools and given the reasons for selecting the TreeTagger. The choice of Annotate as the tree editor for this project was 'dictated' by the tool and annotation scheme used in SMULTRON. I have not tested any other annotation tool since this project is primarily about building a parallel treebank including Spanish and not about tool evaluation. Nevertheless an evaluation of the tools used can be found in section 5.2.1. POS taggers and parsers are the optional support tools mentioned in step 6 of the treebanking process.

Annotate has been developed at the University of Saarland in Saarbrücken. [19] It is the "natural" visual editor for treebanks in Negra/Tiger annotation format. Annotate is an interactive tool that allows for semi-automatic annotation thanks to the built-in tools develop by Thorsten Brants: a TnT POS tagger and a statistical chunker based on cascaded Markov Models, one for each layer of the syntactic structure. These integrated tools use a model to compute annotation suggestions on the user's demand allowing an interactive and thus efficient annotation –if the model is provided.

Model generator There is a model creation program that comes together with the graphical tool Annotate. This "generate-model" program needs an annotated corpus in Negra output format to compute the model from the data. Another script extracts the corpus data from the relational database.

The model can be iteratively refined and completed by generating it again based on a larger set of annotated data. After the Spanish treebank annotation was finalised I generated a new model based on the complete set of 518 sentences. The desired model is identified through its name as registered in `Corpus`, one of the MySQL database tables used by Annotate; the set of model files are all placed under the same `model/` directory. The tool works fine even when the model itself does not exist, but it should be able to access a model under the given name to benefit from it.

[19] www.coli.uni-sb.de/sfb378/negra-corpus/annotate.html

3.2.4 Annotation Phases

3.2.4.1 Production Phase

Besides the annotation examples from SMULTRON, the annotators of the English, German and Swedish treebanks had complete guidelines at their disposal. As the Spanish guidelines were still under development, the annotator began to work without any help. After about 20 sentences we discussed the uncertain cases and defined a subset of rules to follow in the original AnCora guidelines. After having annotated 150 sentences, I built a model on these data, copied the resulting model under the corresponding path, i.e. 'switched the interactive annotation on', and instructed the annotator on how to use this feature to ease and speed up the work.

Annotation control by cross-checking is not possible since we have recruted only one annotator per treebank. However we have developed other means of quality control as explained in section 3.3. During the production phase we discussed the difficult cases together, looking at the available guidelines and taking into account the other treebank annotations. I documented the decisions in the DVD or Spanish specific guidelines and the annotator added a comment to the sentences presenting difficulties and intrinsic ambiguities.

3.2.4.2 Revision Phase

The revision phase for Spanish has run almost in parallel to the production phase. The main goal was to facilitate the manual annotation, relieve the annotator from making unnecessary decisions and, at the same time, ensure a high quality: in other words, to optimise the annotation process itself. The creation of the Spanish model and the consecutive "switching on" of the interactive annotation together with the simplification of the function scheme go in that direction. Another concern was the optimisation of the dataflow and workflow for all treebanks.

Guidelines Adaptation: Part II After annotating the first 200 Spanish sentences, we definitively decided to reduce the set of edge labels from 59 to 17 to reflect the simple practice we had followed until then. This decision also relies on the results I obtained querying the AnCora treebank for the special cases of repetitive or discontinuous function marks. The low occurrences corroborated my belief and reinforced the decision to eliminate these complex function labels from the DVD-ES scheme. Table 14 lists the essential syntactic functions used in the Spanish treebank. The only undercategories left are the verb adjuncts of location and time. Another

edge label	function
SUJ	subject (sujeto)
CD	direct object (complemento directo)
CI	indirect object (complemento indirecto)
ATR	attribute (atributo)
CPRED	predicative complement (complemento predicativo)
CAG	agent complement (complemento agente)
CREG	prepositional complement (complemento de régimen verbal)
CC	verb adjunct (complemento circunstancial)
CCL	verb adjunct of location (cc de lugar)
CCT	verb adjunct of time (cc de tiempo)
MOD	verb modifier (adverbio modalizador)
NEG	negation (adverbio negativo)
PASS	passive mark (marca de oración pasiva refleja)
IMPERS	impersonal mark (marca de impersonalidad)
AO	clause adjunct (adyunto oracional)
ET	text element (elemento textual)
VOC	vocative (vocativo)

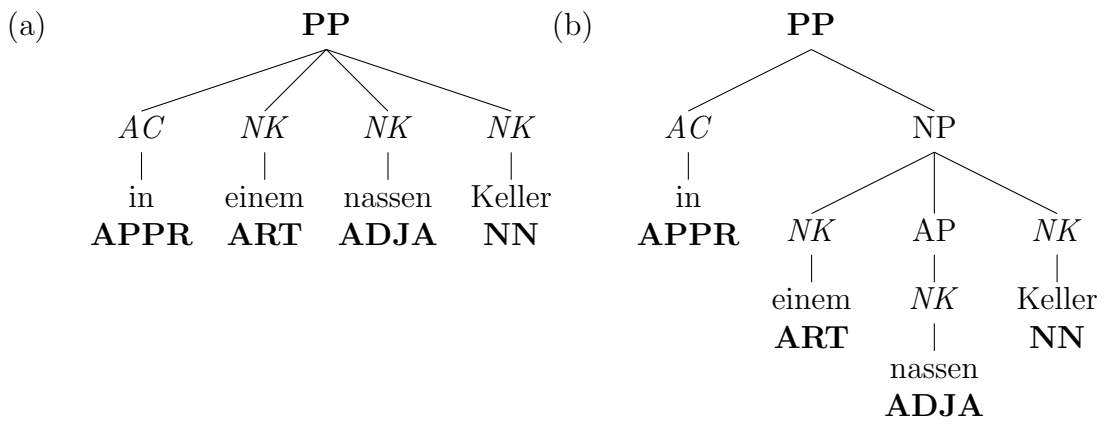
Table 14: DVD-ES edge labels

label still present though not yet used is the vocative. This function is in general rare and totally inexistent in our user manual, but I kept the label in prevision of further annotations of Spanish texts in different genres.

Dataflow/Workflow Optimisation Until December 2008, an additional manual intervention was necessary in order to prepare the data for the alignment step.[20] A Python program[21] is now available that directly converts the treebank files in Negra output format to the desired TIGER XML format. This automatic conversion improves the dataflow of the parallel treebank building process. Not only the format but also the character encoding is different: a preposed step in the pipeline converts the characters from the ISO-8859-1 (aka Latin-1) encoding used in Annotate to the Unicode UTF-8 encoding used in the format transformation script.

[20] It was necessary to import the treebank extracted from Annotate into the external tool TIGER-Registry just to export it again in another format.

[21] Thanks to Stephanie Odok who wrote the Python program `negra_to_tiger.py` (see Annex B)



Example 3.1: German (a) *flat* and (b) *deepened* PP

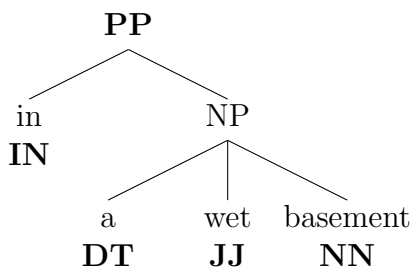
3.2.5 Automatic Deepening

In SMULTRON, German and Swedish trees are automatically deepened, i.e. nodes are inserted into the syntactic structures. This node insertion results in an enriched annotation, mirror of an adequate linguistic analysis of the sentences. In the following, the reasons and results of this automatic annotation are explained and illustrated.

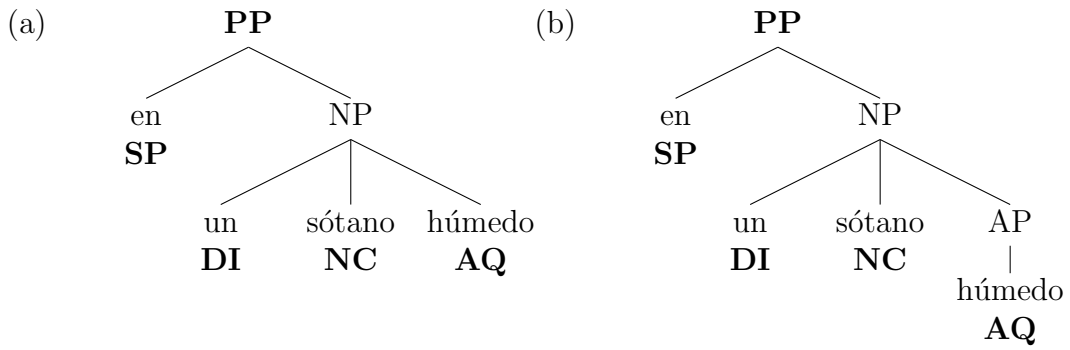
3.2.5.1 Flat versus Deep Trees

Why do we need to deepen the flat structures of the German and Swedish treebank sentences? And to begin with: Why are these structures flat? The arguments evoked in Samuelsson and Volk (2004) in favour of flat trees include practical reasons: fewer annotation decisions and better tree overview ease the annotator's work and augment the annotation throughput. Another reason is that SMULTRON was built on the legacy of existing corpora, tagsets and annotation guidelines for German and English; the flat annotation has been chosen for Swedish, because it is more similar to German than to English. But then why not flatten the English trees? This simplification ignores linguistic adequacy and interferes with the declared goal of fine-grained translation correspondences. SMULTRON wanted to follow an existing practice based on the long experience of the Penn Treebank. Additionally, every treebank should stand for itself. And last, if the information is already present, we certainly don't want to discard it, mostly because the more detailed structures we have, the more alignments we can find between the parallel treebanks.

