A Tree-to-Tree Model for Statistical Machine Translation

by

Brooke Alissa Cowan

B.A. American Studies
Stanford University (1994),
S.M. Electrical Engineering and Computer Science
Massachusetts Institute of Technology (2004)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2008

Author .....
Department of Electrical Engineering and Computer Science
May 23, 2008

Certified by .....
Michael J. Collins
Associate Professor
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

A Tree-to-Tree Model for Statistical Machine Translation

by

Brooke Alissa Cowan

Submitted to the Department of Electrical Engineering and Computer Science
on May 23, 2008, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science and Engineering

Abstract

In this thesis, we take a statistical tree-to-tree approach to solving the problem of machine translation (MT). In a statistical tree-to-tree approach, first the source-language input is parsed into a syntactic tree structure; then the source-language tree is mapped to a target-language tree. This kind of approach has several advantages. For one, parsing the input generates valuable information about its meaning. In addition, the mapping from a source-language tree to a target-language tree offers a mechanism for preserving the meaning of the input. Finally, producing a target-language tree helps to ensure the grammaticality of the output.

A main focus of this thesis is to develop a statistical tree-to-tree mapping algorithm. Our solution involves a novel representation called an *aligned extended projection*, or AEP. The AEP, inspired by ideas in linguistic theory related to tree-adjoining grammars, is a parse-tree like structure that models clause-level phenomena such as verbal argument structure and lexical word-order. The AEP also contains alignment information that links the source-language input to the target-language output. Instead of learning a mapping from a source-language tree to a target-language tree, the AEP-based approach learns a mapping from a source-language tree to a target-language AEP.

The AEP is a complex structure, and learning a mapping from parse trees to AEPs presents a challenging machine learning problem. In this thesis, we use a linear structured prediction model to solve this learning problem. A human evaluation of the AEP-based translation approach in a German-to-English task shows significant improvements in the grammaticality of translations. This thesis also presents a statistical parser for Spanish that could be used as part of a Spanish/English translation system.

Thesis Supervisor: Michael J. Collins
Title: Associate Professor

Acknowledgments

Michael Collins — Glorious Seven Bags Associate Professor, Martial World-Class Road Assembly Champion — it's so trite it kills me, but it's so true I feel obliged to say it: I could not have done this without you. Thank you so much.

Tommi "Most Amazing Smile" Jaakkola and Leslie "How Does She Do It All?" Kaelbling — My Committee: You didn't make me rerun experiments! You didn't make me re-write the whole thesis! You smiled and nodded during my defense![1] You are wonderful!

Victor Zue and whoever else happened to read my MIT application: Thank you for taking a chance on someone with a background in The Humanities.

And a whole host of others at MIT who make it hard to leave: Stephanie Seneff, Jim Glass, Lee Hetherington, and others in the Spoken Language Systems Group who have supported me from start to finish; My Officemates: Ed Filisko! Vlad Gabovich! Kate Saenko! Xialong Mou! Jason Rennie! John Barnett! Natasha Singh! Terry Koo! John Lee!; MSG (yeah, that's the best we could do: **M**ike's **S**tudent **G**roup): Luke Zettlemoyer, Xavier Carreras, Ariadna Quattoni, Natasha Singh, Terry Koo, Dan Wheeler, Jason Katz-Brown, Ali Mohammad, Percy Liang; and then there's: Sarah Finney, Meg Aycinena, Emma Brunskill, Natalia Hernandez-Gardiol, Ghinwa Choueiter, Paulina Varshavskaya, Raquel Urtasun (oh my god, I didn't realize there were so many women in this program!), Albert Huang, Matt Walter, Mario Christoudias, Brian Milch, Fawah Akwo, Shaunalynn Duffy, Colleen Russell, Teresa Cataldo; Louis-Philippe "LPLMNOP" Morency, Mike "Molty" Oltmans, Ivona "Big Foot" Kučerová!!!, Raquel "Rocky" Romano, Brett "Scooby" Kubicek (oh yeah, you guys already left!!); The Human Annotators: Sarah Finney, Matt Walter, Tom Kollar, Gremio Marton, Mario Christoudias, Meg Aycinena, Luke Zettlemoyer, Terry Koo, Harr Chen (actually, you guys made it easier for me to leave. thank you!); TIG (oh my god, nightmare situation: two weeks before my defense, the hard drive on my laptop fails, and these guys SAVED me.): Noah Meyerhans, Jon Proulx, Eric Schwartz, Arthur Prokosch, Anthony Zolnik, Ron Wiken[2], Henry Gonzalez; and last but not least: Rodney Daughtrey, Eric Jordan, Lisa and Alexey.

And then there are all the people in California who make it so damn hard to stay! Jocey, Lise, Marisa and Bern, Paul and Annette, Ron and Nanci, Sue and Rob, Holly and Bill,

[1] Actually, for all I know they were nodding off. The little I remember from my defense is fading fast.
[2] Nothing to do with the laptop, actually. Ron is just the nicest guy in the lab.

Thea, Frances, and of course...

MI MAMA Y MI PAPA. More on you later. First...

My newest family members (a lot has happened in the last SEVEN years!): Johanna, Tashi, Teta Silva, Teta Liza, Uncle Kevin, Baka, Kari, Lisa and Colt, Dominic, Andrew, Colin, Aunt Mary and Uncle Steve. I love you guys!

My oldest family members: Grum (I miss you), Grandpa Ernest, Grandpa Ben, Grandma Bess.

And now. Mi mamá y mi papa (The Potato) and Zach, my favorite only brother. I love you so much it makes my heart feel three sizes too small.

And finally. Matto. My best friend and my husband! There have been times when I doubted why I ever came here — to MIT where I didn't speak the language, to Massachusetts where the winter lasts eight months — but there's never been any doubt that I came here to meet you. I love you ferociously. I love you gently. Thank you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The goal of automatic translation (also called *machine translation*, or MT) is to translate text from one human language into another using computers. There are many ways to achieve this goal. This thesis develops a method that integrates syntactic information into a machine learning framework.

We follow other recent work in statistical MT by taking a supervised learning approach to the problem. We assume access to a corpus of translation examples: a *bilingual parallel corpus*. The parallel corpus contains pairs of sentences — one sentence in the *source language* and the other (its translation) in the *target language*. These examples serve as the training data for our translation model. The goal of MT in the machine learning context is to learn a model that can predict a translation in the target language given some novel text in the source language.

This thesis incorporates syntactic information into a machine learning framework using a *statistical tree-to-tree* approach. The tree-to-tree approach is one of a larger class of *syntax-based* approaches to translation. Syntax-based approaches may in general be considered disjoint from the class of *phrase-based* approaches, another very popular class of statistical approaches that solves the machine translation problem by making use of a probabilistic phrase-pair dictionary, often without direct access to syntactic information.

A statistical tree-to-tree approach can be described in two steps, also summarized in Figure 1-1:

1. The Parsing Step: The source-language text is parsed to generate a syntactic parse tree. The parse tree contains grammatical information about the input sentence.

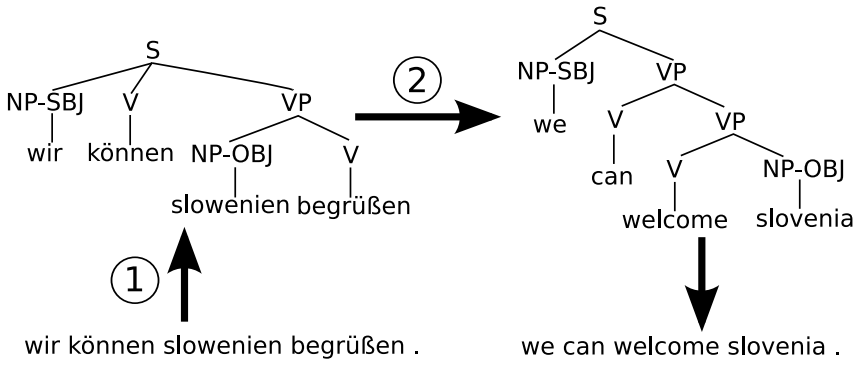


Figure 1-1: In a tree-to-tree statistical model, the source-language input (in this case German) is first parsed (1). Then, a target-language parse tree (in this case English) is predicted, and the leaves of the tree are read off the parse tree to produce a translation (2).

2. The Tree-to-Tree Mapping Step: The source-language parse tree is used to predict a target-language parse tree that explains the structure of the target-language translation. The translation is generated by reading off the leaves of the tree.

Each of these steps represents a significant challenge. The first, the parsing problem, is a classic natural language processing (NLP) problem that has traditionally garnered a lot of attention. In the past couple of decades, significant progress has been made in automatic parsing using statistical methods. Statistical parsers for a variety of different languages, with performance ranging from adequate to excellent, are now freely available. One part of this thesis is to develop a statistical parser for Spanish that could be used in a tree-to-tree translation framework.

The second step, the tree-to-tree mapping problem, is also a significant challenge. Ideally, the structural information should do two things: one, it should constrain the output in terms of the target-language syntax; and two, it should preserve any dependencies exhibited in the source-language counterpart. These requirements are what make the mapping step challenging, but while they do underscore the complexity of the tree-to-tree approach, they also illustrate its appeal. The use of syntactic parsing on the source-language side can give us valuable information about the meaning of the input. The use of syntax on the target-language side can help us to generate grammatically-correct output. A mapping between the two trees gives the model a mechanism for ensuring that the meaning of the

input is projected across into the output. The tree-to-tree mapping problem has not yet received as much attention as the parsing problem. A significant contribution of this thesis is to develop a solution to this problem.

1.1 A Solution to the Tree-to-Tree Mapping Problem

To solve the tree-to-tree mapping problem, we introduce a representation called an *aligned extended projection*, or AEP. Instead of directly predicting a target-language parse tree, our model learns to predict a target-language AEP from a source-language parse tree. The AEP is a parse-tree like structure with many properties useful for solving MT. First, it models clause-level syntactic phenomena — such as verbal argument structure and lexical word order — crucial for generating grammatical output. Second, it models alignment information between the source-language input and the target-language output, providing a mechanism for preserving the meaning of the input. The AEP prediction step is a major contribution of this thesis.

An AEP is like a parse tree in that it specifies the syntactic structure of the target-language translation. However, it has three key differences:

1. an AEP is always associated with a clause (where a clause has a single main verb, for instance an independent clause, a subordinate or relative clause, etc.);
2. an AEP may have "holes" — non-terminals that do not terminate in leaves;
3. an AEP is annotated with alignment information that relates the aforementioned holes to subtrees in a source-language parse tree.

Figure 1-2 shows an example AEP. In this case, the clause that the AEP is associated with is the German *wir können slowenien begrüßen* with gloss *we can slovenia welcome*. An appropriate English translation would be *we can welcome slovenia*. The English AEP, however, does not generate a complete translation. Instead, it produces something like 1 *can welcome* 2, where the noun phrases (NPs) labelled 1 and 2 represent holes in the translation. In this case, the two holes represent the subject and object arguments of the English clause's main verb *welcome*. That is, the AEP relays information about the argument structure of the English translation's main verb, and it places the verb and its arguments in a satisfying syntactic order. These properties help to ensure that the output

will be grammatical. The AEP also contains alignment information that couples the source-language input and the target-language output. In the example, 1 in the English AEP is connected to 1 in the German tree (*wir*), and 2 in the English AEP is connected to 2 in the German tree (*slowenien*). These links help to establish a mechanism for enforcing the preservation of meaning from the input sentence to the output translation.

The AEP is inspired in part by work in tree-adjoining grammars (TAG) on *extended projections* (Frank, 2002; Grimshaw, 1991). Extended projections are lexicalized TAG elementary trees, each of which is anchored by a single content word (that word is *welcome* in the AEP in Figure 1-2). Extended projections may also contain any function words associated with the content word, where function words may be auxiliary verbs, complementizers, prepositions, etc. In our example AEP, the word *can* is a function word that is incorporated into the AEP associated with *welcome*. In general, extended projections can be associated with any content word (e.g., nouns, adjectives, adverbs). However, in our AEP-based translation model, we focus on extended projections of verbs, which naturally correspond to clauses. Chapter 2 describes extended projections in more detail, and Chapter 5 explains how AEPs are constructed using verbal extended projections.

The decision to focus on the clause as the unit of translation is substantiated in part by ideas linguistic theory which emphasize the importance of verbs and clauses (e.g., (Haegeman & Guéron, 1999; Joshi, 1985)): verbs subcategorize for certain types and numbers of arguments; these arguments tend to be placed close to (i.e., within the same clause as) the verb. The verbal AEPs that form the basis of our approach model precisely this kind of information in target language clauses. In fact, incorporating verbal subcategorization information has been shown to contribute substantially to improved performance of lexicalized automatic parsing models of English (e.g., (Collins, 1999)).

AEPs are complex structures with many advantages for MT, but learning to map to them from parse trees is far from trivial. In this thesis, we use a discriminative linear structured prediction model (e.g., (Collins, 2002)) to solve this learning problem. Linear structured prediction models have many advantages: to name a few, they are simple to implement, well-grounded theoretically, and quite effective in practice. This type of model affords us a highly-flexible framework in which to define features that allow the model to make predictions involving syntactic correspondences between the source and target languages. The features of our model are crucial to its success: they enable it to capture

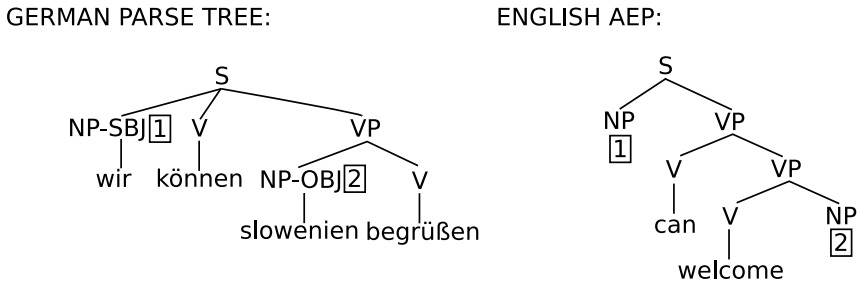


Figure 1-2: An aligned extended projection, or AEP, associated with the German *wir können slowenien begrüßen* (gloss: *we can slovenia welcome.*) The AEP aligns the English subject NP to the German *wir/we* and the object NP to *slowenien/slovenia*.

dependencies between the AEP and the parse tree, and within the AEP itself. We include a detailed description of a set of features for German-English translation in Appendix E. Chapter 2 presents background on discriminative linear structured prediction models as well as the perceptron algorithm (Rosenblatt, 1958; Freund & Schapire, 1998), which we use for training, and incremental beam search (Collins & Roark, 2004), which we use for decoding. To train the model, we need source-language parse trees and target-language AEP examples. To generate the training data, we take a parallel corpus of sentences and parse both the source and target-language examples. We then extract AEPs from the target-language parses using an algorithm described in Chapter 5.

As we have seen, an AEP may contain holes; it is not necessarily a complete parse tree and may not generate a full target-language translation. Furthermore, each AEP is associated with only a clause of the original source-language input, and not necessarily the whole sentence. Thus, in order to use AEPs to generate full translations, we add an auxiliary step to the tree-to-tree approach as previously defined. This third step — the generation step — translates any verbal arguments or modifiers of each English clause's main verb and combines clauses into whole sentences. Thus, translation using AEPs involves three steps:

1. The Parsing Step: The source-language text is parsed to generate a syntactic parse tree. The parse tree contains grammatical information about the input sentence.

2. The AEP Prediction Step: The source-language parse tree is used to predict one target-language AEP per clause.

3. The Generation Step: A translation is generated by translating arguments and modifiers, and combining the target-language clauses.

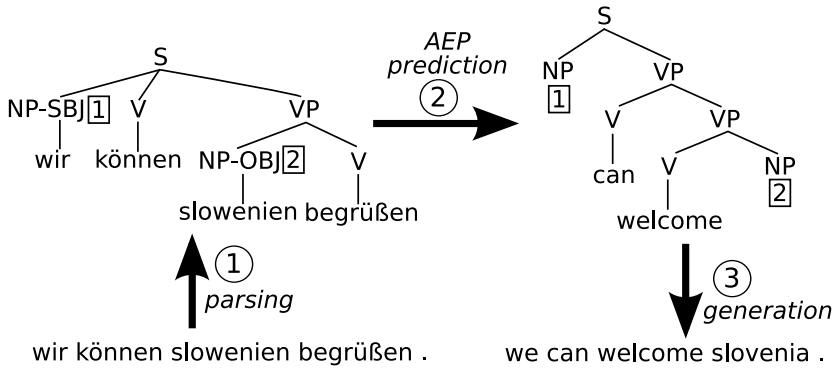


Figure 1-3: In tree-to-tree translation using AEPs, the tree-to-tree mapping problem is broken into three steps: in the first step, a target-language AEP is predicted from the source-language parse tree; in the second step, an AEP is predicted for each clause; in the third step, a translation is generated from the AEP.

Figure 1-3 presents an overview of the AEP-based translation process.

The remainder of this chapter is structured as follows: Section 1.2 provides contextual information for better understanding the statistical tree-to-tree problem we are addressing in this thesis as well as our method for solving it; Section 1.3 presents an end-to-end overview of AEP-based translation; Section 1.4 states the contributions of this thesis, and Section 1.5 contains a summary of each chapter.

1.2 Background

The AEP-based translation approach presented in this thesis is a solution to the more general statistical tree-to-tree problem, which in turn constitutes a syntax-based approach to MT. In this section, we first place statistical tree-to-tree approaches in particular, and syntax-based approaches in general, in some historical context. We then introduce a class of statistical approaches called *phrase-based* models. Phrase-based models are widely used today in the translation community; we also use them in this thesis to help generate translations from AEPs and as a baseline when we evaluate the performance of our system. In the last part of this background section, we discuss a crucial problem in MT known as *reordering* and explain how it is manifested in German and English, the language pair we choose to evaluate our model. German-English is a particularly befitting pair to test the AEP-based translation because of the interesting syntactic divergences between the two languages.

1.2.1 Some Historical Context

The use of computers to translate documents is a challenge that has intrigued people for at least half a century. However, statistical tree-to-tree approaches, and statistical approaches in general, have emerged only in the past couple of decades. The time line in Figure 1-4 summarizes the history of MT, including the emergence of statistical approaches.

Prior to statistical approaches, most MT systems were *rule-based*, meaning that their development consisted at least in part of the hand-crafting of rules to carry out translation. The number and character of these rules vary considerably from one rule-based system to another. For instance, one system might consist of rules based on a specific syntactic formalism, while another might not be based on any syntactic formalism at all.

One of the earliest MT projects, a collaboration between Georgetown University and IBM (the "Georgetown-IBM Experiment" in Figure 1-4), was an example of a very small rule-based system. The system, which specialized in the domain of organic chemistry, had a vocabulary of 250 word pairs and used six hand-coded rules to produce translations. However, in spite of its attenuated size, it generated a tremendous amount of enthusiasm for the nascent field of MT. A 1954 public demonstration of the Georgetown-IBM translation system, along with a claim that MT would be solved in three to five years, stimulated government funding for the subsequent decade.

The development of rule-based systems with large sets of hand-crafted, inter-dependent rules dominated the field through at least the mid-nineties. Rule-based models continue to play an important role in the field to this day. For instance, Systran, a leading commercial vendor of translation tools, uses rule-based methods.

Machine learning-based methods for MT emerged in the early 90s with the IBM models (Brown et al., 1990; Brown et al., 1993). A major appeal of machine learning approaches is their ability to automatically extract information and generalize from data. Rather than having to hand-code large sets of rules, people hoped that these new models would be able to learn from example translations by using statistics. The availability of large sets of example translations such as the Canadian Hansards corpus and the European Parliament corpus (Koehn, 2005a) has made supervised machine learning approaches to MT — such as the IBM approach — a real possibility.

A limitation of the IBM models was that they essentially performed translation in a

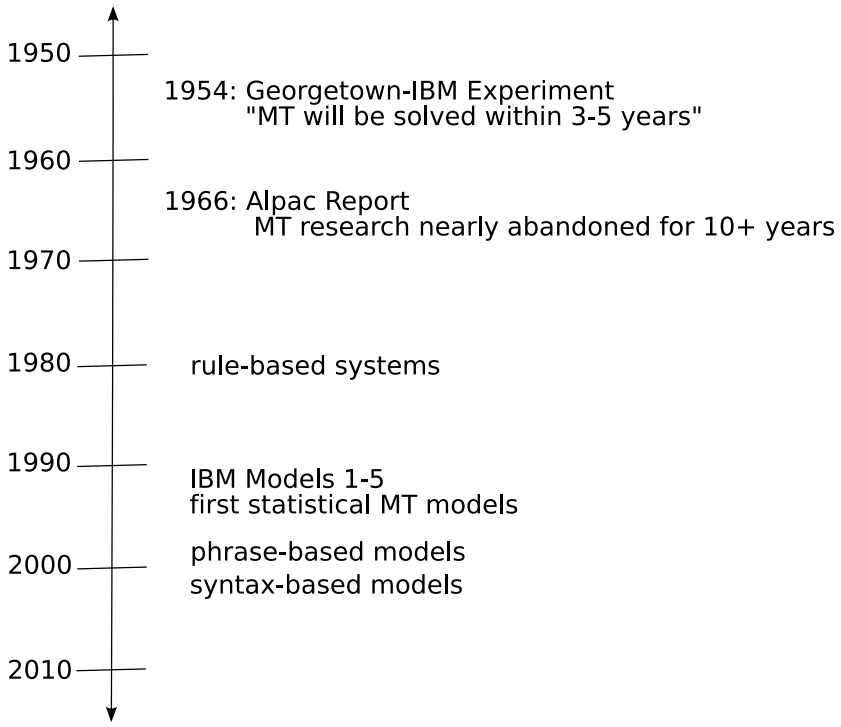


Figure 1-4: A brief history of machine translation. The emergence of machine learning approaches wasn't until the early 1990s with the IBM models. Contemporary statistical approaches include phrase-based and syntax-based systems.

word-for-word manner. Around the late 90s, a new class of statistical models emerged: *phrase-based models* (e.g., (Koehn, 2004; Koehn et al., 2003; Och & Ney, 2002; Och & Ney, 2000)). Phrase-based models directly addressed the limitations of the IBM models by constructing translations using mappings between source-language phrases and target-language phrases. These models have advanced the field of MT considerably in the past decade and are widely used today. For instance, the search engine Google uses phrase-based models in its translation tools.

Syntax-based models are an alternative statistical approach that emerged more or less contemporaneously with phrase-based models. In contrast to phrase-based models, which in general do not have a mechanism for directly incorporating the syntax of the source and target languages, syntax-based models do. Tree-to-tree approaches represent one kind of syntax-based model that learns a mapping from source-language parse trees to target-language parse trees. Other approaches are tree-to-string, which build mappings directly from target-language parse trees to source-language strings (e.g., (Menezes & Quirk, 2007; Collins et al., 2005; Xia & McCord, 2004)), and string-to-tree, which map from source-

language strings to target-language trees (e.g., (Marcu et al., 2006; Galley et al., 2006; Yamada & Knight, 2001)). A fourth class of approaches involves synchronous grammar formalisms, which learn a grammar that can simultaneously generate two trees (e.g., (Chiang, 2005; Wu, 1997)).

There are really only a handful of researchers who have done work in tree-to-tree approaches (e.g., (Nesson et al., 2006; Riezler & Maxwell, 2006; Ding & Palmer, 2005; Gildea, 2003)). Other tree-to-tree methods, and syntax-based models in general, are covered in more detail in Chapter 3.

1.2.2 An Introduction to Phrase-Based Models

Among contemporary machine learning approaches to MT, phrase-based systems (e.g., (Koehn, 2004; Koehn et al., 2003; Och & Ney, 2002; Och & Ney, 2000)) play an important role. They are widely used, highly competitive, and considered to be among the state-of-the-art in statistical approaches today. In this thesis, we use a phrase-based model both as a baseline against which to compare our AEP-based system, and as a sub-component of AEP-based translation (in the generation step). Because the phrase-based model plays a leading role in our work, we include a short discussion of it here. More on how phrase-based systems work can be found in Chapter 3.

Phrase-based models make use of bilingual phrase-pair dictionaries similar to the one in Figure 1-5. (In a real phrase-pair dictionary, there would be a distribution over possible English translations for each German entry.) A *phrase pair* is a bilingual pairing between two corresponding phrases — one in the source language and the other (its translation) in the target language. A *phrase* in this context is any substring of words in a sentence. There are usually no restrictions on phrases other than the number of words they are allowed to contain, a practical constraint imposed for the sake of efficiency. In particular, phrases are not necessarily constrained to adhere to syntactic boundaries. For example, for German and English, phrase-based alignments may include such pairings as *familienlebens* with *family life*, *auch ein* with *a*, and *konkretes problem* with *specific problem*.

One advantage of considering phrase pairs is that they can capture translations that are not word-for-word, or one-to-one. A good example is the German-English pairing *familienlebens* (one word) and *family life* (two words). A second advantage of phrase pairs is that they often encode context which can help with word-sense disambiguation. The context

gestatten	allow
gestatten sie mir	**please allow me**
sie	you
mir	me
mir auch	me
auch	also
auch ein	**a**
ein konkretes	a specific
konkretes problem	**specific problem**
problem	problem
aus	from
aus dem	from the
aus dem bereich	**in the area**
des familienlebens	**of family life**
familienlebens	family life
zu erwähnen	**to mention**
.	**.**

gestatten sie mir auch ein konkretes problem aus dem bereich des familienlebens zu erwähnen . $\Longrightarrow$ please allow me a specific problem in the area of family life to mention .

Figure 1-5: Phrase-based translation systems rely on a bilingual phrase-pair dictionary similar to the one at the top of the figure. In reality, the dictionary would have a distribution over possible English translations of each German phrase entry, whereas for simplicity we have only shown one possible English translation. The phrase-based translator has three major decisions to make when producing a translation: 1. how to segment the input into phrases; 2. which translation to select for each phrase; and 3. how to order the phrases. In the example, the system has chosen the following segmentation: [*gestatten sie mir*] [*auch ein*] [*konkretes problem*] [*aus dem bereich*] [*des familienlebens*] [*zu erwähnen*] [.]. It has chosen to place the translations of each one of these phrases in an order that is monotonic with respect to the original German ordering.

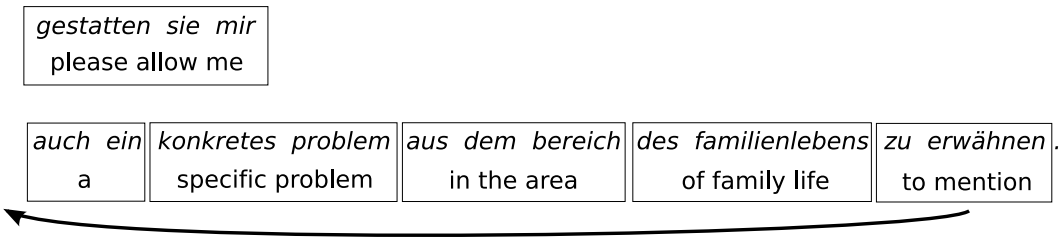


Figure 1-6: The reordering problem in machine translation refers to the phenomenon when the ordering of words in the target-language translation does not match that of the words in the source-language input. In this example, the phrase *zu erwähnen/to mention* appears at the end of the German clause but at the beginning of the English clause.

in which a word appears can contain important clues to its meaning. For instance, in the phrase-pair dictionary in Figure 1-5, the word *aus*, commonly translated as *from*, is better translated as *in* when seen in the context *aus dem bereich* (*in the area*).

In the example in the figure, the phrase-based system translates the sentence *gestatten sie mir auch ein konkretes problem aus dem bereich des familienlebens zu erwähnen* by segmenting the input into phrases, translating the phrases, and choosing an order for the translated phrases. In this particular case, the system has chosen to place the English phrases in an order that is monotonic with respect to the original German ordering, generating *please allow me a specific problem in the area of family life to mention.*

A correct translation of the German sentence in Figure 1-5 would be *please allow me to mention a specific problem in the area of family life.* In order to produce this output, the phrase-based system would have to translate the German phrases in the following order: [*gestatten sie mir*] [*zu erwähnen*] [*auch ein*] [*konkretes problem*] [*aus dem bereich*] [*des familienlebens*] [.] Figure 1-6 demonstrates that the phrase-based system would have to move the phrase *zu erwähnen/to mention* across four intervening phrases in the second clause of the sentence. Since phrase-based systems may either penalize translations that move phrases around too much or explicitly limit the absolute distance that phrases can move (compromising the exactness of search for efficiency), the output can tend to mimic the word ordering of the input.

Based on our observation of phrase-based output, this kind of long-distance reordering is something that phrase-based systems tend to get wrong. Chapter 3 provides a more in-depth analysis of some other types of syntactic errors we have frequently observed in the output of phrase-based systems.

1.2.3 The Reordering Problem for German-English and Other Language Pairs

Reordering of the type seen in Figure 1-6 is very frequent in German-to-English translation (which is the particular translation task we tackle in Chapters 5 and 6). One reason for this is that German and English can have very divergent ways of structuring clauses. First of all, German handles verb phrases much differently than English. In German independent clauses, the finite verb in a verb phrase is often placed in the second position of the clause, and the remaining verbs, if they exist, are placed at the end of the clause. In certain cases, such as when an independent clause is preceded by a subordinate clause, or in interrogatives, a verb must be placed in the first position of the sentence. This means that in independent clauses, there can be an arbitrary number of modifiers separating the finite verb in the first or second position of the clause and the remaining verbs at the end. In subordinate and relative clauses (such as the second clause in Figure 1-6), all verbs come last.

Consider, for example, the following German independent clause (here, "German" is the source-language sentence; "gloss" is a word-for-word translation; and "reference" is a human-generated, gold-standard translation):

GERMAN: sie selber haben die mutige entscheidung getroffen .

GLOSS: they alone have the courageous decision made .

REFERENCE: they alone have made the courageous decision .

In the German, the finite auxiliary verb *haben* is separated from the main verb *getroffen* by the object *die mutige entscheidung* to satisfy German's syntactic requirements for the placement of verbs.

In contrast, English has a different set of syntactic requirements: English syntax usually demands that the verb phrase be placed after the subject and before the object of the clause. This suggests that when translating from German to English, some verbs may have to be moved across an arbitrary number of intervening arguments and modifiers for a satisfactory translation.

As a matter of fact, the reordering problem gets even worse for the German-English pair. This is due to the flexibility that German exhibits in placing arguments like subject and object. Consider, for instance, the following example:

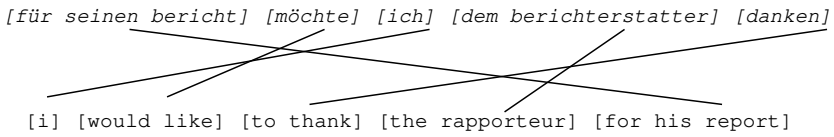


Figure 1-7: The relative positioning of the verbal modifiers and arguments changes considerably in translation from German to English.

```
[für seinen bericht] möchte [ich] [dem berichterstatter] danken
[ich] möchte [dem berichterstatter] [für seinen bericht] danken
[dem berichterstatter] möchte [ich] [für seinen bericht] danken
```

Figure 1-8: In German, arguments and modifiers can move around positionally in the sentence without significantly altering its meaning. In the example, the subject (*ich*), object (*dem berichterstatter*), and a prepositional-phrase modifier (*für seinen bericht*) illustrate such behavior.

GERMAN: für seinen bericht möchte ich dem berichterstatter danken .

GLOSS: for his report would like I the rapporteur to thank .

REFERENCE: i would like to thank the rapporteur for his report .

The English translation of this sentence involves a considerable amount of reordering of the top-level arguments and modifiers in the sentence (see Figure 1-7): the subject *ich* moves from after the German modal verb *möchte* to the front of the English translation; the prepositional-phrase modifier *für seinen bericht* moves from the front of the sentence to after the object in the English. Figure 1-8 shows three variations of the German sentence introduced above with roughly the same meaning, illustrating how the arguments and modifiers are relatively free to move around. In contrast, as we have seen, the placement of phrases in English is generally more constrained, the subject of an English sentence almost always coming before the verb, the object almost always after.

German and English certainly do not constitute the sole language pair with structural divergences that lead to significant reordering challenges. In fact, it is probably fair to say that every language pair will involve some reordering, and many pairs will involve major long-distance reordering of the kind exhibited by German and English. That is to say, long-distance reordering is a problem that any successful MT system will have to solve. The tree-to-tree AEP system described by this thesis directly addresses reordering phenomena by explicitly modeling the placement of sentential elements such as verbs and their arguments and modifiers in the target-language output. The fact that German and English exhibit

these kinds of structural divergences makes this pair particularly appropriate and interesting to use as a testbed for our approach.

1.3 AEP-Based Translation

In this section, we present an overview of AEP-based translation. We begin with an example that demonstrates how AEPs directly address the reordering problem. We then describe the work involved in implementing an AEP-based system for a new language pair. Finally, we preview the results described in Chapter 6 on German-to-English translation. Based on these results, we conclude that AEP-based translation produces more grammatical output than a phrase-based system with almost no syntactic information. We feel these results are very encouraging for future work in AEP-based translation.

1.3.1 An Example

Earlier in this chapter, we said that the AEP-based solution to the tree-to-tree translation problem involves three steps: parsing, AEP prediction, and generation. In this section, we delve a little more closely into each of these steps to see how the approach handles reordering phenomena when decoding a novel sentence. Throughout this section, we will work with the following example:

GERMAN:	ich hoffe , dass wir slowenien in der ersten gruppe de neuen mitglieder begrüßen können .
GLOSS:	i hope , that we slovenia in the first group of new members welcome can .
REFERENCE:	i hope that we can welcome slovenia in the first group of new member states .
AEP-BASED:	i hope we can welcome slovenia in the first group of new members states .

This example has two clauses. The first, *ich hoffe*, involves no reordering relative to English; the second, *dass wir slowenien...*, does. The auxiliary verb *können/can* and the main verb *begrüßen/welcome* need to be reordered with respect to one another, and the verb phrase needs to be moved up in the sentence so that it falls between the subject *wir/we* and the object *slowenien/slovenia*. We now show how AEP-based translation handles this

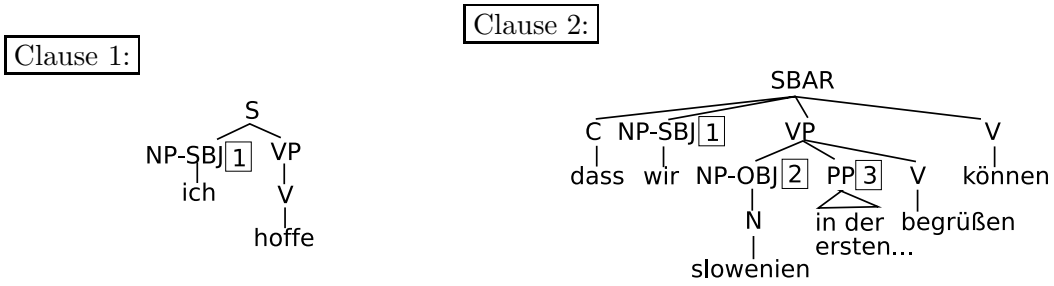


Figure 1-9: When decoding a sentence using AEP-based translation, the first step is to parse the input and break it into a sequence of clauses. In this example, the German sentence is split into two clauses. The numbered boxes represent the arguments and modifiers in each clause. In the first clause, the subject *ich* is the only argument. In the second clause, there are two arguments *wir* and *slowenien*, and one prepositional phrase modifier, *in der ersten gruppe der neuen mitglieder*.

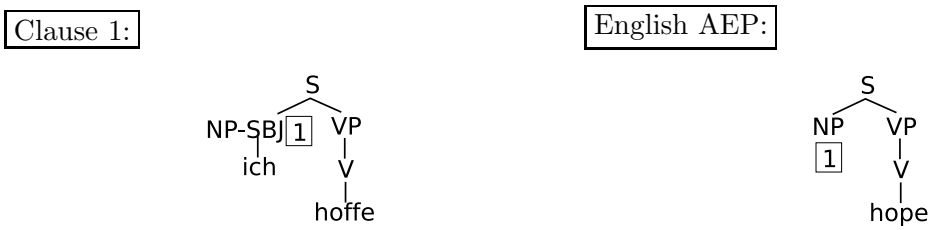


Figure 1-10: This AEP for the first clause has a subject argument for the main verb *hope*. The subject argument is aligned to the word *ich* in the German tree.

input in tree steps.

The Parsing Step The first step in decoding the German sentence is to parse it. In this thesis, we use a state-of-the-art statistical German parser (Dubey, 2005) to do this. After the sentence is parsed, it is broken into a series of clauses, and each of the arguments and modifiers is identified. Figure 1-9 shows the two clausal parse trees for our example; the first clause has one argument (labeled 1), and the second clause has two arguments (1 and 2) and a modifier (3).

The AEP Prediction Step In the second step, the AEP model predicts at least one AEP for each clausal parse tree. Figures 1-10 and 1-11 each show a potential AEP for the two parse trees in our example. The AEP in Figure 1-11 is particularly interesting in that it has correctly reordered the clausal elements that needed reordering.

The Generation Step Following AEP prediction, we can think of the state of the translation as being a mix of target-language and source-language words whose ordering should

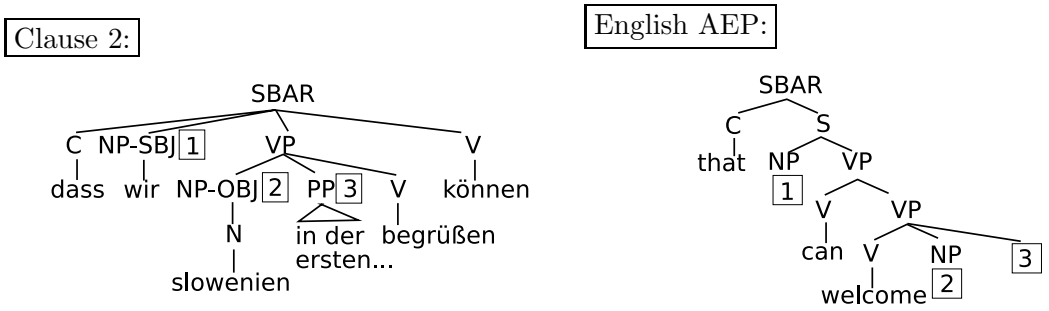


Figure 1-11: This AEP for the second clause has a subject argument (1), an object argument (2), and a modifier (3). The subject is aligned to *wir*, the object to *slowenien*, and the modifier to *in der ersten gruppe der neuen mitglieder*.

more closely resemble English syntax. In our example, the first AEP represents the string *ich hoffe*, and the second AEP represents the string *that wir can welcome slowenien in der ersten gruppe der neuen mitglieder*. The final step in AEP-based translation is to translate the arguments and modifiers and combine the clauses to form a sentence. In this thesis, we use a phrase-based system (that of (Koehn et al., 2003)) to generate lists of candidate translations for each argument and modifier. In Chapter 5, we develop a reranking-based method for selecting a translation from each list, and in Chapter 6 we develop a more sophisticated method involving finite-state machines. Figure 1-12 gives an overview of the finite-state method. In this approach, candidate argument and modifier translations are represented as lattices. These lattices are then used as building blocks to construct AEP lattices. Since the AEP-prediction model can output n-best lists of AEPs, we use n-best AEP lattices to construct a sentence-level finite-state machine. The sentence lattice — which includes scores on the edges that are not shown in the figure — can be searched using well-known dynamic programming methods such as the Viterbi algorithm.

1.3.2 Implementing an AEP-Based Translation System

We now present an overview of AEP-based translation from the point of view of the training and testing phases and the requirements of each. This should give the reader a sense of the work involved for implementing the approach for a new language pair. Figure 1-13 shows the major steps involved in training (left) and testing (right).

In the training phrase, a bilingual parallel corpus of sentences is first parsed. In order to carry out this step, a parser for each of the source language and target language is needed. Next, the parse trees are split into clauses. On the target-language side, this step

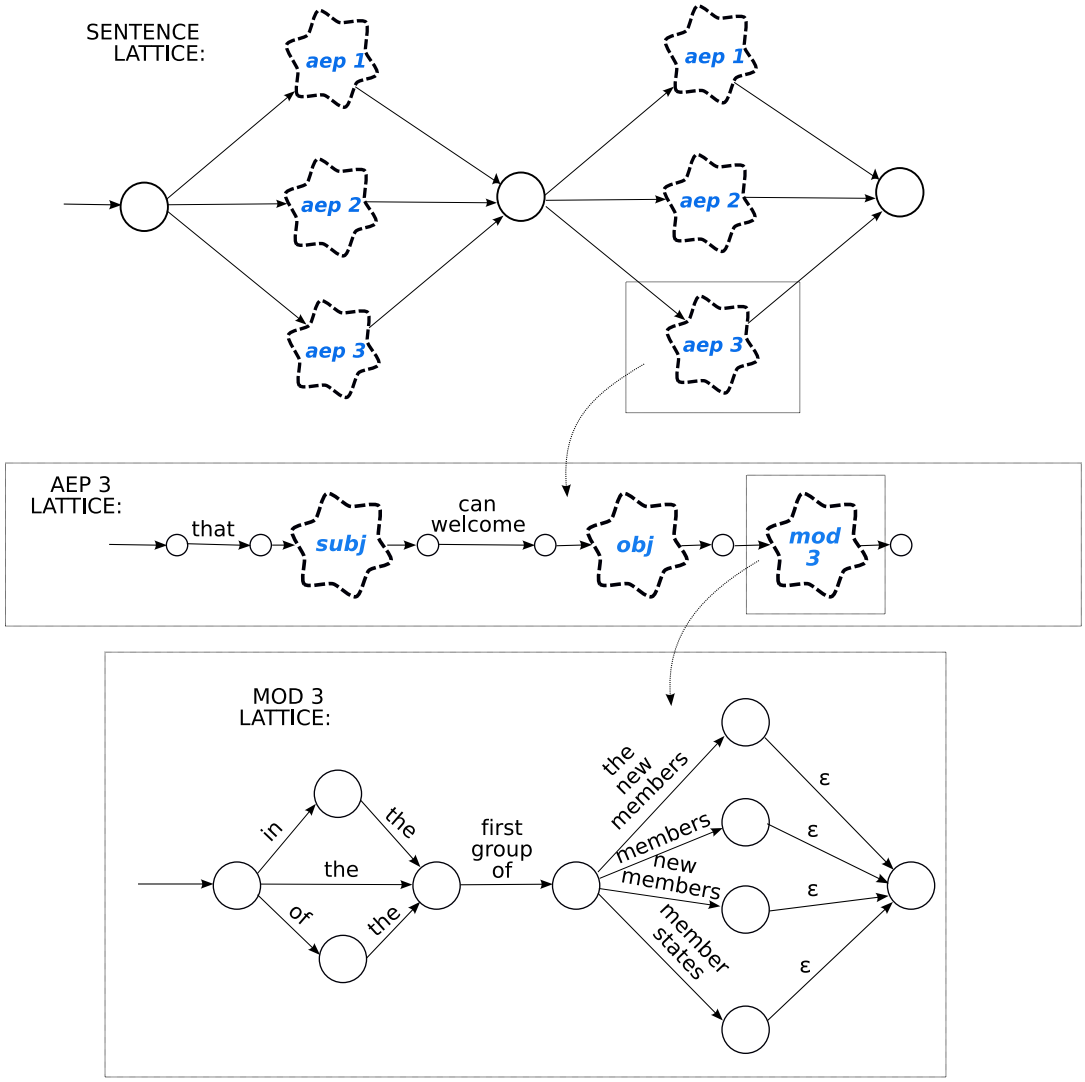


Figure 1-12: A sentence-level finite state machine, or lattice (top) is constructed from n-best AEP lattices ($n = 3$ in the figure). Each AEP lattice (middle) consists of the different pieces of the AEP, concatenated together. Arguments and modifier translation candidates are themselves represented by lattices (bottom). In reality, the lattices also contain scores on the edges that are not shown in the figure.

also involves the extraction of AEPs from the parsed clauses. It is likely in this step that the annotation schemes of the parse trees will be different (depending on the annotation schemes of the language-dependent parsers). This implies that the code used to split the source-language trees could be different from the code used to split the target-language trees. It also implies that for any new languages, new code will have to be developed. Once we have parsed source-language clauses and corresponding target-language AEPs, we can train the perceptron-based AEP prediction model. The perceptron-based training requires a set of language-pair-dependent features.

During testing, we begin with source-language sentences and then parse them. The parse trees are then split into clauses. For each clause, one or more AEPs are predicted. Finally, we generate target-language sentences from the AEPs.

1.3.3 Results

We evaluate the output of our system using both automatic evaluation and human judgments. Each of these methods has its advantages and disadvantages. Automatic metrics provide a quick and easy way of evaluating the performance of an MT system; however, they cannot be sensitive to everything we might like. This comes as no surprise: developing automatic metrics for the evaluation of machine translation output is an extremely challenging endeavor. If, for any translation, we could reliably and automatically determine its quality, we would be a lot closer to solving the machine translation problem.

In this thesis, we use the automatic metric BLEU (Papineni et al., 2001) to compare our system's output with that of a phrase-based system (Koehn et al., 2003). Figure 1-14 shows the BLEU scores (Papineni et al., 2001) computed over the output of the phrase-based system and the AEP-based system in a German-to-English translation task. According to BLEU, the AEP-based system is about 1.2 points behind the phrase-based system (a higher BLEU score is better). Roughly speaking, one BLEU point represents a minor but appreciable difference in the recovery of n-grams.[1] The scores in the figure have been computed on the test set described in Chapter 6.

The BLEU score has been shown to correlate well with human judgments of translation quality (Papineni et al., 2001). However, as a metric it is not necessarily sensitive to the

[1]BLEU is computed by taking the geometric mean of n-gram precisions and weighting it with a brevity penalty that is sensitive to the length of the translation.

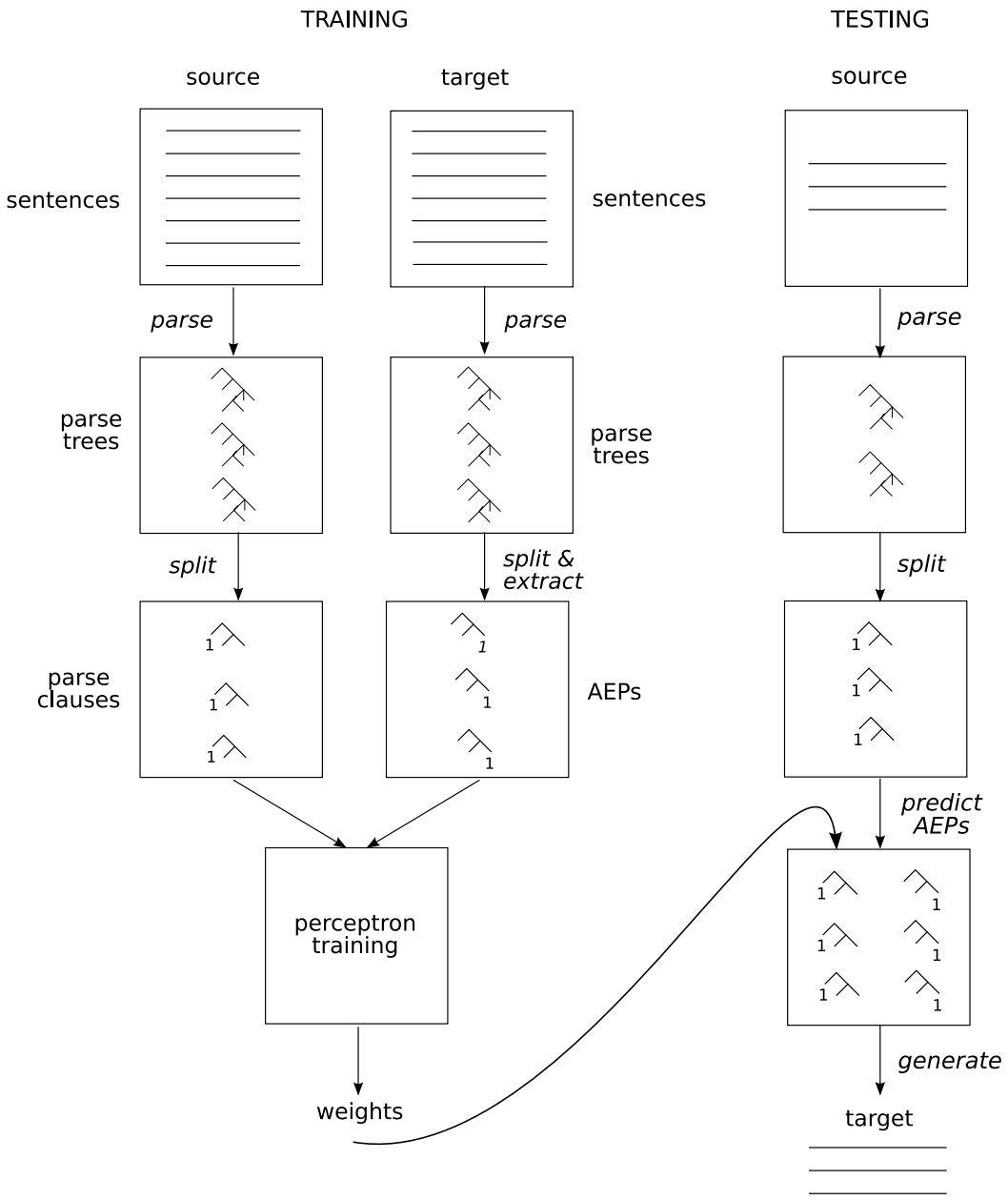


Figure 1-13: The training and testing phases of AEP-based translation.

	BLEU
PB	22.66
AEP	21.42

Figure 1-14: Phrase-based system (PB) and AEP-based system (AEP) BLEU scores computed on the test set of Chapter 6.

