Tailoring Word Alignments to Syntactic Machine Translation

John DeNero
Computer Science Division
University of California, Berkeley
denero@berkeley.edu

Dan Klein
Computer Science Division
University of California, Berkeley
klein@cs.berkeley.edu

Abstract

Extracting tree transducer rules for syntactic MT systems can be hindered by word alignment errors that violate syntactic correspondences. We propose a novel model for unsupervised word alignment which explicitly takes into account target language constituent structure, while retaining the robustness and efficiency of the HMM alignment model. Our model's predictions improve the yield of a tree transducer extraction system, without sacrificing alignment quality. We also discuss the impact of various posterior-based methods of reconciling bidirectional alignments.

1 Introduction

Syntactic methods are an increasingly promising approach to statistical machine translation, being both algorithmically appealing (Melamed, 2004; Wu, 1997) and empirically successful (Chiang, 2005; Galley et al., 2006). However, despite recent progress, almost all syntactic MT systems, indeed statistical MT systems in general, build upon crude legacy models of word alignment. This dependence runs deep; for example, Galley et al. (2006) requires word alignments to project trees from the target language to the source, while Chiang (2005) requires alignments to induce grammar rules.

Word alignment models have not stood still in recent years. Unsupervised methods have seen substantial reductions in alignment error (Liang et al., 2006) as measured by the now much-maligned AER metric. A host of discriminative methods have been introduced (Taskar et al., 2005; Moore, 2005; Ayan and Dorr, 2006). However, few of these methods have explicitly addressed the tension between word alignments and the syntactic processes that employ them (Cherry and Lin, 2006; Daumé III and Marcu, 2005; Lopez and Resnik, 2005).

We are particularly motivated by systems like the one described in Galley et al. (2006), which constructs translations using tree-to-string transducer rules. These rules are extracted from a bitext annotated with both English (target side) parses and word alignments. Rules are extracted from target side constituents that can be projected onto contiguous spans of the source sentence via the word alignment. Constituents that project onto non-contiguous spans of the source sentence do not yield transducer rules themselves, and can only be incorporated into larger transducer rules. Thus, if the word alignment of a sentence pair does not respect the constituent structure of the target sentence, then the minimal translation units must span large tree fragments, which do not generalize well.

We present and evaluate an unsupervised word alignment model similar in character and computation to the HMM model (Ney and Vogel, 1996), but which incorporates a novel, syntax-aware distortion component which conditions on target language parse trees. These trees, while automatically generated and therefore imperfect, are nonetheless (1) a useful source of structural bias and (2) the same trees which constrain future stages of processing anyway. In our model, the trees do not *rule out* any alignments, but rather softly influence the probability of transitioning between alignment positions. In particular, transition probabilities condition upon paths through the target parse tree, allowing the model to prefer distortions which respect the tree structure.

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 17–24,
Prague, Czech Republic, June 2007.

Our model generates word alignments that better respect the parse trees upon which they are conditioned, without sacrificing alignment quality. Using the joint training technique of Liang et al. (2006) to initialize the model parameters, we achieve an AER superior to the GIZA++ implementation of IBM model 4 (Och and Ney, 2003) and a reduction of 56.3% in aligned interior nodes, a measure of agreement between alignments and parses. As a result, our alignments yield more rules, which better match those we would extract had we used manual alignments.

2 Translation with Tree Transducers

In a tree transducer system, as in phrase-based systems, the coverage and generality of the transducer inventory is strongly related to the effectiveness of the translation model (Galley et al., 2006). We will demonstrate that this coverage, in turn, is related to the degree to which initial word alignments respect syntactic correspondences.

2.1 Rule Extraction

Galley et al. (2004) proposes a method for extracting tree transducer rules from a parallel corpus. Given a source language sentence s, a target language parse tree t of its translation, and a word-level alignment, their algorithm identifies the constituents in t which map onto contiguous substrings of s via the alignment. The root nodes of such constituents – denoted *frontier nodes* – serve as the roots and leaves of tree fragments that form *minimal* transducer rules.

