# Using Word Dependent Transition Models in HMM based Word Alignment for Statistical Machine Translation

**Xiaodong He**

Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
xiaohe@microsoft.com

## Abstract

In this paper, we present a Bayesian Learning based method to train word dependent transition models for HMM based word alignment. We present word alignment results on the Canadian Hansards corpus as compared to the conventional HMM and IBM model 4. We show that this method gives consistent and significant alignment error rate (AER) reduction. We also conducted machine translation (MT) experiments on the Europarl corpus. MT results show that word alignment based on this method can be used in a phrase-based machine translation system to yield up to 1% absolute improvement in BLEU score, compared to a conventional HMM, and 0.8% compared to a IBM model 4 based word alignment.

## 1 Introduction

Word alignment is an important step of most modern approaches to statistical machine translation (Koehn et al., 2003). The classical approaches to word alignment are based on IBM models 1-5 (Brown et al., 1994) and the HMM based alignment model (Vogel et al., 1996) (Och and Ney, 2000a, 2000b), while recently discriminative approaches (Moore, 2006) and syntax based approaches (Zhang and Gildea, 2005) for word alignment are also studied. In this paper, we present improvements to the HMM based alignment model originally proposed by (Vogel et al., 1996, Och and Ney, 2000a).

Although HMM based word alignment approaches give good performance, one weakness of it is the coarse transition models. In the HMM based alignment model (Vogel et al., 1996), it is assumed that the HMM transition probabilities depend only on the jump width from the last state to the next state. Therefore, the knowledge of transition probabilities given a particular source word $e$ is not sufficiently modeled.

In order to improve transition models in the HMM based alignment, Och and Ney (2000a) extended the transition models to be word-class dependent. In that approach, words of the source language are first clustered into a number of word classes, and then a set of transition parameters is estimated for each word class. In (2002), Toutanova et al. modeled self-transition (i.e., jump width is zero) probability separately from other transition probabilities. A word dependent self-transition model $\mathbf{P}(\mathbf{stay}|e)$ is introduced to decide whether to stay at the current source word $e$ at the next step, or jump to a different word. It was also shown that with the assumption that a source word with fertility greater than one generates consecutive words in the target language, this probability approximates fertility modeling. Deng and Byrne in (2005) improved this idea. They proposed a word-to-phrase HMM in which a source word dependent phrase length model is used to model the approximate fertility, i.e., the length of consecutive target words generated by the source word. It provides more powerful modeling of approximate fertility than the single $\mathbf{P}(\mathbf{stay}|e)$ parameter.

However, these methods only model the probability of state occupancy rather than a full set of transition probabilities. Important knowledge of jumping from $e$ to another position, e.g., jumping



*Proceedings of the Second Workshop on Statistical Machine Translation*, pages 80–87,
Prague, June 2007. 

forward (monotonic alignment) or jumping backward (non-monotonic alignment), is not modeled.

In this paper, we present a method to further improve the transition models for HMM alignment model. For each source word *e*, we not only model its self-transition probability, but also the probability of jumping from word *e* to a different word. For this purpose, we estimate a full transition model for each source word.

A key problem for detailed word-dependent transition modeling is data sparsity. In (Toutanova et al., 2002), the word dependent self-transition probability **P**(**stay**|*e*) is interpolated with the global HMM self-transition probability to alleviate the data sparsity problem, where an interpolation weight is used for all words and that weight is tuned on a hold-out set. In the proposed word dependent transition model, because there are a large number of parameters to estimate, the data sparsity problem is even more severe. Moreover, since the sparsity of different words are very different, it is difficult to find a one-size-fits-all interpolation weight, and therefore simple linear interpolation is not optimal. In order to address this problem, we use Bayesian learning so that the transition model parameters are estimated by maximum *a posteriori* (MAP) training. With the help of the prior distribution of the model, the training is regularized and results in robust models.

