What's in a translation rule?

Michel Galley
Dept. of Computer Science
Columbia University
New York, NY 10027
galley@cs.columbia.edu

Mark Hopkins
Dept. of Computer Science
University of California
Los Angeles, CA 90024
mhopkins@cs.ucla.edu

Kevin Knight and **Daniel Marcu**
Information Sciences Institute
University of Southern California
Marina Del Rey, CA 90292
{knight,marcu}@isi.edu

Abstract

We propose a theory that gives formal semantics to word-level alignments defined over parallel corpora. We use our theory to introduce a linear algorithm that can be used to derive from word-aligned, parallel corpora the minimal set of syntactically motivated transformation rules that explain human translation data.

1 Introduction

In a very interesting study of syntax in statistical machine translation, Fox (2002) looks at how well proposed translation models fit actual translation data. One such model embodies a restricted, linguistically-motivated notion of word re-ordering. Given an English parse tree, children at any node may be reordered prior to translation. Nodes are processed independently. Previous to Fox (2002), it had been observed that this model would prohibit certain re-orderings in certain language pairs (such as subject-VP(verb-object) into verb-subject-object), but Fox carried out the first careful empirical study, showing that many other common translation patterns fall outside the scope of the child-reordering model. This is true even for languages as similar as English and French. For example, English adverbs tend to move outside the local parent/children in environment. The English word "not" translates to the discontiguous pair "ne ... pas." English parsing errors also cause trouble, as a normally well-behaved re-ordering environment can be disrupted by wrong phrase attachment. For other language pairs, the divergence is expected to be greater.

In the face of these problems, we may choose among several alternatives. The first is to abandon syntax in statistical machine translation, on the grounds that syntactic models are a poor fit for the data. On this view, adding syntax yields no improvement over robust phrase-substitution models, and the only question is how much does syntax hurt performance. Along this line, (Koehn et al., 2003) present convincing evidence that restricting phrasal translation to syntactic constituents yields poor translation performance – the ability to translate non-constituent phrases (such as "there are", "note that", and "according to") turns out to be critical and pervasive.

Another direction is to abandon conventional English syntax and move to more robust grammars that adapt to the parallel training corpus. One approach here is that of Wu (1997), in which word-movement is modeled by rotations at unlabeled, binary-branching nodes. At each sentence pair, the parse adapts to explain the translation pattern. If the same unambiguous English sentence were to appear twice in the corpus, with different Chinese translations, then it could have different learned parses.

A third direction is to maintain English syntax and investigate alternate transformation models. After all, many conventional translation systems are indeed based on syntactic transformations far more expressive than what has been proposed in syntax-based statistical MT. We take this approach in our paper. Of course, the broad statistical MT program is aimed at a wider goal than the conventional rule-based program – it seeks to understand and explain human translation data, and automatically learn from it. For this reason, we think it is important to learn from the model/data explainability studies of Fox (2002) and to extend her results. In addition to being motivated by rule-based systems, we also see advantages to English syntax within the statistical framework, such as marrying syntax-based translation models with syntax-based language models (Charniak et al., 2003) and other potential benefits described by Eisner (2003).

Our basic idea is to create transformation rules that condition on larger fragments of tree structure. It is certainly possible to build such rules by hand, and we have done this to formally explain a number of human-translation examples. But our main interest is in collecting a large set of such rules automatically through corpus

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2004		**00-00-2004 to 00-00-2004**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
What's in a translation rule?	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Computer Science Department,Columbia University,New York City,NY,10027**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**8**	

analysis. The search for these rules is driven exactly by the problems raised by Fox (2002) – cases of crossing and divergence motivate the algorithms to come up with better explanations of the data and better rules. Section 2 of this paper describes algorithms for the acquisition of complex rules for a transformation model. Section 3 gives empirical results on the explanatory power of the acquired rules versus previous models. Section 4 presents examples of learned rules and shows the various types of transformations (lexical and nonlexical, contiguous and noncontiguous, simple and complex) that the algorithms are forced (by the data) to invent. Section 5 concludes. Due to space constraints, all proofs are omitted.

