Extraction of Syntactic Translation Models from Parallel Data using Syntax from Source and Target Languages

Vamshi Ambati, Alon Lavie, Jaime Carbonell
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{vamshi,alavie,jgc}@cs.cmu.edu

Abstract

We propose a generic rule induction framework that is informed by syntax from both sides of a parsed parallel corpus, as sets of structural, boundary and labeling related constraints. Factoring syntax in this manner empowers our framework to work with independent annotations coming from multiple resources and not necessarily a single syntactic structure. We then explore the issue of lexical coverage of translation models learned in different scenarios using syntax from one side vs. both sides. We specifically look at how the non-isomorphic nature of parse trees for the two languages affects coverage. We propose a novel technique for restructuring target-side parse trees, that generates alternate isomorphic target trees that preserve the syntactic boundaries of constituents that were aligned in the original parse trees. We also show that combining rules extracted by restructuring syntactic trees on both sides produces significantly better translation models. The improved precision and coverage of our syntax tables particularly fill in for the lack of lexical coverage in Syntax based Machine Translation approaches.

1 Introduction

Recent approaches to Syntax based Machine Translation (MT) incorporate linguistic syntax for one side of the language pair, and obtain phrase tables and hierarchical translation rules. While this has indeed proven successful (Yamada and Knight, 2001) (Marcu et al., 2006), it has been shown that the word alignments, which are usually extracted using syntactically uninformed generative models, are not optimal for the syntactic phrase extraction problem. Other approaches (Tinsley et al., 2007),(Lavie et al., 2008) have been proposed for using syntactic parse trees for both the languages, employing node alignment techniques to align them and extract hierarchical translation models for syntactic machine translation systems. Using trees for both sides suffers from severe coverage problems, primarily due to the highly restrictive space of constituent segmentations that the trees on two sides introduce.

Phrase based statistical MT (PB-SMT) techniques for extracting phrases although not syntactically motivated, enjoy a very high coverage. In order to bridge the gap some successful approaches to syntax in MT resort to binarization of trees(Wang et al., 2007) that systematically alter the structure of the source side parse tree to increase the space of segmentation allowed. This improves the recall of the syntactic translation models in particular the flat rules corresponding to syntactic phrasal entries. Another promising approach for bridging the coverage gap is combination of non-syntactic phrases with syntactic phrases (Hanneman and Lavie, 2009). Such techniques have shown that starting with large syntactic phrase tables and preferring syntactic phrases when overlapping with non-syntactic ones is beneficial for a syntactic MT system. They show improvements in decoding speeds and also improvement in translation quality that results from the precision of these syntax motivated phrases. The syntactic tables we produce in our work are precise and much high in coverage and can directly sup-

port these approaches. (Hanneman and Lavie, 2009) also show that a small set of manual synchronous grammar rules already benefit from clean syntactic tables to better explain the word movement in MT. The goal of our work is not to compete with non-syntactic systems, but to create better syntactic phrase tables and grammars which can help Syntax based MT approaches.

In this paper, we first formalize the task of rule induction for syntactic translation models as the task of extracting rules, both lexical and structural, respecting constraints imposed by the underlying word alignment and syntax. Syntactic constraints coming from a parse tree can be put into three categories - *structural constraints* related to the hierarchical nature of the tree, *constituency constraints* related to the spans and boundaries of what is a constituent, and *labeling constraints* related to the assignment of labels to the constituents. Decoupling syntax into these components enables working with a variety of syntax models ranging from phrase structure trees to simple dependency structures. We then show how most, if not all existing rule induction approaches can be explained as instances within this framework and observe that different combinations of constraints lead to diverse translation models with varying qualities of translation.

A primary observation is that phrase tables obtained when restricting phrase spans to abide by constituency constraints from both sides are precise since they can, where possible, overcome recall in word-alignment. However they are much smaller in size when compared to non-syntactic phrase tables or those obtained using tree on one side only. The coverage sparsity from using syntax on both sides can be attributed to the non-isomorphic nature of the parse trees for the language pair that come from two completely independent parsers and parsing models. The parser design is a monolingual activity targeted for a specific task of focus and not necessarily well suited for MT. We propose a novel technique for modifying the non-isomorphic parse tree structures for the target language, by introducing an isomorphic backbone parse structure into the target tree and merging the nodes to retain the tree structure. Finally we propose 'symmetric' rule induction technique, which extracts grammar by restructuring trees on both sides of the parallel corpus. Extraction of syntactic translation models using our methods produces superior quality syntactic phrase tables with improved lexical coverage.

