Hidden Markov Tree Model for Word Alignment

Shuhei Kondo **Kevin Duh** **Yuji Matsumoto**
Graduate School of Information Science
Nara Institute of Science and Technology
8916-5, Takayama, Ikoma, Nara, 630-0192, Japan
{shuhei-k,kevinduh,matsu}@is.naist.jp

Abstract

We propose a novel unsupervised word alignment model based on the Hidden Markov Tree (HMT) model. Our model assumes that the alignment variables have a tree structure which is isomorphic to the target dependency tree and models the distortion probability based on the source dependency tree, thereby incorporating the syntactic structure from both sides of the parallel sentences. In English-Japanese word alignment experiments, our model outperformed an IBM Model 4 baseline by over 3 points alignment error rate. While our model was sensitive to posterior thresholds, it also showed a performance comparable to that of HMM alignment models.

1 Introduction

Automatic word alignment is the first step in the pipeline of statistical machine translation. Translation models are usually extracted from word-aligned bilingual corpora, and lexical translation probabilities based on word alignment models are also used for translation.

The most widely used models are the IBM Model 4 (Brown et al., 1993) and Hidden Markov Models (HMM) (Vogel et al., 1996). These models assume that alignments are largely monotonic, possibly with a few jumps. While such assumption might be adequate for alignment between similar languages, it does not necessarily hold between a pair of distant languages like English and Japanese.

Recently, several models have focused on incorporating syntactic structures into word alignment. As an extension to the HMM alignment, Lopez and Resnik (2005) present a distortion model conditioned on the source-side dependency tree, and DeNero and Klein (2007) propose a distortion model based on the path through the source-side phrase-structure tree. Some supervised models receive syntax trees as their input and use them to generate features and to guide the search (Riesa and Marcu, 2010; Riesa et al., 2011), and other models learn a joint model for parsing and word alignment from word-aligned parallel trees (Burkett et al., 2010). In the context of phrase-to-phrase alignment, Nakazawa and Kurohashi (2011) propose a Bayesian subtree alignment model trained with parallel sampling. None of these models, however, can incorporate syntactic structures from both sides of the language pair and can be trained computationally efficiently in an unsupervised manner at the same time.

The Hidden Markov Tree (HMT) model (Crouse et al., 1998) is one such model that satisfies the above-mentioned properties. The HMT model assumes a tree structure of the hidden variables, which fits well with the notion of word-to-word dependency, and it can be trained from unlabeled data via the EM algorithm with the same order of time complexity as HMMs.

In this paper, we propose a novel word alignment model based on the HMT model and show that it naturally enables unsupervised training based on both source and target dependency trees in a tractable manner. We also compare our HMT word alignment model with the IBM Model 4 and the HMM alignment models in terms of the standard alignment error rates on a publicly available English-Japanese dataset.

2 IBM Model 1 and HMM Alignment

We briefly review the IBM Model 1 (Brown et al., 1993) and the Hidden Markov Model (HMM) word alignment (Vogel et al., 1996) in this section. Both are probabilistic generative models that fac-

Proceedings of the Eighth Workshop on Statistical Machine Translation, pages 503–511,
Sofia, Bulgaria, August 8-9, 2013

tor as

$$p(\mathbf{f}|\mathbf{e}) = \sum_{\mathbf{a}} p(\mathbf{a}, \mathbf{f}|\mathbf{e})$$

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

where $\mathbf{e} = \{e_1, ..., e_I\}$ is an English (source) sentence and $\mathbf{f} = \{f_1, ..., f_J\}$ is a foreign (target) sentence. $\mathbf{a} = \{a_1, ..., a_J\}$ is an alignment vector such that $a_j = i$ indicates the j-th target word aligns to the i-th source word and $a_j = 0$ means the j-th target word is null-aligned. $j\_$ is the index of the last non null-aligned target word before the index j.

In both models, $p_t(f_j|e_{a_j})$ is the lexical translation probability and can be defined as conditional probability distributions. As for the distortion probability $p_d(a_j|a_{j\_})$, $p_d(a_j = 0|a_{j\_} = i') = p_0$ where p_0 is NULL probability in both models. $p_d(a_j = i|a_{j\_} = i')$ is uniform in the Model 1 and proportional to the relative count $c(i - i')$ in the HMM for $i \neq 0$. DeNero and Klein (2007) proposed a syntax-sensitive distortion model for the HMM alignment, in which the distortion probability depends on the path from the i-th word to the i'-th word on the source-side phrase-structure tree, instead of the linear distance between the two words.

