Non-Projective Parsing for Statistical Machine Translation

Xavier Carreras **Michael Collins**
MIT CSAIL, Cambridge, MA 02139, USA
{carreras,mcollins}@csail.mit.edu

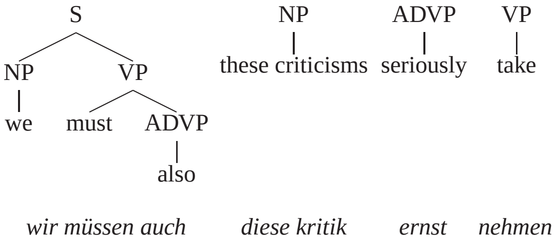
Abstract

We describe a novel approach for syntax-based statistical MT, which builds on a variant of tree adjoining grammar (TAG). Inspired by work in discriminative dependency parsing, the key idea in our approach is to allow highly flexible reordering operations during parsing, in combination with a discriminative model that can condition on rich features of the source-language string. Experiments on translation from German to English show improvements over phrase-based systems, both in terms of BLEU scores and in human evaluations.

1 Introduction

Syntax-based models for statistical machine translation (SMT) have recently shown impressive results; many such approaches are based on either synchronous grammars (e.g., (Chiang, 2005)), or tree transducers (e.g., (Marcu et al., 2006)). This paper describes an alternative approach for syntax-based SMT, which directly leverages methods from non-projective dependency parsing. The key idea in our approach is to allow highly flexible reordering operations, in combination with a discriminative model that can condition on rich features of the source-language input string.

Our approach builds on a variant of tree adjoining grammar (TAG; (Joshi and Schabes, 1997)) (specifically, the formalism of (Carreras et al., 2008)). The models we describe make use of phrasal entries augmented with subtrees that provide syntactic information in the target language. As one example, when translating the sentence *wir müssen auch diese kritik ernst nehmen* from German into English, the following sequence of syntactic phrasal entries might be used (we show each English syntactic fragment above its associated German sub-string):



TAG parsing operations are then used to combine these fragments into a full parse tree, giving the final English translation *we must also take these criticisms seriously*.

Some key aspects of our approach are as follows:

- We impose no constraints on entries in the phrasal lexicon. The method thereby retains the full set of lexical entries of phrase-based systems (e.g., (Koehn et al., 2003)).[1]
- The model allows a straightforward integration of lexicalized syntactic language models—for example the models of (Charniak, 2001)—in addition to a surface language model.
- The operations used to combine tree fragments into a complete parse tree are significant generalizations of standard parsing operations found in TAG; specifically, they are modified to be highly flexible, potentially allowing any possible permutation (reordering) of the initial fragments.

As one example of the type of parsing operations that we will consider, we might allow the tree fragments shown above for *these criticisms* and *take* to be combined to form a new structure with the sub-string *take these criticisms*. This step in the derivation is necessary to achieve the correct English word order, and is novel in a couple of respects: first, *these criticisms* is initially seen to the left of *take*, but after the adjunction this order is reversed; second, and more unusually, the treelet for *seriously* has been skipped over, with the result that the German words translated at this point (*diese*, *kritik*, and *nehmen*) form a non-contiguous sequence. More generally, we will allow any two

[1] Note that in the above example each English phrase consists of a completely connected syntactic structure; this is not, however, a required constraint, see section 3.2 for discussion.

tree fragments to be combined during the translation process, irrespective of the reorderings which are introduced, or the non-projectivity of the parsing operations that are required.

The use of flexible parsing operations raises two challenges that will be a major focus of this paper. First, these operations will allow the model to capture complex reordering phenomena, but will in addition introduce many spurious possibilities. Inspired by work in discriminative dependency parsing (e.g., (McDonald et al., 2005)), we add probabilistic constraints to the model through a discriminative model that links lexical dependencies in the target language to features of the source language string. We also investigate hard constraints on the dependency structures that are created during parsing. Second, there is a need to develop efficient decoding algorithms for the models. We describe approximate search methods that involve a significant extension of decoding algorithms originally developed for phrase-based translation systems.

Experiments on translation from German to English show a 0.5% improvement in BLEU score over a phrase-based system. Human evaluations show that the syntax-based system gives a significant improvement over the phrase-based system. The discriminative dependency model gives a 1.5% BLEU point improvement over a basic model that does not condition on the source language string; the hard constraints on dependency structures give a 0.8% BLEU improvement.

2 Relationship to Previous Work

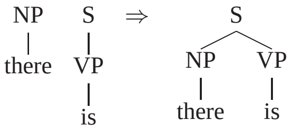
A number of syntax-based translation systems have framed translation as a parsing problem, where search for the most probable translation is achieved using algorithms that are generalizations of conventional parsing methods. Early examples of this work include (Alshawi, 1996; Wu, 1997); more recent models include (Yamada and Knight, 2001; Eisner, 2003; Melamed, 2004; Zhang and Gildea, 2005; Chiang, 2005; Quirk et al., 2005; Marcu et al., 2006; Zollmann and Venugopal, 2006; Nesson et al., 2006; Cherry, 2008; Mi et al., 2008; Shen et al., 2008). The majority of these methods make use of synchronous grammars, or tree transducers, which operate over parse trees in the source and/or target languages. Reordering rules are typically specified through rotations or transductions stated at the level of context-free rules, or larger fragments, within parse trees. These rules can be learned automatically from corpora.