The rest of the paper is organized as follows. In Section 2 we discuss the framework for rule induction using syntax on both sides. In Section 3 we discuss our approach to modifying the non-isomorphic parse trees which can then be used in translation model extraction. Section 4 discusses symmetric rule induction technique for improved translation model extraction. We conclude by discussing the results in Section 5.

2 Rule Induction Framework

One of the requirements for a generic rule induction framework from parallel data is to be able to incorporate any kind of syntactic information that may come from either sides of the language pair. Our rule learning framework is general and works with syntax on both sides or any other annotation that is available, such as dependency structures, part-of-speech tags etc. We achieve this by factorizing syntax into three essential components - structure, constituency and labeling.

2.1 Problem and Inputs

A parallel corpus D is defined as a set of sentence pairs (F_I^1, E_J^1, A), where $A \subseteq \{(i,j) : i = 1..I', j = 1..J'\}$ is the word alignment relation over each pair. We subscribe to the theory of alignments discussed in (Galley et al., 2004), that explains the concept of 'consistency of word-alignment' for minimal rule extraction from a tree. Consistent alignment requires all the words in a particular segment of the source side to align with a particular contiguous segment of the target sentence, as decided by the word-level alignment. Formally, a span (i_s, j_s) projects to a target span(i_t, j_t) if and only if $\forall k_s \in (i_s, j_s), A(k_s) \in (i_t, j_t)$.

We introduce syntax into this induction process by defining every node in the source tree as a tuple of three entities $Tree(F_I^1) = \{n_{si} : \langle c_{si}, l_{si}, p_{si} \rangle\}$, where $c_{si} \in C_S$ is a set of spans of the constituents defined as $C_S = \{c_{s_1}..c_{s_M} : c_{s_m} = (s_i, s_j), s_i, s_j \in 1..I\}$. $l_{si} \in L_S = \{l_{s_1}..l_{s_M}\}$ where L_S is a set of labels which bears a one-one

mapping with constituents in C_S. $p_{si} \subseteq P_S$ is set of constraints related to hierarchical nature of the nodes in the tree. For purpose of rule induction, we only concentrate on predicates such as $unary(C_s)$, $parent(C_{si}, C_{sj})$. Similarly the target side tree can be defined as $Tree(E_J^1) = \{n_{tj} = \langle c_{tj}, l_{tj}, p_{tj} \rangle\}$ where $C_T = \{c_{t_1}..c_{t_N} : c_{t_n} = (t_i, t_j), t_i, t_j \in 1..J\}$, $L_T = \{l_{t_1}..l_{t_N}\}$ and P_T.

When possible, we use trees on both sides for extracting syntax motivated translation models. Most languages, however, do not have syntactic parsers available and the ones that do may not provide reliable syntactic analysis. Independent resources for annotation tasks such as base-np chunking, part-of-speech tagging, etc are often available. Our formalization of the rule induction task empowers the framework to be general and can accommodate partial syntax for at least one side of the language pair. For example, one can imagine a chunking algorithm providing the constituency information for the target side as C_T. A part-of-speech tagger can provide reliable labels for the terminals as L_T. Our framework can thus enable federation of the best available syntactic annotation for that language, possibly coming from multiple sources and not necessarily from a single reliable parse tree.

2.2 Rule Extraction

Our extractor runs in two phases to extract minimal rules. In the first phase we mark nodes in the source tree that align to the target tree and extract syntactic phrases; also called 'lexical rules'. In the second phase we use these marked nodes to extract 'hierarchical rules'.

2.2.1 Lexical Rule Extraction: Phrase tables

The input to this phase is an entry from the parallel corpus $D_i = \langle E, F, A \rangle$ and the corresponding syntax information from both sides $Tree(F_I^1) = \{n_{si} : \langle c_{si}, l_{si}, p_{si} \rangle\}$ and $Tree(E_J^1) = \{n_{ti} : \langle c_{ti}, l_{ti}, p_{ti} \rangle\}$. We start by traversing the target-side syntax tree starting from the root. At each node $c_m \in C_T$ of the target tree, with span (i, j), we calculate the maximal contiguous sub-sentential segment (k, l) in the source sentence that is consistently aligned with all the words in the yield of this target node . We then perform a check to see if this source span corresponds to a node in the source tree that has not already been aligned i.e $\exists c_n \in C_S : c_n = (k, l), \forall c_p : (c_n, c_p) \notin A_N$. A_N is the "Node Alignment" that defines alignment links between the nodes in the parallel trees. If it does, then c_m aligns to c_n and we mark them as a synchronous decomposition node pair. If it does not, we check to see if the span of the immediately higher node in the tree that subsumes the projected span is consistently aligned. If so, we mark that node as aligned. All unaligned words in the word alignment are ignored while checking for consistency.