Examples 3.1 and 3.2 show the structures of equivalent prepositional phrases (PP)



Example 3.2: English PP



Example 3.3: Spanish *flat* (a) and *middle* (b) PP

taken from our DVD treebank.[22]

Obviously the English phrase structure fits better to the automatically deepened German tree than to the flat, manually annotated original one. The Spanish annotation guidelines follow a middle way between the flat German structures and the very deep original AnCora structures. Example 3.3 Spanish version of the same prepositional phrase.

However, regarding adjectives, the English trees do not have the explicit adjective phrase node that the German and Swedish trees acquire through the node insertion procedure; the Spanish trees also entail this additional node level. Another argument in favour of deepening is the better suitability of deeper trees for linguistic complex searches. For example, working with the TIGER corpus, it is difficult to find the head nouns in flat pronominal phrases. This query is easily formulated for the German SMULTRON treebank.

To recapitulate, deepening produces more adequate linguistic structures that allow richer alignments and differentiated searches.

[22] From sentence #12 in both German (DVD_Manual_DE:s12) and English (DVD_Manual_EN:s12) treebanks

3.2.5.2 Spanish Node Insertion

In the Spanish guidelines we follow a middle way between totally flat and extremely deep structures; consider the following three possible parse trees for the same noun phrase:

- flat: NP(DI(*un*) NC(*sótano*) AQ(*húmedo*))
- middle: NP(DI() NC() AP(AQ()))
- deep: NP(spec(DI()) group.nom(NC() AP(group.a(AQ()))))

The flat scheme first planned for single adjectives modifying a noun was ignored in the annotation of the first half, so I decided to continue the middle way to avoid the many manual corrections. If we can enrich tree structures inserting unambiguous nodes, then we can automatically eliminate these nodes without information loss. I have not yet performed a complete deepening of the Spanish treebank, however I think that our annotation structures can –theoretically– be transform into AnCora's scheme. The following examples are meant as a first proof of concept. Which nodes should be inserted in the first place? Potentially all the node categories I eliminated from AnCora's tagset as listed in Table 15. This looks very similar to context-free grammar rules.

upper node	inserted	lower node
XP	group.X	X
NP	group.nom	NC
NP	group.nom	NC AP
group.nom	spec	D (Determiner)
...		

Table 15: DVD-ES node insertion candidates (previously eliminated from the AnCora scheme)

The structures resulting from the simplification of the AnCora scheme have hopefully not lost information, i.e. it is still possible to automatically reconstruct the richer structures based on the local information. What are the required conditions for inserting a given node? It depends on the types of nodes present in the simpler structure. In some cases the annotated syntactic functions available as edge labels between these nodes may be necessary.

3.3 Postprocessing: Improving Quality

A research question issue to be discussed is when to check the quality of the treebank annotation and how. What is the optimal manner to interleave the manual and automatic phases? The alternation is necessary: after each automatic phase there should be a manual control that could deliver new input for optimising the automatic processing; on the other hand an automatic check of the manual work should also improve the quality of the resulting treebank by eliminating human inconsistencies (of one annotator or between annotators). This reciprocal refinement cycle ideally leads to an optimum treebank annotation.

3.3.1 Error Detection and Correction

There are three ways to avoid annotation errors:

- annotate entire corpus several times and test interannotator agreement
- use/define adequate annotation scheme, with good documentation and list of specific problematic cases to allow for 100 % agreement
- automatic methods of error detection

Unfortunately it is impossible to apply the first method in this project because of the financial and organisational restrictions. Each version of the DVD manual is annotated by only one person. We have discussed the difficult cases and doubts about vague guidelines, and there are still points to clarify, the whole annotation process being a work in progress, a cycle of decision, application, revision, question, decision, and so forth.

This leads to the second point: adequate annotation scheme and well documented guidelines. The difficulty for the Spanish annotation was the constant changes to adapt the guidelines to our needs. More than for the other languages, where the guidelines were already well defined and documented, at least for the general, most common language texts.

The third way to 'avoid' errors is to detect and correct them, if possible, automatically. The SMULTRON project used some scripts written in Perl to check for completeness and detect possible inconsistencies (see scripts in Annex B).

A manual way to control the annotated data is to visualise it and search for given structures with the help of TIGERSearch or other graphic query tools (see 4.2 and 4.3). This visualisation step is helpful for developing automatic checks: it helped me

to see what can go 'wrong' in the manual annotation step. Once the typical errors are discovered, the checks can be automated.

3.3.2 Completeness and Consistency Checks

"Completeness and consistency are important characteristics of corpus annotation" state Samuelsson and Volk (2007a). As mentioned in 2.2.1, the SMULTRON parallel treebank was checked for completeness and consistency using Perl scripts. I adapted these scripts to the new XML format used in the alignment tool[23] and wrote the monolingual consistency check in `display_token_ranges.perl`. Before checking its consistency, we control that the corpus annotation is complete.

Completeness What does completeness mean? A treebank is complete when all its sentences are annotated. When is a sentence completely annotated? It depends on the annotation scheme and the relevant guidelines, but the goal is to attach the maximum terminal and non-terminal nodes to the minimum number of trees. As an example of divergent guidelines, punctuation marks are treated differently: in English and Spanish they are attached to the trees whereas the German guidelines recommend trees without punctuation. Like Samuelsson and Volk (2007a) comment, the completeness is easy to check and "should ideally be part of the annotation tool." The sentence annotation status is indicated on the lower right corner of the Annotate window.

However the overall status is not available at once; the only possibility is to step through the whole treebank and check tree after tree for completeness. Since this is not a viable nor a reliable solution, we have to check the completeness by other means: the Perl script `check_completeness.perl` outputs error warnings that the annotator can systematically check. I incorporated some variants for the English and Spanish schemes. For example the root node labels expected in German, Swedish and Spanish are **S** or **CS**, as for English the list is larger: **S**, **SINV**, **SBAR**, **SBARQ**, **SQ**, **S-CLF** and **FRAG**. Another check is performed for special tokens: depending on the language, some punctuation marks should be connected or not. Additionally, the Spanish inversed punctuation symbols are recognised as such. In the current version, the program expects the sentences to have one root node. Since the specific DVD guidelines allow multiple root nodes in one sentence segment, and this is valid for all languages, the check outputs many error messages. The next version will work on the TIGER XML format and have a flexible scheme

[23] the TreeAligner alignment tool (see 4.2)

configuration to avoid these unnecessary warnings and imperfections.

Consistency Introducing the annotation choice, Abeillé Abeillé (2003) comments:

> What is most important is consistency (similar cases must be handled similarly) and expliciteness (a detailled documentation must accompany the annotated corpus).

The documentation is of course part of the project's deliverables, but what about the consistency? Consistency is twofold: the monolingual, internal annotation consistency of each treebank, and the multilingual alignment consistency of each treebank pair. In this section I will address the question of monolingual consistency in the case of Spanish DVD treebank. In section 4.4 we will look at the alignment consistency.

The different kind of consistency checks[24]:

- domain restrictions
- dominance relations and function labels
  - Daughter sequence to Mother node
  - Node pair to intermediate Edge label

The domain restrictions are enforced in Annotate: only the predefined morphology, part-of-speech, phrase constituent and function labels can be selected for the annotation of the corresponding levels. The only consistency problem arises when these predefined domains are updated. This is also easily controlled with the TIGER XML scheme. A sorted list of tokens (or lemma, if available) with the different POS labels assigned to them is a simple method to control the part-of-speech consistency. The Perl script `display_token_ranges.perl` reads a treebank file in TIGER-XML format and prints the token sequence under all the non-terminal nodes of all the trees. The output file is a tab-separated table with nodes and token sequences. The script writes a second file with two lists: identical token sequences under different node labels and single tokens with different POS labels.

The relation between nodes is checked with the script `check_edge_relations.perl` that generates a table for all edge relations in a TIGER-XML treebank. Table 16 shows list of edge relations present in the Spanish DVD treebank (the list is sorted by descending frequency). The remaining control consists of manually checking the unfrequent cases and to correct the obvious linguistic mistakes. The output of unlabeled ("`--`") relations also helps to detect incomplete function annotation.

[24]from lecture notes on treebanks, Martin Volk, Göteborg 2005

Mother category	Function label	Daughter category	Freq	TIGERSearch query
NP	−−	NC	2437	[cat="NP"] > −− [pos="NC"]
PP	−−	SP	1631	[cat="PP"] > −− [pos="SP"]
PP	−−	NP	1342	[cat="PP"] > −− [cat="NP"]
S	−−	VM	1066	[cat="S"] > −− [pos="VM"]
NP	−−	DA	972	[cat="NP"] > −− [pos="DA"]
NP	−−	PP	905	[cat="NP"] > −− [cat="PP"]
AP	−−	AQ	514	[cat="AP"] > −− [pos="AQ"]
CNP	−−	NP	455	[cat="CNP"] > −− [cat="NP"]
S	CD	NP	398	[cat="S"] > CD [cat="NP"]
NP	−−	AP	389	[cat="NP"] > −− [cat="AP"]
...	...	...	...	...
S	SUJ	NP	271	[cat="S"] > SUJ [cat="NP"]
...	...	...	...	...
S	CC	PP	217	[cat="S"] > CC [cat="PP"]
...	...	...	...	...
PP	−−	CPP	1	[cat="PP"] > −− [cat="CPP"]
S	−−	PP	1	[cat="S"] > −− [pos="PP"]
...	...	...	...	...