In the next section we briefly review modeling of transition probabilities in a conventional HMM alignment model (Vogel et al., 1996, Och and Ney, 2000a). Then we describe the equations of MAP training for word dependent transition models. In section 5, we present word alignment results that show significant alignment error rate reductions compared to the baseline HMM and IBM model 4. We also conducted phrase-based machine translation experiments on the Europarl corpus, English – French track, and shown that the proposed method can lead to significant BLEU score improvement compared to the HMM and IBM model 4.

## 2 Baseline HMM alignment model

We briefly review the HMM based word alignment models (Vogel, 1996, Och and Ney, 2000a). Let's denote by $f_1^J = (f_1, ..., f_J)$ as the French sentence, $e_1^I = (e_1, ..., e_I)$ as the English sentence, and $a_1^J = (a_1, ..., a_J)$ as the alignment that specifies the position of the English word aligned to each French word. In the HMM based word alignment, a HMM is built at English side, i.e., each (*position*, *word*) pair, $(a_j, e_{a_j})$, is a HMM state, which emits the French word $f_j$. In order to mitigate the sparse data problem, it is assumed that the emission probability only depends on the English word, i.e., $p(f_j \mid a_j, e_{a_j}) = p(f_j \mid e_{a_j})$, and the transition probability only depends on the position of the last state and the length of the English sentence, i.e., $p(a_j \mid a_{j-1}, e_{a_{j-1}}, I) = p(a_j \mid a_{j-1}, I)$. Then, Vogel et al. (1996) give

$$p(f_1^J \mid e_1^I) = \sum_{a_1^J} \prod_{j=1}^{J} \left[ p(a_j \mid a_{j-1}, I) p(f_j \mid e_{a_j}) \right] \quad (1)$$

In the HMM of (Vogel et al., 1996), it is further assumed these transition probabilities $p(a_j = i \mid a_{j-1} = i', I)$ depend only on the jump width ($i$ - $i'$), i.e.,

$$p(i \mid i', I) = \frac{c(i - i')}{\sum_{l=1}^{I} c(l - i')} \quad (2)$$

Therefore, the transition probability $p(a_j \mid a_{j-1}, I)$ depends on $a_{j-1}$ but only through the distortion set $\{c(i - i')\}$.

In (Och and Ney, 2000a), the word *null* is introduced to generate the French words that don't align to any English words. If we denote by $j\_$ the position of the last French word before $j$ that aligns to a non-null English word, the transition probabilities $p(a_j = i \mid a_{j-1} = i', I)$ in (1) is computed as $p(a_j = i \mid a_{j\_} = i', I) = \tilde{p}(i \mid i', I)$, where

$$\tilde{p}(i \mid i', I) = \begin{cases} p_0 & \text{if } i = 0 \\ (1 - p_0) \cdot p(i \mid i', I) & \text{otherwise} \end{cases}$$

state $i$=0 denotes the state of a null word at the English side, and $p_0$ is the probability of jumping to state 0, which is estimated from hold-out data.

For convenience, we denote by $\Lambda = \left\{ p(i \mid i', I), p(f_j \mid e_i) \right\}$ the HMM parameter set.

In the training stage, $\Lambda$ are usually estimated through maximum likelihood (ML) training, i.e.,

$$\Lambda_{ML} = \arg\max_{\Lambda} p(f_1^J \mid e_1^I, \Lambda) \qquad (3)$$

and the efficient Expectation-Maximization algorithm can be used to optimize $\Lambda$ iteratively until convergence (Rabiner 1989).