These models can be trained efficiently using the EM algorithm. In practice, models in two directions (source to target and target to source) are trained and then symmetrized by taking their intersection, union or using other heuristics. Liang et al. (2006) proposed a joint objective of alignment models in both directions and the probability of agreement between them, and an EM-like algorithm for training.

They also proposed posterior thresholding for decoding and symmetrization, which take

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

with a threshold τ. DeNero and Klein (2007) summarized some criteria for posterior thresholding, which are

- Soft-Union

$$\sqrt{p_f(a_j = i|\mathbf{f}, \mathbf{e}) \cdot p_r(a_i = j|\mathbf{f}, \mathbf{e})}$$

- Soft-Intersection

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

- Hard-Union

$$max(p_f(a_j = i|\mathbf{f}, \mathbf{e}), p_r(a_i = j|\mathbf{f}, \mathbf{e}))$$

- Hard-Intersection

$$min(p_f(a_j = i|\mathbf{f}, \mathbf{e}), p_r(a_i = j|\mathbf{f}, \mathbf{e}))$$

where $p_f(a_j = i|\mathbf{f}, \mathbf{e})$ is the alignment probability under the source-to-target model and $p_r(a_i = j|\mathbf{f}, \mathbf{e})$ is the one under the target-to-source model.

They also propose a posterior decoding heuristic called *competitive thresholding*. Given a $j \times i$ matrix of combined weights c and a threshold τ, it choose a link (j, i) only if its weight $c_{ji} \geq \tau$ and it is connected to the link with the maximum weight both in row j and column i.

3 Hidden Markov Tree Model

The Hidden Markov Tree (HMT) model was first introduced by Crouse et al. (1998). Though it has been applied successfully to various applications such as image segmentation (Choi and Baraniuk, 2001), denoising (Portilla et al., 2003) and biology (Durand et al., 2005), it is largely unnoticed in the field of natural language processing. To the best of our knowledge, the only exception is Žabokrtskỳ and Popel (2009) who used a variant of the Viterbi algorithm for HMTs in the transfer phase of a deep-syntax based machine translation system.

An HMT model consists of an observed random tree $\mathbf{X} = \{x_1, ..., x_N\}$ and a hidden random tree $\mathbf{S} = \{s_1, ..., s_N\}$, which is isomorphic to the observed tree.

The parameters of the model are

- $P(s_1 = j)$, the initial hidden state prior
- $P(s_t = j|s_{\rho(t)} = i)$, transition probabilities
- $P(x_t = h|s_t = j)$, emission probabilities,

where $\rho()$ is a function that maps the index of a hidden node to the index of its parent node. These parameters can be trained via the EM algorithm.

The "upward-downward" algorithm proposed in Crouse et al. (1998), an HMT analogue of the forward-backward algorithm for HMMs, can be used in the E-step. However, it is based on the decomposition of joint probabilities and suffers from numerical underflow problems.

Durand et al. (2004) proposed a smoothed variant of the upward-downward algorithm, which is

based on the decomposition of smoothed probabilities and immune to underflow. In the next section, we will explain this variant in the context of word alignment.

4 Hidden Markov Tree Word Alignment

We present a novel word alignment model based on the HMT model. Given a target sentence $\mathbf{f} = \{f_1, ..., f_J\}$ with a dependency tree $\mathbf{F}$ and a source sentence $\mathbf{e} = \{e_1, ..., e_I\}$ with a dependency tree $\mathbf{E}$, an HMT word alignment model factors as

$$p(\mathbf{f}|\mathbf{e}) = \sum_{\mathbf{a}} p(\mathbf{a}, \mathbf{f}|\mathbf{e})$$

$$p(\mathbf{a}, \mathbf{f}|\mathbf{e}) = \prod_{j=1}^{J} p_d(a_j|a_{j\_}) p_t(f_j|e_{a_j}).$$

While these equations appear identical to the ones for the HMM alignment, they are different in that 1) $\mathbf{e}$, $\mathbf{f}$ and $\mathbf{a}$ are not chain-structured but tree-structured, and 2) $j\_$ is the index of the non null-aligned *lowest ancestor* of the j-th target word[1], rather than that of the last non null-aligned word preceding the j-th word as in the HMM alignment. Note that $\mathbf{A}$, the tree composed of alignment variables $\mathbf{a} = \{a_1, ..., a_J\}$, is isomorphic to the target dependency tree $\mathbf{F}$.

Figure 1 shows an example of a target dependency tree with an alignment tree, and a source dependency tree. Note that English is the target (or foreign) language and Japanese is the source (or English) language here. We introduce the following notations following Durand et al. (2004), slightly modified to better match the context of word alignment.