Table 16: DVD-ES examples of edge relations (desc freq)

All these methods are only checks, they do not automatically correct the inconsistencies; nonetheless they are useful in that they reduce the manual effort and play a decisive role in the quality improvement.

3.3.3 Visualisation

TIGERSearch is a query tool with a graphical interface to search and browse graph/tree structures.[25] TIGERSearch can also be used for quality checks. A simple way to control if the guidelines have been followed correctly is to query for precise cases described in the guidelines. Browsing and searching for specific patterns in linguistic structures may reveal some of the annotation inconsistencies. I used TIGERSearch to view the Spanish treebank after a first automatic check and manual correction cycle. I formulated some simple queries to begin with:

[25]see page 74 for more information about TIGERSearch

- **CAVP** dominates only **AVP** nodes (and terminal coordinations) in TIGERSearch query syntax: `[cat="CAVP"] > [cat="AVP"]`
- generalisation: every coordinated phrase CX dominates X nodes, otherwise it is a coordination of unlike constituents **CO**

This first query detected the coordinated adverb phrase *arriba o abajo*, "up and down", erroneously labeled as **CAP** instead of **CAVP**.

As another example, I formulated a more complex query to check that only certain verbs are predicative and have an object complement with an attribute function. The query in TIGERSearch syntax:

```
#s >ATR #nt &
#nt > #t:[T] &
#s > #v:[pos=("VA"|"VM"|"VS")]
```

results in 105 matched trees and 161 subgraphs, but these include all the terminal nodes under the node with attribute function. The statistics for the verb lemma frequencies reveal that *ser* and *estar*, both verbs meaning "to be" are most frequent, then some modal verbs appear like *deber*, *poder* and *soler* ("must", "can" and "use to", respectively); the only unusual case is the main verb *iluminar*, "to iluminate". Looking at the sentence that unexpectedly matched the query, I immediately realised that the annotation presented a classic error: the prepositional phrase is attached to the main clause and not to the subordinated clause as it should in this particular case; thereafter the correct verb is again *estar*.

As shown in Table 16, the automatic checks write all the dominance relations encountered in a treebank together with the corresponding TIGERSearch queries. This is very useful for the immediate visualisation of suspicious cases.

Another way to control the quality of the data is to try to align it; this alignment process can indeed reveal undetected inconsistencies and unvalidate decisions that were made in one or more treebanks. Before we can align the parallel corpora, we must transform the flat structures of the German and Swedish treebanks to mimick the deeper structures of the English and Spanish counterparts. This transformation is done by automatically adding intermediate nodes to the existing trees. The reasons and further details of this deepening step are explained in section 3.2.5.

3.4 Grammar Extraction

Experiments in grammar extraction from the Penn Treebank (PTB II) have shown that the more sentences were considered, the more rules were extracted. The first hypothese that the "full underlying grammar" overwhelmingly surpassed the coverage of such a large treebank can be rejected (see Charniak (1996)). Another explanation lies in the different –sometimes inconsistent– depths of the trees, some of those structures being "underspecified" or "flatter" than they should or could be after deepening to reflect a reappearing common structure, i.e. a subtree. To handle the "rule growth phenomenon", Krotov et al. (1998) suggest to compact the induced grammar, eliminating redundant rules from it. Some redundant rules should not be eliminated because they express a linguistically correct simpler structure, rather than a partial-structure duplicate. They use the data structure frequencies to retain linguistically valid rules: A rule allowing a simpler structure should only be eliminated when this structure is less frequent than the complex structure parsed with the set of rules that could replace the "flat" rule (i.e. these rules can parse the same sequence).

The Spanish DVD treebank is too small a sample of syntactically analysed sentences to be able to extract a representative grammar of Spanish or even of the Spanish language subset deployed in user manuals. The potential grammar extracted from the Spanish DVD treebank would be far incomplete, even without applying the compaction method, but it could serve to check the annotation consistency.

4 Aligning the DVD Treebanks

This chapter begins with the general purposes of text alignment and the more specific motivation for aligning the DVD parallel treebanks. The second section introduces different alignment tools and presents in greater details the TreeAligner that I used to align a first set of Spanish with English sentences from the corresponding DVD treebanks. The guidelines and experience gained through the evaluation of these first experimental results are the topics of the last two sections.

4.1 Motivation

> Since the reasons for constructing multilingual corpora include being able to correlate individual pieces of one text with corresponding parts of another, their use immediately raises the problem of *text alignment*, or computing which chunk of a text in one language corresponds to a given chunk of the parallel text in another language. (Lawler 2001: p.19)

Much work has been done and many papers have been written on text alignment based on paragraph, sentence and word alignments. The different sentence alignment methods are length-based (number of words or characters) or use lexical information (with/out dictionary). Gale and Church (1993) first described the character length-based technique for text alignment in the best-known paper on that subject. Another development in the alignment field is the so-called phrase-based alignment: this method tries to align the largest possible sequences of tokens. The term 'phrase-based' may be confusing in a phrase constituency environment where the phrases we work on are linguistically motivated phrases. The new approach applied in the SMULTRON project is *sub-sentential alignment* that aligns not only individual words to parallel terminal nodes and whole sentences to the corresponding root nodes, but also the phrase constituents on one side to the equivalent 'nodes in between' on the other side. Note that the alignments are "translationally and not necessarily structurally equivalent" text sequences, i.e. not necessarily continuous sequences of tokens nor similar phrase structures. To align our parallel DVD

treebanks at word, phrase and sentence level, we need tools designed for this task and guidelines to achieve consistency and 'translation usability'.

4.2 Alignment Tools

There are many sentence alignment tools available from the internet.[1] Most of them are meant to be used in a batch process on large amounts of parallel texts; the graphical interface is only important in a second stage to visualise and possibly correct the automatic alignments. The evaluation can be done manually in these visualisation tools or automatically by computing the distance to a gold standard, i.e. some similarity measures between the suggested alignments and a manually aligned reference. The tool we use for tree alignment allows us to go beyond this 'simple' sentence alignment.

TreeAligner TreeAligner[2] is a powerful alignment and query tool for parallel treebanks. It is meant as an enhanced alternative to TIGERSearch for parallel treebank queries.

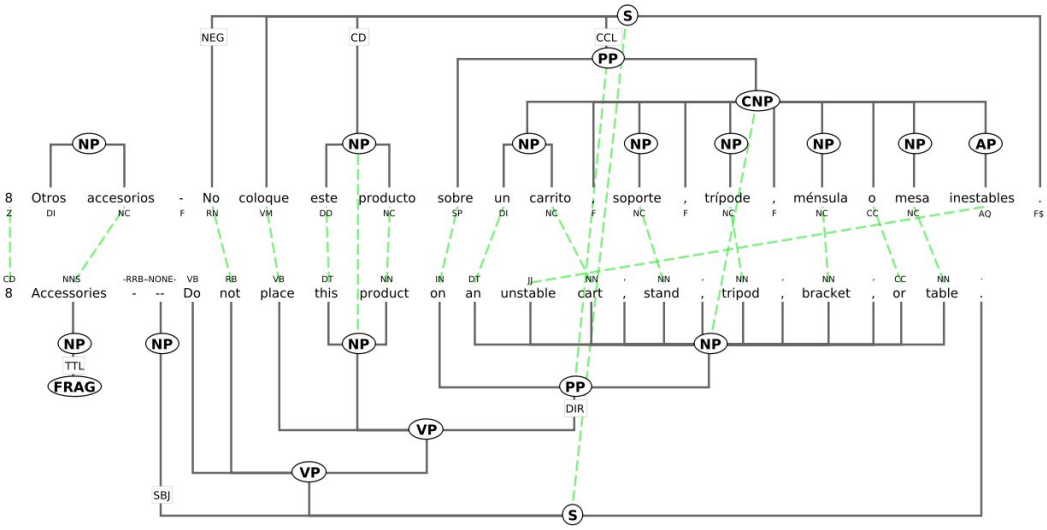


Figure 12: DVD ES-EN alignment example

[1] see e.g. the list of Rada Mihalcea, University of Northern Texas: `http://www.cse.unt.edu/~rada/wa/`

[2] the formerly Stockholm TreeAligner (STA)

TreeAligner integrates an automatic alignment function since version 1.1. This option can be switched on at project setup. The user can ask anytime during the alignment process for new suggestions.[3]

4.3 Automatic Alignment Experiment

I experimented with this additional feature in TreeAligner version 1.1.1 (Bandhagen). I aligned 20 Spanish sentences to the corresponding English sentences. The alignment suggestions proved more and more useful the further I progressed since the anticipated alignments rely on existing ones. The only drawback of the current implementation of automatic alignment are the many lines drawn between frequent words.[4] In effect, some words receive so many alignment suggestions that it becomes unpractical to delete them one by one. Fortunately, TreeAligner has a powerful set of short cuts to manipulate the alignment lines with a low number of operations. In a batch processing as well as in an interactive environment, the superfluous alignments (false positives) are worst than missing alignments (false negatives) "since we prefer to add links manually, with a minimal effort of manually correcting links" as clearly stated in (Samuelsson and Volk 2007b).