	FLUENCY		ADEQUACY	
	better	**worse**	**better**	**worse**
AEP	.45	.29	.36	.33

Figure 1-15: A summary of the results of a human evaluation comparing the output of a phrase-based system to the AEP-based system. The scores are averages over six judges and 1200 translation pairs (randomly selected from test set output with sentences between 10 and 20 words in length). Fluency reflects grammaticality and adequacy reflects overall quality, including the preservation of meaning. The numbers indicate that on average, in 45% of cases the judges thought the AEP-based system's output was more fluent that a phrase-based system's. When judging adequacy, on average the annotators found the AEP-based system's output of higher quality in 36% of cases.

kinds of syntactic improvements we are trying to introduce with the AEP-based method (see, for example, (Callison-Burch et al., 2006)). In Chapter 6 we describe a human evaluation that we conducted to compare the fluency (grammaticality) and adequacy (overall quality, including preservation of meaning) of our system's output and a phrase-based system's output in a German-to-English translation task. The advantage of using a human evaluation to measure performance is that humans are usually quite good at discriminating between syntactic differences. However, like automatic metrics, human evaluations are not perfect: it is difficult to get humans to agree on what a good translation is, and human evaluations are difficult and time-consuming to conduct. That said, human evaluations may be the best way we have at this point to assess the performance of our syntax-based AEP system.

Results of the human evaluation we conducted are summarized in Figure 1-15. The scores indicate that the judges found AEP-based translation to be more fluent than phrase-based translation. In terms of adequacy, the judges found very little difference between the two systems. In Chapter 6, we analyze these results more closely.

1.4 Contributions of this Thesis

This thesis offers the following contributions:

- An approach to machine translation that integrates source and target syntactic infor-

mation in a machine learning framework, thereby solving the statistical tree-to-tree translation problem.

  - An object for the representation of syntactic correspondences between the source-language input and the target-language translation called an *aligned extended projection* or AEP.
  - A language-dependent feature set for AEP-based German-to-English translation.
  - An algorithm for extracting AEPs from a bilingual parallel corpus (used to train the AEP prediction model).
  - Two methods for generating full translations from AEPs.

- A statistical parser for Spanish that could be used in an AEP-based translation system.

1.5 Summary of Each Chapter

Chapter 2 gives a detailed discussion of the background material referred to throughout this thesis. The first section of the chapter covers relevant topics in machine learning — in particular, what linear structured prediction models are, how to train them using the perceptron and exponentiated gradient algorithms, and how to use one to search for a solution after the training regimen is complete. Both the Spanish parser described in Chapter 4 as well as the AEP model in Chapters 5 and 6 are built using a linear structured prediction model. The second section of Chapter 2 focuses on topics from linguistics — in particular context-free grammar (CFG) and tree-adjoining grammar (TAG) and its variants.

Chapter 3 discusses previous work in MT. The chapter begins with an overview of the field. It then moves into a discussion of how phrase-based models work. Finally, it presents a literature review of some representative papers on syntax-based statistical MT models.

Chapter 4 presents a parser for Spanish. The model leverages both the morphological information of the Spanish language and the ability of a linear structured prediction model to define global features of parse trees to improve parsing performance.

Chapter 5 is the first of two chapters describing the AEP-based approach to statistical MT. This chapter gives a detailed description of the AEP model, including technical details related to the AEP object and the discriminative AEP prediction model. The chapter

presents results based on a reranking method for selecting a final translation from the predicted AEPs.

Chapter 6 presents a way of generating translations from AEPs using lattices. Results in this chapter are evaluated using the BLEU metric as well as an extensive human evaluation. An error analysis of the AEP-based system's output, with reference to the results of the human evaluation, is given.

Chapter 7 offers some suggestions for future work and then concludes the thesis.

Chapter 2

What You Need to Know

The background material required to understand the work in this thesis is fairly simple. There's a little bit of machine learning and a little bit of linguistics. This chapter covers what you need to know, including supervised learning, linear structured prediction models, and training and decoding using the perceptron and exponentiated gradient algorithms, context-free grammars and tree-adjoining grammars.

2.1 Machine Learning

Much of machine learning theory is about finding some way of making predictions about inputs. We want to devise a mathematical rule that will enable us to make such predictions. What we hope is that the prediction will be accurate.

In natural language processing (NLP), we often make predictions about objects involving words. For instance, in part-of-speech tagging, we have as input a string of words (a sentence) in a given language, and we want to develop a rule that will tell us the best part-of-speech tag for each word. For a simple input, say the sentence *the protesters amassed outside the governor's office*, the rule would be good, in a predictive sense, if it told us that *the* is a determiner, *protesters* is a noun, *amassed* is a verb, *outside* is an preposition, and so on.

Learning theory encompasses the kinds of rules we can use to make predictions like these, how we can learn these rules, and what guarantees, advantages, and disadvantages there are when we choose a particular type of rule or a particular method for learning it.

Whenever we set out to learn a predictive rule for a given task, we rely on training

data to help us do that. One way to classify learning methods is according to the type of training data we have. *Supervised* methods assume that our data set includes examples with both sample inputs (e.g., sentences) and corresponding sample outputs (e.g., sequences of part-of-speech tags). *Unsupervised* methods assume our data set consists of inputs without any corresponding outputs. In *semisupervised* methods, the data may be mixed, with some of the sample inputs labeled with corresponding outputs, and others unlabeled.

For the most part, the work in this thesis uses supervised learning methods and therefore requires corpora with example inputs and outputs. Chapter 4 uses a treebank of Spanish sentences and their corresponding syntactic trees (Torruella, 2004) to build a parser for Spanish. The supervised task in this context is to learn a function that will predict the best parse tree for an arbitrary Spanish sentence. The treebank is used as evidence in learning the function. Chapters 5 and 6 use a bilingual parallel corpus of German-English translation pairs (Koehn, 2005b). The supervised task in this context is to learn a function that will predict the best English translation of an arbitrary German sentence.

2.2 Linear Structured Prediction Models

In learning theory we often talk about models: building models, learning models, selecting models, etc. A model can be thought of as a representation of the space of inputs. For instance, the model in Figure 2-1 is a line that divides circles from squares, where the circles and squares represent our input space. Once we have this line, or model, we can devise a prediction rule that says *For any new input, if it falls below the line, it is a square; if it falls above the line, it is a circle.*[1] Because this model represents the input space with a line, it is an example of a *linear model*.

A linear model is one of the simplest models we can use. In a space with just two dimensions (think of a Cartesian grid where each dimension represents a different feature of the inputs, as in Figure 2-1), a linear model is just a line. In general, in a space with an arbitrary number of dimensions, it is a hyperplane. The work in this thesis uses linear models almost exclusively.[2]

[1] If it falls on the line, we can devise some method for deciding whether it should be a circle or a square. For instance, we can decide that by convention it should be a circle, or we can flip a coin and decide circle for heads.

[2] That is, all the important models underlying the novel work in this thesis are linear. We do of course make heavy use of parsers and other tools which are based on models that are not necessarily linear.

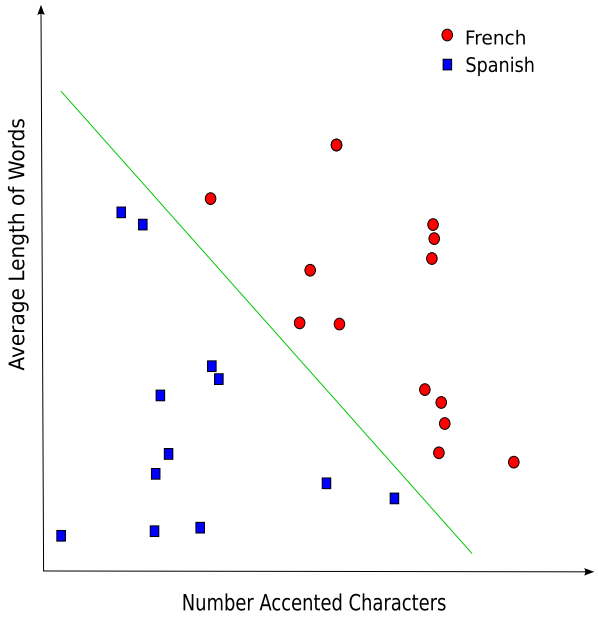


Figure 2-1: Example of a linear model where each point represents a document written in either French or Spanish. The line divides the input space into those inputs (documents) that fall below the line and are classified as Spanish, and those inputs that fall above the line and are classified as French.

It is perhaps simplest to introduce linear models in the context of *classification*. In the simplest case, we can imagine a classification problem in which there are just two classes, and we want to be able to classify any arbitrary input as a member of either class 1 or class 2. Say the inputs are documents written in one of two natural languages, Spanish and French, and we would like to be able to say whether any arbitrary document is written in Spanish or French. Furthermore, say we represent the inputs with two features, perhaps one feature that tracks the number of accented characters in the document, and another feature that tracks the average length of the words in each document (maybe we have some reason to believe that the words of one language are generally longer than the words of the other language). If we have a bunch of sample documents and we know whether each one is French or Spanish, then we can use these as training data to come up with a line such as the one in Figure 2-1. How we come up with that line is a separate problem called training the model; in this thesis we use two methods to train our linear models (see Section 2.3).

While linear models lend themselves easily to classification problems, for the purposes of this thesis, it's important to consider them in terms of a different kind of problem: the *structured prediction problem*. By this we mean that instead of trying to predict a class for our input, we predict a structure, like a parse tree or a translation, or a part-of-speech tag

sequence. The Spanish parsing work described in Chapter 4 trains a linear model to predict the best parse from a set of candidates generated by a probabilistic context-free grammar (PCFG); the aligned extended projection (AEP) prediction model used in Chapters 5 and 6 is underlyingly linear as well. It predicts the best target-language AEP given a source-language parse tree.

Figure 2-2 presents a geometric interpretation of the kind of linear structured prediction model that is used in this thesis. Imagine we're trying to generate translations, say English translations of Spanish sentences, and that we have at our disposal some function that will generate a set of candidate English translations given any Spanish sentence. In the figure, each of these candidate translations (actually, input-candidate pairs) is represented by a shape; the correct candidate is shown as a square, and the remaining candidates as circles.[3]

As in the classification scenario, we represent the candidates with features. Each dimension represents a different feature of the input-candidate pairs. For instance, imagine we had a Spanish-English dictionary to look up all pairs of words in a given input-candidate pair. We could then let the number of pairs that appear in the dictionary be the first feature we track. For the second feature, we could use the difference in the length between the input (Spanish sentence) and the candidate (English sentence). Each candidate is represented by a pair of values, and these pairs are plotted on the graph in the figure.

In the context of the structured prediction problem, a linear model defines a vector onto which we can project the candidates to get a ranking. In the figure, the vector labeled $\mathbf{w}$ is this vector, and the lines extending from each candidate to the vector illustrates the projection.[4] This projection defines a ranking among the candidates such that a superior candidate is farther away from the origin along the vector, while an inferior one is closer. In the figure, the upper graph shows one possible ranking, and the lower graph shows another possible ranking, each one induced by projecting the points onto the vector $\mathbf{w}$ (a different $\mathbf{w}$ for each graph). In the upper graph, the ranking is "correct" insofar as the correct candidate is also #1 in the ranking. In the lower graph, the ranking is incorrect.

Note that in general, we want to find a parameter vector $\mathbf{w}$ that is globally optimal over all training examples. Figure 2-3 illustrates a parameter vector that induces rankings over three different inputs. This parameter vector manages to predict the correct candidates

[3]Note that the set of points in the figure represents candidate translations for a single input. This is very different from the picture in Figure 2-1 in which each point represents a different input and its class.

[4]We assume without loss of generality that the vector is a unit vector.

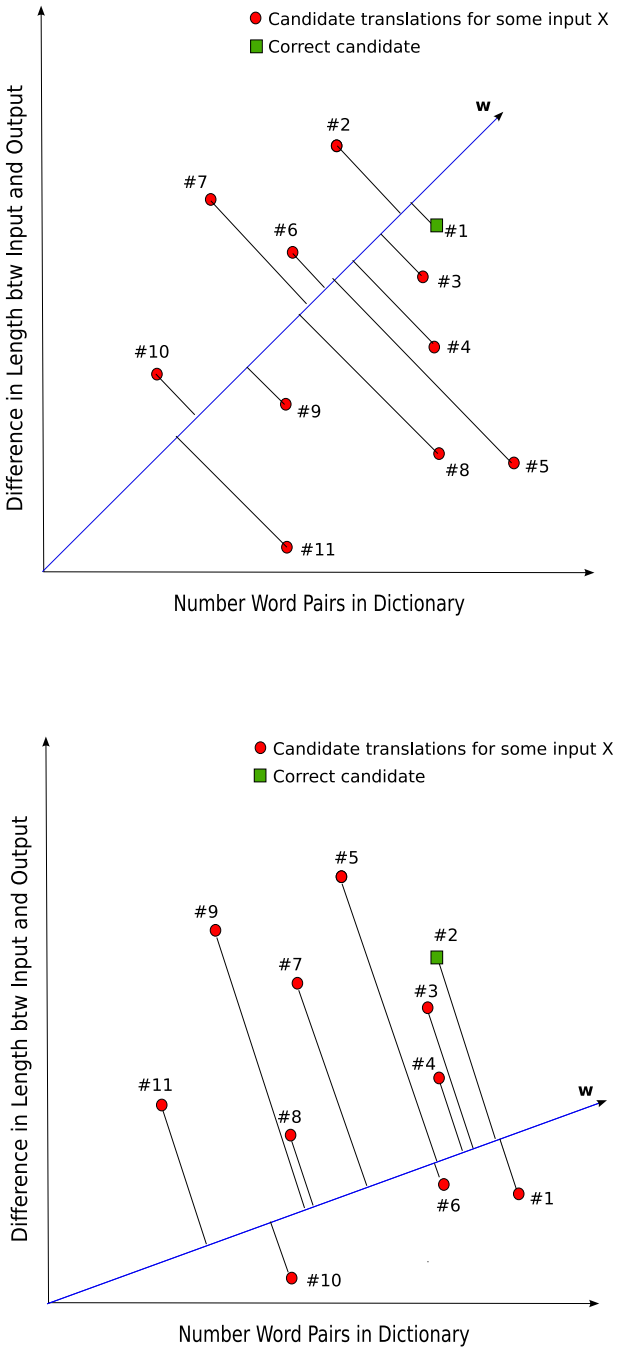


Figure 2-2: Linear models in the context of a structured prediction problem. Each point represents a different candidate output for the same input X. The square point represents the correct candidate. The candidates are represented by two features (labeled on the axes). A ranking over candidates is induced via their projection of the candidates onto the unit parameter vector $\mathbf{w}$. The best candidate is labeled "#1" and the worst "#11." The upper figure illustrates a parameter vector that correctly ranks the best candidate #1; the lower figure illustrates a parameter vector that does not correctly rank the best candidate.

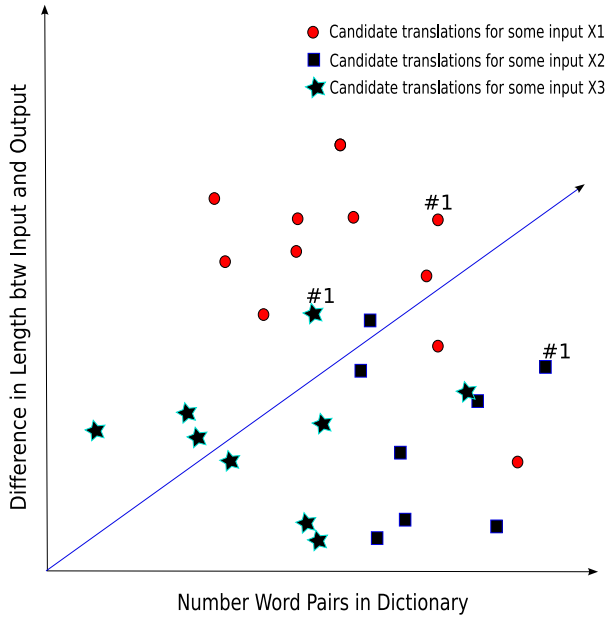


Figure 2-3: Choosing an optimal weight vector. During training, we try to find the parameter vector that optimally ranks each set of candidates for each example input x_i. The candidates labeled "#1" are the best candidates for each input. This particular parameter vector manages to predict the correct output for all three example inputs.

for all three. However, it is not always possible to find a globally-optimal vector that will correctly predict the best candidate on all examples.

2.2.1 Linear Structured Prediction Models: Formal Definition

In this section, we follow the framework for linear structured prediction models described in (Collins, 2002). Formally, the task is to learn a function $F : X \rightarrow Y$, where X is a set of possible inputs, and Y is a set of possible outputs. For example, in the case of parsing, the set X would be the set of all possible strings in a particular natural language, and Y would be a set of parse trees. In the machine translation task, X would be the set of all strings in one language, and Y would be the set of all strings in a second language. Our function tells us what the best output is for any given input — for instance, the best parse tree for some sentence in French, or the best Spanish translation for an English sentence.

In the linear structured prediction framework, we assume we have the following components:

- a set of m training examples (x_i, y_i) for $i = 1, \ldots, m$, where $x_i \in X$ and $y_i \in Y$,
- a function GEN that generates a set of candidates GEN(x) for each input x,

- a function $\mathbf{f}$ that maps a pair $(x, y) \in (X, Y)$ to a feature vector $\mathbf{f}(x, y) \in \mathbb{R}^{\mathbf{d}}$,
- a parameter vector $\mathbf{w} \in \mathbb{R}^{\mathbf{d}}$.

The first item in this list, a set of training examples, implies a supervised learning framework. The second item in this list — the set of candidates GEN(x) — corresponds to the set of points in our example in Figure 2-2. The feature vector $\mathbf{f}(x, y)$ corresponds to the two features (*number of word pairs in dictionary* and *difference in length between input and output*), and the parameter vector $\mathbf{w}$ corresponds to the vector $\mathbf{w}$.[5] We assume that there is some fixed ordering on the components of the d-dimensional feature vector $\mathbf{f}(x, y)$, such that its components and those of the parameter vector $\mathbf{w}$ are aligned. That is, for each feature $f_j(x, y)$, there is a corresponding parameter w_j.

We now define the form of the linear model to be

$$F(x) = \underset{y \in \text{GEN}(x)}{\text{argmax}} \ \mathbf{f}(x, y) \cdot \mathbf{w} \tag{2.1}$$

$\mathbf{f}(x, y) \cdot \mathbf{w}$ represents the inner product of the feature vector and the parameter vector: $\sum_{j=1}^{n} f_j(x, y)\mathbf{w}_j$. In our example in Figure 2-2, this inner product corresponds to the projection of each candidate onto the parameter vector $\mathbf{w}$ and can be thought of as a score for each input-candidate pair. The argmax operation selects the candidate with the highest score.

We discuss two methods for learning a parameter vector in Section 2.3. The decoding or search problem is to find the value of the function $F(x)$ that maximizes the score $\mathbf{f}(x, y) \cdot \mathbf{w}$ over a set of candidates GEN(x). We discuss a method for carrying out this search in Section 2.4.

2.2.2 Advantages to Linear Structured Prediction Models

Linear structured prediction models have many desirable characteristics. Foremost among these is formal simplicity. Equation 2.1 basically says *Find the candidate with the highest score*, where all we need to score each candidate is a feature function, a parameter vector, and a method for computing a dot product.

[5] We have assumed a unit vector in Figure 2-2 so that we can interpret the projection as a simple ranking. In general we can always convert any vector $\mathbf{w}$ to a unit vector by dividing by the norm (Euclidean length) of $\mathbf{w}$: $\mathbf{w}/\|\mathbf{w}\|$.

In spite of being formally quite simple, linear structured prediction models are representationally rich: we can incorporate arbitrary features into these models, and we can use them for a lot of different natural language tasks.[6] We can often obtain superior performance by using features that are sensitive to the linguistic information inherent in the data. The work in this thesis leverages the versatility of the linear structured prediction model by developing an extensive set of features for parse reranking in Chapter 4, and for tree-to-tree translation in Chapters 5 and 6. Because they are able to incorporate arbitrary features, linear structured prediction models are sometimes called *feature-based models*.

2.3 Training Algorithms

Given some example input-output pairs and a model, we need a way to use the examples to train our model. In the case of linear models, this means that we need to find values for the components of the parameter vector $\mathbf{w}$. In the context of classification, this is equivalent to choosing a hyperplane that will separate two classes of objects.[7] In the context of structured prediction, this means selecting the vector on which to project candidates in $\text{GEN}(x)$.

2.3.1 The Perceptron Training Algorithm

The perceptron is a simple yet effective training algorithm. The basic idea is to look at each sample input from the training examples and make a prediction y' according to the current parameters $\mathbf{w}$. Then we compare y' to the true output y_i. If it's correct, we do nothing, but if it's incorrect, we make a change to the parameters. We keep doing this — looping over the training set example by example — until we (hopefully) find a parameter vector that doesn't make any mistakes.

The original perceptron algorithm was described by (Rosenblatt, 1958). The algorithm was intended as a training method for classification problems such as those described in Section 2.2. The perceptron algorithm in Figure 2-4 is a variant that was developed for use in structured prediction problems (Collins, 2002) and is the version we use in this thesis.

The first line of the algorithm says that we're going to repeat the whole thing T times.[8]

[6] A very large number of features will complicate the search process. We address different search techniques in Section 2.4.

[7] The hyperplane can be defined in terms of the parameter vector, which is perpendicular to the hyperplane.

[8] We have a few choices as to how we pick the value of T. If we want, we can just pick an arbitrary value

Inputs: training examples (x_i, y_i)
Initialization: set $\mathbf{w} = \mathbf{0}$
Algorithm:

1 for $t = 1 \ldots T$:
2 for $i = 1 \ldots N$:
3 • $y' = \text{argmax}_{y \in \text{GEN}(x_i)} \mathbf{f}(x_i, y) \cdot \mathbf{w}$
4 • if $y' \neq y_i$
5 – set $\mathbf{w} = \mathbf{w} + \mathbf{f}(x_i, y_i) - \mathbf{f}(x_i, y')$

Output: parameter vector $\mathbf{w}$

Figure 2-4: The perceptron training algorithm for linear structured prediction problems. T is the number of iterations; N is the number of training examples in the training set, where each training example consists of a pair $(x_i, y_i) \in X \times Y$. $\mathbf{f}$ is a feature function that maps input-output pairs to a feature vector $\mathbf{f}(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{\mathbf{d}}$. $\mathbf{w} \in \mathbb{R}^d$ is the parameter vector whose components are set by the algorithm.

The second line says we're going to loop over the entire training set example by example. Lines 3, 4, and 5 form the heart of the algorithm. Basically they say *Find the candidate with the highest score according to the current settings of the parameters* (line 3). *If it's not the same as the output we know to be the best (i.e., y_i)* (line 4), *then make a change to the parameter vector* (line 5). The update in line 5 is additive and simple to implement.

Once the algorithm has completed, the final parameter vector $\mathbf{w}$ is used to evaluate, for any $x \in X$, the candidate in GEN(x) with the best score.

Two Variants: The Voted Perceptron and the Averaged Perceptron

For a long time, the perceptron algorithm was not taken seriously as a viable training method. Even though it seemed really promising initially, it was shown by Minsky and Papert that in reality, it couldn't even learn a simple XOR function (Minsky & Papert, 1969). In the early 90s, Freund and Schapire wrote a paper on a new variant of the perceptron called the *voted perceptron* (Freund & Schapire, 1998), and that caused a resurgence of its use in the field of NLP. In this thesis, we use something called *the averaged perceptron*, a variant whose efficacy has been demonstrated previously (Freund & Schapire, 1998) and (Collins, 2002). The averaged perceptron turns out to be an approximation to Freund and Schapire's voted perceptron.

(maybe each iteration takes a really long time, and we have only a limited amount of time to train, so we pick something small). Another approach is to see how well the algorithm does for different values of T by testing on a validation set.

Inputs: training examples (x_i, y_i)
Initialization: set $\mathbf{w}_0 = \mathbf{0}$, $c_0 = 0$, $j = 0$
Algorithm:
for $t = 1 \dots T$:
  for $i = 1 \dots N$:
    - $y' = \text{argmax}_{y \in \text{GEN}(x_i)} \mathbf{f}(x_i, y) \cdot \mathbf{w}$
    - if $y' = y_i$
    - set $c_j = c_j + 1$
    - else
      - set $\mathbf{w}_{j+1} = \mathbf{w}_j + \mathbf{f}(x_i, y_i) - \mathbf{f}_(x_i, y')$
      - set $c_{j+1} = 1$
      - set $j = j + 1$

Output: list of weighted parameter vectors $\langle (\mathbf{w}_0, c_0) \dots (\mathbf{w}_n, c_n) \rangle$

Figure 2-5: The voted perceptron training algorithm. This version of the algorithm is exactly the same as the one in Figure 2-4 except we keep track of the state of the parameter vector over the course of the algorithm. For each version of the parameter vector $\mathbf{w}_j$, $j = 0 \dots k$, we count the number of examples that are correctly predicted before a mistake is made.

The voted perceptron for structured prediction is given in Figure 2-5 (Collins, 2002).[9] It's exactly the same as the regular perceptron (Figure 2-4), with some added bookkeeping concerning the parameter vector $\mathbf{w}$. Think of $\mathbf{w}$ over the course of the training algorithm: each time an incorrect prediction is made under $\mathbf{w}$, it's considered a mistake and we update the parameters to make a correction. The voted perceptron just keeps track of each version of the parameter vector $\mathbf{w}_j, j = 0 \dots k$. Each time there's a new version (i.e., a mistake is made), the algorithm counts how many examples the current version $\mathbf{w}_j$ was able to predict correctly before updating to a new version $\mathbf{w}_{j+1}$.

To make a prediction for a novel input $x \in X$ using the output of the voted perceptron, we use the parameter vectors $\mathbf{w}_j, j = 0 \dots k$ to see which candidate they vote for, and produce as output the candidate with the highest vote:

$$F_j(x) = \underset{y \in \text{GEN}(x)}{\text{argmax}} \ \mathbf{w}_j \cdot \mathbf{f}(x, y) \tag{2.2}$$

$$Vote(y) = \sum_{j : F_j(x) = y} c_j \tag{2.3}$$

$$Final(x) = \underset{y \in \text{GEN}(x)}{\text{argmax}} \ Vote(y) \tag{2.4}$$

[9] This follows the form of the voted perceptron for classification given in (Freund & Schapire, 1998). Note that in this formulation, $c_0 \geq 0$. In contrast, for every other example (x_i, y_i), $i \neq 0$, $c_i \geq 1$.

where c_j is the count associated with each parameter vector $\mathbf{w}_j$. Equation 2.2 predicts the best candidate under each parameter vector $\mathbf{w}_j$ for a given input $x \in X$. Equation 2.3 tallies the vote for each candidate, and Equation 2.4 selects the candidate with the largest number of votes.

(Freund & Schapire, 1998) showed that the voted perceptron for classification performs a lot better than the original perceptron. There is, however, an obvious disadvantage to the voted perceptron: k, the number of different parameter vectors we have after training, might be really big, and we have to do k decodings for each input x.

What we do instead in this thesis, to avoid this potentially large number of decodings during prediction, is to approximate the voted perceptron with the averaged perceptron. The averaged perceptron has also been shown to perform a lot better than the original perceptron (Collins, 2002). In the averaged perceptron, we take the weighted parameter vectors from the output of the voted perceptron and we combine them using averaging to derive a new parameter vector $\mathbf{w}'$, where

$$\mathbf{w}' = \sum_{j=0}^{k} c_j \mathbf{w}_j / NT,$$

N is the size of the training set, and T is the number of iterations performed during training. We use the averaged parameter vector $\mathbf{w}'$ to predict the best output for x:

$$F(x) = \underset{y \in \mathrm{GEN}(x)}{\operatorname{argmax}} \ \mathbf{f}(x, y) \cdot \mathbf{w}'$$

This approach results in only a single decoding per input.

Properties of the Perceptron Algorithm

There are three important theorems related to properties of the perceptron algorithm for structured prediction tasks like those we address in this thesis (Collins, 2002). The first two are statements about how long it takes for the perceptron to find a good rule. A "good rule" in this case means one that is able to make correct predictions about the training data.

The first theorem says that the perceptron algorithm will find a good rule when trained on data that are *linearly separable*. Data are linearly separable when there exists some

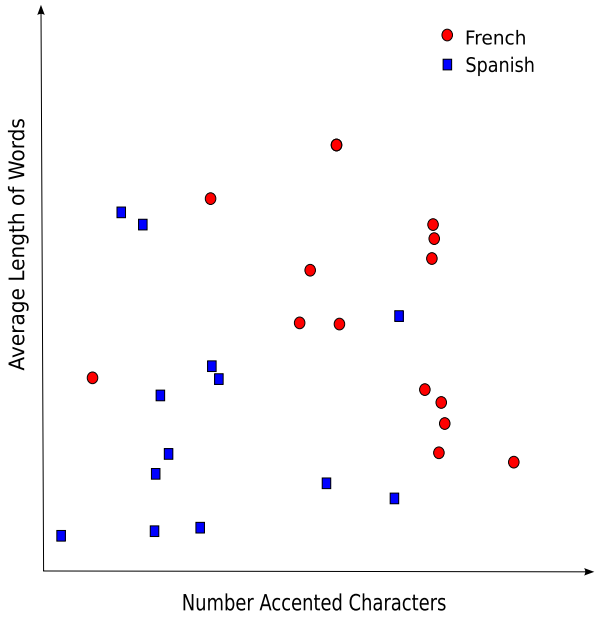


Figure 2-6: Training data that are not linearly separable.

parameter vector $\mathbf{w}$ that will produce the correct output y_i for each input x_i. An easy way to conceptualize linear separability is in the case of classification; for example, the data in Figure 2-1 is linearly separable because there exists a line that will perfectly classify all of the examples. In contrast, the data in Figure 2-6 is not linearly separable: there is no line that can separate the examples in the two classes. In the case of structured prediction, linear separability means that there exists a parameter vector $\mathbf{w}$ that will rank $y_i \in \text{GEN}(x_i)$ higher than any other candidate $y \in \text{GEN}(x_i), y \neq y_i$, for all i.

In addition to making guarantees in the case of linear separability, the first theorem also says how long it may take to find a good rule. The length of time is measured in terms of the number of mistakes the algorithm makes. A mistake in this context refers to a discrepancy between the predicted output and the actual output (y' and y_i, respectively, using the notation from Figure 2-4). The bound on the number of mistakes in the linearly separable case is given in terms of two quantities represented by R and δ:

$$number\ of\ mistakes \leq \frac{R^2}{\delta^2} \tag{2.5}$$

R is just a constant such that $\forall i, \forall z \in \text{GEN}(x_i), \|\mathbf{f}(x_i, y_i) - \mathbf{f}(x_i, z)\| \leq R$.[10] R is an upper bound on the difference between the length of the feature vector belonging to the correct

[10] $\|\mathbf{f}|$ denotes the Euclidean length of $\mathbf{f}$: $\|\mathbf{f}\| = \sqrt{\sum_i \mathbf{f}_i^2}$.

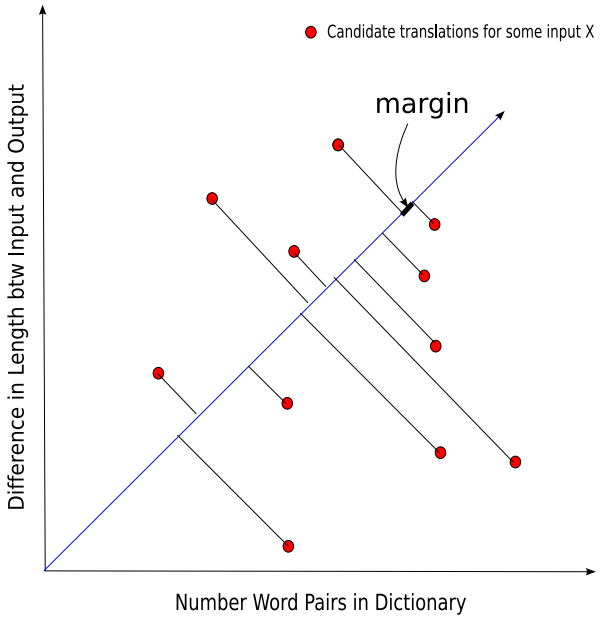


Figure 2-7: Training data containing a single example, where the margin δ is small.

candidate and the length of the feature vectors belonging to all other candidates. δ is a constant known as the *margin* of the data. The margin is a lower bound on the distance between the correct candidate $y_i \in \text{GEN}(x_i)$ and all the other candidates in $\text{GEN}(x_i)$. Formally, training data are linearly separable with margin δ if there exists a parameter vector $\mathbf{U}$, $\|\mathbf{U}\| = \mathbf{1}$, such that $\forall i, \forall z \in \text{GEN}(x_i), z \neq y_i, \mathbf{U} \cdot \mathbf{f}(\mathbf{x_i}, \mathbf{y_i}) - \mathbf{U} \cdot \mathbf{f}(\mathbf{x_i}, \mathbf{z}) \geq \delta$. Figures 2-7 and 2-8 give a sense of what's going on geometrically by depicting the margin in the case where there is only one example in the training set. The datum in Figure 2-7 has a small margin relative to the datum in Figure 2-8. If there were more than one example in the training set, then the margin would be the minimum distance, over all the examples, between the correct output and the next-best output. Intuitively, it should be more difficult to find a parameter vector that will correctly rank candidates in data sets with smaller margins. The bound supports this intuition.

The second theorem is similar to the first, except that it deals with data that are not linearly separable. See (Collins, 2002) for details.

The third theorem states how well the prediction rule that is generated by the perceptron training algorithm *generalizes* to novel inputs. If the algorithm makes only a small number of mistakes during training, then the resulting parameter vector is likely to generalize well to new inputs. The theorem and proof are given in (Collins, 2002).

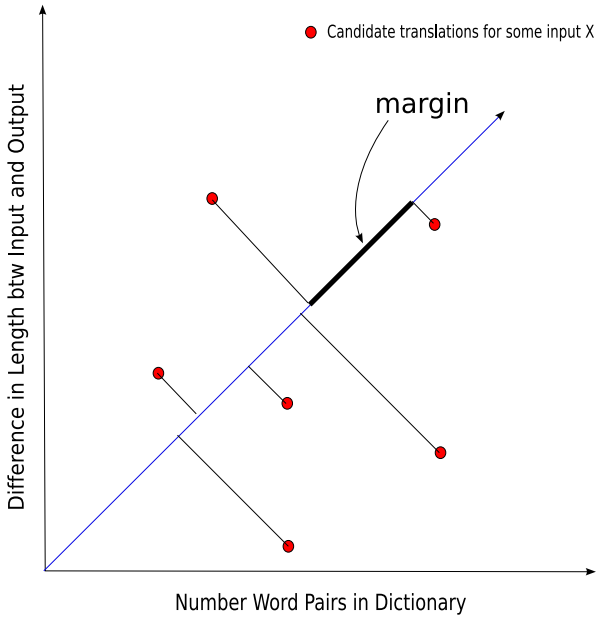


Figure 2-8: Training data containing a single example, where the margin δ is larger.

2.3.2 Exponentiated Gradient

The exponentiated gradient (EG) algorithm (Bartlett et al., 2004) is an alternative method for training a linear structured prediction model. It is this training method that we choose in Chapter 4 to train our Spanish parsing model to select the best parse from a list of candidates.

EG is a technique for solving an optimization problem with a specific objective criterion $\mathcal{L}(\mathbf{w})$ related to the large-margin training criterion of support vector machines (Cortes & Vapnik, 1995):

$$\mathcal{L}(\mathbf{w}) = \frac{c}{2}\|\mathbf{w}\|^2 + \sum_i \max_{y \in \text{GEN}(x_i)} [\ell_i(y) - (\mathbf{w} \cdot \mathbf{f}(\mathbf{x_i}, \mathbf{y_i}) - \mathbf{w} \cdot \mathbf{f}(\mathbf{x_i}, \mathbf{y}))]$$

Here, x_i is an input, y_i is the best output in a set of candidates $\text{GEN}(x_i)$. $\mathbf{w}$ is the parameter vector. The objective function has two terms. The first part, $\frac{c}{2}\|\mathbf{w}\|^2$, is a regularization term that ensures that the size of the weight vector w doesn't grow too large. In this term, c is a constant that is learned on a validation set. The second term makes use of a loss function $\ell_i(y)$ where y is an output. The only constraints on this loss function are that $\ell_i(y_i) = 0$ and $\ell_i(y) \geq 0$. The loss function we employ in our parsing work in Chapter 4 is

the same one described in (Collins & Koo, 2005):

$$\begin{aligned} \ell(y) &= Score(y_i) - Score(y) \\ Score(y) &= \frac{F-measure(y)}{100} \times Size(y) \\ Size(y) &= \text{number of constituents in } y \end{aligned}$$

The critical observation to make regarding the second term of the objective function is that if

$$\mathbf{w} \cdot \mathbf{f}(\mathbf{x_i}, \mathbf{y_i}) - \mathbf{w} \cdot \mathbf{f}(\mathbf{x_i}, \mathbf{y}) < \ell_\mathbf{i}(\mathbf{y})$$

then the objective incurs a penalty.

The solution to the objective function generates the optimal weight vector $\mathbf{w}^*$:

$$\mathbf{w}^* = min_\mathbf{w} \mathcal{L}(\mathbf{w})$$

2.4 Decoding

Training is the process by which we set the components of the parameter vector in our linear structured prediction model; *decoding* is the process we use to generate an output once we have a parameter vector[11]. For our purposes, decoding is the process by which we solve the *argmax* problem:

$$F(x) = \underset{y \in \text{GEN}(x)}{\text{argmax}} \ \mathbf{f}(x, y) \cdot \mathbf{w}$$

The difficulty posed by the *argmax* problem depends in part on the size of GEN(x). By definition, the *argmax* means that we have to find the candidate with the best score in all of GEN(x). If this set of candidates is small, then maybe the problem can be solved by exhaustive enumeration of the candidates: we score and then rank each one. This would be the brute force approach.

If, on the other hand, the set of candidates is very big (as with the NLP tasks we tackle in this thesis), it may not be possible to use brute force. Both parsing and translation are

[11] Note that even during training, we may need to perform decoding, so a hard distinction between training and decoding can be a little artificial. For example, the perceptron algorithm in Figure 2-4 requires a decoding of each example to compute the best candidate y' under the current parameters.

examples of natural language tasks where the size of GEN(x) may be exponential in the length of the input sentence x.

In the following sections, we introduce three decoding approaches that avoid brute force search. The first uses dynamic programming and is useful in circumstances where the form of the features is naturally constrained. However, since in our work we use arbitrary, unconstrained features, we discuss two alternatives to dynamic programming — reranking and beam search — that avoid the intractability of brute search. These are the two decoding methods that we use in our parsing and translation work.

2.4.1 Dynamic Programming

There are algorithms that circumvent the problem of search spaces that are too big to be explored by brute force. Among these is a technique called *dynamic programming*, which can be used in finding optimal solutions to problems that can be broken down into smaller subproblems with repeating substructure. While the work we present in this thesis does not emphasize dynamic programming approaches to decoding, we include a brief discussion here since the approach is used heavily in NLP.

When a dynamic programming approach can be applied to a problem, it may be possible to search the space of candidate outputs and find an optimal solution in significantly less time than searching by brute force. For example, the *CKY algorithm* (see, for example, (Manning & Schütze, 1999)) is a dynamic programming algorithm that can find the best parse for a sentence in $O(n^3)$ time, that is, time polynomial in the length n of the sentence (rather than exponential in the length of the sentence, as in the brute force approach).

Dynamic programming, when it can be used, is a great technique. However, efficient decoding with a dynamic programming algorithm like CKY cannot be done for arbitrary feature representations. Since the work in this thesis relies on the power of an arbitrary feature representation, we consider alternative methods.

2.4.2 Reranking

One alternative to dynamic programming is a *reranking* approach. It's fairly straightforward. The idea is that we complete the decoding task — say parsing (though it could be anything) — in two passes. In the first pass, we use a parser that is pretty good but not necessarily as powerful as one that would allow use of an arbitrary feature representation.

So, for instance, we could use a parser with a restricted set of features, facilitating efficient decoding using CKY. Instead of looking for the single best parse, however, we extract a set of top candidate parses — often called an n-best list.

In the second pass, we parse again, only this time, we use the n-best list as our search space. If its size is not too big, we can search the space using brute force with any parser we want. The approach is called "reranking" because the first-pass parser's ranking on the set of candidates might differ from second-pass parser's ranking. The second-pass parser is presumably a more powerful model that can select the correct candidate from the list.

Reranking is the decoding approach we use in Chapter 4 in our work on Spanish parsing. In that chapter, we use a weaker (i.e., one with fewer features) generative model based on a PCFG (see Section 2.5.1) as the first-pass parser, and a more powerful linear feature-based model as the second-pass parser.

2.4.3 Beam Search

A problem with reranking is that the first-pass model may not find the best solution in the n-best list it generates. If the best solution is nowhere on this list, then there is no way for the second-pass model to find it.

An alternative approach, called *beam search*, allows us to search the space of inputs in a single pass. Like reranking, it places no constraints on the features of the model. Also like reranking, it does not necessarily guarantee the discovery of an optimal solution. It can, however, speed up the search process considerably compared to brute search. Beam search is the technique we use in our translation work in Chapter 5.

The mechanics of beam search may vary in different contexts (see, for example, (Russell & Norvig, 2002) for an introduction to basic beam search). The definition of beam search that we use in this thesis is derived from work done on incremental parsing in (Collins & Roark, 2004). The basic idea is to decode a candidate structure, like a parse tree or a translation, in incremental steps, or as a sequence of N decisions $\langle d_1, d_2, \ldots, d_N \rangle$. Each decision d_i is a member of a set $\mathcal{D}_i$ representing all possible values of the ith decision. At each stage i, there is a list of incomplete candidates $\langle d_1, \ldots, d_i \rangle$ that need to be expanded. The expansion in this context means that from a candidate $\langle d_1, \ldots, d_i \rangle$ a new set of candidates each of the form $\langle d_1, \ldots, d_{i+1} \rangle$ will be derived. A *beam* of size m is applied to the list of candidates at each stage, such that the size of the list is never greater than m.

Collins and Roark assume two functions to carry out the incremental beam search:

- ADVANCE($x, \langle d_1, d_2, \ldots, d_{i-1}, d_i \rangle$)
- FILTER($\{\langle d_1, \ldots, d_i \rangle\}$

The ADVANCE function maps an input x and an incomplete candidate $d_1, \ldots, d_i$ to a set of candidates that incorporate decision $i+1$. Specifically, ADVANCE is used to enumerate the possible candidates for decision $i+1$. The function FILTER filters the set of candidates generated by ADVANCE such that the size of the new set is at most m. Thus, it maps a set of candidates $\mathcal{C}_{i+1} = \{\langle d_1, \ldots, d_{i+1} \rangle\}$ at stage i to a subset of $\mathcal{C}_{i+1}$ that is of an appropriate size. FILTER does its work by scoring each of the candidates and keeping only the top-ranking ones.

In the work in Chapters 5 and 6, the set of candidates we are generating during decoding consists of English AEPs from which translations can be derived. The candidates are broken down into a sequence of decisions, or parts, and at each step in the beam search, a single part is predicted. As the beam search unravels, only the top m partial candidates are retained at each step.

2.5 Linguistic Theory

This thesis borrows heavily from ideas in linguistic theory, particularly context-free grammars (CFG) and tree-adjoining grammars (TAG). CFG is the formalism underlying our first-pass parser in the reranking work in Chapter 4; TAG is the formalism underlying the tree-to-tree translation work in Chapters 5 and 6.

2.5.1 Context-Free Grammar

A *grammar* in the field of linguistics is a system that has atomic elements or objects as well as rules describing how to combine those elements. Combining the atomic elements in the grammar produces derived structures or objects of some sort that may or may not look like the original atomic elements.

Context-free grammar (CFG) and its statistical variant *probabilistic context-free grammar* (PCFG) are very simple formalisms that have been used liberally and with much success in parsing. For a very good formal introduction to CFGs, see (Sipser, 1997).

1.	S	→	NP VP	6.	D	→	*the*
2.	NP	→	D N	7.	N	→	*cat*
3.	VP	→	VP PP	8.	N	→	*dog*
4.	VP	→	V	9.	V	→	*leapt*
5.	PP	→	P NP	10.	P	→	*over*

Figure 2-9: A context-free grammar.

The atomic elements of a CFG are context-free rules of the form $X \rightarrow \xi$, where X represents a nonterminal symbol, and ξ represents a string of terminal and/or nonterminal symbols, or the empty string ϵ. In NLP, the nonterminal symbols are often phrasal categories such as *noun phrase* (`NP`) or *adverbial phrase* (`ADVP`), or part-of-speech tags such as *determiner* (`D`) or *verb* (`V`); the terminal symbols are words in the language, such as *water*.

The rule for combining the atomic elements of a CFG is a simple substitution rule. Specifically, we can replace a non-terminal γ appearing on the right-hand side of rule ρ_1 with the entire right-hand side of rule ρ_2 iff γ appears on the right-hand side of ρ_2. More concretely, say the rules in our grammar are those in Figure 2-9. Then we can substitute the right-hand side of rule 2 for the `NP` in rule 1, resulting in a new right-hand side: S → D N VP. We can continue substituting in this way until the right-hand side consists of only terminal symbols (words).

In a CFG, we can visualize the process of rule-substitution with a tree structure like the one in Figure 2-10. This tree is called a CFG *derivation tree* because we can use it to explain how the grammar generated the string *the cat leapt over the dog*. Specifically, we can see that the grammar used rule 1 once, rule 2 twice, rule 3 once, etc.; in addition we can see which substitutions were carried out (rule 2 substituted into rule 1, rule 3 into rule 1, etc.)

To form a PCFG from a CFG, we simply assign a probability to each one of the rules in the grammar. For instance, the PCFG in Figure 2-11 is a valid PCFG, where each rule can be seen as a conditional probability. For instance, rule 1 says $P(\text{NP VP}|S) = 1.0$, and rule 2 says $P(\text{D N}|\text{NP}) = 0.7$. The probabilities on the rules are subject to the following constraint: $\forall X, \sum_{\xi} P(\xi|X) = 1.0$.

Given probabilities on each one of the rules, we can associate derivations such as the one in Figure 2-10 with probabilities: $P(\text{tree}) = \prod_{\rho \in \{\text{rules in tree}\}} P(\rho)$, or equivalently $\log P(\text{tree}) = \sum_{\rho \in \{\text{rules in tree}\}} \log P(\rho)$. Probabilities on trees give us a quantitative

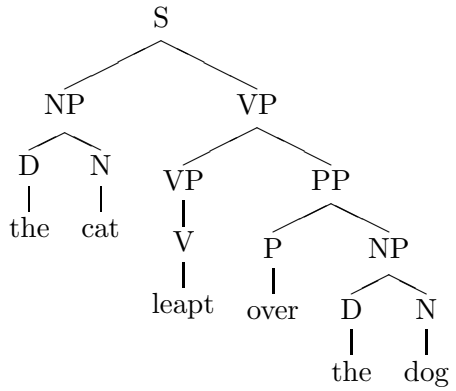


Figure 2-10: A context-free grammar derivation tree.

1.	S	→	NP VP	1.0	8.	D	→	*the*	0.5
2.	NP	→	D N	0.7	9.	D	→	*a*	0.5
3.	NP	→	N	0.3	10.	N	→	*cat*	0.7
4.	VP	→	VP PP	0.6	11.	N	→	*dog*	0.15
5.	VP	→	V NP	0.3	12.	N	→	*raccoon*	0.15
6.	VP	→	V	0.1	13.	V	→	*leapt*	1.0
7.	PP	→	P NP	1.0	14.	P	→	*over*	1.0

Figure 2-11: A probabilistic context-free grammar.

mechanism for comparing alternative parses for a given sentence. For a more complete discussion of PCFGs (how to learn the probabilities, etc.), see any good introductory text on NLP, such as (Manning & Schütze, 1999).

2.5.2 Tree-Adjoining Grammar

Tree adjoining grammar (TAG) (e.g., (Joshi & Schabes, 1996)) is a grammar formalism that has heavily influenced the work on tree translation we present in Chapters 5 and 6. In this section, we give an overview of the formalism and some of its variants.

One intuitive way to see the atomic elements of a TAG is as a splicing of CFG derivation trees. For instance, we can take the tree in Figure 2-10 and chop it up into pieces to form the *elementary trees* (i.e., the atomic units) of a TAG. The resulting set of elementary trees is depicted in Figure 2-12.

TAG elementary trees are classified as two basic types: *initial trees* and *auxiliary trees*. All phrasal-category nodes (e.g., `NP`, `ADJP`, etc) that appear on the frontier of an initial tree are labeled with a downarrow (↓), indicating that a substitution can take place there. This is also true of auxiliary trees, with the exception of a distinguished phrasal-type frontier

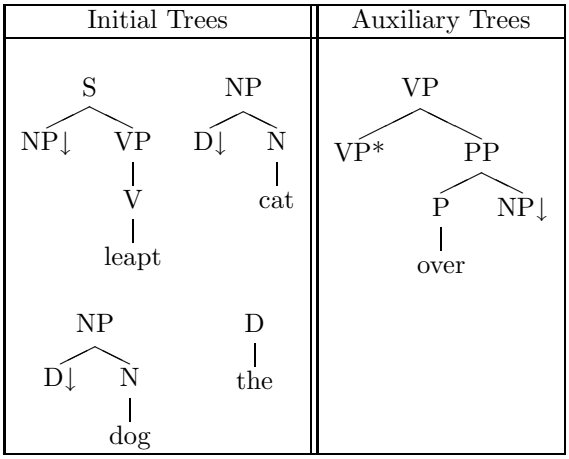


Figure 2-12: Some tree-adjoining grammar elementary trees.

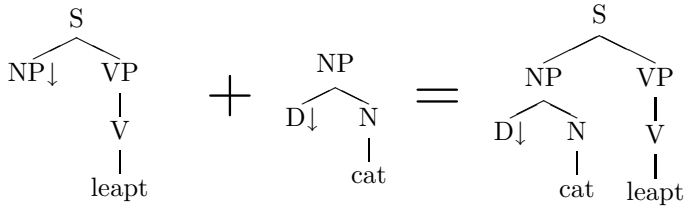


Figure 2-13: TAG substitution: the tree headed by NP is substituted into the tree headed by S at the NP↓ node.

node marked with an asterisk (*). This special node is called a *foot node*. Its label always matches that of the root node of the auxiliary tree to which it belongs.

There are two operations for combining TAG elementary trees: *substitution* and *adjunction*. Whenever a node is marked with a downarrow for substitution, it can be replaced by another tree that is headed by a root node with the same label, so long as that tree is either an initial tree or derived from an initial tree. An example of substitution in action is shown in Figure 2-13.

