Dependencies vs. Constituents for Tree-Based Alignment

Daniel Gildea
Computer Science Department
University of Rochester
Rochester, NY 14627

Abstract

Given a parallel parsed corpus, statistical tree-to-tree alignment attempts to match nodes in the syntactic trees for a given sentence in two languages. We train a probabilistic tree transduction model on a large automatically parsed Chinese-English corpus, and evaluate results against human-annotated word level alignments. We find that a constituent-based model performs better than a similar probability model trained on the same trees converted to a dependency representation.

1 Introduction

Statistical approaches to machine translation, pioneered by Brown et al. (1990), estimate parameters for a probabilistic model of word-to-word correspondences and word re-orderings directly from large corpora of parallel bilingual text. In recent years, a number of syntactically motivated approaches to statistical machine translation have been proposed. These approaches assign a parallel tree structure to the two sides of each sentence pair, and model the translation process with reordering operations defined on the tree structure. The tree-based approach allows us to represent the fact that syntactic constituents tend to move as unit, as well as systematic differences in word order in the grammars of the two languages. Furthermore, the tree structure allows us to make probabilistic independence assumptions that result in polynomial time algorithms for estimating a translation model from parallel training data, and for finding the highest probability translation given a new sentence.

Wu (1997) modeled the reordering process with binary branching trees, where each production could be either in the same or in reverse order going from source to target language. The trees of Wu's Inversion Transduction Grammar were derived by synchronously parsing a parallel corpus, using a grammar with lexical translation probabilities at the leaves and a simple grammar with a single nonterminal providing the tree structure. While this grammar did not represent traditional syntactic categories such as verb phrases and noun phrases, it served to restrict the word-level alignments considered by the system to those allowable by reordering operations on binary trees.

Yamada and Knight (2001) present an algorithm for estimating probabilistic parameters for a similar model which represents translation as a sequence of re-ordering operations over children of nodes in a syntactic tree, using automatic parser output for the initial tree structures. This gives the translation model more information about the structure of the source language, and further constrains the reorderings to match not just a possible bracketing as in Wu (1997), but the specific bracketing of the parse tree provided.

Recent models of alignment have attempted to exploit syntactic information from both languages by aligning a pair of parse trees for the same sentence in either language node by node. Eisner (2003) presented such a system for transforming semantic-level dependecy trees into syntactic-level dependency trees for text generation. Gildea (2003) trained a system on parallel constituent trees from the Korean-English Treebank, evaluating agreement with hand-annotated word alignments. Ding and Palmer (2004) align parallel dependency trees with a divide and conquer strategy, choosing a highly likely word-pair as a splitting point in each tree. In addition to providing a deeper level of representation for the transformations of the translation model to work with, tree-to-tree models have the advantage that they are much less computationally costly to train than models which must induce tree structure on one or both sides of the translation pair. Because Expectation Maximization for tree-to-tree models iterates over pairs of nodes in the two trees, it is $O(n^2)$ in the sentence length, rather than $O(n^6)$ for Wu's Inversion Transduction Grammar or $O(n^4)$ for the Yamada and Knight tree-to-string model.

In this paper, we make a comparison of two tree-to-tree models, one trained on the trees produced by automatic parsers for both our English and Chinese corpora, and one trained on the same parser output converted to a dependency representation. The trees are converted using a set of deterministic head rules for each language. The dependency representation equalizes some differences in the annotation style between the English and Chinese treebanks. However, the dependency representation makes the assumption that not only the bracketing structure, but also the head word choices, will correspond in the two trees. Our evaluation is in terms of agreement with word-level alignments created by bilingual human annotators. Our model of alignment is that of Gildea (2003), reviewed in Section 2 and extended to dependency trees in Section 3. We describe our data and experiments in Section 4, and discuss results in Section 5.