As evoked in section 3.3.3, the semi-automatic alignment process reveals structural errors. Although TreeAligner is primarily meant to align, browse and search parallel aligned bilingual corpora, it also helps to detect some "monolingual" errors in one or the other corpus. In this first experiment, I was able to spot annotation errors, e.g. a wrong function label **CD** instead of **SUJ**, and inconsistencies, e.g. a node labeled **NP** dominating a single coordinated nominal phrase **CNP**, or different phrase attachments that manifest diverging interpretations of a sentence. It also clearly confirmed that structural ambiguity is omnipresent, for example through PP-attachments or coordinations. I will present these results in more details in the next section (4.4). Another impression I had is that the parallelism between those two versions of the manual is high; I suppose that the Spanish version is a translation of the English original text.

[3]Roth (2009) describes in details how these suggested alignments are computed.

[4]It is a known bug that shall be fixed in the next release, as acknowledged by the developer.

(a) al (b) del

to the of the

Example 4.1: 1:m alignments of amalgam prepositions (a) *al* and (b) *del*

(a) Noun piscina (b) Verb moverse

swimming pool be moved

Example 4.2: 1:m alignments of (a) noun *piscina* and (b) verb *moverse*

4.4 Alignment Evaluation

Although I have not completed the Spanish-English treebank alignment, I have still gained some interesting insights aligning only a few sentences.

Spanish-English 1-to-many terminal node alignments As we do not split the Spanish agglutinated forms in separate tokens, we must align them with many English tokens. Prototypical cases are the monosyllabic prepositions *a* and *de* followed by the definite article *el* that merge to the contractions *al* resp. *del*. Example 4.1 illustrates the alignment of these Spanish amalgams to the single English tokens.

There are other cases of multiple alignments with common nouns and verb forms as shown in Example 4.2

The more general case of multiple alignments, called many-to-many alignments, also appears among the first sentences: the Spanish pseudo-reflexive verb form *moverse* (that expresses the passive voice *ser movido*) is aligned to *be moved*; when the clitic *se* comes as a separate pronoun, then there are 2 tokens on both sides. How do we align them? Following the practice described in Samuelsson (2007) for SMULTRON, we leave these words unaligned and assume that "[i]f many-to-many alignment is required, the alignment on phrase level is [...] sufficient."

se	*debe*	*mover*
p0	vm.fin	vm.inf
(self)	must	move

"should be moved"

Gold Standard The goal of aligning a small portion of the Spanish text with the English text is to serve as gold standard for the evaluation of automatic alignment

Example 4.3: ES-EN example of unallowed n:m alignments

experiments. One idea is to automatically transfer a pair of aligments that have a language in common. Tests have already been performed between German and Swedish based on the manual English-German and English-Swedish alignment pairs. Samuelsson (2007) reports on a machine learning method to predict alignments for a parallel treebank:

> It is interesting that the variations between the languages are minor. This may in part be attributable to the fact that the three languages used, German, English and Swedish, are related.

In another paper Samuelsson and Volk (2007a) describe a rule-based alignment transfer "where the alignment between L2 and L3 is derived from the alignment between L1 and L2 and between L1 and L3".[5] The transfer of one-to-one word aligments on both sides is straightforward. The difficult cases arise when there are different kinds of alignment in the two aligned pairs. In some of these cases the alignments cannot be automatically mapped, but the authors counted only few 'untransferable' alignments. Samuelsson and Volk (2007b) show the results of "Using statistical n-gram alignment for syntactic phrase alignment" and conclude that the discontinuous constituents responsible for crossing edges (not allowed in the English annotation guidelines) represent a problem for their n-gram approach.

Zhechev and Way (2008) report on the intrinsic and extrinsic evaluation of alignments and conclude that the automatic alignments should lead to improve translation quality. So the question that arises is if the manual alignments are a 'gold' standard after all. Linguistically they may be more correct, better reflect the annotator's language understanding, but for the final goal of machine translation they may be inadequate/suboptimal. On the other hand, the argument based on evaluation results only shifts the problem one step further to the machine translation evaluation metrics whose validity has been discussed in Callison-Burch et al. (2006). In other words: are we humans more satisfied with the translations produced by machine translation systems based on automatic alignments or based on manual alignments? If we judge the former to translate better, than we should ask ourselves why we still spend any effort to align parallel treebanks manually.

[5]This automatic alignment transfer is briefly explained in section 2.2.1, p. 27.

4.5 Alignment Guidelines

The current version 2 of the alignment guidelines has been released in September 2009. [6] It is too early to provide a definitive version of the alignment guidelines including the experiences made aligning the DVD treebanks. Some DVD specific cases like the buttons, menus, functions, product and company names are to be aligned like the other named entities thus far. Alignment with the new Spanish scheme may impose guidelines modifications, though until now I did not discover any incompatibility. How do we align for example the Spanish multiword units? If the other side is only a single token, then we can draw alignment lines between the single token and every token from the MWU; if there are equivalent MWUs on both sides, then we align the non-terminal nodes.

[6] The previous release of the alignment guidelines goes back to mid-2007.

5 Results

The following results cannot be compared with larger projects like the Prague Dependency Treebank with 50,000 sentences for an estimated overall cost of $2Mio or the 20,000 sentences of the SALSA I project annotated with frames in about 10 person years. This chapter sums up what we have achieved in this project. I present the overall figures of the DVD parallel treebank, report on the tool and annotation evaluations, describe the qualitative and quantitative linguistic characteristics of the Spanish monolingual and the DVD parallel treebanks and expose some enhancement ideas.

5.1 DVD Parallel Treebank

The expanded SMULTRON treebank is, following the criteria in Borja (2008), based on an electronic, off-line, written, static, synchronic, small, mixed (general/specialised), organised around three genres, non-representative corpus with text fragments of both literary and non-literary sources; it is additionally an aligned parallel multilingual syntactically annotated corpus for general research purposes and applications. Tables 17 and 18 show what kind of annotation and how much annotated data we now have in the DVD parallel treebanks.

We saw in section 2.2.1 about SMULTRON that the average sentence length (or more precisely, the average number of tokens per sentence) is dependent on the source: the distinct fiction and economy text genres show very different average length: approx. 14 resp. 21-22 tokens per sentence (see Table 5). The DVD figures in Table 18 show that there is also a difference between the languages within the user manual genre: German and Swedish sentences are about 16 tokens long whereas English and Spanish sentences are on average larger than 20.[1] Some figures can directly be extracted with the help of regular expressions from the XML file, e.g. the number of nodes (tokens or constituents) and edges. But for more elaborate evaluations we

[1] These figures depend on the text structure (many or few headings and enumerations) and, in general, on the sentence segmentation.

Linguistic level	Annotation	DE	EN	SV	ES
Morphosyntax	POS tag(set)	STTS	Penn	SUC	mAnCora
Morphology	morph tags	-	-	-	-
	lemmas	SMULTRON	-	-	DVD
Syntax	constituents + functions	TIGER	PTB II	SWE-TIGER	mAnCora
	null elements	-	trace	-	no trace
	surface order (edges)	cross+sec	cross+sec	cross+sec	cross+sec
Semantics	named entities	pos, types	-	-	-
	word senses	-	-	-	-
	roles, frames	-	-	-	-
Pragmatics	coreference	-	-	-	-
	discourse	-	-	-	-

Table 17: DVD treebanks annotation layers

lang	tokens	lemmata	phrases	sentences	tok/sent
DE	8,987	1,562	6,308	547	16.4
EN	10,283	-	7,860	512	20.1
ES	10,835	1,430	7,304	518	21.0
SV	8,882	-	7,018	533	16.7

Table 18: DVD Parallel Treebank Statistics

need specialised structure query tools. TIGERSearch already offers general statistics information about a given treebank, but this treebank has to be loaded before with the TIGERRegistry tool. TreeAligner gives more specific information about the different sorts of nodes and edges. These tools are evaluated in section 5.2.1.

5.2 Evaluation

Evaluation is based on the principle of comparison; it is important to balance the subjective impressions with second opinions, community expertise and quantitative measures and more objective ranking. Even if both issues are intricated, I will try to evaluate separately the annotation results and the tools that supported the annotation process. The quality obtained by automatic annotation and the quality finally achieved after manual revision differ, hopefully to the advantage of the human, although I think the NLP community would welcome at least an equal situation.

5.2.1 Tools Evaluation

The tools used in different treebank building projects are as various as the underlying formats combined with the desired features multiplied by other idiosyncrasies and preferences. Like for the annotation schemes, there is no apriori standard, but some tools can draw on many years of experience and a long list of successful projects. For each tool evaluation I will comment on the same aspects, wherever applicable: availability, ease of installation, user friendliness, and quality. Additionally I will describe some alternative tools, for the sake of comparison, though I can only present theoretical, i.e. reading insights that do not rely on direct experience with them.

TreeTagger is certainly one of the well-established POS tagging tools.

Availability. TreeTagger is not open source code but the tool is freely available (for research, education and evaluation) from the internet together with tagging –and in some cases even chunking– parameter files for many languages, among them English, German and Spanish, but not Swedish.[2] Language independency: TreeTagger can be trained on any tagged corpus and is therefore language independent. As announced on the TreeTagger homepage, the executable code and the parameter files (though the coverage thereof depends on the system) are available for Sparc workstations, Linux and Windows PCs and Macs.