- $\rho(j)$ denotes the index of the head of the j-th target word.
- $c(j)$ denotes the set of indices of the dependents of the j-th target word.
- $\overline{\mathbf{F}}_j = \overline{\boldsymbol{f}}_j$ denotes the target dependency subtree rooted at the j-th word.

As for the parameters of the model, the initial hidden state prior described in Section 3 can be defined by assuming an artificial ROOT node for both dependency trees, forcing the target ROOT node to be aligned only to the source ROOT node and prohibiting other target nodes from being aligned to the source ROOT node. The lexical translation probability $p_t(f_j|e_{a_j})$, which corresponds to the emission probability, can be defined as conditional probability distributions just like in the IBM Model 1 and the HMM alignment.

The distortion probability $p_d(a_j = i|a_{j\_} = i')$, which corresponds to the transition probability, depends on the distance between the i-th source word and the i'-th source word on the source dependency tree $\mathbf{E}$, which we denote $d(i, i')$ hereafter. We model the dependence of $p_d(a_j = i|a_{j\_} = i')$ on $d(i, i')$ with the counts $c(d(i, i'))$.

In our model, $d(i, i')$ is represented by a pair of non-negative distances $(up, down)$, where up is the distance between the i-th word and the lowest common ancestor (*lca*) of the two words, $down$ is the one between the i'-th word and the *lca*. For example in Figure 1b, $d(0, 2) = (0, 4)$, $d(2, 5) = (2, 2)$ and $d(4, 7) = (3, 0)$. In practice, we clip the distance by a fixed window size w and store $c(d(i, i'))$ in a two-dimensional $(w + 1) \times (w + 1)$ matrix. When $w = 3$, for example, the distance $d(0, 2) = (0, 3)$ after clipping.

We can use the smoothed variant of upward-downward algorithm (Durand et al., 2004) for the E-step of the EM algorithm. We briefly explain the smoothed upward-downward algorithm in the context of tree-to-tree word alignment below. For the detailed derivation, see Durand et al. (2004).

In the smoothed upward-downward algorithm, we first compute the state marginal probabilities

$$p(a_j = i) = \sum_{i'} p(a_{\rho(j)} = i') p_d(a_j = i|a_{\rho(j)} = i')$$

for each target node and each state, where

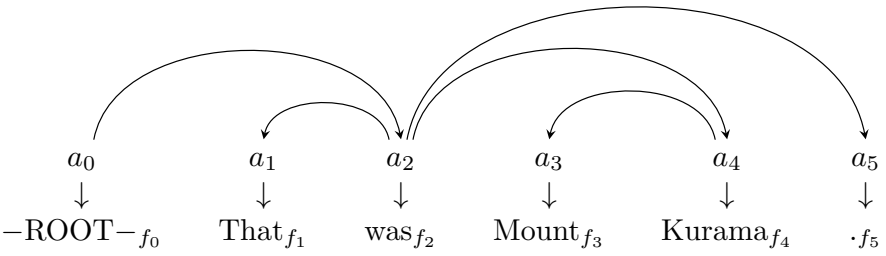
$$p_d(a_j = i|a_{\rho(j)} = i') = p_0$$

if the j-th word is null-aligned, and

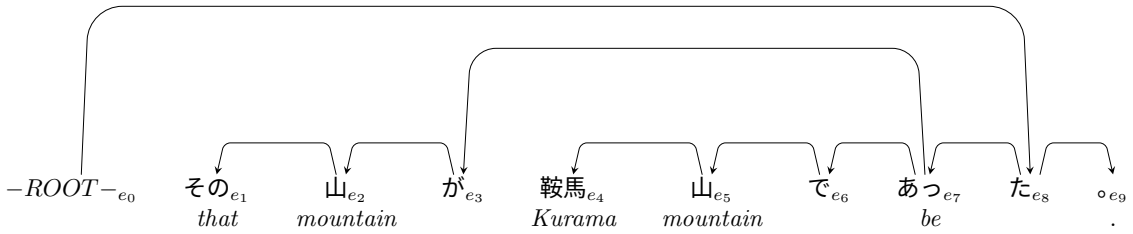
$$p_d(a_j = i|a_{\rho(j)} = i') = (1 - p_0) \cdot \frac{c(d(i', i))}{\sum_{i'' \neq 0} c(d(i', i''))}$$

if the j-th word is aligned. Note that we must artificially normalize $p_d(a_j = i|a_{\rho(j)} = i')$, because unlike in the case of the linear distance, multiple words can have the same distance from the j-th word on a dependency tree.