A critical difference in our work is to allow arbitrary reorderings of the source language sentence (as in phrase-based systems), through the use of flexible parsing operations. Rather than stating reordering rules at the level of source or target language parse trees, we capture reordering phenomena using a discriminative dependency model. Other factors that distinguish us from previous work are the use of all phrases proposed by a phrase-based system, and the use of a dependency language model that also incorporates constituent information (although see (Charniak et al., 2003; Shen et al., 2008) for related approaches).

3 A Syntactic Translation Model

3.1 Background

Our work builds on the variant of tree adjoining grammar (TAG) introduced by (Carreras et al., 2008). In this formalism the basic units in the grammar are spines, which associate tree fragments with lexical items. These spines can be combined using a *sister-adjunction* operation (Rambow et al., 1995), to form larger pieces of structure.[2] For example, we might have the following operation:



In this case the spine for *there* has sister-adjoined into the S node in the spine for *is*; we refer to the spine for *there* as being the modifier spine, and the spine for *is* being the head spine. There are close connections to dependency formalisms: in particular in this operation we see a lexical dependency between the modifier word *there* and the head word *is*. It is possible to define syntactic language models, similar to (Charniak, 2001), which associate probabilities with these dependencies, roughly speaking of the form $P(w_m, s_m|w_h, s_h, pos, \sigma)$, where w_m and s_m are the identities of the modifier word and spine, w_h and s_h are the identities of the head word and spine, pos is the position in the head spine that is being adjoined into, and σ is some additional state (e.g., state that tracks previous modifiers that have adjoined into the same spine).

[2] We also make use of the r-adjunction operation defined in (Carreras et al., 2008), which, together with sister-adjunction, allows us to model the full range of structures found in the Penn treebank.

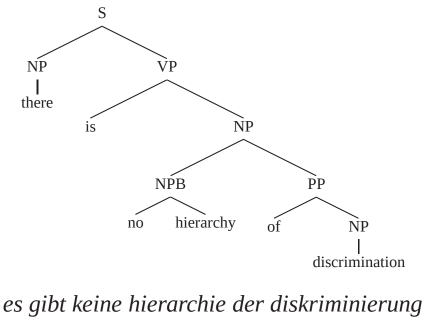
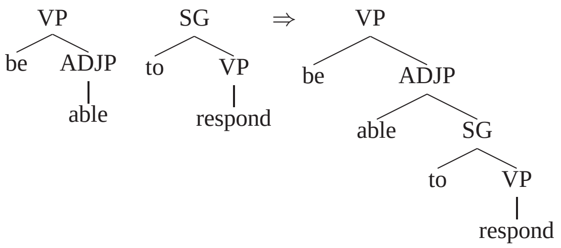


Figure 1: A training example consisting of an English (target language) tree and a German (source language) sentence.

In this paper we will also consider *treelets*, which are a generalization of spines, and which allow lexical entries that include more than one word. These treelets can again be combined using a sister-adjunction operation. As an example, consider the following operation:



In this case the treelet for *to respond* sister-adjoins into the treelet for *be able*. This operation introduces a bi-lexical dependency between the modifier word *to* and the head word *able*.

3.2 S-phrases

This section describes how phrase entries from phrase-based translation systems can be modified to include associated English syntactic structures. These syntactic phrase-entries (from here on referred to as "s-phrases") will form the basis of the translation models that we describe.

We extract s-phrases from training examples consisting of a source-language string paired with a target-language parse tree. For example, consider the training example in figure 1. We assume some method that enumerates a set of possible *phrase entries* for each training example: each phrase entry is a pair $\langle (i,j),(k,l) \rangle$ specifying that source-language words $f_i \ldots f_j$ correspond to target-language words $e_k \ldots e_l$ in the example. For example, one phrase entry for the example might be $\langle (1,2),(1,2) \rangle$, representing the pair $\langle$*es gibt* $\Rightarrow$ *there is*$\rangle$. In our experiments we use standard methods in phrase-based systems (Koehn et al., 2003) to define the set of phrase entries for each sentence in training data.

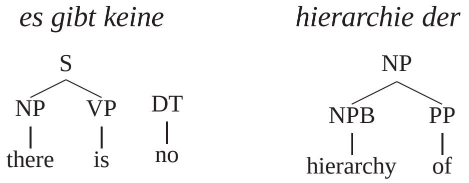
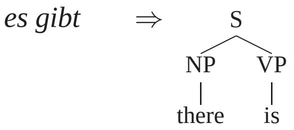


Figure 2: Example syntactic phrase entries. We show German sub-strings above their associated sequence of treelets.[4]

For each phrase entry, we add syntactic information to the English string. To continue our example, the resulting entry would be as follows:



To give a more formal description of how syntactic structures are derived for phrases, first note that each parse tree t is mapped to a TAG derivation using the method described in (Carreras et al., 2008). This procedure uses the head finding rules of (Collins, 1997). The resulting derivation consists of a TAG spine for each word seen in the sentence, together with a set of adjunction operations which each involve a modifier spine and a head spine. Given an English string $\mathbf{e} = e_1 \ldots e_n$, with an associated parse tree t, the syntactic structure associated with a substring $e_k \ldots e_l$ (e.g., *there is*) is then defined as follows:

- For each word in the English sub-string, include its associated TAG spine in t.
- In addition, include any adjunction operations in t where both the head and modifier word are in the sub-string $e_j \ldots e_k$.