Installation. As for my experience, the downloaded software package is easy to install on a Linux system. There is an installation script and some additional hints.

Usability. TreeTagger is a plain command line tool that comes with default batch programs minimally configured for tagging texts in the languages currently included. It is also possible to train the tagger to obtain a new parameter file, but I have not used that program as the Spanish parameters already exist. There are actually two add-on Windows graphic interfaces[3], one for the tagger and another for the training program. The user can select the options and launch the tagging or training process from a comfortable graphic window without bothering about the command option syntax. However, the command line interface is better for pipeline processing. In both cases, the tagging itself is automatically performed without any interaction with the user. That is one reason why I think our treebank building process does not need the graphic interfaces at this stage of annotation. A second advantage of

[2]There is no Swedish parameter file available on the project homepage, however the Computational Linguistics Group at Stockholm University trained TreeTagger on the manually tagged SUC and we used this Swedish parameter file in our project.

[3]Windows Interface for Stuttgart Tree Tagger, developed by Ciarán Ó Duibhín, available at `http://www.smo.uhi.ac.uk/~oduibhin/oideasra/interfaces/winttinterface.htm`

launching the program from the command line, or more precisely from a batch file, is the reproducibility of the tagging experiments: the configuration can be saved and is thus reusable.

Quality. Schmid (1994) reports an accuracy of 96.36% on the Penn-Treebank data. As stated in the "readme" file delivered with the tool, the accuracy of the TreeTagger "usually improves, if different settings of the [...] parameters are tested and the best combination is chosen". To evaluate the accuracy attained for Spanish I compared the automatic POS tags from TreeTagger with the manually checked/corrected POS tags of the first 200 sentences in our Spanish DVD treebank. The confusion matrix reports an macro-average of 85.44% precision for 80.77% recall, and a micro-average of 93.53% precision for a recall of 93.53% (see confusion matrix in Annex A.3).

Annotate is the semi-automatic tool for syntactic annotation on the Negra/Tiger model.

Availability. Annotate is available for research purposes and can be obtained free of charge after signing a license agreement. The tool runs only under Solaris and Linux. This is a limitation in comparison to "VISL's java-based tree-visualiser"[4] that should run on every system having a Java virtual machine.

Installation. Annotate is installed on one of our Solaris servers, ready for our annotators to use. I have not tried to install it elsewhere, so I cannot judge how easy the installation is. Several programs (GNU C-Compiler, Tcl/Tk, Embedded Tk, MySQL) must be pre-installed for the tool to run. The user administration is hand-work, there is no specific graphic interface to the database behind the scene. The general web tool phpMyAdmin can of course be used to manage any MySQL database. It permits saving the operations executed during an administration session, so the commands (SQL statements) can be repeated later.

Usability. For the designer/administrator, the tool is configurable in many ways: external taggers and parsers can be set up; for every corpus, the user rights, the annotation labels for terminal nodes, non-terminal nodes and edges (type and number), and the trained model for the tagger and the parser/chunker can be defined. For the end users, the multi-user system enforces that only one person is annotating a corpus at a time; this certainly makes sense, although a finer locking mechanism could be possible and may be needed for larger teams; however the corpus can be split into multiple subcorpora temporally dedicated to individual annotators. Functions

[4]Note that the goal of VISL is the Visual Interactive Syntax Learning; see `http://beta.visl.sdu.dk/treebanks.html`.

for manipulating the structures speed up the annotation process, but manipulating long sentences is cumbersome, even on a big screen.[5] As the structures get bigger, it is difficult to keep an overview of the whole underlying sentence, many window scrolling and mouse movements become necessary, and this back and forth accounts for an important time loss. One solution to this problem is found in other annotation tools (e.g. VISL's tree-visualiser and AnCoraPipe): node folding. Another convenient graphic facility for browsing large structures is zooming. TreeAligner includes both node folding and zooming. An expert user who uses the key functions saves time, yet I think that building trees would be more efficient if the annotator could also drag and drop the elements –one manipulation possibility of VISL's tool. A drawback of Annotate is the ISO character encoding: the data is encoded in Latin-1 and not UTF-8. As explained in section 3.1.4 on text segmentation, it is not possible to correct sentence breaks within the tool. The lack of this facility impedes to streamline the annotation processing and compromises its quality. An advantage of Annotate is that it allows crossing branches.

A comparison with AnCoraPipe[6] confirms the pros and cons of Annotate. On the negative side, AnCoraPipe has no statistics computed for the corpus and the annotation is not interactive; the additional features are not easily configurable. The positive sides are that the user sees what remains to be done and the 'side trees' can be folded[7]; there are additional annotation features like arguments, thematic roles, named entities and NE classes, and buttons and menus for all annotation operations; AnCoraPipe also integrates WordNet and coreference interfaces.

Quality. The accuracy of the external tagger and parser could serve as an evaluation of the interactive annotation tool. However, I did not experiment systematically to gain reliable evaluation results nor is it the goal of this project. The following comments only relate my personal experience annotating and revising three of the four DVD treebanks. Changing the label of a constituent node also resets or at least may suggest new edge labels: this well-intended help typically impedes a quick label correction in these cases where it does not imply a radical structural nor functional change. This often seemed counterproductive.

TIGERSearch I present TIGERSearch here because of its past popularity. It is now superseded by the TreeAligner query tool.

[5] I confess that most of the time I annotated or corrected the structures on my laptop with the two-buttons mouse from the touchpad.

[6] AnCoraPipe is a multilevel annotation tool developed by Bertran from the AnCora project group.

[7] It is though difficult to get used to the 'side trees' after the 'upside-down' trees.

Availability. TIGERSearch is freely available as binaries for Windows, Linux, Solaris, and MacOS X, as well as source code from the project page. It is not developed nor supported anymore but still enjoys great popularity. The tool suite consists of two GUI programs: TIGERRegistry to register the corpora, i.e. convert them in the right format, if necessary, and build the internal data structures, optionally an index, needed by the TIGERSearch browsing and querying tool.

Installation. The installation of TIGERSearch is straightforward since it is distributed as self-extracting archives (together with a Java Virtual Machine where necessary).[8]

Usability. It is possible to draw queries (graphical mode), although I have not used this feature for any complex queries, the text query language being more powerful. Additionally, I find it easier and faster to type the query in textual mode. The query results are displayed in the GraphViewer window; another useful view on results matching terminal nodes is given in the "Statistics" window. The display options can be set to show/hide selected elements of the sentence structure, context sentences, etc. Matches or whole corpora can be exported in TIGER-XML or plain text format.

To query monolingual treebanks I use TIGERSearch for two precise reasons: to quickly get the number of matches and to export them as a subcorpus. Indeed, the total of subgraph matches is not immediately visible in TreeAligner (though it can be manually computed) in contrast to the practical overview in TIGERSearch. And there is no export function to save a set of matched trees in a separate corpus.

Quality. For a query language, the quality is hopefully 100% correct answers! A serious issue is the expressive power of the query language: TIGERSearch has some known limitations that have been addressed in TreeAligner, for example the universal quantified negation. Another important issue is the performance of the system, i.e. how fast does it answer and present the results to the user. This is to a certain degree also a matter of usability, but the experienced response time has never been long enough to be of relevance.

TreeAligner "is a tool for annotating and browsing correspondences between elements of syntactical trees".[9]

Availability. TreeAligner is an open-source program written in Python; the tool has been tested on Linux, MacOS and Windows.

[8]Thanks to Alexandra Bünzli for installing TIGERSearch on a Mac.

[9]`http://www.cl.uzh.ch/kitt/treealigner`

Installation. The sources are easy to download and install on a Linux Ubuntu[10] system.

Usability. Ease of use and configurability not at the price of power. This is how I have experienced the new versions of TreeAligner. The integrated query facility is powerful and still easy to learn. It is built upon the TIGER query language and has been extended to allow universal quantified queries. The TreeAligner query language introduces the notion of node sets to implement universal quantification (see Marek et al. (2008)). Apart from the extension and some minor changes (equality operators, negation of predicates, constraint stacking), TreeAligner uses the same syntax as TIGERSearch for querying the monolingual treebanks. In addition to word, dominance and precedence relations, and node constraints for each treebank, the user can define alignment constraints on the treebank pair. The query results appear as browsable tree subset with the number of graphs (under "Pairs:") matched by the query; stepping through the result trees, the number of matches within the same tree ("Subgraphs:") is updated. The total number of matches can only be computed by hand. This feature together with the possibility to export the matched trees –like in TIGERSearch– could also be integrated in a next release. Comparison with AnCora's tree visualisation tool: it does not show/mark the exact query match, only the sentences that matches the query; the exact match is "visible" in the first result list, but it disappear when the tree is drawed.

Quality. The automatic alignment feature is very useful although it includes some imperfections. The exact evaluation scores can be consulted in Roth (2009).

5.2.2 Annotation Evaluation

The annotation evaluation takes as comparison base the rough estimates[11] for:

- a trained and experienced annotator
- supported by a good treebank editor and good tools
- on newspaper texts (avg. sentence length 20 words)

Under these conditions, the estimated annotation time lies between 2-5 minutes per sentence, i.e. a throughput of 12-30 sentences per hour.