Frontier nodes are distinguished by their compatibility with the word alignment. For a constituent c of t, we consider the set of source words s_c that are aligned to c. If none of the source words in the linear closure s_c^* (the words between the leftmost and rightmost members of s_c) aligns to a target word outside of c, then the root of c is a frontier node. The remaining *interior* nodes do not generate rules, but can play a secondary role in a translation system.[1] The roots of null-aligned constituents are not frontier nodes, but can attach productively to multiple minimal rules.

Two transducer rules, $t_1 \rightarrow s_1$ and $t_2 \rightarrow s_2$, can be combined to form larger translation units by composing t_1 and t_2 at a shared frontier node and appropriately concatenating s_1 and s_2. However, no technique has yet been shown to robustly extract smaller component rules from a large transducer rule. Thus, for the purpose of maximizing the coverage of the extracted translation model, we prefer to extract many small, minimal rules and generate larger rules via composition. Maximizing the number of frontier nodes supports this goal, while inducing many aligned interior nodes hinders it.

2.2 Word Alignment Interactions

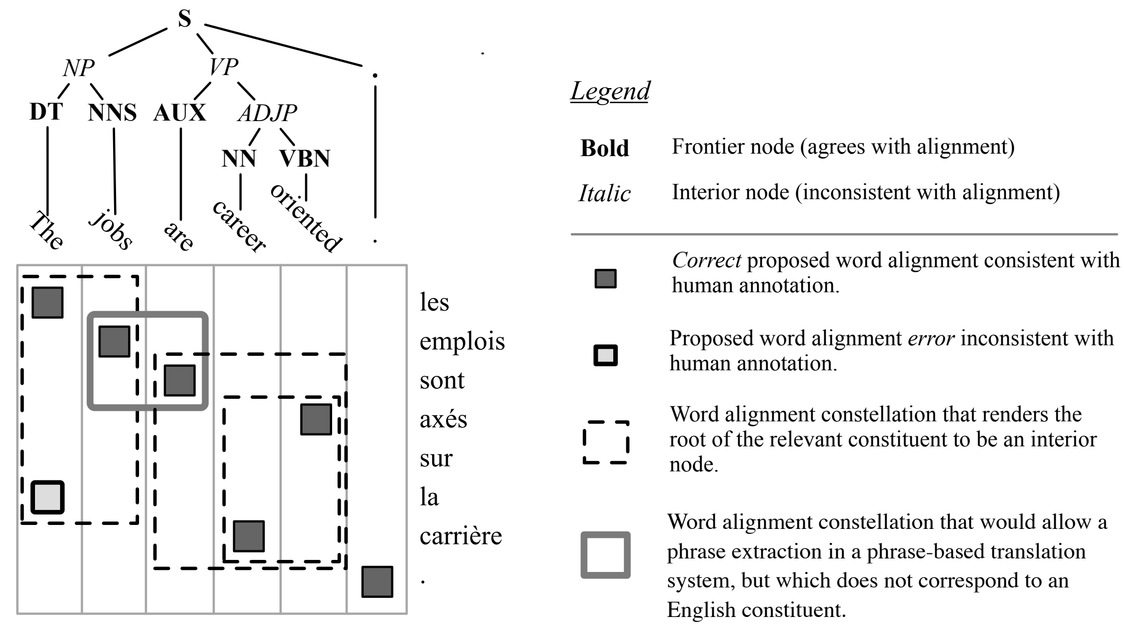
We now turn to the interaction between word alignments and the transducer extraction algorithm. Consider the example sentence in figure 1A, which demonstrates how a particular type of alignment error prevents the extraction of many useful transducer rules. The mistaken link [*la* $\Rightarrow$ *the*] intervenes between *axés* and *carrièr*, which both align within an English adjective phrase, while *la* aligns to a distant subspan of the English parse tree. In this way, the alignment violates the constituent structure of the English parse.