2 Rule Acquisition

Suppose that we have a French sentence, its translation into English, and a parse tree over the English translation, as shown in Figure 1. Generally one defines an alignment as a relation between the words in the French sentence and the words in the English sentence. Given such an alignment however, what kinds of rules are we entitled to learn from this instance? How do we know when it is valid to extract a particular rule, especially in the presence of numerous crossings in the alignment? In this section, we give principled answers to these questions, by constructing a theory that gives formal semantics to word alignments.

2.1 A Theory of Word Alignments

We are going to define a generative process through which a string from a source alphabet is mapped to a rooted tree whose nodes are labeled from a target alphabet. Henceforth we will refer to symbols from our source alphabet as *source symbols* and symbols from our target alphabet as *target symbols*. We define a *symbol tree* over an alphabet Δ as a rooted, directed tree, the nodes of which are each labeled with a symbol of Δ.

We want to capture the process by which a symbol tree over the target language is derived from a string of source symbols. Let us refer to the symbol tree that we want to derive as the *target tree*. Any subtree of this tree will be called a *target subtree*. Furthermore, we define a *derivation string* as an ordered sequence of elements, each of which is either a source symbol or a target subtree.

Now we are ready to define the derivation process. Given a derivation string S, a *derivation step* replaces a substring S' of S with a target subtree T that has the following properties:

1. Any target subtree in S' is a subtree of T.

2. Any target subtree in S but not in S' does not share nodes with T.

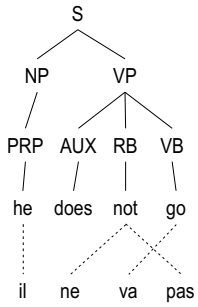

Figure 1: A French sentence aligned with an English parse tree.

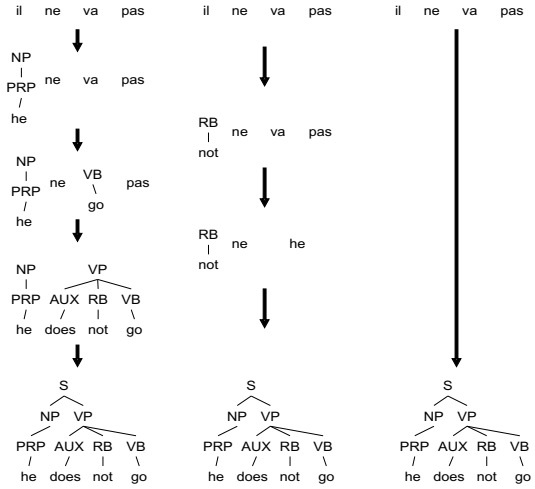

Figure 2: Three alternative derivations from a source sentence to a target tree.

Moreover, a *derivation* from a string S of source symbols to the target tree T is a sequence of derivation steps that produces T from S.

Moving away from the abstract for a moment, let us revisit the example from Figure 1. Figure 2 shows three derivations of the target tree from the source string "il ne va pas", which are all consistent with our definitions. However, it is apparent that one of these derivations seems much more "wrong" than the other. Specifically, in the second derivation, "pas" is replaced by the English word "he," which makes no sense. Given the vast space of possible derivations (according to the definition above), how do we distinguish between good ones and bad ones? Here is where the notion of an alignment becomes useful.

Let S be a string of source symbols and let T be a target tree. First observe the following facts about derivations from S to T (these follow directly from the definitions):

1. Each element of S is replaced at exactly one step of the derivation.

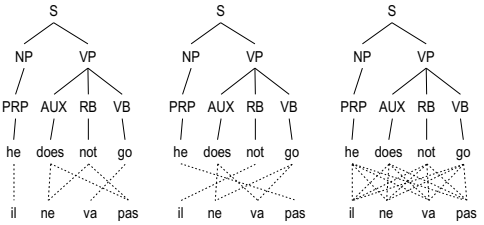

Figure 3: The alignments induced by the derivations in Figure 2

2. Each node of T is created at exactly one step of the derivation.

Thus for each element s of S, we can define $replaced(s, D)$ to be the step of the derivation D during which s is replaced. For instance, in the leftmost derivation of Figure 2, "va" is replaced by the second step of the derivation, thus $replaced(va, D) = 2$. Similarly, for each node t of T, we can define $created(t, D)$ to be the step of derivation D during which t is created. For instance, in the same derivation, the nodes labeled by "AUX" and "VP" are created during the third step of the derivation, thus $created(AUX, D) = 3$ and $created(VP, D) = 3$.