2 The Tree-to-Tree Model

A tree-to-tree alignment model has tree transformation operations for reordering a node's children, inserting and deleting nodes, and translating individual words at the leaves of the parse trees. The transformed tree must not only match the surface string of the target language, but also the tree structure assigned to the string by the parser. In order to provide enough flexibility to make this possible, tree transformation operations allow a single node in the source tree to produce two nodes in the target tree, or two nodes in the source tree to be grouped together and produce a single node in the target tree. The model can be thought of as a synchronous tree substitution grammar, with probabilities parameterized to generate the target tree conditioned on the structure of the source tree.

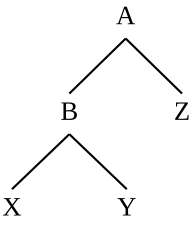
The probability $P(T_b|T_a)$ of transforming the source tree T_a into target tree T_b is modeled in a sequence of steps proceeding from the root of the target tree down. At each level of the tree:

1. At most one of the current node's children is grouped with the current node in a single *elementary tree*, with probability $P_{elem}(t_a|\varepsilon_a \Rightarrow children(\varepsilon_a))$, conditioned on the current node ε_a and its children (ie the CFG production expanding ε_a).

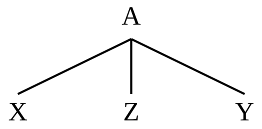
2. An alignment of the children of the current elementary tree is chosen, with probability $P_{align}(\alpha|\varepsilon_a \Rightarrow children(t_a))$. This alignment operation is similar to the re-order operation in the tree-to-string model, with the extension that 1) the alignment α can include insertions and deletions of individual children, as nodes in either the source or target may not correspond to anything on the other side, and 2) in the case where two nodes have been grouped into t_a, their children are re-ordered together in one step.

In the final step of the process, as in the tree-to-string model, lexical items at the leaves of the tree are translated into the target language according to a distribution $P_t(f|e)$.

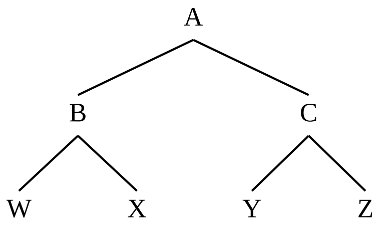
Allowing non-1-to-1 correspondences between nodes in the two trees is necessary to handle the fact that the depth of corresponding words in the two trees often differs. A further consequence of allowing elementary trees of size one or two is that some reorderings not allowed when reordering the children of each individual node separately are now possible. For example, with our simple tree



if nodes A and B are considered as one elementary tree, with probability $P_{elem}(t_a|\mathrm{A} \Rightarrow \mathrm{BZ})$, their collective children will be reordered with probability $P_{align}(\{(1,1)(2,3)(3,2)\}|\mathrm{A} \Rightarrow \mathrm{XYZ})$



giving the desired word ordering XZY. However, computational complexity as well as data sparsity prevent us from considering arbitrarily large elementary trees, and the number of nodes considered at once still limits the possible alignments. For example, with our maximum of two nodes, no transformation of the tree



is capable of generating the alignment WYXZ.

In order to generate the complete target tree, one more step is necessary to choose the structure on the target side, specifically whether the elementary tree has one or two nodes, what labels the nodes have, and, if there are two nodes, whether each child attaches to the first or the second. Because we are

Operation	Parameterization
elementary tree grouping	$P_{elem}(t_a \vert \varepsilon_a \Rightarrow children(\varepsilon_a))$
re-order	$P_{align}(\alpha \vert \varepsilon_a \Rightarrow children(t_a))$
insertion	α can include “insertion” symbol
lexical translation	$P_t(f \vert e)$
cloning	$P_{makeclone}(\varepsilon)$
	α can include “clone” symbol

Table 1: The probabilistic tree-to-tree model

ultimately interested in predicting the correct target string, regardless of its structure, we do not assign probabilities to these steps. The nonterminals on the target side are ignored entirely, and while the alignment algorithm considers possible pairs of nodes as elementary trees on the target side during training, the generative probability model should be thought of as only generating single nodes on the target side. Thus, the alignment algorithm is constrained by the bracketing on the target side, but does not generate the entire target tree structure.