While alignment errors are undesirable in general, this error is particularly problematic for a syntax-based translation system. In a phrase-based system, this link would block extraction of the phrases [*axés sur la carrièr* $\Rightarrow$ *career oriented*] and [*les emplois* $\Rightarrow$ *the jobs*] because the error overlaps with both. However, the intervening phrase [*emplois sont* $\Rightarrow$ *jobs are*] would still be extracted, at least capturing the transfer of subject-verb agreement. By contrast, the tree transducer extraction method fails to extract any of these fragments: the alignment error causes all non-terminal nodes in the parse tree to be interior nodes, excluding preterminals and the root. Figure 1B exposes the consequences: a wide array of desired rules are lost during extraction.

The degree to which a word alignment respects the constituent structure of a parse tree can be quantified by the frequency of interior nodes, which indicate alignment patterns that cross constituent boundaries. To achieve maximum coverage of the translation model, we hope to infer tree-violating alignments only when syntactic structures truly diverge.

[1] Interior nodes can be used, for instance, in evaluating syntax-based language models. They also serve to differentiate transducer rules that have the same frontier nodes but different internal structure.

(A)



(B) (i)

(S NP[0] VP[1] .[2]) → [0] [1] [2]
(S (NP (DT[0] NNS[1]) VP[2] .[3]) → [0] [1] [2] [3]
(S (NP (DT[0] (NNS *jobs*)) VP[2] .[3]) → [0] *emplois* [2] [3]
(S (NP (DT[0] (NNS *jobs*)) (VP AUX[1] ADJV[2]) .[3]) → [0] *emplois* [1] [2] [3]
(S (NP (DT[0] (NNS *jobs*)) (VP (AUX *are*) ADJV[1]) .[2]) → [0] *emplois sont* [1] [2]

(ii)

(S (NP (DT[0] NNS[1]) (VP AUX[2] (ADJV NN[3] VBN[4]) .[5]) → [0] [1] [2] [3] [4] [5]
(S (NP (DT[0] (NNS *jobs*)) (VP AUX[1] (ADJV NN[2] VBN[3]) .[4]) → [0] *sont* [1] [2] [3] [4]
(S (NP (DT[0] (NNS *jobs*)) (VP (AUX *are*) (ADJV NN[1] VBN[2]) .[3]) → [0] *emplois sont* [1] [2] [3]

Figure 1: In this transducer extraction example, (A) shows a proposed alignment from our test set with an alignment error that violates the constituent structure of the English sentence. The resulting frontier nodes are printed in bold; all nodes would be frontier nodes under a correct alignment. (B) shows a small sample of the rules extracted under the proposed alignment, (ii), and the correct alignment, (i) and (ii). The single alignment error prevents the extraction of all rules in (i) and many more. This alignment pattern was observed in our test set and corrected by our model.

3 Unsupervised Word Alignment

To allow for this preference, we present a novel conditional alignment model of a foreign (source) sentence $\mathbf{f} = \{f_1, ..., f_J\}$ given an English (target) sentence $\mathbf{e} = \{e_1, ..., e_I\}$ and a target tree structure t. Like the classic IBM models (Brown et al., 1994), our model will introduce a latent alignment vector $\mathbf{a} = \{a_1, ..., a_J\}$ that specifies the position of an aligned target word for each source word. Formally, our model describes $p(\mathbf{a}, \mathbf{f}|\mathbf{e}, t)$, but otherwise borrows heavily from the HMM alignment model of Ney and Vogel (1996).

The HMM model captures the intuition that the alignment vector $\mathbf{a}$ will in general progress across the sentence $\mathbf{e}$ in a pattern which is mostly local, perhaps with a few large jumps. That is, alignments are locally monotonic more often than not.

Formally, the HMM model factors as:

$$p(\mathbf{a}, \mathbf{f}|\mathbf{e}) = \prod_{j=1}^{J} p_d(a_j|a_{j_-}, j) p_\ell(f_j|e_{a_j})$$

where j_- is the position of the last non-null-aligned source word before position j, p_ℓ is a lexical transfer model, and p_d is a local distortion model. As in all such models, the lexical component p_ℓ is a collection of unsmoothed multinomial distributions over foreign words.