[1] This dependence on $a_{j\_}$ can be implemented as a first-order HMT, analogously to the case of the HMM alignment (Och and Ney, 2003).



(a) Target sentence with its dependency/alignment tree. Target words $\{f_0, ..., f_5\}$ are emitted from alignment variables $\{a_0, ..., a_5\}$. Ideally, $a_0 = 0, a_1 = 1, a_2 = 7, a_3 = 5, a_4 = 4$ and $a_5 = 9$.



(b) Source sentence with its dependency tree. None of the target words are aligned to e_2, e_3, e_6 and e_8.

Figure 1: An example of sentence pair under the Hidden Markov Tree word alignment model. If we ignore the source words to which no target words are aligned, the dependency structures look similar to each other.

In the next phase, the upward recursion, we compute $p(a_j = i|\overline{\mathbf{F}}_j = \overline{\boldsymbol{f}}_j)$ in a bottom-up manner. First, we initialize the upward recursion for each leaf by

$$\begin{aligned}\beta_j(i) &= p(a_j = i|F_j = f_j) \\ &= \frac{p_t(f_j|e_i)p(a_j = i)}{N_j},\end{aligned}$$

where

$$N_j = p(F_j = f_j) = \sum_i p_t(f_j|e_i)p(a_j = i).$$

Then, we proceed from the leaf to the root with the following recursion,

$$\begin{aligned}\beta_j(i) &= p(a_j = i|\overline{\mathbf{F}}_j = \overline{\boldsymbol{f}}_j) \\ &= \frac{\{\prod_{j' \in c(j)} \beta_{j,j'}(i)\} p_t(f_j|e_i)p(a_j = i)}{N_j},\end{aligned}$$

where

$$\begin{aligned}N_j &= \frac{p(\overline{\mathbf{F}}_j = \overline{\boldsymbol{f}}_j)}{\prod_{j' \in c(j)} p(\overline{\mathbf{F}}_{j'} = \overline{\boldsymbol{f}}_{j'})} \\ &= \sum_i \{ \prod_{j' \in c(j)} \beta_{j,j'}(i)\} p_t(f_j|e_i)p(a_j = i)\end{aligned}$$

and

$$\begin{aligned}\beta_{\rho(j),j}(i) &= \frac{p(\overline{\mathbf{F}}_j = \overline{\boldsymbol{f}}_j|a_{\rho(j)} = i)}{p(\overline{\mathbf{F}}_j = \overline{\boldsymbol{f}}_j)} \\ &= \sum_{i'} \frac{\beta_j(i')p_d(a_j = i'|a_{\rho(j)} = i)}{p(a_j = i')}.\end{aligned}$$

After the upward recursion is completed, we compute $p(a_j = i|\overline{\mathbf{F}}_0 = \overline{\boldsymbol{f}}_0)$ in the downward recursion. It is initialized at the root node by

$$\xi_0(i) = p(a_0 = i|\overline{\mathbf{F}}_0 = \overline{\boldsymbol{f}}_0).$$

Then we proceed in a top-down manner, computing

$$\begin{aligned}&\xi_j(i) = p(a_j = i|\overline{\mathbf{F}}_0 = \overline{\boldsymbol{f}}_0) \\ &= \frac{\beta_j(i)}{p(a_j = i)} \cdot \\ &\sum_{i'} \frac{p_d(a_j = i|a_{\rho(j)} = i')\xi_{\rho(j)}(i')}{\beta_{\rho(j),j}(i')}.\end{aligned}$$

for each node and each state.

The conditional probabilities

$$\begin{aligned}&p(a_j = i, a_{\rho(j)} = i'|\overline{\mathbf{F}}_0 = \overline{\boldsymbol{f}}_0) \\ &= \frac{\beta_j(i)p_d(a_j = i|a_{\rho(j)} = i')\xi_{\rho(j)}(i')}{p(a_j = i)\beta_{\rho(j),j}(i')},\end{aligned}$$

which is used for the estimation of distortion probabilities, can be extracted during the downward recursion.

