Syntax-driven Learning of Sub-sentential Translation Equivalents and Translation Rules from Parsed Parallel Corpora

Alon Lavie
alavie@cs.cmu.edu
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Alok Parlikar
aup@cs.cmu.edu
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Vamshi Ambati
vambati@cs.cmu.edu
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Abstract

We describe a multi-step process for automatically learning reliable sub-sentential syntactic phrases that are translation equivalents of each other and syntactic translation rules between two languages. The input to the process is a corpus of parallel sentences, word-aligned and annotated with phrase-structure parse trees. We first apply a newly developed algorithm for aligning parse-tree nodes between the two parallel trees. Next, we extract all aligned sub-sentential syntactic constituents from the parallel sentences, and create a syntax-based phrase-table. Finally, we treat the node alignments as tree decomposition points and extract from the corpus all possible synchronous parallel tree fragments. These are then converted into synchronous context-free rules. We describe the approach and analyze its application to Chinese-English parallel data.

1 Introduction

Phrase-based Statistical MT (PB-SMT) (Koehn et al., 2003) has become the predominant approach to Machine Translation in recent years. PB-SMT requires broad-coverage databases of phrase-to-phrase translation equivalents. These are commonly acquired from large volumes of automatically word-aligned sentence-parallel text corpora. Accurate identification of sub-sentential translation equivalents, however, is a critical process in all data-driven MT approaches, including a variety of data-driven syntax-based approaches that have been developed in recent years. (Chiang, 2005) (Imamura et al., 2004) (Galley et al., 2004).

In this paper, we describe a multi-step process for automatically learning reliable sub-sentential syntactic phrases that are translation equivalents of each other and syntactic translation rules between two languages. The input to the process is a corpus of parallel sentences, word-aligned and annotated with phrase-structure parse trees for both languages. Our method consists of three steps. In the first step, we apply a newly developed algorithm for aligning parse-tree nodes between the two parallel trees. In the second step, we extract all aligned sub-sentential syntactic constituents from the parallel sentences, and create a syntax-based phrase-table. Our syntactic phrases come with constituent "labels" which can guide their syntactic function during decoding. In the final step, we treat the node alignments as tree decomposition points and extract from the corpus all possible synchronous parallel tree fragments. These are then converted into synchronous context-free rules. Our methods do not depend on any specific properties of the underlying phrase-structure representations or the parsers used, and were designed to be applicable even when these representations are quite different for the two languages.

The approach described is used to acquire the resources for a statistical syntax-based MT approach that we have developed (Stat-XFER), briefly described below. The resulting resources can, however, be used in any syntax-based data-driven MT approach other than our own. The focus of this paper is on our syntax-driven process for extracting phrases and rules from data. We describe the approach and analyze its effectiveness when applied to large-volumes of Chinese-English parallel data.

1.1 The Stat-XFER MT Framework

Stat-XFER is a search-based syntax-driven framework for building MT systems. The underlying formalism is based on synchronous context-free grammars. The synchronous rules can optionally be augmented by unification-style feature constraints. The synchronous grammars can be acquired automatically from data, but also manually developed by experts. A simple example transfer-rule (for Chinese-to-English) can be seen below:

```
{NP,1062753}
NP::NP [DNP NP] -> [NP PP]
(
(*score* 0.946640316205534)
(X2::Y1)
(X1::Y2)
)
```

Each rule has a unique identifier followed by a synchronous rule for both source and target sides. The alignment of source-to-target constituents is explicitly represented using 'X' indices for the source side, and 'Y' indices for the target side. Rules can also have lexical items on either side, in which case no alignment information is required for these elements. Feature constraints can optionally be specified for both source and target elements of the rule. We do not address the learning of feature constraints in the work described here, and concentrate only on the acquisition of the synchronous CFG rules. The rules can be modeled statistically and assigned scores, which can then be used as decoding features.