The distortion model $p_d(a_j|a_{j_-}, j)$ is a distribution over the signed distance $a_j - a_{j_-}$, typically parameterized as a multinomial, Gaussian or exponential distribution. The implementation that serves as our baseline uses a multinomial distribution with separate parameters for $j = 1$, $j = J$ and shared parameters for all $1 < j < J$. Null alignments have fixed probability at any position. Inference over **a** requires only the standard forward-backward algorithm.

3.1 Syntax-Sensitive Distortion

The broad and robust success of the HMM alignment model underscores the utility of its assumptions: that word-level translations can be usefully modeled via first-degree Markov transitions and independent lexical productions. However, its distortion model considers only string distance, disregarding the constituent structure of the English sentence.

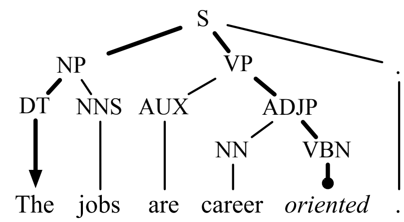
To allow syntax-sensitive distortion, we consider a new distortion model of the form $p_d(a_j|a_{j_-}, j, t)$. We condition on t via a generative process that transitions between two English positions by traversing the unique shortest path $\rho_{(a_{j_-}, a_j, t)}$ through t from a_{j_-} to a_j. We constrain ourselves to this shortest path using a staged generative process.

Stage 1 ($\textbf{P}\textsc{op}(\hat{n}), \textbf{S}\textsc{top}(\hat{n})$**):** Starting in the leaf node at a_{j_-}, we choose whether to STOP or POP from child to parent, conditioning on the type of the parent node $\hat{n}$. Upon choosing STOP, we transition to stage 2.

Stage 2 ($\textbf{M}\textsc{ove}(\hat{n}, d)$**):** Again, conditioning on the type of the parent $\hat{n}$ of the current node n, we choose a sibling $\bar{n}$ based on the signed distance $d = \phi_{\hat{n}}(n) - \phi_{\hat{n}}(\bar{n})$, where $\phi_{\hat{n}}(n)$ is the index of n in the child list of $\hat{n}$. Zero distance moves are disallowed. After exactly one MOVE, we transition to stage 3.

Stage 3 ($\textbf{P}\textsc{ush}(n, \phi_n(\breve{n}))$**):** Given the current node n, we select one of its children $\breve{n}$, conditioning on the type of n and the position of the child $\phi_n(\breve{n})$. We continue to PUSH until reaching a leaf.

This process is a first-degree Markov walk through the tree, conditioning on the current node and its immediate surroundings at each step. We enforce the property that $\rho_{(a_{j_-}, a_j, t)}$ be unique by staging the process and disallowing zero distance moves in stage 2. Figure 2 gives an example sequence of tree transitions for a small parse tree.



Stage 1: { Pop(VBN), Pop(ADJP), Pop(VP), Stop(S) }

Stage 2: { Move(S, -1) }

Stage 3: { Push(NP, 1), Push(DT, 1) }

Figure 2: An example sequence of staged tree transitions implied by the unique shortest path from the word *oriented* ($a_{j_-} = 5$) to the word *the* ($a_j = 1$).

The parameterization of this distortion model follows directly from its generative process. Given a path $\rho_{(a_{j_-}, a_j, t)}$ with $r = k + m + 3$ nodes including the two leaves, the nearest common ancestor, k intervening nodes on the ascent and m on the descent, we express it as a triple of staged tree transitions that include k POPs, a STOP, a MOVE, and m PUSHes:

$$\left(\begin{array}{c} \{\textsc{Pop}(n_2), ..., \textsc{Pop}(n_{k+1}), \textsc{Stop}(n_{k+2})\} \\ \{\textsc{Move}\,(n_{k+2}, \phi(n_{k+3}) - \phi(n_{k+1}))\} \\ \{\textsc{Push}\,(n_{k+3}, \phi(n_{k+4}))\,, ..., \textsc{Push}\,(n_{r-1}, \phi(n_r))\} \end{array} \right)$$

Next, we assign probabilities to each tree transition in each stage. In selecting these distributions, we aim to maintain the original HMM's sensitivity to target word order:

- Selecting POP or STOP is a simple Bernoulli distribution conditioned upon a node type.
- We model both MOVE and PUSH as multinomial distributions over the signed distance in positions (assuming a starting position of 0 for PUSH), echoing the parameterization popular in implementations of the HMM model.