In the above example, the resulting structure (i.e., the structure for *there is*) is a single treelet. In other cases, however, we may get a sequence of treelets, which are disconnected from each other. For example, another likely phrase-entry for this training example is $\langle$*es gibt keine* $\Rightarrow$ *there is no*$\rangle$ resulting in the first lexical entry in figure 2, which has two treelets. Allowing s-phrases with multiple treelets ensures that all phrases used by phrase-based systems can be used within our approach.

As a final step, we add additional *alignment* information to each s-phrase. Consider an s-phrase which contains source-language words $f_1 \ldots f_n$ paired with target-language words $e_1 \ldots e_m$. The alignment information is a vector $\langle (a_1,b_1) \ldots (a_m,b_m) \rangle$ that specifies for each word e_i its alignment to words $f_{a_i} \ldots f_{b_i}$ in the source language. For example, for the phrase entry ⟨*es gibt* ⇒ *there is*⟩ a correct alignment would be $\langle(1,1),(2,2)\rangle$, specifying that *there* is aligned to *es*, and *is* is aligned to *gibt* (note that in many, but not all, cases $a_i = b_i$, i.e., a target language word is aligned to a single source language word).

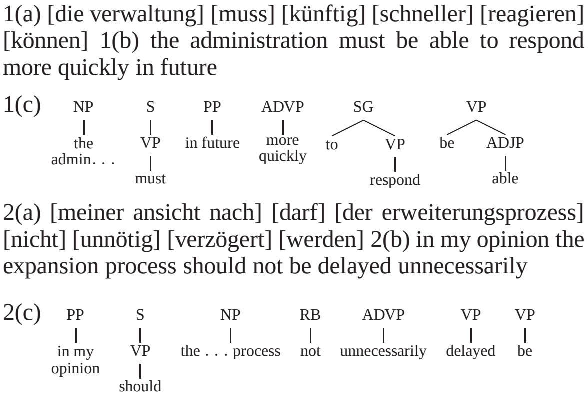
The alignment information in s-phrases will be useful in tying syntactic dependencies created in the target language to positions in the source language string. In particular, we will consider discriminative models (analogous to models for dependency parsing, e.g., see (McDonald et al., 2005)) that estimate the probability of target-language dependencies conditioned on properties of the source-language string. Alignments may be derived in a number of ways; in our method we directly use phrase entries proposed by a phrase-based system. Specifically, for each target word e_i in a phrase entry $\langle f_1 \dots f_n, e_1 \dots e_m\rangle$ for a training example, we find the smallest[5] phrase entry in the same training example that includes e_i on the target side, and is a subset of $f_1 \dots f_n$ on the source side; the word e_i is then aligned to the subset of source language words in this "minimal" phrase.

In conclusion, s-phrases are defined as follows:

Definition 1 *An s-phrase is a 4-tuple* $\langle f, e, t, a\rangle$ *where:* f *is a sequence of foreign words;* e *is a sequence of English words;* t *is a sequence of treelets specifying a TAG spine for each English word, and potentially some adjunctions between these spines; and* a *is an alignment. For an s-phrase* q *we will sometimes refer to the 4 elements of* q *as* $f(q)$, $e(q)$, $t(q)$ *and* $a(q)$.

3.3 The Model

We now introduce a model that makes use of s-phrases, and which is flexible in the reorderings that it allows. To provide some intuition, and some motivation for the use of reordering operations, figure 3 gives several examples of German strings which have different word orders from English.

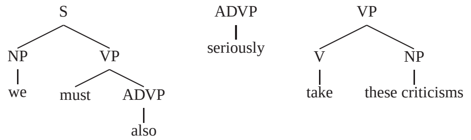


1(a) [die verwaltung] [muss] [künftig] [schneller] [reagieren] [können] 1(b) the administration must be able to respond more quickly in future

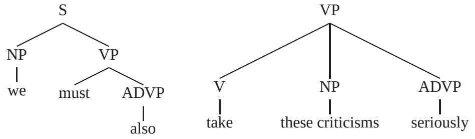
2(a) [meiner ansicht nach] [darf] [der erweiterungsprozess] [nicht] [unnötig] [verzögert] [werden] 2(b) in my opinion the expansion process should not be delayed unnecessarily

Figure 3: Examples of translations. In each example (a) is the original German string, with a possible segmentation marked with "[" and "]"; (b) is a translation for (a); and (c) is a sequence of phrase entries, including syntactic structures, for the segmentation given in (a).

The crucial idea will be to use TAG adjunction operations to combine treelets to form a complete parse tree, but with a complete relaxation on the order in which the treelets are combined. For example, consider again the example given in the introduction to this paper. In the first step of a derivation that builds on these treelets, the treelet for *these criticisms* might adjoin into the treelet for *take*, giving the following new sequence:



In the next derivation step *seriously* is adjoined to the right of *take*, giving the following treelets:



In the final step the second treelet adjoins into the VP above *must*, giving a parse tree for the string *we must also take these criticisms seriously*, and completing the translation.