The Stat-XFER framework also includes a fully-implemented transfer engine that applies the transfer grammar to a source-language input sentence at runtime, and produces collections of scored word and phrase-level translations according to the grammar. These are collected into a lattice data-structure. Scores are based on a log-linear combination of several features, and a beam-search controls the underlying parsing and transfer process. A second-stage monotonic decoder is responsible for combining translation fragments into complete translation hypotheses (Lavie, 2008)

2 PFA Algorithm for Node Aligment

2.1 Objectives of the Algorithm

Our objective of the first stage of our approach is to detect sub-sentential constituent correspondences in parallel sentences, based on phrase-structure parses for the two corresponding sentences. Given a pair of parallel sentences and their corresponding parse trees, our goal is to find pairings of nodes in the source and target trees whose yields are translation equivalents of each other. Our current approach only considers complete constituents and their contigious yields, and will therefore not align discontiguous phrases or partial constituents. Similar to phrase extraction methods in PB-SMT, we rely on word-level alignments (derived manually or automatically) as indicators for translation equivalence. The assumption applied is that if two words are aligned with each other, they carry the same meaning and can be treated as translation equivalents. Constituents are treated as compositional units of meaning and translation equivalence.

2.2 Related Work

Aligning nodes in parallel trees has been investigated by a number of previous researchers. (Samuelsson and Volk, 2007) describe a process for manual alignment of nodes in parallel trees. This approach is well suited for generating reliable parallel treebanks, but is impractical for accumulating resources from large parallel data. (Tinsley et al., 2007) use statistical lexicons derived from automatic statistical word alignment for aligning nodes in parallel trees. In our approach, we use the word alignment information directly, which we believe may be more reliable than the statistical lexicon. (Groves et al., 2004) propose a method of aligning nodes between parallel trees automatically, based on word alignments. In addition to the word alignment information, their approach uses the constituent labels of nodes in the trees, and the general structure of the tree. Our approach is more general in the sense that we only consider the word alignments, thereby making the approach applicable to any parser or phrase-structure representation, even ones that are quite different for the two languages involved.

2.3 Unaligned Words and Contiguity

Word-level alignment of phrase-level translation equivalents often leaves some words unaligned. For example, some languages have articles, while others do not. It is thus reasonable to expect that constituent pairs in parallel trees that are good translation equivalents of each other may contain some unaligned words. Our PFA node-alignment algorithm allows for such constituents to be matched.

Different languages have different word orders. In English, an adjective always comes before a noun, while in French, in most cases, the adjective follows its noun. Our node alignment algorithm allows aligning of constituents regardless of the word order expressed by the linear precedence relation of their sub-constituents. As long as one piece of contiguous text dominated by a node covers the same word-level alignments as the yield of a node in the parallel tree, the two nodes can be aligned.

2.4 Wellformedness constraints

Given a pair of word-aligned sentences and their corresponding parse trees S and T, represented as sets of constituent nodes, our PFA node alignment algorithm produces a collection of aligned node-pairs (S_i, T_j). The underlying assumptions of compositionality in meaning and word-level alignments being indicative of translation equivalence lead directly to the following node alignment wellformedness criteria:

1. If a node S_i is linked to a node T_j, then any node within the subtree of node S_i can only be linked to nodes within the subtree of node T_j.

2. If a node S_i is linked to a node T_j, then any node that dominates the node S_i can only be linked to nodes that dominate the node T_j.

3. If a node S_i is linked to a node T_j, then the word alignments of the yields of the two constituents must satisfy the following:

   (a) Every word in the yield of the node S_i must be aligned to one or more words in the yield of the node T_j, or it should be unaligned.

   (b) Every word in the yield of the node T_j must be aligned to one or more words in the yield of the node S_i, or it should be unaligned.

   (c) There should be at least one alignment between the yields of nodes S_i and T_j. Thus, the words in the yields can not all be unaligned.

2.5 Arithmetic Representation

Our PFA algorithm uses a arithmetic mapping that elegently carries over the constraints characterized by the wellformedness constraints elaborated above. This mapping is designed to ensure that each aligned word, which carries a distinct "piece of meaning" can be uniquely identified, and also inherently reflects the compositional properties of constituent translation equivalence. This is accomplished by assigning numerical values to the nodes of the two parse trees being aligned, in a bottom-up fashion, starting from the leaf nodes of the trees. Leaf nodes that correspond to words that are aligned are each assigned a unique prime number. Unaligned leaf nodes are assigned a value of "1". Constituent nodes in the parse trees are then assigned a value that is the product of all its sub-constituent nodes. Because of the arithmetic property that any composite number can be uniquely factored into primes, it should be evident that the value of every constituent node uniquely identifies the aligned words that are covered by its yield. Consequently, by assigning *the same* prime values to the aligned words of both trees, retrieving aligned constituent nodes is as simple as finding the set of nodes in the two trees that carry the same numerical value. Note that by assigning values of "1" to unaligned words, these unaligned words do not influence the numerical values assigned to constituent nodes, thus reflecting their treatment as "don't cares" with respect to the translation equivalence of constituent nodes.