This model reduces to the classic HMM distortion model given minimal English trees of only uniformly labeled pre-terminals and a root node. The classic 0-distance distortion would correspond to the

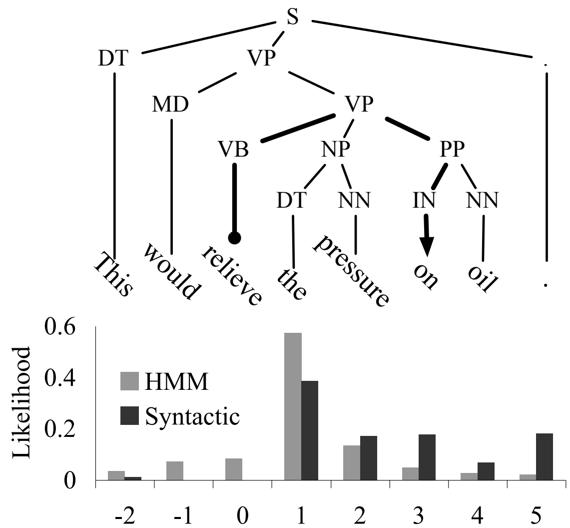


Figure 3: For this example sentence, the learned distortion distribution of $p_d(a_j|a_{j\_}, j, t)$ resembles its counterpart $p_d(a_j|a_{j\_}, j)$ of the HMM model but reflects the constituent structure of the English tree t. For instance, the short path from *relieve* to *on* gives a high transition likelihood.

STOP probability of the pre-terminal label; all other distances would correspond to MOVE probabilities conditioned on the root label, and the probability of transitioning to the terminal state would correspond to the POP probability of the root label.

As in a multinomial-distortion implementation of the classic HMM model, we must sometimes artificially normalize these distributions in the deficient case that certain jumps extend beyond the ends of the local rules. For this reason, MOVE and PUSH are actually parameterized by three values: a node type, a signed distance, and a range of options that dictates a normalization adjustment.

Once each tree transition generates a score, their product gives the probability of the entire path, and thereby the cost of the transition between string positions. Figure 3 shows an example learned distribution that reflects the structure of the given parse.

With these derivation steps in place, we must address a handful of special cases to complete the generative model. We require that the Markov walk from leaf to leaf of the English tree must start and end at the root, using the following assumptions.

1. Given no previous alignment, we forego stages 1 and 2 and begin with a series of PUSHes from the root of the tree to the desired leaf.

2. Given no subsequent alignments, we skip stages 2 and 3 after a series of POPs including a pop conditioned on the root node.

3. If the first choice in stage 1 is to STOP at the current leaf, then stage 2 and 3 are unnecessary. Hence, a choice to STOP immediately is a choice to emit another foreign word from the current English word.

4. We flatten unary transitions from the tree when computing distortion probabilities.

5. Null alignments are treated just as in the HMM model, incurring a fixed cost from any position.

This model can be simplified by removing all conditioning on node types. However, we found this variant to slightly underperform the full model described above. Intuitively, types carry information about cross-linguistic ordering preferences.

3.2 Training Approach

Because our model largely mirrors the generative process and structure of the original HMM model, we apply a nearly identical training procedure to fit the parameters to the training data via the Expectation-Maximization algorithm. Och and Ney (2003) gives a detailed exposition of the technique.

In the E-step, we employ the forward-backward algorithm and current parameters to find expected counts for each potential pair of links in each training pair. In this familiar dynamic programming approach, we must compute the distortion probabilities for each pair of English positions.