Given a string S of source symbols and a target tree T, an *alignment* A with respect to S and T is a relation between the leaves of T and the elements of S. Choose some derivation D from S to T. The *alignment A induced by D* is created as follows: an element s of S is aligned with a leaf node t of T iff $replaced(s, D) = created(t, D)$. In other words, a source word is aligned with a target word if the target word is created during the same step in which the source word is replaced. Figure 3 shows the alignments induced by the derivations of Figure 2.

Now, say that we have a source string, a target tree, and an alignment A. A key observation is that the set of "good" derivations according to A is precisely the set of derivations that induce alignments A' such that A is a subalignment of A'. By subalignment, we mean that $A \subseteq A'$ (recall that alignments are simple mathematical relations). In other words, A is a subalignment of A' if A aligns two elements only if A' also aligns them.

We can see this intuitively by examining Figures 2 and 3. Notice that the two derivations that seem "right" (the first and the third) are superalignments of the alignment given in Figure 1, while the derivation that is clearly wrong is not. Hence we now have a formal definition of the derivations that we are interested in. We say that a derivation is *admitted* by an alignment A if it induces a superalignment of A. The set of derivations from source string S to target tree T that are admitted by alignment A can be denoted $\delta_A(S, T)$. Given this, we are ready to obtain a formal characterization of the set of rules that can be inferred from the source string, target tree, and alignment.

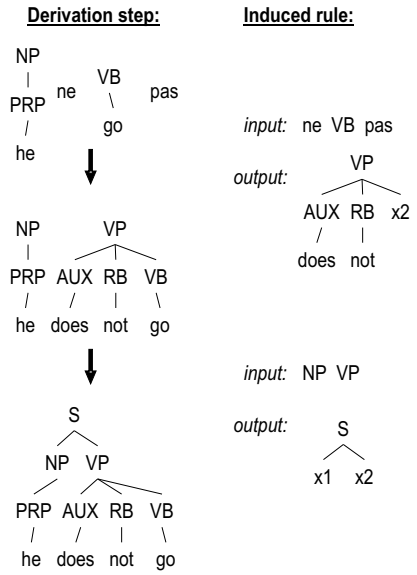

Figure 4: Two derivation steps and the rules that are induced from them.

2.2 From Derivations to Rules

In essence, a derivation step can be viewed as the application of a rule. Thus, compiling the set of derivation steps used in any derivation of $\delta_A(S, T)$ gives us, in a meaningful sense, all relevant rules that can be extracted from the triple (S, T, A). In this section, we show in concrete terms how to convert a derivation step into a usable rule.

Consider the second-last derivation step of the first derivation in Figure 2. In it, we begin with a source symbol "ne", followed by a target subtree rooted at VB, followed by another source symbol "pas." These three elements of the derivation string are replaced with a target subtree rooted at VP that discards the source symbols and contains the target subtree rooted at VB. In general, this replacement process can be captured by the rule depicted in Figure 4. The input to the rule are the roots of the elements of the derivation string that are replaced (where we define the root of a symbol to be simply the symbol itself), whereas the output of the rule is a symbol tree, except that some of the leaves are labeled with variables instead of symbols from the target alphabet. These variables correspond to elements of the input to the rule. For instance, the leaf labeled $x2$ means that when this rule is applied, $x2$ is replaced by the target subtree rooted at VB (since VB is the second element of the input). Observe that the second rule induced in Figure 4 is simply a CFG rule expressed in the opposite direction, thus this rule format can (and should) be viewed as a strict generalization of CFG rules.