2.6 Description of the PFA Algorithm

The PFA algorithm uses the concept of 'composite meaning as prime factorization', and hence the name (Prime Factorization and Alignments). The algorithm assigns values to the leaf nodes, propogates the values up the tree, and then compares the node values across the trees to align the nodes. As described above, leaf nodes which have word alignments are assigned unique prime numbers, and the

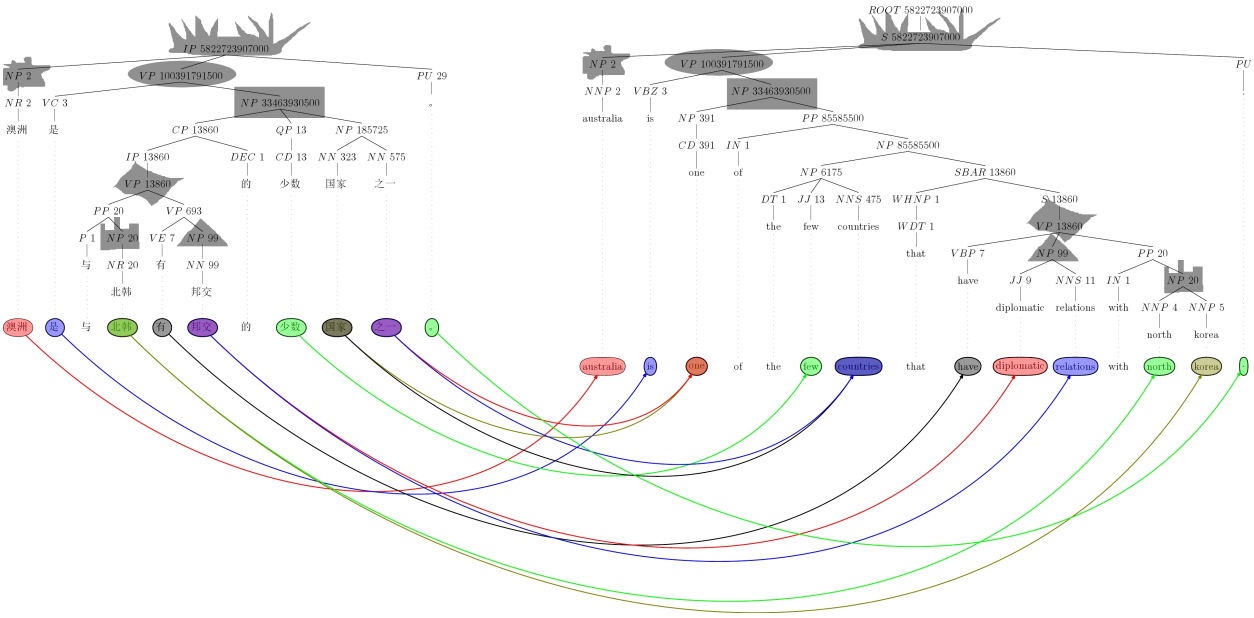

Figure 1: Node-Aligned Parallel Sentences

same prime is assigned to the corresponding aligned words in the parallel sentences. Leaf nodes corresponding to unaligned words are assigned the value "1". The treatment of "one-to-many" word alignments is a special case. Such alignments are considered to carry the same meaning, and should thus be assigned the same value. To accomplish this, if a single word is aligned to multiple words in the other language, we assign the same prime number to all words on the "multiple" side, and assign the product of these to the single word equivalent.

Another special case is when the parse trees contain unary productions. In this case, the values of both nodes involved in this production are the same. Our node alignment algorithm breaks this "tie" by selecting the node that is "lower" in the tree (the daughter node of the unary production). A similar situation with two nodes being assigned identical values can arise when one or more unaligned words are attached directly to the parent node. Here too, our algorithm aligns the "lower" node and leaves the "higher" node unaligned. These decisions reflect our desire to be conservative with respect to such ambiguous cases, and their implications on the notion of translational equivalence. This also provides some robustness against noisy alignments.

It is straightfoward to verify that the PFA algorithm satisfies the wellformedness constraints described above. Also, since multiplication is commutative, the algorithm is not effected by differing word orders within parallel constituent structures.