In the M-step, the lexical translation model can be updated with

$$p_t(f|e) = \frac{c(f,e)}{c(e)},$$

just like the IBM Models and HMM alignments, where $c(f,e)$ and $c(e)$ are the count of the word pair (f,e) and the source word e. However, the update for the distortion model is a bit complicated, because the matrix that stores $c(d(i,i'))$ does not represent a probability distribution. To approximate the maximum likelihood estimation, we divide the counts $c(d(i,i'))$ calculated during the E-step by the number of distortions that have the distance $d(i,i')$ in the training data. Then we normalize the matrix by

$$c(d(i,i')) = \frac{c(d(i,i'))}{\sum_{i=0}^{\mathbf{w}} \sum_{i'=0}^{\mathbf{w}} c(d(i,i'))}.$$

Given initial parameters for the lexical translation model and the distortion counts, an HMT aligner collects the expected counts $c(f,e)$, $c(e)$ and $c(d(i,i'))$ with the upward-downward algorithm in the E-step and re-estimate the parameters in the M-Step. Dependency trees for the sentence pairs in the training data remain unchanged during the training procedure.

5 Experiment

We evaluate the performance of our HMT alignment model in terms of the standard alignment error rate[2] (AER) on a publicly available English-Japanese dataset, and compare it with the IBM Model 4 (Brown et al., 1993) and HMM alignment with distance-based (HMM) and syntax-based (S-HMM) distortion models (Vogel et al., 1996; Liang et al., 2006; DeNero and Klein, 2007).

We use the data from the Kyoto Free Translation Task (KFTT) version 1.3 (Neubig, 2011). Table 1 shows the corpus statistics. Note that these numbers are slightly different from the ones observed under the dataset's default training procedure because of the difference in the preprocessing scheme, which is explained below.

[2]Given sure alignments S and possible alignments P, the alignment error rate of alignments A is $1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|}$ (Och and Ney, 2003).

The tuning set of the KFTT has manual alignments. As the KFTT doesn't distinguish between sure and possible alignments, F-measure equals $1 - \mathrm{AER}$ on this dataset.

5.1 Preprocessing

We tokenize the English side of the data using the Stanford Tokenizer[3] and parse it with the Berkeley Parser[4] (Petrov et al., 2006). We use the phrase-structure trees for the Berkeley Aligner's syntactic distortion model, and convert them to dependency trees for our dependency-based distortion model[5]. As the Berkeley Parser couldn't parse 7 (out of about 330K) sentences in the training data, we removed those lines from both sides of the data. All the sentences in the other sets were parsed successfully.

For the Japanese side of the data, we first concatenate the function words in the tokenized sentences using a script[6] published by the author of the dataset. Then we re-segment and POS-tag them using MeCab[7] version 0.996 and parse them using CaboCha[8] version 0.66 (Kudo and Matsumoto, 2002), both with UniDic. Finally, we modify the CoNLL-format output of CaboCha where some kind of symbols such as punctuation marks and parentheses have dependent words. We chose this procedure for a reasonable compromise between the dataset's default tokenization and the dependency parser we use.

As we cannot use the default gold alignment due to the difference in preprocessing, we use a script[9] published by the author of the dataset to modify the gold alignment so that it better matches the new tokenization.

5.2 Training

We initialize our models in two directions with jointly trained IBM Model 1 parameters (5 iterations) and train them independently for 5 iterations

[3]`http://nlp.stanford.edu/software/`

[4]We use the model trained on the WSJ portion of Ontonotes (Hovy et al., 2006) with the default setting.

[5]We use Stanford's tool (de Marneffe et al., 2006) with options `-conllx -basic -makeCopulaHead -keepPunct` for conversion.

[6]`https://github.com/neubig/util-scripts/blob/master/combine-predicate.pl`

[7]`http://code.google.com/p/mecab/`

[8]`http://code.google.com/p/cabocha/`

[9]`https://github.com/neubig/util-scripts/blob/master/adjust-alignments.pl`

	Sentences	English Tokens	Japanese Tokens
Train	329,974	5,912,543	5,893,334
Dev	1,166	24,354	26,068
Tune	1,235	30,839	33,180
Test	1,160	26,730	27,693

Table 1: Corpus statistics of the KFTT.

	Precision	Recall	AER
HMT (Proposed)	71.77	55.23	37.58
IBM Model 4	60.58	57.71	40.89
HMM	69.59	56.15	37.85
S-HMM	71.60	56.14	37.07

Table 2: Alignment error rates (AER) based on each model's peak performance.

with window size $w = 4$ for the distortion model. The entire training procedure takes around 4 hours on a 3.3 GHz Xeon CPU.

We train the IBM Model 4 using GIZA++ (Och and Ney, 2003) with the training script of the Moses toolkit (Koehn et al., 2007).

The HMM and S-HMM alignment models are initialized with jointly trained IBM Model 1 parameters (5 iterations) and trained independently for 5 iterations using the Berkeley Aligner. We find that though initialization with jointly trained IBM Model 1 parameters is effective, joint training of HMM alignment models harms the performance on this dataset (results not shown).