While the probability model for tree transformation operates from the top of the tree down, probability estimation for aligning two trees takes place by iterating through pairs of nodes from each tree in bottom-up order, as sketched below:

```
for all nodes $\varepsilon_a$ in source tree $T_a$ in bottom-up order do
  for all elementary trees $t_a$ rooted in $\varepsilon_a$ do
    for all nodes $\varepsilon_b$ in target tree $T_b$ in bottom-up order do
      for all elementary trees $t_b$ rooted in $\varepsilon_b$ do
        for all alignments $\alpha$ of the children of $t_a$ and $t_b$ do
          $\beta(\varepsilon_a, \varepsilon_b)$ += $P_{elem}(t_a|\varepsilon_a) P_{align}(\alpha|\varepsilon_i) \prod_{(i,j)\in\alpha} \beta(\varepsilon_i, \varepsilon_j)$
        end for
      end for
    end for
  end for
end for
```

The outer two loops, iterating over nodes in each tree, require $O(|T|^2)$. Because we restrict our elementary trees to include at most one child of the root node on either side, choosing elementary trees for a node pair is $O(m^2)$, where m refers to the maximum number of children of a node. Computing the alignment between the $2m$ children of the elementary tree on either side requires choosing which subset of source nodes to delete, $O(2^{2m})$, which subset of target nodes to insert (or clone), $O(2^{2m})$, and how to reorder the remaining nodes from source to target tree, $O((2m)!)$. Thus overall complexity of the algorithm is $O(|T|^2 m^2 4^{2m} (2m)!)$, quadratic in the size of the input sentences, but exponential in the fan-out of the grammar.

2.1 Clone Operation

Both our constituent and dependency models make use of the “clone” operation introduced by Gildea (2003), which allows words to be aligned even in cases of radically mismatched trees, at a cost in the probability of the alignment. Allowing m-to-n matching of up to two nodes on either side of the parallel treebank allows for limited non-isomorphism between the trees. However, even given this flexibility, requiring alignments to match two input trees rather than one often makes tree-to-tree alignment more constrained than tree-to-string alignment. For example, even alignments with no change in word order may not be possible if the structures of the two trees are radically mismatched. Thus, it is helpful to allow departures from the constraints of the parallel bracketing, if it can be done in without dramatically increasing computational complexity.

The clone operation allows a copy of a node from the source tree to be made anywhere in the target tree. After the clone operation takes place, the transformation of source into target tree takes place using the tree decomposition and subtree alignment operations as before. The basic algorithm of the previous section remains unchanged, with the exception that the alignments α between children of two elementary trees can now include cloned, as well as inserted, nodes on the target side. Given that α specifies a new cloned node as a child of ε_j, the choice of which node to clone is made as in the tree-to-string model:

$$P_{clone}(\varepsilon_i | \text{clone} \in \alpha) = \frac{P_{makeclone}(\varepsilon_i)}{\sum_k P_{makeclone}(\varepsilon_k)}$$

Because a node from the source tree is cloned with equal probability regardless of whether it has already been “used” or not, the probability of a clone operation can be computed under the same dynamic programming assumptions as the basic tree-to-tree model. As with the tree-to-string cloning operation, this independence assumption is essential to keep the complexity polynomial in the size of the input sentences.

3 Dependency Tree-to-Tree Alignments

Dependencies were found to be more consistent than constituent structure between French and English by Fox (2002), though this study used a tree representation on the English side only. We wish to investigate whether dependency trees are also more suited to tree-to-tree alignment.

Figure 1 shows a typical Xinhua newswire sentence with the Chinese parser output, and the sentence's English translation with its parse tree. The conversion to dependency representation is shown below the original parse trees.

Examination of the trees shows both cases where the dependency representation is more similar across the two languages, as well as its potential pitfalls. The initial noun phrase, "14 Chinese open border cities" has two subphrases with a level of constituent structure (the QP and the lower NP) not found in the English parse. In this case, the difference in constituent structure derives primarily from differences in the annotation style between the original English and Chinese treebanks (Marcus et al., 1993; Xue and Xia, 2000; Levy and Manning, 2003). These differences disappear in the constituent representation. In general, the number of levels of constituent structure in a tree can be relatively arbitrary, while it is easier for people (whether professional syntacticians or not) to agree on the word-to-word dependencies.