The PFA algorithm run on a sample Chinese-English parallel sentence is shown in Figure 1. The value of each node as shown as a part of its label. The aligned nodes are marked by shapes. A triangle aligns to a triangle, and squares to squares.

3 Syntax-based Sub-sentential Phrase Extraction

The alignment of nodes as described in the previous section allows us to build a comprehensive syntax-based phrase-to-phrase translation lexicon from a parallel corpus. To build a syntax-based "phrase table", we simply extract all aligned constituent nodes along with their yields and enter them into a database, while accumulating frequency counts. In addition to the source-to-target phrase correspondences, we record the constituent labels of the aligned constituent nodes on both the source and target sides (which may be different). These labels "connect" the phrases with synatactic transfer rules during decoding. The set of phrases extracted from the example sentence in Figure 1 is shown in Figure 2.

Source Category	Target Category	Source	Target
IP	S	澳洲 是 与 北韩 有 邦交 的 少数 国家 之一 。	Australia is one of the few countries that have diplomatic relations with North Korea.
VP	VP	是 与 北韩 有 邦交 的 少数 国家 之一	is one of the few countries that have diplomatic relations with North Korea
NP	NP	与 北韩 有 邦交 的 少数 国家 之一	one of the few countries that have diplomatic relations with North Korea
VP	VP	与 北韩 有 邦交	have diplomatic relations with North Korea
NP	NP	邦交	diplomatic relations
NP	NP	北韩	North Korea
NP	NP	澳洲	Australia

Figure 2: Phrases extracted from Aligned Nodes

The process of building syntax-based "phrase tables" from large corpora of sentence-parallel data is quite similar to the corresponding process in phrase-based SMT systems. Our phrase correspondences, however, only reflect contiguous and complete constituent correspondences. We also note that the extracted phrase tables in both approaches can be modeled statistically in similar ways. Similar to common practice in PB-SMT, we currently use the frequency counts of the phrases to calculate relative likelihood estimates and use these as features in our Stat-XFER decoder.

4 Evaluation of the PFA algorithm

The accuracy of our node alignment algorithm depends on both the quality of the word alignments as well as the accuracy of the parse trees. We performed several experiments to assess the effects of these underlying resources on the accuracy of our approach. The most accurate condition is when the parallel sentences are manually word-aligned, and when verified correct parse trees are available for both source and target sentences. Performance is expected to degrade when word alignments are produced using automatic methods, and when correct parse trees are replaced with automatic parser output. In these experiments, we used a manually word-aligned parallel Chinese-English TreeBank consisting of 3342 parallel sentences.

4.1 Manual Constituent Node Alignments

We first investigated the accuracy of our approach under the most accurate condition. We sampled 30 sentences from the Chinese-English treebank corpus. A bilingual expert from our group then manually aligned the nodes in these trees. These node alignments were then used as a "gold standard". We then used the accurate parse trees and the manually created word alignments for these sentence pairs, and ran the PFA node algorithm, and compared the resulting node alignments with the gold standard alignments. The Precision, Recall, F-1 and F-0.5 results are reported in Table 1.

Precision	Recall	F-1	F-0.5
0.8129	0.7325	0.7705	0.7841

Table 1: Accuracy of PFA Node Alignments against Manual Node Alignments

We manually inspected cases where there was a mismatch between the manual and automatic node alignments, and found several trends. Many of the alignment differences were the result of one-to-many or many-to-many word alignemnts. For example, in some cases a verb in Chinese was word-aligned to an auxiliary and a head verb on the english side (e.g. `have` and `put`). The PFA algorithm in this case node-aligns the `VP` that governs the Chinese verb to the `VP` that contains both auxiliary and head verbs on the English side. The gold standard human alignments, however, in some cases, aligned the `VP` of the Chinese verb to the English `VP` that governs just the main verb. Other mismatches were attributed to errors or inconsistencies in the manual word alignment and to the treatment of traces and fillers in the parse trees.