The design of the alignment algorithm itself is constrained by the 'structural constraints' introduced by the trees as P_T and P_S. One design assumption is that if two nodes are aligned, then all the child nodes below one node can only align to child nodes below the other node. Another assumption we make is when faced with unary rules, we always choose the lower node for alignment. Yet another constraint that we follow is that pre-terminal nodes can not align to non-terminal nodes and vice-versa. One can imagine relaxing such structural constraints or adding more as variations in the rule extraction framework.

When no nodes can be aligned any further, we output the 'lexical rules' from the sentence pair by traversing all the aligned nodes in A_N and gathering the yield of the source node and the corresponding target node as a "flat rule". The lexical rules are of the format below, where $c_s \in L_S$ and $c_t \in L_T$ represent syntactic categories and w_s and w_t are the word or phrase strings for the source- and target-sides correspondingly. The phrasal entries are collected from the entire corpus and scored by conditioning on the source side together with the label for assigning probabilities and form a syntactic phrase table.

$$c_s :: c_t \rightarrow [w_s] :: [w_t]$$

Our alignment algorithm does not depend on the label sets L_S or L_T. Given that the original syntactic labels associated with the two parse trees are designed independent from each other, they may be sub-optimal for MT. (Huang and Knight, 2006) achieved improved translation quality by relabeling trees from which translation models were extracted. Decoupling the choice of labels from the alignment

algorithm and delaying the assignment of labels until the output phase, enables our framework to experiment with various labeling strategies. In all our experiments, we retain the labels from both the sets. Exploring other possibilities for labeling, although interesting, is beyond the scope of this paper.

2.2.2 Hierarchical Rule Extraction: Synchronous Grammar

Given two synchronous trees and their node alignment A_N, we developed a tree traversal algorithm that decomposes parallel trees into all minimal tree fragments. Our tree fragment extraction algorithm operates by an in-order traversal of the trees top down, starting from the root nodes. The traversal can be guided by either the source or target parse tree. Each node in the tree that is marked as an aligned node triggers a decomposition. The subtree that is rooted at this node is removed from the currently traversed tree. A copy of the removed subtree is then recursively processed for top-down decomposition. If the current tree node being explored is not an aligned node (and thus is not a decomposition point), the traversal continues down the tree, possibly all the way to the leaves of the tree. Decomposition is performed on the corresponding parallel tree at the same time. We apply this process on all the aligned constituent nodes (decomposition points) to obtain all possible decomposed synchronous tree fragment pairs from the original parallel parse trees. This results in a collection of all minimal synchronous subtree fragments. Flattening these subtree fragments and tracking the movement of constituents between languages as alignment variables produces our synchronous context-free grammar (SCFG) shown as below. These can be scored similar to the lexical rules to produce a probabilistic-SCFG.

$$NP :: NP \rightarrow [DET^1 N^2 deN^3] :: [DET^1 N^3 N^2]$$

2.3 Other configurations

The rule induction algorithm can be run in several other 'configurations' which vary over the spectrum of constraints imposed by the syntactic trees. The effect of word-alignment on rule extraction is an important one and is explored in (Lavie et al., 2008). In this paper, we are interested in the constraints introduced by the parse trees and study them in detail.

Syntax trees from both sides introduce constraints on the possible segmentations. One variation to the above tree-tree model is to extract rules using target tree only. This can be seen as an instance of the framework, where $C_S, L_S, P_S = NULL$. A dependency tree can be seen as constituting a hidden unlabeled constituent tree. In such a case, the rule extraction can be used in a configuration where $L_T, L_S =$ 'X', simulating a 'hiero' style rule extraction (Chiang, 2005). As seen previously, variations with the choice of label sets L_S, L_T are also possible. For a language where only base-np chunkers and part-of-speech taggers are available, the framework can be instantiated as $C_T = \{(i, j)\}$ as the set of span boundaries from the chunker and $L_T = L_{pos} \cup \{'X'\}$ where L_{pos} come from the part-of-speech tagger and can be assigned to the preterminals, and 'X' label to the non-terminals. We experiment with two configurations, using tree on one side vs. both sides.