For the interest of this paper, we elaborate transition parameter estimation with more details. These transition probabilities $\{p(i \mid i', I)\}$ is a multinomial distribution estimated according to (2), where at each iteration the distortion set $\{c(i - i')\}$ is the fractional count of transitions with jump width $d = i - i'$, i.e.,

$$c(d) = \sum_{j=1}^{J-1} \sum_{i=1}^{I} \Pr(a_j = i, a_{j+1} = i + d \mid f_1^J, e_1^I, \Lambda') \qquad (4)$$

where $\Lambda'$ is the model obtained from the immediate previous iteration and these terms in (4) can be efficiently computed by using the Forward-Backward algorithm (Rabiner 1989). In practice, we can bucket the distortion parameters $\{c(d)\}$ into a few buckets as implemented in (Liang et al., 2006). In our implementation, 15 buckets are used for $c(\leq -7)$, $c(-6)$, ... $c(0)$, ..., $c(\geq 7)$. The probability mass for transitions with jump width larger than 6 is uniformly divided. As suggested in (Liang et al., 2006), we also use two separate sets of distortion parameters for transitioning into the first state, and for transitioning out of the last state, respectively. Finally, we further smooth transition probabilities with a uniform distribution as described in (Och and Ney, 2000a),

$$p'(a_j \mid a_{j\_}, I) = \alpha \cdot \frac{1}{I} + (1 - \alpha) \cdot p(a_j \mid a_{j\_}, I).$$

After training, Viterbi decoding is used to find the best alignment sequence $\hat{a}_1^J$. i.e.,

$$\hat{a}_1^J = \arg\max_{a_1^J} \prod_{j=1}^{J} \left[ p(a_j \mid a_{j\_}, I) p(f_j \mid e_{a_j}) \right].$$

# 3 Word-dependent transition models in HMM based alignment model

As discussed in the previous sections, conventional transition models that only depend on source word positions are not accurate enough. There are only limited distortion parameters to model the transition between HMM states for all English words, and the knowledge of transition probabilities given a particular source word is not represented. In order to improve the transition model in HMM, we extend the transition probabilities to be word dependent so that the probability of jumping from state $a_{j\_}$ to $a_j$ not only depends on $a_{j\_}$, but also depends on the English word at position $a_{j\_}$. This gives

$$p(f_1^J \mid e_1^I) = \sum_{a_1^J} \prod_{j=1}^{J} \left[ p(a_j \mid a_{j\_}, e_{a_{j\_}}, I) p(f_j \mid e_{a_j}) \right].$$

Compared to (1), we need to estimate the transition parameter $p(a_j \mid a_{j\_}, e_{a_{j\_}}, I)$ which is $e_{a_{j\_}}$ dependent. Correspondingly, the HMM parameters we need to estimate are $\Lambda = \{p(i \mid i', e_{i'}, I), p(f_j \mid e_i)\}$, which provides a much richer set of free parameters to model transition probabilities.

# 4 Bayesian Learning for word-dependent transition models

## 4.1 Maximum *a posteriori* training

Using ML training, we can obtain the estimation formula for word dependent transition probabilities $\{p(i \mid i', e, I)\}$ similar as (2), i.e.,

$$p_{ML}(i \mid i', e, I) = \frac{c(i - i'; e)}{\sum_{l=1}^{I} c(l - i'; e)} \qquad (5)$$

where at each training iteration the word dependent distortion set $\{c(i - i'; e)\}$ is computed by

$$c(d; e) = \sum_{j=1}^{J-1} \sum_{i=1}^{I} \delta(e_{a_j} = e) \Pr(a_j = i, a_{j+1} = i + d \mid f_1^J, e_1^I, \Lambda') \qquad (6)$$

where $d = i - i'$ is the jump width, and $\delta(e_{a_j} = e)$ is the Kronecker delta function that equals one if $e_{a_j} = e$, and zero otherwise.