Formally, given an input sentence $\mathbf{f}$, a derivation d is a pair $\langle \mathbf{q}, \pi\rangle$ where:

- $\mathbf{q} = q_1 \dots q_n$ is a sequence of s-phrases such that $\mathbf{f} = f(q_1) \oplus f(q_2) \oplus \dots \oplus f(q_n)$ (where $u \oplus v$ denotes the concatenation of strings u and v).
- π is a set of adjunction operations that connects the sequence of treelets contained in $\langle t(q_1), t(q_2), \dots, t(q_n)\rangle$ into a parse tree in the target language. The operations allow a complete relaxation of word order, potentially allowing any of the $n!$ possible orderings of the n s-phrases. We make use of both sister-adjunction and r-adjunction operations, as defined in (Carreras et al., 2008).[6]

[5] The "size" of a phrase entry is defined to be $n_s + n_t$ where n_s is the number of source language words in the phrase, n_t is the number of target language words.

[6] In principle we allow any treelet to adjoin into any other treelet—for example there are no hard, grammar-based constraints ruling out the combination of certain pairs of nonterminals. Note however that in some cases operations will have probability 0 under the syntactic language model introduced later in this section.

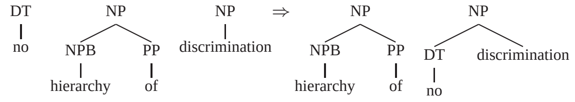


Figure 4: A spurious derivation step. The treelets arise from *[keine] [hierarchie der] [diskriminierung]*.

Given a derivation $d = \langle \mathbf{q}, \pi \rangle$, we define $e(d)$ to be the target-language string defined by the derivation, and $t(d)$ to be the complete target-language parse tree created by the derivation. The most likely derivation for a foreign sentence $\mathbf{f}$ is $\arg\max_{d \in G(\mathbf{f})} score(d)$, where $G(\mathbf{f})$ is the set of possible derivations for $\mathbf{f}$, and the score for a derivation is defined as[7]

$$\begin{aligned} score(d) &= score_{LM}(e(d)) + score_{SYN}(t(d)) \\ &+ score_R(d) + \sum_{j=1}^{n} score_P(q_j) \end{aligned} \quad (1)$$

The components of the model are as follows:

- $score_{LM}(e(d))$ is the log probability of the English string under a trigram language model.
- $score_{SYN}(t(d))$ is the log probability of the English parse tree under a syntactic language model, similar to (Charniak, 2001), that associates probabilities with lexical dependencies.
- $score_R(d)$ will be used to score the parsing operations in π, based on the source-language string and the alignments in the s-phrases. This part of the model is described extensively in section 4.1 of this paper.
- $score_P(q)$ is the score for an s-phrase q. This score is a log-linear combination of various features, including features that are commonly found in phrase-based systems: for example $\log P(f(q)|e(q))$, $\log P(e(q)|f(q))$, and lexical translation probabilities. In addition, we include a feature $\log P(t(q)|f(q), e(q))$, which captures the probability of the phrase in question having the syntactic structure $t(q)$.

Note that a model that includes the terms $score_{LM}(e(d))$ and $\sum_{j=1}^{n} score_P(q_j)$ alone would essentially be a basic phrase-based model (with no distortion terms). The terms $score_{SYN}(t(d))$ and $score_R(d)$ add syntactic information to this basic model.

A key motivation for this model is the flexibility of the reordering operations that it allows. However, the approach raises two major challenges:

Constraints on reorderings. Relaxing the operations in the parsing model will allow complex reorderings to be captured, but will also introduce many spurious possibilities. As one example, consider the derivation step shown in figure 4. This step may receive a high probability from a syntactic or surface language model—*no discrimination* is a quite plausible NP in English—but it should be ruled out for other reasons, for example because it does not respect the dependencies in the original German (i.e., *keine*/no is not a modifier to *diskriminierung*/discrimination in the German string). The challenge will be to develop either hard constraints which rule out spurious derivation steps such as these, or soft constraints, encapsulated in $score_R(d)$, which penalize them.

Efficient search. Exact search for the derivation which maximizes the score in Eq. 1 cannot be accomplished efficiently using dynamic programming (as in phrase-based systems, it is easy to show that the decoding problem is NP-complete). Approximate search methods will be needed.

The next two sections of this paper describe solutions to these two challenges.

4 Constraints on Reorderings

4.1 A Discriminative Dependency Model

We now describe the model $score_R$ introduced in the previous section. Recall that π specifies k adjunction operations that are used to build a full parse tree, where $k \geq n$ is the number of treelets within the sequence of s-phrases $\mathbf{q} = \langle q_1 \dots q_n \rangle$.

Each of the k adjunction operations creates a dependency between a modifier word w_m within a phrase q_m, and a head word w_h within a phrase q_h. For example, in the example in section 3.3 where *these criticisms* was combined with *take*, the modifier word is *criticisms* and the head word is *take*. The modifier and head words have TAG spines s_m and s_h respectively. In addition we can define (a_m, b_m) to be the start and end indices of the words in the foreign string to which the word w_m is aligned; this information can be recovered because the s-phrase q_m contains alignment information for all target words in the phrase, including w_m. Similarly, we can define (a_h, b_h) to be alignment information for the head word w_h. Finally, we can define ρ to be a binary flag specifying whether or not the adjunction operation involves reordering (in the *take criticism* example, this flag is set to `true`, because the order in En-

[7] In practice, MERT training (Och, 2003) will be used to train relative weights for the different model components.