3 Restructuring Parse Trees

The parsers that generate syntactic analysis are built under varying assumptions of grammar, label granularity and structural constructions, and most importantly purpose of the grammar , which most often is not MT. We discuss our approach to restructuring which has two primary operations, the first is creating extra parse nodes that are licensed by the word alignment and the target-side parse tree and introducing them into the original source-side parse tree. The second operation is to merge some of these nodes that improve the quality of translation. We will discuss this in context of the example in Figure 1 where the initial alignment A_N is shown.

The input to the restructuring process is the source-side tree $Tree(F_I^1) = \{n_{si} : \langle c_{si}, l_{si}, p_{si} \rangle\}$ and target-side tree $Tree(E_J^1) = \{n_{ti} : \langle c_{ti}, l_{ti}, p_{ti} \rangle\}$, the word alignment mapping A, the subtree alignment information A_N. Given this, we now describe the two primary operations to be performed in a sequence.

3.1 Introduce Operation

We first traverse the target side parse tree $Tree(E)$ in depth-first fashion. For each node $n_t \in Tree(E)$ that is not already aligned as given by A_N, we find

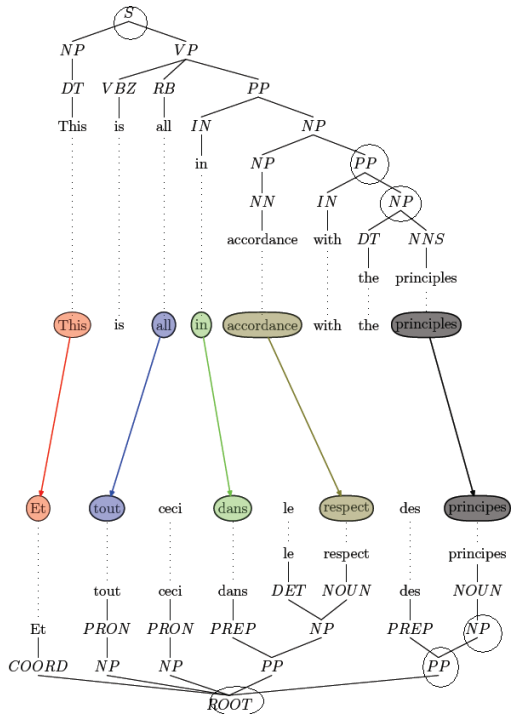

Figure 1: English and French parse trees with initial node-alignment

a valid projection for the yield of the node in the source sentence as licensed by the word alignment. Let the indices of the node be i and j. We now introduce a new node instance $n_s =< c_s, l_s, p_s >$ where $c_s = (i, j)$ into the source tree that respects the following two conditions expressed as structural constraints in p_s -

- Already existing nodes that cover the complete or partial span of i and j, are made as children to the new node. The new node is a parent node.
- The new node n_s is then attached to the immediate parent that governs the yield from i to j.

Figure 2 shows introduction of nodes into the original French parse tree, that make it isomorphic to the English parse tree after combination operation. Choice of label l_s for this node comes from the label l_t of the projected node n_t. One could imagine other methods of assigning labels for new nodes, or have a generic 'X' label. These decisions will affect scoring and downstream application of grammar, which we currently do not explore in this paper.

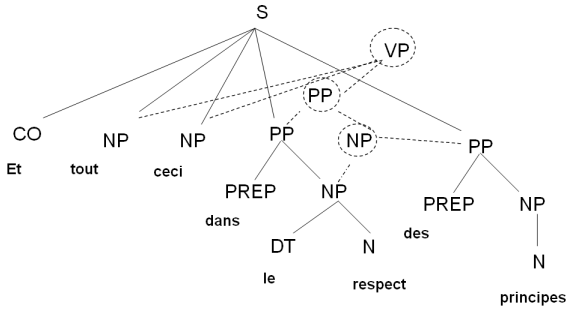

Figure 2: Creating isomorphic structure with the English Tree

3.2 Combine Operation

The graph like structure obtained after the above operation has spurious nodes some of which are not necessarily unique. The structure can be seen as a packed forest with two trees in it - the original tree and the projected structure tree. In this step, we produce a tree from it by performing a set of merging operations which make sure that we end up with a final tree structure We perform the below two operations, which basically ensure that every node in the tree has only one parent:

- For each introduced node n_s, we pick its parent node n'_s in the source tree $Tree(F)$. If n'_s is aligned to the same target-side node as n_s, we drop n_s .
- All the nodes in the original tree which do not correspond to any decomposition points as decided by the tree-tree alignment function A_N are dropped.