5.3 Result

We use posterior thresholding for the HMT and HMM alignment models, and the *grow-diag-final-and* heuristic for the IBM Model 4.

Table 2 and Figure 2 show the result. As the Soft-Union criterion performed best, we don't show the results based on other criteria. On the other hand, as the peak performance of the HMT model is better with competitive thresholding and those of HMM models are better without it, we compare Precision/Recall curves and AER curves both between the same strategy and the best performing strategy for each model.

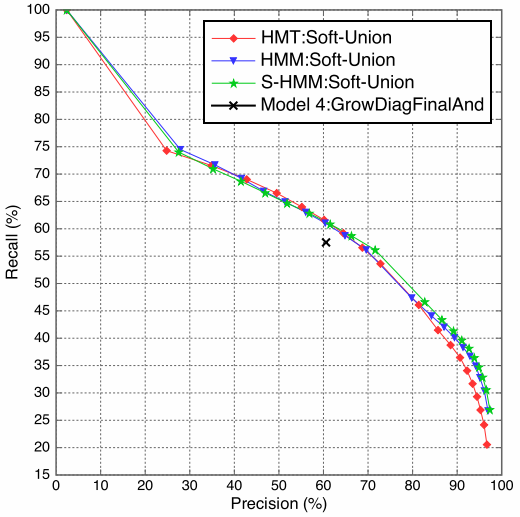
As shown in Table 2, the peak performance of the HMT alignment model is better than that of the IBM Model 4 by over 3 point AER, and it was somewhere between the HMM and the S-HMM. Taking into account that our distortion model is simpler than that of S-HMM, these results seem natural, and it would be reasonable to expect that replacing our distortion model with more sophisticated one might improve the performance.

When we look at Precision/Recall curves and AER curves in Figures 2a and 2d, the HMT model is performing slightly better in the range of 50 to 60 % precision and 0.15 to 0.35 posterior threshold with the Soft-Union strategy. Results in Figures 2b and 2e show that the HMT model performs better around the range around 60 to 70 precision and it corresponds to 0.2 to 0.4 posterior threshold with the competitive thresholding heuristic. In addition, results on both strategies show that performance curve of the HMT model is more peaked than those of HMM alignment models.

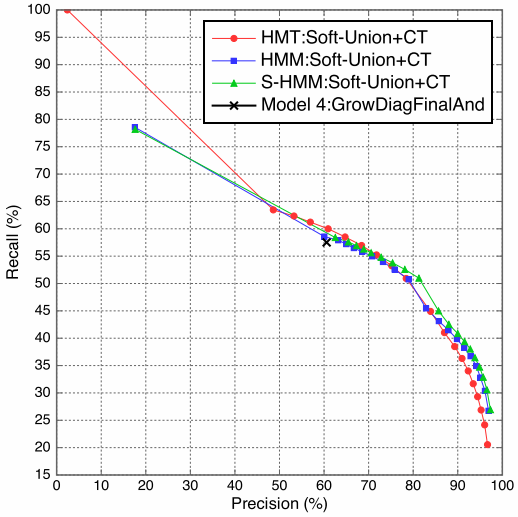
We suspect that a part of the reason behind such behavior can be attributed to the fact that the HMT model's distortion model is more uniform than that of HMM models. For example, in our model, all sibling nodes have the same distortion probability from their parent node. This is in contrast with the situation in HMM models, where nodes within a fixed distance have different distortion probabilities. With more uniform distortion probabilities, many links for a target word may have a considerable amount of posterior probability. If that is true, too many links will be above the threshold when it is set low, and too few links can exceed the threshold when it is set high. More sophisticated distortion model may help mitigate such sensitivity to the posterior threshold.

6 Related Works

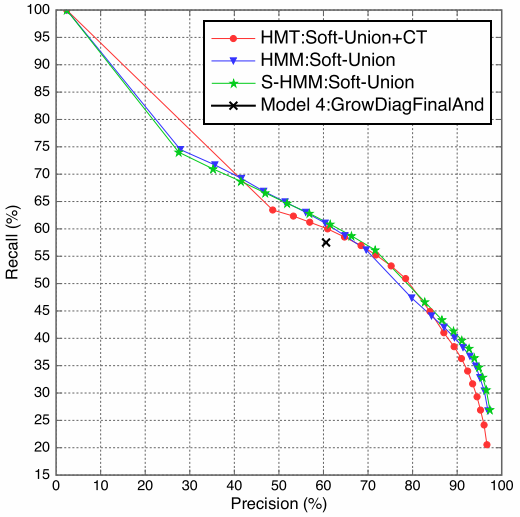
Lopez and Resnik (2005) consider an HMM model with distortions based on the distance in dependency trees, which is quite similar to our model's distance. DeNero and Klein (2007) propose another HMM model with syntax-based distortions based on the path through constituency trees, which improves translation rule extraction for tree-to-string transducers. Both models as-