TAG adjunction allows the splicing of an auxiliary tree into any other type of tree (initial, auxiliary, or derived, where a *derived tree* is any tree resulting from the combination of two trees). For instance, we can adjoin the auxiliary tree in Figure 2-12 into the derived tree in Figure 2-13, as is depicted in Figure 2-14.

Lexicalized Tree-Adjoining Grammar (LTAG)

There are a lot of interesting properties of tree-adjoining grammars. For one thing, they can easily be lexicalized. A *lexicalized grammar* is one where each atomic structure is associated

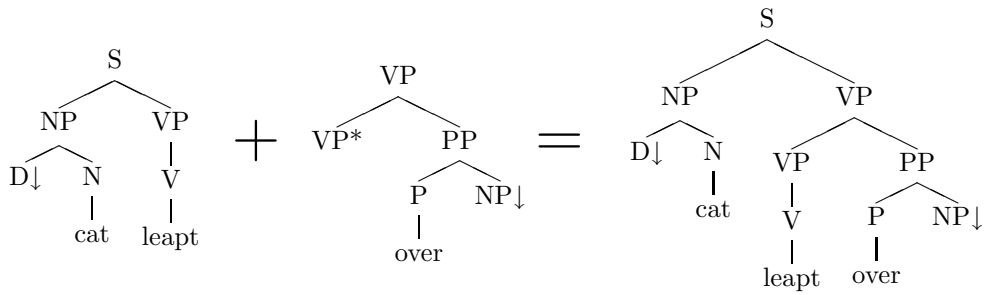


Figure 2-14: TAG adjunction: the auxiliary tree headed by VP is adjoined into the derived tree headed by S.

with at least one lexical item. For instance, each of the TAG elementary trees in Figure 2-12 has a frontier node labeled with a lexical item (*leapt, dog,* etc.). In contrast, only half of the PCFG rules in Figure 2-11 (those in the right-hand column) are lexicalized, and so the grammar as a whole is not lexicalized. In general, CFGs do not lend themselves naturally to lexicalization, while TAGs do (Joshi & Schabes, 1996). A TAG that is lexicalized is called a *lexicalized tree-adjoining grammar*, or LTAG.

One of the most important things about lexicalization is that it lets us express syntactic properties of lexical items. For instance, if we look at the initial tree anchored by the word *leapt* in Figure 2-12, it tells us that *leapt* requires a subject and doesn't take an object. In other words, we know something about the argument structure of this verb. Another way of saying this is that TAG elementary trees have an *extended domain of locality*, meaning that it is possible to express the long-distance syntactic dependencies of lexical items in atomic structures.[12] This means that in a sentence like *The cat who killed the bird who swallowed the mouse who nibbled the cheese leapt*, where an arbitrary number of relative clauses of the form *who V the N* might be inserted between the verb *leapt* and its subject *the cat*, TAG is still able to express the dependency between verb and subject.

The AEPs that we develop in Chapter 5 are like lexicalized TAG elementary trees in the sense that each of the elementary trees contains at least one lexical item (the main verb of the clause).

[12]Of course, it's not only lexicalization that gives us extended domain of locality; it's also the fact that TAG's atomic structures are trees of unrestricted depth.

Extended Projections

Extended projections (EPs) play a crucial role in the lexicalized tree-adjoining grammar (LTAG) (Joshi, 1985) approach to syntax described by Frank (Frank, 2002), and Frank's work plays a crucial role in the translation framework established in Chapter 5. EPs can be thought of as one way to define the elementary trees of an LTAG.

An EP is associated with exactly one content word (noun, adjective, etc.). As an example, a parse tree for the sentence *the cat who killed the bird leapt over the dog* would make use of EPs for the words *cat, killed, bird, leapt,* and *dog.* Function words (in this sentence *the, who,* and *over*) do not have EPs; instead, each function word is incorporated in the EP of some content word.

Figures 2-15 and 2-16 have examples of EPs. Each EP is an LTAG elementary tree which contains a single content word as one of its leaves. Substitution nodes (such as `NP` or `SBAR`) in the elementary trees specify the positions of arguments of the content words. Each EP may contain one or more function words that are associated with the content word. For verbs, these function words include items such as modal verbs and auxiliaries (e.g., *should* and *has*); complementizers (e.g., *that*); and *wh*-words (e.g., *which*). For nouns, function words include determiners and prepositions.

The AEPs in our translation work are extended projections of verbs. Each AEP contains at least one verb, slots for subject and object if they exist, and any function words associated with the verb. These verbal extended projections are used to model the argument structure of the verb.

Synchronous Tree-Adjoining Grammar (STAG)

Synchronous tree-adjoining grammar (STAG) is another interesting variant of TAG, with applications in machine translation and semantic parsing (Shieber & Schabes, 1990). The main idea in a STAG is to build elementary structures in the grammar that consist of aligned pairs of elementary trees. Some STAG elementary tree pairs are shown in Figure 2-17. The alignment between the trees is represented with numbered boxes. For instance, the first row of the table contains a pair of trees whose root nodes are aligned (S 1) and whose subjects are aligned (NP 2). This pair of trees represents a correspondence between the English verb *leapt* and its Spanish translation *brincó*, as well as the argument structures of the two verbs.

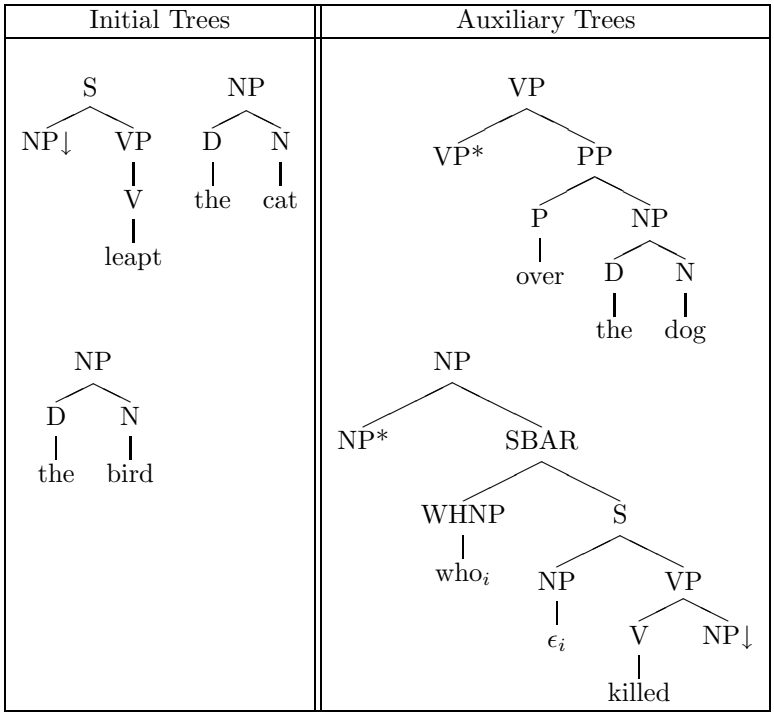


Figure 2-15: Extended projections for the verbs *leapt* and *killed*, and for the nouns *cat*, *dog*, and *bird*. These extended projections could be used to form the sentence *The cat who killed the bird leapt over the dog*, among others.

The operations in a STAG are the same as in a regular TAG (i.e., substitution and adjunction) except that they are synchronized with respect to the alignment specified in the elementary structures. That is, the aligned tree nodes must be substituted or adjoined simultaneously into corresponding tree structures, such that when completed, the derivation process will result in two aligned trees.

From STAG we borrow the notion of alignment between trees that are translations of one another. However, our translation work does not simultaneously generate two tree structures as in a STAG. Rather, we begin with a parse in the source language and generate a parse in the target language. However, our translation process does include a step where it is necessary to predict an alignment between two parsed structures.

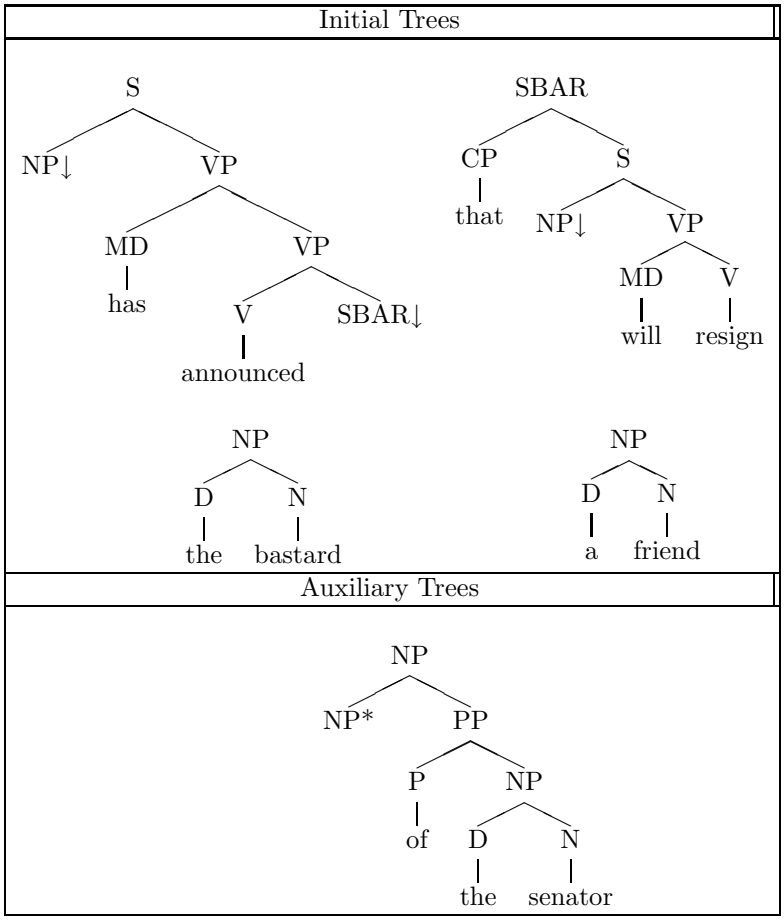


Figure 2-16: Extended projections for the verbs *announced* and *resign*, and for the nouns *bastard*, *friend*, and *senator*. These extended projections could be used to form the sentence *A friend of the senator has announced that the bastard will resign*, among others. This example was adapted from (Frank, 2002)

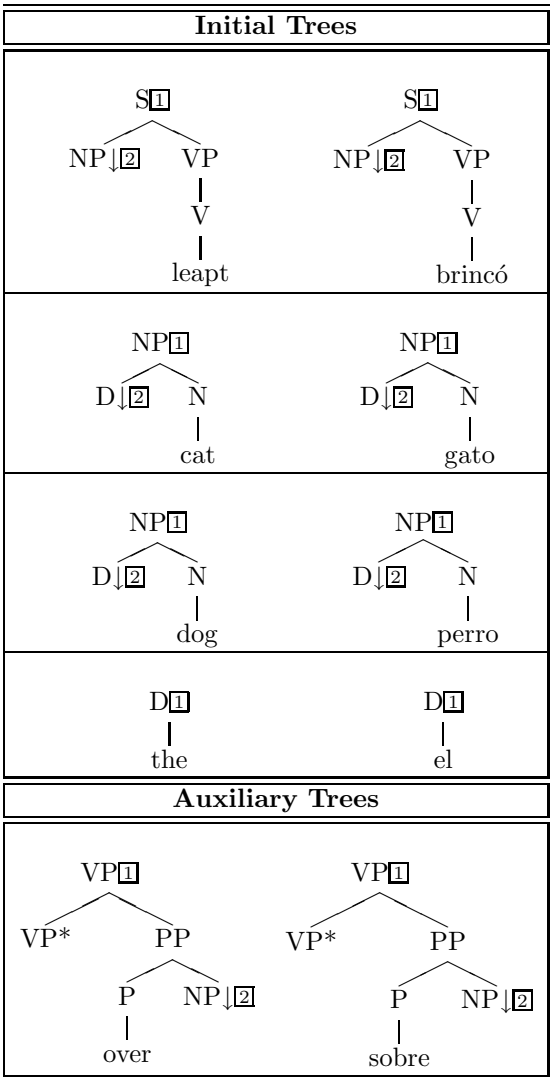


Figure 2-17: Some STAG elementary trees. Each row of the table represents a single atomic structure consisting of an aligned pair of English and Spanish elementary TAG trees. The alignment of each pair is given by the numbered boxes.

Chapter 3

Previous Work

Machine translation is a problem that has intrigued researchers for many years. The history of MT begins more or less in the 1950s, contemporaneously with the assembly of the first computers. We begin this chapter with a brief overview of that history. We devote the remainder of the chapter to a detailed discussion of two contemporary approaches to statistical MT: phrase-based systems and syntax-based systems.

3.1 A Brief History of Machine Translation

Ever since the era of the first digital computers, the notion of using them to perform translation has intrigued researchers. The earliest approaches to MT, in the 1950s, used dictionaries to look up words and hand-coded rules to place them in an order appropriate to the target language. This approach, when demonstrated on a limited-vocabulary, 250-word translation task with only six rules (known as the Georgetown-IBM experiment of 1954) was impressive enough at the time to initiate US government investment in the nascent field of computational linguistics.

Among the early MT pioneers was Yehoshua Bar-Hillel, a mathematician, linguist, and philosopher. In his article *The Present State of Research on Mechanical Translation* (Bar-Hillel, 1951), he proposed that MT systems ought to include steps involving automatic stemming and morphological analysis of the source-language words, automatic chunking of the source text into syntactic units or phrases, followed by rearrangement of the source-sentence chunks into a logically-equivalent target-language sentence.

By the 1980s and 1990s, many researchers were in fact working on large-scale systems

that performed such steps as Bar-Hillel suggested several decades earlier. The systems of that era consisted of large sets of hand-coded, inter-dependent rules; hence the name *rule-based*. Many of today's most successful commercial systems are rule-based.

In general, rule-based systems take either a *transfer* or *interlingua* approach to MT. The transfer approach performs automatic syntactic analysis of the input and transforms the source-language parse into a target-language parse via hand-coded transfer rules. Verbmobil is a well-known MT system that has its roots in transfer (Wahlster, 1993). The competing theoretical framework is the interlingual approach: the meaning of the source-language input is analyzed with respect to a language-independent interlingua, and the target language output is generated directly from the interlingual representation. The interlingua expresses syntactic, semantic, and pragmatic language universals. Dorr (Dorr, 1993) describes an interlingual approach.

The transfer approach has often been criticized for being overly labor-intensive: transfer rules have to be written for each language pair, in each direction (i.e., for German and English, one rule set for German-to-English translation and one for English-to-German translation). For translation involving n languages, the number of rule sets is $n(n-1)$. Using an interlingua reduces rule-set complexity to $O(n)$, where n is the number of target languages. This is because all source languages are mapped to a language-independent representation during analysis. However, the interlingua approach has often received criticism for being too abstract: actually defining a language-independent representation has proven to be a very difficult task.

In the mid-1990s, researchers at IBM proposed an alternative to rule-based systems with five statistical MT models of increasing representational power (Brown et al., 1990; Brown et al., 1993). The power of these statistical models was their ability to generalize from large amounts of data and to take into account many varied sources of information within a probabilistic framework. The gleaning of information from text itself turned out to be a viable alternative to writing complex sets of hand-crafted rules, particularly with the advent of phrase-based models in the early 2000s (e.g., (Koehn, 2004; Koehn et al., 2003; Och & Ney, 2002; Och & Ney, 2000)). The success of these systems has had a tremendous impact on the field of statistical MT.

Recently, and in large part as a reaction to some of the limitations of phrase-based systems (see Chapter 1), there has been increasing interest among researchers in directly

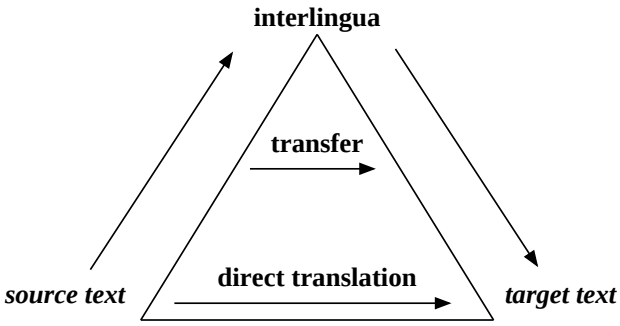


Figure 3-1: The MT pyramid.

modelling syntactic constraints within a statistical framework. Our discriminative, tree-to-tree model explicitly uses the syntactic information of both the source and target languages. Other approaches have proposed making use of syntax in the source or target languages alone.

Approaches to MT have traditionally been summarized by a pyramid such as the one in Figure 3-1 (from (Hutchins & Somers, 1992)). At the bottom of the pyramid is *direct translation*. Direct translation involves little to no intermediate analysis (e.g., syntactic or semantic). For example, a direct approach would be to translate the words in the source text using a bilingual dictionary and then reorder them according to lexicalized rules.

The transfer and interlingua approaches depend on an increasing amount of analysis of the source text and are shown higher up on the pyramid. Transfer involves a language-dependent analysis of the source at the syntactic and/or semantic level. The resulting structure is then rearranged to represent a structure conforming to the constraints of the target language. An interlingual representation, in contrast, should be language-independent and represent syntactic and semantic universals. Interlingual analyses of the source and target are supposed to result in the same representation, based on these universals. Translation therefore occurs immediately following the interlingual analysis.

The IBM models (Brown et al., 1990; Brown et al., 1993) are often described as direct translation models, and phrase-based methods are in large part direct models as well. Many of the statistical syntax-based models can be described as transfer approaches, particularly those that make use of syntactic analyses of both the source and target languages, such as our tree-to-tree method.

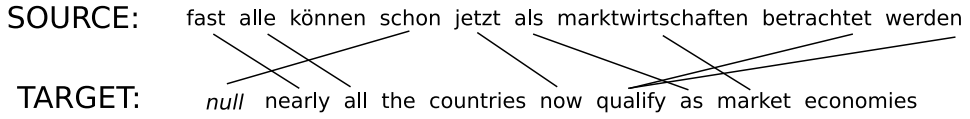


Figure 3-2: A sample IBM Model alignment.

3.2 Statistical MT: From Word-Based to Phrase-Based Models

The five statistical models proposed by researchers at IBM (Brown et al., 1990; Brown et al., 1993) in the early part of the decade had a strong influence on the direction of the field of MT. The IBM models induce alignments between the words in the source-language input and those in the target-language translation. These hidden-variable alignments are induced using a form of the expectation-maximization (EM) algorithm (Dempster et al., 1977) and help to explain the derivation of the translation. An example IBM model alignment is shown in Figure 3-2. Note that in this example, the German compound noun *markwirtschaften* would ideally be aligned with both *market* and *economies*. However, the IBM models constrain alignments to be many-to-one from from source words to target words, and so the model has chosen to align *markwirtschaften* to *market*. Another property of the IBM models is the inclusion of a special word, the *null* word, on the target-language side. The *null* word is used to account for words in the source language without any natural target-language correspondence.

The IBM models are commonly thought of as *word-based*, or word-for-word, in the sense that the translational unit is the word. That is, the translation process involves words in the source language being probabilistically transformed into words in the target language according to $P(f|e)$, the probability of a source-language word given the target-language word to which it is aligned.

Phrase-based systems (e.g., (Koehn, 2004; Koehn et al., 2003; Och & Ney, 2002; Och & Ney, 2000)) advanced the state of the art in statistical MT in around the late nineties. They surpassed the performance of the word-based IBM models in part by extending the notion of an *alignment* to allow phrasal translations. Phrase-based models start with alignments induced by the IBM Models and augment them to allow alignments between substrings of the source and target texts (see, for instance, (Koehn et al., 2003)). These aligned substrings are used in a bilingual phrase-pair dictionary during phrase-based search. The phrase pairs

do not have to be phrases in any linguistic sense.

Phrase-based models address a severe limitation of the IBM models by allowing many-to-many alignment mappings between source and target words. Phrase-based alignments may include such pairings as *marktwirtschaften* with *market economies*, *fast alle* with *nearly all*, and *als marktwirtschaften betrachtet werden* with *qualify as market economies*.

The use of many-to-many phrasal alignments means that these systems can capture some lexical reorderings as well as the translation of idiomatic expressions. For example, consider the English-Spanish phrasal translation pair:

SPANISH: la casa roja
GLOSS: the house red
REFERENCE: the red house

The positions of the English noun *house* and the adjective *red* are reversed relative to the positions of the Spanish noun *casa* and adjective *roja*. By learning that these phrases are common translations of one another, the system has a better chance of outputting the words in this phrase in the correct order. In contrast, the IBM models have to learn how to reorder all of the words in the sentence as distinct units.

While the introduction of phrasal alignments was certainly an improvement upon strict word-based alignments, one criticism of early phrase-based systems has been an over-dependence on lexical phrase pairs. For instance, as we have seen, Spanish nouns tend to precede adjectival dependents, while English nouns usually follow. The original phrase-based systems model this swapping by memorizing lexical phrasal correspondences. However, they have no mechanism for generalizing this phenomenon to novel input phrases. So if the phrase

SPANISH: la manzana roja
GLOSS: the apple red
REFERENCE: the red apple

appears during testing but not in the training data used to induce the phrase-pair dictionary, the system will be unable to use a phrasal translation. Even if the system has seen a similar phrase (say, *la casa roja* from above, it has no way of recognizing the similarity between these phrases.

Some work has been done to give phrase-based systems access to sources of linguistic information beyond lexical items (such as part-of-speech tags), and thereby endow them

with increased generalization capability. For instance, the work by (Liang et al., 2006) has shown some improvement in translation quality by using part-of-speech information in the distortion model. Also, some important work has been done with *factor-based models*, which behave much like phrase-based models while being able take into account additional information such as morphology, lemmas, and phrase bracketing in the source and/or target languages during decoding ((Koehn et al., 2007a; Koehn et al., 2007b)).

3.2.1 Phrase-Based Systems: The Underlying Probability Model

In this section, we review the design of a typical phrase-based system. The underlying probability model is log-linear. It models the conditional probability of a target-language sentence (T) given a source-language sentence (S):

$$Pr(T|S) = \frac{\exp \sum_i \lambda_i h_i(S,T)}{\sum_{T'} \exp \sum_i \lambda_i h_i(S,T')} \tag{3.1}$$

The λ_i are weights associated with the feature functions $h_i(S,T)$. The weights are usually trained using minimum error rate training (Och, 2003), although (Liang et al., 2006) found an online perceptron training algorithm to be effective as well. There are usually on the order of ten feature functions, among which may be

- bidirectional phrase models (models that score phrasal translations);
- bidirectional lexical models (models that look up pairs of words from phrasal translations in a translation dictionary);
- target-language models (often n-gram language models (Jelinek & Mercer, 1980) — nth order Markov models whose parameters are trained on large amounts of monolingual data; the probability of each word in a phrase is conditioned on the preceding $n-1$ words, and the probability of the phrase is the product of the probabilities the words);
- distortion models (models that consider how target-language phrases get reordered in the translation).

3.2.2 Phrase-Based Search

In general, phrase-based models use a search algorithm similar to the one used by Pharaoh (Koehn, 2004). Pharaoh relies on a variation of beam search to generate the best target-language translation given a sentence in the source language. The basic process is a loop that takes a partially-complete translation and

1. identifies all possible untranslated phrases in the source-language sentence;
2. for each source-language phrase, forms new partial translations by appending a target-language phrase in the dictionary to the end of the partially-complete translation;
3. scores the newly-formed partial translations;
4. places each partial translation in a stack containing other partial translations with the same number of already-translated source words.

At the start of the search process, the partial translation is the empty string.

Figure 3-3 shows phrase-based search in progress. The system is expanding a partial translation of the Spanish sentence *Sin embargo, no queda claro si el acuerdo llevará a progresos políticos en la región.* It has already produced translations of four phrases spanning nine words, and it now must choose some substring of the remaining Spanish text to translate and append to the partial translation. In the example, it chooses to translate the phrase *a progresos políticos.* One possible translation of this phrase, *to political progress*, is appended to the partial translation, forming a new partial translation. The new partial translation will be placed in a stack containing other partial translations that have covered twelve source words. The stacks are continually pruned so as to retain only the highest-scoring partial translations. The search continues until there are only complete translations (i.e., translations which have accounted for all of the words in the source sentence). See (Koehn, 2004) for a more detailed technical description of the search process used by Pharaoh.

An important characteristic of phrase-based search is that the feature functions (see Section 3.2.1) must decompose more or less along phrasal boundaries.[1] This is so that the feature functions may be used to score partial translations for pruning the stacks.

[1]The n-gram language model score can be derived using the previous target-language phrase because the translation is constructed monotonically.

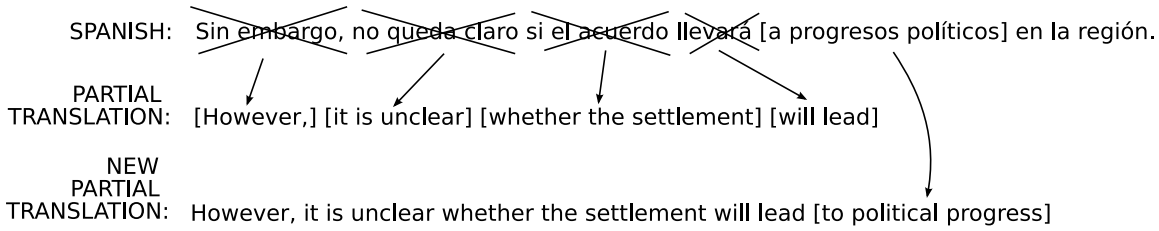


Figure 3-3: Phrase-based search.

This constraint on feature functions makes it difficult to evaluate global properties of a translation. For instance, it is difficult to evaluate a translation during decoding for certain syntactic properties such as whether each clause contains a verb, or whether the subject follows or precedes the verb, etc.

3.2.3 Strengths and Limitations of Phrase-Based Systems

From observing the output of phrase-based systems, in our view they are often quite good at modeling phrasal translations and conveying content words. However, they can also fail to produce syntactically-appropriate output, output that conveys the roles of these content words and their meaning in the original sentence.

Consider, for instance, the following English translation, generated by a phrase-based system trained using over 700,000 German-English sentence pairs from the European Parliament dataset (the baseline system described in (Collins et al., 2005)):

GERMAN: die berichte sollen den kandidaten helfen , notwendige reformen fortzusetzen und noch bestehende defizite abzubauen .

GLOSS: the reports should the candidates help , necessary reforms continue and still existing shortcomings to abolish.

REFERENCE: the reports are intended to help candidate countries continue with the necessary reforms and rectify persistent shortcomings .

PHRASE-BASED: the reports will help the candidate countries , the necessary reforms continue and residual reducing deficits .

Note, first of all, that the first segment of the phrase-based translation — *the reports will help the candidate countries* — is very coherent. Comparing this to the gloss, *the reports should the candidates help*, we can see quite clearly that the phrase-based system is capturing some important phrasal reorderings: the verb *help* has been repositioned so that it sits between the auxiliary verb *will* and the object *the candidate countries*. Bear in mind that the phrase-based system achieves this reordering with no explicit knowledge of source-language or target-language syntax.

In the remainder of the sentence, the phrase-based system outputs all of the important content words: *necessary, reforms, continue, residual, reducing,* and *deficits*. However, their roles in the sentence are not clear. For instance, it is not clear that the phrase *the necessary reforms* is the object of the verb *continue*. Nor is it clear what role the segment *and residual reducing deficits* has to the remainder of the sentence. As a matter of fact, this segment could be translated as a clause (something like *and reduce residual deficits*), but the word *reducing* is behaving more like an adjective modifying *deficits* than the main verb of a clause.

A few additional examples of phrase-based output, taken from the test set described in (Collins et al., 2005), illustrate some of the syntactic errors this system tends to make when translating from German to English. We present two examples in detail here before making some general observations.

Example 1

GERMAN: damit die umsetzung der menschenrechte jedoch auch praktisch überwacht werden kann , müssen der union eindeutige regelungen über mögliche sanktionsmechanismen gegenüber den mitgliedstaaten zur verfügung stehen .

GLOSS: for the implementation of human rights however also practical monitored be can , must the union clear rules regarding possible sanction mechanisms against the member states to be used .

REFERENCE: for the implementation of human rights to be monitored in practice also , however, the union must have clear rules regarding possible sanction mechanisms to be used against member states .

PHRASE-BASED: thus the implementation of human rights practical can be monitored , the union must have clear rules on possible penalties against the member states are available .

The German sentence in this example contains two clauses: *damit die umsetzung...* and *müssen der union....* In the original text, the first clause is a subordinate clause that modifies the second, independent clause. In the phrase-based translation, however, this relationship has been lost, and both clauses read more like independent clauses. Furthermore, in the first clause, the phrase-based system has introduced the adjective *practical*, a translation of *praktisch*, and placed it before the verb phrase (*can be monitored*). In the reference translation, this same word has been translated as the adverbial *in practice*, modifying the verb *monitored*. In the second clause, the system has introduced two finite verb phrases: (*must have* and *are*). This mishandling of verb phrases and their modifiers seems to be a common error of phrase-based systems. Phrase-based systems do not overtly model the appropriate handling of such phenomena.

Example 2

GERMAN: herr präsident , meine sehr verehrten kolleginnen und kollegen ! unter dem beifall der gerichtshöfe in luxemburg und straßburg hat der konvent einen entwurf einer grundrechtecharta in bemerkenswert ausgewogener weise formuliert und in die hände des rates gelegt .

GLOSS: mr president , my very honored ladies and gentlemen ! with the approval of the courts in luxembourg and strasbourg has the convention a draft of a charter of fundamental rights in a remarkably balanced way formulated and in the hands of the council delivered .

REFERENCE: mr president , ladies and gentlemen, with the approval of the courts in luxembourg and strasbourg, the convention has put together a draft charter of fundamental rights which is remarkably balanced , and has delivered it into the hands of the council .

PHRASE-BASED: mr president , ladies and gentlemen , under the applause of the courts in luxembourg and strasbourg , the convention has a draft a charter of fundamental rights in a balanced way remarkable and formulated in the hands of the council laid .

This example contains two clauses: *unter dem beifall...* and *und in die hände....* The phrase-based system introduces the conjunction *and* before *formulated*, the main verb, thereby isolating it from the first clause. This is another example of a poorly-constructed verb phrase. In the second clause, the prepositional phrase *in the hands of the council* appears before the verb it modifies (*laid*), which is unusual in English. This clause is also a missing an object, in this case the referring pronoun (*it*). I.e., the final part of the sentence should say something like *...laid it in the hands of the council*. These are examples of misplaced or missing verbal arguments and modifiers.

Missing arguments and poorly-constructed verb phrases are problems that are likely to arise in phrase-based systems due to their indifference to syntax. In examples 2, 4, and 5 (examples 3, 4, and 5 can be found in the table in Figure 3-4), the phrase-based system has produced clauses that are missing verbal arguments. Either the system has dropped them from the original text (example 4) or failed to produce an argument for a verb that requires one (examples 2 and 5). Examples 1, 4, and 5 demonstrate what can go wrong with verb phrases in a phrase-based system: in example 1, there are two finite verbs in the clause; example 4 contains two ungrammatical verb phrases; example 5 is missing a verb.

As we saw in the first example, the misplacement of modifiers and arguments with respect to other elements in the sentence can also be a problem. For instance, the misplacement of verbs with respect to their modifiers, as in examples 1, 2, and 3 (see Figure 3-4 for examples 3, 4, and 5), is a very common error for German-to-English translation. Arguments such as subjects and objects may also need to be repositioned relative to the source-language input in order to produce syntactically-sound output (see, for instance, examples 2 and 3).

Phrase-based systems generally do not model the syntactically-correct construction of clauses and verb phrases; nor do they model the argument structure of verbs. As we saw during the discussion of phrase-based search in Section 3.2.2, it would be difficult to incorporate features to sensitize the model to these phenomena: it would require analysis that could cross phrasal boundaries, and phrase-based search generally uses features that decompose along phrasal (non-syntactic) boundaries. Our AEP approach directly addresses the construction of satisfactory verb phrases by explicitly modeling clausal main verbs and their accompanying modals and auxiliaries.

3.3 Syntax-Based Statistical Models

The challenge of incorporating syntactic information in a statistical framework has been of increasing interest to many researchers in the past several years. The tree-to-tree discriminative model we present in this thesis is one possible approach to statistical syntax-based translation. In this section we present a review of several representative papers in the field to offer a sense of the range of possibilities.

3.3.1 Some Differences Among Syntax-Based Systems

The body of work we present resists a definitive taxonomy. Rather, there appear to be a few broad dimensions along which the approaches tend to differ. We describe two of these dimensions here before presenting the literature review, arranged simply in chronological order.

Approaches to Reordering A concern of general interest to many researchers working on MT is the reordering problem: how to find a syntactically-satisfying order for the words in the target-language translation. In syntax-based statistical models in particular, there may be structural information available to the system to help it make appropriate decisions. Researchers developing such models have devised different ways of leveraging this information. For instance, restructuring approaches such as (Collins et al., 2005; Xia & McCord, 2004; Nießen & Ney, 2004) make use of linguistic information (syntactic and/or morphological) to restructure the source-language sentence in some way to make reordering easier. The approach of (Yamada & Knight, 2001) implements reordering by modeling permuta-

EXAMPLE 3

GR: ich gehe mit ihm konform , dass es notwendig ist , das institutionelle gleichgewicht durch eine politische aufwertung sowohl der kommission als auch des rates aufrechtzuerhalten .

GL: i go with him conform , that it necessary is , the institutional balance by a political revaluation both of the commission as also the council to maintain .

R: i agree with him on the need to maintain the institutional balance and enhance the role , in political terms , of both the commission and the council .

P: i am with him that it is necessary , the institutional balance by means of a political revaluation of both the commission and the council to maintain .

EXAMPLE 4

GR: unabhängig davon möchte ich sie jedoch bitten , diese zukunft nicht schon für morgen zu erwarten , denn sie wird schrittweise , stein für stein , errichtet .

GL: independent thereof would like i you however ask , this future not yet for tomorrow to expect , because it will be gradually , stone for stone , constructed .

R: that is why i invite you , in spite of everything , not to throw yourselves headlong into the future , as this is a future that should be built step by step , and stone by stone .

P: irrespective of this , however , i should like to ask for this future is not already expect tomorrow , because they will gradually , stone for stone , erected .

EXAMPLE 5

GR: fünftens scheint uns auch die haushaltsvorausschau nicht angemessen zu sein .

GL: fifthly feel we also the budgetary forecasts not adequate are .

R: fifthly , we also feel that the budgetary forecasts are not sufficient .

P: fifthly , seems to us the budget perspectives not appropriate .

Figure 3-4: More examples of phrase-based output (P). R is a human-generated translation, GL is a word-for-word translation, and GR is the German input.

tions of children nodes in parse trees. Several approaches (e.g., (Chiang, 2005; Wu, 1997)) use synchronous grammar formalisms to model reordering. Some approaches involving dependency structures may use heuristics (Lin, 2004) or separate reordering models (Menezes & Quirk, 2007) to flatten and linearize target-language dependency structures. (Riezler & Maxwell, 2006) use a grammar-based generator to flatten dependency structures, then apply language and reordering models to generate a translation. (Alshawi et al., 2000) construct transducers that process and linearize automatically-induced dependency structures. Some tree-to-string models using constituent trees (e.g., (Galley et al., 2006; Huang et al., 2006)) embed reordering information in tree-to-string transfer rules. (Marcu et al., 2006) use target-language syntax over phrases that may give their model more detailed information when it comes to their reordering. The model of (Chiang, 2005) induces CFG-like rules from an unannotated parallel corpus, allowing for variables injected in lexical phrases that may help with reordering.

The Use of Syntax An important distinction between the various syntax-based models is whether syntax is used on the source side only, the target side only, or on both sides. We call these approaches *tree-to-string*, *string-to-tree*, and *tree-to-tree*, respectively. Among tree-to-string models, for instance, are restructuring models (Collins et al., 2005; Xia & McCord, 2004; Nießen & Ney, 2004) that use source-side syntax to rearrange the input to try to simplify the translation task. The dependency models of (Menezes & Quirk, 2007) and (Lin, 2004) use source-side dependency parses to induce target-language dependency structures. Alternatively, many systems make use of target-side syntax. Among these are (Marcu et al., 2006; Galley et al., 2006; Yamada & Knight, 2001). In general, there has been a good deal of literature on tree-to-string theory and methods, for example (Huang et al., 2006; Galley et al., 2006; Graehl & Knight, 2004; Gildea, 2003).

There is also a some literature on tree-to-tree models. With the steady improvement in performance of parsers of various languages, tree-to-tree systems have arguably become more feasible. (Nesson & Shieber, 2007; Melamed, 2004; Graehl & Knight, 2004; Eisner, 2003) offer some important theoretical results involving tree-to-tree algorithms. (Ding & Palmer, 2005) present a system that maps dependency trees in the source language to dependency trees in the target language. The work of (Gildea, 2003) also involves methods that make use of syntactic information in both the source and target languages. (Riezler

& Maxwell, 2006) employ LFG grammars of both the source and target languages, and (Nesson et al., 2006) present a method using synchronous tree-insertion grammars.

Finally, there are some methods that use syntactic formalisms such as synchronous context-free grammars (CFG) without committing to any particular grammar. Chiang (Chiang, 2005) calls these systems *formally syntactic* as opposed to *linguistically syntactic*. Among such approaches are (Chiang, 2005) and (Wu, 1997), both of which describe systems that learn a grammar that can simultaneously generate two CFG trees.

3.3.2 Literature Review

(Wu, 1997)

Wu describes early work on a formalism called *inversion transduction grammar* that can be used to parse two languages synchronously with a very simple inversion ordering rule. The formalism consists of context-free grammar rules that simultaneously generate both nonterminals (common to both languages) and terminal symbols. The child nodes of any left-hand side nonterminal may be generated in inverted order (left-to-right in one language and right-to-left in the other). Wu presents an EM-style (Dempster et al., 1977) algorithm for carrying out the synchronous parsing.

(Alshawi et al., 2000; Alshawi, 1996)

The work of (Alshawi et al., 2000) describes methods for building a translation model from automatically-learned dependency head transducers. The model takes as input an unannotated parallel corpus and, using a dynamic programming algorithm, learns dependency structures over the translation string pairs. The dependency structures allow for a synchronous, hierarchical alignment between the two languages. Head transducers are then constructed using the induced dependency structures.

(Yamada & Knight, 2001; Yamada & Knight, 2002; Charniak et al., 2001)

Yamada and Knight describe methods that make use of syntactic information in the target language alone. Working within a noisy-channel framework, they present a model for $P(f|e)$, the probability of a source-language string given a target-language parse tree. The nodes of the input tree are probabilistically subjected to three transformative operations: a reordering operation, an insertion operation, and a translation operation. The source-language string is then read off of the leaves of the transformed tree. To train their model,

Yamada and Knight use a variant of EM. A human evaluation found word alignments induced by their syntax-based model to be of higher quality than those induced by IBM Model 5. In (Yamada & Knight, 2002), the authors present a decoding algorithm for their model. They also extend the earlier model to handle phrasal translations. In Charniak et al., the translation model of (Yamada & Knight, 2001) is combined in a noisy-channel framework with an English language model based on the syntactic parser of (Charniak, 2000).

(Fox, 2002)

Fox carries out a study on phrasal cohesion between French and English. Using a manually-aligned corpus with automatically-generated parses of the English sentences, Fox examines the number of head and modifier crossings between the two languages. While the numbers differ dramatically depending on the experimental setup, the author concludes that there is overall less cohesion between French and English than one might expect. Much of the difference appears to be due to three identifiable structural divergences between the languages, while a good portion is due to errors, rewordings, and reorderings in the corpus. The number of crossings is smaller when dealing with dependency structures instead of constituent structures.

(Eisner, 2003)

Eisner discusses methods for learning non-isomorphic tree mappings in the context of the synchronous tree-substitution grammar (TSG) formalism.[2] He presents algorithms for training, decoding, and deriving the single best syntactic alignment between two trees. The methods are based on an EM-type algorithm (Dempster et al., 1977). Eisner mentions that the methods described were implemented in a Czech-to-English translation system, although no results are reported.

(Gildea, 2003)

Gildea develops tree-to-string and tree-to-tree models for machine translation that specifically address non-isomorphic alignments. Like (Ding & Palmer, 2005; Eisner, 2003; Fox, 2002), Gildea makes the point that for many language pairs, syntactic divergences may be quite common, either because the languages truly differ syntactically or because the corpus translations are sufficiently different. Gildea addresses the problem of non-isomorphism

[2]TSG is like tree-adjoining grammar, TAG, except that it uses only the substitution operation, not the adjunction operation.

between tree structures by introducing a *clone* operation that allows for the copying of a subtree in position α to a different position β. Cloning, combined with a deletion operation, permits more dramatic restructuring of a parse tree than a model that permits solely insertion, reordering of child nodes, and translation (e.g., (Yamada & Knight, 2001)). Gildea tests the alignments generated by his tree-to-string and tree-to-tree models in a Korean-to-English translation task. His models show an improvement (in terms of alignment error rate) over IBM Models 1, 2, and 3, and other syntax-based statistical systems trained without the clone operation.

(Galley et al., 2004)

Galley et al. aim to test the complexity of transformation rules necessary to carry out automatic translation. The authors describe methods for the extraction of tree-based rules. Their methods depend on the availability of a target-language parser and a word-aligned parallel corpus. They conclude that rules that perform only local child reorderings are not adequate for producing high-quality translations.

(Graehl & Knight, 2004)

Graehl and Knight describe methods for training tree transducers. A tree transducer is a generalization of a finite-state transducer: a finite-state transducer is restricted to transformations on strings, while a tree transducer is able to compute transformations on trees. The paper focuses on an algorithm for training a tree transducer, i.e., for assigning weights to each of the productions. The algorithm is a generalization of the forward-backward algorithm for training a finite-state transducer. A tree transduction framework can be used as a model for a tree-to-tree or tree-to-string machine translation system. For example, Graehl and Knight describe a set of tree-transducer productions to implement the tree-to-string MT system of (Yamada & Knight, 2001).

(Lin, 2004)

Lin presents a method for translation using dependency trees in a probabilistic framework that models $P(T|S)$, the probability of a target-language tree given a source-language tree. The method involves inducing a set of transfer rules that are used to produce translations. The transfer rules express a mapping from source-language to target-language dependency trees. Lin presents an algorithm for extracting transfer rules from a parallel corpus of parsed source-language sentences and unannotated target-language sentences. He also presents a

method for decoding and formulates it as a graph-theoretic problem he calls *Minimum Path Covering of Trees*.

(Melamed, 2004)

Melamed establishes a theoretical framework for generalized synchronous parsing and translation using multitext grammars (a generalization of context-free grammars to the synchronous case). He describes an abstract design for a full, n-dimensional translation system starting with nothing other than some parallel text and at least one monolingual treebank. With these resources, the system induces a parallel treebank; from the parallel treebank, it induces a probabilistic multitree grammar; the grammar enables the system to decode new inputs and produce translations.

(Nießen & Ney, 2004; Nießen & Ney, 2001)

Nießen and Ney investigate two reordering rules for translation between English and German. The rules address the phenomena of (1) German separable verb prefixes,[3] and (2) interrogative word order.[4] To deal with German separable verb prefixes, the authors preprocess their training data to identify the prefixes and prepend them to the verb to which they belong (for German-to-English translation). When translating in the opposite direction, they train their MT model to attach the prefix to the verb and then use a separate postprocessor to find the most likely placement. For interrogatives, they preprocess the training data to normalize their form such that they more closely resemble declaratives. Postprocessing is necessary to map the normalized interrogatives to their correct form in the target language. Both of these techniques generate improved scores according to four different scoring metrics compared to a baseline system that does not use pre-translation reordering.

(Och et al., 2004)

The work by Och et al. incorporates syntactic information through reranking approaches applied to n-best output from phrase-based systems. Several features are defined, including simple word-level feature functions (e.g., IBM Model 1 score), shallow syntactic feature functions (e.g., probabilities derived from part-of-speech sequences), and tree-based feature functions (including one feature that uses the score derived from the tree-to-string model

[3] In German, certain verbs contain prefixes that are removable and may be placed elsewhere in the sentence

[4] Interrogatives in both English and German have different word order than declaratives.

described in (Yamada & Knight, 2001) and (Yamada & Knight, 2002)). Of all the features described, the one that has the largest impact on BLEU score is the IBM Model 1 score.

(Xia & McCord, 2004)

Xia and McCord present a method for automatically learning rewrite rules that reorder a source-language parse tree such that it more closely resembles the word order of the target language. This work is very similar to the approach by (Collins et al., 2005), except that in that work the six rules are specified by hand, whereas in this work, over 3 million rewrite patterns are automatically generated (only 56K are retained for use after filtering, and only 1.4 patterns per sentence are actually employed, on average). The experimental results show more noted improvement on out-of-domain test data than in-domain.

(Chiang, 2005)

Chiang shows significant improvements in translation accuracy using learned hierarchical phrases in a synchronous probabilistic context-free grammar. The technique he describes involves inducing PCFG-like rules from a parallel corpus devoid of any *a priori* syntactic annotations. Chiang uses a feature-based log-linear model to determine the weights on the extracted rules, and a form of beam-search to implement decoding. Chiang's model allows for the learning of templates involving a mixture of lexical items and phrasal placeholders, thereby facilitating the reordering problem.

(Collins et al., 2005)

Collins et al. describe an approach that involves parsing and reordering the input prior to translating it. In their experiments, they work with translation from German to English, two languages that differ sufficiently in word order to make the results meaningful. The reordering transformations that are applied to the parsed input are intended to alter the input such that it more closely resembles the word order of the target language. The six transformations they apply are hand-written. The results show an improvement over a standard phrase-based system both in terms of BLEU score and in terms of a human evaluation.

(Ding & Palmer, 2005)

Ding and Palmer describe a method for producing target-language translations via a tree-to-tree transduction method using dependency trees. They induce a grammar from an aligned,

parsed parallel corpus (Chinese, English); the grammar relates subtrees of the dependency trees (*elementary trees*) to one another in an isomorphic manner. To decode a sentence, it is first parsed into a dependency tree and then transduced into a target-language dependency via an HMM-like probabilistic transduction process that uses the induced grammar. The ordering of related elementary trees is kept constant through the transduction from, in this case, Chinese to English, both SOV languages.

(Galley et al., 2006)

Working within a noisy-channel framework, the authors define a probability distribution for $Pr(f|\pi)$, the probability of a source-language string f given an target-language tree π. Like (Galley et al., 2004), this model defines this probability in terms of a derivation dependent on the extraction of tree-based rules. Unlike the earlier work, however, this model's extracted rules are larger in scope and more complex, conditioning on more syntactic context. The reported BLEU scores (on both Arabic-to-English and Chinese-to-English) are around six points lower than a state-of-the-art alignment template model. However, they represent a 3.63 BLEU point increase over the (Galley et al., 2004) model.

(Huang et al., 2006)

The authors make use of parse trees on the source-language side in an English-to-Chinese translation system. The parsing is done in an independent step, and then a derivation from the English parse tree to a Chinese string is computed. The left-hand sides of the rules (used to form the derivations) are allowed to span multiple levels in the tree; in other words, they exhibit the TAG property of *extended domain of locality*. The tree-to-string probability model is combined with a Chinese language model and a sentence-length model in a log-linear framework. When compared with a phrase-based model, the Huang et al. model scores about three BLEU points higher.

(Marcu et al., 2006)

Marcu et al. make use of target-side syntax to augment source-target phrase pairs. The syntactic information is used in the extraction of tree-to-string rules that are used to form derivations of ⟨source-string,target-tree,alignment⟩ triples. Probabilities on these rules are estimated using frequency counts. The syntax-augmented models (the authors define five variations) are employed in the context of a log-linear framework. All of the models outperform a phrase-based system in Chinese-to-English translation, both according to BLEU

score and in a human evaluation.

(Nesson et al., 2006)

The work by Nesson et al. is representative of a larger body of work that has been conducted by Shieber, Nesson, and Rush in the past few years (Shieber, 2007; Nesson & Shieber, 2007; Nesson & Shieber, 2006) with the larger goal of exploring the use of synchronous grammars for machine translation. (Nesson & Shieber, 2006) and (Nesson & Shieber, 2007) both investigate the use of synchronous tree-adjoining grammar, in particular, as a formalism with sufficient expressivity, efficiency, and simplicity to meet the demands of the translation task. In (Nesson et al., 2006) the authors implement an MT system based on probabilistic tree-insertion grammar[5] and trained on a small amount of data (around 15K sentences). The resulting system outperforms a phrase-based system, when trained using the same data, in terms of BLEU score and in human evaluations for fluency and adequacy.

(Quirk & Corston-Oliver, 2006)

Quirk and Corston-Oliver investigate the impact of parse quality on the output of a statistical syntax-based MT system. The system they test on is the dependency treelet system described in (Menezes & Quirk, 2007). The experimental method involves using varying amounts of training data to produce English parsers of varying quality. These parsers were then used in the context of the syntax-based system. For both English-to-Japanese and English-to-German translation, the MT systems trained with better-quality parsers produced higher-quality translations.

(Riezler & Maxwell, 2006)

Riezler and Maxwell describe a method for learning a probabilistic model that maps LFG parse structures in German into LFG parse structures in English. The method uses LFG parsers along with word alignment information to extract aligned LFG transfer rules. A chart-based algorithm is used to generate candidate translations, and a translation is selected using a beam search over the transfer chart. Features are computed over the candidate translations, with weights on the features being trained using minimum error-rate training. Riezler and Maxwell's system performs comparably to a standard phrase-based system using the NIST metric for evaluation when the test set is restricted to in-coverage sentences

[5]Tree-insertion grammar (Schabes & Waters, 1995) is a tree-based formalism similar to TAG with both substitution and adjunction. Restrictions on adjunction and the form of elementary trees allows for cubic-time monolingual parsing.

(roughly, sentences that parse completely), but performs comparably to IBM Model 4 with an unrestricted test set. In a human evaluation, their system was found to improve on in-coverage sentences in terms of both translational adequacy and grammaticality.