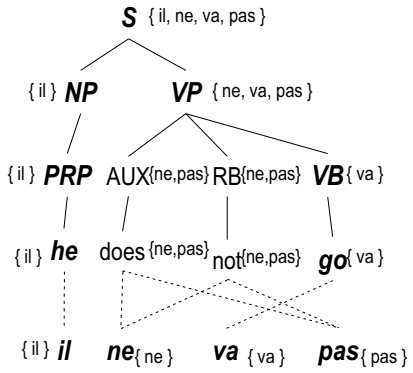

Figure 5: An alignment graph. The nodes are annotated with their spans. Nodes in the frontier set are boldfaced and italicized.

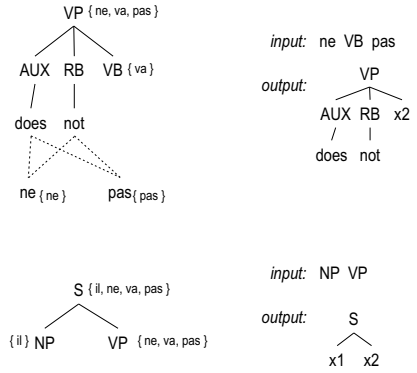

Figure 6: Two frontier graph fragments and the rules induced from them. Observe that the spans of the sink nodes form a partition of the span of the root.

Every derivation step can be mapped to a rule in this way. Hence given a source string S, a target tree T, and an alignment A, we can define the set $\rho_A(S,T)$ as the set of rules in any derivation $D \in \delta_A(S,T)$. We can regard this as the set of rules that we are entitled to infer from the triple (S,T,A).

2.3 Inferring Complex Rules

Now we have a precise problem statement: learn the set $\rho_A(S,T)$. It is not immediately clear how such a set can be learned from the triple (S,T,A). Fortunately, we can infer these rules directly from a structure called an *alignment graph*. In fact, we have already seen numerous examples of alignment graphs. Graphically, we have been depicting the triple (S,T,A) as a rooted, directed, acyclic graph (where direction is top-down in the diagrams). We refer to such a graph as an *alignment graph*. Formally, the alignment graph corresponding to S, T, and A is just T, augmented with a node for each element of S, and edges from leaf node $t \in T$ to element $s \in S$ iff A aligns s with t. Although there is a difference between a node of the alignment graph and its label, we will not make a distinction, to ease the notational burden.

To make the presentation easier to follow, we assume throughout this section that the alignment graph is connected, i.e. there are no unaligned elements. All of the results that follow have generalizations to deal with unaligned elements, but unaligned elements incur certain procedural complications that would cloud the exposition.

It turns out that it is possible to systematically convert certain fragments of the alignment graph into rules of $\rho_A(S,T)$. We define a *fragment* of a directed, acyclic graph G to be a nontrivial (i.e. not just a single node) subgraph G' of G such that if a node n is in G' then either n is a sink node of G' (i.e. it has no children) or all of its children are in G' (and it is connected to all of them). In Figure 6, we show two examples of graph fragments of the alignment graph of Figure 5.

The *span* of a node n of the alignment graph is the subset of nodes from S that are reachable from n. Note that this definition is similar to, but not quite the same as, the definition of a span given by Fox (2002). We say that a span is *contiguous* if it contains all elements of a contiguous substring of S. The *closure* of $span(n)$ is the shortest contiguous span which is a superset of $span(n)$. For instance, the closure of $\{s_2, s_3, s_5, s_7\}$ would be $\{s_2, s_3, s_4, s_5, s_6, s_7\}$ The alignment graph in Figure 5 is annotated with the span of each node.

Take a look at the graph fragments in Figure 6. These fragments are special: they are examples of *frontier graph fragments*. We first define the *frontier set* of an alignment graph to be the set of nodes n that satisfy the following property: for every node n' of the alignment graph that is connected to n but is neither an ancestor nor a descendant of n, $span(n') \cap closure(span(n)) = \emptyset$.

We then define a *frontier graph fragment* of an alignment graph to be a graph fragment such that the root and all sinks are in the frontier set. Frontier graph fragments have the property that the spans of the sinks of the fragment are each contiguous and form a partition of the span of the root, which is also contiguous. This allows the following transformation process:

1. Place the sinks in the order defined by the partition (i.e. the sink whose span is the first part of the span of the root goes first, the sink whose span is the second part of the span of the root goes second, etc.). This forms the input of the rule.