The average speed at the end of the experiment, with almost all conditions above met –as the tools still could be better and the annotation conditions ameliorated–,

[10]However the latest version is not compatible anymore with my older Ubuntu 8.04 installation.
[11]established by Martin Volk

lies at the lower-bound estimation of 12.5 sentences per hour. Some sentences are of course faster "annotatable". I suppose that the general 80/20-rule also governs the annotation process: some sentences with special structures take so much decision time, that the average elapsed time grows above the optimistic estimates. The resulting decreased average speed is an important factor that should not be neglected in planning and supporting the next annotation effort, otherwise the annotation team could be demotivated by the discouraging tempo. Another crucial parameter is the reasonable maximum working time on the task: I entirely endorse Martin Volk's sound recommendation of at most 4 to 5 hours per day.

Apart from the time factor, the annotation evaluation should report on the quality. With only one annotator per treebank we have no data to compute the inter-annotator agreement. Moreno et al. (2003) question the validity of their annotation scheme after having annotated 1,600 sentences in the UAM Spanish Treebank project. Whether our 500 sentences are sufficient to consolidate an annotation scheme or not is theoretically unanswered, but we have proven the practical validity building and integrating our Spanish treebank into SMULTRON.

5.3 Linguistic Observations

To ampliate the existing treebank we chose a new text genre: user manuals. We looked for a third kind of text that would be 1) different from the first ones, that consist of economy and fiction, almost essay texts; and 2) available in at least all four languages we wanted to analyse: English, Swedish, German, and Spanish. The goal here is to diversify the text genres so that the underlying corpus results be better balanced. To validate or invalidate this hypothesis thorough corpus linguistic investigations are needed. The purpose of this section is more modest. It delineates the possible linguistic research questions applicable to our treebanks.

5.3.1 Qualitative Subcorpus Description

Integrating a new language into the parallel treebank, new interesting questions arise from Spanish specific constructions: the various ways to express passive and phenomena relative to the categorial chameleon pronoun *se*.

Presenting official documents as one of the parallel texts often used in statistical NLP, (Manning and Schütze 2000: 467) "suspect that the nature of these texts has also been helpful to Statistical NLP researchers" because of their "very consistent,

literal translations". As mentioned in section 4.3, I see a high parallelism between the Spanish and English versions of the manual; the Spanish text still is a good literal translation and not a word by word translation. About the issue of consistency, one naively may expect that instruction manuals are unambiguously written, follow a clearly predefined terminology. In comparison, advertising texts often play with ambiguities, searching double meanings, using puns, inventing new words, and creating new expressions to catch the attention of the potential customers. But once the product is acquired, there is no need to be witty anymore. On the contrary, for safety reasons or simply to maximize the customer's satisfaction, the instructions must be precise and complete, almost repetitive. Is this naive intuition indeed true? Does the collected data confirm that hypothesis? Another impression I had reading newspapers is that journalists use a figurative language. Do user manuals also contain language figures? Are there metaphors in the DVD treebanks? And in SMULTRON? These questions are difficult to answer systematically. They may need explicit semantic annotation. Metaphors are for example hardly searchable through syntactic and functional information alone, though we may use the support verb construction information annotated as **SVC** in our scheme. The only other 'alternatives' I see is either to look at all sentences in search of such a metaphor, or to guess their possible heads and use a concordancer together with TigerSearch or TreeAligner.

5.3.2 Quantitative Subcorpus Description

We already saw some general statistics about our parallel treebanks. This section presents various linguistic phenomena, their formulation and the quantitative results based on the Spanish DVD treebank. The corpus size is an important issue in quantitative analysis. When is a corpus big enough to allow results to be significant? It depends on the frequency of the feature we want to observe and quantify. For features occurring frequently, a relatively small sample of about 1,000 words is enough.

The typical methods used in quantitative studies are the log-likelihood (more reliable with low frequencies) and the chi-squared test (unreliable with very small frequencies).

The 48 unlabeled terminal nodes are not taken into account. The query in TIGERSearch: #t:[T] gives 10,885 matches, i.e. terminal nodes. The statistics export reports the same part-of-speech label frequencies.

POS	freq (abs)	freq (%)	with punct (%)	AnCora (%)	part of speech
A	678	7.17	6.26	7.25	Adjectives
C	612	6.48	5.65	5.50	Conjunctions
D	1,407	14.89	12.98	16.11	Determiners
F	1,389	n.a.	12.82	12.80	Punctuation marks
N	2,995	31.70	27.64	23.16	Nouns
P	289	3.06	2.67	4.51	Pronouns
R	333	3.52	3.07	3.86	Adverbs
SP	1,715	18.15	15.83	15.19	Prepositions
V	1,248	13.21	11.52	11.73	Verbs
W,XF,Y,Z	171	1.81	1.58	< 1	Misc
Total	**10,837**		100	100	POS tags
	9,448	100			w/o punctuation

Table 19: Spanish DVD Statistics: parts-of-speech

5.4 Enhancements

I see enhancement possibilities at different levels. A version control system would improve the project management, development and maintenance of the various program and data resources.

The current dataflow with manual interventions between the tools is a potential source of error besides being a time factor. The automatic pipelining could be improved. We need a single graphical tool which allows integrating annotation tools (at least POS tagger and chunker/parser), alignment tools and control mechanisms. As a consequence, the dataflow would be reduced to a minimum. The integrated checks could detect the inconsistencies on user's demand and guide the annotator directly to the suspicious cases. Impossible combinations of labels and structures and intrinsic errors (e.g. wrong POS tag of closed class words) could be registered in a "black list" and updated, if allowed. The user permission rights are an important issue for controlled quality. The ideal tool should include a user rights administration feature.

The Spanish annotation scheme may have to incorporate the morphology and the semantic information to better serve the potential treebank exploitations.

6 Conclusion

In this project of integrating Spanish into SMULTRON we have built four monolingual treebanks on the parallel English, German, Spanish and Swedish versions of a DVD manual.[1] The alignment of the English-Swedish and English-German treebank pairs has also been completed.[2]

I developed annotation guidelines, i.e. adopted and adapted the AnCora guidelines for Spanish. After first experiments, I revised these guidelines and enhanced the annotation workflow. The data workflow of the parallel treebanking process was also optimised. As a side effect I evaluated the different tools used in the project.

The expansion of SMULTRON with the 4 monolingual DVD treebanks and the 6 aligned treebank pairs have not yet been distributed; a new SMULTRON release with a complete documentation of the expansion is planned for the end of the year 2009.

These deliverables are the tangible answers to the research questions of integrating Spanish as a new language into the existing SMULTRON parallel treebank ensuring high annotation and alignment quality while discovering optimisation opportunities.

There are other potential future work areas besides the planned release: we could 'backward' expand SMULTRON with the Spanish versions of the novel and economy texts; transform the Spanish treebank in a AnCora-compatible format to query it with their graphic online interface; complete the morphosyntactic annotation with detailed morphology information; annotate additional semantic and discourse information like named entity classes, WordNet synsets, thematic roles, anaphora and coreferences; convert the hybrid constituency treebank into a dependency treebank, etc.

As Nivre (2004: 373) points out "most of the effort so far has been put into the development of resources, whereas we have really only seen the beginning when it comes to the exploitation of these resources". The concrete exploitation of the

[1] Thanks to Pierpaolo Frasa, Annette Rios, Claudia Lorencez Arreguín and Vladimir Kornev

[2] Thanks to Elisabet Joensson-Steiner and Moira Kindlimann

SMULTRON expansion and of manually annotated parallel treebanks in general remains indeed an open question for me. Some experts tend to entirely eliminate manual annotation, other trends still set on human intervention for the annotation of higher levels and larger treebanks. Both tendencies may come to a compromise using the power of the internet community with an online annotation-alignment tool[3]. The future applications will tell.

Allow me to end on a personal note. In the realm of this project, though I did not make any new "findings about human nature that were unsuspected before these resources [became] available"[4], I certainly gained interesting insights on treebank annotation: building high quality resources is hard work that requires collaboration as much as individual efforts. As a compensation, it is a reward to look at the produced trees and an encouragement to use 'all that wood' in MT systems of the next generation.

[3]Like the phrase detective game `http://anawiki.essex.ac.uk/phrasedetectives/?lb=y`

[4](Sampson 2003: 23)

References

Abeillé, A. (2003). *Treebanks. Building and Using Parsed Corpora.* Text, Speech and Language Technology. Kluwer Academic Publishers, Dordrecht; Boston; London.

Ahrenberg, L. (2007). LinES: An English-Swedish Parallel Treebank. In Nivre, J., Kaalep, H.-J., Muischnek, K., and Koit, M., editors, *Proceedings of the 16th Nordic Conference of Computational Linguistics NODALIDA*, pages 270–273, Tartu, Estonia.

Anderman, G. and Rogers, M. (2008). *Incorporating Corpora. The Linguist and the Translator.* Translating Europe. Multilingual Matters, Clevedon Buffalo Toronto.

Baker, P., Hardy, A., and McEnery, T. (2006). *A Glossary of Corpus Linguistics.* Edinburgh University Press.

Borja, A. (2008). *Incorporating Corpora. The Linguist and the Translator*, chapter 13: Corpora for Translators in Spain. In Anderman and Rogers (2008).

Brill, E. (1995). Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging. *Computational Linguistics*, 21(4):543–565.

Callison-Burch, C., Osborne, M., and Koehn, P. (2006). Re-evaluating the Role of BLEU in Machine Translation Research. In *Proceedings of the European Association of Computational Linguistics (EACL)*, pages 249–256.