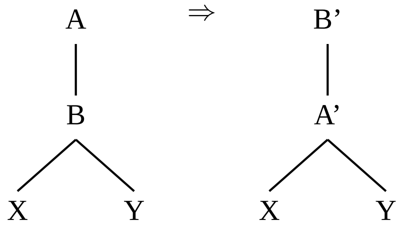
In some cases, differences in the number of level may be handled by the tree-to-tree model, for example by grouping the subject NP and its base NP child together as a single elementary tree. However, this introduces unnecessary variability into the alignment process. In cases with large difference in the depths of the two trees, the aligner may not be able to align the corresponding terminal nodes because only one merge is possible at each level. In this case the aligner will clone the subtree, at an even greater cost in probability.

The rest of our example sentence, however, shows cases where the conversion to dependency structure can in some cases exacerbate differences in constituent structure. For example, *jingji* and *jianshe* are sisters in the original constituent structure, as are their English translations, *economic* and *construction*. In the conversion to Chinese dependency structure, they remain sisters both dependent on the noun *chengjiu* (*achievements*) while in English, *economic* is a child of *construction*. The correspondence of a three-noun compound in Chinese to a noun modified by prepositional phrase and an adjective-noun relation in English means that the conversion rules select different heads even for pieces of tree that are locally similar.

3.1 The Dependency Alignment Model

While the basic tree-to-tree alignment algorithm is the same for dependency trees, a few modifications to the probability model are necessary.

First, the lexical translation operation takes place at each node in the tree, rather than only at the leaves. Lexical translation probabilities are maintained for each word pair as before, and the lexical translation probabilities are included in the alignment cost for each elementary tree. When both elementary trees contain two words, either alignment is possible between the two. The direct alignment between nodes within the elementary tree has probability $1 - P_{swap}$. A new parameter P_{swap} gives the probability of the upper node in the elementary tree in English corresponding to the lower node in Chinese, and vice versa. Thus, the probability for the following transformation:



is factored as $P_{elem}(\text{AB}|\text{A}{\Rightarrow}\text{B})$ P_{swap} $P_t(\text{A}'|\text{A})$ $P_t(\text{B}'|\text{B})$ $P_{align}(\{(1,1)(2,2)\}|\text{A} \Rightarrow \text{X}Y)$.

Our model does not represent the position of the head among its children. While this choice would have to be made in generating MT output, for the purposes of alignment we simply score how many tree nodes are correctly aligned, without flattening our trees into a string.

We further extended the tree-to-tree alignment algorithm by conditioning the reordering of a node's children on the node's lexical item as well as its syntactic category at the categories of its children. The lexicalized reordering probabilities were smoothed with the nonlexicalized probabilities (which are themselves smoothed with a uniform distribution). We smooth using a linear interpolation of lexicalized and unlexicalized probabilities, with weights proportional to the number of observations for each type of event.

4 Experiments

We trained our translation models on a parallel corpus of Chinese-English newswire text. We re-