2. Replace sink nodes of the fragment with a variable corresponding to their position in the input, then take the tree part of the fragment (i.e. project the fragment on T). This forms the output of the rule.

Figure 6 shows the rules derived from the given graph fragments. We have the following result.

Theorem 1 *Rules constructed according to the above procedure are in* $\rho_A(S,T)$.

Rule extraction: Algorithm 1. Thus we now have a simple method for extracting rules of $\rho_A(S,T)$ from the alignment graph: search the space of graph fragments for frontier graph fragments.

Unfortunately, the search space of all fragments of a graph is exponential in the size of the graph, thus this procedure can also take a long time to execute. To arrive at a much faster procedure, we take advantage of the following provable facts:

1. The frontier set of an alignment graph can be identified in time linear in the size of the graph.
2. For each node n of the frontier set, there is a unique minimal frontier graph fragment rooted at n (observe that for any node n' not in the frontier set, there is no frontier graph fragment rooted at n', by definition).

By minimal, we mean that the frontier graph fragment is a subgraph of every other frontier graph fragment with the same root. Clearly, for an alignment graph with k nodes, there are at most k minimal frontier graph fragments. In Figure 7, we show the seven minimal frontier graph fragments of the alignment graph of Figure 5. Furthermore, all other frontier graph fragments can be created by composing 2 or more minimal graph fragments, as shown in Figure 8. Thus, the entire set of frontier graph fragments (and all rules derivable from these fragments) can be computed systematically as follows: compute the set of minimal frontier graph fragments, compute the set of graph fragments resulting from composing 2 minimal frontier graph fragments, compute the set of graph fragments resulting from composing 3 minimal graph fragments, etc. In this way, the rules derived from the minimal frontier graph fragments can be regarded as a basis for all other rules derivable from frontier graph fragments. Furthermore, we conjecture that the set of rules derivable from frontier graph fragments is in fact equivalent to $\rho_A(S,T)$.

Thus we have boiled down the problem of extracting complex rules to the following simple problem: find the set of minimal frontier graph fragments of a given alignment graph.

The algorithm is a two-step process, as shown below.

Rule extraction: Algorithm 2

1. Compute the frontier set of the alignment graph.
2. For each node of the frontier set, compute the minimal frontier graph fragment rooted at that node.

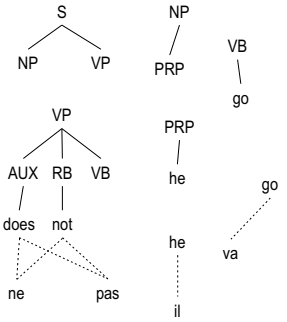

Figure 7: The seven minimal frontier graph fragments of the alignment graph in Figure 5

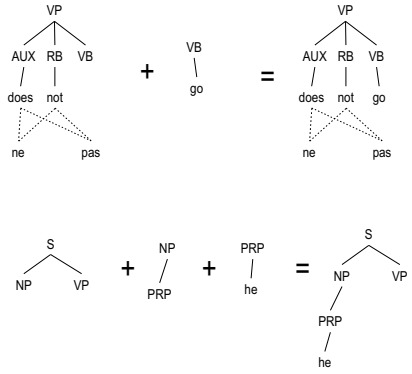

Figure 8: Example compositions of minimal frontier graph fragments into larger frontier graph fragments.

Step 1 can be computed in a single traversal of the alignment graph. This traversal annotates each node with its span and its *complement span*. The *complement span* is computed as the union of the complement span of its parent and the span of all its siblings (siblings are nodes that share the same parent). A node n is in the frontier set iff $complement\_span(n) \cap closure(span(n)) = \emptyset$. Notice that the complement span merely summarizes the spans of all nodes that are neither ancestors nor descendents of n. Since this step requires only a single graph traversal, it runs in linear time.

Step 2 can also be computed straightforwardly. For each node n of the frontier set, do the following: expand n, then as long as there is some sink node n' of the resulting graph fragment that is not in the frontier set, expand n'. Note that after computing the minimal graph fragment rooted at each node of the frontier set, every node of the alignment graph has been expanded at most once. Thus this step also runs in linear time.