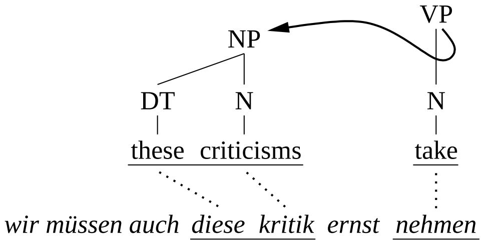


Figure 5: An adjunction operation that involves the modifier *criticisms* and the head *take*. The phrases involved are underlined; the dotted lines show alignments within s-phrases between English words and positions in the German string. The Γ-dependency in this case includes the head and modifier words, together with their spines, and their alignments to positions in the German string (*kritik* and *nehmen*).

glish is reversed from that in German). This leads to the following definition:

Definition 2 *Given a derivation* $d = \langle \mathbf{q}, \pi \rangle$, *we define* $\Gamma(d)$ *to be the set of* Γ-dependencies *in* d. *Each* Γ*-dependency is a tuple* $\langle w_m, s_m, a_m, b_m, w_h, s_h, a_h, b_h, \rho \rangle$ *of elements as described above.*

Figure 5 gives an illustration of how an adjunction creates one such Γ-dependency.

The model is then defined as

$$score_R(d) = \sum_{\gamma \in \Gamma(d)} score_r(\gamma, \mathbf{f})$$

where $score_r(\gamma, \mathbf{f})$ is a score associated with the Γ-dependency γ. This score can potentially be sensitive to any information in γ or the source-language string **f**; in particular, note that the alignment indices (a_m, b_m) and (a_h, b_h) essentially anchor the target-language dependency to positions in the source-language string, allowing the score for the dependency to be based on features that have been widely used in discriminative dependency parsing, for example features based on the proximity of the two positions in the source-language string, the part-of-speech tags in the surrounding context, and so on. These features have been shown to be powerful in the context of regular dependency parsing, and our intent is to leverage them in the translation problem.

In our model, we define $score_r$ as follows. We estimate a model $P(y|\gamma, \mathbf{f})$ where $y \in \{-1, +1\}$, and $y = +1$ indicates that a dependency does exist between w_m and w_h, and $y = -1$ indicates that a dependency does not exist. We then define

$$score_r(\gamma, \mathbf{f}) = \log P(+1|\gamma, \mathbf{f})$$

To estimate $P(y|\gamma, \mathbf{f})$, we first extract a set of labeled training examples of the form $\langle y_i, \gamma_i, \mathbf{f}_i \rangle$ for $i = 1 \ldots N$ from our training data as follows: for each pair of target-language words (w_m, w_h) seen in the training data, we can extract associated spines (s_m, s_h) from the relevant parse tree, and also extract a label y indicating whether or not a head-modifier dependency is seen between the two words in the parse tree. Given an s-phrase in the training example that includes w_m, we can extract alignment information (a_m, b_m) from the s-phrase; we can extract similar information (a_h, b_h) for w_h. The end result is a training example of the form $\langle y, \gamma, \mathbf{f} \rangle$.[8] We then estimate $P(y|\gamma, \mathbf{f})$ using a simple backed-off model that takes into account the identity of the two spines, the value for the flag r, the distance between (a_m, b_m) and (a_h, b_h), and part-of-speech information in the source language.

4.2 Contiguity of π-Constituents

We now describe a second type of constraint, which limits the amount of non-projectivity in derivations. Consider again the k adjunction operations in π, which are used to connect treelets into a full parse tree. Each adjunction operation involves a head treelet that *dominates* a modifier treelet. Thus for any treelet t, we can consider its *descendants*, that is, the entire set of treelets that are directly or indirectly dominated by t. We define a π*-constituent* for treelet t to be the subset of source-language words dominated by t and its descendants. We then introduce the following constraint on π-constituents:

Definition 3 *(*π*-constituent constraint.) A* π*-constituent is* contiguous *iff it consists of a contiguous sequence of words in the source language. A derivation* π *satisfies the* π*-constituent constraint iff all* π*-constituents that it contains are contiguous.*

In this paper we constrain all derivations to satisfy the π-constituent constraint (future work may consider probabilistic versions of the constraint).

The intuition behind the constraint deserves more discussion. The constraint specifies that the modifiers to each treelet can appear in any order around the treelet, with arbitrary reorderings or non-projective operations. However, once a treelet has taken all its modifiers, the resulting π-constituent must form a contiguous sub-sequence

[8] To be precise, there may be multiple (or even zero) s-phrases which include w_m or w_h, and these s-phrases may include conflicting alignment information. Given n_m different alignments seen for w_m, and n_h different alignments seen for w_h, we create $n_m \times n_h$ training examples, which include all possible combinations of alignments.

of the source-language string. As one set of examples, consider the translations in figure 3, and the example given in the introduction. These examples involve reordering of arguments and adjuncts within clauses, a very common case of reordering in translation from German to English. The reorderings in these translations are quite flexible, but in all cases satisfy the π-constituent constraint.