4.2 Effect of Using Automatic Word Alignments

We next tested how sensitive the PFA algorithm is to errors in automatic word alignment. We use the entire 3342 sentences in the parallel treebank for this experiment. We first ran the algorithm with the correct parse trees and manual word-alignments as input. We use the resulting node alignments as the gold standard in this case. We then used GIZA++ to get bidirectional word alignments, and combined them using various strategies. In this scenario, the trees are high-quality (from the treebank), but the alignments are noisy. The results obtained are shown in Table 2. Unsurprisingly, the "Union" combination method has the best precision but worst recall, while the "Intersection" combination method has the best recall but worst precision. The four

Comb Method	Prec	Rec	F-1	F-0.5
Intersection	0.6382	0.5395	0.5846	0.6014
Union	0.8114	0.2915	0.4288	0.5087
Sym1	0.7142	0.4534	0.5546	0.5992
Sym2	0.7135	0.4631	0.5616	0.6045
Grow-Diag-Final	0.7777	0.3462	0.4790	0.5493
Grw-Diag-Fin-And	0.6988	0.4700	0.5619	0.6011

Table 2: Manual Trees, Automatic Node Alignments

other methods for combining word alignments fall in between. Three of the four (all except "grow-diag-final") behave quite similarly. We generally believe that precision is somewhat more important than recall for this task, and have thus used the "sym2" method (Ortiz-Martínez et al., 2005) (which has the best F-0.5 score) for our translation experiments.

4.3 Effect of Using Automatic Parses

We evaluated the effect of parsing errors (as reflected in automatically derived parse trees) on the quality of the node alignments. We parsed the treebank corpus on both English and Chinese using the Stanford parser, and extracted phrases using manual word alignments. Compared to the phrases extracted from the manual trees, we obtained a precision of 0.8749, and a recall of 0.7227, that is, an F-0.5 measure of 0.8174. We then evaluated the most 'noisy' condition that involves both automatic word alignments and automatic parse trees. We evaluated the phrase extraction with different Viterbi combination strategies. The 'sym2' combination gave the best results, with a precision of 0.6251, recall of 0.3566, thus an F-0.5 measure of 0.4996.

5 Synchronous Tree Fragment and CFG Rule Extraction

5.1 Related Work

Syntax-based reordering rules can be used as a preprocessing step for PB-SMT (and other approaches), to decrease the word-order and syntactic distortion between the source and target languages (Xia and McCord, 2004). A variety of hierarchical and syntax-based models, which are applied during decoding, have also been developed. Many of these approaches involve automatic learning and extraction of the underlying syntax-based rules from data. The underlying formalisms used has been quite broad and include simple formalisms such as ITGs (Wu, 1997), hierarchical synchronous rules (Chiang, 2005), string to tree models by (Galley et al., 2004) and (Galley et al., 2006), synchronous CFG models such (Xia and McCord, 2004) (Yamada and Knight, 2001), synchronous Lexical Functional Grammar inspired approaches (Probst et al., 2002) and others.

Most of the previous approaches for acquiring syntactic transfer or reordering rules from parallel corpora use syntactic information from only one side of the parallel corpus, typically the target side. (Hearne and Way, 2003) describes an approach that uses syntactic information from the source side to derive reordering subtrees, which can then be used within a "data-oriented translation" (DOT) MT system, similar in framework to (Poutsma, 2000). Our work is different from the above in that we use syntactic trees for both source and target sides to infer constituent node alignments, from which we then learn synchronous trees and rules. Our process of extraction of rules as synchronous trees and then converting them to synchronous CFG rules is most similar to that of (Galley et al., 2004).

5.2 Synchronous Tree Fragment Pair Extraction

The main concept underlying our syntactic rule extraction process is that we treat the node alignments discovered by the PFA algorithm (described in previous sections) as synchronous tree decomposition points. This reflects the fact that these nodes denote points in the synchronous parse trees where translation correspondences can be put together compositionally. Using the aligned nodes as decomposition points, we break apart the synchronous trees into collections of minimal synchronous tree fragments. Finally, the synchronous fragments are also converted into synchronous context-free rules. These are then collected into a database of synchronous rules.

The input to our rule extraction process consists of the parallel parse trees along with their node alignment information. The constituent nodes in the parallel trees that were aligned by the PFA node alignment algorithm are treated as tree decomposition points. At each such decomposition point, spliting the two parallel trees results in two partial trees or tree fragments. One synchronous pair consists of

the subtrees that are headed by the aligned nodes where the decomposition took place. Since the subtrees are rooted at aligned nodes, their yields are translation equivalents of each other. The other synchronous tree fragment pair consists of the remaining portions of the trees. The translation equivalence of the complete tree (or subtree) prior to decomposition implies that these tree fragments (which exclude the detached subtrees) also correspond to translation equivalents. The tree fragments that are obtained by decomposing the synchronous trees in this fashion are similar to the Synchronous Tree Insertion Grammar of (Shieber and Schabes, 1990).