(Menezes & Quirk, 2007; Quirk et al., 2005)

The methods of Menezes and Quirk build on an earlier dependency treelet model (Quirk et al., 2005) that makes use of dependency parses of the source language. In the treelet model, source-side parses are projected onto the target-language translation via a set of word alignments. Dependency treelets are then extracted from the aligned, parallel dependency-tree corpus. Several features including an order model are trained and combined using a log-linear framework. The best results for the system (translating from English into French) are obtained using a bottom-up exhaustive search method; however, the search space has to be pruned substantially in order for the method to work. In the newer work, Menezes and Quirk take advantage of the fact that reordering is decoupled from lexical choice in the treelet model and focus on improving the reordering model so as to reduce the size of the search space in a more informed manner. While the results show only a slight improvement (in terms of BLEU score) over the original treelet method, the authors point out that the new method is a lot more time-efficient.

(Wang et al., 2007)

This work suggests that restructuring the trees in a treebank used in syntax-based MT may lead to improved translation quality. The authors present several methods for binarizing trees, including a method that induces binarizations using the EM algorithm (Dempster et al., 1977). These methods are tested in the context of a syntax-based system based on the work described in (Zhang et al., 2006) and (Galley et al., 2006). The system performs translation from Chinese to English and yields a BLEU score of 37.94, the highest score on this particular test set to date.

Chapter 4

A Discriminative Model for Statistical Parsing

In this chapter, we present a syntactic parser for Spanish that could easily be used for AEP-based MT. The parser uses a linear structured prediction model to rerank parses in an n-best list generated by a PCFG-based first-pass model.[1] The linear model allows us to incorporate arbitrary global features of parse trees in the second pass to select the best parse among the candidates. The reranking model reaches 85.1% F1 accuracy on the Spanish parsing task.

The first-pass parser is a lexicalized PCFG model that incorporates features based on Spanish morphology. The morphologically-informed first-pass parser itself achieves an F1 constituency score of 83.6%, an absolute improvement of 1.4% over a baseline which makes little use of morphology.

Throughout this chapter, we refer to the morphological model as Model M and the reranking model as Model R. This chapter is based on work originally described in (Cowan & Collins, 2005).

4.1 Introduction

A clear application of statistical parsing is to tree-to-tree MT approaches. However, early methods for statistical parsing were mainly developed through experimentation on English data sets. While subsequent research has focused on applying these methods to other lan-

[1] See Chapter 2 for an explanation of linear structured prediction models, reranking, and PCFGs.

guages, there is often a gap between the performance of English parsers and non-English parsers. There has been widespread evidence that new languages exhibit linguistic phenomena that pose considerable challenges to techniques originally developed for English; because of this, an important area of current research concerns how to model these phenomena more accurately within statistical approaches. In this chapter, we investigate this question within the context of parsing Spanish. We develop two models that incorporate detailed features in a Spanish parser. The baseline model, which also forms the underlying model for our Model M parser, is a lexicalized PCFG originally developed for English.

We incorporate morphology into the PCFG model (the Model 1 parser in (Collins, 1999)) by modifying its part-of-speech (POS) tagset; in this chapter, we explain how this mechanism allows the parser to better capture syntactic constraints.

All of the experiments in this paper have been carried out using a freely-available Spanish treebank produced by the 3LB project (Navarro et al., 2003). The version of the treebank used in this chapter contains around 3,500 hand-annotated trees. Each tree encodes a large amount of morphological information. One of the research questions posed by this chapter is to what extent we can make use of all this information: does it help to use as much information as we can? We expect there to be diminishing returns since adding features adds complexity to the model, complexity that it may not be able to handle given a limited amount of training data. If this is the case, then is there a subset of morphological features with strong predictive power? If so, why is this subset more predictive than other subsets?

To get at these questions, we use development data to test the performance of several first-pass Model M parsers, each incorporating a subset of morphological information. We have found that those models sensitive to less morphology do exhibit better parsing performance than those sensitive to more. The highest-accuracy model on the development set is sensitive to the mode and number of verbs, as well as the number of adjectives, determiners, nouns, and pronouns. On test data, this model obtains an F1 accuracy of 83.6%/83.9%/79.4% for labeled constituents, unlabeled dependencies, and labeled dependencies, respectively. The baseline model, which makes almost no use of morphology, achieves 81.2%/82.5%/77.0% in these same measures.

In a set of reranking experiments, we use the highest-performing Model M parser as the first-pass parser. Here we investigate the efficacy of a reranking approach for parsing Spanish by using arbitrary structural features in a second pass. Previous work in statistical

parsing (Collins & Koo, 2005) has shown that applying reranking techniques to the n-best output of a base parser can improve parsing performance. Applying the exponentiated gradient reranking algorithm (Bartlett et al., 2004) to a linear model trained using the n-best output of our morphologically-informed Model M gives us similar improvements. This reranking model – Model R – performs at 85.1%/84.7%/80.2% F1 accuracy for labeled constituents, unlabeled dependencies, and labeled dependencies.

4.2 Background: Spanish Morphology

Model M makes use of Spanish morphology to improve the performance of a baseline PCFG model. Relative to English, Spanish has a rich set of morphological characteristics. Figure 4-1 illustrates how the forms of Spanish nouns, determiners, and adjectives reflect both number and gender. Spanish pronouns also reflect gender, number, person, and case, and Spanish verbs inflect for number and person. Figure 4-3 gives a more global picture of Spanish morphological features for various parts of speech. In general, even when both Spanish and English exhibit the same morphological phenomenon, Spanish is usually far more morphologically complex. For instance, English verbs usually only have two present-tense forms, one for the 3rd-person singular (e.g., *he/she/it eats*), and one for everything else *I/you/we/they eat*); Spanish, on the other hand, generally has six distinct inflected forms for the present tense alone, one for each possible pairing of number (singular or plural) and person (1st, 2nd, or 3rd, see Figure 4-2).

The morphological features used in Figure 4-3 are explained below:

- Gender: Grammatical gender signifies noun classes, which in Spanish are *masculine* or *feminine*; in rare cases, Spanish gender can also be *neutral*. Use of gender in English is fairly limited (e.g., words like *actor* and *actress* have a gender associated with them), whereas in Spanish, nearly every noun has a gender.

- Number: Grammatical number in general refers to count distinctions, which in Spanish (like English) is the difference between *singular* or *plural*. Spanish inflects determiners, nouns, pronouns, adjectives, and verbs for number, whereas English only does so for nouns, pronouns, and, to a very limited extent, verbs.

- Person: Grammatical person distinguishes between different types of referents in a

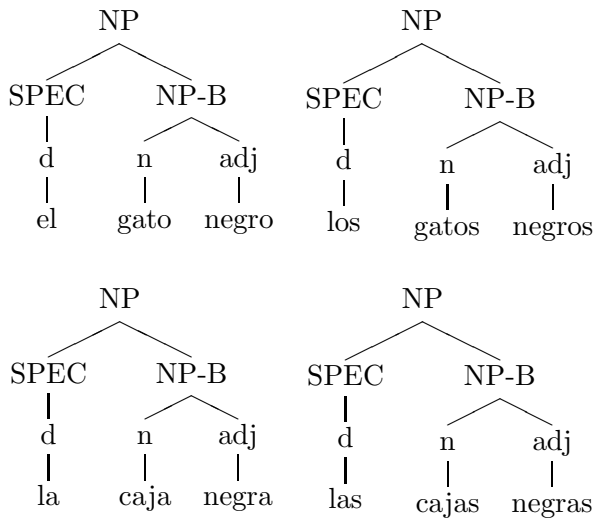


Figure 4-1: Morphological agreement in Spanish is sometimes constrained by syntax. The tree in the upper left (*el gato negro*, or *the black cat*) shows how the number (singular) and gender (masculine) of the determiner and adjective are constrained to agree with the noun. Likewise for the tree in the upper right, headed by a plural masculine noun (*gatos*, or *cats*): here the determiner and noun both take the plural, masculine form. The trees on the bottom illustrate the same concept for a head noun that is feminine (*caja*, or *box*).

sentence; in Spanish (as in English), these different types are *first person* (i.e., *I, we*), *second person* (i.e., *you*), or *third person* (i.e., *she, he, it*). Verbal inflection for person is far more extensive in Spanish than in English.

- Mode: Grammatical mode or mood describes the relationship of a verb with reality or intent; Spanish is usually said to have two modes: *indicative* and *subjunctive*. While English could also be said to distinguish these modes, use of the subjunctive is more frequent in Spanish than in English.

- Case: The grammatical case of a noun or pronoun indicates its function in a phrase or sentence. In Spanish, pronomial case distinctions are made for the nominative, accusative, and dative cases. This is also true for English, although there are more Spanish pronomial forms than there are English.

Spanish morphology is often constrained by syntax, a fact which may be exploited in automatic syntactic parsing. For instance, any constituent noun, adjective, and determiner in a noun phrase is constrained to agree in number and gender with the head noun (see Figure 4-1); a verb is constrained to agree in number and person with its subject (see Figure 4-2). This means that morphology offers important structural clues about the syntactic

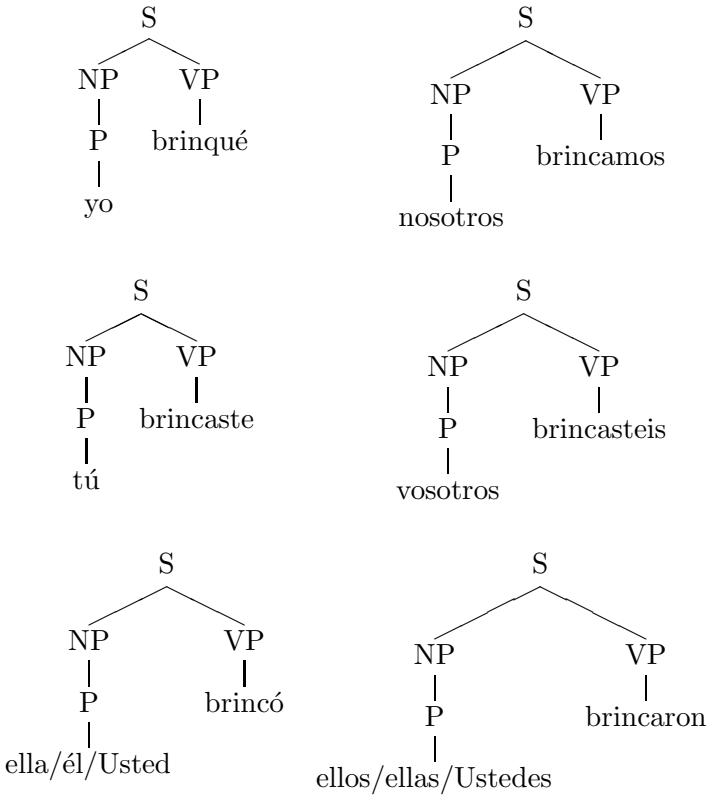


Figure 4-2: Spanish verb forms are inflected to agree with the person and number of their subject. In this figure, six preterite (past tense) verb forms are shown for each of first, second, and third person, singular and plural.

CATEGORY	**gender**	**number**	**person**	**mode**	**case**
noun	√	√			
verb		√	√	√	
determiner	√	√			
adjective	√	√			
pronoun	√	√	√		√

Figure 4-3: Spanish morphological features according to part-of-speech category.

dependencies in a Spanish sentence.

4.3 Models

This section details our two approaches for adding features to a baseline parsing model. First, we describe how morphological information can be added to a parsing model by modifying the POS tagset. Second, we describe an approach that reranks the n-best output of the morphologically-rich parser, using arbitrary, general features of the parse trees as additional information.

4.3.1 Model M: Adding Morphological Information

We incorporate morphological information into our Model M parser by modifying the POS tagset of a lexicalized PCFG[2] — the Model 1 parser described in (Collins, 1999) (hereafter Model 1).

As we mentioned in Section 4.1, the trees in the 3LB corpus contain a lot of annotated morphological information. One of the questions we aim to address in this chapter is to what extent we can make use of this information. In order to answer to this question, we created different models corresponding to distinct POS tagsets, each one encoding a subset of the information available to us. We will evaluate the performance of each of these parsers to test the efficacy of each subset.

The table in Figure 4-4 shows the complete subset of morphological features we considered when forming Spanish POS tagsets.[3] A parser trained using a certain tagset will be sensitive to the morphological features it encodes. There are 22 morphological features in total in this table; different POS tagsets can be created by deciding whether or not to include each of these 22 features, meaning that there are 2^{22} different tagsets we could have experimented with. For instance, one particular tagset might encode the modal information of verbs. This tagset would contain six POS tags for verbs (one for each mode – indicative, subjunctive, imperative, infinitive, gerund, and participle) instead of just one. A tagset that encodes both the number and mode of verbs would have 18 verbal POS tags, assuming three values (singular, plural, and neutral) for the number feature.

[2]Hand-crafted head rules are used to lexicalize the trees (see Appendix A).

[3]In other words, this is the union over all the subsets of features we experimented with. There are some features included in the treebank that we never used; they are not included in the table.

Category	Attributes	Values
Adjective	gender	masculine, feminine, common
	number	singular, plural, invariable
	participle	yes, no
Determiner	gender	masculine, feminine, common, neutral
	number	singular, plural, invariable
	person	first, second, third
	possessor	singular, plural
Noun	gender	masculine, feminine, common
	number	singular, plural, invariable
Verb	gender	masculine, feminine
	number	singular, plural
	person	first, second, third
	mode	indicative, subjunctive, imperative, infinitive, gerund, participle
	tense	present, imperfect, future, conditional,past
Preposition	gender	masculine, feminine
	number	singular, plural
	form	simple, complex
Pronoun	gender	masculine, feminine, common, neutral
	number	singular, plural, invariable
	person	first, second, third
	case	nominative, accusative, dative, oblique
	possessor	singular, plural

Figure 4-4: A list of the morphological features we used to create our models. For succinctness, we only list attributes with at least two values. See (Torruella, 2000) for a comprehensive list of the morphological attributes included in the Spanish treebank.

The Effect of the Tagset on Collins' Model 1

Modifying the POS tagset allows Model 1 to better distinguish events that are unlikely from those that are likely, on the basis of morphological evidence. An example will help to illustrate this point.

Model 1 relies on statistics conditioned on lexical headwords for practically all parameters in the model. This sensitivity to headwords is achieved by propagating lexical heads and POS tags to the non-terminals in the parse tree. Thus, any statistic based on headwords may also be sensitive to the associated POS tag. For instance, consider the subtree in Figure 4-5. Note that this structure is ungrammatical because the subject, *gatos* (*cats*), is plural, but the verb, *corrió* (*ran*), is singular. In Model 1, the probability of generating

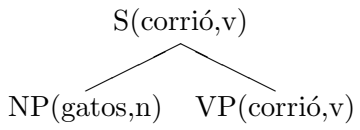


Figure 4-5: An ungrammatical dependency: the plural noun *gatos* is unlikely to modify the singular verb *corrió*.

the noun phrase (NP) with headword *gatos* and headtag noun (n) is defined as follows:[4]

$$\begin{aligned} P(\text{gatos, n, NP} \mid \text{corrió, v, S, VP}) &= P_1(\text{n, NP} \mid \text{corrió, v, S, VP}) \times \\ & \quad P_2(\text{gatos} \mid \text{n, NP, corrió, v, S, VP}) \end{aligned}$$

The parser smooths parameter values using backed-off statistics, and in particular smooths statistics based on headwords with coarser statistics based on POS tags alone. This allows the parser to effectively use POS tags as a way of separating different lexical items into subsets or classes depending on their syntactic behavior. In our example, each term is estimated as follows:

$$\begin{aligned} P_1(\text{n, NP} \mid \text{corrió, v, S, VP}) &= \lambda_{1,1}\hat{P}_{1,1}(\text{n, NP} \mid \text{corrió, v, S, VP}) + \\ & \quad \lambda_{1,2}\hat{P}_{1,2}(\text{n, NP} \mid \text{v, S, VP}) + \\ & \quad \lambda_{1,3}\hat{P}_{1,3}(\text{n, NP} \mid \text{S, VP}) \end{aligned}$$

$$\begin{aligned} P_2(\text{gatos} \mid \text{n, NP, corrió, v, S, VP}) &= \lambda_{2,1}\hat{P}_{2,1}(\text{gatos} \mid \text{n, NP, corrió, v, S,VP}) + \\ & \quad \lambda_{2,2}\hat{P}_{2,2}(\text{gatos} \mid \text{n, NP, v, S, VP}) + \\ & \quad \lambda_{2,3}\hat{P}_{2,3}(\text{gatos} \mid \text{n}) \end{aligned}$$

Here the $\hat{P}_{i,j}$ terms are maximum likelihood estimates derived directly from counts in the training data. The $\lambda_{i,j}$ parameters are defined so that $\lambda_{1,1}+\lambda_{1,2}+\lambda_{1,3} = \lambda_{2,1}+\lambda_{2,2}+\lambda_{2,3} = 1$. They control the relative contribution of each level of back-off to the final estimate.

Note that thus far our example has not included any morphological information in the POS tags. Because of this, we will see that there is a danger of the estimates P_1 and P_2 both being high, in spite of the dependency being ungrammatical. P_1 will be high because

[4]Note that the parsing model includes other features such as distance which we omit from the parameter definition for simplicity.

all three estimates $\hat{P}_{1,1}, \hat{P}_{1,2}$ and $\hat{P}_{1,3}$ will most likely be high: there is no indication in any of these estimates that the NP sister is headed by a plural noun, while the VP on which it is dependent is headed by a singular verb.

Next, consider P_2. Of the three estimates $\hat{P}_{2,1}, \hat{P}_{2,2}$, and $\hat{P}_{2,3}$, only $\hat{P}_{2,1}$ contains any indication that the noun is plural and the verb is singular. Thus P_2 will be sensitive to the morphological clash between *gatos* and *corrió* only if $\lambda_{2,1}$ is high, reflecting a high level of confidence in the estimate of $\hat{P}_{2,3}$. This will only happen if the context ⟨corrió, v, S, VP⟩ is seen frequently enough for $\lambda_{2,1}$ to take a high value. This is unlikely, given that this context is quite specific. In summary, the model can only capture morphological restrictions through lexically-specific estimates based on extremely sparse statistics.

Now consider a model that incorporates morphological information — in particular, number information — in the noun and verb POS tags. *gatos* will have the POS tag pn, signifying a plural noun; *corrió* will have the POS tag sv, signifying a singular verb. All estimates in the previous equations will reflect these part-of-speech changes. For example, P_1 will now be estimated as follows:

$$\begin{aligned} P_1(\texttt{pn, NP} \mid \text{corrió, } \texttt{sv, S, VP}) \;=\; & \lambda_{1,1}\hat{P}_{1,1}(\texttt{pn, NP} \mid \text{corrió, } \texttt{sv, S, VP}) + \\ & \lambda_{1,2}\hat{P}_{1,2}(\texttt{pn, NP} \mid \texttt{sv, S, VP}) + \\ & \lambda_{1,3}\hat{P}_{1,3}(\texttt{pn, NP} \mid \texttt{S, VP}) \end{aligned}$$

Note that the two estimates $\hat{P}_{1,1}$ and $\hat{P}_{1,2}$ include an (unlikely) dependency between the POS tags pn and sv. Both of these estimates will be 0, assuming that a plural noun is never seen as the subject of a singular verb, and even if the context ⟨corrió, sv, S, VP⟩ isn't seen frequently enough during training for $\lambda_{1,1}$ to be very high, the context ⟨sv, S, VP⟩ should be. Therefore we'd expect $\hat{P}_{1,2}$ to be a reliable estimate, thus correctly assigning low probability to the ungrammatical dependency.

The backed-off statistics used to estimate P_2 will of course also be affected:

$$\begin{aligned} P_2(\text{gatos} \mid \texttt{pn, NP, } \text{corrió, } \texttt{sv, S, VP}) \;=\; & \lambda_{2,1}\hat{P}_{2,1}(\text{gatos} \mid \texttt{pn, NP, } \text{corrió, } \texttt{sv, S,VP}) + \\ & \lambda_{2,2}\hat{P}_{2,2}(\text{gatos} \mid \texttt{pn, NP, sv, S, VP}) + \\ & \lambda_{2,3}\hat{P}_{2,3}(\text{gatos} \mid \texttt{pn}) \end{aligned}$$

Note, however, that the effect on P_2 is likely not to be as powerful as that on P_1: assuming it is unlikely to see a plural noun coupled with a singular verb during training, both $\lambda_{2,1}$ and $\lambda_{2,2}$ will probably be low, leaving most of the estimate for P_2 to be determined by $\hat{P}_{2,3}$. This is just the probability that the word *gatos* is a plural noun, which should be high.

In summary, the morphologically-rich Model M can make use of non-lexical statistics such as $\hat{P}_{1,2}(\texttt{pn}, \texttt{NP} \mid \texttt{sv}, \texttt{S}, \texttt{VP})$ which contain dependencies between POS tags and which will most likely be estimated reliably by the model. The hope is that for unlikely dependencies, the corresponding loss in value for the estimate P_1 will be greater than any potential gain in value for P_2.

4.3.2 Model R: The Reranking Model

In the reranking model, we use an n-best version of the Model M parser to generate a number of candidate parse trees for each sentence in training and test data. These parse trees are then represented through a combination of the log probability under the initial model, together with a large number of global features. A linear reranking model uses the information from these features to derive a new ranking of the n-best parses, with the hope of improving upon the first-pass model. This same approach has been used with success in parsing English (e.g., (Collins & Koo, 2005)). There are a variety of methods for training the parameters of the model. In this work, we use the exponentiated gradient algorithm described in (Bartlett et al., 2004).

The motivation for using the reranking model is that a wide variety of features, which can essentially be sensitive to arbitrary context in the parse trees, can be incorporated by choosing a feature-based model such as a linear model. In our work, we included all features described in (Collins & Koo, 2005). As far as we are aware, this is the first time that a reranking model has been applied to parsing a language other than English. One goal was to investigate whether the improvements seen on English parsing can be carried across to another language. We have found that features in (Collins & Koo, 2005), initially developed for English parsing, also give appreciable gains in accuracy when applied to Spanish.

Non-Terminal	Significance
aq	*adjective*
cc	*conjunction*
COORD	*coordinated phrase*
ESPEC	*determiner*
GRUP	*base noun phrase*
GV	*verb phrase*
MORF	*impersonal pronoun*
p	*pronoun*
PREP	*base prepositional phrase*
RELATIU	*relative pronoun phrase*
s	*adjectival phrase*
SN	*noun phrase*
SP	*prepositional phrase*
SADV	*adverbial phrase*
S	*sentence*
sps	*preposition*
v	*verb*

Figure 4-6: The non-terminals and part-of-speech labels from the Spanish 3LB corpus used in this chapter.

4.4 Data

The corpus we use is a version of the Spanish 3LB treebank, a freely-available resource with about 3,500 sentence/tree pairs. The average sentence length is 28 tokens. It is a mixed–genre corpus containing data from 38 complete articles and short texts. Roughly 27% of the texts are news articles, 27% scientific articles, 14% narrative, 11% commentary, 11% sports articles, 6% essays, and 5% articles from weekly magazines. The trees contain information about both constituency structure and syntactic functions.

4.4.1 Preprocessing

It is well-known that tree representation influences parsing performance (Johnson, 1998). Prior to training our models, we made some systematic modifications to the corpus trees in an effort to make it easier for Model 1 to represent the linguistic phenomena present in the trees. The table in Figure 4-6 provides a key to the non-terminal labels in the 3LB treebank that are used in this section's discussion of preprocessing and in the remainder of the chapter as well.

Relative and Subordinate Clauses Cases of relative and subordinate clauses appearing in the corpus trees have the basic structure of the top tree in Figure 4-7. The bottom tree in the figure shows the modifications we make to such structures. The modified structure has the advantage that the `SBAR` selects the `CP` node as its head, making the relative pronoun *quien* the headword for the root of the subtree. This change allows, for example, better modeling of verbs that select for particular complementizers. In addition, the new subtree rooted at the `S` node now looks like a top-level sentence, making sentence types more uniform in structure and easier to model statistically. Also, the new structure differentiates phrases embedded in the complementizers of `SBAR`s from those used in other contexts, allowing relative pronouns like *quien* in Figure 4-7 to surface as lexical headwords when embedded in larger phrases beneath the `CP` node.[5]

Coordination In the 3LB treebank, coordinated constituents and their coordinating conjunction are placed as sister nodes in a flat structure. The modifications we make to this structure are illustrated in Figure 4-8. The modifications help to rule out unlikely phrases such as *cats and dogs and*: the model trained with the original treebank structures will assign non-zero probability to ill-formed structures such as these.

Pro-Drop Subject pronouns in Spanish are often dropped, a phenomenon common to several languages and known as *pro-drop*. The Spanish 3LB treebank marks this phenomenon explicitly. In all of our experiments, we remove nodes representing these elliptical subjects. Otherwise, our parser would have to predict structure for words missing from the input.

Punctuation Prior to training we delete all punctuation at the beginning and end of sentences. In addition, we eliminate a punctuation node if it is the only child of its parent. This guarantees that every non-terminal has a legitimate headchild. Finally, we raise any remaining punctuation to the highest possible position in the tree. Specifically, anywhere in the tree where punctuation is the left- or right-most child of its parent, we raise that node in the tree until it falls between two non-punctuation non-terminals.

[5]This is achieved through our head rules.

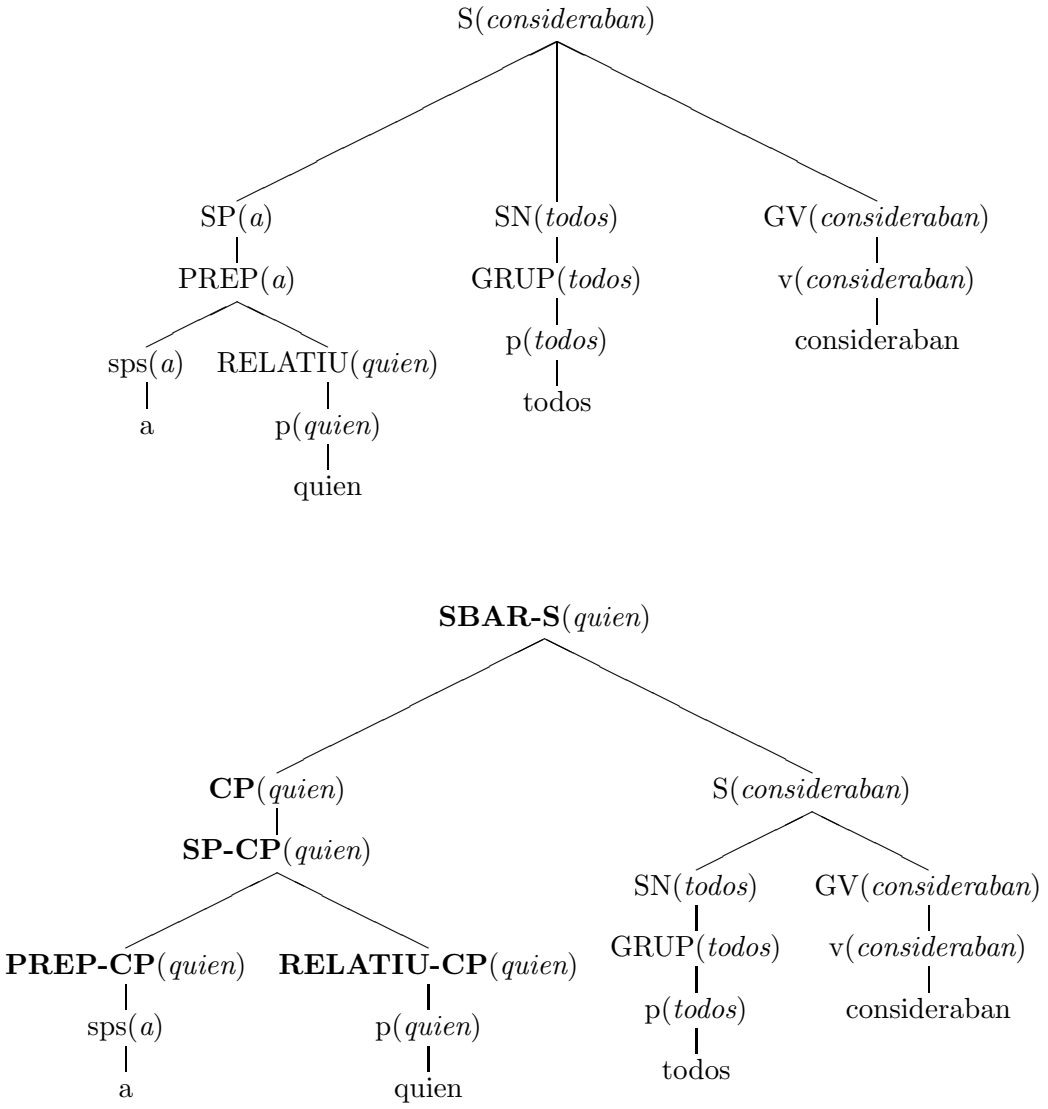


Figure 4-7: Preprocessing of relative and subordinate clauses: the top tree in the figure is the original structure from the 3LB treebank for the phrase *a quien todos consideraban* or *whom everyone considered*. The lower tree shows the modifications we make (new nodes are shown in bold): we insert `SBAR` and `CP` nodes and mark all non-terminals below the `CP` with a `-CP` tag. In both trees, the headwords are shown in italics next to the nonterminal labels.

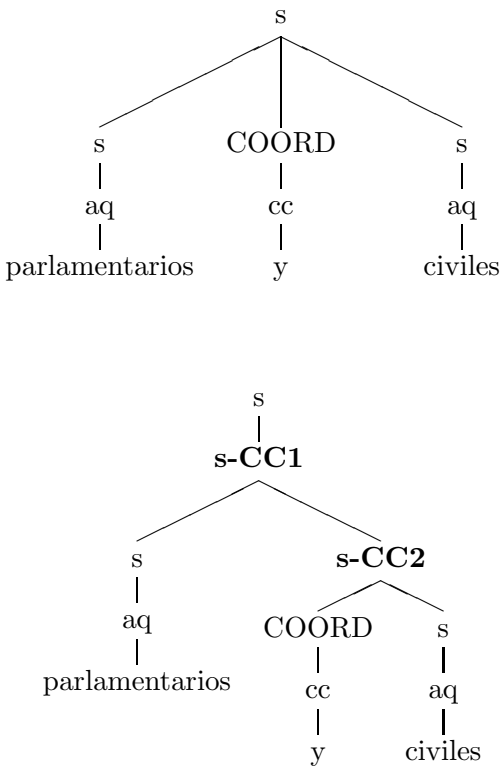


Figure 4-8: In the 3LB corpus, phrases involving coordination are represented with a flat structure as in the top tree. For coordination involving a non-terminal X (X = `s` in the example), we insert new nodes X`-CC1` and X`-CC2` to form the structure in the bottom tree. The new nodes are shown in bold face.

4.5 Experiments

4.5.1 Experimental Setup

Our baseline model and Model M (the morphological model) were both trained using a set consisting of 80% of the data (2,800 sentence/tree pairs, 75,372 words) available in the 3LB treebank. Both models used the Collins' thesis parser code, Model 1 (Collins, 1999). The only modifications to the code involved changes to accommodate the expanded part-of-speech tagsets.

We reserved the remaining 20% of the 3LB data (692 sentences, 19,343 words) to use as unseen data in a test set. We selected these subsets with two criteria in mind: first, we wanted to respect the boundaries of the texts by placing articles in their entirety into either one subset or the other; and second, we wanted to maintain, in each subset, the same proportion of genres found in the original set of trees.

During development, we used a cross-validation approach on the training set to test different tagsets. We divided the 2,800 training data trees into 14 different development data sets, where each of these data sets consisted of 2,600 training sentences and 200 development sentences. We took the average over the results of the 14 splits to gauge the effectiveness of the tagset being tested.

4.5.2 Evaluation Metrics

To evaluate our models, we considered the recovery of labeled dependencies, unlabeled dependencies, and labeled constituents. Unlabeled dependencies capture how the words in a sentence depend on one another. Formally, they are tuples {*headchild index*, *modifier index*}, where the indices indicate position in the sentence. Labeled dependencies include the labels of the modifier, headchild, and parent non-terminals. The root of the tree has a special dependency: {*head index*} in the unlabeled case and {*TOP*, *headchild index*, *root non-terminal*} in the labeled case.

The labeled constituents in a tree are all of the non-terminals and, for each, the positions of the words it spans.

We use the standard definitions of precision, recall, and F-measure:[6]

$$\text{precision} = 100 * \frac{num\ correct}{num\ output\ by\ parser}$$

$$\text{recall} = 100 * \frac{num\ correct}{num\ in\ gold\ standard}$$

$$\text{F-measure} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

where *num correct* represents the number of correct dependencies/constituents output by the parser; *num output by parser* is the total number of dependencies/constituents output by the parser; and *num in gold standard* is the total number of dependencies/constituents found in the gold standard parses.

4.5.3 The Effects of Morphology

In our first set of experiments, we trained over 50 models, incorporating different morphological information into each via their tagsets. Prior to running the parsers, we trained the POS tagger described in (Collins, 2002). The output from the tagger was used to assign POS tags to unknown words. During both training and testing, we made no attempt to parse sentences with length greater than 70 words.

The tables in Figures 4-9 and 4-10 give results for several of the tagsets we tried during development. Our baseline model, which we used to evaluate the effects of using morphology, was Model 1 (Collins, 1999) with a simple POS tagset containing almost no morphological information.[7] The morphological models we show are meant to be representative of both the highest-scoring models and the performance of various morphological features. For instance, we found that, in general, gender had only a slight impact on the performance of the parser. Note that gender is not a morphological attribute of Spanish verbs, and that the inclusion of verbal features, particularly number, mode, and person, generated the

[6] When extracting dependencies, we replaced all non-punctuation POS tags with a generic tag *TAG* to avoid conflating tagging errors with dependency errors. We also included the structural changes that we imposed during preprocessing. Results for constituent precision and recall were computed after we restored the trees to the original treebank structure.

[7] I.e., a tagset with just the most basic POS labels: V for verb, N for noun, etc. We say this tagset contains "almost no" morphological information because POS tags of any type can be said to carry some morphological information.

	Model	Labeled Dep		Unlabeled Dep	
		Prec/Rec	**Gain**	**Prec/Rec**	**Gain**
1	Baseline	76.0	—	82.1	—
2	NUM(p,n,v)	78.4	2.4	83.6	1.5
3	NUM(a,d,n,p,v)	78.2	2.2	83.5	1.4
4	NUM(v)	77.8	1.8	82.9	0.8
5	MODE(v)	78.4	2.4	83.1	1.0
6	TENSE(v)	77.6	1.6	82.7	0.6
7	PERS(v)	78.1	2.1	83.3	1.2
8	GEN(v)	76.3	0.3	82.2	0.1
9	**NUM(a,d,n,v,p)+MODE(v)**	**79.0**	**3.0**	**84.0**	**1.9**
10	NUM(p,n,v)+MODE(v)	78.9	2.9	83.7/83.8	1.6/1.7
11	NUM(a,d,n,v,p)+MODE(v)+PERS(v)	78.7	2.7	83.6	1.5
12	NUM(a,d,n,v,p)+PERS(v)	78.4	2.4	83.5/83.6	1.4/1.5
13	NUM(a,d,n,v,p)+GEN(a,d,n,v,p)	78.1	2.1	83.2	1.1

Figure 4-9: Results on recovery of labeled and unlabeled dependencies after training morphological models during development. Note that dependencies are computed over parsed structures which include the structural changes we described in Section 4.4.1, while the dependencies extracted over the gold-standard structures have not undergone this preprocessing. Because of this difference, precision and recall of dependencies may differ slightly. Row 1 shows results on a baseline model containing almost no morphological information. The subsequent rows represent the different tagsets we used for experimentation: NUM is *number*, PERS *person*, and GEN *gender*. The letters in parentheses represent different POS categories. We use "p" for *pronoun*, "n" for *noun*, "d" for *determiner*, "v" for *verb*, and "a" for *adjective*. The results of the best-performing tagset are in bold.

strongest-performing models in our experiments.

The table in Figure 4-11 shows the results of running two models on the test set: the baseline model and Model M — the best-performing morphological model from the development stage. This model uses the number and mode of verbs, as well as the number of adjectives, determiners, nouns, and pronouns.

The results in Figures 4-9, 4-10, and 4-11 show that adding some amount of morphological information to a parsing model is beneficial. We found, however, that adding more information does not always lead to improved performance (see, for example, rows 11 and 13 in Figures 4-9 and 4-10). Presumably this is because the training set is too small to adequately model all of the new tags in the tagset.[8]

[8]The number of new tags formed is the size of the set that results from taking the cross product between the sets containing the possible values of each morphological feature added to the model.

	Model	Labeled Const			
		<=70 words		<=40 Words	
		Prec	Rec	Prec	Rec
1	Baseline	81.6	80.4	82.6	81.4
2	NUM(p,n,v)	83.1	82.5	84.1	83.4
3	NUM(a,d,n,p,v)	83.3	82.4	84.2	83.3
4	NUM(v)	82.3	81.6	83.1	82.2
5	MODE(v)	82.8	82.0	83.8	82.9
6	TENSE(v)	82.4	81.4	83.2	82.3
7	PERS(v)	82.9	82.0	83.8	82.8
8	GEN(v)	81.6	80.6	82.7	81.7
9	**NUM(a,d,n,v,p)+MODE(v)**	**83.9**	**83.2**	**84.7**	**84.1**
10	NUM(p,n,v)+MODE(v)	83.6	82.8	84.6	83.7
11	NUM(a,d,n,v,p)+MODE(v)+PERS(v)	83.6	82.9	84.4	83.8
12	NUM(a,d,n,v,p)+PERS(v)	83.3	82.6	84.2	83.5
13	NUM(a,d,n,v,p)+GEN(a,d,n,v,p)	83.1	82.5	83.9	83.4

Figure 4-10: Results on development set for recovery of labeled constituents for sentences of 70 words or less and 40 words or less. When precision and recall differ in labeled or unlabeled dependencies, both scores are shown. Row 1 shows results on a baseline model containing almost no morphological information. The subsequent rows represent the different tagsets we used for experimentation: NUM means *number*, PERS means *person*, and GEN means *gender*. The letters in parentheses represent different POS categories. We use "p" for *pronoun*, "n" for *noun*, "d" for *determiner*, "v" for *verb*, and "a" for *adjective*. The results of the best-performing tagset are in bold.

	Model	Labeled Dep	Unlabeled Dep	Labeled Const			
				<=70 words		<=40 Words	
		Prec/Rec	Prec/Rec	Prec	Rec	Prec	Rec
1	Baseline	77.0	82.5	81.7	80.8	83.1	82.0
2	Model M	79.4	83.9	83.9	83.4	85.1	84.4
3	Model R	80.2	84.7	85.2	85.0	86.3	85.9
4	Model 1, Eng	–	–	82.9	82.9	83.5	83.4

Figure 4-11: Results after running the morphological and reranking models on test data. Row 1 is our baseline model. Row 2 is the model with the highest-achieving tagset during development (NUM(a,d,n,v,p)+MODE(v)). Row 3 gives the accuracy of the reranking approach, when applied to n-best output from the model in Row 2. Row 4 shows the results from training Model 1 (Collins, 1999) using 2,800 English trees from the Penn treebank and testing on section 23. This row gives us an idea how Model 1 performs when trained with the same amount of data we have to train the Spanish parser.

4.5.4 Experiments with Reranking

In the reranking experiments, we follow the procedure described in (Collins & Koo, 2005) for creation of a training set with n-best parses for each sentence. This method involves jack-knifing the data: the training set of 2,800 sentences was parsed in 200-sentence chunks

by an n-best version of the Model M parser trained on the remaining 2,600 sentences. This ensured that each sentence in the training data had n-best output from a baseline model that was not trained on that sentence. We used the optimal morphological model (NUM(a,d,n,v,p)+MODE(v)) to generate the n-best lists, and we used the feature set described in (Collins & Koo, 2005). The test results are given in Figure 4-11.[9]

4.5.5 Statistical Significance

We tested the significance of the labeled precision and recall results in Figure 4-11 using the sign test. When applying the sign test, for each sentence in the test data we calculated the sentence-level F1 constituent score for the two parses being compared. This indicates whether one model performs better on that sentence than the other model, or whether the two models perform equally well, information used by the sign test. All differences were found to be statistically significant at the level $p = 0.01$. When comparing the baseline model to the morphological model on the 692 test sentences, F1 scores improved on 314 sentences, and became worse on 164 sentences. When comparing the baseline model to the reranked model, 358 sentences had improved parses, while 157 had worse parses. When comparing the morphological model to the reranked model, 199 sentences had improved parses and 106 had worse parses.

4.6 Further Analysis of Model M

Figure 4-12 takes a closer look at the performance of Model M in the recovery of particular labeled dependencies. The breakdown shows the top 15 dependencies in the gold-standard trees across the entire training set. Collectively, these dependencies represent around 72% of the dependencies seen in this data.

We see a large gain in the recovery of some of these dependencies when we add morphological information. Among these are the two involving postmodifiers to verbs (⟨`S GV SP` R⟩ and ⟨`S GV SN` R⟩). When examining the output of the morphological model, we found that much of this gain is due to the fact that there are two non-terminal labels used in the treebank that specify modal information of verbs they dominate (infinitivals and gerunds):

[9] We also created development sets for development of the reranking approach, and for cross-validation of the single parameter C for the exponentiated gradient training of (Bartlett et al., 2004).

Dependency	Count	Model	Prec/Rec
Determiner modifier	9680	BL	95.0/95.4
SN GRUP ESPEC L	(15.5%)	M	95.4/95.7
Complement of SP	9052	BL	92.4/92.9
SP PREP SN R	(14.5%)	M	93.2/93.9
SP modifier to noun	4500	BL	83.9/78.1
GRUP TAG SP R	(7.2%)	M	82.9/79.9
Subject	3106	BL	77.7/86.1
S GV SN L	(5.0%)	M	83.1/87.5
Sentential head	2758	BL	75.0/75.0
TOP S	(4.4%)	M	79.7/79.7
S modifier under SBAR	2728	BL	83.3/82.1
SBAR CP S R	(4.4%)	M	86.0/84.7
SP modifier to verb	2685	BL	62.4/78.8
S GV SP R	(4.3%)	M	72.6/82.5
SN modifier to verb	2677	BL	71.6/75.6
S GV SN R	(4.3%)	M	81.0/83.0
Adjective postmodifier	2522	BL	76.3/83.6
GRUP TAG s R	(4.0%)	M	76.4/83.5
Adjective premodifier	980	BL	79.2/80.0
GRUP TAG s L	(1.6%)	M	80.1/79.3
SBAR modifier to noun	928	BL	62.2/60.6
GRUP TAG SBAR R	(1.4%)	M	61.3/60.8
Coordination	895	BL	65.2/72.7
S-CC2 S coord L	(1.4%)	M	66.7/74.2
Coordination	870	BL	52.4/56.1
S-CC1 S-CC2 S L	(1.4%)	M	60.3/63.6
Impersonal pronoun	804	BL	93.3/96.4
S GV MORF L	(1.3%)	M	92.0/95.6
SN modifier to noun	736	BL	47.3/39.5
GRUP TAG SN R	(1.2%)	M	51.7/50.8

Figure 4-12: Labeled dependency accuracy for the top fifteen dependencies (representing around 72% of all dependencies) in the gold-standard trees across all training data. The first column shows the type and subtype, where the subtype is specified as the 4-tuple {*parent non-terminal*, *head non-terminal*, *modifier non-terminal*, *direction*}; the second column shows the count for that subtype and the percent of the total that it represents (where the total is 62,372) . The model BL is the baseline, and M is the morphological model NUM(a,d,n,v,p)+MODE(v).

with insufficient morphological information, the baseline parser was unable to distinguish regular verb phrases from these more specific verb phrases.

In order to see just how much mode information was being used to distinguish these verbal non-terminals, we tried replacing all INFINITIU and GERUNDI non-terminals with the

Dependency	Count	Model	Prec/Rec
SN modifier to verb	3401	BL	79.9/81.6
S GV SP R	(5.5%)	M	82.1/83.3
SP modifier to verb	3313	BL	68.9/81.1
S GV SN R	(5.3%)	M	72.7/81.8

Figure 4-13: Accuracy for two labeled dependencies involving verbal modifiers using training data that collapses three verb-phrase non-terminals (INFINITIU, GERUNDI, and GV) to a single verb-phrase non-terminal. Results are given for the baseline model and Model M.

non-terminal GV in the training/development set, and then we retrained the baseline model and Model M. In the baseline model, the recovery of labeled dependencies improved considerably (77.8%), and the recovery of unlabeled dependencies was about constant (82.4%); in the morphological model, performance remained fairly constant (79.4% labeled dependencies, 84.0% unlabeled dependencies). Figure 4-13 shows how each model did in the recovery of verbal postmodifiers in particular. We see that the behavior of Model M is roughly equivalent to what it was when we had three different verb-phrase non-terminals, but that the baseline model is doing much better. However, even though the baseline has improved, it still does not do as well as Model M, suggesting that there is additional information supplied by mode.

We've looked at how mode information is contributing to Model M's improved performance, but how is number being used? Number could help the parser decide not to attach elements with differing number information in situations where they are constrained to agree. Figure 4-14 suggests that this is (at least in part) what's happening: of the top 20 dependencies with the largest difference in error when comparing Model M to the baseline, half of them involve modifiers and heads with differing number information. For example, the baseline model, without any notion of number, tries to attach an plural adjective to a singular noun, creating a false dependency. This kind of error accounts for a smaller proportion of the errors induced by the morphological model.

4.6.1 Is More Data Better?

The results in Figure 4-11 show that the morphological model we employ performs at least as well as the Model 1 trained on a comparable amount of English data. Model 1 shows a substantial gain in performance with an increased amount of training data (on the order of 90% precision and recall with a training set of roughly 40,000 trees); would our model

Diff	Dependency
0.4	S GV(vsi) SN(n0) **R**
0.4	S GV(vsi) SP(sps) R
0.4	S GV(aqs) SP(sps) R
0.4	S#CC1 S#CC2(v0n) S(vsi) **L**
0.3	SBAR CP(cs) S(v0n) **R**
0.3	TOP(v0g) S(v0g)
0.3	TOP(v0n) S(v0n)
0.2	SP prep(sps) S(vsi) R
0.2	GRUP TAG(ns) s(aqp) R
0.2	S GV(v0g) SN(ns) **L**
0.2	S GV(vsi) S(vsi) R
0.2	GRUP TAG(n0) s(ns) **L**
0.2	GRUP TAG(ns) SN(np) **R**
0.2	SN GRUP(n0) ESPEC(dp) **L**
0.2	S GV(v0n) SN(ns) **L**
0.2	S GV(v0g) SN(np) **L**
0.2	GRUP TAG(n0) SP(sps) R
0.1	S GV(vpi) SN(n0) **R**
0.1	S GV(v0g) SADV(rg) L
0.1	S GV(vpi) SP(sps) R

Figure 4-14: The effects of adding number information to a morphologically-sensitive parsing model. The table shows the twenty dependencies having the largest difference in error when comparing the baseline model to a morphological model in which `GERUNDI` and `INFINITIU` have been collapsed to a single nonterminal `GV`. Half of these dependencies involve a mismatch in number information, indicating that the morphological model is doing a better job of predicting syntactic constraints based on number. In parentheses next to each headchild and modifier in column 2 is the gold-standard POS tag of the headword of that non-terminal. For adjectives, determiners, nouns, verbs, and pronouns, the second character indicates the number attribute (*s* is *singular*; *p* is *plural*; and *0* is *no number info*).

exhibit a similar gain with more training data?

Figure 4-15 shows a plot of the performance of our best morphological model when trained on data sets of varying sizes. For these experiments, we used the same approach described in section 4.5.4; however, rather than use the entire training set of 2,600 trees at each iteration, we selected a random sample of our target size. The development sets used for testing were constant across experiments of different training-set sizes. Note that we retrained the POS tagger employed by our parser to reflect a restricted amount of available data.

The plot shows a substantial gain in accuracy when we increase the size of the training set from 400 to 800 and from 800 to 1600. Between 1600 and 2800, however, the growth rate slows considerably. Whether the performance would continue to improve, albeit at a

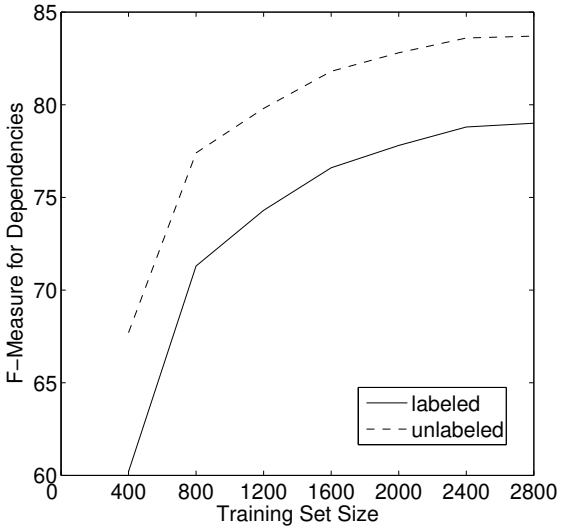


Figure 4-15: The performance of the morphological model n(A,D,N,V,P)+m(V) when trained on data sets of varying sizes (400, 800, 1200, 1600, 2000, 2400, 2800). The plot shows size vs. F-measure accuracy of both labeled and unlabeled dependencies.

decreased rate, to reach the levels of English parsing performance given larger amounts of data, or whether it would reach a plateau at an inferior level of performance is unknown. However, we should also keep in mind that with larger amounts of data, we might be able to employ more sophisticated morphological models that could also help parsing performance.

Chapter 5

A Discriminative Model for Tree-to-Tree Translation

This chapter proposes a statistical tree-to-tree model for producing translations. Two main contributions are as follows: (1) the extraction of syntactic structures with alignment information from a parallel corpus of translations, and (2) the use of a discriminative, feature-based model for prediction of these target-language syntactic structures — which we call *aligned extended projections*, or AEPs. An evaluation of the method on translation from German to English shows similar performance to the phrase-based model of Koehn et al. (Koehn et al., 2003). This chapter is based on work originally described in (Cowan et al., 2006).

5.1 Introduction

Phrase-based approaches (Och & Ney, 2004) to statistical machine translation have recently achieved impressive results, leading to significant improvements in accuracy over the original IBM models (Brown et al., 1993). However, phrase-based models lack a direct representation of syntactic information in the source or target languages; this has prompted several researchers to consider various approaches that make use of syntactic information.