As an illustration of a derivation that violates the constraint, consider again the derivation step shown in figure 4. This step has formed a partial hypothesis, *no discrimination*, which corresponds to the German words *keine* and *diskriminierung*, which do not form a contiguous substring in the German. Consider now a complete derivation, which derives the string *there is hierarchy of no discrimination*, and which includes the π-constituent *no discrimination* shown in the figure (i.e., where the treelet *discrimination* takes *no* as its only modifier). This derivation will violate the π-constituent constraint.[9]

5 Decoding

We now describe decoding algorithms for the syntactic models: we first describe inference rules that are used to combine pieces of structure, and then describe heuristic search algorithms that use these inference rules. Throughout this section, for brevity and simplicity, we describe algorithms that apply under the assumption that each s-phrase has a single associated treelet. The generalization to the case where an s-phrase may have multiple treelets is discussed in section 5.3.

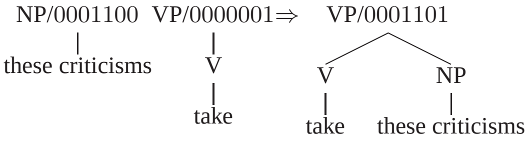
5.1 Inference Rules

Parsing operations for the TAG grammars described in (Carreras et al., 2008) are based on the dynamic programming algorithms in (Eisner, 2000). A critical idea in dynamic programming algorithms such as these is to associate constituents in a chart with *spans* of the input sentence, and to introduce inference rules that combine constituents into larger pieces of structure. The crucial step in generalizing these algorithms to the non-projective case, and to translation, will be to make use of *bit-strings* that keep track of which words in the German have already been translated in a chart entry. To return to the example from the introduction, again assume that the selected s-phrases segment the German input into *[wir müssen auch] [diese kritik] [ernst] [nehmen]*, and the treelets are as shown in the introduction. Each of these treelets will form a basic entry in the chart, and will have an associated bit-string indicating which German words have been translated by that entry.

```
Data structures: $\mathcal{Q}_i$ for $i = 1 \ldots n$ is a set of hypotheses
  for each length $i$, $\mathcal{S}$ is a set of chart entries
$\mathcal{S} \leftarrow \emptyset$
Initialize $\mathcal{Q}_1 \ldots \mathcal{Q}_n$ with basic chart entries derived
  from phrase entries
**For** $i = 1 \ldots n$
  **For** any $A \in$ BEAM$(\mathcal{Q}_i)$
      **If** $\mathcal{S}$ contains a chart entry with the same signature
        as $A$, and which has a higher inside score,
         **continue**
      **Else**
         Add $A$ to $\mathcal{S}$
         For any chart entry $C$ that can be derived from
           $A$ together with another chart entry $B \in \mathcal{S}$,
           add $C$ to the set $\mathcal{Q}_j$ where $j =$ length$(C)$
**Return** $\mathcal{Q}_n$, a set of items of length $n$
```

Figure 6: A beam search algorithm. A dynamic-programming *signature* consists of the regular dynamic-programming state for the parsing algorithm, together with the span (bit-string) associated with a constituent.

These basic chart entries can then be combined to form larger pieces of structure. For example, the following inferential step is possible:



We have shown the bit-string representation for each consituent: for example, the new constituent has the bit-string 0001101 representing the fact that the non-contiguous sub-strings *diese kritik* and *nehmen* have been translated at this point. Any two constituents can be combined, providing that the logical AND of their bit-strings is all 0's.

Inference steps such as that shown above will have an associated score corresponding to the TAG adjunction that is involved: in our models, both $score_{SYN}$ and $score_R$ will contribute to this score. In addition, we add state—specifically, word bigrams at the start and end of constituents—that allows trigram language model scores to be calculated as constituents are combined.

5.2 Approximate Search

There are 2^n possible bit-strings for a sentence of length n, hence the search space is of exponential size; approximate algorithms are therefore required in search for the highest scoring derivation. Figure 6 shows a beam search algorithm which makes use of the inference rules described in the

[9] Note, however, that the derivation step show in figure 4 will be considered in the search, because if *discrimination* takes additional modifiers, and thereby forms a π-constituent that dominates a contiguous sub-string in the German, then the resulting derivation will be valid.

previous section. The algorithm stores sets $\mathcal{Q}_i$ for $i = 1 \ldots n$, where n is the source-language sentence length; each set $\mathcal{Q}_i$ stores hypotheses of length i (i.e., hypotheses with an associated bit-string with i ones). These sets are initialized with basic entries derived from s-phrases.

The function BEAM$(\mathcal{Q}_i)$ returns all items within $\mathcal{Q}_i$ that have a high enough score to fall within a beam (more details for BEAM are given below). At each iteration (step 4), each item in turn is taken from BEAM$(\mathcal{Q}_i)$ and added to a chart; the inference rules described in the previous section are used to derive new items which are added to the appropriate set $\mathcal{Q}_j$, where $j > i$.