However, for many non-frequent words, the data samples for $c(d;e)$ is very limited and therefore may lead to a biased model that severely overfits to the sparse data. In order to address this issue, maximum *a posteriori* (MAP) framework is applied (Gauvain and Lee, 1994). In MAP training, an appropriate prior distribution is used to incorporate prior knowledge into the model parameter estimation,

$$\Lambda_{MAP} = \arg\max_{\Lambda} p(f_1^J \mid e_1^I, \Lambda) g(\Lambda \mid e_1^I) \quad (7)$$

where the prior distribution $g(\Lambda \mid e_1^I)$ characterizes the distribution of the model parameter set $\Lambda$ given the English sentence. The relation between ML and MAP estimation is through the Bayes' theorem where the posterior distribution $p(\Lambda \mid f_1^J, e_1^I) \propto p(f_1^J \mid e_1^I, \Lambda) g(\Lambda \mid e_1^I)$, and $p(f_1^J \mid e_1^I, \Lambda)$ is the likelihood function.

In transition model estimation, the transition model $\{p(i \mid i', e_{i'}, I)\}$ is a multinomial distribution. Its conjugate prior distribution is a Dirichlet distribution taking the following form (Bishop 2006),

$$g\left(p(i \mid i', e_{i'}, I) \mid e_1^I\right) \propto \prod_{i=1}^{I} p(i \mid i', e_{i'}, I)^{v_{i',i}-1} \quad (8)$$

where $\{v_{i',i}\}$ is the set of hyper-parameters of the prior distribution. Note that for mathematic tractability, $v_{i',i}$ needs to be greater than 1, which is usually the case in practice.

Substitute (8) into (7) and using EM algorithm, we can obtain the iterative MAP training formula for transition models (Gauvain and Lee, 1994)

$$p_{MAP}(i \mid i', e, I) = \frac{c(i-i'; e) + v_{i',i} - 1}{\sum_{l=1}^{I} c(l-i'; e) + \sum_{l=1}^{I} v_{i',l} - I} \quad (9)$$

### 4.2 Setting hyper-parameters for the prior distribution

In Bayesian learning, the hyper-parameter set $\{v_{i',i}\}$ of the prior distribution is assumed known based on a subjective knowledge about the model. In our method, we set the prior with word-independent transition probabilities.

$$v_{i',i} = \tau \cdot p(i \mid i', I) + 1 \quad (10)$$

where $\tau$ is a positive parameter that needs to tune on a hold-out data set. We will investigate the effect of $\tau$ with experimental results in later sections.

Substituting (10) into (9), the MAP based transition model training formula becomes

$$p_{MAP}(i \mid i', e, I) = \frac{c(i-i'; e) + \tau \cdot p(i \mid i', I)}{\sum_{l=1}^{I} c(l-i'; e) + \tau} \quad (11)$$

Note that for frequent words that have a large amount of data samples for $c(d;e)$, the sum of $\sum_{l=1,\ldots,I} c(l-i'; e)$ is large, so that $p_{MAP}(i \mid i', e, I)$ is dominated by the data distribution. For rare words that have low counts of $c(d;e)$, $p_{MAP}(i \mid i', e, I)$ will approach to the word independent model. On the other hand, for the same word, when a small $\tau$ is used, a weak prior is applied, and the transition probability is more dependent on the training data of that word. When $\tau$ becomes larger and larger, a stronger prior knowledge is applied, and the word dependent transition model will approach to the word-independent transition model. Therefore, we can vary the parameter $\tau$ to control the contribution of prior distribution in model training and tune the word alignment performance.

## 5 Experimental Results

### 5.1 Word alignment on the Canadian Hansards English-French corpus

We evaluated our word dependent transition models for HMM based word alignment on the English-French Hansards corpus. Only a subset of 500K sentence pairs was used in our experiments including 447 test sentence-pairs. Tests sentence-pairs were manually aligned and were marked with both *sure* and *possible* alignments (Och and Ney 2000a). Using this annotation, we report the word alignment performance in terms of alignment error rate (AER) as defined by Och and Ney (2000a):

$$AER = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|} \quad (12)$$

where $S$ denotes the set of *sure* gold alignments, $P$ denotes the set of *possible* gold alignments, $A$ denotes the set of alignments generated by the word alignment method under test.