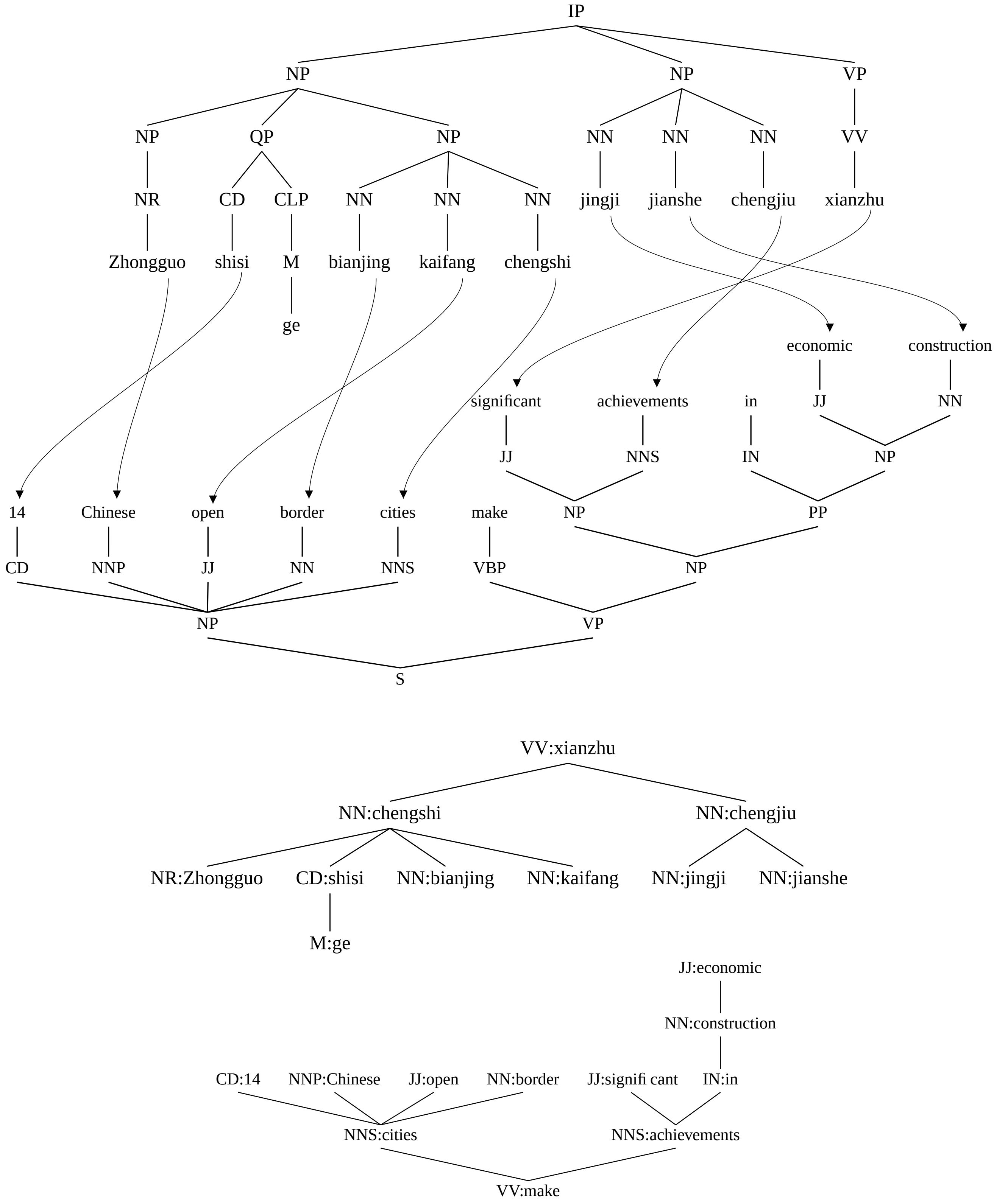


Figure 1: Constituent and dependency trees for a sample sentence

			Alignment
	Precision	*Recall*	*Error Rate*
IBM Model 1	.56	.42	.52
IBM Model 4	.67	.43	.47
Constituent Tree-to-Tree	.51	.48	.50
Dependency Tree-to-Tree	.44	.38	.60
Dependency, lexicalized reordering	.41	.37	.61

Table 2: Alignment results on Chinese-English corpus. Higher precision and recall correspond to lower alignment error rate.

stricted ourselves to sentences of no more than 25 words in either language, resulting in a training corpus of 18,773 sentence pairs with a total of 276,113 Chinese words and 315,415 English words. The Chinese data were automatically segmented into tokens, and English capitalization was retained. We replace words occurring only once with an unknown word token, resulting in a Chinese vocabulary of 23,783 words and an English vocabulary of 27,075 words. Chinese data was parsed using the parser of Bikel (2002), and English data was parsed using Collins (1999). Our hand-aligned test data were those used in Hwa et al. (2002), and consisted of 48 sentence pairs also with less than 25 words in either language, for a total of 788 English words and 580 Chinese words. The hand aligned data consisted of 745 individual aligned word pairs. Words could be aligned one-to-many in either direction. This limits the performance achievable by our models; the IBM models allow one-to-many alignments in one direction only, while the tree-based models allow only one-to-one alignment unless the cloning operation is used. A separate set of 49 hand-aligned sentence pairs was used to control overfitting in training our models.