We can now use the modified parse tree for the source-side and the original target-side parse tree to extract lexical rules. Table 1 shows all lexical rules that are extracted by this approach. We notice that the translations are more precise, as the phrasal boundaries are provided by the source and target-side syntax.

4 Restructuring applied to Source and Target trees

In the case where we use trees on both sides we only extract spans that are consistent with the word alignment, and form nodes in the trees on both sides. In restructuring scenario, the space of phrases we consider for extraction is the entire set of nodes in the

English	French
This	Et
the principles	principes
with the principles	des principes
accordance with the ...	respect des principes
in accordance with the ...	dans le respect des ...
is all in accordance with..	tout ceci dans le ...

Table 1: Lexical rules extracted using the restructured parse tree

target tree C_T, and prefer nodes in source tree to define the span boundaries. Therefore we extract considerably more phrases than by using trees on both sides, and more precise translation equivalents than by using tree on one side.

This however has a drawback which is that we overwrite syntactic constituent boundaries coming from the second side of the language pair. We therefore propose a "symmetric" rule induction technique that performs the restructuring process not just in one direction but also in the opposite direction in an 'inclusion' mode to not throw away any syntactic information. The extraction is done independently in both directions and the scores are accumulated over the obtained rules. To ensure that we are not double counting nodes during symmetrization, we aggregate all the phrases in the 'direction of translation', French-English for example, and remove any resulting duplicates.

4.1 Effect on Grammar Extraction

Restructuring has two main effects on syntactic tree. Firstly, if we chose to restructure the source tree to be isomorphic to the target tree, then we introduce syntactic nodes into the source tree, which were absent, perhaps due to differences in parser design. Secondly, during restructuring we may modify or over-write existing structure of the tree in a preference for the 'isomorphism' property, which is crucial for extracting generalized synchronous grammars.

This is acceptable in a case where parser for one language is evidently better in quality than the parser for the second language, or the parser produces completely shallow structures that are not suitable for generalized grammar extraction. While working with languages where both the parsers are reliable isomorphism may still be desirable for generalization. However, placing a prior preference on one

Rule	Res	Sym
PRP::CL [I] $\rightarrow [Je]$	1	1
AUX::V [am] $\rightarrow [sui]$	1	1
JJ::A [sure] $\rightarrow$ [certaine]	1	1
VP::VP [am sure] $\rightarrow$ [sui certaine]	1	1
VP::VP [AUX$_1 ADJP_2$] $\rightarrow [V_1 A_2]$	1	1
NP::NP [PRP$_1$] $\rightarrow [CL_1]$	1	1
S::Sint [NP$_1 VP_2$] $\rightarrow [NP_1 VP_2]$	1	1
VN::VN [I am] $\rightarrow$ [Je sui]	-	1
VN::VN [PRP$_1 AUX_2$] $\rightarrow [CL_1 V_2]$	-	1
AP::AP [sure] $\rightarrow$ [certaine]	-	1
S::Sint [VN$_1 AP_2$] $\rightarrow [VN_1 AP_2]$	-	1

Table 2: Rules induced by "Res"-Restructuring from English side and "Sym"-Symmetric extraction

parser over the other suppresses generalization over constituents from the second side. Symmetrization comes to the rescue here, by not losing any syntax as we restructure from both sides. This also has the effect of introducing syntax from one tree into the other, which reduces divergences resulting from parser differences. It is clear that 'symmetrization' produces larger syntactic phrase tables than any other method of syntax based extraction. We also argue here that the grammars extracted by restructuring and symmetrization are useful for their increased expressiveness and compositionality both of which are required for MT.