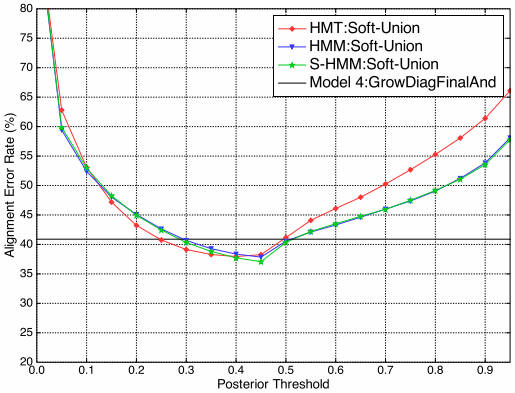
(a) Precision/Recall Curve with Soft-Union.



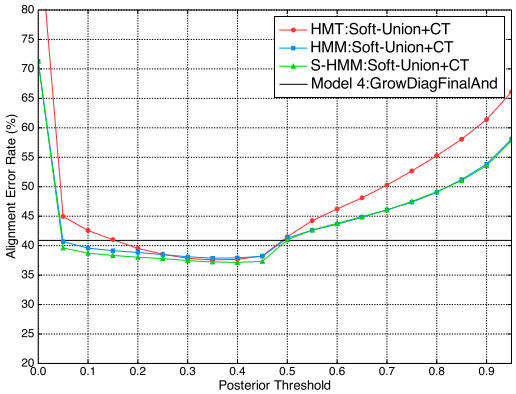
(b) Precision/Recall Curve with Soft-Union + Competitive Thresholding.



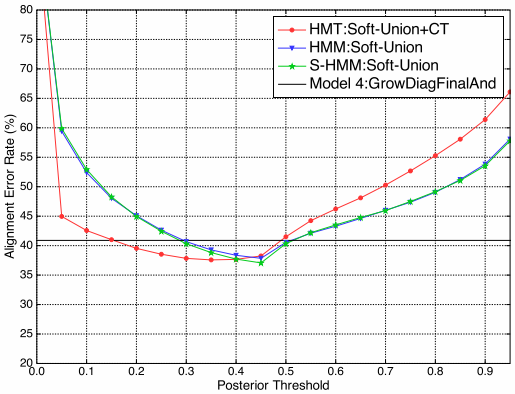
(c) Precision/Recall Curve with the Best Strategy.



(d) Alignment Error Rate with Soft-Union.



(e) Precision/Recall Curve with Soft-Union + Competitive Thresholding.



(f) Alignment Error Rate with with the Best Strategy.

Figure 2: Precision/Recall Curve and Alignment Error Rate with Different Models and Strategies.

sume a chain structure for hidden variables (alignment) as opposed to a tree structure as in our model, and condition distortions on the syntactic structure only in one direction.

Nakazawa and Kurohashi (2011) propose a dependency-based phrase-to-phrase alignment model with a sophisticated generative story, which leads to an increase in computational complexity and requires parallel sampling for training.

Several supervised, discriminative models use syntax structures to generate features and to guide the search (Burkett et al., 2010; Riesa and Marcu, 2010; Riesa et al., 2011). Such efforts are orthogonal to ours in the sense that discriminative alignment models generally use statistics obtained by unsupervised, generative models as features and can benefit from their improvement. It would be interesting to incorporate statistics of the HMT word alignment model into such discriminative models.

Žabokrtskỳ and Popel (2009) use HMT models for the transfer phase in a tree-based MT system. While our model assumes that the tree structure of alignment variables is isomorphic to target side's dependency tree, they assume that the deep-syntactic tree of the target side is isomorphic to that of the source side. The parameters of the HMT model is given and not learned by the model itself.

7 Conclusion

We have proposed a novel word alignment model based on the Hidden Markov Tree (HMT) model, which can incorporate the syntactic structures of both sides of the language into unsupervised word alignment in a tractable manner. Experiments on an English-Japanese dataset show that our model performs better than the IBM Model 4 and comparably to the HMM alignment models in terms of alignment error rates. It is also shown that the HMT model with a simple tree-based distortion is sensitive to posterior thresholds, perhaps due to the flat distortion probabilities.