This chapter describes a framework for tree-to-tree based statistical translation. Our goal is to learn a model that maps parse trees in the source language to parse trees in the target language. The model is learned from a corpus of translation pairs, where each sentence in the source or target language has an associated parse tree. We see two major

benefits of tree-to-tree based translation. First, it is possible to explicitly model the syntax of the target language, thereby improving grammaticality. Second, we can build a detailed model of the correspondence between the source and target parse trees, thereby attempting to construct translations that preserve the meaning of source language sentences.

Our translation framework involves a process where the target-language parse tree is broken down into a sequence of clauses, and each clause is then translated separately. A central concept we introduce in the translation of clauses is that of an *aligned extended projection* (AEP). AEPs are derived from the concept of an *extended projection* in lexicalized tree adjoining grammars (LTAG) (Frank, 2002), with the addition of alignment information that is based on work in synchronous LTAG (Shieber & Schabes, 1990). A key contribution of this chapter is a method for learning to map parsed German clauses to AEPs using a feature-based model with a perceptron learning algorithm.

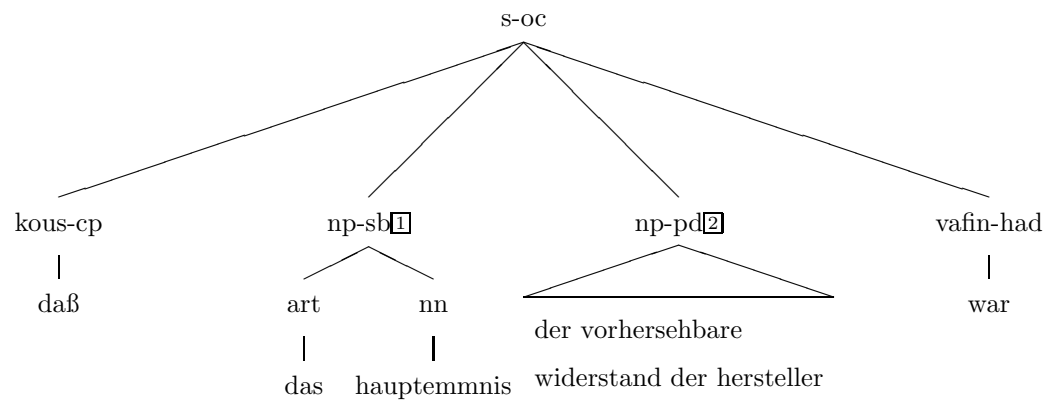
We performed experiments on translation from German to English on the Europarl data set. Evaluation in terms of both BLEU scores and human judgments shows that our system performs similarly to the phrase-based model of Koehn et al. (Koehn et al., 2003).

5.1.1 A Sketch of the Approach

This section provides an overview of the translation process. We will use the German sentence *wir wissen daß das haupthemmnis der vorhersehbare widerstand der hersteller war* as a running example. For this example we take the desired translation to be *we know that the main obstacle has been the predictable resistance of manufacturers*. Translation of a German sentence proceeds in the following four steps:

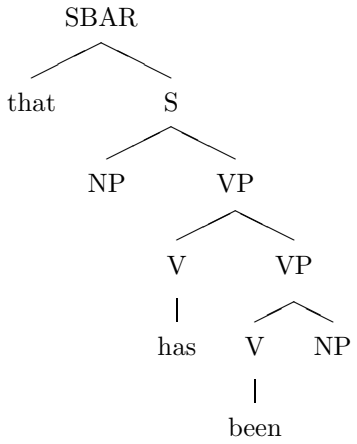
Step 1: The German sentence is parsed and then broken down into separate parse structures for a sequence of clauses. For example, the German example above is broken into a parse structure for the clause *wir wissen* followed by a parse structure for the subordinate clause *daß...war*. Each of these clauses is then translated separately, using steps 2–3 below.

Step 2: A structure that we will call an *aligned extended projection* (AEP) is predicted for each German clause. To illustrate this step, consider translation of the second German clause, which has the following parse structure:



Note that we use the symbols 1 and 2 to identify the two modifiers (arguments or adjuncts) in the clause, in this case a subject and an object.

A major part of the AEP is a parse-tree fragment similar to a TAG elementary tree (see also Figure 5-1):



Following the work of (Frank, 2002), we will refer to a structure like this as an *extended projection* (EP). The EP encapsulates the core syntactic structure in the English clause. It contains the main verb *been*, as well as the function words *that* and *has*. It also contains a parse tree "spine" which has the main verb *been* as one of its leaves, and has the clause label `SBAR` as its root. In addition, it specifies positions for arguments in the clause — in this case NPs corresponding to the subject and object.

An AEP contains an EP, as well as *alignment information* about where the German modifiers should be placed in the extended projection. For example, the AEP in this case would contain the tree fragment shown above, together with an alignment specifying that the modifiers 1 and 2 from the German parse will appear in the EP as subject and object, respectively.

Step 3: The German modifiers are translated and placed in the appropriate positions within the AEP. For example, the modifiers *das haupthemmnis* and *der vorhersehbare widerstand der hersteller* would be translated as *the main obstacle*, and *the predictable resistance of manufacturers*, respectively, and then placed into the subject and object positions in the AEP.

Step 4: The individual clause translations are combined to give a final translation. For example, the translations *we know* and *that the main obstacle has been ...* would be concatenated to give *we know that the main obstacle has been ...*.

The main focus of this chapter will be Step 2: the prediction of AEPs from German clauses. AEPs are detailed structural objects, and their relationship to the source-language clause can be quite complex. We use a discriminative feature-based model, trained with the perceptron algorithm, to incrementally predict the AEP in a sequence of steps. At each step we define features that allow the model to capture a wide variety of dependencies within the AEP itself, or between the AEP and the source-language clause.

5.1.2 Motivation for the Approach

Our approach to tree-to-tree translation is motivated by several observations. Breaking the source-language tree into clauses (Step 1) considerably simplifies the difficult problem of defining an alignment between source and target trees. Our impression is that high-quality translations can be produced in a clause-by-clause fashion.[1] The use of a feature-based model for AEP prediction (Step 2) allows us to capture complex syntactic correspondences between English and German, as well as grammaticality constraints on the English side.

In this chapter, we implement the translation of modifiers (Step 3) with the phrase-based system of (Koehn et al., 2003). The modifiers in our data set are generally small chunks of text such as NPs, PPs, and ADJPs, which by definition do not include clauses or verbs. In our approach, we use the phrase-based system to generate n-best lists of candidate translations and then rerank the translations based on grammaticality, i.e., using criteria that judge how well they fit the position in the AEP. In future work, we might use finite state machines in place of a reranking approach, or recursively apply the AEP approach to

[1] We do not assume that all of the translations in the training data have been produced in a clause-by-clause fashion. Rather, we assume that good translations for test examples can be produced in this way.

the modifiers.

Stitching translated clauses back together (Step 4) is a relatively simple task: in a substantial majority of cases, the German clauses are not embedded, but instead form a linear sequence that accounts for the entire sentence. In these cases we can simply concatenate the English clause translations to form the full translation. Embedded clauses in German are slightly more complicated, but it is not difficult to form embedded structures in the English translations.

Section 5.4.2 of this chapter provides an overview of the features we use for AEP prediction in translation from German to English (see Appendix E for an in-depth description). Many of the features of the AEP prediction model are specifically tuned to the choice of German and English as the source and target languages. However, it should be easy to develop new feature sets to deal with other languages or treebanking styles. This is one of the strengths of the feature-based approach.

In the work presented in this chapter, we focus on the prediction of clausal AEPs, i.e., AEPs associated with main verbs. One reason for this is that clause structures are particularly rich and complex from a syntactic perspective. This means that there should be considerable potential in improving translation quality if we can accurately predict these structures. It also means that clause-level AEPs are a good test-bed for the discriminative approach to AEP prediction; future work may consider applying these methods to other structures such as NPs, PPs, ADJPs, and so on.

5.2 A Translation Architecture Based on AEPs

5.2.1 Aligned Extended Projections (AEPs)

We now build on the idea of extended projections (see Section 2.5.2) to give a detailed description of AEPs. Figure 5-1 shows examples of German clauses paired with the AEPs found in training data.[2] The German clause is assumed to have n (where $n \geq 0$) modifiers. For example, the first German parse in Figure 5-1 has two arguments, indexed as 1 and 2.[3] Each of these modifiers must either have a translation in the corresponding English clause,

[2]Because we consider translation from German to English in Section 5.6 of this chapter, we take *English* to be synonymous with the target language in translation and *German* to be synonymous with the source language.

[3]The parses in the figure are shown in an indented format, where each indentation corresponds to a level in the tree.

or must be deleted. In the second example, the correspondence between the German clause and its English translation is not entirely direct. The subject in the English is the expletive *there*; the subject in the German clause becomes the object in English. This is a typical pattern for the German verb *bestehen*. The German `PP` *zwischen ...* appears at the start of the clause in German, but is post-verbal in the English. The modifier *also* — whose English translation is *so* — is in an intermediate position in the German clause, but appears in the `pre-subject` position in the English clause. The AEP representation is able to model these and many other German/English correspondences.

An AEP consists of the following parts:

STEM: A string specifying the stemmed form of the main verb in the clause.

SPINE: A syntactic structure associated with the main verb. The structure has the symbol `V` as one of its leaf nodes; this is the position of the main verb. It includes higher projections of the verb such as `VP`s, `S`s, and `SBAR`s. It also includes leaf nodes `NP-A` in positions corresponding to noun-phrase arguments (e.g., the subject or object) of the main verb. In addition, it may contain leaf nodes labeled with categories such as `WHNP` or `WHADVP` where a *wh*-phrase may be placed. It may include leaf nodes corresponding to one or more complementizers (common examples being *that*, *if*, *so that*, and so on).

VOICE: One of two alternatives, `active` or `passive`, specifying the voice of the main verb.

SUBJECT: This variable can be one of three types. If there is no subject position in the `SPINE` variable, then the value for `SUBJECT` is `NULL`. Otherwise, `SUBJECT` can either be a string, for example *there*,[4] or an index of one of the n modifiers in the German clause.

OBJECT: This variable is similar to `SUBJECT`, and can also take three types: `NULL`, a specific string, or an index of one of the n German modifiers. It is always `NULL` if there is no object position in the `SPINE`; it can never be a modifier index that has already been assigned to `SUBJECT`.

[4]This happens in the case where there exists a subject in the English clause but not in the German clause. See, for instance, the second example in Figure 5-1.

WH: This variable is always `NULL` if there is no *wh*-phrase position within the `SPINE`; it is always a non-empty string (such as *which*, or *in which*) if a *wh*-phrase position does exist.

MODALS: This is a string of verbs that constitute the modals that appear within the clause. We use `NULL` to signify that there are no modals.

INFL: The inflected form of the verb.

MOD(i): There are n modifier variables `MOD(1), MOD(2), ..., MOD(n)` that specify the positions for German arguments that have not already been assigned to the `SUBJECT` or `OBJECT` positions in the spine. Each variable `MOD(i)` can take one of five possible values:

- `null`: This value is chosen if and only if the modifier has already been assigned to the subject or object position.
- `deleted`: This means that a translation of the i'th German modifier is not present in the English clause.
- `pre-sub`: The modifier appears after any complementizers or *wh*-phrases, but before the subject of the English clause.
- `post-sub`: The modifier appears after the subject of the English clause, but before the modals.
- `in-modals`: The modifier appears after the first modal in the sequence of modals, but before the second modal or the main verb.
- `post-verb`: The modifier appears somewhere after the main verb.

5.3 Extracting AEPs from a Corpus

A crucial step in our approach is the extraction of training examples from a parsed translation corpus. Each training example consists of a parsed German clause paired with an English AEP (see Figure 5-1). In our experiments, we used the Europarl bilingual parallel corpus (Koehn, 2005a), parsed it, and extracted German clauses and English AEP from the parses. For each sentence pair from the original data, we used a version of the German

German Clause	English AEP
s-oc kous-cp daß np-sb 1 art das nn haupthemmnis np-pd 2 art der adja vorhersehbare nn widerstand np-ag art der nn hersteller vafin-hd war Paraphrase: *that [np-sb the main obstacle] [np-pd the predictable resistance of manufacturers] was*	STEM: be SPINE: SBAR-A IN that S NP-A VP V NP-A VOICE: active SUBJECT: 1 OBJECT: 2 WH: NULL MODALS: has INFL: been MOD1: null MOD2: null
s pp-mo 1 appr zwischen piat beiden nn gesetzen vvfin-hd bestehen adv-mo 2 also np-sb 3 adja erhebliche adja rechtliche $, , adja praktische kon und adja wirtschaftliche nn unterschiede Paraphrase: *[pp-mo between the two pieces of legislation] exist so [np-sb significant legal, practical and economic differences]*	STEM: be SPINE: S NP-A VP V NP-A VOICE: active SUBJECT: “there” OBJECT: 3 WH: NULL MODALS: NULL INFL: are MOD1: post-verb MOD2: pre-sub MOD3: null
s-rc prels-sb die vp pp-mo 1 appr an pdat jenem nn tag pp-mo 2 appr in ne tschernobyl vvpp-hd gezündet vafin-hd wurde Paraphrase: *which [pp-mo on that day] [pp-mo in chernobyl] released were*	STEM: release SPINE: SBAR WHNP SG-A VP V VOICE: passive SUBJECT: NULL OBJECT: NULL WH: which MODALS: was INFL: released MOD1: post-verb MOD2: post-verb

Figure 5-1: Three examples of German parse trees, together with their aligned extended projections (AEPs) in the training data.

parser described in (Dubey, 2005) to parse the German component, and a version of the English parser described in (Collins, 1999) to parse the English component. To extract AEPs, we first align all German and English NPs and PPs. We then use these alignments to constrain the alignment of clauses. After inducing an alignment at the clause level, we extract AEPs from the English clauses. We explain these steps in further detail below.

NP and PP Alignment To align NPs and PPs, first all German and English nouns, personal and possessive pronouns, numbers, and adjectives are identified in each sentence and aligned using GIZA++ (Och & Ney, 2003). Next, each NP in an English tree is aligned to an NP or PP in the corresponding German tree in a way that is *consistent* with the word-alignment information. That is, the words dominated by the English node must be aligned only to words dominated by the German node, and vice versa. Note that if there is more than one German node that is consistent, then the one rooted at the minimal subtree is selected.

Clause alignment, and AEP Extraction The next step in the extraction process is to identify German/English clause pairs. We first break each English or German parse tree into a set of clauses (see Appendix B for a description of how we identify clauses). We retain only those training examples where the English and German sentences have the same number of clauses. For these retained examples, we say the English sentence contains the clause sequence $\langle e_1, e_2, \ldots, e_n \rangle$, and the German sentence contains the clause sequence $\langle g_1, g_2, \ldots, g_n \rangle$. The clauses are ordered according to the position of their main verbs in the original sentence. We create n candidate pairs $\langle (e_1, g_1), (e_2, g_2), \ldots, (e_n, g_n) \rangle$ (i.e., force a one-to-one correspondence between the two clause sequences). We then discard any clause pairs (e, g) which are inconsistent with the NP/PP alignments for that sentence.[5]

Note that this method is deliberately conservative (i.e., high precision, but lower recall), in that it discards sentence pairs where the English/German sentences have different numbers of clauses. In practice, we have found that the method yields a large number of training examples, and that these training examples are of relatively high quality. Future work may consider improved methods for identifying clause pairs, for example methods that make use of labeled training examples.

[5] A clause pair is inconsistent with the NP/PP alignments if it contains an NP/PP on either the German or English side which is aligned to another NP/PP which is not within the clause pair.

Once we have clause pairs, an AEP can be extracted. The EP for the clause is first extracted, giving values for all variables except for SUBJECT, OBJECT, and MOD(1), ..., MOD(n). The values for the SUBJECT, OBJECT, and MOD(i) variables are derived from the alignments between NPs/PPs, and an alignment of other clauses (ADVPs, ADJPs, etc.) derived from GIZA++ alignments. If the English clause has a subject or object which is not aligned to a German modifier, then the value for SUBJECT or OBJECT is taken to be the full English string.

5.4 The Model

5.4.1 Beam search and the perceptron

In this section we describe linear history-based models with beam search, and the perceptron algorithm for learning these models. These methods will form the basis for our model that maps German clauses to AEPs.

We have a training set of n examples, (x_i, y_i) for $i = 1 \ldots n$, where each x_i is a German parse tree, and each y_i is an AEP. We follow previous work on history-based models, by representing each y_i as a series of N decisions $\langle d_1, d_2, \ldots d_N \rangle$. In our approach, N will be a fixed number for any input x: we take the N decisions to correspond to the sequence of variables STEM, SPINE, ..., MOD(1), MOD(2), ..., MOD(n) described in section 5.2. Each d_i is a member of a set $\mathcal{D}_i$ which specifies the set of allowable decisions at the i'th point (for example, $\mathcal{D}_2$ would be the set of all possible values for SPINE). We assume a function ADVANCE$(x, \langle d_1, d_2, \ldots, d_{i-1} \rangle)$ which maps an input x together with a prefix of decisions $d_1 \ldots d_{i-1}$ to a subset of $\mathcal{D}_i$. ADVANCE is a function that specifies which decisions are allowable for a past history $\langle d_1, \ldots, d_{i-1} \rangle$ and an input x. In our case the ADVANCE function implements hard constraints on AEPs (for example, the constraint that the SUBJECT variable must be NULL if no subject position exists in the SPINE). For any input x, a *well-formed* decision sequence for x is a sequence $\langle d_1, \ldots, d_N \rangle$ such that for $i = 1 \ldots n$, $d_i \in$ ADVANCE$(x, \langle d_1, \ldots, d_{i-1} \rangle)$. We define GEN$(x)$ to be the set of all decision sequences (or AEPs) which are well-formed for x.

The model that we will use is a discriminatively-trained, feature-based model. A significant advantage to feature-based models is their flexibility: it is very easy to sensitize the model to dependencies in the data by encoding new features. To define a feature-based

model, we assume a function $\mathbf{f}(x, \langle d_1, \ldots, d_{i-1} \rangle, d_i) \in \mathbb{R}^d$ which maps a decision d_i in context $(x, \langle d_1, \ldots, d_{i-1} \rangle)$ to a *feature vector*. We also assume a vector $\mathbf{w} \in \mathbb{R}^d$ of parameter values. We define the *score* for any partial or complete decision sequence $y = \langle d_1, d_2, \ldots, d_m \rangle$ paired with x as:

$$\text{SCORE}(x, y) = \mathbf{f}(x, y) \cdot \mathbf{w} \tag{5.1}$$

where $\mathbf{f}(x, y) = \sum_{i=1}^{m} \mathbf{f}(x, \langle d_1, \ldots, d_{i-1} \rangle, d_i)$. In particular, given the definitions above, the output structure $F(x)$ for an input x is the highest–scoring well–formed structure for x:

$$F(x) = \arg \max_{y \in \text{GEN}(x)} \text{SCORE}(x, y) \tag{5.2}$$

To decode with the model we use a beam-search method. The method incrementally builds an AEP in the decision order $d_1, d_2, \ldots, d_N$. At each point, a beam contains the top M highest–scoring partial paths for the first m decisions, where M is taken to be a fixed number. The score for any partial path is defined in Equation 5.1. The ADVANCE function is used to specify the set of possible decisions that can extend any given path in the beam.

To train the model, we use the averaged perceptron algorithm described in (Collins, 2002). This combination of the perceptron algorithm with beam-search is similar to that described in (Collins & Roark, 2004).[6] The perceptron algorithm is a convenient choice because it converges quickly — usually taking only a few iterations over the training set (Collins, 2002), (Collins & Roark, 2004).

5.4.2 The Features of the Model

The model's features allow it to capture dependencies between the AEP and the German clause, as well as dependencies between different parts of the AEP itself. The features included in $\mathbf{f}$ can consist of any function of the decision history $\langle d_1, \ldots, d_{i-1} \rangle$, the current decision d_i, or the German clause. In defining features over AEP/clause pairs, we make use of some basic functions which look at the German clause and the AEP (see Tables 5.1 and 5.2). We use various combinations of these basic functions in the prediction of each decision d_i, as described below.

[6] Future work may consider alternative algorithms, such as those described in (Daumé & Marcu, 2005).

1	main verb
2	any verb in the clause
3	all verbs, in sequence
4	spine
5	tree
6	preterminal label of left-most child of subject
7	terminal label of left-most child of subject
8	suffix of terminal label of right-most child of subject
9	preterminal label of left-most child of object
10	terminal label of left-most child of object
11	suffix of terminal label of right-most child of object
12	preterminal label of the negation word *nicht* (*not*)
13	is either of the strings *es gibt* (*there is/are*) or *es gab* (*there was/were*) present?
14	complementizers and wh-words
15	labels of all wh-nonterminals
16	terminal labels of all wh-words
17	preterminal label of a verb in first position
18	terminal label of a verb in first position
19	terminal labels of all words in any relative pronoun under a PP
20	are all of the verbs at the end?
21	nonterminal label of the root of the tree
22	terminal labels of all words constituting the subject
23	terminal labels of all words constituting the object
24	the leaves dominated by each node in the tree
25	each node in the context of a CFG rule
26	each node in the context of the RHS of a CFG rule
27	each node with its left and right sibling
28	the number of leaves dominated by each node in the tree

Table 5.1: Functions of the German clause used for making features in the AEP prediction model.

1	does the SPINE have a subject?
2	does the SPINE have an object?
3	does the SPINE have any wh-words?
4	the labels of any complementizer nonterminals in the SPINE
5	the labels of any wh-nonterminals in the SPINE
6	the nonterminal labels SQ or SBARQ in the SPINE
7	the nonterminal label of the root of the SPINE
8	the grammatical category of the finite verbal form INFL (i.e., infinitive, 1st-, 2nd-, or 3rd-person pres, pres participle, sing past, plur past, past participle)

Table 5.2: Functions of the English AEP used for making features in the AEP prediction model.

STEM: Features for the prediction of STEM conjoin the value of this variable with each of the functions in lines 1–13 of Table 5.1. For example, one feature is the value of STEM conjoined with the main verb of the German clause. In addition, **f** includes features sensitive to the rank of a candidate stem in an externally-compiled lexicon.[7]

SPINE: Spine prediction features make use of the values of the variables SPINE and STEM from the AEP, as well as functions of the spine in lines 1–7 of Table 5.2, conjoined in various ways with the functions in lines 4, 12, and 14–21 of Table 5.1. Note that the functions in Table 5.2 allow us to look at substructure in the spine. For instance, one of the features for SPINE is the label SBARQ or SQ, if it exists in the candidate spine, conjoined with a verbal preterminal label if there is a verb in the first position of the German clause. This feature captures the fact that German yes/no questions begin with a verb in the first position.

VOICE: Voice features in general combine values of VOICE, SPINE, and STEM, with the functions in lines 1–5, 22, and 23 of Table 5.1.

SUBJECT: Features used for subject prediction make use of the AEP variables VOICE and STEM. In addition, if the value of SUBJECT is an index i (see section 5.2), then **f** looks at the nonterminal label of the German node indexed by i as well as the surrounding context in the German clausal tree. Otherwise, **f** looks at the value of SUBJECT. These basic features are combined with the functions in lines 1, 3, and 24–27 of Table 5.1.

[7]The lexicon is derived from GIZA++ and provides, for a large number of German main verbs, a ranked list of possible English translations.

OBJECT: We make similar features to those for the prediction of `SUBJECT`. In addition, **f** can look at the value predicted for `SUBJECT`.

WH: Features for `WH` look at the values of `WH` and `SPINE`, conjoined with the functions in lines 1, 15, and 19 of Table 5.1.

MODALS: For the prediction of `MODALS`, **f** looks at `MODALS`, `SPINE`, and `STEM`, conjoined with the functions in lines 2–5 and 12 of Table 5.1.

INFL: The features for `INFL` include the values of `INFL`, `MODALS`, and `SUBJECT`, and `VOICE`, and the function in line 8 of Table 5.2.

MOD(i): For the `MOD(i)` variables, **f** looks at the value of `MODALS`, `SPINE` and the current `MOD(i)`, as well as the nonterminal label of the root node of the German modifier being placed, and the functions in lines 24 and 28 of Table 5.1.

5.5 Deriving Full Translations

As we described in Section 5.1.1, the translation of a full German sentence proceeds in a series of steps: a German parse tree is broken into a sequence of clauses; each clause is individually translated; and finally, the clause-level translations are combined to form the translation for a full sentence. The first and last steps are relatively straightforward. We now show how the second step is achieved — i.e., how AEPs can be used to derive English clause translations from German clauses.

We will again use the following translation pair as an example: *daß das haupthemmnis der vorhersehbare widerstand der hersteller war./that the main obstacle has been the predictable resistance of manufacturers.*

First, an AEP like the one at the top of Figure 5-1 is predicted. Then, for each German modifier which does not have the value `deleted`, an English translation is predicted. In the example, the modifiers *das haupthemmnis* and *der vorhersehbare widerstand der hersteller* would be translated to *the main obstacle*, and *the predictable resistance of manufacturers*, respectively.

A number of methods could be used for translation of the modifiers. In this chapter, we use the phrase-based system of (Koehn et al., 2003) to generate n-best translations for each of the modifiers, and we then use a discriminative reranking algorithm (Bartlett et al., 2004) to choose between these modifiers. The features in the reranking model can be sensitive to various properties of the candidate English translation, for example the words, the part-of-speech sequence or the parse tree for the string. The reranker can also take into account the original German string. Finally, the features can be sensitive to properties of the AEP, such as the main verb or the position in which the modifier appears (e.g., subject, object, `pre-sub`, `post-verb`, etc.) in the English clause. See Appendix C for a full description of the features used in the modifier translation model. The reranking stage allows us to filter translation candidates which do not fit syntactically with the position in the English tree. For example, we can parse the members of the n-best list, and then learn a feature which strongly disprefers prepositional phrases if the modifier appears in subject position.

Finally, the full string is predicted. In our example, the AEP variables `SPINE`, `MODALS`, and `INFL` in Figure 5-1 give the ordering `<that SUBJECT has been OBJECT>`. The AEP and modifier translations would be combined to give the final English string. In general, any modifiers assigned to the `pre-sub`, `post-sub`, `in-modals` or `post-verb` are placed in the corresponding position within the spine. For example, the second AEP in Figure 5-1 has a spine with ordering `<SUBJECT are OBJECT>`; modifiers 1 and 2 would be placed in positions `pre-sub` and `post-verb`, giving the ordering `<MOD2 SUBJECT are OBJECT MOD1>`. All modifiers assigned the `post-verb` position are placed after the object. If multiple modifiers appear in the same position (e.g., `post-verb`), then they are placed in the order seen in the original German clause.

5.6 Experiments

We applied the approach to translation from German to English, using the Europarl corpus (Koehn, 2005a) for our training data. This corpus contains over 750,000 training sentences; we extracted over 441,000 training examples for the AEP model from this corpus, using the method described in section 5.3. We reserved 35,000 of these training examples as development data for the model. We used a set of features derived from the those described in section 5.4.2. This set was optimized using the development data through experimentation

with several different feature subsets.

Modifiers within German clauses were translated using a phrase-based model (Koehn et al., 2003). We first generated n-best lists for each modifier. We then built a reranking model — see section 5.5 — to choose between the elements in the n-best lists. The reranker was trained using around 800 labeled examples from a development set.

The test data for the experiments consisted of 2,000 sentences, and was the same test set as that used by (Collins et al., 2005). We use the model of (Koehn et al., 2003) as a baseline for our experiments. The AEP-driven model was used to translate all test set sentences where all clauses within the German parse tree contained at least one verb and there was no embedding of clauses — there were 1,335 sentences which met these criteria. The remaining 665 sentences were translated with the baseline system. This set of 2,000 translations had a BLEU score of 23.96. The baseline system alone achieved a BLEU score of 25.26 on the same set of 2,000 test sentences. We also obtained judgments from two human annotators on 100 randomly-drawn sentences on which the baseline and AEP-based outputs differed. For each example the annotator viewed the reference translation, together with the two systems' translations presented in a random order. Annotator 1 judged 62 translations to be equal in quality, 16 translations to be better under the AEP system, and 22 to be better for the baseline system. Annotator 2 judged 37 translations to be equal in quality, 32 to be better under the baseline, and 31 to be better under the AEP-based system.

Chapter 6

Machine Translation with Lattices

In this chapter, we present an alternative lattice-based framework for deriving full translations from AEPs. The new framework has three important advantages over the one presented in Chapter 5: 1. it accommodates n-best lists of AEPs; 2. it allows for the use of a target language model; and 3. it can select argument and modifier translations in context. In the first part of the chapter, we describe the new lattice-based framework. In the second part, we present experimental results on a German-to-English translation task. Finally, in the last part, we give an analysis of our system's performance.

6.1 The Lattice-Based Framework

The lattice-based framework can be described in the following sequence of steps:

1. parse the input and break it into a series of clauses;
2. predict n-best AEPs for each clause;
3. generate translations for subjects, objects, and modifiers;
4. construct a sentence-level lattice of possible translations;
5. select a full translation by searching the lattice.

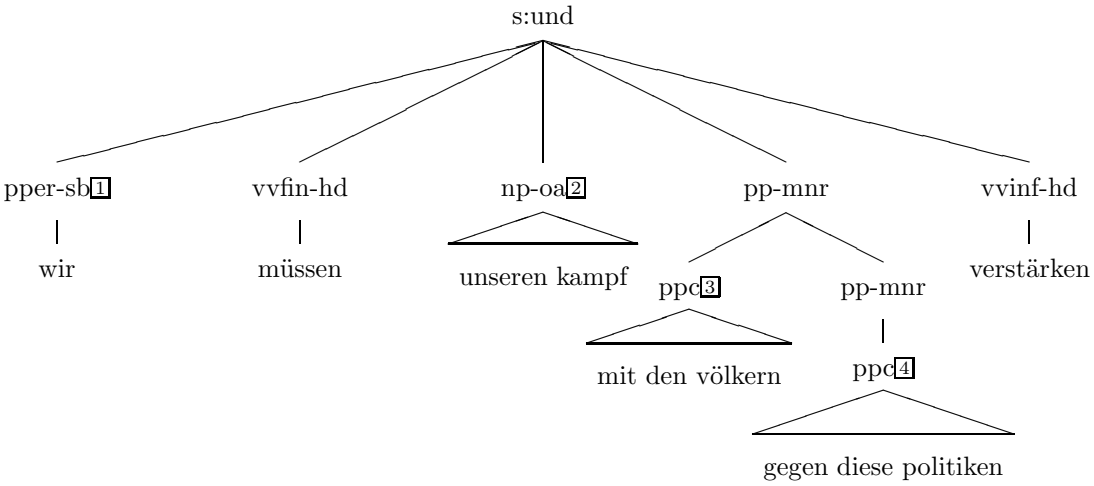
In this section, we provide an example to demonstrate each one of these steps. Throughout the example, we consider the input to be the following German sentence:

GERMAN: wir sind zornig und wir müssen unseren kampf mit den völkern gegen diese politiken verstärken .

GLOSS: we are angry and we must our fight with the people against these policies extend .

REFERENCE: we are angry and we must extend our fight with the peoples against these policies .

Step 1: Parse input and break into clauses This step is identical to the first step in Chapter 5. The complete German input is parsed and then broken into a series of clauses. In our example there are two clauses, *wir sind zornig* and *und wir müssen....* The second clause has the following parse structure:



Step 2: Predict AEPs for each clause In this step, we generate an n-best list of AEPs for each clause. The AEP prediction model is the same as that described in Chapter 5, except that it has been modified to produce an n-best list of AEPs rather than the single best AEP. This modification is simple to implement since the AEP prediction model's search maintains a beam of the top-scoring sequences of decisions, as described in Section 5.4.

The following list of three top AEPs corresponds to the second clause in the example:

STEM: extend		STEM: strengthen		STEM: extend	
SPINE:		SPINE:		SPINE:	
S:and S NP-A		S:and S NP-A		S:and S NP-A	
VP V		VP V		VP V	
NP-A		NP-A		NP-A	
VOICE:	active	VOICE:	active	VOICE:	active
SUBJECT:	1	SUBJECT:	1	SUBJECT:	1
OBJECT:	2	OBJECT:	2	OBJECT:	2
WH:	NULL	WH:	NULL	WH:	NULL
MODALS:	must	MODALS:	must	MODALS:	are
INFL:	extend	INFL:	strengthen	INFL:	extending
MOD1:	null	MOD1:	null	MOD1:	null
MOD2:	null	MOD2:	null	MOD2:	null
MOD3:	post-verb	MOD3:	post-verb	MOD3:	post-verb
MOD4:	post-verb	MOD4:	pre-subj	MOD4:	post-verb

Step 3: Generate modifier translations A lattice of possible translations is generated for each modifier. In the second clause of our example there are four modifiers: the subject *wir*, the object *unsere kampf*, and the prepositional phrases *mit den völkern* and *gegen diese politiken*. A simplified lattice is shown at the bottom of Figure 6-1 for the modifier *mit den völkern*. This lattice represents the phrases *with the citizens*, *with the nations*, *with the people*, *with the peoples*, *by the people*, *by the peoples*, *of the people*, and *of the peoples*. Note that the lattices we use contain both words and scores on the arcs, while the schematic shows only words.

Step 4: Construct sentence-level lattice A sentence-level lattice is constructed using the AEPs from Step 2 and the modifier lattices from Step 3. First, lattices are constructed for each AEP in the n-best list by concatenating the modifier lattices with the main verb and any conjunctions, complementizers, or *wh*-words in the AEP. The subject and object modifier lattices are placed before and after the verb, respectively; the remaining modifier lattices are placed according to the positioning designated by the AEP. The schematic labelled *AEP 3 lattice*, in the middle of Figure 6-1, represents an AEP lattice for the third AEP shown above.

Once each of the AEP lattices have been formed, they are combined using the union operation for finite-state machines. This operation is illustrated at the top of Figure 6-1.

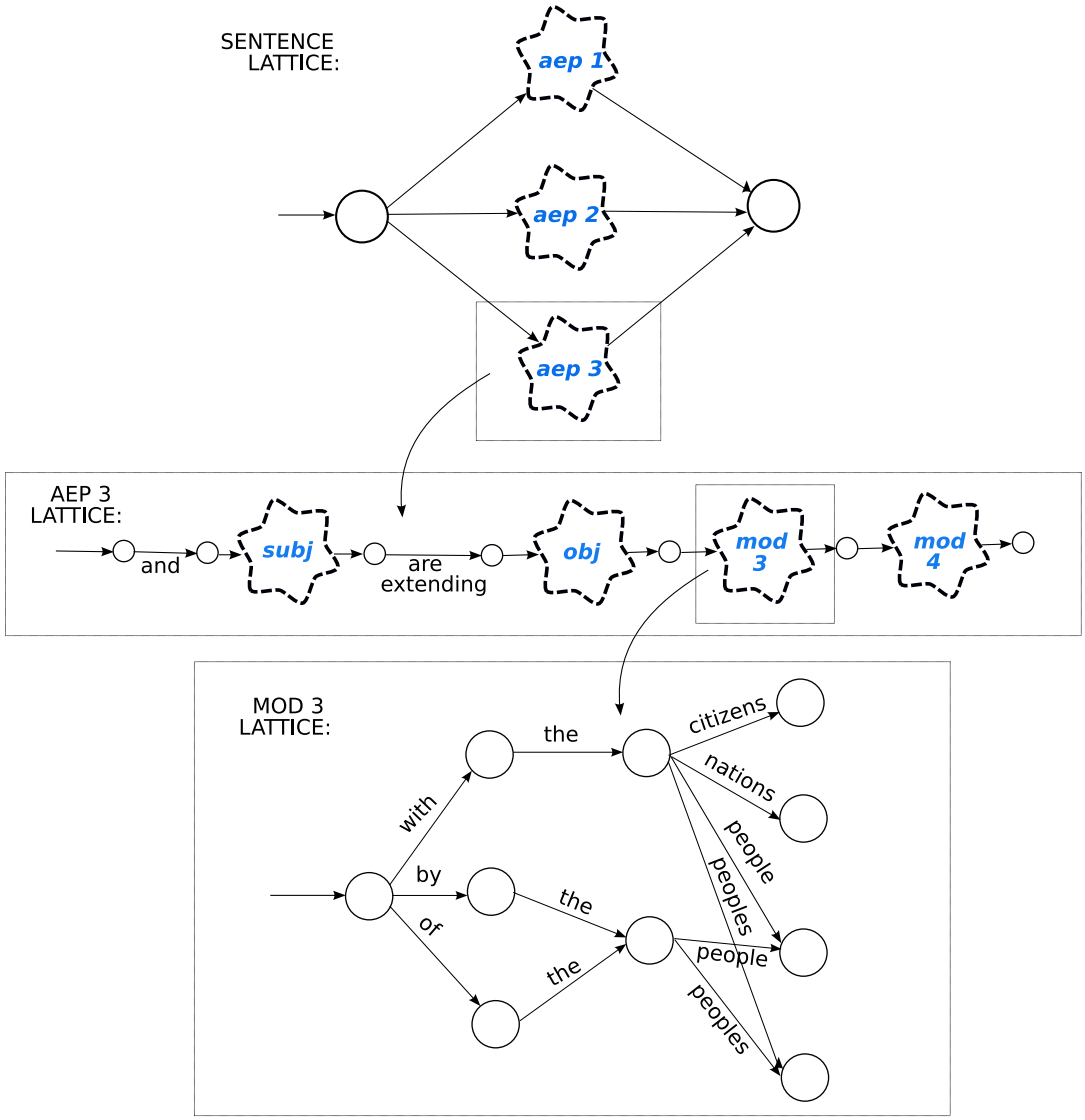


Figure 6-1: A sentence lattice (top) is constructed from n-best AEP lattices. Each AEP lattice (middle) consists of the different pieces of the AEP, concatenated together. Arguments and modifier translation candidates are themselves represented by lattices (bottom).

The final step in constructing the sentence-level lattice is to combine it with a language model using the composition operation.

Step 5: Select translation Finally, a translation is selected by searching for the best possible path through the lattice, as determined by the scores on the arcs. In Section 6.2 we describe how we compute these scores. The lattice can be searched using the Viterbi algorithm.

6.2 Experiments

6.2.1 Data

We tested the lattice-based model on a German-to-English translation task. Our model was trained and tested using the Europarl corpus (version 2) of (Koehn, 2005b). The data was partitioned into four segments: TRAIN/DEV1 (751,088 sentence pairs),[1] DEV2 (2000 sentence pairs), and TEST (7718 sentence pairs). The partitioning is summarized in Figure 6-2.

TRAIN/DEV1, taken from the Europarl V2 standard training set, was used to train the n-best AEP model: the first 701,014 sentences were used to extract 411,634 AEP-German parse tree pairs to train the model; the remaining examples were used to generate 30K AEP-German parse tree pairs for use as development data (see Figure 6-3).[2]

DEV2 and TEST were both extracted from the Europarl V2 standard test data (i.e., files in the range 10/00–12/00).[3] DEV2 was used as development data to train weights for the lattice search. TEST was used to test the lattice-based framework. For both of these sets, we only included German sentences where the parser of (Dubey, 2005) produced a parse, and where the German clauses involved no embedding. After this filtering step, DEV2 contained 1494 sentences, and TEST contained 4694 sentences (see Figure 6-4).

TRAIN	DEV1	DEV2	TEST
700K	50K	2K	8K
EUROPARL V2 train		EUROPARL V2 test	

Figure 6-2: The data was partitioned into four segments of roughly 700K, 50K, 2K, and 8K. TRAIN and DEV1 were taken from the standard Europarl version 2 training set; DEV2 and TEST were taken from the standard test set (10/00–12/00). DEV2 consists of the first 2000 examples from files 10/2/00–10/6/00; TEST is the remaining 8000 examples.

TRAIN	DEV1
411K	30K

Figure 6-3: Approximate number of English AEP-German clause pairs in TRAIN and DEV1.

6.2.2 From German Sentences to English Translations

The full experimental path from German sentences to English translations is shown in Figure 6-5. The first step is to train an n-best AEP model (top) using data in TRAIN and DEV1. We ran the averaged perceptron trainer described in Section 5.4 for ten iterations and selected the model with the smallest number of total errors (where errors are summed over all decisions).

The next step is to generate English translations from German sentences in TEST (middle and bottom of Figure 6-5). We used n-best AEP model of the first step to generate n-best AEPs for the TEST German sentences, which were first parsed with (Dubey, 2005) and broken into clauses. To translate the arguments and modifiers of the n-best AEPs, we used the phrase-based model of (Koehn et al., 2003) trained on the first 727,770 examples from the TRAIN/DEV1 set. We used a different trigram language model with this system[4] according to the type of the argument or modifier being translated. That is, we selected one of six trigram language models, each of which was trained on examples of the appropriate argument type (subject or object) or modifier type (pre-subject, post-subject, in-modals, or post-verb). The arguments and modifiers used to train the language models were extracted from the parse trees of the 727,770 examples used to train the system. The position-specific translation candidates were stored as lattices and used as building blocks in the clause and

[1]This is the same 751,088 used to train the phrase-based and AEP-based systems in Chapter 5.

[2]The lexicon used in the AEP model's feature set was trained on all 751,088 sentences in TRAIN/DEV1.

[3]Before extracting DEV2 and TEST from the Europarl V2 test data, we preprocessed it in three steps: 1. we filtered out XML lines; 2. we filtered out any line that appeared in the test set used in Section 5.6; and 3. we filtered out 10 lines from the file dated 12/13/2000, which appeared as part of a speech we found reiterated in part of the training data (file 06/14/2000).

[4]This phrase-based system includes a language model as one of its features.

DEV2	TEST
1500	4700

Figure 6-4: Approximate number of sentence pairs in DEV2 and TEST after filtering.

sentence-level lattices (as described in Section 6.1).

There are a few details worth mentioning regarding the construction of the lattices:

- When building clause lattices, we first filtered the n-best list of AEPs: we kept only those that had the same modifiers placed and deleted as the top-ranked AEP. This ensured that the scores of the parallel clause paths in the sentence lattice were comparable. Then we took the top five AEPs in the resulting list and created lattices for them.

- For any AEP lattice, if there was more than one modifier in any of the possible modifier positions, we concatenated them in the same order they appeared in the German sentence.

- If there was more than one clause in the original German sentence, we concatenated the corresponding clause-level lattices in the same order of their German correspondents.

- The sentence-level lattice was intersected with a trigram language model trained on the same 727,770 sentences that were used to train the phrase-based modifier translators.

To extract a translation from the sentence lattice, we used the Viterbi algorithm for finite-state machines. The scores on the edges of the sentence lattices were a linear combination of two component scores: a language model score and a word insertion penalty. The weights on these scores were trained using the examples in DEV2 and the simplex algorithm (Nelder & Mead, 1964).

6.3 Results

We compared output from the lattice-based model to output from a phrase-based model (Koehn et al., 2003) trained on the first 727,770 sentences from the TRAIN/DEV1 set described above in Section 6.2.[5] Output for both systems was generated using the final

[5]All BLEU scores and results from the human evaluation in this section were derived using output from this phrase-based system. A phrase-based system trained on all 751,088 sentences in TRAIN/DEV generates a BLEU score of 22.74 on the final test set.

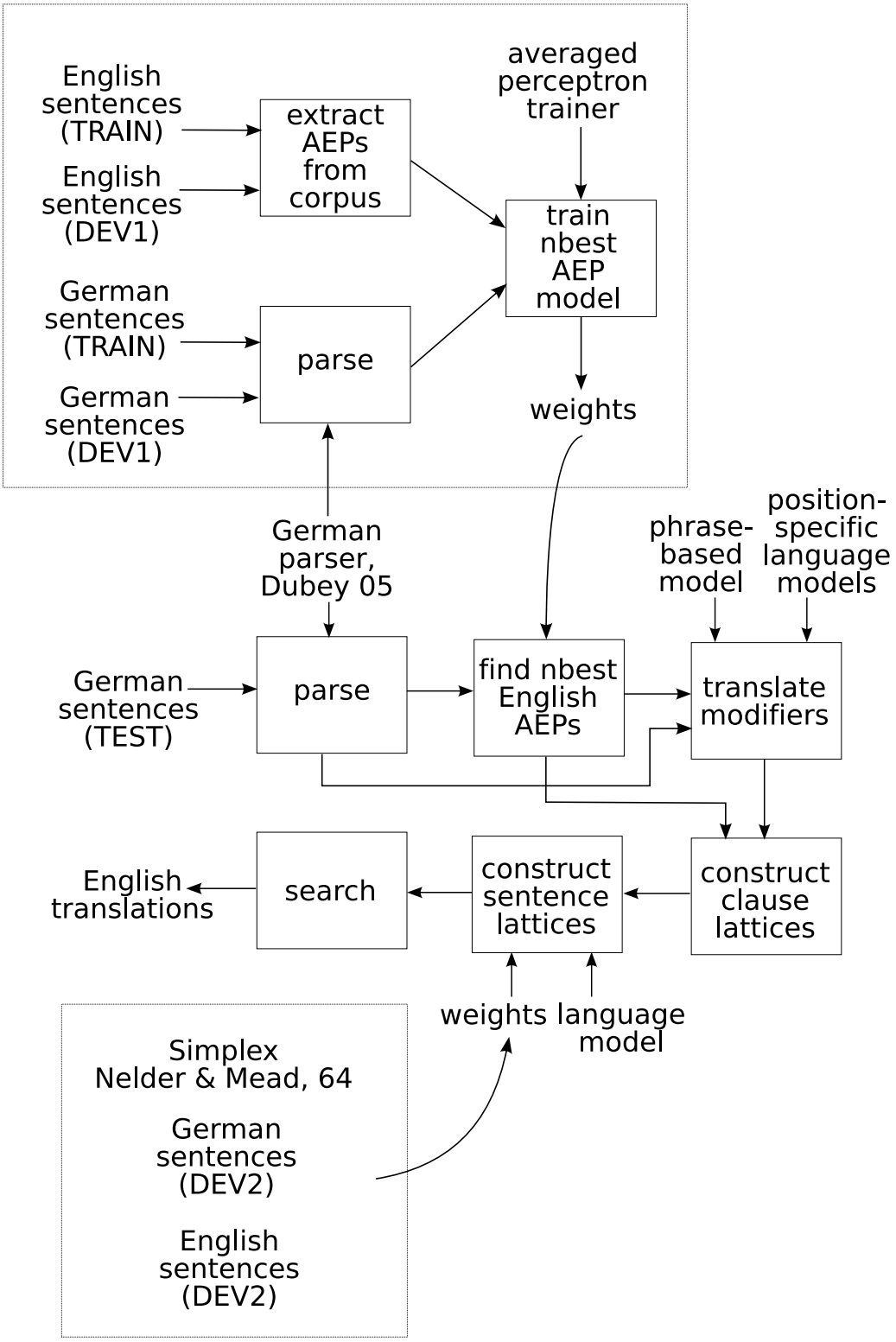


Figure 6-5: The path from German sentences to English translations within the lattice-based frame

	BLEU
PB	22.66
AEP	21.42

Figure 6-6: BLEU scores on test set for both the phrase-based (PB) and the AEP-based (AEP) system.

development and test sets (containing 1494 and 4694 sentences, respectively) that resulted following the filtering regimen described above in Section 6.2. On the test set, the lattice-based system achieved a BLEU score of 21.42 and the phrase-based system a score of 22.66. These results are summarized in Figure 6-6.

6.3.1 Human Evaluation

We conducted a human evaluation to compare the fluency and adequacy of our system's output and a phrase-based system's output on the test set. Fluency refers to the degree to which a translation is well-formed according to the rules of standard written English. A fluent sentence is one that is well-formed grammatically, contains correct spellings, adheres to common use of terms, titles, and names, is intuitively acceptable and can be sensibly interpreted by a native speaker of English.

We asked six native speakers of English to compare 200 translation pairs on the basis of fluency and adequacy. Judges were only shown translations of sentences with length between ten and twenty words (roughly 35% of the test set). The six judges were randomly paired so that there would be two independent judgments of both fluency and adequacy for each of 600 translation pairs. The 600 pairs were randomly ordered before being segmented into chunks of 200. When providing fluency judgments, the annotators were presented with a pair of sentences consisting of two randomly-ordered translations, one from each system. The judges were asked to decide whether the first translation was more fluent, the second was more fluent, or the two were the same. When providing adequacy judgments, the annotators were shown, in addition to the output from the two MT systems, a reference translation. They were asked to decide which translation was better, the first or the second, or whether they were of the same quality, given the reference translation. They were told that an ideal translation should correctly communicate the meaning of the reference translation and should also be fluent/grammatically well-formed. We include the complete instructions given to the judges in Appendix F.

Results of the evaluation are summarized in Figure 6-7. The top tables show the correlation between pair 1, annotator one and pair 1, annotator two's judgments for fluency (left) and adequacy (right). The middle tables are the results for pair 2, and the bottom tables are pair three. Overall, averaged over all judges, they found the output of the AEP-based system to be more fluent than the phrase-based system's in 45% of the translation pairs; in 29% of cases they thought it was worse, and in 26% they thought the two were the same. When judging adequacy, on average the annotators found the AEP-based system's output of higher quality in 36% of cases, of lesser quality in 33%, and of the same quality in 31%.

These averages are more or less in accord with what we found in the judgments of the individual annotators: for fluency, all six preferred the AEP-based system, and for five of the six the difference between the two systems was statistically significant using the sign test (see, for example, (Lehmann, 1986)). For adequacy, only one annotator's preference for the AEP-based system was statistically significant using the sign test; for the remaining five, the difference between the two systems was not statistically significant.

Figure 6-8 shows the correlation between each individual annotator's fluency and adequacy judgments. The top tables are pair 1; the middle ones are pair 2, and the bottom are pair 3.

Figure 6-9 shows the correlation between fluency and adequacy on examples where the two annotators in each pair were in complete agreement. The pair 1 annotators agreed on both fluency and adequacy in 47.5% of cases, the pair 2 annotators in 45.5%, and the pair 3 annotators in 48%. The tables show the strongest correlation along the diagonal, suggesting that fluency and adequacy may be related in people's judgment of translation quality.

6.4 Analysis of Translation Output

In this section, we perform an analysis of the translation output of the lattice-based system. The observations we make are based on our examination of the annotated data resulting from the human evaluation we presented in the preceding section. Before discussing the system's errors, we begin with a discussion of its strengths.