We evaluate our translation models in terms of agreement with human-annotated word-level alignments between the sentence pairs. For scoring the viterbi alignments of each system against gold-standard annotated alignments, we use the alignment error rate (AER) of Och and Ney (2000), which measures agreement at the level of pairs of words:[1]

$$AER = 1 - \frac{2|A \cap G|}{|A| + |G|}$$

where A is the set of word pairs aligned by the automatic system, and G the set aligned in the gold standard. For a better understanding of how the models differ, we break this figure down into precision:

$$P = \frac{|A \cap G|}{|A|}$$

and recall:

$$R = \frac{|A \cap G|}{|G|}$$

Since none of the systems presented in this comparison make use of hand-aligned data, they may differ in the overall proportion of words that are aligned, rather than inserted or deleted. This affects the precision/recall tradeoff; better results with respect to human alignments may be possible by adjusting an overall insertion probability in order to optimize AER.

Table 2 provides a comparison of results using the tree-based models with the word-level IBM models. IBM Models 1 and 4 refer to Brown et al. (1993). We used the GIZA++ package, including the HMM model of Och and Ney (2000). We trained each model until AER began to increase on our held-out cross validation data, resulting in running Model 1 for three iterations, then the HMM model for three iterations, and finally Model 4 for two iterations (the optimal number of iterations for Models 2 and 3 was zero). “Constituent Tree-to-Tree” indicates the model of Section 2 trained and tested directly on the trees output by the parser, while “Dependency Tree-to-Tree” makes the modifications to the model described in Section 3. For reasons of computational efficiency, our constituent-based training procedure skipped sentences for which either tree had a node with more than five children, and the dependency-based training skipped trees with more than six children. Thus, the tree-based models were effectively trained on less data than IBM Model 4: 11,422 out of 18,773 sentence pairs for the constituent model and 10,662 sentence pairs for the dependency model. Our tree-based models were initialized with lexical translation probabilities trained using IBM Model 1, and uniform probabilities for the tree reordering operations. The models were trained until AER began to rise on our held-out

[1] While Och and Ney (2000) differentiate between *sure* and *possible* hand-annotated alignments, our gold standard alignments come in only one variety.

cross-validation data, though in practice AER was nearly constant for both tree-based models after the first iteration.

5 Discussion

The constituent-based version of the alignment model significantly outperforms the dependency-based model. The IBM models outperform the constituent tree-to-tree model to a lesser degree, with tree-to-tree achieving higher recall, and IBM higher precision. It is particularly significant that the tree-based model gets higher recall than the other models, since it is limited to one-to-one alignments unless the clone operation is used, bounding the recall it can achieve.

In order to better understand the differences between the constituent and dependency representations of our data, we analyzed how well the two representations match our hand annotated alignment data. We looked for *consistently aligned* pairs of constituents in the two parse trees. By consistently aligned, we mean that all words within the English constituent are aligned to words inside the Chinese constituent (if they are aligned to anything), and vice versa. In our example in Figure 1, the NP “14 Chinese border cities” and the Chinese subject NP “Zhongguo shisi ge bianjing kaifang chengshi” are consistenly aligned, but the PP “in economic construction” has no consistently aligned constituent in the Chinese sentence. We found that of the 2623 constituents in our English parse trees (not counting unary consituents, which have the same boundaries as their children), for 1044, or 40%, there exists some constituent in the Chinese parse tree that is consistently aligned. This confirms the results of Fox (2002) and Galley et al. (2004) that many translation operations must span more than one parse tree node. For each of our consistently aligned pairs, we then found the head word of both the Chinese and English constituents according to our head rules. The two head words correspond in the annotated alignments 67% of the time (700 out of 1044 consistently aligned constituent pairs). While the head-swapping operation of our translation model will be able to handle some cases of differing heads, it can only do so if the two heads are adjacent in both tree structures.

