A Discriminative Model for Tree-to-Tree Translation

Brooke Cowan
MIT CSAIL
brooke@csail.mit.edu

Ivona Kučerová
MIT Linguistics Department
kucerova@mit.edu

Michael Collins
MIT CSAIL
mcollins@csail.mit.edu

Abstract

This paper proposes a statistical, tree-to-tree model for producing translations. Two main contributions are as follows: (1) a method for the extraction of syntactic structures with alignment information from a parallel corpus of translations, and (2) 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. (2003).

1 Introduction

Phrase-based approaches (Och and Ney, 2004) to statistical machine translation (SMT) 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 paper 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, with the aim of constructing 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 and Schabes, 1990). A key contribution of this paper is a method for learning to map 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. (2003).

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.

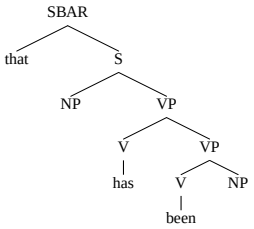
Step 2: An *aligned extended projection* (AEP) is predicted for each German clause. To illustrate this step, consider translation of the second German clause, which has the following parse structure:

Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006), pages 232–241,
Sydney, July 2006.

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

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, that is similar to a TAG elementary tree (see also Figure 2):



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 paper 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.

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 paper, 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 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 5.2 of this paper describes the features

[1] Note that 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.

we use for AEP prediction in translation from German to English. 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. We see this as one of the strengths of the feature-based approach.

In the work presented in this paper, 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.

2 Related Work

There has been a substantial amount of previous work on approaches that make use of syntactic information in statistical machine translation. Wu (1997) and Alshawi (1996) describe early work on formalisms that make use of transductive grammars; Graehl and Knight (2004) describe methods for training tree transducers. Melamed (2004) establishes a theoretical framework for generalized synchronous parsing and translation. Eisner (2003) discusses methods for learning synchronized elementary tree pairs from a parallel corpus of parsed sentences. Chiang (2005) has recently shown significant improvements in translation accuracy, using synchronous grammars. Riezler and Maxwell (2006) describe a method for learning a probabilistic model that maps LFG parse structures in German into LFG parse structures in English.

Yamada and Knight (2001) and Galley et al. (2004) describe methods that make use of syntactic information in the target language alone; Quirk et al. (2005) describe similar methods that make use of dependency representations. Syntactic parsers in the target language have been used as language models in translation, giving some improvement in accuracy (Charniak et al., 2001). The work of Gildea (2003) involves methods that make use of syntactic information in both the source and target languages.

Other work has attempted to incorporate syntactic information through reranking approaches applied to n-best output from phrase-based systems (Och et al., 2004). Another class of approaches has shown improvements in translation through reordering, where source language strings are parsed and then reordered, in an attempt to recover a word order that is closer to the target language (Collins et al., 2005; Xia and McCord, 2004).

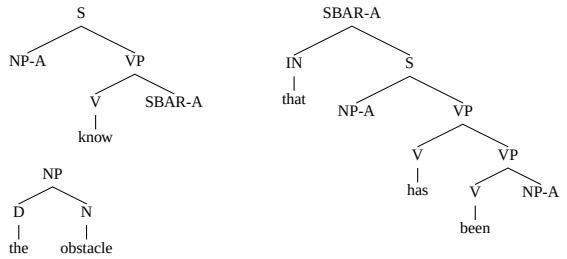


Figure 1: Extended projections for the verbs *know* and *been*, and for the noun *obstacle*. The EPs were taken from the parse tree for the sentence *We know that the main obstacle has been the predictable resistance of manufacturers*.

Our approach is closely related to previous work on synchronous tree adjoining grammars (Shieber and Schabes, 1990; Shieber, 2004), and the work on TAG approaches to syntax described by Frank (2002). A major departure from previous work on synchronous TAGs is in our use of a discriminative model that incrementally predicts the information in the AEP. Note also that our model may include features that take into account any part of the German clause.

3 A Translation Architecture Based on Aligned Extended Projections

3.1 Background: Extended Projections (EPs)

Extended projections (EPs) play a crucial role in the lexicalized tree adjoining grammar (LTAG) (Joshi, 1985) approach to syntax described by Frank (2002). In this paper we focus almost exclusively on extended projections associated with main verbs; note, however, that EPs are typically associated with all content words (nouns, adjectives, etc.). As an example, a parse tree for the sentence *we know that the main obstacle has been the predictable resistance of manufacturers* would make use of EPs for the words *we, know, main, obstacle, been, predictable, resistance*, and *manufacturers*. Function words (in this sentence *that, the, has*, and *of*) do not have EPs; instead, as we describe shortly, each function word is incorporated in an EP of some content word.

Figure 1 has examples of EPs. Each one is an LTAG elementary tree which contains a sin-

gle content word as one of its leaves. Substitution nodes (such as NP-A or SBAR-A) 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.