The minimal path between two leaves in a tree can be computed efficiently by first finding the path from the root to each leaf, then comparing those paths to find the nearest common ancestor and a path through it – requiring time linear in the height of the tree. Computing distortion costs independently for each pair of words in the sentence imposed a computational overhead of roughly 50% over the original HMM model. The bulk of this increase arises from the fact that distortion probabilities in this model must be computed for each unique tree, in contrast to the original HMM which has the same distortion probabilities for all sentences of a given length.

In the M-step, we re-estimate the parameters of the model using the expected counts collected during the E-step. All of the component distributions of our lexical and distortion models are multinomials. Thus, upon assuming these expectations as values for the hidden alignment vectors, we maximize likelihood of the training data simply by computing relative frequencies for each component multinomial. For the distortion model, an expected count $c(a_j, a_{j\_})$ is allocated to all tree transitions along the path $\rho_{(a_{j\_}, a_j, t)}$. These allocations are summed and normalized for each tree transition type to complete re-estimation. The method of re-estimating the lexical model remains unchanged.

Initialization of the lexical model affects performance dramatically. Using the simple but effective joint training technique of Liang et al. (2006), we initialized the model with lexical parameters from a jointly trained implementation of IBM Model 1.

3.3 Improved Posterior Inference

Liang et al. (2006) shows that thresholding the posterior probabilities of alignments improves AER relative to computing Viterbi alignments. That is, we choose a threshold τ (typically $\tau = 0.5$), and take

$$\mathbf{a} = \{(i, j) : p(a_j = i|\mathbf{f}, \mathbf{e}) > \tau\}.$$

Posterior thresholding provides computationally convenient ways to combine multiple alignments, and bidirectional combination often corrects for errors in individual directional alignment models. Liang et al. (2006) suggests a soft intersection of a model m with a reverse model r (foreign to English) that thresholds the product of their posteriors at each position:

$$\mathbf{a} = \{(i, j) : p_m(a_j = i|\mathbf{f}, \mathbf{e}) \cdot p_r(a_i = j|\mathbf{f}, \mathbf{e}) > \tau\}.$$

These intersected alignments can be quite sparse, boosting precision at the expense of recall. We explore a generalized version to this approach by varying the function c that combines p_m and p_r: $\mathbf{a} = \{(i, j) : c(p_m, p_r) > \tau\}$. If c is the $\max$ function, we recover the (hard) union of the forward and reverse posterior alignments. If c is the $\min$ function, we recover the (hard) intersection. A novel, high performing alternative is the soft union, which we evaluate in the next section:

$$c(p_m, p_r) = \frac{p_m(a_j = i|\mathbf{f}, \mathbf{e}) + p_r(a_i = j|\mathbf{f}, \mathbf{e})}{2}.$$

Syntax-alignment compatibility can be further promoted with a simple posterior decoding heuristic we call *competitive thresholding*. Given a threshold and a matrix c of combined weights for each possible link in an alignment, we include a link (i, j) only if its weight c_{ij} is above-threshold and it is connected to the maximum weighted link in both row i and column j. That is, only the maximum in each column and row and a contiguous enclosing span of above-threshold links are included in the alignment.

3.4 Related Work

This proposed model is not the first variant of the HMM model that incorporates syntax-based distortion. Lopez and Resnik (2005) considers a simpler tree distance distortion model. Daumé III and Marcu (2005) employs a syntax-aware distortion model for aligning summaries to documents, but condition upon the roots of the constituents that are jumped over during a transition, instead of those that are visited during a walk through the tree. In the case of syntactic machine translation, we want to condition on crossing constituent boundaries, even if no constituents are skipped in the process.

4 Experimental Results

To understand the behavior of this model, we computed the standard alignment error rate (AER) performance metric.[2] We also investigated extraction-specific metrics: the frequency of interior nodes – a measure of how often the alignments violate the constituent structure of English parses – and a variant of the CPER metric of Ayan and Dorr (2006).