For clarity of exposition and lack of space, a couple of issues have been glossed over. Briefly:

- As previously stated, we have ignored here the issue of unaligned elements, but the procedures can be easily generalized to accommodate these. The

results of the next two sections are all based on implementations that handle unaligned elements.

- This theory can be generalized quite cleanly to include derivations for which substrings are replaced by sets of trees, rather than one single tree. This corresponds to allowing rules that do not require the output to be a single, rooted tree. Such a generalization gives some nice power to effectively explain certain linguistic phenomena. For instance, it allows us to immediately translate "va" as "does go" instead of delaying the creation of the auxiliary word "does" until later in the derivation.

3 Experiments

3.1 Language Choice

We evaluated the coverage of our model of transformation rules with two language pairs: English-French and English-Chinese. These two pairs clearly contrast by the underlying difficulty to understand and model syntactic transformations among pairs: while there is arguably a fair level of cohesion between English and French, English and Chinese are syntactically more distant languages. We also chose French to compare our study with that of Fox (2002). The additional language pair provides a good means of evaluating how our transformation rule extraction method scales to more problematic language pairs for which child-reordering models are shown not to explain the data well.

3.2 Data

We performed experiments with two corpora, the FBIS English-Chinese Parallel Text and the Hansard French-English corpus.We parsed the English sentences with a state-of-the-art statistical parser (Collins, 1999). For the FBIS corpus (representing eight million English words), we automatically generated word-alignments using GIZA++ (Och and Ney, 2003), which we trained on a much larger data set (150 million words). Cases other than one-to-one sentence mappings were eliminated. For the Hansard corpus, we took the human annotation of word alignment described in (Och and Ney, 2000). The corpus contains two kinds of alignments: S (sure) for unambiguous cases and P (possible) for unclear cases, e.g. idiomatic expressions and missing function words ($S \subseteq P$). In order to be able to make legitimate comparisons between the two language pairs, we also used GIZA++ to obtain machine-generated word alignments for Hansard: we trained it with the 500 sentences and additional data representing 13.7 million English words (taken from the Hansard and European parliament corpora).

3.3 Results

From a theoretical point of view, we have shown that our model can fully explain the transformation of any parse tree of the source language into a string of the target language. The purpose of this section is twofold: to provide quantitative results confirming the full coverage of our model and to analyze some properties of the transformation rules that support these derivations (linguistic analyses of these rules are presented in the next section).

Figure 9 summarizes the coverage of our model with respect to the Hansard and FBIS corpora. For the former, we present results for the three alignments: S alignments, P alignments, and the alignments computed by GIZA++. Each plotted value represents a percentage of parse trees in a corpus that can be transformed into a target sentence using transformation rules. The x-axis represents different restrictions on the size of these rules: if we use a model that restrict rules to a single expansion of a non-terminal into a sequence of symbols, we are in the scope of the child-reordering model of (Yamada and Knight, 2001; Fox, 2002). We see that its explanatory power is quite poor, with only 19.4%, 14.3%, 16.5%, and 12.1% (for the respective corpora). Allowing more expansions logically expands the coverage of the model, until the point where it is total: transformation rules no larger than 17, 18, 23, and 43 (in number of rule expansions) respectively provide enough coverage to explain the data at 100% for each of the four cases.

It appears from the plot that the quality of alignments plays an important role. If we compare the three kinds of alignments available for the Hansard corpus, we see that much more complex transformation rules are extracted from noisy GIZA++ alignments. It also appears that the language difference produces quite contrasting results. Rules acquired for the English-Chinese pair have, on average, many more nodes. Note that the language difference in terms of syntax might be wider than what the plot seems to indicate, since word alignments computed for the Hansard corpus are likely to be more errorful than the ones for FBIS because the training data used to induce the latter is more than ten times larger than for the former.