We have found the definition of BEAM$(\mathcal{Q}_i)$ to be critical to the success of the method. As a first step, each item in $\mathcal{Q}_i$ receives a score that is a sum of an inside score (the cost of all derivation steps used to create the item) and a future score (an estimate of the cost to complete the translation). The future score is based on the source-language words that are still to be translated—this can be directly inferred from the item's bit-string—this is similar to the use of future scores in Pharoah (Koehn et al., 2003), and in fact we use Pharoah's future scores in our model. We then give the following definition, where N is a parameter (the beam size):

Definition 4 *(BEAM) Given $\mathcal{Q}_i$, define $\mathcal{Q}_{i,j}$ for $j = 1 \ldots n$ to be the subset of items in $\mathcal{Q}_i$ which have their j'th bit equal to one (i.e., have the j'th source language word translated). Define $\mathcal{Q}'_{i,j}$ to be the N highest scoring elements in $\mathcal{Q}_{i,j}$. Then BEAM$(\mathcal{Q}_i) = \cup_{j=1}^{n} \mathcal{Q}'_{i,j}$.*

To motivate this definition, note that a naive method would simply define BEAM$(\mathcal{Q}_i)$ to be the N highest scoring elements of $\mathcal{Q}_i$. This definition, however, assumes that constituents which form translations of different parts of a sentence have scores that can be compared—an assumption that would be true if the future scores were highly accurate, but which quickly breaks down when future scores are inaccurate. In contrast, the definition above ensures that the top N analyses for each of the n source language words are stored at each stage, and hence that all parts of the source sentence are well represented. In experiments, the naive approach was essentially a failure, with parsing of some sentences either failing or being hopelessly inefficient, depending on the choice of N. In contrast, definition 4 gives good results.

System	BLEU score
Syntax-based	25.2
Syntax (no $Score_R$)	23.7 (-1.5)
Syntax (no π-c constraint)	24.4 (-0.8)

Table 1: Development set results showing the effect of removing $Score_R$ or the π-constituent constraint.

5.3 Allowing Multiple Treelets per s-Phrase

The decoding algorithms that we have described apply in the case where each s-phrase has a single treelet. The extension of these algorithms to the case where a phrase may have multiple treelets (e.g., see figure 2) is straightforward, but for brevity the details are omitted. The basic idea is to extend bit-string representations with a record of "pending" treelets which have not yet been included in a derivation. It is also possible to enforce the π-constituent constraint during decoding, as well as a constraint that ensures that reordering operations do not "break apart" English sub-strings within s-phrases that have multiple treelets (for example, for the s-phrase in figure 2, we ensure that *there is no* remains as a contiguous sequence of words in any translation using this s-phrase).

6 Experiments

We trained the syntax-based system on 751,088 German-English translations from the Europarl corpus (Koehn, 2005). A syntactic language model was also trained on the English sentences in the training data. We used Pharoah (Koehn et al., 2003) as a baseline system for comparison; the s-phrases used in our system include all phrases, with the same scores, as those used by Pharoah, allowing a direct comparison. For efficiency reasons we report results on sentences of length 30 words or less.[10] The syntax-based method gives a BLEU (Papineni et al., 2002) score of 25.04, a 0.46 BLEU point gain over Pharoah. This result was found to be significant ($p = 0.021$) under the paired bootstrap resampling method of Koehn (2004), and is close to significant ($p = 0.058$) under the sign test of Collins et al. (2005).

Table 1 shows results for the full syntax-based system, and also results for the system with the discriminative dependency scores (see section 4.1) and the π-contituent constraint removed from the system. In both cases we see a clear impact of these components of the model, with 1.5 and 0.8 BLEU point decrements respectively.

[10] Both Pharoah and our system have weights trained using MERT (Och, 2003) on sentences of length 30 words or less, to ensure that training and test conditions are matched.

R: in our eyes , the opportunity created by this directive of introducing longer buses on international routes is efficient .
S: the opportunity now presented by this directive is effective in our opinion , to use long buses on international routes .
P: the need for this directive now possibility of longer buses on international routes to is in our opinion , efficiently .
R: europe and asia must work together to intensify the battle against drug trafficking , money laundering , international crime , terrorism and the sexual exploitation of minors .
S: europe and asia must work together in order to strengthen the fight against drug trafficking , money laundering , against international crime , terrorism and the sexual exploitation of minors .
P: europe and asia must cooperate in the fight against drug trafficking , money laundering , against international crime , terrorism and the sexual exploitation of minors strengthened .
R: equally important for the future of europe - at biarritz and later at nice - will be the debate on the charter of fundamental rights .
S: it is equally important for the future of europe to speak on the charter of fundamental rights in biarritz , and then in nice .
P: just as important for the future of europe , it will be in biarritz and then in nice on the charter of fundamental rights to speak .
R: the convention was thus a muddled system , generating irresponsibility , and not particularly favourable to well-ordered democracy .
S: therefore , the convention has led to a system of a promoter of irresponsibility of the lack of clarity and hardly coincided with the rules of a proper democracy .
P: the convention therefore led to a system of full of lack of clarity and hardly a promoter of the irresponsibility of the rules of orderly was a democracy .

Figure 7: Examples where both annotators judged the syntactic system to give an improved translation when compared to the baseline system. 51 out of 200 translations fall into this category. These examples were chosen at random from these 51 examples. **R** is the human (reference) translation; **S** is the translation from the syntax-based system; **P** is the output from the baseline (phrase-based) system.