We evaluated the performance of our model on both French-English and Chinese-English manually aligned data sets. For Chinese, we trained on the FBIS corpus and the LDC bilingual dictionary, then tested on 491 hand-aligned sentences from the 2002

[2]The hand-aligned test data has been annotated with both *sure* alignments S and *possible* alignments P, with $S \subseteq P$, according to the specifications described in Och and Ney (2003). With these alignments, we compute AER for a proposed alignment A as: $\left(1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|}\right) \times 100\%$.

French	Precision	Recall	AER
Classic HMM	93.9	**93.0**	6.5
Syntactic HMM	95.2	91.5	**6.4**
GIZA++	**96.0**	86.1	8.6
Chinese	Precision	Recall	AER
Classic HMM	81.6	78.8	**19.8**
Syntactic HMM	**82.2**	76.8	20.5
GIZA++*	61.9	**82.6**	29.7

Table 1: Alignment error rates (AER) for 100k training sentences. The evaluated alignments are a soft union for French and a hard union for Chinese, both using competitive thresholding decoding. *From Ayan and Dorr (2006), *grow-diag-final* heuristic.

French	w/o CT	with CT
Hard Intersection (Min)	8.4	8.4
Hard Union (Max)	12.3	7.7
Soft Intersection (Product)	6.9	7.1
Soft Union (Average)	**6.7**	**6.4**
Chinese	w/o CT	with CT
Hard Intersection (Min)	27.4	27.4
Hard Union (Max)	25.0	**20.5**
Soft Intersection (Product)	25.0	25.2
Soft Union (Average)	**21.1**	21.6

Table 2: Alignment error rates (AER) by decoding method for the syntactic HMM model. The competitive thresholding heuristic (CT) is particularly helpful for the hard union combination method.

NIST MT evaluation set. For French, we used the Hansards data from the NAACL 2003 Shared Task.[3] We trained on 100k sentences for each language.

4.1 Alignment Error Rate

We compared our model to the original HMM model, identical in implementation to our syntactic HMM model save the distortion component. Both models were initialized using the same jointly trained Model 1 parameters (5 iterations), then trained independently for 5 iterations. Both models were then combined with an independently trained HMM model in the opposite direction: $\mathbf{f} \rightarrow \mathbf{e}$.[4] Table 1 summarizes the results; the two models perform similarly. The main benefit of our model is the effect on rule extraction, discussed below.

We also compared our French results to the public baseline GIZA++ using the script published for the NAACL 2006 Machine Translation Workshop Shared Task.[5] Similarly, we compared our Chinese results to the GIZA++ results in Ayan and Dorr (2006). Our models substantially outperform GIZA++, confirming results in Liang et al. (2006).

Table 2 shows the effect on AER of competitive thresholding and different combination functions. The most dramatic effect of competitive thresholding is to improve alignment quality for hard unions. It also impacts rule extraction substantially.

4.2 Rule Extraction Results

While its competitive AER certainly speaks to the potential utility of our syntactic distortion model, we proposed the model for a different purpose: to minimize the particularly troubling alignment errors that cross constituent boundaries and violate the structure of English parse trees. We found that while the HMM and Syntactic models have very similar AER, they make substantially different errors.

To investigate the differences, we measured the degree to which each set of alignments violated the supplied parse trees, by counting the frequency of interior nodes that are not null aligned. Figure 4 summarizes the results of the experiment for French: the Syntactic distortion with competitive thresholding reduces tree violations substantially. Interior node frequency is reduced by 56% overall, with the most dramatic improvement observed for clausal constituents. We observed a similar 50% reduction for the Chinese data.

Additionally, we evaluated our model with the transducer analog to the consistent phrase error rate (CPER) metric of Ayan and Dorr (2006). This evaluation computes precision, recall, and F1 of the rules extracted under a proposed alignment, relative to the rules extracted under the gold-standard sure alignments. Table 3 shows improvements in F1 by using

[3]Following previous work, we developed our system on the 37 provided validation sentences and the first 100 sentences of the corpus test set. We used the remainder as a test set.