Charniak, E. (1996). Tree-bank grammars. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1031–1036.

Gale, W. A. and Church, K. W. (1993). A Program for Aligning Sentences in Bilingual Corpora. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 177–184.

Hansen-Schirra, S. (2008). *Topics in Language Resources for Translation and Localisation*, volume 79 of *Benjamins translation library*, chapter 2: Interactive

reference grammars. Exploiting parallel and comparable treebanks for translation, pages 23–36. John Benjamins Publishing Company, Amsterdam.

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 Multi-dimensional Markup in Natural Language Processing (NLPXML-2006) held at EACL*, pages 35–42, Trento, Italy. EACL.

Jurafsky, D. and Martin, J. H. (2009). *Speech and Language Processing.* Prentice Hall, 2nd edition.

Karlsson, F. (1992). SWETWOL: A Comprehensive Morphological Analyzer for Swedish. *Nordic Journal of Linguistics*, 15:1–45.

Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the MT Summit 2005*, pages 79–86.

Krotov, A., Hepple, M., Gaizauskas, R., and Wilks, Y. (1998). Compacting the Penn Treebank Grammar. In *Proceedings of the COLING-ACL '98 Joint Conference*, pages 699–703, Montreal, Canada.

Lawler, J. M. (2001). Review of Multilingual Corpora in Teaching and Research. *Language Learning & Technology*, 5(3):19–23.

Leech, G. (1993). Corpus annotation schemes. *Literary and Linguistic Computing*, 8(4):275–281.

Leech, G. and Garside, R. (1991). *English Computer Corpora: Selected Papers and Research Guide*, chapter Running a grammar factory: The production of syntactically analysed corpora or "treebanks", pages 15–32. Mouton de Gruyter.

Lemnitzer, L. and Zinsmeister, H. (2006). *Korpuslinguistik. Eine Einführung.* Gunter Narr Verlag, Tuebingen.

Manning, C. D. and Schütze, H. (2000). *Foundations of Statistical Natural Language Processing.* The MIT Press, Cambridge, Massachusetts; London, England.

Marek, T., Lundborg, J., and Volk, M. (2008). Extending the tiger query language with universal quantification. In *Proceedings of KONVENS.*

Martí, M. A., Taulé, M., Bertran, M., and Màrquez, L. (2007). Ancora: Multilingual and multilevel annotated corpora. Technical report, CLiC-UB (Centre de Llenguatge i Computació, Universitat de Barcelona); TALP, Software Department, Universitat Politècnica de Catalunya.

McEnery, T. and Wilson, A. (2001). *Corpus Linguistics: An Introduction*. Edinburgh textbooks in empirical linguistics. Edinburgh University Press, Edinburgh, 2nd edition.

McEnery, T. and Xiao, R. (2008). *Incorporating Corpora: The Linguist and the Translator*, chapter Parallel and Comparable Corpora: What is Happening? 2: Parallel and Comparable Corpora: What is Happening? - Tony McEnery and Richard Xiao. In Anderman and Rogers (2008).

Moreno, A., Grishman, R., López, S., Sánchez, F., and Sekine, S. (2000). A Treebank of Spanish and its Application to Parsing. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation LREC-2000*, pages 107–112, Athens, Greece.

Moreno, A., López, S., and Sánchez, F. (1999). *Spanish Tree Bank: Specifications*. Laboratorio de Lingüística Informática, Universidad Autónoma de Madrid, Spain. Version 5.

Moreno, A., López, S., Sánchez, F., and Grishman, R. (2003). *Treebanks: Building and Using Parsed Corpora*, chapter 9: Developing a Syntactic Annotation Scheme and Tools for a Spanish Treebank, pages 149–163. In Abeillé (2003).

Nivre, J. (2004). Book Review: Abeillé, Anne (ed.), Treebanks: Building and Using Parsed Corpora. *Machine Translation*, 18(4):373–376.

Nivre, J., Nilsson, J., and Hall, J. (2006). Talbanken05: A Swedish Treebank with Phrase Structure and Dependency Annotation. In *Proceedings of the 5th International Conference on Linguistic Resources and Evaluation (LREC2006)*, Genoa, Italy.

Roth, S. (2009). Automatisches alignieren in parallelen baumbanken. Master's thesis, Zurich University.

Sampson, G. (2003). *Treebanks. Building and Using Parsed Corpora*, chapter 2: Thoughts on two decades of drawing trees, pages 23–41. In Abeillé (2003).

Samuelsson, Y. (2007). Automatic Phrase Alignment Aided by a Third Language. A Machine Learning Approach. GSLT ML.

Samuelsson, Y. and Volk, M. (2004). Automatic Node Insertion for Treebank Deepening. In *Proceedings of the 3rd Workshop on Treebanks and Linguistic Theories (TLT2004)*, Tübingen, Germany.

Samuelsson, Y. and Volk, M. (2007a). Alignment Tools for Parallel Treebanks. In *Proceedings of GLDV Frühjahrstagung 2007*, Tübingen, Germany.

Samuelsson, Y. and Volk, M. (2007b). Automatic Phrase Alignment: Using Statistical N-Gram Alignment for Syntactic Phrase Alignment. In *Proceedings of the Sixth Workshop on Treebanks and Linguistic Theories (TLT 2007)*, Bergen, Norway.

Schmid, H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing*.

Taulé, M., Martí, M. A., and Recasens, M. (2008). AnCora: Multilevel Annotated Corpora for Catalan and Spanish. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odjik, J., Piperidis, S., and Tapias, D., editors, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. ELRA.

Tinsley, J., Hearne, M., and Way, A. (2007). Exploiting parallel treebanks to improve phrase-based statistical machine translation. In *Proceedings of Treebanks and Linguistic Theories (TLT)*, Bergen, Norway.

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.

Zhechev, V. and Way, A. (2008). Automatic generation of parallel treebanks. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 1105–1112, Manchester.

A Part-of-Speech Tagsets

A.1 Spanish POS Tagsets: DVD vs. AnCora

Part of speech	POS type	POS	Nb of labels in AnCora
Adjective	Ordinal	AO	4
	Qualificative	AQ	12
Conjunction	Coordinate	CC	1
	Subordinate	CS	1
Determiner	Definite Article	DA	5
	Demonstrative	DD	6
	Exclamative	DE	1
	Indefinite	DI	6
	Numeral	DN	6
	Possessive	DP	14
	Interrogative	DT	3
Punctuation mark	Sentence end	F$	3
	Others	F	12
Interjection		I	1
Noun	Common	NC	10
	Proper	NP	5
Pronoun	Se	P0	5
	Demonstrative	PD	7
	Exclamative	PE	1
	Indefinite	PI	6
	Numeral	PN	5
	Personal	PP	27
	Relative	PR	8
	Interrogative	PT	5
	Possessive	PX	9
Adverb	General	RG	1

Part of speech	POS type	POS	Nb of labels in AnCora
	Negation	RN	1
Preposition		SP	2
Verb	Auxiliary	VA	31
	Main	VM	47
	Semiauxiliary	VS	30
Date		W	1
Abbreviation		Y	1
Number		Z	3
Foreign word		XF	1
14	Total	**35**	280

Table 20: Spanish POS tagsets: DVD vs. AnCora

A.2 POS Tag Mapping: TreeTagger to DVD-ES

```
{
    downcase_input_tags  = 0
    words_case_sensitive = 0
}
#DVD(new) constraint(word) TreeTagger(old)
-- * ACRNM
AQ * ADJ
RG * ADV
-- * ALFP
-- * ALFS
DA el ART
DA la ART
DA lo ART
DA los ART
DA las ART
DI un ART
DI una ART
DI unos ART
DI unas ART
F * BACKSLASH
# BEWARE: more general case before the more specific ones
# tt.CARD is also mact.Z when numeric like "2"
Z * CARD
```

```
# but tt.CARD is mact.DN when written out like "dos"
# or rather mact.PI for "uno"
PI uno CARD
DN dos CARD
DN tres CARD
DN cuatro CARD
DN cinco CARD
DN seis CARD
DN siete CARD
DN ocho CARD
DN nueve CARD
DN diez CARD
DN once CARD
DN doce CARD
DN trece CARD
DN catorce CARD
DN quince CARD
# etc...
CC * CC
CC * CCAD
CC * CCNEG
F * CM
Z * CODE
F * COLON
CS * CQUE
CS * CSUBF
# Sub conj introducing non-finite clauses tt.CSUBI are prep. mact.SP
SP * CSUBI
CS * CSUBX
F * DASH
PD * DM
DD esta DM
DD este DM
DD estas DM
DD estos DM
DD aquel DM
DD aquellos DM
DD aquella DM
DD aquellas DM
DD ese DM
DD esa DM
DD esos DM
DD esas DM
F * DOTS
-- * FO
F$ * FS
PT * INT
```

```
I * ITJN
F * LP
NC * NC
RN * NEG
NC * NMEA
W * NMON
NP * NP
AO * ORD
SP * PAL
SP * PDEL
XF * PE
-- * PERCT
-- * PNC
PP * PPC
DP * PPO
PP * PPX
SP * PREP
SP * PREP
SP * PREP/DEL
F * QT
DI * QU
# specific quantifiers tt.Q as indef pronouns mact.PI (when #PI > #DI)
# e.g. nada is always PI in our corpus but otro is mostly DI
PI algo QU
PI alguien QU
PI nada QU
# nadie is tagged as NC (un nadie?)
PI nadie *
PI uno QU
PR * REL
F * RP
PO * SE
F * SEMICOLON
F * SLASH
-- * SYM
Y * UMMX
VM * VCLIger
VM * VCLIinf
VM * VCLIfin
# tt.V.adj are past participles, most of them used as adjectives
# tt.VEadj rather verb mact.VM or adjective mact.AQ ?
VM * VEadj
# AQ * VEadj
VM * VEfin
VM * VEger
VM * VEinf
# tt.VHadj : VM or AQ ?
```

```
VA * VHadj
# AQ * VHadj
VA * VHfin
VA * VHger
VA * VHinf
# tt.VLadj is more often mact.AQ
AQ * VLadj
# VM * VLadj
VM * VLfin
VM * VLger
VM * VLinf
# tt.VMadj : VM or AQ ?
VM * VMadj
VM * VMfin
VM * VMger
VM * VMinf
# tt.VMadj : VM or AQ ?
VS * VSadj
VS * VSfin
VS * VSger
VS * VSinf
```