	Syntax	PB	=	Total
Syntax	51	3	7	61
PB	1	25	11	37
=	21	14	67	102
Total	73	42	85	200

Table 2: Human annotator judgements. Rows show results for annotator 1, and columns for annotator 2. *Syntax* and *PB* show the number of cases where an annotator respectively preferred/dispreferred the syntax-based system. $=$ gives counts of translations judged to be equal in quality.

In addition, we obtained human evaluations on 200 sentences chosen at random from the test data, using two annotators. For each example, the reference translation was presented to the annotator, followed by translations from the syntax-based and phrase-based systems (in a random order). For each example, each annotator could either decide that the two translations were of equal quality, or that one translation was better than the other. Table 2 shows results of this evaluation. Both annotators show a clear preference for the syntax-based system: for annotator 1, 73 translations are judged to be better for the syntax-based system, with 42 translations being worse; for annotator 2, 61 translations are improved with 37 being worse; both annotators' results are statistically significant with $p < 0.05$ under the sign test. Figure 7 shows some translation examples where the syntax-based system was judged to give an improvement.

7 Conclusions and Future Work

We have described a translation model that makes use of flexible parsing operations, critical ideas being the definition of s-phrases, Γ-dependencies, the π-constituent constraint, and an approximate search algorithm. A key area for future work will be further development of the discriminative dependency model (section 4.1). The model of $score_r(\gamma, \mathbf{f})$ that we have described in this paper is relatively simple; in general, however, there is the potential for $score_r$ to link target language dependencies to arbitrary properties of the source language string $\mathbf{f}$ (recall that γ contains a head and modifier spine in the target language, along with positions in the source-language string to which these spines are aligned). For example, we might introduce features that: a) condition dependencies created in the target language on dependency relations between their aligned words in the source language; b) condition target-language dependencies on whether they are aligned to words that are in the same clause or segment in the source language string; or, c) condition the grammatical roles of nouns in the target language on grammatical roles of aligned words in the source language. These features should improve translation quality by giving a tighter link between syntax in the source and target languages, and would be easily incorporated in the approach we have described.

Acknowledgments We would like to thank Ryan McDonald for conversations that were influential in this work, and Meg Aycinena Lippow and Ben Snyder for translation judgments. This work was supported under the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-C-0022.

References

H. Alshawi. 1996. Head automata and bilingual tiling: Translation with minimal representations. In *Proceedings of ACL*, pages 167–176.

X. Carreras, M. Collins, and T. Koo. 2008. TAG, dynamic programming and the perceptron for efficient, feature-rich parsing. In *Proc. of CoNLL*.

E. Charniak, K. Knight, and K. Yamada. 2003. Syntax-based language models for machine translation. In *Proceedings of MT Summit IX*.

E. Charniak. 2001. Immediate-head parsing for language models. In *Proceedings of ACL 2001*.

C. Cherry. 2008. Cohesive phrase-based decoding for statistical machine translation. In *Proceedings of ACL-08: HLT*, pages 72–80, Columbus, Ohio, June. Association for Computational Linguistics.

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

M. Collins, P. Koehn, and I. Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proceedings of ACL*.

M. Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 16–23, Madrid, Spain, July. Association for Computational Linguistics.

J. Eisner. 2000. Bilexical grammars and their cubic-time parsing algorithms. In H. C. Bunt and A. Nijholt, editors, *New Developments in Natural Language Parsing*, pages 29–62. Kluwer Academic Publishers.

J. Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. In *Proceedings of ACL*.

A.K. Joshi and Y. Schabes. 1997. Tree-adjoining grammars. In G. Rozenberg and K. Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 169–124. Springer.

P. Koehn, F.J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT/NAACL*.

P. Koehn. 2004. Statistical significance tests for machine translation evaluation. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July. Association for Computational Linguistics.

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

D. Marcu, W. Wang, A. Echihabi, and K. Knight. 2006. Spmt: Statistical machine translation with syntactified target language phrases. In *Proceedings of EMNLP*.

R. McDonald, K. Crammer, and F. Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of ACL*.

D. Melamed. 2004. Statistical machine translation by parsing. In *Proceedings of ACL*.

H. Mi, L. Huang, and Q. Liu. 2008. Forest-based translation. In *Proceedings of ACL-08: HLT*, pages 192–199. Association for Computational Linguistics.

R. Nesson, S.M. Shieber, and A. Rush. 2006. Induction of probabilistic synchronous tree-insertion grammars for machine translation. In *Proceedings of the 7th AMTA*.

F.J. Och. 2003. Minimum error rate training for statistical machine translation. In *Proceedings of ACL*.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318. Association for Computational Linguistics.

C. Quirk, A. Menezes, and Colin Cherry. 2005. Dependency tree translation: Syntactically informed phrasal smt. In *Proceedings of ACL*.

O. Rambow, K. Vijay-Shanker, and D. Weir. 1995. D-tree grammars. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 151–158, Cambridge, Massachusetts, USA, June. Association for Computational Linguistics.

L. Shen, J. Xu, and R. Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of ACL*.

D. Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–404.

K. Yamada and K. Knight. 2001. A syntax-based statistical translation model. In *Proceedings of ACL*.

H. Zhang and D. Gildea. 2005. Stochastic lexicalized inversion transduction grammar for alignment. In *Proceedings of ACL*, pages 473–482.

A. Zollmann and A. Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings of NAACL 2006 Workshop on Statistical Machine Translation*.