We developed a tree traversal algorithm that decomposes parallel trees into all minimal tree fragments. Given two synchronous trees and their node alignment decomposition information, 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. These synchronous subtree fragments are minimal in the sense that they do not contain any internal aligned nodes. Another property of the synchronous subtree fragments is that their frontier nodes are either aligned nodes from the original tree or leaf nodes (corresponding to lexical items). Figure 3 shows some sample tree fragment pairs that were obtained from the example discussed earlier in Figure 1.

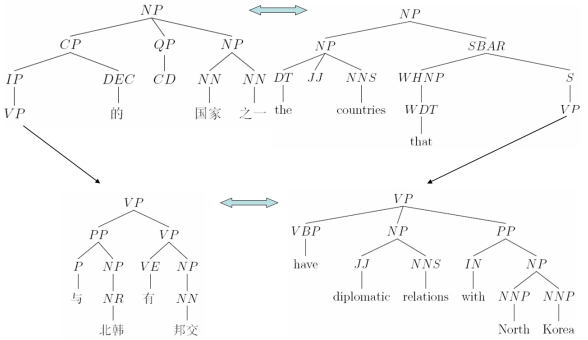

Figure 3: Tree Fragment Pairs Extracted from Aligned Nodes

5.3 Synchronous Transfer Rule Creation

In the last step, we convert the synchronous tree fragment pairs obtained as described above into synchronous context-free rules. This creates rules in a format that is compatible with the Stat-XFER formalism that was described in Section 1. Our system currently does not use the internal tree structure information that is contained in the synchronous tree fragments. Therefore, only the syntactic category labels of the roots of the tree fragments, and the nodes on the fragment frontier are relevant to decoding. This in essense corresponds to a "flattening" of the synchronous tree fragment into a synchronous context free style rule.

The flattening of the tree fragments is accomplished by an "in-order" traversal on each of the tree fragments to produce a string representation. Frontier nodes in the fragment are either labeled constituent nodes or leaf nodes of the original parse tree. These form the right-hand sides of the flattened rule. The positions of the constituent nodes in the output string are numbered to keep track of alignment of the nodes, which is often non-monotonic due to reordering between the source and target languages. Finally the root constituent label of the source tree fragment becomes the source-side parent category of the rule, while the root label of the target tree fragment becomes the target side parent category.

Accurate automatic transfer rule learning requires accurate word alignments and parse structures. Thus, to favor high precision (at the expense of some loss of recall), in our work to date on Chinese and other languages, while we extract syntactic phrases from all available parallel data, we extract

rules only from manually word-aligned parsed parallel data. To compensate for the limited amount of data, we generalize the rules as much as possible. Elements in the rules that originate from leaf nodes in the parse trees are generalized to their part-of-speech categories, if the corresponding words were one-to-one aligned in the parallel sentences. Unaligned words and words that are part of one-to-many alignments are not generalized to the POS level and remain lexicalized in the final rule.

The phrase table extracted from the corpus and the rules are scored together to ensure that they are consistent when used in our translation system. For all Stat-XFER experiments to date, we have used just the source side conditionig with a constant smoothing factor for robustness to noise.

6 Extraction Applied to Chinese-English Parallel Data

We used the pipeline of PFA node alignment followed by rule extraction to build resources for a Stat-XFER Chinese-to-English MT system. The syntax-based phrase table was constructed from two large parallel corpora released by LDC for the DARPA/GALE program. The parallel sentences for both English and Chinese were parsed using the Stanford parser. The first corpus consists of about 1.2 million sentence pairs. Our extraction process applied to this corpus resulted in a syntax-based phrase table of about 9.2 million entries. The other data source used was a parallel corpus of about 2.6 million sentences, but many of its entries were from a Chinese-English lexicon. From this corpus, we extracted 8.75 million phrases.