[4]Null emission probabilities were fixed to $\frac{1}{|\mathbf{e}|}$, inversely proportional to the length of the English sentence. The decoding threshold was held fixed at $\tau = 0.5$.

[5]Training includes 16 iterations of various IBM models and a fixed null emission probability of .01. The output of running GIZA++ in both directions was combined via intersection.

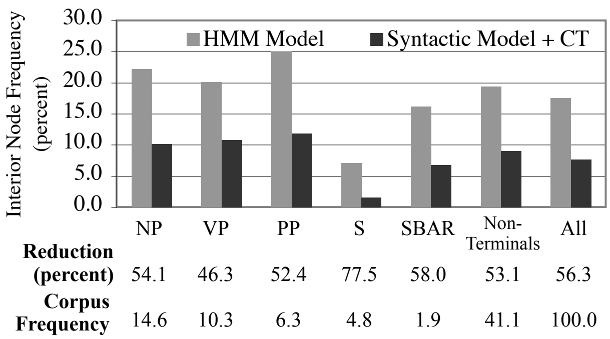


Figure 4: The syntactic distortion model with competitive thresholding decreases the frequency of interior nodes for each type and the whole corpus.

the syntactic HMM model and competitive thresholding together. Individually, each of these changes contributes substantially to this increase. Together, their benefits are partially, but not fully, additive.

5 Conclusion

In light of the need to reconcile word alignments with phrase structure trees for syntactic MT, we have proposed an HMM-like model whose distortion is sensitive to such trees. Our model substantially reduces the number of interior nodes in the aligned corpus and improves rule extraction while nearly retaining the speed and alignment accuracy of the HMM model. While it remains to be seen whether these improvements impact final translation accuracy, it is reasonable to hope that, all else equal, alignments which better respect syntactic correspondences will be superior for syntactic MT.

French	Prec.	Recall	F1
Classic HMM Baseline	40.9	17.6	24.6
Syntactic HMM + CT	33.9	22.4	27.0
Relative change	-17%	27%	10%

Chinese	Prec.	Recall	F1
HMM Baseline (hard)	**66.1**	14.5	23.7
HMM Baseline (soft)	36.7	39.1	37.8
Syntactic + CT (hard)	48.0	41.6	**44.6**
Syntactic + CT (soft)	32.9	**48.7**	39.2
*Relative change**	31%	6%	18%

Table 3: Relative to the classic HMM baseline, our syntactic distortion model with competitive thresholding improves the tradeoff between precision and recall of extracted transducer rules. Both French aligners were decoded using the best-performing soft union combiner. For Chinese, we show aligners under both soft and hard union combiners. *Denotes relative change from the second line to the third line.

References

Necip Fazil Ayan and Bonnie J. Dorr. 2006. Going beyond aer: An extensive analysis of word alignments and their impact on mt. In *ACL*.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1994. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19:263–311.

Colin Cherry and Dekang Lin. 2006. Soft syntactic constraints for word alignment through discriminative training. In *ACL*.

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

Hal Daumé III and Daniel Marcu. 2005. Induction of word and phrase alignments for automatic document summarization. *Computational Linguistics*, 31(4):505–530, December.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *HLT-NAACL*.

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*.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *HLT-NAACL*.

A. Lopez and P. Resnik. 2005. Improved hmm alignment models for languages with scarce resources. In *ACL WPT-05*.

I. Dan Melamed. 2004. Algorithms for syntax-aware statistical machine translation. In *Proceedings of the Conference on Theoretical and Methodological Issues in Machine Translation*.

Robert C. Moore. 2005. A discriminative framework for bilingual word alignment. In *EMNLP*.

Hermann Ney and Stephan Vogel. 1996. Hmm-based word alignment in statistical translation. In *COLING*.

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

Ben Taskar, Simon Lacoste-Julien, and Dan Klein. 2005. A discriminative matching approach to word alignment. In *EMNLP*.

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