A.3 TreeTagger Evaluation: Confusion Matrix

```
CLASSIFICATION STATISTICS
*************************
Date: Thu Sep 17 23:47:14 CEST 2009
Key File (with reference tags):
----------------------------
   /home/anne/liz/data_ES/postagtests/ES_20090914.mantag
Test File (with wrong tags):
----------------------------
   /home/anne/liz/data_ES/postagtests/t05m15l21_ES_20090914.auttag
+----------------------------------------+-----------------+
|confusion matrix                        |     error ratio |
+--------------+-------------+------------+--------+--------+
| wrong tag    | correct tag |  frequency | rel ct |  total |
+--------------+-------------+------------+--------+--------+
| NP           | NC          |        111 |   4.48 |   1.02 |
| CS           | PR          |         73 |  97.33 |   0.67 |
| AQ           | VM          |         67 |   6.03 |   0.62 |
| XF           | NP          |         53 |  10.27 |   0.49 |
| AQ           | NC          |         53 |   2.14 |   0.49 |
| NC           | NP          |         46 |   8.91 |   0.42 |
| NC           | AQ          |         30 |   4.45 |   0.28 |
```

CS	SP	23	1.34	0.21
--	NC	19	0.77	0.17
--	NP	16	3.10	0.15
NP	XF	14	60.87	0.13
VM	NC	14	0.57	0.13
AQ	DI	11	3.67	0.10
--	CC	11	2.81	0.10
--	SP	10	0.58	0.09
XF	SP	9	0.52	0.08
CS	RG	9	3.64	0.08
XF	NC	9	0.36	0.08
RG	PT	7	77.78	0.06
XF	VM	7	0.63	0.06
VM	VA	6	14.63	0.06
DI	PI	6	33.33	0.06
NC	VM	6	0.54	0.06
NC	XF	5	21.74	0.05
AQ	RG	5	2.02	0.05
NP	AQ	5	0.74	0.05
VM	F	5	0.51	0.05
RG	SP	4	0.23	0.04
XF	RN	4	4.65	0.04
DD	PD	4	33.33	0.04
RG	AQ	4	0.59	0.04
RG	NC	3	0.12	0.03
F	NP	3	0.58	0.03
PR	RG	3	1.21	0.03
AQ	NP	3	0.58	0.03
NC	Y	2	10.53	0.02
Z	W	2	100.00	0.02
XF	PP	2	4.76	0.02
SP	NC	2	0.08	0.02
Z	NC	2	0.08	0.02
NC	RG	2	0.81	0.02
Z	Y	2	10.53	0.02
SP	RG	2	0.81	0.02
NP	VM	2	0.18	0.02
RG	CS	2	0.90	0.02
F$	F	2	0.21	0.02
--	XF	1	4.35	0.01
VM	VS	1	1.05	0.01
DA	PP	1	2.38	0.01
DN	Z	1	0.79	0.01
DI	CS	1	0.45	0.01
XF	DA	1	0.10	0.01
AQ	XF	1	4.35	0.01
VM	AQ	1	0.15	0.01

RG	DI	1	0.33	0.01
RG	VM	1	0.09	0.01
NC	AO	1	25.00	0.01
NP	CC	1	0.26	0.01
DI	RG	1	0.40	0.01
PI	NC	1	0.04	0.01
SP	VM	1	0.09	0.01
SP	CS	1	0.45	0.01
DI	NC	1	0.04	0.01
NC	F	1	0.10	0.01
DI	DD	1	1.64	0.01
CC	NP	1	0.19	0.01
NP	SP	1	0.06	0.01
DI	AQ	1	0.15	0.01
RG	CC	1	0.26	0.01
AQ	F	1	0.10	0.01
all	all	704		6.47

total error ratio = proportion of current line to total error
relative error ratio ct = proportion of this confusion to correct tag
relative error ratio wt = contribution of this confustion to wrong tag

tag	present	found	wrong	missed	prec	recal	f-mea
--	48	105	57	0	45.71	100.00	62.75
AO	4	3	0	1	100.00	75.00	85.71
AQ	674	774	141	41	81.78	93.92	87.43
CC	391	379	1	13	99.74	96.68	98.18
CS	221	322	105	4	67.39	98.19	79.93
DA	974	974	1	1	99.90	99.90	99.90
DD	61	64	4	1	93.75	98.36	96.00
DI	300	299	11	12	96.32	96.00	96.16
DN	10	11	1	0	90.91	100.00	95.24
DP	63	63	0	0	100.00	100.00	100.00
F	975	969	3	9	99.69	99.08	99.38
F$	414	416	2	0	99.52	100.00	99.76
NC	2477	2355	93	215	96.05	91.32	93.63
NP	516	528	134	122	74.62	76.36	75.48
PO	131	131	0	0	100.00	100.00	100.00
PD	12	8	0	4	100.00	66.67	80.00
PI	18	13	1	6	92.31	66.67	77.42
PP	42	39	0	3	100.00	92.86	96.30
PR	75	5	3	73	40.00	2.67	5.00
PT	9	2	0	7	100.00	22.22	36.36

RG	247	248	23	22	90.73	91.09	90.91
RN	86	82	0	4	100.00	95.35	97.62
SP	1715	1674	6	47	99.64	97.26	98.44
VA	41	35	0	6	100.00	85.37	92.11
VM	1111	1054	27	84	97.44	92.44	94.87
VS	95	94	0	1	100.00	98.95	99.47
W	2	0	0	2	0.00	0.00	0.00
XF	23	87	85	21	2.30	8.70	3.64
Y	19	15	0	4	100.00	78.95	88.24
Z	127	132	6	1	95.45	99.21	97.30
macr-avg	10881	10881	704	704	85.44	80.77	80.91
micr-avg	10881	10881	704	704	93.53	93.53	93.53

B List of scripts

This appendix contains a list of all scripts used to build the parallel DVD treebank.

Preprocessing

- Tokenisation, sentence segmentation, pretagging and lemmatisation:
  - German original:
    `GermAnno`
  - DVD batch:
    `process_all_steps_GermAnno_DVD_manual.bat`
- Format conversion:
  `negra_to_tiger.py`
- POS tag mapping program: `tagfixes`
  - compiling of the mapping file:
    `tagfixes -c mapping.fx`
  - fixing the tags with compiled mapping file:
    `tagfixes -f mapping.fxc CORPUS.oldtags > CORPUS.newtags`

Automatic Annotation

- POS tagging (within Annotate):
  model generation: `generate-cmm -1 modelname 3 CORPUS.negraformat`
  generated model: `modelname-{edge|level(1|2|3)|pos|seq}.{123|lex}`
- Deepening
  - German: `enrich_DE_trees.perl`
  - Swedish: `enrich_SV_trees.perl`

Postprocessing

- Completeness check: `check_completeness.perl`
- Monolingual consistency check: `display_token_ranges.perl`
- Multilingual consistency check: `check_edge_relations.perl`
- Alignment check:`display_parallel_word_alignment.perl`

Lebenslauf

Persönliche Angaben

Anne Göhring
Dienerstrasse 81
8004 Zürich
anne_goehring@access.uzh.ch
Geboren am 18. März 1967

Schulbildung

1973–1979	Primarschule Collonge-Bellerive GE
1979–1982	Sekundarschule Cycle d'Orientation Bois-Caran GE
1982–1986	Kantonsschule Collège Calvin GE, Typus C
1986–1991	Informatikstudium ETH Zürich
2006–2007	Erasmus Austauschjahr in Spanien (Universidad Complutense Madrid)
seit 2001	Studium der Spanischen Sprach- und Literaturwissenschaft und Computerlinguistik an der Universität Zürich

Berufliche und nebenberufliche Tätigkeiten

1991–2000	Ecofin AG, Zürich
2000–2002	Teilzeitanstellung bei Eurospider AG, Zürich
2003–2006	Hilfsassistenz am Institut für Informatik, DDIS
2005–2006	Tutorate PCL I+II
2008–2009	Tutorate "Computergestützte Textanalyse", "Maschinelle Übersetzung und parallele Korpora", "Semantische Annotation paralleler Korpora" und "Multilinguale Textanalyse"
2001–2009	Vorstandsmitglied des Fachvereins für Computerlinguistik, seit 2005 Kassierin
2009–2010	Mitglied im OK für die Tagung der Computerlinguistik Studierenden TaCoS 2010 in Zürich