Figure 3 shows symmetric rule extraction on a sub-sentence from our training set. Nodes introduced by restructuring are shown in the figure as boxes and dotted lines. This is a case of 'copula' which is handled differently by the two parsers. The English parser (Klein and Manning, 2002) attaches the main verb to the adjectival phrase, while the French parser [1] makes a 'verbal nucleus' constituent out of the 'clitic pronoun' and the main verb. Table 2 shows the output from symmetric rule induction and compare it with rules extracted by restructuring only the French parse tree using the English parse tree. We can observe that when restructuring from one side we lose grammar rules related to composing the verb 'am' with the pronoun 'I', but symmetrization fills the gap. The new generalized rule $VN :: VN[PRP_1 AUX_2] \rightarrow [CL_1 V_2]$ can now combine other verbs like 'see','think' to form constituents 'I see', 'I think' etc, which were earlier not possible.

[1] http://code.google.com/berkeleyparser

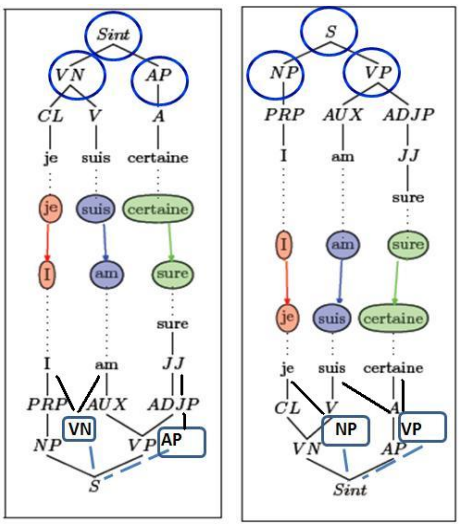

Figure 3: Symmetric Rule Induction: Joint restructuring

5 Evaluation

5.1 Experimental Setup

We build a French to English translation system using our decoding framework. We do not exploit the hierarchical nature of the decoder, as the translation models with which we would like to experiment are flat syntactic phrases. The parallel data we used to build our translation models is the Europarl data consisting of 1.3M translation sentence pairs. The English side of the corpus is parsed using the Stanford parser(Klein and Manning, 2002). The French side of the corpus was parsed by the Berkeley Parser. Word alignments for the parallel corpus were obtained using GIZA++ (Och and Ney, 2003) followed by a symmertrization technique called 'sym2' from the Thot toolkit (Ortiz-Martínez et al., 2005). We used this technique as it was shown to provide good node alignment results across trees in both (Lavie et al., 2008) and (Tinsley et al., 2007). We then perform the extraction of the phrase pairs under all modes of rule learning - tree on one side, trees on both sides, restructuring tree and symmetric rule induction. Since we are interested in studying the affect of the lexical coverage licensed by these different extraction scenarios, in all our experiments we run our decoder in a monotonic mode without any hierarchical models.

We performed translation experiments using the experimental setup defined above and use Stat-XFER (Lavie, 2008) as the decoding framework. We built a suffix array language model (SALM) (Zhang and Vogel, 2006) over 430 million words including the English side of the parallel corpus. The weights on the features are tuned using standard MERT (Och, 2003) techniques over a 600-sentence dev set. The test set used was released by the WMT shared task 2007 and consists of 2000 sentences. When run without hierarchical syntax, this decoder is very similar to Moses decoder (et al, 2007).

5.2 Results

The results are shown in Table 3. The overall problem of low coverage from using trees on both sides can be seen in the final translation quality too. Using syntax on one side produces syntax tables of larger coverage which also reflects in MT quality as judged both by BLEU and METEOR(Banerjee and Lavie, 2005) metrics. Our non-isomorphic tree restructuring technique that benefits from syntactic boundaries from both trees shows significant [2] improvement over the two other approaches. Symmetric rule induction that is a union of rules extracted by restructuring both trees has similar benefits from restructuring and also produces the largest possible syntactic phrase table. Although these results are slightly worse when compared to standard PB-SMT baseline (30.18 BLEU), it is to be noted that we are working with only syntactic phrase tables which are clean resources and are still relatively smaller in size. The goal of our work is not to compete with non-syntactic systems, but to create better syntactic phrase tables and grammars which can then help building of syntax based MT systems. One can leverage from our translation models by introducing hierarchical syntax and/or augmenting non-syntactic phrase tables as observed by (Hanneman and Lavie, 2009).