As the next step, we plan to improve the distortion component of our HMT alignment model. Something similar to the syntax-sensitive distortion model of DeNero and Klein (2007) might be a good candidate.

It is also important to see the effect of our model on the downstream translation. Applying our model to recently proposed models that directly incorporate dependency structures, such as string-to-dependency (Shen et al., 2008) and dependency-to-string (Xie et al., 2011) models, would be especially interesting.

Last but not least, though the dependency structures don't pose a hard restriction on the alignment in our model, it is highly likely that parse errors have negative effects on the alignment accuracy. One way to estimate the effect of parse errors on the accuracy is to parse the input sentences with inferior models, for example trained on a limited amount of training data. Moreover, preserving some ambiguities using k-best trees or shared forests might help mitigate the effect of 1-best parse errors.

Acknowledgments

We thank anonymous reviewers for insightful suggestions and comments.

References

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational linguistics*, 19(2):263–311.

David Burkett, John Blitzer, and Dan Klein. 2010. Joint Parsing and Alignment with Weakly Synchronized Grammars. In *Proceedings of NAACL HLT 2010*, pages 127–135.

Hyeokho Choi and Richard G. Baraniuk. 2001. Multiscale Image Segmentation Using Wavelet-Domain Hidden Markov Models. *IEEE Transactions on Image Processing*, 10(9):1309–1321.

Matthew S. Crouse, Robert D. Nowak, and Richard G. Baraniuk. 1998. Wavelet-Based Statistical Signal Processing Using Hidden Markov Models. *IEEE Transactions on Signal Processing*, 46(4):886–902.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of LREC'06*, pages 449–454.

John DeNero and Dan Klein. 2007. Tailoring Word Alignments to Syntactic Machine Translation. In *Proceedings of ACL 2007*, pages 17–24.

Jean-Baptiste Durand, Paulo Gonçalvès, and Yann Guédon. 2004. Computational Methods for Hidden Markov Tree Models-An Application to Wavelet Trees. *IEEE Transactions on Signal Processing*, 52(9):2551–2560.

J.-B. Durand, Y. Guédon, Y. Caraglio, and E. Costes. 2005. Analysis of the plant architecture via tree-structured statistical models: the hidden Markov tree models. *New Phytologist*, 166(3):813–825.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% Solution. In *Proceedings of HLT-NAACL 2006, Short Papers*, pages 57–60.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of ACL 2007, Demonstration Session*, pages 177–180.

Taku Kudo and Yuji Matsumoto. 2002. Japanese Dependency Analysis using Cascaded Chunking. In *Proceedings of CoNLL-2002*, pages 63–69.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by Agreement. In *Proceedings of HLT-NAACL 2006*, pages 104–111.

Adam Lopez and Philip Resnik. 2005. Improved HMM Alignment Models for Languages with Scarce Resources. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 83–86.

Toshiaki Nakazawa and Sadao Kurohashi. 2011. Bayesian Subtree Alignment Model based on Dependency Trees. In *Proceedings of IJCNLP 2011*, pages 794–802.

Graham Neubig. 2011. The Kyoto Free Translation Task. http://www.phontron.com/kftt.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational linguistics*, 29(1):19–51.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning Accurate, Compact, and Interpretable Tree Annotation. In *Proceedings of COLING/ACL 2006*, pages 433–440.

Javier Portilla, Vasily Strela, Martin J. Wainwright, and Eero P. Simoncelli. 2003. Image Denoising Using Scale Mixtures of Gaussians in the Wavelet Domain. *IEEE Transactions on Image Processing*, 12(11):1338–1351.

Jason Riesa and Daniel Marcu. 2010. Hierarchical Search for Word Alignment. In *Proceedings of ACL 2010*, pages 157–166.

Jason Riesa, Ann Irvine, and Daniel Marcu. 2011. Feature-Rich Language-Independent Syntax-Based Alignment for Statistical Machine Translation. In *Proceedings of EMNLP 2011*, pages 497–507.

Libin Shen, Jinxi Xu, and Ralph Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of ACL-08: HLT*, pages 577–585.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-Based Word Alignment in Statistical Translation. In *Proceedings of COLING 1996*, pages 836–841.

Jun Xie, Haitao Mi, and Qun Liu. 2011. A Novel Dependency-to-String Model for Statistical Machine Translation. In *Proceedings of EMNLP 2011*, pages 216–226.

Zdeněk Žabokrtskỳ and Martin Popel. 2009. Hidden Markov Tree Model in Dependency-based Machine Translation. In *Proceedings of ACL-IJCNLP 2009, Short Papers*, pages 145–148.