FLUENCY

		A1			
		PB	AEP	=	
A2	PB	0.2	0.065	0.02	**0.285**
	AEP	0.065	0.315	0.035	**0.415**
	=	0.08	0.11	0.11	**0.3**
		0.345	**0.49**	**0.165**	

ADEQUACY

		A1			
		PB	AEP	=	
A2	PB	0.25	0.08	0.025	**0.355**
	AEP	0.02	0.28	0.005	**0.305**
	=	0.12	0.095	0.125	**0.34**
		0.39	**0.455**	**0.155**	

FLUENCY

		A1			
		PB	AEP	=	
A2	PB	0.195	0.045	0.005	**0.245**
	AEP	0.06	0.405	0.035	**0.5**
	=	0.105	0.08	0.07	**0.255**
		0.36	**0.53**	**0.11**	

ADEQUACY

		A1			
		PB	AEP	=	
A2	PB	0.225	0.03	0.02	**0.275**
	AEP	0.045	0.285	0.025	**0.355**
	=	0.145	0.09	0.135	**0.37**
		0.415	**0.405**	**0.18**	

FLUENCY

		A1			
		PB	AEP	=	
A2	PB	0.155	0.015	0.04	**0.21**
	AEP	0.04	0.275	0.07	**0.385**
	=	0.105	0.07	0.23	**0.405**
		0.3	**0.36**	**0.34**	

ADEQUACY

		A1			
		PB	AEP	=	
A2	PB	0.18	0.005	0.045	**0.23**
	AEP	0.04	0.235	0.075	**0.35**
	=	0.1	0.06	0.26	**0.42**
		0.32	**0.3**	**0.38**	

Figure 6-7: Confusion matrices comparing fluency and adequacy judgments of the two annotators in pairs 1 (top), 2 (middle), and 3 (bottom). For each pair, A1 is annotator one, and A2 is annotator two. The nine interior cells show inter-annotator agreement, and the outer six cells summarize the judgments of each individual annotator.

ANNOTATOR 1

		AD		
		PB	AEP	=
FL	PB	0.235	0.065	0.045
	AEP	0.115	0.345	0.03
	=	0.04	0.045	0.08

ANNOTATOR 2

		AD		
		PB	AEP	=
FL	PB	0.215	0.015	0.055
	AEP	0.06	0.245	0.11
	=	0.08	0.045	0.175

ANNOTATOR 1

		AD		
		PB	AEP	=
FL	PB	0.26	0.04	0.06
	AEP	0.11	0.355	0.065
	=	0.045	0.01	0.055

ANNOTATOR 2

		AD		
		PB	AEP	=
FL	PB	0.16	0.015	0.07
	AEP	0.045	0.305	0.15
	=	0.07	0.035	0.15

ANNOTATOR 1

		AD		
		PB	AEP	=
FL	PB	0.215	0.02	0.065
	AEP	0.045	0.235	0.08
	=	0.06	0.045	0.235

ANNOTATOR 2

		AD		
		PB	AEP	=
FL	PB	0.145	0.025	0.04
	AEP	0.015	0.24	0.13
	=	0.07	0.085	0.25

Figure 6-8: The tables show the correlation between fluency (FL) and adequacy (AD) for each annotator. The top two tables are from pair one; the middle are from pair 2, and the bottom are from pair three.

PAIR 1

		AD		
		PB	AEP	=
FL	PB	0.115	0.005	0.005
	AEP	0.005	0.17	0.025
	=	0.01	0.005	0.135

PAIR 2

		AD		
		PB	AEP	=
FL	PB	0.145	0.005	0.01
	AEP	0.01	0.21	0.01
	=	0.005	0.005	0.055

PAIR 3

		AD		
		PB	AEP	=
FL	PB	0.12	0.005	0.02
	AEP	0.015	0.24	0.03
	=	0.01	0.0	0.04

Figure 6-9: Each table shows the correlation between fluency (FL) and adequacy (AD) for only those examples where the annotators in each pair agreed. The proportion in each cell was calculated out of the 200 examples each pair annotated. Pair 1 agreed on 47.5% of the examples, pair 2 on 45.5%, and pair 3 on 48%.

6.4.1 Strengths

Cases when the AEP-based system's output is notably more fluent and more adequate than that of the phrase-based system tend to be when the German word order diverges dramatically from standard English *subject-verb-object* structure. The following is an example:

GERMAN:	ich hoffe , dass wir slowenien in der ersten gruppe der neuen mitglieder begrüßen können .
GLOSS:	i hope , that we slovenia in the first group of new members welcome can .
REFERENCE:	i hope slovenia will be in the first group of new members .
PHRASE-BASED:	i hope that we slovenia in the first group of the new members welcome that fact .
AEP-BASED:	i hope we can welcome slovenia in the first group of the new member states .

In this example, the second clause of the German (beginning *dass wir slowenien...*) follows the basic word order *subject-object-verb*.[6] Furthermore, the object *slowenien* and the verb phrase *begrüßen können* are separated by two prepositional phrases. As we pointed out in Chapter 1, the phrase-based system is likely to rely on a reordering model to move the verb phrase to an appropriate position (between the subject *we* and the object *slovenia*). In this example, that reordering model is too weak to carry out the desired result. As a consequence, the phrase-based system ends up mimicking the word order of the German. In contrast, the AEP-based system correctly orders the verb and its arguments.

Figure 6-10 shows the parse tree for the clause we have been discussing. An interesting property of this tree is that the parser has mistakenly placed the prepositional phrase modifier *in der ersten gruppe...* in a position of dependence on the object *slowenien* instead of the verb *begrüßen*. This means that the discriminative AEP model considers the entire phrase *slowenien in der ersten gruppe...* to be the object. In this particular case, the AEP model is robust to this error, and it has no negative impact on the translation. However, we will see that parsing errors can have a deleterious effect on translation.

[6] It is required for verbs to come at the end of German subordinate clauses, so this is a frequently-occurring phenomenon.

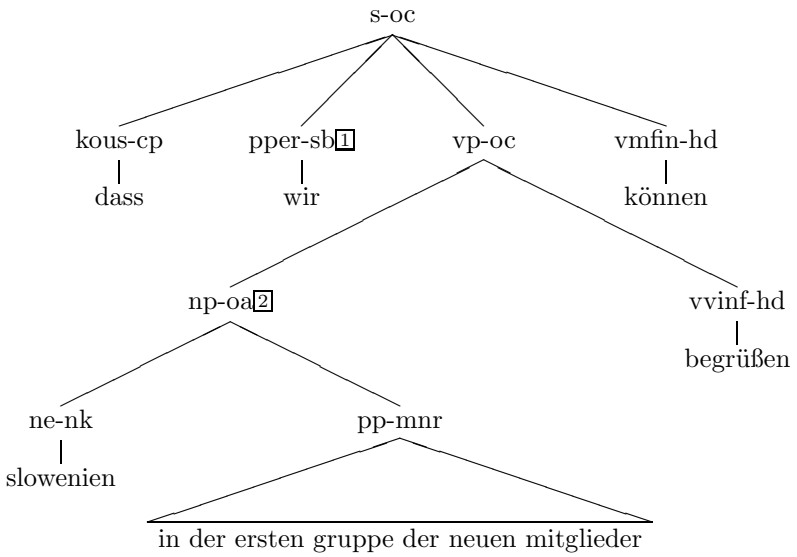


Figure 6-10: Parse of the German clause *dass wir slowenien in der ersten gruppe der neuen mitglieder begrüßen können*. Note that the parser has incorrectly placed the prepositional phrase modifier as a dependent of the object *slowenien*; it should be a dependent of the verb *begrüßen*.

6.4.2 Errors

Based on the human evaluation, the AEP-based system is producing output more fluent than a phrase-based system's roughly 45% of the time; however, the AEP-based system produces more adequate output only 36% of the time. It seems intuitive that more fluent output should also be more adequate. Since our system's output seems to be overall more fluent than the phrase-based system, a natural question to ask is why it is not also more adequate. In addition, the phrase-based system produces more fluent output than the AEP-based system roughly 29% of the time. Since one of the explicit goals of the AEP approach is to improve grammaticality, we would like to understand what is going on in those cases where fluency is still worse than a system that does not explicitly take syntax into account. In the sections that follow, we present some of the errors that our system is making, based on observations we have made in reviewing the data generated during the human evaluation. As we analyze these errors, we pay particularly close attention to phenomena that may help to answer the two questions posed here.

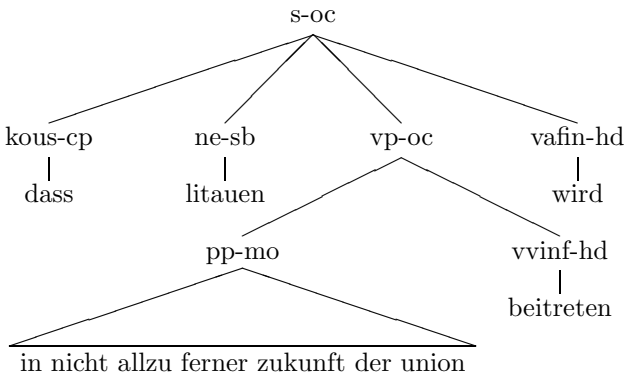


Figure 6-11: Parse of the German clause *dass litauen in nicht allzu ferner zukunft der union beitreten wird*.

Misidentification of Arguments

One problem that appears with some frequency in the AEP-based system's output is the misidentification of arguments (subject and object). For example, in the following case, the AEP prediction model generated at least one top-ranked AEP with subject *they* (filled in from a list of common subjects), and placed the real subject *litauen* in the object position, which ultimately leads to the following translation:

GERMAN:	ich hoffe , dass litauen in nicht allzu ferner zukunft der union beitreten wird .
GLOSS:	i hope , that lithuania in not too distant future the union join will .
REFERENCE:	i look forward to lithuania joining the union in the not too distant future .
PHRASE-BASED:	i hope that lithuania in the not too distant future , join the union .
AEP-BASED:	i hope they will join lithuania in the not too distant future of the european union .

In this particular case, the parse for the second clause where the error occurs does not have any glaring errors (see Figure 6-11), and it seems that the fault lies with the AEP prediction model itself. Parse errors can lead to poor translations, however. Here is an example where a parse error very likely accounts for the obfuscated translation:

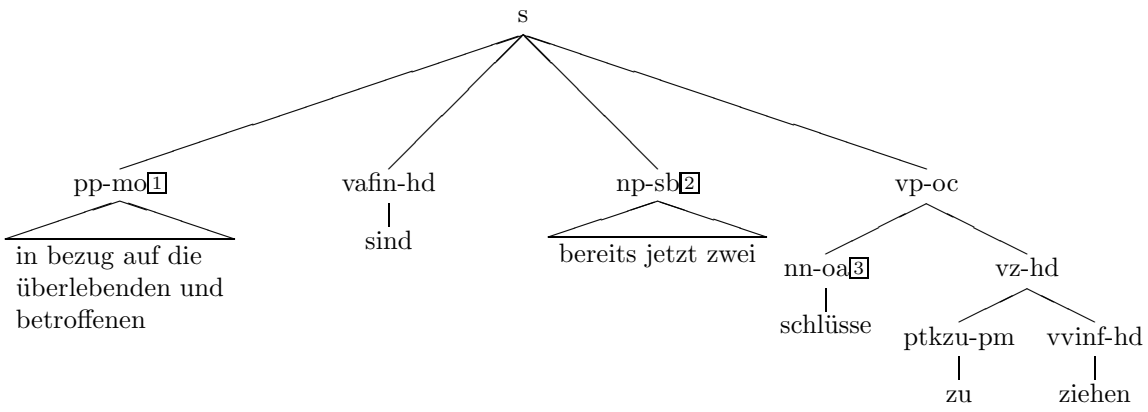


Figure 6-12: Parse of the German sentence *in bezug auf die überlebenden und betroffenen sind bereits jetzt zwei schlüsse zu ziehen.*

GERMAN: in bezug auf die überlebenden und betroffenen sind bereits jetzt zwei schlüsse zu ziehen .

GLOSS: in relation to the survivors and affected are already now two conclusions to draw .

REFERENCE: judging from the survivors and those involved , we can already draw two conclusions .

PHRASE-BASED: in relation to the survivors and affected are already two conclusions .

AEP-BASED: we are required to draw two already in relation to the survivors and those conclusions .

The correct object would be *conclusions*, but the AEP prediction module places the phrase dominated by `np-sb 2` in the object position. At least part of the problem in this particular example is an imperfect parse tree (shown in Figure 6-13), in which the modifier *zwei/two* is separated from its head *schlüsse/conclusions.*

Poor Translations of Arguments and Modifiers

While the lattice-based approach does allow for more contextual information in the selection of argument and modifier translations, it is not perfect: the phrase-based system can introduce errors in producing the n-best list of candidate translations. Here's one example:

GERMAN: durch eine entscheidung der natur ist die frau trägerin der zukunft der menschheit .

GLOSS: by a decision of nature is the woman carrier of the future of humanity .

REFERENCE: nature has determined that woman should carry the future of humanity in her womb .

PHRASE-BASED: by a decision of the nature of the madam indeed a recipient of the future of humanity .

AEP-BASED: the commissioner is indeed a recipient of the future of mankind , by a decision of the environment .

In this case, the AEP-prediction module has correctly predicted that the phrase *die frau* is the subject of the sentence in every AEP in the n-best list. The phrase-based system used to translate *die frau* must then have introduced *the commissioner* as a possible translation, and it was then selected during the lattice search.

Problems Related to Verb Stem Choice

Another source of errors is related to AEP stem prediction. One instance of this problem has to do with suboptimal selection of a stem candidate, resulting in translations such as the following, in which the valence of the verb *summoned* is not quite right with respect to the reference translation *increased*:

GERMAN: der verhandlungsprozess als solcher ist im letzten jahr noch einmal deutlich beschleunigt worden .

GLOSS: the negotiation process as such is the last year again considerably increased been .

REFERENCE: the pace of the negotiating process itself increased considerably again last year .

PHRASE-BASED: the negotiation process , as such is the last year clear once again been speeded up .

AEP-BASED: last year , the process as such , has been summoned once again clearly .

Errors such as this one can have a strong impact on a human's perception of adequacy. Another instance of stem-choice problems occurs when the AEP-prediction module decides

to translate a contentful German verb with an English verbal construction involving a light verb such as *to be*, while dropping the verbal complement:

GERMAN:	was uns allerdings beunruhigt , ist das schicksal von vladimiro montesino .
GLOSS:	what us however worries , is the fate of vladimiro montesino .
REFERENCE:	but we are concerned about the fate of mr montesinos .
PHRASE-BASED:	what we are , however , is concerned , the fate of vladimiro montesino .
AEP-BASED:	what is to us , however , is the fate of vladimiro montesino .

In the next section, we analyze stem prediction and other errors being made during AEP prediction.

6.5 Analysis of AEP Model Output

Section 6.4 identifies three major sources of errors in the lattice-based translation system: parsing errors, AEP-prediction errors, and modifier translation errors (made by the phrase-based system). In this section, we take a closer look at errors made by the AEP prediction model. During training, errors are detected by comparing the value of each decision in the predicted AEP to those in the gold-standard (extracted) AEP. While this is a strict criterion (the predicted AEP's main verb might be synonymous with that of the gold-standard AEP; a verb in the passive voice might work just as well as one in the active voice, etc.), it is no more strict than most supervised MT methods with a single reference translation. That said, it would be useful to know how many predicted AEPs might actually generate reasonable translations, even if they differ from the gold standard.

To this end, we have undertaken an analysis of 200 randomly-selected AEPs from the development set used to train the prediction model (the DEV1 set from Section 6.2). The errors generated by the model during training on the full set are shown in Figure 6-13. Again, these numbers were generated using the strict criterion of an exact match. According to this criterion, 85.3% of the 30,000 AEPs in DEV1 had at least one decision error. Under a different set of criteria, however, roughly 40% have at least one error.

STEM	SPINE	VOICE	SUBJ	OBJ	WH	MODALS	INFL	MOD
48.5%	45.3%	0.09%	28.6%	29.9%	0.08%	39.5%	35.7%	33.0%

Figure 6-13: Percentage of errors by decision. These numbers were taken from the AEP model used to generate n-best AEPs for DEV2 and TEST. The errors were made on the DEV1 set (size 30K), during training.

To derive this number, we use a less stringent set of criteria than the exact match. Under this new set of criteria, an AEP is considered satisfactory if

- it is *formally equivalent* to the gold-standard AEP, or
- it is *functionally equivalent* to the gold-standard AEP, or
- it is *semantically equivalent* to the gold-standard AEP.

The first criterion, formal equivalence, is the same as the exact match criterion of before: a predicted AEP is formally equivalent to its gold-standard AEP if each one of its decision values is an exact match with the gold. With the second criterion, functional equivalence, the two AEPs may have one or more decision values that are different, but these differences have no real effect. For example, the AEPs in Figure 6-14 are functionally equivalent. Semantic equivalence is the most lenient of all the criteria. Two AEPs that are semantically equivalent may differ considerably; however, the predicted AEP is deemed satisfactory in the sense that it could lead to a perfectly valid translation. Semantic equivalence covers cases where

- modifiers are placed in different but equally satisfying positions,
- the main verb is different but synonymous,
- the predicted output is passive and the gold is active (or vice versa) but the two are semantically equivalent,
- the verbal inflection is different but sound,

and so on. The AEPs in Figure 6-15 are semantically equivalent.

Using these less stringent criteria, we evaluated the randomly-generated set of 200 AEP described above. We found 121/200 (60%) of the AEPs to be satisfactory, and 79/200 (40%) to have at least one error. Of these, 18/79 (22.8%) would be satisfactory if the model had

German Clause

s-cc np-oa 1 npb art-oa ein
adja-nk klares
nn-nk signal
pp-mnr appr-ad an
cnp-nk npb nn-nk rat
kon-cd und
nn-nk kommission
vvfin-hd senden

Paraphrase: *[[np-oa a clear signal] [pp-mnr to the council and the commission]] send*

Gold English AEP

STEM: send
SPINE:
SBAR-A IN in
SG-A VP V
NP-A
VOICE: active
SUBJECT: BLANK
OBJECT: 1
WH: NULL
MODALS: BLANK
INFL: sending
MOD1: null

Predicted English AEP

STEM: send
SPINE:
SBAR-A IN in
SG-A VP V
VOICE: active
SUBJECT: BLANK
OBJECT: BLANK
WH: NULL
MODALS: BLANK
INFL: sending
MOD1: post-verb

Figure 6-14: The gold AEP and the predicted AEP are functionally equivalent. The gold AEP predicts an object in the spine and fills it with the German modifier labeled 1; the predicted AEP predicts no object in the spine, but places the modifier labeled 1 after the verb. Strictly speaking, the predicted AEP is incorrect, but functionally it produces the same output as the gold AEP.

German Clause

s np-sb 1 npb art-sb die
nn-nk kommission
vafin-hd hat
pp-mo 2 ppc apprart-da beim
nn-nk lernen
npb 3 nn-oa schwierigkeiten

Paraphrase: *[np-sb the commission] has [pp-mo with learning] [npb difficulties]*

Gold English AEP

STEM: learn
SPINE:
S NP-A
VP V
VOICE: active
SUBJECT: 1
OBJECT: BLANK
WH: NULL
MODALS: BLANK
INFL: learns
MOD1: null
MOD2: delete
MOD3: post-verb

Predicted English AEP

STEM: have
SPINE:
S NP-A
VP V
NP-A
VOICE: active
SUBJECT: 1
OBJECT: 3
WH: NULL
MODALS: BLANK
INFL: has
MOD1: null
MOD2: post-verb
MOD3: null

Figure 6-15: The gold AEP and the predicted AEP are semantically equivalent. The gold AEP maps to the English translation *the commission learns with difficulty*, while the predicted AEP might lead to a translation such as *the commission has difficulty learning*.

chosen a better main verb. These numbers suggest that by focusing on improving the main verb decision, the AEP prediction model could be achieving roughly 69.4% accuracy.

Chapter 7

Conclusion

We have presented an approach to tree-to-tree machine translation which leverages a new representation — the aligned extended projection — within a discriminative, feature-based framework. The AEP approach makes it possible to explicitly model the syntax of the target language. The approach also makes it possible to build a detailed model of the correspondence between the source and target-language parse trees, thereby constructing a translation that preserves the meaning of the source-language sentence.

In the human evaluation of Chapter 6, we have seen evidence for a significant improvement in the grammaticality of the AEP-based system's output when compared with a phrase-based system. We have also seen evidence that the system could be improved with respect to adequacy. The preservation of meaning is a complicated goal that can be thwarted by poor parses, errors in the AEP-prediction model, or inadequate translations of arguments and modifiers. Our hope is that future work may alleviate these problems and help bring the AEP approach closer to its full potential.

7.1 Future Work

The AEP-based framework presents many opportunities for future work. We explore some of these ideas here in the hope that it will be helpful for those wanting to make improvements down the line.

7.1.1 Improved AEP Prediction

Improving AEP prediction could have a significant impact on the overall quality of AEP-based translation. Based on the errors of the current system (see Section 6.5), it is clear that the prediction of particular parts of the AEP could be improved. For instance, we have routinely observed that the choice of the main verb may not be optimal or even correct.

The feature-driven linear structured prediction framework allows a wide range of features to be tested in attempting to improve AEP prediction accuracy. An overhaul of the feature set would be an obvious place to start when thinking about the improvement of any individual AEP decision. Alternatively, one might consider different definitions of AEPs. For example, one might consider AEPs that include larger chunks of phrase structure, or ones that contain more detailed information about the relative ordering of modifiers. Also, not much has been done to test the order in which the decisions are made, or whether certain decisions (e.g., voice, inflection) might be unnecessary or even harmful. Inflection is something that might be better determined during the lattice search.

Something that we didn't mention earlier but that ought to be documented are those systematic bugs in the AEP extraction phase that are propagating through the AEP prediction phase to the final translations:

1. interrogative structures are not being handled correctly in AEP extraction;

2. the prepositions or particles that certain verbs subcategorize for (e.g., *on* in *vote on*, *to* in *subject to*, *account* in *take account*) are not being adequately modeled;

3. there's a bug stemming from errors in GIZA++ alignments that causes gold-standard AEPs to model two subjects (e.g., *we we will vote*);

The result of the first bug is that the word-order of interrogatives is almost always incorrect (e.g., *you do want* instead of *do you want*). This should be fairly simple to fix once the appropriate spines, subjects, etc. are systematically applied to interrogatives during AEP extraction. The second problem might be a bit more tricky. One idea is to identify these prepositions with the verb, such that the verb is selected as *vote on* instead of just *vote*. Another solution would be to create a separate decision that predicts these prepositions. Or, they could be predicted as part of modifier translation, perhaps in a recursive AEP-prediction stage (see Section 7.1.3). The last bug in the list is critical in that, in ad-

dition to producing repeated words in inappropriate places, it may have other inadvertent detrimental effects, like influencing the system's tendency to introduce new subject strings ("hallucinated" subjects) rather than find a good translation for some word in the German input. For instance, the fact that the system sometimes introduces common English subjects such as *the european union* when the true subject is, say, *romania* may be due to an overestimated prior on these "hallucinated" subjects.

Finally, one last way of potentially improving AEP prediction would be to investigate alternative learning methods. For example, a large-margin optimization technique such as the exponentiated gradient approach to structured prediction (Bartlett et al., 2004) might be adapted to the AEP prediction problem. We might also look into improving the method by which training examples are extracted. Our current method involves heuristics for determining modifier alignment; this might be done instead by using the EM algorithm to induce the best alignment, or by annotating examples and using supervised learning techniques.

In approaching any of the suggested improvements above, it would be good to think about what people expect from automatic translations. I.e., what errors are people most sensitive to? What errors can be most tolerated? I don't think these questions have yet been answered.

7.1.2 Better Integration with the Phrase-Based System

The translation framework we have developed in this thesis uses a phrase-based system to produce n-best lists of modifier translations. One way that might improve modifier translation within this framework would be to develop a better integration between the AEP prediction model and the phrase-based system. For instance, rather than producing modifier translations in isolation, we could use the syntax-based system to produce modified MT input to the phrase-based system. Figure 7-1 illustrates how the input to the MT system might be rearranged by the syntax-based AEP model. Note that in step two of the figure, some of the input has been translated to English (the verb sequence *would like to thank*), while some if it has merely been reordered. This allows the phrase-based system to decode the entire AEP at once. This approach is similar to a clause-restructuring approach such as that of (Collins et al., 2005). A different approach would be to build a classifier that selects the best output after both the AEP-based and the phrase-based systems have decoded the

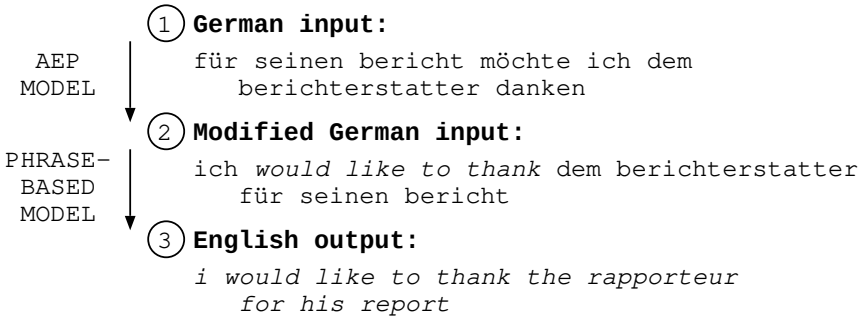


Figure 7-1: The input to the MT system — *für seinen bericht möchte ich dem berichterstatter danken* — is rearranged by the syntax-based system to produced modified input to the phrase-based system.

input.

7.1.3 Recursive Prediction of Modifier AEPs

We might consider methods that relieve the AEP-based translation framework from any reliance whatsoever on a phrase-based system. A natural modification to the current AEP framework would be to apply AEP prediction to the translation of modifiers. The new AEP models would be trained to predict NPs, PPs, ADJPs, and ADVPs. Nouns, adjectives, etc. (like verbs) subcategorize lexically for certain types of arguments (Haegeman & Guéron, 1999), and extended projections may be formed for many types of words. For example, given a German modifier aligned to the English subject, we might try to predict a corresponding AEP, which would likely involve the extended projection of a noun. This approach is compelling not only because it suggests an AEP framework that is entirely self-contained, without reliance on a phrase-based system, but also because it facilitates the modeling of syntactic structure conditioned on the intended role (subject, object, etc) of the target-language modifier.

7.1.4 Better Parsers

The German parser we are currently using (Dubey, 2005) achieves an F1 score of 76.2% in the recovery of labeled brackets and is trained using the Negra German treebank (Skut et al., 1997). In some work by (Kübler et al., 2006), experimentation was conducted with parsers trained using Negra and another treebank TüBa-D/Z (Telljohann et al., 2005); the performance of the parsers trained using Negra was consistently around 20% worse.

The best parser trained with TüBa-D/Z performed at around 89% F1 in the recovery of labeled constituents (compared with 69% for the same parser trained with Negra data) when using gold-standard part-of-speech tags as input. It is therefore possible that we might see a significant improvement in German parsing performance, and hence German/English translation quality, by switching to a parser trained using the TüBa-D/Z treebank.

7.1.5 Spanish/English Translation

One language pair that would be interesting to work with is Spanish/English. (This thesis does develop a Spanish parser, after all!) Spanish is interesting for a number of reasons (Zagona, 2002), some of which are discussed in Chapter 4:

- it has a fairly rich morphology;
- constituent order is not fixed in declarative sentences (constituents may appear as S-V-O, V-O-S, or V-S-O);
- if understood from context, the subject does not have to appear in a sentence;
- the use of the subjunctive mood is quite prevalent and may be triggered by a phrase that appears far away from the inflected verb;
- the passive voice is used in many cases where active voice would be the natural choice in English;
- Spanish uses two forms of past tense (preterite and imperfect) where English uses only one (simple);
- Spanish has two forms of the verb *to be*;
- Spanish uses double clitics (that is, two clitics with the same referent), and there are two possible positions when placing clitics in sentences where a verb appears in the progressive or infinitive form.

7.2 Final Remarks

The goal of this research has been to make explicit use of syntactic information to produce high-quality automatic translations within a discriminative statistical framework. We as-

sume that we have at our disposal a sentence-aligned parallel corpus and automatic parsers for the two languages involved. We believe that using syntax in both the source and target languages is the correct way to approach MT, and as parsers for languages other than English continue to improve, our approach can only benefit.

In this thesis, we have developed methods for automatically extracting and predicting target-language aligned extended projections, which are syntactic structures that contain global information about the output. We have also presented a method for training the AEP prediction model. The results of a human evaluation are very encouraging for the approach, which we believe still holds much untapped potential. We hope that the work in this thesis will lead to future work that not only produces better machine translation output, but also sheds more light on what might be necessary and useful syntactic information in the automatic translation process.

Appendix A

Head Rules For Spanish Parsing

Table A.1 shows the set of head-finding rules used in the Spanish parsing models of Chapter 4. This set of deterministic head rules specifies which child is the head of its parent. A recursive algorithm then propagates headwords (lexical items) and headtags (POS tags) up the tree from the leaves to the root via the head-children (see (Collins, 1999)).

The head rules identify the headchild for each non-terminal. Any non-terminal not in this list takes the left-most child as its head. This is also the default case for all non-terminals listed here (except `*CC2` and `*CP` — where * is a wildcard— which always take the right-most child to be the head). We illustrate the workings of the head rules with an example: to find the headchild for the non-terminal `SN`, we first search for the right-most child with label `GRUP`. If no such child exists, we look for the left-most `SN`. If no such child exists, we look for the left-most `S`, `RELATIU`, or `SP`, and finally, the right-most `N`. In the case that no head child has yet been assigned, we assign the left-most child to be the head. Punctuation nodes are never assigned as head-children.

Non-Terminal	Dir	Headchild
CONJ	left	CS
COORD	left	CC
GRUP	right	N,W,P,Z,AQ,GRUP
INC	left	S
INFINITIU	left	V
INTERJECCIO	left	I
NEG	left	RN
PREP	left	SP
SA	left	AO,AQ
SN	right	GRUP
	left	SN
	left	S,RELATIU,SP
	right	N
SP	left	PREP
	left	SP
SADV	left	RG
	left	P
	left	SADV
s	right	AO,AQ
	left	s
S	left	GV
	left	INFINITIU
	left	S
	left	GERUNDI
SBAR	left	CP
*CC1	left	CC2
*CC2	right	any child
*CP	left	RELATIU-CP,CONJ-CP
	right	any child

Table A.1: Spanish head rules.

Appendix B

Identification of Clauses for AEP-Based Translation

To generate training data for the AEP prediction model of Chapter 5, we need to extract clause-level German parse trees and English AEPs from the parsed parallel corpus. Before extracting AEPs using the algorithm in Section 5.3, we first identify the clauses in the parse trees as follows. Any non-terminal labeled by the parser of (Collins, 1999) as `SBAR` or `SBAR-A` is labeled as a clause root. Any node labeled by the parser as `S` or `S-A` is also labeled as the root of a clause, unless it is directly dominated by a non-terminal labeled `SBAR` or `SBAR-A`. Any node labeled `SG` or `SG-A` by the parser is labeled as a clause root, unless (1) the node is directly dominated by `SBAR` or `SBAR-A`; or (2) the node is directly dominated by a VP, and the node is not directly preceded by a verb (POS tag beginning with `V`) or modal (POS tag beginning with `M`). Any node labeled `VP` is marked as a clause root if (1) the node is not directly dominated by a `VP`, `S`, `S-A`, `SBAR`, `SBAR-A`, `SG`, or `SG-A`; or (2) the node is directly preceded by a coordinating conjunction (i.e., a POS tag labeled as `CC`).

In German parse trees, we identify any nodes labeled as `S` or `CS` as clause roots. In addition, we mark any node labeled as `VP` as a clause root, provided that (1) it is preceded by a coordinating conjunction, i.e., a POS tag labeled as `KON`; or (2) it has one of the functional tags `-mo`, `-re` or `-sb`.

Appendix C

Reranking Modifier Translations

The n-best reranking model for the translation of modifiers of Chapter 5 considers a list of candidate translations. We hand-labeled 800 examples, marking the element in each list that would lead to the best translation. The features of the n-best reranking algorithm are combinations of the basic features in Tables C.1 and C.2.

Each list contained the n-best translations produced by the phrase-based system of (Koehn et al., 2003). The lists also contained a supplementary candidate — "DELETED" — signifying that the modifier should be deleted from the English translation. In addition, each candidate derived from the phrase-based system contributed one new candidates to the list signifying that the first word of the candidate should be deleted. These additional candidates were motivated by our observation that the optimal candidate in the n-best list produced by the phrase-based system often included an unwanted preposition at the beginning of the string.

1	candidate string
2	should the first word of the candidate be deleted?
3	POS tag of first word of candidate
4	POS tag of last word of candidate
5	top nonterminal of parse of candidate
6	modifier deleted from English translation?
7	first candidate on n-best list
8	first word of candidate
9	last word of candidate
10	rank of candidate in n-best list
11	is there punctuation at the beginning, middle, or end of the string?
12	if the first word of the candidate should be deleted, what is the string that is deleted?
13	if the first word of the candidate should be deleted, what is the POS tag of the word that is deleted?

Table C.1: Functions of the candidate modifier translations used for making features in the n-best reranking model.

1	the position of the modifier (0–4) in AEP
2	main verb in AEP
3	voice in AEP
4	subject in AEP
5	German input string

Table C.2: Functions of the German input string and predicted AEP output used for making features in the n-best reranking model.

Appendix D

German NEGRA Corpus

Throughout this thesis, we refer to syntactic parses generated from the parser of (Dubey, 2005). This parser was trained using the German NEGRA treebank (Skut et al., 1997). The annotation scheme of the NEGRA treebank includes nonterminals representing phrasal and functional categories, and preterminals from the standard "Stuttgart/Tübinger Tagsets" (STTS). For the sake of completeness, in this appendix we include a comprehensive list of the NEGRA nonterminals and preterminals, many of which are used in this thesis. Table D.1 lists the phrasal categories; Table D.2 lists the functional categories; and Table D.3 lists the part-of-speech tags.

AA	*superlative phrase with "am"*
AP	*adjective phrase*
AVP	*adverbial phrase*
CAC	*coordinated adpositions*
CAP	*coordinated adjective phrase*
CAVP	*coordinated adverbial phrase*
CCP	*coordinated complementizer*
CH	*chunk*
CNP	*coordinated noun phrase*
CO	*coordination*
CPP	*coordinated adpositional phrase*
CS	*coordinated sentence*
CVP	*coordinated non-finite verb phrase*
CVZ	*coordinated zu-marked infinitive*
DL	*discourse-level contituent*
ISU	*idiosyncratic unit*
MPN	*multi-word proper noun*
MTA	*multi-token adjective*
NM	*multi-token number*
NP	*noun phrase*
PP	*adpositional phrase*
QL	*quasi-language*
S	*sentence*
VP	*verb phrase*
VZ	*zu-marked infinitive*

Table D.1: The phrasal categories used in the NEGRA corpus.

AC	*adpositional case marker*	MR	*rhetorical modifier*
ADC	*adjective component*	MW	*way (directional modifier)*
AMS	*measure argument of an adj/adv*	NG	*negation*
APP	*apposition*	NK	*noun kernel modifier*
AVC	*adverbial phrase component*	NMC	*numerical component*
CC	*comparative complement*	OA	*accusative object*
CD	*coordinating conjunction*	OA2	*second accusative object*
CJ	*conjunct*	OC	*clausal object*
CM	*comparative conjunction*	OG	*genitive object*
CP	*complementizer*	PD	*predicate*
DA	*dative*	PG	*pseudo-genitive*
DH	*discourse-level head*	PH	*placeholder*
DM	*discourse marker*	PM	*morphological particle*
EP	*expletive*	PNC	*proper noun component*
GL	*prenominal genitive*	RC	*relative clause*
GR	*postnominal genitive*	RE	*repeated element*
HD	*head*	RS	*reported speech*
JU	*junctor*	SB	*subject*
MC	*comitative*	SBP	*passivized subject (PP)*
MI	*instrumental*	SP	*subject or predicate*
ML	*locative*	SVP	*separable verb prefix*
MNR	*postnominal modifier*	UC	*unit component*
MO	*modifier*	VO	*vocative*

Table D.2: The functional categories used in the NEGRA corpus.

ADJA	*attributive adj*	PWS	*subst interrogative pron*
ADJD	*adverbial or predicative adj*	PWAT	*attr interrogative pron*
ADV	*adverb*	PWAV	*adv interrogative or rel pron*
APPR	*preposition*	PAV	*pronomial adverb*
APPRART	*preposition with article*	PTKZU	*zu for infinitive*
APPO	*postposition*	PTKNEG	*negation particle*
APZR	*circumposition right part*	PTKVZ	*separated verbal particle*
ART	*article*	PTKANT	*answer particle*
CARD	*cardinal number*	PTKA	*particle with adjective*
FM	*foreign text*	SGML	*SGML markup*
ITJ	*interjection*	SPELL	*word spelled out*
ORD	*ordinal number*	TRUNC	*truncated*
KOUI	*subordinating conj with zu*	VVFIN	*finite verb*
KOUS	*subordinating conj with sent*	VVIMP	*imperative*
KON	*coordinating conjunction*	VVINF	*infinitive*
KOKOM	*comparison particle*	VVIZU	*infinitive with zu*
NN	*common noun*	VVPP	*perfect participle*
NE	*proper noun*	VAFIN	*finite auxiliary*
PDS	*substitutive demonstrative pron*	VAIMP	*auxiliary imperative*
PDAT	*attributive demonstrative pron*	VAINF	*auxiliary infinitive*
PIS	*substitutive indefinite pron*	VAPP	*auxiliary perfect participle*
PIAT	*attributive indefinite pron*	VMFIN	*finite modal*
PIDAT	*attributive indefinite pron*	VMINF	*modal infinitive*
	with determiner		
PPER	*irreflexive personal pron*	VMPP	*modal perfect participle*
PPOSS	*substitutive possessive pron*	XY	*not a word*
PPOSAT	*attributive possessive pron*	$,	*comma*
PRELS	*substitutive rel pron*	$.	*end of sentence punc*
PRELAT	*attributive rel pron*	$(	*sentence-internal punc*
PRF	*reflexive personal pron*		

Table D.3: The part-of-speech categories used in the NEGRA corpus. These are the standard "Stuttgart/Tübinger Tagsets" (STTS).

Appendix E

AEP Prediction Model Features for German-to-English Translation

In any feature-based model, the features themselves are crucial to the performance of the model. Without good features, the model will not have the information it needs to understand the domain and make strong predictions. This appendix provides an in-depth description of the features used for the decisions in our AEP prediction model for German-to-English translation.

In this appendix, we make a distinction between feature types and features. A *feature type* is the template used for stamping out a particular *feature*, which is constructed for an individual example. For instance, a feature type for the English stem decision might be the pair ⟨English stem candidate value, German main verb⟩. Then, given an individual English stem candidate and a German parse, the actual feature might be ⟨be, ist⟩.

Throughout this chapter, we provide many examples of German trees that have been parsed using the annotation scheme of the NEGRA corpus (Skut et al., 1997). Readers unfamiliar with this annotation may wish to consult Appendix D.

E.1 Stem Prediction

Choosing the English stem is the first decision made during AEP prediction. The size of the complete set of feature types for the stem decision is thirteen including the bias feature. These thirteen feature types are listed in Figure E-1, and we describe each one in more detail below. There are several example German trees and their stem features at the end

Stem Features	
0 (bias)	STEM
1	STEM + German main verb
2	STEM + each German verb
3	STEM + German spine
4	STEM + German tree
5	STEM + German verb sequence
6	lexicon features
7	STEM + *es gibt/es gab* + German object
8	STEM + German object
9	STEM + German object + German main verb
10	STEM + *es gibt/es gab* + German subject
11	STEM + German subject
12	STEM + German negation

Figure E-1: The features used for prediction of the stem, the first decision in AEP prediction.

of this section to which we refer throughout the description for concreteness. Note that throughout this chapter, we use the symbol "STEM" to refer to the value of the candidate under consideration (e.g., *take*, *put*, *be*, etc.). The number of stems in the candidate set we use in our model is 1,639. This list is generated from the training data.

0. STEM The bias feature type for the stem decision contains only the value of the candidate English stem. The example in Figure E-2 instantiates this feature with the English stem candidate *take*. Each different stem candidate would instantiate a different bias feature according to its value (e.g., *be*, *give*, etc.).

1. STEM + German main verb This feature type pairs the value of the English stem candidate and the value of the German main verb (extracted from the German parsed input). This feature is intended to model likely English-German verb pairs. For example, the clause in Figure E-3 has two verbs, *können* and *übernehmen*, the second of which is the main verb. In Figure E-2, we see the feature instantiated as the pair ⟨*take*, *übernehmen*⟩.

2. STEM + each German verb This feature type is a set of sub-feature types, one for each of the verbs in the German input conjoined with the value of the candidate stem. The clause in Figure E-3 has two verbs and therefore two features in the set, ⟨*take*,*übernehmen*⟩ and ⟨*take*, *können*⟩, shown in Figure E-2. In contrast, the clause in Figure E-4 has only

one verb and therefore one feature in the set for the stem candidate *be*, $\langle$*be*,*gibt*$\rangle$, shown in Figure E-6. The intuition behind this feature is that sometimes the English verb translation is better paired with a verb other than the main verb of the German input. For example, in certain verbal phrases like the German *ist befaßt*, or *is seized*, the German main verb (*befaßt*) does not directly correspond the correct English stem (*be*), but another verb (in this case *ist*) does.

3. STEM + German spine This feature type conjoins the value of the English stem candidate with the German spine. The spine-extraction algorithm extracts a subtree of the parsed German input, keeping intact any structure involving a complementizer or wh word, a subject, an object, and any verbs, and throwing away any remaining structure. A sample German spine is shown in Figure E-2, extracted from the tree in Figure E-3. The tag "kous" represents a complementizer; "sb" represents the subject; "oa" the object, and "vb" the verbs. The spine encodes the information that the German input is a subordinate clause of some kind, with a subject, an object, and two verbs. In contrast, the sample spine in Figure E-6, extracted from the tree in Figure E-4, represents an input that is an independent clause beginning with a conjunction (*und/and*) and has a subject, an object, and a single verb.

4. STEM + German tree This feature type conjoins the value of the English stem candidate with the entire parsed German input. The tables in Figures E-2 and E-6 each show the value stem candidate with a tree, copied from Figures E-3 and E-4, respectively.

5. STEM + German verb sequence This feature type conjoins the value of the English stem candidate with all of the verbs in the German input as a single sequence. For the tree in Figure E-3, the two verbs *übernehmen* and *können* form the verb sequence that gets paired with the candidate *take* in Figure E-2. The intuition here is that there may be certain sequences of verbs that commonly translate to some English stem.

6. lexicon features This feature type is a set of sub-feature types each of which relates the German main verb to the value of the candidate stem via its ranking in an externally-compiled lexicon. The full set of features is as follows, where "rank" is the rank of a candidate in a list of possible translations (stored in the lexicon) for the German main verb:

- rank=NOT_FOUND, if the candidate stem is not on the list of possible translations for the German main verb.
- rank=1, rank=2, rank=3, rank=4, rank=5, if the rank of the candidate is 1–5.
- rank≤2, rank≤5, rank≤10, rank≤15, rank≤20, rank≤30, rank≤40, rank≤50, if the rank of the candidate is at least 50.

Each of the above features has a corresponding negative feature that is instatiated in the case that the positive one does not. For instance, if the stem is of rank one in the lexicon, then in addition to having a feature "rank=1," there will be features "rank≠2," "rank≠3," etc. In the example in Figure E-2, the rank of the candidate *take* is one in the lexicon for all possible translations of the main verb *übernehmen*. In Figure E-6, the rank of *be* for the German verb *gibt* is "NOT FOUND."[1]

The lexicon itself is derived from GIZA++ translation probabilities.[2] These probabilities are used to rank potential English translations of German words. English translations with equal probability are given the same rank. The lexicon contains close to 160K German entries (158,329), not all of which are verbs. The highest number of translations for any single entry is 771 (for the word *sich*). The average number of translations per entry is 5.3. 82% of the translations are in rank 1–10, and 92% are in rank 1–20. 38% are rank 1.

7. STEM + es gibt/es gab + German object This feature type only applies to German clauses of the form *es gibt...* or *es gab...* (*there is...*, *there was..*), or some permutation thereof (e.g., *is there...?*, *were there...?*). It is a set of three sub-feature types, each of which conjoins some piece of information about the German object (below) with the value of the candidate stem:

- grandparent tag information of the left-most word in a noun phrase object
- parent tag information of the left-most word in a noun phrase object
- suffix information of the right-most word in a noun phrase object

[1] This is not because *be* is a poor candidate; rather it is because in the particular lexicon we used in our experiments, we removed the stem candidate *be* from the lexicon to try to discourage the model from selecting this stem too frequently. In fact, in an identical lexicon that still contains the candidate *be*, it is the number one candidate for *gibt*.

[2] All of our training data was used to create this lexicon. The subset of this data that was used as a development set to train the AEP models both in this appendix and Chapters 5 and 6 was also used to create the lexicon.

For example, the feature set in Figure E-4 (taken from the subtree in Figure E-5) contains information about the parent and grandparent tags of *allzu*, the left-most word under the noun phrase object`np-oa`; it also contains the final three characters of the right-most word, *anzeichen*.

8. STEM + German object This feature type is identical to feature type 7 except that the input is not constrained to be of any particular form. Therefore, this feature type is instantiated for the example in Figure E-3 (shown in Figure E-2).

9. STEM + German object + German main verb This feature type is identical to feature type 8, except that each subfeature is conjoined with the German main verb in addition to the value of the English stem candidate. Examples are given in Figures E-2 and E-6.

10. STEM + es gibt/es gab + German subject This feature type applies only to German clauses of the form *es gibt* or *es gab*. It contains the same information as feature type 7, only the information is extracted from a noun-phrase subject (`np-sb`).

11. STEM + German subject This feature type is identical to feature type 10 except that the input is not constrained to be of any particular form. It is therefore instantiated for the example in Figure E-3 (shown in Figure E-2).

12. STEM + German negation This feature type conjoins the value of the English stem candidate with the parent and grandparent labels of the German negation word *nicht* if it appears in the German input. The example in Figure E-4 contains a negation that triggers the use of this feature, shown in Figure E-6. Here, a list containing the parent tag `ptkneg-ng` and grandparent tag `s:sc:und` is paired with the candidate *be*. The intuition behind this feature is that explicit negation on the German side may be embedded in the English stem in the translation.

E.2 Spine Prediction

The English spine is selected as the second decision in the AEP prediction process, after the stem. The spine contains crucial information about complementizers, *wh*-words, subjects,

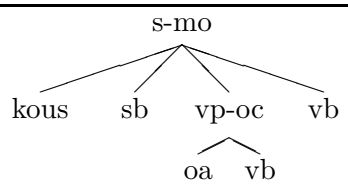
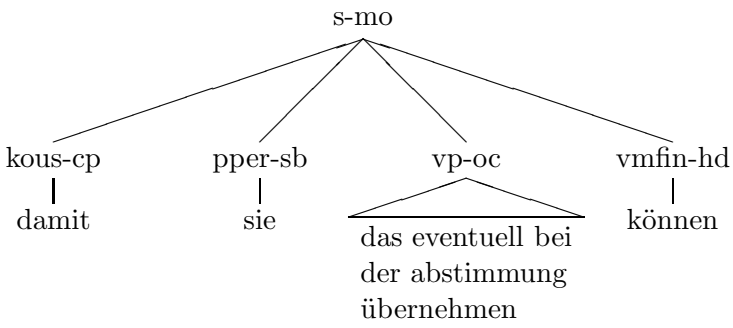
Stem Features	
0	take
1	take + übernehmen
2	take + übernehmen take + können
3	take + 
4	take + 
5	take + übernehmen,können
6	rank≠NOT_FOUND rank=1, rank≠2 , rank≠3, rank≠4, rank≠5 rank≤2, rank≤5, rank≤10, rank≤15, rank≤20, rank≤30, rank≤40, rank≤50
7	N/A
8	take + das take + NULL take + NULL
9	take + das + übernehmen take + NULL + übernehmen take + NULL + übernehmen
10	N/A
11	take + sie take + NULL take + NULL
12	take + NULL

Figure E-2: Stem features for the German clause *damit sie das eventuell bei der abstimmung übernehmen können* (shown in Figure E-3) with the English stem candidate *take*.

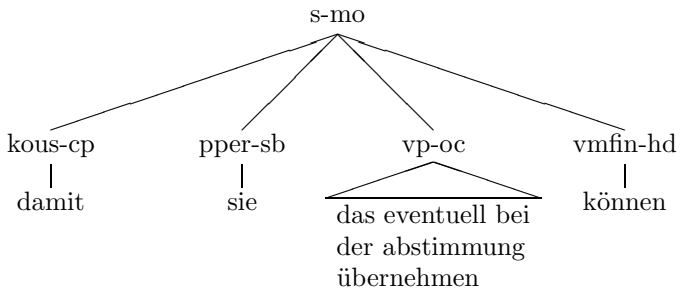


Figure E-3: Parsed input for the German clause *damit sie das eventuell bei der abstimmung übernehmen können*, or *so that they can take control in the election.* Figure E-2 shows the stem features for this example tree.

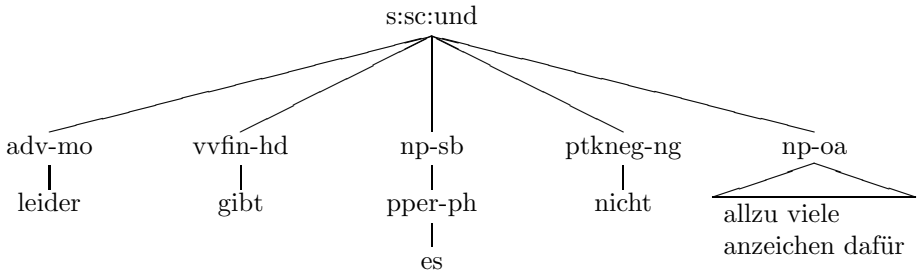


Figure E-4: Parsed input for the German clause *leider gibt es nicht allzu viele anzeichen dafür*, or *unfortunately, there is not very much evidence for it.* Figure E-6 shows the stem features for this example tree.