Our system is trained and test on automatically generated parse trees, which may contribute to the mismatches in the tree structures. As our test data was taken from the Chinese Treebank, hand-annotated parse trees were available for the Chinese, but not the English, sentences. Running the analysis on hand-annotated Chinese trees found slightly better English/Chinese agreement overall, but there were still disagreements in the head words choices for a third of all consistently aligned constuent pairs. Running our alignment system on gold standard trees did not improve results. The comparison between parser output and gold standard trees is summarized in Table 3.

We used head rules developed for statistical parsers in both languages, but other rules may be better suited to the alignment task. For example, the tensed auxiliary verb is considered the head of English progressive and perfect verb phrases, rather than the present or past particple of the main verb. Such auxiliaries carry agreement information relevant to parsing, but generally have no counterpart in Chinese. A semantically oriented dependency structure, such as Tree Adjoining Grammar derivation trees, may be more appropriate for alignment.

6 Conclusion

We present a comparison of constituent and dependency models for tree-to-tree alignment. Despite equalizing some mismatches in tree structure, the dependency representation does not perform as well, likely because it is less robust to large differences between the tree structures.

Acknowledgments We are very grateful to Rebecca Hwa, Hao Zhang, everyone at the 2003 John Hopkins speech and language summer research workshop, and EMNLP’s reviewers for their assistance, criticism, and data. This work was partially supported by NSF ITR IIS-09325646, NSF research infrastructure grant EIA-0080124, and NSF grant 0121285 to the summer workshop.

References

Daniel M. Bikel. 2002. Design of a multi-lingual, parallel-processing statistical parsing engine. In *Proceedings ARPA Workshop on Human Language Technology*.

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, June.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Michael John Collins. 1999. *Head-driven Statisti-*

	Chinese Parse Trees	
	Automatic	*Treebank*
Proportion of English constits w/ consistently aligned Chinese constit	.40	.42
Proportion of above with heads words aligned	.67	.66
Constituent-Based AER	.50	.51
Dependency-Based AER	.60	.62

Table 3: Comparison of automatically generated and hand-annotated Chinese parse trees.

cal Models for Natural Language Parsing. Ph.D. thesis, University of Pennsylvania, Philadelphia.

Yuan Ding and Martha Palmer. 2004. Automatic learning of parallel dependency treelet pairs. In *The First International Joint Conference on Natural Language Processing (IJCNLP)*.

Jason Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics, companion volume*, Sapporo, Japan.

Heidi J. Fox. 2002. Phrasal cohesion and statistical machine translation. In *In Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 304–311.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *Proceedings of the Human Language Technology Conference/North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*.

Daniel Gildea. 2003. Loosely tree-based alignment for machine translation. In *Proceedings of the 41th Annual Conference of the Association for Computational Linguistics (ACL-03)*, pages 80–87, Sapporo, Japan.

Rebecca Hwa, Philip Resnik, Amy Weinberg, and Okan Kolak. 2002. Evaluating translational correspondence using annotation projection. In *Proceedings of the 40th Annual Conference of the Association for Computational Linguistics (ACL-02)*.

Roger Levy and Christopher Manning. 2003. Is it harder to parse Chinese, or the Chinese Treebank? In *Proceedings of the 41th Annual Conference of the Association for Computational Linguistics (ACL-03)*, Sapporo, Japan.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2):313–330, June.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Conference of the Association for Computational Linguistics (ACL-00)*, pages 440–447, Hong Kong, October.

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

Nianwen Xue and Fei Xia. 2000. The bracketing guidelines for the penn chinese treebank. Technical Report IRCS-00-08, IRCS, University of Pennsylvania.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Conference of the Association for Computational Linguistics (ACL-01)*, Toulouse, France.