Elementary trees corresponding to EPs form the basic units in the LTAG approach described by Frank (2002). They are combined to form a full parse tree for a sentence using the TAG operations of substitution and adjunction. For example, the EP for *been* in Figure 1 can be substituted into the SBAR-A position in the EP for *know*; the EP for *obstacle* can be substituted into the subject position of the EP for *been*.

3.2 Aligned Extended Projections (AEPs)

We now build on the idea of extended projections to give a detailed description of AEPs. Figure 2 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 2 has two arguments, indexed as 1 and 2. Each of these modifiers must either have a translation in the corresponding English clause, or must be deleted.

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 VPs, Ss, and SBARs. 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*,[3] 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.

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 an absence of 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.

[2] Note that in this paper we consider translation from German to English; in the remainder of the paper we take *English* to be synonymous with the target language in translation and *German* to be synonymous with the source language.

[3] This happens in the case where there exists a subject in the English clause which is not aligned to a modifier in the German clause. See, for instance, the second example in Figure 2.

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 2: Three examples of German parse trees, together with their aligned extended projections (AEPs) in the training data. Note that 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.

4 Extracting AEPs from a Corpus

A crucial step in our approach is the extraction of training examples from a translation corpus. Each training example consists of a German clause paired with an English AEP (see Figure 2).

In our experiments, we used the Europarl corpus (Koehn, 2005). For each sentence pair from this data, we used a version of the German parser described by Dubey (2005) to parse the German component, and a version of the English parser described by Collins (1999) to parse the English component. To extract AEPs, we perform the following steps:

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 and 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 training process is to identify German/English clause pairs which are translations of each other. We first break each English or German parse tree into a set of clauses; see Appendix A 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, define the English sentence to contain the clause sequence $\langle e_1, e_2, \ldots, e_n \rangle$, and the German sentence to contain 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.[4]

[4] 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.

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.

An AEP can then be extracted from each clause pair. The EP for the English 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 The Model

5.1 Beam search and the perceptron

In this section we describe linear history-based models with beam search, and the perceptron algorithm for learning in 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 3. 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 $\bar{\phi}(x, \langle d_1, \ldots, d_{i-1} \rangle, d_i) \in \mathbb{R}^d$ 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 $\bar{\alpha} \in \mathbb{R}^d$ 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:

$$\text{SCORE}(x, y) = \Phi(x, y) \cdot \bar{\alpha} \quad (1)$$

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

$$F(x) = \arg \max_{y \in \text{GEN}(x)} \text{SCORE}(x, y) \quad (2)$$

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 Eq. 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 by Collins (2002). This combination of the perceptron algorithm with beam-search is similar to that described by Collins and Roark (2004).[5] The perceptron algorithm is a convenient choice because it converges quickly — usually taking only a few iterations over the training set (Collins, 2002; Collins and Roark, 2004).

5.2 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 $\bar{\phi}$

[5] Future work may consider alternative algorithms, such as those described by Daumé and 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 1: Functions of the German clause used for making features in the AEP prediction model.

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 1 and 2). We use various combinations of these basic functions in the prediction of each decision d_i, as described below.

STEM: Features for the prediction of STEM conjoin the value of this variable with each of the functions in lines 1–13 of Table 1. For example, one feature is the value of STEM conjoined with the main verb of the German clause. In addition, $\bar{\phi}$ includes features sensitive to the rank of a candidate stem in an externally-compiled lexicon.[6]

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 2, conjoined in various ways with the functions in lines 4, 12, and 14–21 of Table 1. Note that the functions in Table 2 allow us to look

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

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

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 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 3), then $\bar{\phi}$ looks at the nonterminal label of the German node indexed by i as well as the surrounding context in the German clausal tree. Otherwise, $\bar{\phi}$ looks at the value of SUBJECT. These basic features are combined with the functions in lines 1, 3, and 24–27 of Table 1.

OBJECT: We make similar features to those for the prediction of SUBJECT. In addition, $\bar{\phi}$ 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 1.

MODALS: For the prediction of MODALS, $\bar{\phi}$ looks at MODALS, SPINE, and STEM, conjoined with the functions in lines 2–5 and 12 of Table 1.

INFL: The features for INFL include the values of INFL, MODALS, and SUBJECT, and VOICE, and the function in line 8 of Table 2.

MOD(i): For the MOD(i) variables, $\bar{\phi}$ 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 1.

6 Deriving Full Translations

As we described in section 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 2 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 paper, 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 B for a full description of the features used in the modifier translation model. Note that 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 2 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 `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 2 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>`. Note that modifiers assigned `post-verb` 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.

7 Experiments

We applied the approach to translation from German to English, using the Europarl corpus (Koehn, 2005) 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 4. 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 5.2. This set was optimized using the development data through experimentation with several different feature subsets.

Modifiers within German clauses were translated using the phrase-based model of Koehn et al. (2003). We first generated n-best lists for each modifier. We then built a reranking model—see section 6—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.

8 Conclusions and Future Work

We have presented an approach to tree-to-tree based translation which models a new representation—aligned extended projections—within a discriminative, feature-based framework. Our model makes use of an explicit representation of syntax in the target language, together with constraints on the alignments between source and target parse trees.