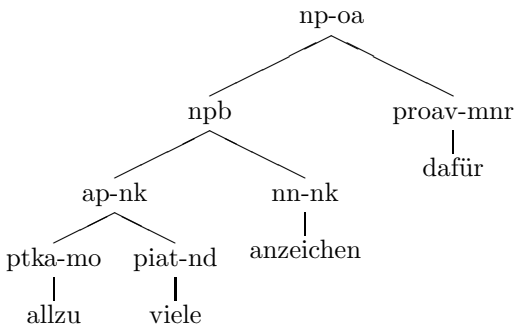


Figure E-5: Detail of the object *allzu viele anzeichen dafür* (translation: *very much evidence for it*) from Figure E-4, used to generate features 7 and 8 (shown in Figure E-6).

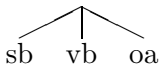
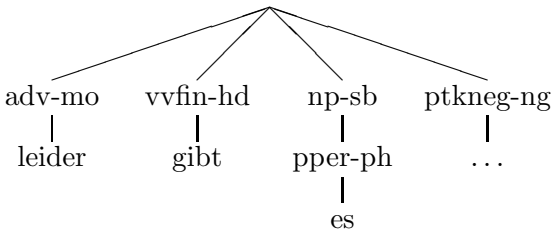
Stem Features	
0	be
1	be + gibt
2	be + gibt
3	be + 
4	be + 
5	be + gibt
6	rank=NOT_FOUND rank≠1, rank≠2 , rank≠3, rank≠4, rank≠5
7	be + ap-nk be + ptka-mo be + hen
8	be + ap-nk be + ptka-mo be + hen
9	be + ap-nk + gibt be + ptka-mo + gibt be + hen + gibt
10	be + pper-ph be + es be + es
11	be + pper-ph be + es be + es
12	be + ptkneg-ng,s:sc:und

Figure E-6: Stem features for the input *leider gibt es nicht allzu viele anzeichen dafür* (in Figure E-4) with the English stem candidate *be*.

Spine Feature Types	
0	SPINE
1	SPINE + STEM
2	SPINE + German spine
3	SPINE + German complementizer
4	SPINE + German relative pronoun
5	SPINE + German first verb
6a	SPINE: has subj? + German spine
6b	SPINE: has obj? + German spine
6c	SPINE: has wh? + German spine
7	SPINE: has wh stub? + German relative pronoun
8	SPINE + German verbs at end?
9	SPINE: top NT + German spine: top NT
10	SPINE: top NT + German complementizer
11	SPINE: top NT + German relative pronoun
12	SPINE + German negation
13	English question NTs + German first verb
14	English question NTs + German first verb NT
15	English wh-NTs + German wh-NTs
16	English wh-NTs + German wh-words
17	English complementizer + German complementizer
18	English complementizer + German tree: top NT

Figure E-7: The full set of spine feature types.

and objects. It is a representation of the argument structure of the English translation, and it contains many of the translation's function words.

In total there are nineteen feature types in the complete set listed in Figure E-7. Each of these feature types is explained in more detail below, and examples are provided at the end of the section. Throughout this appendix, the symbol "SPINE" is used to signify the value of the English spine candidate. The size of English spine candidate list is 385.

0. SPINE The bias feature type for the spine decision contains only the value of the English spine candidate. In Figure E-8, the spine under consideration is a subordinate clause (`sbar`) with a *wh*-noun phrase at the beginning, followed by a sentence containing a subject and verb (e.g., *what Mary ate*). The spine in Figure E-11 is a subordinate clause with the complementizer *that* at the beginning, followed by a sentence with both a subject and an object (e.g., *that Mary ate a banana*).

1. SPINE + STEM This feature type pairs the value of the English spine candidate with the value of the stem predicted in the preceding decision. For example, the feature shown in Figure E-8 shows the spine paired with the stem *be*. The idea behind this feature is to model the relationship between stems and their argument structure. For instance, transitive stems such as *hold* usually take an object, whereas intransitive stems such as *function* do not.

2. SPINE + German spine This feature type conjoins the value of the English spine candidate with the German spine, as defined in feature type 3 of the stem decision (Section E.1). For example, the German spine in Figure E-8 contains an object (`oa`), a subject (`sb`), and two verbs. The feature will try to measure how well that German spine pairs with the English spine candidate, depicted in the bias feature of the same figure. The German spine in Figure E-11 contains a complementizer (`kous`), an expletive (`ep`), and a verb.

3. SPINE + German complementizer This feature type conjoins the value of the English spine candidate with the value and part-of-speech label of any complementizer found in the German input. The idea here to model the relationship between English spines (in particular those containing complementizers), and German inputs with complementizers. Figures E-10 and E-11 show an example of a German input with complementizer *daß* and part-of-speech tag `kous-cp`, paired with a candidate English spine that also contains a complementizer.

4. SPINE + German relative pronoun This feature type conjoins the value of the English spine candidate with the value of any relative pronoun found under a prepositional phrase in the first position of the German input tree (the left-most child under the root). The example in Figure E-16 has the prepositional phrase *in denen* (*where*) in the target position, and the prepositional phrase contains the relative pronoun *denen*. The feature instantiated for this example pairs the English spine candidate with the German prepositional phrase. As in feature type 3 in this section, the idea is to model the relationship between specific spines and German inputs containing relative pronouns.

5. SPINE + German first verb This feature type conjoins the value of the English spine candidate and any verb that comes at the beginning of the German input. For

example, in Figure E-12, the German input contains the verb *lassen* in the beginning of an imperative sentence. The purpose of this feature type is to relate those German sentence types in which verbs are placed in the first position of the sentence (such as imperatives and interrogatives) with appropriate English spines.

6. SPINE questions + German spine This is a set of three sub-feature types, each of which asks a question about the English spine candidate, conjoined with the value of the German spine as defined in feature type 3 of the stem decision (Section E.1). The questions are

- Does the spine have a subject?
- Does the spine have an object?
- Does the spine have a *wh*-word?

This feature is instantiated in Figures E-8 and E-11. In the first example, the English spine candidate contains a subject and *wh*-word but not an object; in the second example, it contains a subject and object but not a *wh*-word.

7. SPINE: has *wh*-word? + German relative pronoun This feature type asks whether the English spine candidate has a *wh*-word, and conjoins the answer with the value of the German relative pronoun, if there is one. The German relative pronoun is defined in the same way as feature type 4 in this section. The example in Figure E-16 shows an input that contains the prepositional phrase *in denen*, which contains a relative pronoun (`prels-ad`). In this case, the English spine candidate under consideration does not contain a *wh*-word.

8. SPINE + German verbs at end? This feature type asks whether all of the verbs in the German input come at the end or not, and conjoins the answer with the value of the English spine candidate. Both Figures E-9 and E-10 depicts inputs in which this situation occurs. Since verbs come at the end of German subordinate clauses, this feature is trying to model the relationship between subordinate clause inputs and specific types of English spines.

9. SPINE: top NT + German spine: top NT This feature type conjoins the nonterminal label at the top of the English spine candidate with the nonterminal label at the top of the German spine. For the example in Figure E-8, this feature pairs the English label `sbar-a` (signifying a subordinate clause) with the German label `s-oc` (signifying a clausal object).

10. SPINE: top NT + German complementizer This feature type conjoins the nonterminal label at the top of the English spine candidate with the German complementizer if it appears in the input. For example, Figure E-11 pairs the English label `sbar-a` with the German complementizer *daß* and its part-of-speech label `kous-cp`.

11. SPINE: top NT + German relative pronoun This feature type conjoins the nonterminal label at the top of the English spine candidate with the German relative pronoun if it appears in the input. The German relative pronoun is defined in the same way as feature type 4 of this section. The example in Figure E-17 depicts a case in which a relative pronoun exists in the German input. In this case, the label at the top of the English spine is `s`, and the German relative pronoun is *in denen*. The feature will model the relative merit of choosing a spine rooted at `s` or a spine rooted at some other nonterminal (e.g., `sbar-a`) for an input containing *in denen* (translated roughly as *where*).

12. SPINE + German negation This feature type conjoins the value of the English spine candidate with the parent and grandparent labels of the German negation word *nicht* if it appears in the German input. The example in Figure E-10 contains a negation that instantiates this feature. In this case, the English spine is paired with the labels `ptkneg-ng` and `s-oc`.

13. English question NTs + German first verb This feature type conjoins the values of any nonterminals in the English spine candidate that are indicative of the interrogative (i.e., `sq` and `sbarq`) with the value of any verb that comes at the beginning of the German input. The input in Figure E-12 is an example of an input that contains a verb at the beginning of the sentence. The verb, *lassen*, is in the imperative and means *let*.[3] As in

[3]Since the feature was designed to model relationships between certain German sentential structures and certain question-related English nonterminals, unfortunately this particular example adds noise to the training procedure.

feature type 5, the purpose of this feature is to try to choose appropriate English spines for question-type inputs.

14. English question NTs + German first verb NT This feature type conjoins the value of any nonterminals in the English spine candidate that are indicative of the interrogative (i.e., `sq` and `sbarq`) with the part-of-speech tag of any verb that comes at the beginning of the German input. This feature type is similar to feature type 13 above, except that its purpose is to generalize over specific verb strings to their part-of-speech labels. The example in Figure E-13 shows the difference between feature type 13 and this feature type: whereas the former contains the German verb string *lassen*, this feature contains it tag `vvimp-hd`. In both cases, there are no question-type nonterminals in the English spine candidate.

15. English *wh*-NTs + German *wh*-NTs This feature type looks for *wh*-nonterminal labels in both the English spine candidate and the German tree. The example in Figure E-15 is a case where they exist in both: the English spine contains the label `whnp`, and the German tree contains the label `pws-oa`.

16. English *wh*-NTs + German *wh*-words This feature type looks for *wh*-nonterminal labels in the English spine candiate, and conjoins them with any *wh*-words in the German input. The example in Figure E-8 shows that the English spine has a *wh* label (`whnp`) but the German input does not contain any *wh*-words. The example in Figure E-14 is an example where both exist.

17. English complementizer + German complementizer This feature type pairs any complementizers in the English spine candidate with any complementizers in the German input. In Figure E-11, this feature is used to model the relationship between the English complementizer *that* and the German *daß*.

18. English complementizer + German tree: top NT This feature type pairs any complementizers in the English spine candidate with the nonterminal label at the top of the German tree. Again in Figure E-11 we see the English complementizer *that* in relation to the German label `s-oc` (clausal object).

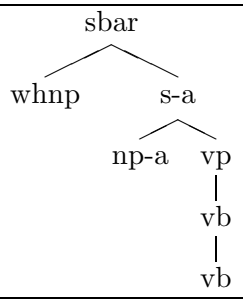
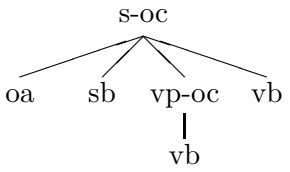
Spine Features	
0	
1	SPINE + have
2	SPINE + 
3	SPINE + NULL
4	SPINE + NULL
5	SPINE + NULL
6	yes + German spine no + German spine yes + German spine
7	yes + NULL
8	SPINE + yes
9	sbar-a + s-oc
10	sbar-a + NULL
11	sbar-a + NULL
12	SPINE + NULL
13	NULL + NULL
14	NULL + no
15	whnp + NULL
16	whnp + NULL
17	whnp + NULL
18	whnp + s-oc

Figure E-8: Spine features for the German clause *was drittländer und nachbarstaaten die ganze zeit über tun sollten* from Figure E-9.

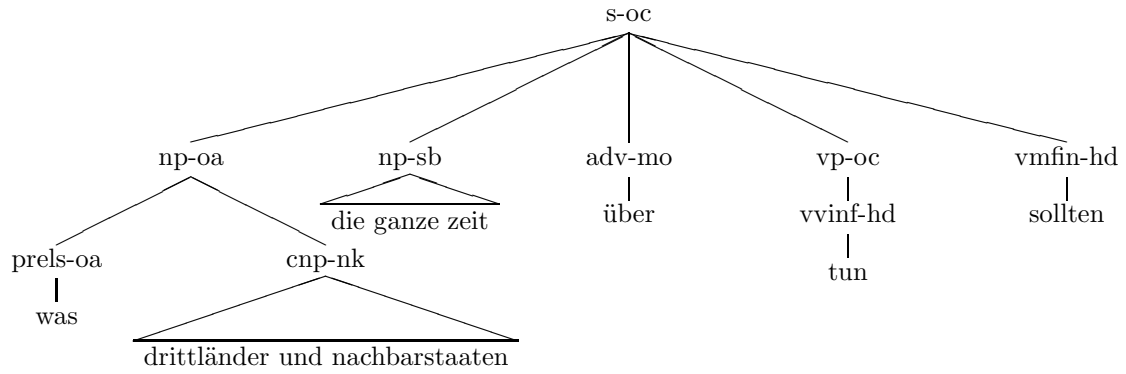


Figure E-9: German example *was drittländer und nachbarstaaten die ganze zeit über tun sollten*, or *what third countries and peripheral states should do all the time.*

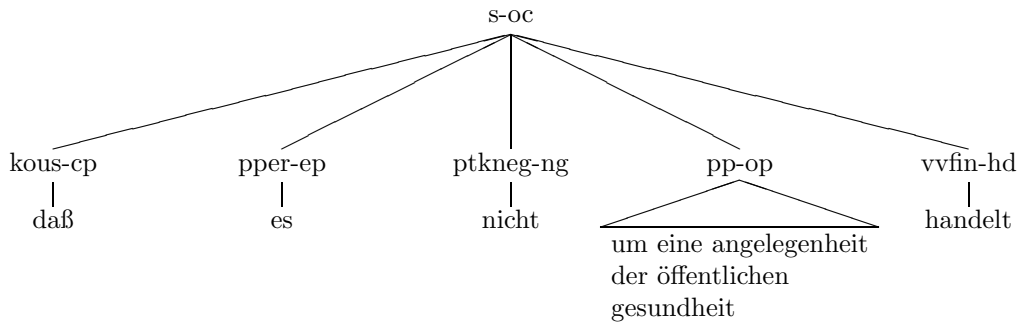


Figure E-10: German input *daß es nicht um eine angelegenheit der öffentlichen gesundheit handelt*, or *that it is not a matter of public health.*

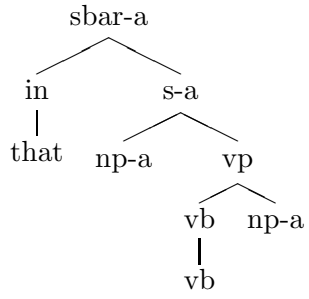
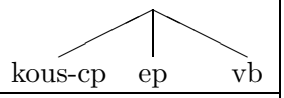
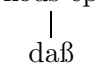
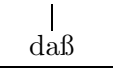
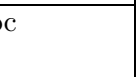
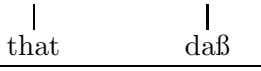
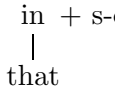
Spine Features	
0	
1	SPINE + be
2	SPINE + 
3	SPINE + 
4	SPINE + NULL
5	SPINE + NULL
6	yes + German spine yes + German spine no + German spine
7	no + NULL
8	SPINE + yes
9	sbar-a + s-oc
10	sbar-a + 
11	sbar-a + NULL
12	SPINE + 
13	NULL + NULL
14	NULL + no
15	NULL + NULL
16	NULL + NULL
17	
18	

Figure E-11: Spine features for the German clause *daß es nicht um eine angelegenheit der öffentlichen gesundheit handelt* from Figure E-10.

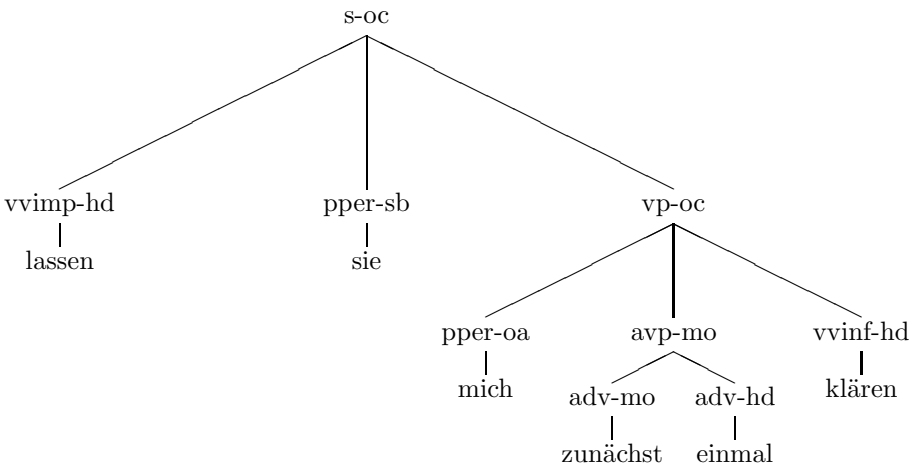


Figure E-12: German input *lassen sie mich zunächst einmal klären*, or *let me first of all clarify.*

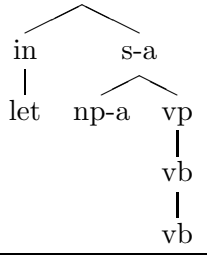
Spine Features	
0	
5	SPINE + lassen
13	NULL + lassen
14	NULL + vvimp-hd

Figure E-13: Spine features for the German clause *lassen sie mich zunächst einmal klären* from Figure E-12.

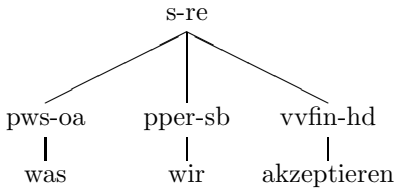


Figure E-14: German input *was wir akzeptieren*, or *what we accept.*

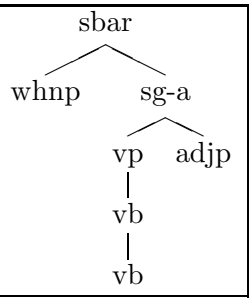
Spine Features	
0	
15	whnp + pws-oa
16	whnp + was

Figure E-15: Spine features for the German clause *was wir akzeptieren* from Figure E-14.

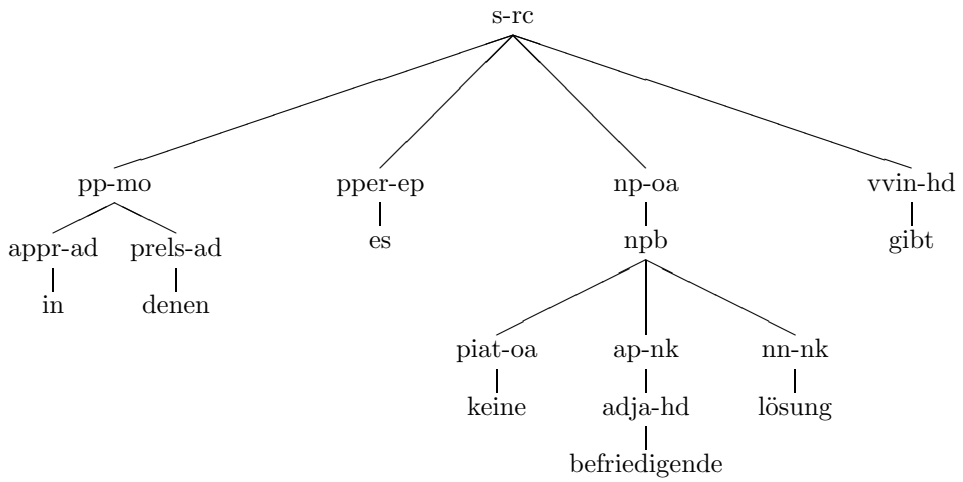


Figure E-16: German input *in denen es keine befriedigende lösung gibt*, or *there is no satisfactory solution*.

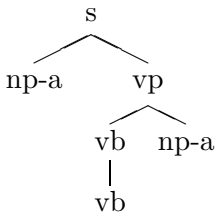
Spine Features	
0	
4	SPINE + in_denen
7	no + in_denen
11	s + in_denen

Figure E-17: Spine features for the German clause *in denen es keine befriedigende lösung gibt* from Figure E-16.

Voice Feature Types	
0	VOICE
1	VOICE + STEM
2	VOICE + STEM + SPINE
3	VOICE + German verb sequence
4	VOICE + each German verb
5	VOICE + German main verb
6	VOICE + German spine
7	VOICE + German tree
8	VOICE + German subject words
9	VOICE + German object words

Figure E-18: The full set of voice feature types.

E.3 Voice Prediction

Choosing the voice of the English clause is the third decision made during AEP prediction. The voice can be either `active` or `passive`. The size of the complete set of feature types for the voice decision is ten including the bias feature. These ten feature types are listed in Figure E-18, and we describe each one in more detail below.

0. VOICE The bias feature type for the voice decision contains only the value of the English voice candidate (either `passive` or `active`). For the example in Figure E-20, the bias feature is `active`.

1. VOICE + STEM This feature type conjoins the value of the voice decision with the value of the earlier stem decision. In Figure E-20, the value of the stem was *see*.

2. VOICE + STEM + SPINE This feature type conjoins the English voice, stem, and spine decisions. The spine in Figure E-20 contains slots for a subject and a verb.

3. VOICE + German verb sequence This feature type pairs the value of the English voice decision with all of the verbs in the German input as a single sequence. In the example of Figure E-20, the two verbs are *kann* and *erkennen*.

4. VOICE + each German verb This feature type is a set of sub-feature types, one for each of the verbs in the German input conjoined with the value of the voice. The clause in

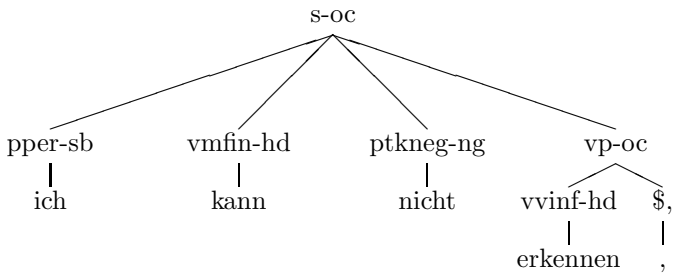


Figure E-19: German input *ich kann nicht erkennen*, or *i cannot see*.

Figure E-19 has two verbs and therefore two features in the set, ⟨*active*,*kann*⟩ and ⟨*active*, *erkennen*⟩, shown in Figure E-20.

5. VOICE + German main verb This feature type conjoins the value of the voice and the value of the main verb in the German input. For example, in Figure E-20, the voice `active` is paired with the German verb *erkennen*.

6. VOICE + German spine This feature type conjoins the value of the voice with the German spine. The spine extraction algorithm is the same as that described in feature #3 of Section E.1. An example is shown in Figure E-20, extracted from the tree in Figure E-19.

7. VOICE + German tree This feature type conjoins the value of the voice with the entire parsed German input. The table in Figure E-20 shows the value of the stem candidate with a tree, copied from Figure E-19.

8. VOICE + German subject words This feature type pairs the value of the voice with the value of the German subject, if there is one. The German subject is a top-level nonterminal with function tag "sb," for example the subject personal pronoun *ich* with label `pper-sb` in the example in Figure E-20.

9. VOICE + German object words This feature type pairs the value of the voice with the value of the German object, if there is one. The German object is a top-level nonterminal with function tag "oa," (accusative object). The example in Figures E-19 has no such tag.

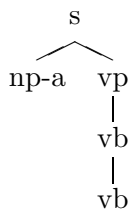
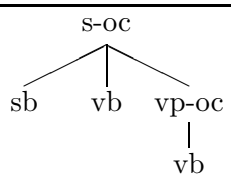
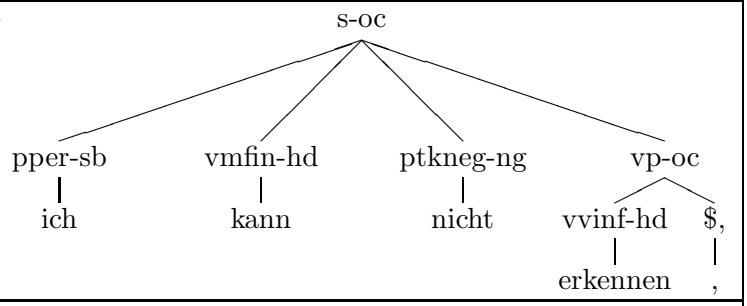
Voice Features	
0	active
1	active + see
2	active + see + 
3	active + erkennen,kann
4	active + erkennen active + kann
5	active + erkennen
6	active + 
7	active + 
8	active + ich
9	active + `null`

Figure E-20: Voice features for the German clause *ich kann nicht erkennen* from Figure E-19.

E.4 Subject Prediction

The subject is chosen fourth in the AEP prediction process, following the stem, spine, and voice of the English translation. Whether or not the English translation has a subject is determined by the form of the spine chosen in the second decision. If there is a noun phrase in the subject position of the spine, then some value for the subject must be predicted. That value can come from one of two places: either a phrase in the German input, or a list of 45 common English subjects without German counterpart (a list generated from the training examples).

Whenever the value of the English subject is taken from the German input tree, it is represented by the nonterminal label dominating the German phrase that will generate the English subject. This phrase may or may not be the subject in the German input. For example, if the German input is in the active voice, but the English translation is in the passive voice, then the German object will most likely become the English subject, and vice versa. In the example in Figures E-22 and E-23, the proposed subject of the English translation is the object of the German input (labelled with *np-oa*).

Whenever the value of the English subject is taken instead from the list of 45 common subjects, it is represented by the English string itself. For example, in Figures E-24 and E-25, the proposed subject of the English translation is the string *there*, which has no direct correspondence in the German input.

The size of the feature type set used for subject prediction is seventeen, and each of the feature types is listed in Figure E-21. In the remainder of this section, we describe each of these feature types in detail and provide examples of each one. Throughout this appendix, we use the symbol "SBJ" to refer to the value of the candidate subject.

0. SBJ The bias feature type for the subject decision contains only the value of the English subject candidate. For the example in Figure E-23, the bias feature contains a pointer into the German tree (shown in Figure E-22): the nonterminal label `np-oa` that dominates the phrase *eine große konferenz*. This phrase will be translated and placed in the subject position in the English translation. For the example in Figure E-25, the bias feature contains the string *there*, a candidate subject that has been taken from a list of 45 common English subjects with no correspondence in the German tree.

Subject Feature Types	
0	SBJ
1	SBJ + German verb sequence
2	SBJ + German verb sequence + STEM
3	SBJ + parent label
4	SBJ + German subject words
5	leaves under SBJ
6	SBJ + leaves under SBJ
7	SBJ + STEM
8	SBJ + German main verb
9	SBJ + STEM + German main verb
10	SBJ + PASSIVE/ACTIVE
11	SBJ: phrase label
12	SBJ: function label
13	number of leaves under SBJ
14	SBJ + SBJ label in CFG rule
15	SBJ + SBJ label in RHS context
16	SBJ + SBJ label with left and right sibs

Figure E-21: The full set of subject feature types.

1. SBJ + German verb sequence This feature type pairs the value of the English subject candidate with all of the verbs in the German input as a single sequence. Both the examples at the end of the section (in Figures E-23 and E-25) are taken from inputs with only a single verb (*gab*). For inputs with more than one verb, such as the input in Figure E-3 containing both *übernehmen* and *können*, the verbs would be gathered into a list and paired with the value of the English subject.

2. SBJ + German verb sequence + STEM This feature type is the similar to feature type 1, but has, in addition to the English subject and German verb sequence, the value of the stem chosen earlier in the decision sequence (e.g., *take*, *be*, etc.).

3. SBJ + parent label This feature type conjoins the value of the English subject candidate with its parent label, if there is one. In Figure E-23, this feature pairs the German `s` (the parent of the `np-oa`) with the English subject. If the subject has no correspondence in the German input (as in Figure E-25), then this feature type does not apply.

4. SBJ + German subject words This feature type pairs the value of the English subject candidate with the value of the German subject, if there is one. The German

subject is a top-level nonterminal with function tag "sb," for example the subject personal pronoun *es* with label `pper-sb` in the example in Figure E-25.

5. leaves under SBJ For English subject candidates that are derived from a phrase in the German tree, this feature type contains just the German phrase dominated by that nonterminal label. In that way, it's similar to the bias feature type but more specific. For example, the feature as instatiated in Figure E-23 contains the phrase under the `np-oa`, *eine große konferenz*. If the value of the subject is derived from the list of 45 common subjects (as in the example in Figure E-25), then this feature type does not apply.

6. SBJ + leaves under SBJ This feature type is similar to feature type 5 above except that it contains the value of the English subject candidate in addition to the string dominated by the German nonterminal. In Figure E-23, this means that the label `np-oa` is added to the feature. In Figure E-25, this feature is not instantiated because the English subject candidate does not have any direct correspondence to the German tree.

7. SBJ + STEM This feature type pairs the value of the English subject candidate with the value of the stem chosen in the first decision (e.g., *take*, *be*, etc.). The idea is that certain subjects may be commonly seen with certain stems (e.g., *there* and *be*, as in Figure E-25).

8. SBJ + German main verb This feature type conjoins the value of the candidate subject and the value of the main verb in the German input. For example, in Figure E-23, the English subject pointer `np-oa` is paired with the German verb *gab*. In Figure E-25, the English subject *there* is paired with the same German verb, *gab*. It might be helpful to know, in this last case, that the phrase *es gab* (*there is*) is commonly seen with the English subject *there*, for instance.

9. SBJ + STEM + German main verb This feature type is the same as feature type 8 above except that, in addition to the English subject and the German main verb, it contains the value of the stem chosen earlier in the decision process (e.g., *take*, *be*, etc.).

10. SBJ + PASSIVE/ACTIVE This feature type pairs the value of the English subject candidate with the value of the third decision, the voice of the English translation (*active*

or *passive*). In Figure E-23, the voice was predicted to be *active*, whereas in Figure E-25 is was *passive*.

11. SBJ: phrase label This feature type splits the nonterminal label of the English subject candidate into two parts — a phrase category and a function category — and lists the phrase category. It only applies in the case where the English subject corresponds to some phrase in the German tree, for example Figure E-23. In this example, the value of the feature is the noun phrase category `np`, taken from the full tag `np-oa`. This feature type was designed to try to generalize the labels seen in the bias feature for this decision.

12. SBJ: function label This feature type lists the functional category of the nonterminal label of the subject. It only applies in the case where the English subject corresponds to some phrase in the German tree. Again in Figure E-23, we see that the function label `oa` has been stripped from the complete tag `np-oa`. The motivation behind this feature type was the same as feature type 11: to try to generalize the labels seen in the bias feature for this decision.

13. number of leaves under SBJ This feature type counts the number of leaves dominated by the German nonterminal in cases where the English subject candidate is derived from a phrase in the German tree, as in Figure E-23. In this case, the number of leaves dominated by the `np-oa` is three. The feature was motivated by the hypothesis that the model might favor phrases of certain length as English subjects.

14. SBJ + SBJ label in CFG rule This feature type and the next two feature types look at the context in which the nonterminal label of the English subject candidate is positioned as a node in the German tree. This feature type looks at the position of the node in a CFG rule, where the node's parent corresponds to the left-hand side of the rule, and the node and its siblings correspond to the right-hand side. In Figure E-23, the CFG rule in which the label `np-oa` appears is `s`→`pp-mo vvfin-hd pper-ep np-oa`. The idea here is simply to try and put the German nonterminal in some context that might give the model more information when making the subject decision.

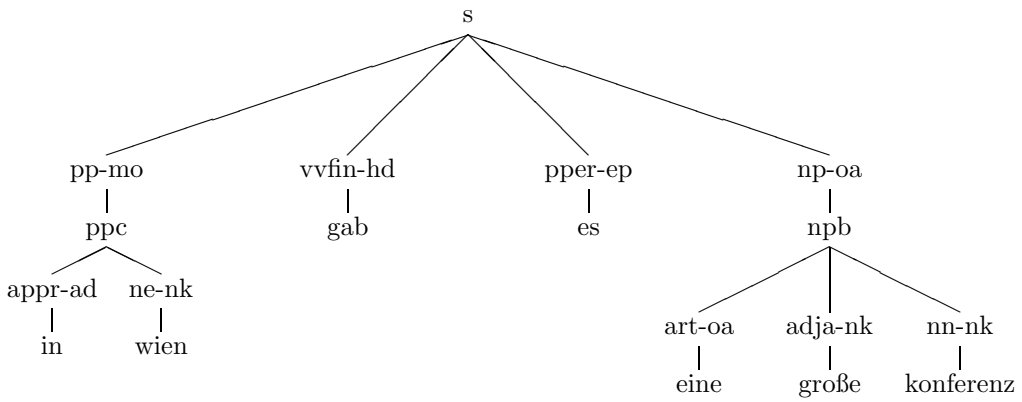


Figure E-22: German input *in wien gab es eine große konferenz*, or *in vienna there was a big conference*.

15. SBJ + SBJ label in RHS context This feature type looks only at the right-hand side of the CFG rule described above. In other words, it looks at the node and its siblings only (i.e., `pp-mo vvfin-hd pper-ep np-oa`). The idea here was to back off from the specificity of feature type 14.

16. SBJ + SBJ label with left and right sibs This feature type looks just at the left and right siblings of the node (i.e., `pper-ep np-oa STOP`). This feature type is even more general than feature type 15.

E.5 Object Prediction

The object is predicted fifth in the AEP prediction process, following the stem, spine, voice, and subject. As is the case with the subject, the presence of an English object in the translation is determined by the spine: if there is a noun phrase in the object position, then some value for the object must be selected. Just like the subject, that value can come from one of two places, either from the German input or from a list of common English objects without German counterpart. This list contains 54 of the most frequently-seen objects of this kind (derived from the training set).

The two examples at the end of this section, in Figures E-22 and E-23, show two different scenarios: what happens when the proposed object of the English translation has a correspondence in the German input, and what happens when there is no object slot in the spine.

Subject Features	
0	np-oa
1	np-oa + gab
2	np-oa + gab + hold
3	np-oa + s
4	np-oa + NULL
5	eine große konferenz
6	np-oa + eine große konferenz
7	np-oa + hold
8	np-oa + gab
9	np-oa + gab + hold
10	np-oa + passive
11	np
12	oa
13	3
14	np-oa + s→pp-mo vvfin-hd pper-ep np-oa*
15	np-oa + pp-mo vvfin-hd pper-ep np-oa*
16	np-oa + pper-ep np-oa* STOP

Figure E-23: Subject features for the German clause *in wien gab es eine große konferenz* from Figure E-22.

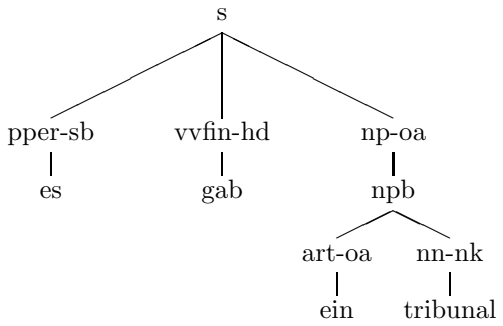


Figure E-24: German input *es gab ein tribunal*, or *There was a tribunal.*

Subject Features	
0	*there*
1	*there* + gab
2	*there* + gab + be
4	*there* + es
7	*there* + be
8	*there* + gab
9	*there* + gab + be
10	*there* + active

Figure E-25: Subject features for the German clause *es gab ein tribunal* from Figure E-24.

Object Feature Types	
0	OBJ
1	OBJ + German verb sequence
2	OBJ + German verb sequence + STEM
3	SBJ + parent label
4	OBJ + German object words
5	OBJ + leaves under OBJ
6	OBJ + STEM
7	OBJ + German main verb
8	OBJ + SBJ
9	OBJ + PASSIVE/ACTIVE
10	OBJ: phrase label
11	OBJ: function label
12	number of leaves under OBJ
13	OBJ + OBJ label in CFG rule
14	OBJ + OBJ label in RHS context
15	OBJ + OBJ label with left and right sibs

Figure E-26: The full set of object feature types.

The object decision has a feature type set of size sixteen (Figure E-26). These are described in detail in this section, with examples at the end. Due to a perceived similarity between subject and object, many of these features have a direct correspondence with some feature in the subject prediction section (Section E.4). In this section, we have starred feature types that are different from any feature described in Section E.4 so that the reader may focus on just the new features if he or she so desires.. In the remainder of this appendix, we use the symbol "OBJ" to refer to the value of the candidate object.

0. OBJ The bias feature type for the object decision contains only the value of the English object candidate. For the example in Figure E-28, the bias feature contains a pointer into the German tree (shown in Figure E-27): the nonterminal label `np-pd` (predicate nominative) that dominates the phrase *der vorhersehbare widerstand der hersteller*, *the predictable resistance of manufacturers*. This phrase will be translated and placed in the object position in the English translation. For the example in Figure E-25, the bias feature contains the string *BLANK* indicating the spine has no object slot.

1. OBJ + German verb sequence This feature type pairs the value of the English object candidate with all of the verbs in the German input as a single sequence. The

example in Figure E-28 is taken from an input with a single verb (*war*). For inputs with more than one verb, such as the input in Figure E-29 containing both *stellen* and *fest*, the verbs are gathered into a list and paired with the value of the English object.

2. OBJ + German verb sequence + STEM This feature type is the similar to feature type 1, but has, in addition to the English object and German verb sequence, the value of the stem chosen earlier in the decision sequence (e.g., *take*, *be*, etc.).

3. OBJ + parent label This feature type conjoins the value of the English object candidate with its parent label, if there is one. In Figure E-28, this feature pairs the German `s-oc` (the parent of the `np-pd`) with the English object. If there is no object in the spine, (as in Figure E-30), then this feature type has the value *BLANK* for the English object.

4. OBJ + German object words This feature type pairs the value of the English object candidate with the value of the German object, if there is one. The German object is a top-level nonterminal with function tag "oa," (accusative object). Neither examples in Figures E-27 and E-29 have such a tag.

5. OBJ + leaves under OBJ For English object candidates that are derived from a phrase in the German tree, this feature type contains just the German phrase dominated by that nonterminal label. In that way, it's similar to the bias feature type but more specific. For example, the feature as instatiated in Figure E-28 contains the phrase under the `np-pd`, *der vorhersehbare widerstand der hersteller*.

6. OBJ + STEM This feature type pairs the value of the English object candidate with the value of the stem chosen in the first decision (e.g., *take*, *be*, etc.). The idea is that certain objects may be commonly seen with certain stems, although this may be less true than for English subjects.

7. OBJ + German main verb This feature type conjoins the value of the candidate object and the value of the main verb in the German input. For example, in Figure E-28, the English object pointer `np-pd` is paired with the German verb *war*.

8. OBJ + SBJ This feature pairs the value of the English object candidate and the value of the object chosen in the fourth decision.

9. OBJ + PASSIVE/ACTIVE This feature type pairs the value of the English object candidate with the value of the third decision, the voice of the English translation (*active* or *passive*). In both examples at the end of the section, the voice was predicted to be *active*.

10. OBJ: phrase label This feature type splits the nonterminal label of the English object candidate into two parts — a phrase category and a function category — and lists the phrase category. It only applies in the case where the English object corresponds to some phrase in the German tree, for example Figure E-28. In this example, the value of the feature is the noun phrase category `np`, taken from the full tag `np-pd`. This feature type was designed to try to generalize the labels seen in the bias feature for this decision.

11. OBJ: function label This feature type lists the functional category of the nonterminal label of the object. It only applies in the case where the English object corresponds to some phrase in the German tree. Again in Figure E-28, we see that the function label `pd` has been stripped from the complete tag `np-pd`. The motivation behind this feature type was the same as feature type 10: to try to generalize the labels seen in the bias feature for this decision.

12. number of leaves under OBJ This feature type counts the number of leaves dominated by the German nonterminal in cases where the English object candidate is derived from a phrase in the German tree, as in Figure E-28. In this case, the number of leaves dominated by the `np-pd` is five. The feature was motivated by the hypothesis that the model might favor phrases of certain length as English objects.

13. OBJ + OBJ label in CFG rule This feature type and the next two feature types look at the context in which the nonterminal label of the English object candidate is positioned as a node in the German tree. This feature type looks at the position of the node in a CFG rule, where the node's parent corresponds to the left-hand side of the rule, and the node and its siblings correspond to the right-hand side. In Figure E-28, the CFG rule in which the label `np-pd` appears is `s-oc`→`kous-cp np-sb np-pd vafin-hd`. The idea

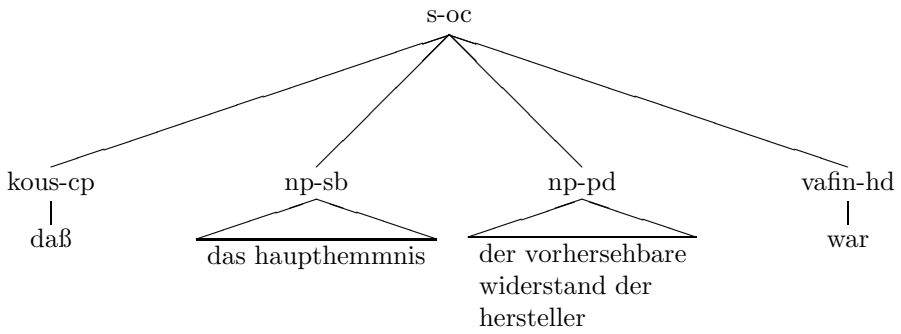


Figure E-27: German input *daß das haupthemmnis der vorhersehbare widerstand der hersteller war*, or *that the main obstacle has been the predictable resistance of manufacturers.*

here is simply to try and put the German nonterminal in some context that might give the model more information when making the object decision.

14. OBJ + OBJ label in RHS context This feature type looks only at the right-hand side of the CFG rule described above. In other words, it looks at the node and its siblings only (i.e., `kous-cp np-sb np-pd vafin-hd`). The idea here was to back off from the specificity of feature type 13.

15. OBJ + OBJ label with left and right sibs This feature type looks just at the left and right siblings of the node (i.e., `np-sb np-pd vafin-hd`). This feature type is even more general than feature type 14.

E.6 WH Prediction

The *wh* words in the English clause are predicted sixth in the AEP prediction process just after the object. As is the case with subject and object, the presence of an English *wh* phrase in the translation is determined by the spine: if there is slot for a *wh* phrase (e.g., `whnp` or `whadj`), then some value must be selected. That value is selected from a list of common English *wh* phrases. This list contains 79 phrases (derived from the training set). There are eight feature types used to predict *wh* words.

0. WH The bias feature type for the *wh* decision contains just the proposed *wh* phrase. In the example in Figure E-33, the bias feature is the word *which*. There has to be a slot in the spine for a *wh*-phrase (e.g., a `whnp` or `whadjp`).

Object Features	
0	np-pd
1	np-pd + war
2	np-pd + war + be
3	np-pd + s-oc
4	np-pd + NULL
5	np-pd + der vorhersehbare widerstand der hersteller
6	np-pd + be
7	np-pd + war
8	np-pd + das hauptemmnis
9	np-pd + ACTIVE
10	np
11	pd
12	5
13	np-pd + s-oc→kous-cp np-sb np-pd* vafin-hd
14	np-pd + kous-cp np-sb np-pd* vafin-hd
15	np-pd + np-sb np-pd* vafin-hd

Figure E-28: Object features for the German clause *daß das haupthemmnis der vorhersehbare widerstand der hersteller war* from Figure E-29.

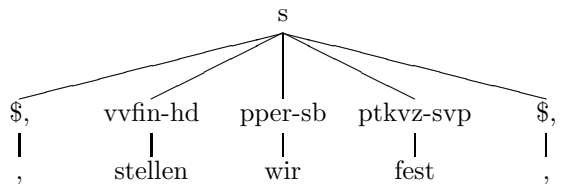


Figure E-29: German input *stellen wir fest*, or *we note*.

Object Features	
0	*BLANK*
1	*BLANK* + stellen_fest
2	*BLANK* + stellen_fest + see
4	*BLANK* + NULL
6	*BLANK* + see
7	*BLANK* + stellen_fest
8	*BLANK* + wir
9	*BLANK* + ACTIVE

Figure E-30: Object features for the German clause *stellen wir fest* from Figure E-29.

WH Feature Types	
0	WH
1	WH + SPINE
2	WH + German wh-NTs
3	WH + German main verb
4	WH + German wh-NTs + German main verb
5	WH + German wh-words
6	WH + German wh-words + German main verb
7	WH + German pp-rel words

Figure E-31: The full set of WH feature types.

1. WH + SPINE This feature type conjoins the candidate *wh* phrase with the value of the spine predicted in decision two. Figure E-33 shows the spine that has a `whnp` slot and an `np` subject.

2. WH + German *wh*-NTs This feature type looks for *wh*-nonterminal labels in the German tree and conjoins them if they exist with the English *wh* candidate. The example in Figure E-33 is a case where they do exist: the German tree contains the label `pwat-nk`.

3. WH + German main verb This feature type pairs the value of the candidate *wh* phrase and the value of the German main verb (extracted from the German parsed input). For example, the main verb in Figure E-32 is entfaltet.

4. WH + German *wh*-NTs + German main verb This feature type conjoins the value of the *wh* phrase with any *wh* nonterminals in the German tree and the German main verb.

5. WH + German *wh*-words This feature type looks for the leaves dominated by any *wh*-nonterminal labels in the German tree and conjoins them, if they exist, with the English *wh* candidate. The example in Figure E-33 conjoints *welche* with *which*.

6. WH + German *wh*-words + German main verb This feature type looks for the leaves dominated by any *wh*-nonterminal labels in the German tree and conjoins them, if they exist, with the English *wh* candidate and the German main verb.

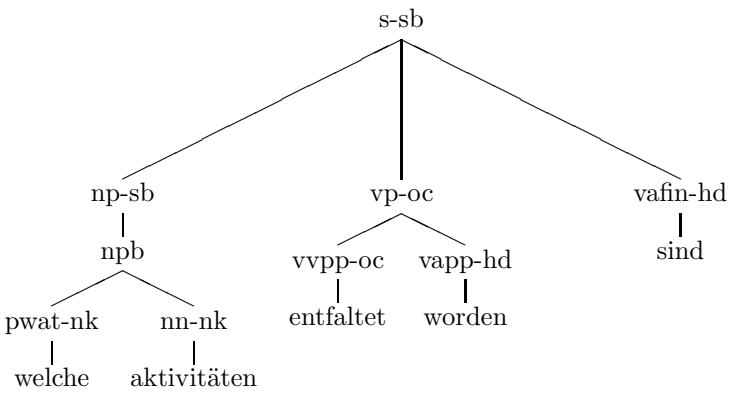


Figure E-32: German input *welche aktivitäten entfaltet worden sind*, or *which activities have been developed*.

7. WH + German relative pronoun This feature type conjoins the value of the English *wh* candidate with the value of any relative pronoun found under a prepositional phrase in the first position of the German input tree (the left-most child under the root). The example in Figure E-32 doesn't have a relative pronoun.

E.7 Modals Prediction

Predicting the modal sequence in the English translation is the seventh decison in the AEP model. The modal sequence is selected from a list of 338 common sequences, taken from the training data. There are nine features types used to predict modals, listed in Figure E-41 and described in more detail below.

0. MODALS The bias feature type for the modals decision contains just the proposed English modal sequence. In the example in Figure E-36, the bias feature contains the sequence *wanted to*.

1. MODALS + German verb sequence This feature type pairs the value of the English modal sequence with all of the verbs in the German input as a single sequence. The example in Figure E-36 is taken from an input with two verbs (*wollte* and *vorschlagen*).

2. MODALS + German spine This feature type conjoins the value of the English modals with the German spine. The spine extraction algorithm is the same as that described in feature #3 of Section E.1. An example is shown in Figure E-36, extracted from the tree

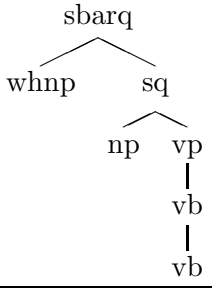
WH Features	
0	which
1	which + 
2	which + pwat-nk
3	which + entfaltet
4	which + pwat-nk + entfaltet
5	which + welche
6	which + welche + entfaltet
7	which + BLANK

Figure E-33: Object features for the German clause *welche aktivitäten entfaltet worden sind* from Figure E-32.

Modals Feature Types	
0	MODALS
1	MODALS + German verb sequence
2	MODALS + German spine
3	MODALS + German tree
4	MODALS + SPINE
5	MODALS + SPINE + STEM
6	MODALS + German negation
7	MODALS + each German verb
8	MODALS + STEM

Figure E-34: The full set of modals feature types.

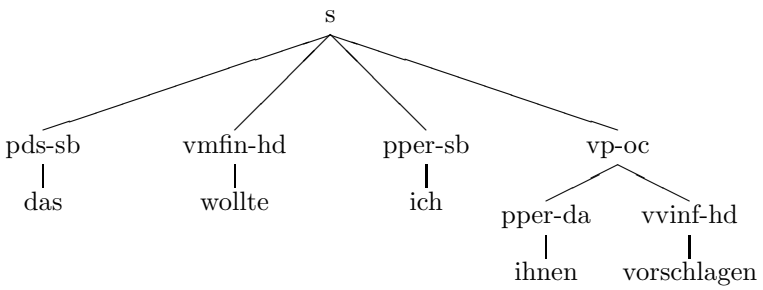


Figure E-35: German input *das wollte ich ihnen vorschlagen*, or *i wanted to propose that.*

in Figure E-35.

3. MODALS + German tree This feature type conjoins the value of the English modals with the entire parsed German input. The table in Figure E-36 shows the value of the modals candidate with the tree from Figure E-35.

4. MODALS + SPINE This feature type conjoins the value of the English modals with the value of the spine predicted in decision two. Figure E-36 shows the spine that has a subject and a verb.

5. MODALS + SPINE + STEM This feature type conjoins the value of the English modals with the value of the spine predicted in decision two and the stem predicted in decision one.

6. MODALS + German negation This feature type conjoins the value of the English modals candidate with the parent and grandparent labels of the German negation word *nicht* if it appears in the German input. The example in Figure E-10 does not contain any negation.

7. MODALS + each German verb This feature type is a set of sub-feature types, one for each of the verbs in the German input conjoined with the value of the voice. The clause in Figure E-35 has two verbs and therefore two features in the set, ⟨*wanted to*,*wollte*⟩ and ⟨*wanted to*, *vorschlagen*⟩, shown in Figure E-36.

8. MODALS + STEM This feature type pairs the value of the English modals candidate with the value of the stem chosen in the first decision (e.g., *take*, *be*, etc.).