In Figure 10, we show the explanatory power of our model at the node level. At each node of the frontier set, we determine whether it is possible to extract a rule that doesn't exceed a given limit k on its size. The plotted values represent the percentage of frontier set internal nodes that satisfy this condition. These results appear more promising for the child-reordering model, with coverage ranging from 72.3% to 85.1% of the nodes, but we should keep in mind that many of these nodes are low in the tree (e.g. base NPs); extraction of 1-level transformation rules generally present no difficulties when child nodes are pre-terminals, since any crossings can be resolved by lexicalizing the elements involved in it. How-

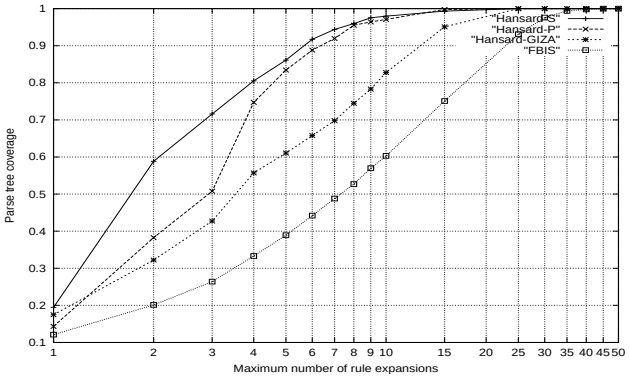

Figure 9: Percentage of parse trees covered by the model given different constraints on the maximum size of the transformation rules.

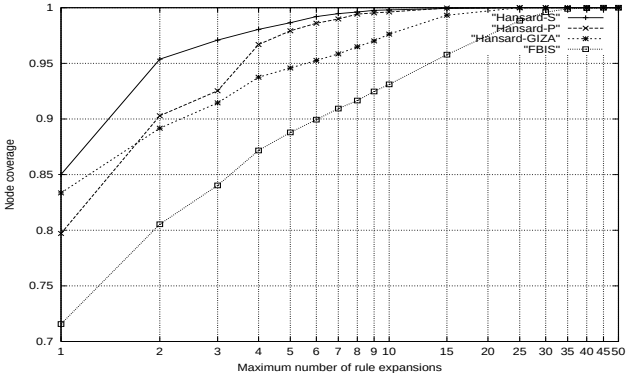

Figure 10: Same as Figure 9, except that here coverage is evaluated at the node level.

ever, higher level syntactic constituents are more problematic for child-reordering models, and the main reasons they fail to provide explanation of the parses at the sentence level.

Table 1 shows that the extraction of rules can be performed quite efficiently. Our first algorithm, which has an exponential running time, cannot scale to process large corpora and extract a sufficient number of rules that a syntax-based statistical MT system would require. The second algorithm, which runs in linear time, is on the other hand barely affected by the size of rules it extracts.

	k=1	3	5	7	10	20	50
I	4.1	10.2	57.9	304.2	-	-	-
II	4.3	5.4	5.9	6.4	7.33	9.6	11.8

Table 1: Running time in seconds of the two algorithms on 1000 sentences. k represent the maximum size of rules to extract.

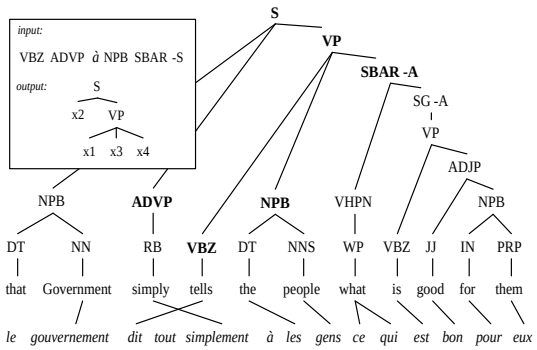

Figure 11: Adverb-verb reordering.

4 Discussions

In this section, we present some syntactic transformation rules that our system learns. Fox (2002) identified three major causes of crossings between English and French: the "ne ... pas" construct, modals and adverbs, which a child-reordering model doesn't account for. In section 2, we have already explained how we learn syntactic rules involving "ne ... pas". Here we describe the other two problematic cases.

Figure 11 presents a frequent cause of crossings between English and French: adverbs in French often appear after the verb, which is less common in English. Parsers generally create nested verb phrases when adverbs are present, thus no child reordering can allow a verb and an adverb to be permuted. Multi-level reodering as the rule in the figure can prevent crossings. Fox's solution to the problem of crossings is to flatten verb phrases. This is a solution for this sentence pair, since this accounts for adverb-verb reorderings, but flattening the tree structure is not a general solution. Indeed, it can only apply to a very limited number of syntactic categories, for which the advantage of having a deep syntactic structure is lost.