In all our experiments with restructuring we retain labels from both sides, and so the probability space of scores in segmented among $L_S \cup L_T$ labels. Non-syntactic phrase tables are modeled as being generated from a single label leading to reliable estimates. In future, we wish to explore the label granularity problem to choose the right label set for syntax that

[2] For the final version of the paper we would be performing statistical significance tests

	Dev-Set	Test-Set	
Syntax Trees	**BLEU**	**BLEU**	**METEOR**
Target side only	25.12	25.52	57.25
Source,Target	20.85	21.13	53.05
Restructured Source	26.03	26.17	57.75
Symmetrized	26.52	26.45	59.79

Table 3: Evaluation of French-English MT System

enables application of grammar rules, but curbs the fragmentation problem.

6 Conclusion

In this paper we proposed a generic rule induction framework that incorporates syntactic constraints from both sides. We decompose syntactic tree as sets of structural, constituent and labeling constraints which empowers the framework to work with independent annotation schemes. We observed that using trees on both sides generates syntactically motivated and precise phrase tables, but of low coverage due to non-isomorphic nature of the parse trees that come from two completely independent parsers and parsing models. We proposed a novel technique for modifying the non-isomorphic parse tree structures for the target language, by introducing an isomorphic backbone parse structure into the target tree We finally evaluated applying our technique from both sides to produce large syntactic phrase tables. We have evaluated the syntax motivated phrase models in a French-English MT system and the results show significant improvements in translation quality.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proc. of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, June.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proc. 43rd ACL*, pages 263–270, Morristown, NJ, USA.

Philipp Koehn et al. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL Demonstration Session*, pages 177–180, Jun.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In Susan Dumais; Daniel Marcu and Salim Roukos, editors, *Proc. HLT-NAACL 2004*, pages 273–280, Boston, Massachusetts, USA, May 2 - May 7.

Greg Hanneman and Alon Lavie. 2009. Decoding with syntactic and non-syntactic phrases in a syntax-based machine translation system. In *Proc. SSST-3 Workshop at NAACL 2009*.

Bryant Huang and Kevin Knight. 2006. Relabeling syntax trees to improve syntax-based machine translation quality. In *Proc. of the main conference on HLT-NAACL*, pages 240–247, Morristown, NJ, USA. Association for Computational Linguistics.

Dan Klein and Christopher D. Manning. 2002. Fast extract inference with a factored model for natural language parsing. In *Proc. of NIPS 15*.

Alon Lavie, Alok Parlikar, and Vamshi Ambati. 2008. Syntax-driven learning of sub-sentential translation equivalents and translation rules from parsed parallel corpora. In *Proc. of the ACL-08: HLT SSST-2 Workshop*, pages 87–95, Columbus, Ohio, June. Association for Computational Linguistics.

Alon Lavie. 2008. Stat-xfer: A general search-based syntax-driven framework for machine translation. In Alexander F. Gelbukh, editor, *CICLing*, volume 4919 of *Lecture Notes in Computer Science*, pages 362–375. Springer.

Daniel Marcu, Wei Wang, Abdessamad Echihabi, and Kevin Knight. 2006. Spmt: Statistical machine translation with syntactified target language phrases. In *Proc. of the 2006 Conference on EMNLP*, pages 44–52, Sydney, Australia, July. Association for Computational Linguistics.

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

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of the Association for Computational Linguistics*, Sapporo, Japan, July 6-7.

D. Ortiz-Martínez, I. García-Varea, and F. Casacuberta. 2005. Thot: a toolkit to train phrase-based statistical translation models. In *Tenth Machine Translation Summit*. AAMT, Phuket, Thailand, September.

John Tinsley, Venstislav Zhechev, Mary Hearne, and Andy Way. 2007. Robust Language-Pair Independent Sub-Tree Alignment. In *Proc. of Machine Translation Summit XI*, pages 467–474, Copenhagen, Denmark.

Wei Wang, Kevin Knight, and Daniel Marcu. 2007. Binarizing syntax trees to improve syntax-based machine translation accuracy. In *Proc. of the 2007 Joint Conference of EMNLP-CoNLL*, pages 746–754.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proc. of ACL '01*, pages 523–530, Morristown, NJ, USA.

Ying Zhang and Stephan Vogel. 2006. Salm: Suffix array and its applications in empirical language processing. Technical Report CMU-LTI-06-010, Pittsburgh PA, USA, Dec 2006.