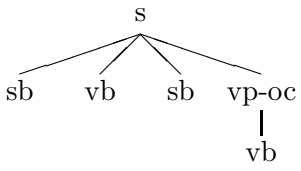
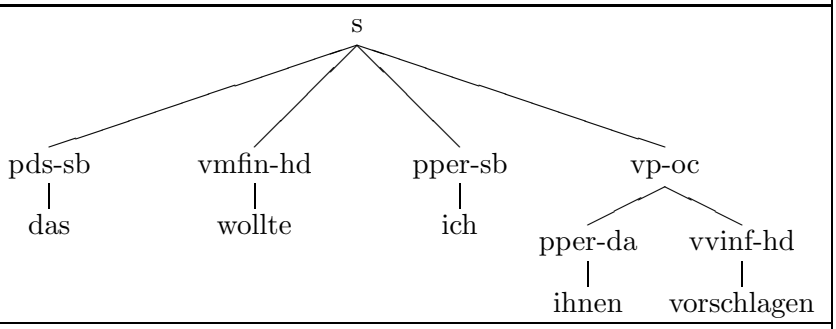
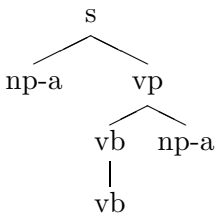
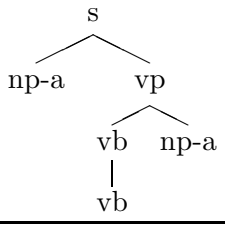
Modals Features	
0	wanted to
1	wanted to + say
2	wanted to + say + wollte, vorschlagen
3	wanted to + say + 
4	wanted to + 
5	wanted to + 
6	wanted to + say + 
7	wanted to + NULL
8	wanted to + wollte wanted to + vorschlagen

Figure E-36: Object features for the German clause *das wollte ich ihnen vorschlagen* from Figure E-35.

Inflection Feature Types	
0	INFL
1	INFL + STEM
2	INFL + STEM + MODALS are blank
3	INFL + are MODALS blank?
4	INFL + STEM + SBJ
5	INFL + SBJ
6	INFL + STEM + MODALS
7	INFL + MODALS
8	INFL + STEM + VOICE
9	INFL + VOICE

Figure E-37: The full set of inflection feature types.

E.8 Inflection Prediction

Predicting the inflection of the English main verb is the second-to-last decison in the AEP model. Options for the inflection decision are *root*, *first-person present*, *second-person present*, *third-person present*, *past singular*, *past plural*, *present participle*, *past participle*. There are ten features types used to predict modals, listed in Figure E-41 and described in more detail below.

0. INFL The bias feature type for the inflection decision contains just the proposed inflection of the English main verb. In the example in Figure E-36, the bias feature is *past singular*, which would cause the inflected form of the main verb *say* to be *said*.

1. INFL + STEM This feature type pairs the value of the inflection decision with the value of the stem chosen in the first decision (e.g., *say*, *be*, etc.).

2. INFL + STEM + are MODALS blank? This feature type pairs the value of the inflection decision with the value of the stem and whether or not this AEP has any predicted modal verbs.

3. INFL + are MODALS blank? This feature type pairs the value of the inflection decision with whether or not this AEP has any predicted modal verbs. Certain inflected forms might be more commonly paired modals than others. For example, it would be common for a present or past participle to be accompanied by some modal sequence.

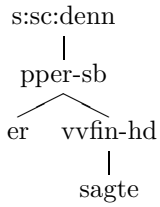


Figure E-38: German input *denn er sagte*, or *because he said*.

4. INFL + STEM + German subject words This feature type pairs the value of the inflection decision with the value of the stem and the German subject, if there is one. The German subject is a top-level nonterminal with function tag "sb," for example the subject personal pronoun *er* with label `pper-sb` in the example in Figure E-39.

5. INFL + German subject words This feature type pairs the value of the inflection decision with the value of the German subject, if there is one.

6. INFL + STEM + MODALS This feature type pairs the value of the inflection decision with the value of the stem and modals.

7. INFL + MODALS This feature type pairs the value of the inflection decision with the value of the modals decision.

8. INFL + STEM + VOICE This feature type pairs the value of the inflection decision with the value of the stem and voice decisions.

9. INFL + VOICE This feature type pairs the value of the inflection decision with the value of the voice decision.

E.9 Modifier Prediction

Predicting placement of German modifiers in the English translation is the final decison in the AEP model. For each modifier that has not already been designated the English subject or object, features are generated that help predict where it should be placed: before the English subject, between the subject and the modals, within the modals (specifically, between the first and the second modal), or after the main verb. A schematic of these

Inflection Features	
0	past singular
1	past singular + say
2	past singular + say + `true`
3	past singular + `true`
4	past singular + say + er
5	past singular + er
6	past singular + say + BLANK
7	past singular + BLANK
8	past singular + say + active
9	past singular + active

Figure E-39: Inflection features for the German clause *denn er sagte* from Figure E-38.

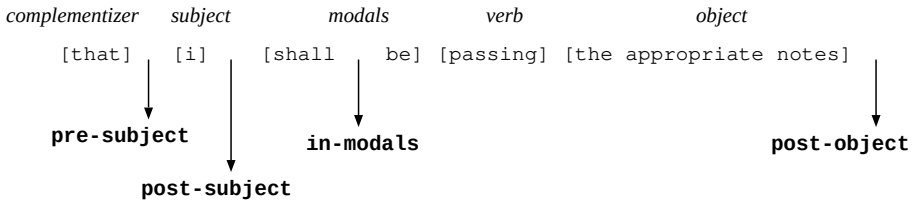


Figure E-40: Possible positions for modifier placement.

options is given in Figure E-40. Sometimes, a German modifier does not appear in the English translation, and in that case it is predicted as "deleted."

There are nine features used to predict modifier placement, listed in Figure E-41 and described in more detail below.

0. MOD The bias feature type for the modifier decision contains just the proposed position of the German modifier. In the examples in Figure E-43, the bias feature is only created if the modifier in question has not already been placed in some other role in the sentence (i.e., subject or object). In the table, just *nicht* and *so* are in need of placement. The features generated place *nicht* in an inter-modal position and *so* in a pre-subject position.

1. MOD + MOD: NT label This feature type conjoins the proposed placement of the German modifier with the label of the nonterminal that dominates the modifier in the German tree. For instance, in Figure E-43, the modifier *nicht* is dominated by the label `ptkneg`, and the modifier *so* is dominated by the label `adv-mo`. The idea informing this feature is that modifiers of certain classes might tend to be placed in similar positions.

Modifier Feature Types	
0	MOD
1	MOD + MOD: label
2	MOD + SPINE
3	MOD + leaves under MOD
4	MOD + child is leaf?
5	MOD + number of leaves under MOD
6	MOD + MOD: phrase label
7	MOD + MOD: function label
8	MOD + MODALS

Figure E-41: The complete set of modifier feature types.

2. MOD + SPINE This feature type conjoins the proposed placement of the German modifier with the value of the spine predicted in decision two. Figure E-43 shows the spine that has a subject and a verb. The idea here is that modifiers might tend to appear in certain positions according to spinal structure.

3. MOD + leaves under MOD This feature type looks at the proposed placement of the German modifier and its string value (i.e., the German actual phrase being placed). It is very likely that, if seen with sufficient frequency, certain words and phrases might tend to appear in certain positions in English sentences.

4. MOD + child is leaf? This feature type looks at whether or not the child of the modifier label is a leaf node, i.e., whether the modifier is a single-word or a multi-word phrase. Both modifiers in the figure are single-word phrases.

5. MOD + number of leaves under MOD This feature type examines the number of words in the modifier phrase. Again, in the examples both phrases contain only one word.

6. MOD + MOD: phrase label This feature type splits the nonterminal label of the modifier being placed into phrase category and functional category, and lists only the phrasal category. For *nicht* in Figure E-43, the phrase category is `ptkneg`; for *so*, it's `adv`.

7. MOD + MOD: function label This feature type contains just the functional category of the nonterminal label of the modifier being placed (`ng` for *nicht* in the example and `mo` for *so*).

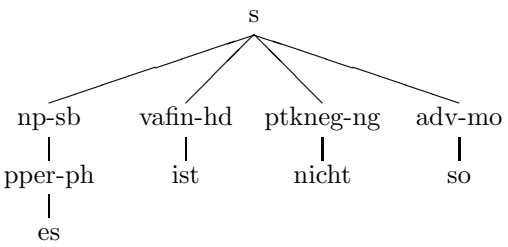


Figure E-42: German input *es ist nicht so*, or *it is not so*.

Modifier Features			
	es	nicht	so
0	NULL	in-modals	pre-subj
1	NULL	in-modals + ptkneg-ng	pre-subj + adv-mo
2	NULL	in-modals + s (np-a, vp (vb (vb)))	pre-subj + s (np-a, vp (vb (vb)))
3	NULL	in-modals + nicht	pre-subj + so
4	NULL	in-modals + yes	pre-subj + yes
5	NULL	in-modals + 1	pre-subj + 1
6	NULL	in-modals + ptkneg	pre-subj + adv
7	NULL	in-modals + ng	pre-subj + mo
8	NULL	in-modals + have been	pre-subj + BLANK

Figure E-43: Modifier features generated for two modifiers being placed: *nicht* and *so*. The modifier *es* has already been placed as the subject of the translation, so it does not generate any modifier features.

8. MOD + MODALS This feature type pairs the placement of the German modifier with the value of the preceding modals decision. In Figure E-43, *nicht* is being considered for an inter-modal position between *have* and *been*, which would generate the ordering *have* ⟨*nicht*⟩ *been*. For *so*, no modals were selected.

Appendix F

Instructions for Judges

This appendix contains the instructions that were given to the six judges in the human evaluation of Chapter 6. The evaluation was carried out in two separate stages. In stage 1, the judges were asked to compare the translations on the basis of fluency. In stage 2, they were asked to compare the translations on the basis of adequacy.

F.1 Fluency Instructions

Machine Translation Evaluation, Stage 1
Instructions For Judges

F.1.1 Goal

In this evaluation, you will compare the performance of two experimental German-to-English machine translation systems. The evaluation will occur in two stages. In this stage (the first of two), translations will be compared on the basis of fluency, which will be discussed below.

F.1.2 Stage One: Fluency

In the first stage, you will receive a file containing 200 numbered pairs of translations separated by a line of hyphens:

```
15---------------------------------------------

however , we want a comprehensive , serious and practicable
regulation for all concerned .

however , we want a comprehensive , serious and practical
arrangements for all the parties concerned .

---------------------------------------------
```

Each pair consists of two automatically-generated translations, one from each system. In each pair, the translations have been randomly ordered.

Your task in this part of the evaluation is to compare the translations on the basis of fluency. **Fluency refers to the degree to which a translation is well-formed according to the rules of standard written English. A fluent sentence is one that is well-formed grammatically, contains correct spellings, adheres to common use of terms, titles, and names, is intuitively acceptable and can be sensibly interpreted by a native speaker of English.** For each example, you should decide whether the first translation is more fluent, the second is more fluent, or the two are the same.

You should spend, on average, no more than 30 seconds assessing the fluency of a translation pair. We strongly encourage you to provide your intuitive assessment of each pair and not to ponder your decisions.

F.1.3 How to Record Judgments

Once you've judged an example, you should record that judgment using the markings shown in Figure F-1. The markings should be recorded directly in the text file, in the space between the numbered line of hyphens at the top of the example and the first translation. I.e.,

First is better	**F**
Second is better	**S**
They are the same	=

Figure F-1: Use these markings to indicate whether the first translation is more fluent, the second is more fluent, or the fluency is the same.

```
15-----------------------------------------------

F|S|=       <-------MARK GOES HERE

however , we want a comprehensive , serious and practicable

regulation for all concerned .


however , we want a comprehensive , serious and practical

arrangements for all the parties concerned .


-----------------------------------------------
```

When you have finished the fluency judgments, please send the annotated file to Brooke. Shortly thereafter, you will receive instructions for the second stage of the evaluation. Please don't hesitate to send mail if you have any questions during the evaluation!

F.1.4 Points To Be Aware Of

German words in the translations

Some automatically-generated translations will contain German words for which the system was unable to find a translation. Not all translations are equally affected by these foreign words. In sentences where foreign words appear, do your best to assess whether one of the translations is better than the other in terms of fluency, or whether they are the same.

F.1.5 Notes

1. We strongly recommend not doing more than 100 judgments in one sitting.

F.2 Adequacy Instructions

Machine Translation Evaluation, Stage 2

Instructions For Judges

F.2.1 Goal

In this evaluation, you will compare the performance of two experimental German-to-English machine translation systems. The evaluation will occur in two stages. In this stage (the second of two), translations will be compared on the basis of adequacy, which will be discussed below.

F.2.2 Stage Two: Adequacy

In the second stage, you will receive a file containing 200 numbered triples of translations separated by a line of hyphens:

```
15----------------------------------------------

REFERENCE:
what we want is a comprehensive , prudent and practicable
arrangement for all concerned .

however , we want a comprehensive , serious and practicable
regulation for all concerned .

however , we want a comprehensive , serious and practical
arrangements for all the parties concerned .

----------------------------------------------
```

Each triple consists of a reference translation followed by two automatically-generated translations, one from each system. **In this stage, you should decide which translation is better, the first or the second, or whether they are of the same quality, given the reference translation. Use your intuition when deciding whether one translation**

First is better	**F**
Second is better	**S**
They are the same	=

Figure F-2: Use these markings to indicate whether the first translation is better, the second is better, or the two are of the same quality.

is better than the other: an ideal translation should correctly communicate the meaning of the reference translation and should also be fluent/grammatically well-formed.

You should spend, on average, no more than 30 seconds assessing the adequacy of a translation pair. We strongly encourage you to provide your intuitive assessment of each pair and not to ponder your decisions.

F.2.3 How to Record Judgments

Once you've judged an example, you should record that judgment using the markings shown in Figure F-2. The markings should be recorded directly in the text file, in the space between the numbered line of hyphens at the top of the example and the first translation. I.e.,

```
15-----------------------------------------------
F|S|=       <-------MARK GOES HERE
REFERENCE:
what we want is a comprehensive , prudent and practicable
arrangement for all concerned .

however , we want a comprehensive , serious and practicable
regulation for all concerned .

however , we want a comprehensive , serious and practical
arrangements for all the parties concerned .

-----------------------------------------------
```

When you have finished the adequacy judgments, please send the annotated file to Brooke. Please don't hesitate to send mail if you have any questions during the evaluation!

F.2.4 Points To Be Aware Of

German words in the translations

Some automatically-generated translations will contain German words for which the system was unable to find a translation. Not all translations are equally affected by these foreign words. In sentences where foreign words appear, do your best to assess whether one of the translations is better than the other, or whether they are of the same quality.

Mismatched reference translation

Sometimes you might find that the reference translation doesn't match the automatically-generated output. Here's an example:

```
43----------------------------------------------

democratic basic values and rules are securely anchored .

of course , we are following further potential risks and
unresolved issues .

we are , of course , continue to potential risks and
unresolved issues .

----------------------------------------------
```

This happens infrequently, but if you run into such a case, just say the translations are the same quality.

F.2.5 Notes

1. We strongly recommend not doing more than 100 judgments in one sitting.

Appendix G

Examples from Human Evaluation

The following nine tables each contain ten randomly-selected examples from the human evaluation discussed in Chapter 6. In the first six tables, all of the examples have at least one judgment in common. For example, in the first table, the AEP-based system's output was judged more fluent by at least one annotator; in the second table, the phrase-based system's output was more fluent, and so on. In the last three tables, all of the examples were given the same fluency and adequacy judgment by two annotators. Overall, 47% of the 600 sentence pairs were given the same fluency and adequacy judgments by two annotators.

The theme of each table is stated above the table itself. On the left-hand side of the table, there are two columns, one labeled F (fluency) and the other A (adequacy). The cells in these columns contain check marks ($\surd$) indicating how the judge marked that example for fluency and adequacy. In cases where the pair was judged of equal quality (either fluency or adequacy), there is a check mark next to both the AEP systems output (labeled A) and the phrase-based system's output (labeled P). The human-generated reference translation in each example is labeled R.

It is possible for a single example to appear in two different tables. This can happen if the same example was randomly selected for both fluency and adequacy; or, it might be selected more than once for the same metric (either fluency or adequacy) in the case that the two annotators disagreed. In the case that two annotators agreed, the example might appear in the first set of six tables and again in the last three.

F	A	AEP SYSTEM IS MORE FLUENT	
		R:	that , in a few words , should have been our message today .
√	√	**A:**	this should have been our message with a few words .
		P:	this would have today , with a few words , our signal should be .
		R:	a great deal of pressure has been placed upon traditional social values in the former planned economies of central and eastern europe .
√	√	**A:**	there is considerable pressure in the former planned economies of central and eastern europe to the traditional social values .
		P:	in the former planned economies of central and eastern europe , the traditional social values considerable pressure .
		R:	it is also best to give local examples to show how this legislation is relevant .
√	√	**A:**	it is best to clarify the basis of a local examples such legislation is as important .
	√	**P:**	it is best , on the basis of local examples to show how important such legislation .
		R:	however strict quality criteria need to be laid down at community level for these decontamination procedures .
√	√	**A:**	stringent quality criteria must be set for this dekontaminierungsverfahren at community level .
		P:	for this dekontaminierungsverfahren must , however , at community level strict quality criteria to be established .
		R:	it was also clear that the staff who work in romanian orphanages need further child care opportunities .
√		**A:**	it is the betreuungsmöglichkeiten for the staff in the romanian orphanages must be improved .
	√	**P:**	became clear that the betreuungsmöglichkeiten for the staff in the romanian orphanages must be improved .
		R:	ever since the beginning , it has been more or less clear that researchers could not be part of the recommendation .
√		**A:**	from the outset , it is quite clear it can not be included in the recommendation .
	√	**P:**	it was quite clear from the outset that researchers not included in the recommendation .
		R:	it is foreign intervention which is to blame for the partition of the island .
√	√	**A:**	the intervention of the outside world is responsible for the division of the island .
		P:	responsible for the division of the island are the intervention from outside .
		R:	democratic elections were held there in 1990 , which the international community deemed fair and free .
√	√	**A:**	democratic elections have been held in 1990 which is in international judgment fair and free .
	√	**P:**	in 1990 there democratic elections were held under international judgment fair and free goods .
		R:	this has been achieved by a fundamental reorganisation of the border police , once a demotivated conscript force ; it is now a professional volunteer service .
√	√	**A:**	this has been achieved by way of a motivationslosen wehrpflichttruppe to a berufskorps on a voluntary basis .
		P:	this overtaken by transformation of a motivationslosen wehrpflichttruppe a berufskorps on a voluntary basis to be made .

F	A	PHRASE-BASED SYSTEM IS MORE FLUENT	
		R:	i should like to point out that all the greek meps have now reacted against this situation .
	√	**A:**	i must emphasize now all greek members of parliament is in the european parliament on this issue .
√	√	**P:**	i must emphasise that already all greek members of the european parliament on this matter have expressed .
		R:	some of the members have referred to the particular problems that turkey has with the kurdish language and the problems in turkey .
		A:	a number of members have referred to the specific problems which is in turkey in relation to the kurdish language .
√	√	**P:**	a number of members have referred to the specific problems in turkey in relation to the kurdish language there .
		R:	i thank mr mann for his support on this point .
	√	**A:**	i thank mr mann for his support on this point .
√		**P:**	i would like to thank mr mann for his support on this point .
		R:	the family , for instance , remains a foundation and a pillar of our society .
		A:	the family , for example , is the basis of society and pillar unchanged .
√	√	**P:**	the family , for example , is unchanged basis and pillar of society .
		R:	farmers are still suffering today from that utterly criminal failure to protect consumers .
	√	**A:**	even today , the farmers are suffering for example , because the protection of consumers is to the kriminellste .
√		**P:**	the farmers suffering even today , including , as the consumer protection on the kriminellste abused .
		R:	the pace of the negotiating process itself increased considerably again last year .
		A:	last year , the process as such , has been summoned once again clearly .
√	√	**P:**	the negotiation process , as such is the last year clear once again been speeded up .
		R:	i am sure that we will manage it , because we must !
	√	**A:**	i am sure we will manage it because we must build it !
√	√	**P:**	i am sure that we will succeed because we must create !
		R:	we are certain that mr milosevic will fall , possibly this evening , certainly soon .
		A:	we are of course , that mr milosevic will fall perhaps this evening determine as soon as possible .
√	√	**P:**	we are sure that mr milosevic will fall , perhaps this evening , certainly soon .
		R:	i would like to underline this , also on behalf of my group .
		A:	on behalf of my group , i would also like to stress this this .
√	√	**P:**	on behalf of my group , i would like to stress that too .
		R:	if you allow me , i would like to elaborate on two aspects .
	√	**A:**	please you i go on two aspects of this mentioned in more detail .
√	√	**P:**	allow me to refer to two aspects mentioned in dwell .

F	A	FLUENCY IS THE SAME	
		R:	this is more than a question ; it is a real challenge , and it is up to us to accept it .
√	√	**A:**	this is more than a question , but the real challenge , we have to face .
√		**P:**	this is more than a question , but the real challenge we must ask ourselves .
		R:	its political and macro-economic benefits will be greater than the additional costs the member states will incur .
√		**A:**	the political and macroeconomic use will be as an additional costs incurred to the members .
√	√	**P:**	the political and macroeconomic use will be greater than the additional costs of the members to emerge .
		R:	there is no doubt at all that right-wing forces have been involved .
√		**A:**	of course , is the right to a very forces actively .
√	√	**P:**	there is no doubt that were the right forces very active .
		R:	the kozloduy nuclear power plant accounts for half the internal power generation in bulgaria .
√	√	**A:**	about half of the production of bulgaria falls to the nuclear power station koslodui should be adhered .
√	√	**P:**	about half of the production of bulgaria falls to the nuclear power plant in koslodui should be adhered to .
		R:	as mr titley pointed out - rightly so - the code can be made more transparent by providing more and better information .
√		**A:**	as mr titley pointed right the code of conduct , more and better information can be made more transparent .
√	√	**P:**	as mr titley rightly points out , can the code by more and better information more transparent .
		R:	the regime 's treatment of the nobel laureate , aung san suu kyi , is all of a piece with their treatment of the burmese people .
√		**A:**	the other birmesen will also be taken as the nobelpreisträgerin aung san suu kyi .
√	√	**P:**	as the nobelpreisträgerin aung san suu kyi will also be the other birmesen dealt with .
		R:	i too believe that the staffing levels in those institutions are too low to warrant extra manpower .
√	√	**A:**	the very limited staff of these institutions it is certainly not to provide additional resources .
√		**P:**	the very limited staffing of these institutions , it is difficult to be additional resources available .
		R:	the good sense of the eea model is based on the idea that efforts to implement legislation should be rewarded immediately . benefits and responsibilities go hand in hand .
√	√	**A:**	this applies both to the citizens of the applicant countries and the current member states .
√	√	**P:**	this applies both to the citizens of the applicant countries and the current member states .
		R:	nonetheless , we have decided to vote against mr titley 's report for the following reasons :
√	√	**A:**	however , we are voting against the report by mr titley , for the following reasons :
√	√	**P:**	however , we are voting against the report by mr titley , for the following reasons :

F	A	AEP SYSTEM IS MORE ADEQUATE	
		R:	we would be embarking on a magical mystery tour if we were to abandon the project or postpone it indefinitely .
√	√	**A:**	an adventure would be it , or to abolish the project or postponing indefinitely .
√		**P:**	an adventure , it would be the project halt or indefinitely to be postponed .
		R:	obviously , this crisis may have dangerous repercussions on the peace process .
√	√	**A:**	it is quite clear that this crisis can have dangerous effects on the peace process .
		P:	it is quite clear that this crisis dangerous effects on the peace process .
		R:	that , in a few words , should have been our message today .
√	√	**A:**	this should have been our message with a few words .
		P:	this would have today , with a few words , our signal should be .
		R:	a great deal of pressure has been placed upon traditional social values in the former planned economies of central and eastern europe .
√	√	**A:**	there is considerable pressure in the former planned economies of central and eastern europe to the traditional social values .
		P:	in the former planned economies of central and eastern europe , the traditional social values considerable pressure .
		R:	this brings to mind another sound rule of thumb : ' no one can achieve the impossible ' .
√	√	**A:**	there is a good principle : ' l ' impossible , nul n est tenu "
		P:	there are good principle : ' l ' impossible , nul n ' est tenu . '
		R:	on a sociological basis , moreover , there is discrimination according to birth .
√	√	**A:**	and there is discrimination on the basis of birth in terms of the sociological .
		P:	and in sociological and there is discrimination on the basis of birth .
		R:	farmers are still suffering today from that utterly criminal failure to protect consumers .
√	√	**A:**	even today , the farmers are suffering for example , because the protection of consumers is to the kriminellste .
		P:	the farmers suffering even today , including , as the consumer protection on the kriminellste abused .
		R:	secondly , the euro is an artificial currency and it would be a disaster to make a complete changeover to the euro in the present circumstances .
√	√	**A:**	in the current context , the actual changeover to the euro would be a disaster .
		P:	under present conditions would be the complete changeover to the euro a disaster .
		R:	i am sure that we will manage it , because we must !
√	√	**A:**	i am sure we will manage it because we must build it !
		P:	i am sure that we will succeed because we must create !
		R:	nuclear power stations of first generation soviet construction are a particular danger as far as europe is concerned .
√	√	**A:**	nuclear power stations soviet design of the first generation applies for europe as a particularly dangerous .
		P:	nuclear power stations soviet design , the first generation apply for europe as especially dangerous .

F	A	PHRASE-BASED SYSTEM IS MORE ADEQUATE	
		R:	today the lithuanian economy is on the mend .
		A:	the general direction of the economy in lithuania is today , positive .
√	√	**P:**	the general direction of the economy in lithuania today is positive .
		R:	this relates to the specific statement made about turkey 's behaviour in this regard .
		A:	i say this to this in relation to the specific proposal on the conduct of turkey in this particular issue .
√	√	**P:**	i say this in relation to the specific proposal on turkey 's conduct in this particular issue .
		R:	allow me to make one more comment on the energy sector , which has already been addressed once today .
		A:	let it dropped and made a comment on the energy sector which has already been raised today .
√	√	**P:**	let me make a comment on the energy sector , the earlier today has been mentioned .
		R:	democratic elections were held there in 1990 , which the international community deemed fair and free .
		A:	democratic elections have been held in 1990 which is in international judgment fair and free .
√	√	**P:**	in 1990 there democratic elections were held under international judgment fair and free goods .
		R:	its political and macro-economic benefits will be greater than the additional costs the member states will incur .
√		**A:**	the political and macroeconomic use will be as an additional costs incurred to the members .
√	√	**P:**	the political and macroeconomic use will be greater than the additional costs of the members to emerge .
		R:	will there be waves of immigrants and what impact will this have on job markets ?
		A:	it is going to happen to migration what it will be the impact on the labour market ?
√	√	**P:**	there will migration and what impact on the labour market is that ?
		R:	to respect and value these differences shows that we still have our feet firmly on the ground .
√		**A:**	it must be prepared to recognize this difference sense of reality .
	√	**P:**	it is a sign of realism , this difference to respect and recognise .
		R:	judging from the survivors and those involved , we can already draw two conclusions .
√		**A:**	we are required to draw two already in relation to the survivors and those conclusions .
	√	**P:**	in relation to the survivors and affected are already two conclusions .
		R:	without wishing to anticipate the results , i can report on some of the trends in the new reports today .
√		**A:**	in order to want the results today , i can already give trends of the new reports .
	√	**P:**	without the results of pre-empt , i can trends of the new reports already today to give .

F	A	ADEQUACY IS THE SAME	
		R:	this is why it is necessary for the economy to be rebuilt .
√	√	**A:**	we have to start with the reconstruction of the economy .
	√	**P:**	therefore , must also with the reconstruction of the economy begun .
		R:	nuclear power stations of first generation soviet construction are a particular danger as far as europe is concerned .
√	√	**A:**	nuclear power stations soviet design of the first generation applies for europe as a particularly dangerous .
√	√	**P:**	nuclear power stations soviet design , the first generation apply for europe as especially dangerous .
		R:	it will retain its specific character . this , moreover , is the wish of those who negotiated the ecsc research programmes with the commission .
√	√	**A:**	in fact , this is the will of those who have negotiated the continuation of the egks-forschungsprogramme with the commission .
	√	**P:**	this was also the will of those who are the continuation of the egks-forschungsprogramme with the commission negotiated .
		R:	this is something which everyone at each stage in the food chain should in fact help with .
√	√	**A:**	this would be at all stages of the food chain good .
√	√	**P:**	this would at all stages of the food chain benefit .
		R:	the candidate states have spared no effort to meet the requirements .
√	√	**A:**	the candidate countries have made enormous efforts in order to meet the requirements .
√	√	**P:**	the candidate countries have made enormous efforts in order to meet the requirements .
		R:	only this afternoon i witnessed a fine demonstration .
√	√	**A:**	precisely this afternoon i have conducted this this on impressive way in mind .
√	√	**P:**	this very afternoon , i was on impressive way brought .
		R:	independent counter-checks are essential in order to avoid errors .
√	√	**A:**	an independent gegenprüfung is essential in order to avoid them mistake .
√	√	**P:**	an independent gegenprüfung is essential to ensure mistake to avoid .
		R:	the kozloduy nuclear power plant accounts for half the internal power generation in bulgaria .
	√	**A:**	about half of the production of bulgaria falls to the nuclear power station koslodui should be adhered .
√	√	**P:**	about half of the production of bulgaria falls to the nuclear power plant in koslodui should be adhered to .
		R:	the regime 's treatment of the nobel laureate , aung san suu kyi , is all of a piece with their treatment of the burmese people .
√	√	**A:**	the other birmesen will also be taken as the nobelpreisträgerin aung san suu kyi .
√	√	**P:**	as the nobelpreisträgerin aung san suu kyi will also be the other birmesen dealt with .
		R:	the only reference to companies is to do with a code of self-regulation .
√	√	**A:**	the only with regard to these companies is the call for the introduction of a selbstkontroll-kodex .
√	√	**P:**	the only with regard to these companies is the call for the introduction of a selbstkontroll-kodex .

F	A	AEP SYSTEM IS MORE FLU, MORE ADE (2 JUDGES)	
		R:	however strict quality criteria need to be laid down at community level for these decontamination procedures .
√	√	**A:**	stringent quality criteria must be set for this dekontaminierungsverfahren at community level .
		P:	for this dekontaminierungsverfahren must , however , at community level strict quality criteria to be established .
		R:	as the commissioner has indicated , the reality is often quite different .
√	√	**A:**	as you have said the reality is often quite different .
		P:	as you said , is the reality is often quite different .
		R:	the fact is that elk hunting is extremely widespread in sweden and finland .
√	√	**A:**	the elchjagd has reached a considerable extent in sweden and finland .
		P:	the elchjagd has in sweden and finland is a considerable proportions .
		R:	amendment no 38 is unacceptable as it falls outside the scope of this directive .
√	√	**A:**	we can not accept amendment no 38 because it does not fall within the scope of the directive .
		P:	amendment no 38 we cannot accept , because it does not in the scope of the directive falls .
		R:	i hope slovenia will be in the first group of new members .
√	√	**A:**	i hope we can welcome slovenia in the first group of the new member states .
		P:	i hope that we slovenia in the first group of the new members welcome that fact .
		R:	as a french woman and citizen , this vision of the world is particularly alien to me .
√	√	**A:**	as a french citizen such a point of view of the world is particularly alien .
		P:	as a french citizen is me such a point of view of the world , particularly alien .
		R:	all three can be accepted for reasons of legal consistency .
√	√	**A:**	all three amendments can be accepted for reasons of legal coherence .
		P:	all three amendments can for reasons of legal coherence adopted .
		R:	the european parliament has failed in its duty to reflect a balanced approach which i very much regret .
√	√	**A:**	the european parliament has not fulfilled its commitment to a balanced approach which i very much regret .
		P:	the european parliament has its commitment to a balanced approach is not fulfilled , which i very much regret .
		R:	all afghan women , whatever their age , are faced with systematic violation of their most fundamental rights .
√	√	**A:**	all the afghan women , whatever their age are facing a systematic violations of their basic rights .
		P:	all the afghan women , irrespective of age , systematic violations of their most basic rights suspended .
		R:	the commission is convinced of the usefulness of a positive list .
√	√	**A:**	in fact , the commission is convinced of the appropriateness of such a positive list .
		P:	the commission is of the appropriateness of such a positive list even convinced .

F	A	PHRASE-BASED SYSTEM MORE FLU, MORE ADE (2 JUDGES)	
		R:	however , i would like to say that this fight is not the exclusive property of the left .
		A:	however , i would also like to say this fight is no domain of the left .
√	√	**P:**	but i would also like to say that this fight is not a domain of the left .
		R:	the information society is important to old as well as new sectors .
		A:	the information society is important for the elderly , as for the new industry .
√	√	**P:**	the information society is important for the old as for the new industry .
		R:	will there be waves of immigrants and what impact will this have on job markets ?
		A:	it is going to happen to migration what it will be the impact on the labour market ?
√	√	**P:**	there will migration and what impact on the labour market is that ?
		R:	this is ground-breaking legislation and it is a gigantic step forward but we must go even further .
		A:	this directive sets all existing legislation and is a major step forward but we need to go .
√	√	**P:**	this directive exceeds all existing legislation and is a tremendous step forward , but we must go further .
		R:	croatia and the region deserve this in the interests of peace within europe .
		A:	croatia and the region it deserves the interests of the european peace .
√	√	**P:**	croatia and the region deserve it in the interests of the european peace .
		R:	the european method , imposing uniformity , is wrong .
		A:	the method of harmonisation at european level is the right way not .
√	√	**P:**	the method of harmonisation at european level is not the right way .
		R:	thank you for the very important statement you have just made .
		A:	we thank you very much interesting statement , which you have just given .
√	√	**P:**	we thank you for the very interesting statement you have just given .
		R:	i also say that very clearly in the context of the two reports we are dealing with here today .
		A:	i am also saying that the very clearly in connection with the two reports , we are dealing today .
√	√	**P:**	i would also say very clearly in connection with the two reports we are debating today .
		R:	prosperity is created by restoring economic freedoms and nothing else .
		A:	what creating rise is the restoration of the economic freedoms , and nothing else .
√	√	**P:**	what creates prosperity , is to restore the economic freedoms and nothing else .
		R:	otherwise i do agree very much with this overall policy and the basic idea behind it .
		A:	otherwise , i agree to a large extent , the overall strategy and the persecuted .
√	√	**P:**	otherwise , i agree with most of the overall strategy and the persecuted line .

F	A	EQUAL FLUENCY, EQUAL ADEQUACY (2 JUDGES)	
		R:	i do not want to encumber this agenda any more , nor do i want to lighten it .
√	√	**A:**	i this do not want to complicate this , which does help but not .
√	√	**P:**	this , i would not even more difficult , but not easier .
		R:	enlargement is also good for the level of social security in europe .
√	√	**A:**	enlargement is also good for the level of social security in europe .
√	√	**P:**	enlargement is also good for the level of social security in europe .
		R:	a common arms exports policy is essential to this process .
√	√	**A:**	a common arms export policy is essential for this process .
√	√	**P:**	a common arms export policy is essential for this process .
		R:	to start , with she could not get into parliament except through the swing doors .
√	√	**A:**	the problems are of achieving them jenny only by the pendeltüren at the european parliament .
√	√	**P:**	the problems fingen already so that jenny only by the pendeltüren into parliament would reach .
		R:	mr president , commissioner , social policy is a cornerstone of the social model within europe .
√	√	**A:**	mr president , commissioner , the social policy is a cornerstone of the european social model .
√	√	**P:**	mr president , commissioner , the social policy is a cornerstone of the european social model .
		R:	it can do that in the domain of foodstuffs , and it must also be able to do it in the domain of animal feed .
√	√	**A:**	you can do this in the area of food and they must also be the in animal feed .
√	√	**P:**	it must in the area of food , and must have the same in the feedingstuffs .
		R:	there we can really see the change that takes place when a country 's democracy is consolidated .
√	√	**A:**	the changes will be here truly clearly that is if a democracy is to prevail in a country .
√	√	**P:**	this is a truly change clearly that is if a democracy in a country prevail .
		R:	the next item is the joint debate on the following motions for resolutions :
√	√	**A:**	the next item is the joint debate on the following motions for resolutions :
√	√	**P:**	the next item is the joint debate on the following motions for resolutions :
		R:	we should therefore remain level-headed and focused in our attempts to create openings for a development which will bring serbia nearer to europe , also from a moral-political point of view .
√	√	**A:**	we must therefore create öffnungen for development soberly , and specifically the serbia , expressed morally and politically in europe .
√	√	**P:**	we must therefore soberly specifically öffnungen for development create the serbia morally and politically to europe brings .
		R:	earlier , the idea of abuse of the provisions was mentioned . i feel that this may provide a major foothold for those most reactionary ideas .
√	√	**A:**	it could as is to support for such rückschrittlichen approach .
√	√	**P:**	it could , it seems to me that on support for such rückschrittlichen approach serve .

References

Alshawi, H. (1996). Head automata and bilingual tiling: translation with minimal representations. In *ACL 96*.

Alshawi, H., Bangalore, S., & Douglas, S. (2000). Learning dependency translation models as collections of finite-state head transducers. *Computational Linguistics*, *26*(1), 45–60.

Bar-Hillel, Y. (1951). The present state of research on mechanical translation. *American Documentation*, *2*(4), 229–237.

Bartlett, P., Collins, M., Taskar, B., & McAllester, D. (2004). Exponentiated gradient algorithms for large-margin structured classification. In *Proceedings of NIPS 2004*.

Brown, P. F., Cocke, J., Pietra, S. A. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., Mercer, R. L., & Roossin, P. S. (1990). A statistical approach to machine translation. *Computational Linguistics*, *16*(2), 79–85.

Brown, P. F., Pietra, S. A. D., Pietra, V. J. D., & Mercer, R. L. (1993). The mathematics of statistical machine translation. *Computational Linguistics*, *22*(1), 39–69.

Callison-Burch, C., Osborne, M., & Koehn, P. (2006). Re-evaluating the role of bleu in machine translation research. In *Proceedings of EACL-2006*.

Charniak, E. (2000). A maximum-entropy-inspired parser. In *Proceedings of the European Chapter of the Association for Computational Linguistics*, (pp. 132–139).

Charniak, E., Knight, K., & Yamada, K. (2001). Syntax-based language models for statistical machine translation. In *ACL 01*.

Chiang, D. (2005). A hierarchical phrase-based model for statistical machine translation. In *ACL 05*.

Collins, M. (1999). *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania.

Collins, M. (2002). Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of Empirical Methods in Natural Language Processing 2002*.

Collins, M., Koehn, P., & Kučerová., I. (2005). Clause restructuring for statistical machine translation. In *ACL 05*.

Collins, M. & Koo, T. (2005). Discriminative reranking for natural language parsing. *Computational Linguistics*, *31*(1), 25–69.

Collins, M. & Roark, B. (2004). Incremental parsing with the perceptron algorithm. In *ACL 04*.

Cortes, C. & Vapnik, V. (1995). Support vector networks. *Machine Learning*, *20*, 273–297.

Cowan, B. & Collins, M. (2005). Morphology and reranking for the statistical parsing of spanish. In *Proceedings of EMNLP 2005*.

Cowan, B., Kučerová, I., & Collins, M. (2006). A discriminative model for tree-to-tree translation. In *Proceedings of EMNLP 2006*.

Daumé, III, H. & Marcu, D. (2005). Learning as search optimization: approximate large margin methods for structured prediction. In *ICML 05*.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, *39*(B), 1–38.

Ding, Y. & Palmer, M. (2005). Machine translation using probabilistic synchronous dependency insertion grammars. In *Proceedings of the 43rd Annual Meeting of the ACL*, (pp. 541–548)., Ann Arbor, MI.

Dorr, B. J. (1993). *Machine Translation: A View from the Lexicon*. Cambridge, Massachusetts: The MIT Press.

Dubey, A. (2005). What to do when lexicalization fails: parsing german with suffix analysis and smoothing. In *ACL 2005*, Ann Arbor, MI.

Eisner, J. (2003). Learning non-isomorphic tree mappings for machine translation. In *ACL 03, Companion Volume*.

Fox, H. J. (2002). Phrasal cohesion and statistical machine translation. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing*, (pp. 304–311).

Frank, R. (2002). *Phrase Structure Composition and Syntactic Dependencies*. MIT Press.

Freund, Y. & Schapire, R. E. (1998). Large margin classification using the perceptron algorithm. *Machine Learning*, *37*(3), 277–296.

Galley, M., Graehl, J., Knight, K., Marcu, D., DeNeefe, S., Wang, W., & Thayer, I. (2006). Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, (pp. 961–968)., Sydney, Australia.

Galley, M., Hopkins, M., Knight, K., & Marcu, D. (2004). What's in a translation rule? In *HLT-NAACL 04*.

Gildea, D. (2003). Loosely tree-based alignment for machine translation. In *ACL 03*.

Graehl, J. & Knight, K. (2004). Training tree transducers. In *NAACL-HLT 04*.

Grimshaw, J. (1991). Extended projection. Master's thesis, Brandeis University, Waltham, MA.

Haegeman, L. & Guéron, J. (1999). *English Grammar: A Generative Perspective*. Oxford, UK: Blackwell Publishers Ltd.

Huang, L., Knight, K., & Joshi, A. (2006). Statistical syntax-directed translation with extended domain of locality. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, (pp. 66–73)., Cambridge, MA.

Hutchins, W. J. & Somers, H. L. (1992). *An Introduction to Machine Translation*. Academic Press.

Jelinek, F. & Mercer, R. (1980). Interpolated estimation of markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, (pp. 381–397)., Amsterdam, The Netherlands.

Johnson, M. (1998). Pcfg models of linguistic tree representations. *Computational Linguistics*, *24*(4), 613–632.

Joshi, A. (1985). *Natural Language Processing—Theoretical, Computational, and Psychological Perspectives*, chapter How much context-sensitivity is necessary for characterizing structural descriptions — tree-adjoining grammars. Cambridge University Press.

Joshi, A. & Schabes, Y. (1996). *Handbook of Formal Languages and Automata*, chapter Tree-Adjoining Grammars. Berlin, Germany: Springer-Verlag.

Koehn, P. (2004). Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *Proceedings of the Sixth Conference of the Association for Machine Translation in the Americas*, (pp. 115–124).

Koehn, P. (2005a). Europarl: A parallel corpus for statistical machine translation. In *MT Summit 05*.

Koehn, P. (2005b). Europarl: A parallel corpus for statistical machine translation. In *Proceedings of MT Summit 2005*.

Koehn, P., Federico, M., Shen, W., Bertoldi, N., Bojar, O., Callison-Burch, C., Cowan, B., Dyer, C., Hoang, H., Zens, R., Constantin, A., Moran, C. C., & Herbst, E. (2007a). Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL 2007*, (pp. 177–180)., Prague, Czech Republic.

Koehn, P., Federico, M., Shen, W., Bertoldi, N., Bojar, O., Callison-Burch, C., Cowan, B., Dyer, C., Hoang, H., Zens, R., Constantin, A., Moran, C. C., & Herbst, E. (2007b). Open source toolkit for statistical machine translation: Factored translation models and confusion network decoding. Technical report, Johns Hopkins University, Center for Speech and Language Processing.

Koehn, P., Och, F. J., & Marcu, D. (2003). Statistical phrase-based translation. In *HLT-NAACL 03*.

Kübler, S., Hinrichs, E. W., & Maier, W. (2006). Is it really that difficult to parse german? In *Proceedings of the 2006 Conference on Empirical Methods in Natura l Language Processing (EMNLP 2006)*, (pp. 111–119)., Sydney, Australia. Association for Computational Linguistics.

Lehmann, E. (1986). *Testing Statistical Hypotheses* (2 ed.). Springer-Verlag.

Liang, P., Bouchard-Côté, A., Klein, D., & Taskar, B. (2006). An end-to-end discriminative approach to machine translation. In *Proceedings of ACL 2006*, (pp. 761–768)., Sydney, Australia.

Lin, D. (2004). A path-based transfer model for machine translation. In *Proceedings of the 20th COLING*.

Manning, C. D. & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.

Marcu, D., Wang, W., Echihabi, A., & Knight, K. (2006). Spmt: Statistical machine translation with syntactified target language phrases. In *Proceedings of EMNLP 2006*, (pp. 44–52)., Sydney, Australia.

Melamed, I. D. (2004). Statistical machine translation by parsing. In *ACL 04*.

Menezes, A. & Quirk, C. (2007). Using dependency order templates to improve generality in translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, (pp. 1–8)., Prague, Czech Republic.

Minsky, M. & Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry*. The MIT Press.

Navarro, B., Civit, M., Martí, M. A., Marcos, R., & Fernández, B. (2003). Syntactic, semantic and pragmatic annotation in cast3lb. In *Shallow Processing of Large Corpora (SProLaC), a Workshop of Corpus Linguistics*, Lancaster, UK.

Nelder, J. & Mead, R. (1964). A simplex method for function minimization. *The Computer Journal*, *7*, 308–313.

Nesson, R. & Shieber, S. (2006). Simpler tag semantics through synchronization. In *Proceedings of the 11th Conference on Formal Grammar*, Malaga, Spain.

Nesson, R., Shieber, S., & Rush, A. (2006). Induction of probabilistic synchronous tree-insertion grammars for machine translation. In *Proceedings of the 7th Conferecne of the Association for Machine translation in the Americas*, (pp. 128–137).

Nesson, R. & Shieber, S. M. (2007). Extraction phenomena in synchronous tag syntax and semantics. In *Proceedings of the Workshop on Syntax and Structure in Statistical Translation*, Rochester, New York.

Nießen, S. & Ney, H. (2001). Morpho-syntactic analysis for reordering in statistical machine translation. In *Proceedings of MT Summit VIII*, (pp. 247–252)., Santiago de Compostela, Galicia, Spain.

Nießen, S. & Ney, H. (2004). Statistical machine translation with scarce resources using morpho-syntactic information. *Computational Linguistics*, *30*(2), 181–204.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, (pp. 160–167).

Och, F. J., Gildea, D., Khudanpur, S., Sarkar, A., Yamada, K., Fraser, A., Kumar, S., Shen, L., Smith, D., Eng, K., Jain, V., Jin, Z., & Radev, D. (2004). A smorgasbord of features for statistical machine translation. In *NAACL-HLT 04*.

Och, F. J. & Ney, H. (2000). A comparison of alignment models for statistical machine translation. In *COLING '00: The 18th International Conference on Computational Linguistics*, (pp. 1086–1090)., Saarbrcken, Germany.

Och, F. J. & Ney, H. (2002). Discriminative training and maximum entropy models for statistical machine translation. In *ACL 2002: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, (pp. 295–302)., Philadelphia, PA. Association for Computational Linguistics.

Och, F. J. & Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, *29*(1), 19–51.

Och, F. J. & Ney, H. (2004). The alignment template approach to statistical machine translation. *Computational Linguistics*, *30*(4), 417–449.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2001). BLEU: a method for automatic evaluation of machine translation. Technical Report RC22176 (W0109-022), IBM T.J. Watson Research Center.

Quirk, C. & Corston-Oliver, S. (2006). The impact of parse quality on syntactically-informed statistical machine translation. In *Proceedings of EMNLP 2006*, (pp. 62–69)., Sydney, Australia.

Quirk, C., Menezes, A., & Cherry, C. (2005). Dependency treelet translation: syntactically informed phrasal smt. In *EACL 05*.

Riezler, S. & Maxwell, J. T. (2006). Grammatical machine translation. In *NAACL-HLT 06*.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*, 386–408. Reprinted in *Neurocomputing* (MIT Press, 1998).

Russell, S. & Norvig, P. (2002). *Artificial Intelligence: A Modern Approach* (2 ed.). Prentice Hall.

Schabes, Y. & Waters, R. (1995). Tree insertion grammar: A cubic-time, parsable formalism that lexicalizes context-free grammar without changing the trees produced. *Computational Linguistics*, *21*(4), 479–513.

Shieber, S. & Schabes, Y. (1990). Synchronous tree-adjoining grammars. In *Proceedings of the 13th conference on Computational Linguistics*, (pp. 253–258)., Helsinki, Finland.

Shieber, S. M. (2007). Probabilistic synchronous tree-adjoining grammars for machine translation: The argument from bilingual dictionaries. In *Proceedings of SSST, NAACL-HLT 2007/AMTA Workshop on Syntax and Structure in Statistical Translation*, (pp. 88–95)., Rochester, New York.

Sipser, M. (1997). *Introduction to the Theory of Computation*. PWS Publishing Company.

Skut, W., Krenn, B., Brants, T., & Uszkoreit, H. (1997). An annotation scheme for free word order languages. In *Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP-97)*, Washington, DC, USA.

Telljohann, H., Hinrichs, E. W., Kübler, S., & Zinsmeister, H. (2005). Stylebook for the tübingen treebank of written german (tü ba-d/z). Seminar für Sprachwissenschaft, Universität Tübing en, German.

Torruella, M. C. (2000). Guía para la anotación morfosintáctica del corpus clic-talp. Technical report, X-Tract Working Paper, WP-00/06.

Torruella, M. C. (2004). 3LB: Guía para la anotación sintáctica de Cast3LB: un corpus del español con anotación sintáctica, semántica y pragmática. Technical report, 3LB-WP 02-01, X-Tract-II WP 03-06 (Versión 4).

Wahlster, W. (1993). Verbmobil: Translation of face-to-face dialogs. In *The Fourth Machine Translation Summit: MT Summit IV*, (pp. 127–135)., Kobe, Japan.

Wang, W., Knight, K., & Marcu, D. (2007). Binarizing syntax trees to improve syntax-based machine translation accuracy. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, (pp. 746–754)., Prague, Czech Republic.

Wu, D. (1997). Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, *23*(3), 377–403.

Xia, F. & McCord, M. (2004). Improving a statistical mt system with automatically learned rewrite patterns. In *COLING 04*.

Yamada, K. & Knight, K. (2001). A syntax-based statistical translation model. In *ACL 01*.

Yamada, K. & Knight, K. (2002). A decoder for syntax-based mt. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, PA.

Zagona, K. (2002). *The Syntax of Spanish*. Cambridge, UK: Cambridge University Press.

Zhang, H., Huang, L., Gildea, D., & Knight, K. (2006). Synchronous binarization for machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*, (pp. 256–263)., New York, NY.