Figure 12 (dotted lines are P alignments) shows an interesting example where flattening the tree structure cannot resolve all crossings in node-reordering models. In these models, a crossing remains between MD and AUX no matter how VPs are flattened. Our transformation rule model creates a lexicalized rule as shown in the figure, where the transformation of "will be" into "sera" is the only way to resolve the crossing.

In the Chinese-English domain, the rules extracted by our algorithm often have the attractive quality that they are the kind of common-sense constructions that are used in Chinese language textbooks to teach students. For instance, there are several that illustrate the complex reorderings that occur around the Chinese marker word "de."

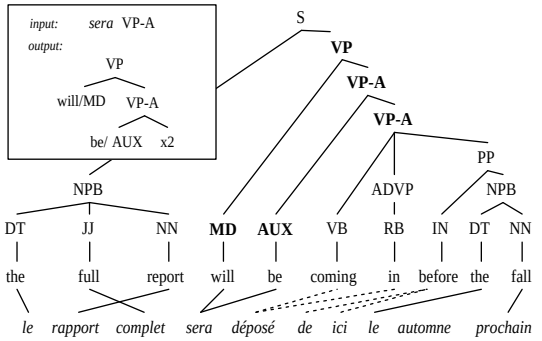

Figure 12: Crossing due to a modal.

5 Conclusion

The fundamental assumption underlying much recent work in statistical machine translation (Yamada and Knight, 2001; Eisner, 2003; Gildea, 2003) is that local transformations (primarily child-node re-orderings) of one-level parent-children substructures are an adequate model for parallel corpora. Our empirical results suggest that this may be too strong of an assumption. To explain the data in two parallel corpora, one English-French, and one English-Chinese, we are often forced to learn rules involving much larger tree fragments. The theory, algorithms, and transformation rules we learn automatically from data have several interesting aspects.

1. Our rules provide a good, realistic indicator of the complexities inherent in translation. We believe that these rules can inspire subsequent developments of generative statistical models that are better at explaining parallel data than current ones.

2. Our rules put at the fingertips of linguists a very rich source of information. They encode translation transformations that are both syntactically and lexically motivated (some of our rules are purely syntactic; others are lexically grounded). A simple sort on the counts of our rules makes explicit the transformations that occur most often. A comparison of the number of rules extracted from parallel corpora specific to multiple language pairs provide a quantitative estimator of the syntactic "closeness" between various language pairs.

3. The theory we proposed in this paper is independent of the method that one uses to compute the word-level alignments in a parallel corpus.

4. The theory and rule-extraction algorithm are also well-suited to deal with the errors introduced by the word-level alignment and parsing programs one uses. Our theory makes no a priori assumptions about the transformations that one is permitted to learn. If a parser, for example, makes a systematic error, we expect to learn a rule that can nevertheless be systematically used to produce correct translations.

In this paper, we focused on providing a well-founded mathematical theory and efficient, linear algorithms for learning syntactically motivated transformation rules from parallel corpora. One can easily imagine a range of techniques for defining probability distributions over the rules that we learn. We suspect that such probabilistic rules could be also used in conjunction with statistical decoders, to increase the accuracy of statistical machine translation systems.

Acknowledgements

This work was supported by DARPA contract N66001-00-1-9814 and MURI grant N00014-00-1-0617.

References

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

M. Collins. 1999. *Head-driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania, Philadelphia.

J. Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. In *Proc. of the 41st Meeting of the Association for Computational Linguistics*.

H. Fox. 2002. Phrasal cohesion and statistical machine translation. In *Proc. of Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

D. Gildea. 2003. Loosely tree-based alignment for machine translation. In *Proc. of the 41th Annual Conference of the Association for Computational Linguistics*.

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

F. Och and H. Ney. 2000. Improved statistical alignment models. *Proc. of the 38th Annual Meeting of the Association for Computational Linguistics*.

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

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 *ACL*, pages 523–530.