Rule learning was performed on a 10K-sentence parallel corpus that was manually word-aligned, released by LDC for the DARPA/GALE program. This manually word-aligned corpus includes the parallel Chinese-English treebank of 3,343 sentence pairs. The treebank sentences come with verified correct parse trees for English and Chinese. The rest of the 10K corpus was parsed by the Stanford parser. The complete 10K parallel corpus was node aligned and rules were extracted as described in Section 5. Figure 3 shows two synchronous tree fragments that were extracted from the example node-aligned sentence pair in Figure 1. After generalization and flattening, we obtain rules such as those shown in Figure 4. The above process resulted in a collection of almost 100K rules. Some statistics on this rule set are shown in Table 3. Analysis of this rule set indicates that only about 4% of these rules were observed more than once in the data. These include the most general and useful rules for mapping Chinese syntactic structures to their corresponding English structures. Most of the "singleton" rules are highly lexicalized. A large portion of the singleton rules are noisy rules, but many of them are good and useful rules. Experiments indicate that removing all singleton rules hurts translation performance.

```
VP::VP [北 NP VE NP] -> [ VBP NP with NP]
(
(X2::Y4) (X3::Y1) (X4::Y2)
)

NP::NP [VP 北 CD 有 邦交 ] -> [one of the CD countries that VP]
(
(X1::Y7)  (X3::Y4)
)
```

Figure 4: Rules Extracted from Tree Pairs

Corpus	Size (sens)	Rules with Structure	Rules (count>=2)	Complete Lexical rules
Parallel Treebank (3K)	3,343	45,266	1,962	11,521
993 sentences	993	12,661	331	2,199
Parallel Treebank (7K)	6,541	41,998	1,756	16,081
Merged Corpus set	10,877	99,925	4,049	29,801

Table 3: Statistics for Chinese-English Rules

7 Conclusions

The process described in this paper provides a fully automated solution for extracting large collection of reliable syntax-based phrase tables and syntactic synchronous transfer rules from large volumes of parsed parallel corpora. In conjunction with the Stat-XFER syntax-based framework, this provides a fully automated solution for building syntax-based MT systems. The current performance of this approach still lags behind state-of-the-art phrase-based systems when trained on the same parallel data but is showing encouraging improvements. Furthermore, the resources extracted by our process can be used by various other syntax-based MT approaches.

References

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270, Morristown, NJ, USA. Association for Computational Linguistics.

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, *HLT-NAACL 2004: Main Proceedings*, pages 273–280, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *ACL '06: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 961–968, Morristown, NJ, USA. Association for Computational Linguistics.

Declan Groves, Mary Hearne, and Andy Way. 2004. Robust sub-sentential alignment of phrase-structure trees. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 1072, Morristown, NJ, USA. Association for Computational Linguistics.

M. Hearne and A. Way. 2003. Seeing the wood for the trees: Data-oriented translation.

Kenji Imamura, Hideo Okuma, Taro Watanabe, and Eiichiro Sumita. 2004. Example-based machine translation based on syntactic transfer with statistical models. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 99, Morristown, NJ, USA. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54, Morristown, NJ, USA. Association for Computational Linguistics.

Alon Lavie. 2008. A general search-based syntax-driven framework for machine translation. In *Invited paper in Proceedings of CICLing-2008*, pages 362–375. Computational Linguistics and Intelligent Text Processing, LNCS 4919,Springer.

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.

Arjen Poutsma. 2000. Data-oriented translation. In *Proceedings of the 18th conference on Computational linguistics*, pages 635–641, Morristown, NJ, USA. Association for Computational Linguistics.

Katharina Probst, Lori Levin, Erik Peterson, Alon Lavie, and Jaime Carbonell. 2002. Mt for minority languages usingelicitation-based learning of syntactic-transfer rules. *Machine Translation*, 17(4):245–270.

Yvonne Samuelsson and Martin Volk. 2007. Alignment tools for Parallel Treebanks. In *Proceedings of the GLDV Fruhjahrstagung*.

Stuart M. Shieber and Yves Schabes. 1990. Synchronous tree-adjoining grammars. In *Proceedings of the 13th Conference on Computational Linguistics*, pages 253–258, Morristown, NJ, USA. Association for Computational Linguistics.

John Tinsley, Mary Hearne, and Andy Way. 2007. Exploiting Parallel Treebanks to Improve Phrase-Based Statistical Machine Translation. In *Proceedings of the Sixth International Workshop on Treebanks and Linguistic Theories (TLT-07)*, pages 175–187, Bergen, Norway.

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

Fei Xia and Michael McCord. 2004. Improving a statistical machine translation system with automatically learned rewrite patterns. In *COLING '04: Proceedings of the 20th International Conference on Computational Linguistics*, page 508, Morristown, NJ, USA. Association for Computational Linguistics.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *ACL '01: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 523–530, Morristown, NJ, USA. Association for Computational Linguistics.