The current system presents many opportunities for future work. For example, improvement in accuracy may come from a tighter integration of modifier translation into the overall translation process. The current method—using an n-best reranking model to select the best candidate—chooses each modifier independently and then places it into the translation. We intend to explore an alternative method that combines finite-state machines representing the n-best output from the phrase-based system with finite-state machines representing the complementizers, verbs, modals, and other substrings of the translation derived from the AEP. Selecting modifiers using this representation would correspond to searching the finite-state network for the most likely path. A finite-state representation has many advantages, including the ability to easily incorporate an n-gram language model.

Future work may also consider expanded definitions of AEPs. For example, we might consider AEPs that include larger chunks of phrase structure, or we might consider AEPs that contain more detailed information about the relative ordering of modifiers. There is certainly room for improvement in the accuracy with which AEPs are predicted in our data; the feature-driven approach allows a wide range of features to be tested. For example, it would be relatively easy to incorporate a syntactic language model (i.e., a prior distribution over AEP structures) induced from a large amount of English monolingual data.

Appendix A: Identification of Clauses

In the English parse trees, we identify clauses 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 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 B: Reranking Modifier Translations

The n-best reranking model for the translation of modifiers 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 3 and 4.

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 candidate 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 3: 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
3	voice
4	subject prediction
5	German input string

Table 4: Functions of the German input string and predicted AEP output used for making features in the n-best reranking model.

Acknowledgements

We would like to thank Luke Zettlemoyer, Regina Barzilay, Ed Filisko, and Ben Snyder for their valuable comments and help during the writing of this paper. Thanks also to Jason Rennie and John Barnett for providing human judgments of the translation output. This work was funded by NSF grants IIS-0347631, IIS-0415030, and DMS-0434222, as well as a grant from NTT, Agmt. Dtd. 6/21/1998.

References

H. Alshawi. 1996. Head automata and bilingual tiling: translation with minimal representations. *ACL 96*.

P. Bartlett, M. Collins, B. Taskar, and D. McAllester. 2004. Exponentiated gradient algorithms for large-margin structured classification. *Proceedings of NIPS 2004*.

P. Brown, S. Della Pietra, V. Della Pietra, and R. Mercer. 1993. The mathematics of statistical machine translation. *Computational Linguistics*, 22(1):39–69.

E. Charniak, K. Knight, and K. Yamada. 2001. Syntax-based language models for statistical machine translation. *ACL 01*.

D. Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. *ACL 05*.

M. Collins. 1999. Head-Driven Statistical Models for Natural Language Parsing. University of Pennsylvania.

M. Collins. 2002. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. *EMNLP 02*.

M. Collins and B. Roark. 2004. Incremental parsing with the perceptron algorithm. *ACL 04*.

M. Collins, P. Koehn, and I. Kučerová. 2005. Clause restructuring for statistical machine translation. *ACL 05*.

H. Daumé III and D. Marcu. 2005. Learning as search optimization: approximate large margin methods for structured prediction. *ICML 05*.

A. Dubey. 2005. What to do when lexicalization fails: parsing German with suffix analysis and smoothing. *ACL 05*.

J. Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. *ACL 03, Companion Volume*.

R. Frank. 2002. *Phrase Structure Composition and Syntactic Dependencies*. Cambridge, MA: MIT Press.

M. Galley, M. Hopkins, K. Knight, and D. Marcu. 2004. What's in a translation rule? *HLT-NAACL 04*.

D. Gildea. 2003. Loosely tree-based alignment for machine translation. *ACL 03*.

J. Graehl and K. Knight. 2004. Training tree transducers. *NAACL-HLT 04*.

A. Joshi. 1985. How much context-sensitivity is necessary for characterizing structural descriptions – tree-adjoining grammar. Cambridge University Press.

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical phrase based translation. *HLT-NAACL 03*.

P. Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. *MT Summit 05*.

I. D. Melamed 2004. Statistical machine translation by parsing. *ACL 04*.

F. J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, D. Radev. 2004. A smorgasbord of features for statistical machine translation. *HLT/NAACL 04*

F. J. Och and H. Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449.

F. J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

C. Quirk, A. Menezes, and C. Cherry. 2005. Dependency tree translation: syntactically informed phrasal SMT. *EACL 05*.

S. Riezler and J. Maxwell. 2006. Grammatical machine translation. In *NLT-NAACL 06*.

S. Shieber. 2004. Synchronous grammars as tree transducers. In *Proceedings of the Seventh International Workshop on Tree Adjoining Grammar and Related Formalisms*.

S. Shieber and Y. Schabes. 1990. Synchronous tree-adjoining grammars. In *Proceedings of the 13th International Conference on Computational Linguistics*.

D. Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403.

F. Xia and M. McCord. 2004. Improving a statistical MT system with automatically learned rewrite patterns. *COLING 04*.

K. Yamada and K. Knight. 2001. A syntax-based statistical translation model. *ACL 01*.