We first trained the IBM model 1 and then a baseline HMM model as described in section 2 on the Hansards corpus. As the common practice, we initialized the translation probabilities of model 1 with uniform distribution over word pairs occur together in a same sentence pair. HMM was initialized with uniform transition probabilities and model 1 translation probabilities. Both model 1 and HMM were trained with 5 iterations. For the proposed word dependent transition model based HMM (WDHMM), we used the same settings as the HMM baseline except that the transition probability is computed according to (11). We also trained IBM model 4 using GIZA++ provided by Och and Ney (2000c), where 5 iterations of model 4 training was performed after 5 iterations of model 1 plus 5 iterations of HMM.

The effect of hyper-parameters in the prior distribution for WDHMM is shown in Figure 1. The horizontal dot line represents the AER given by the baseline HMM. The dash-line curve represents the AERs of WDHMM given different $\tau$'s. We vary the value of $\tau$ in the range from 0 to 1E5 and present that range in a log-scale in the figure. Since $\tau = 0$ is not a valid value in the log domain, we actually use the left-most point in the figure to represent the case of $\tau = 0$. From Fig. 1 it is shown that when $\tau$ is zero, we actually use the ML trained word-dependent transition model. Due to the sparse data problem, the model is poorly estimated and lead to a high AER. When increase $\tau$ to a larger value, a stronger prior is applied to give a more robust model. Then in a large range of $\tau \in [100, 2000]$, WDHMM outperforms baseline HMM significantly. When $\tau$ gets even larger, MAP model training becomes being over-dominated by the prior distribution, and that eventually results in a performance approaching to that of the baseline HMM. Fig. 1 only presents AER results that are calculated after combination of word alignments of both E→F and F→E directions based on a set of heuristics proposed by Och and Ney (2000b). We have observed the similar trend of AER change for the E→F and F→E alignment directions, respectively. However, due to the limit of the space, we didn't include them in this paper.

In table 1-3, we give a detailed comparison between baseline HMM, WDHMM (with $\tau = 1000$), and IBM model 4. Compared to the baseline HMM, the proposed WDHMM can reduce AER by more than 13%. It even outperforms IBM model 4 after two direction word alignment combination. Meanwhile we noticed that although IBM model 4 gives superior performance over the baseline HMM on both of the two alignment directions, its AER after combination is almost the same as that of the baseline HMM. We hypothesize that it may due to the modeling mechanism difference between HMM and model 4.

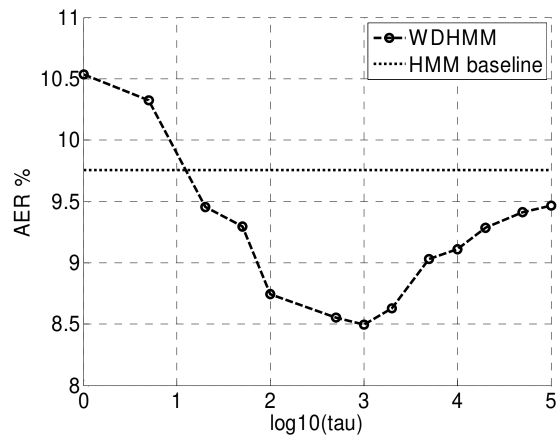


Figure 1: The AER of HMM baseline and the AER of WDHMM as the prior parameter $\tau$ is varied from 0 to 1E5. Note that the $x$ axis is in log scale and we use the left-most point in the figure to represent the case of $\tau = 0$. These results are calculated after combination of word alignments of both E→F and F→E directions.

| model | E → F | F → E | combined |
|---|---|---|---|
| baseline HMM | 12.7 | 13.7 | 9.8 |
| WDHMM ($\tau = 1000$) | 11.6 | 12.7 | 8.5 |
| IBM model 4 (GIZA++) | 11.3 | 12.1 | 9.7 |

Table 1: Comparison of test set **AER** between various models trained on 500K sentence pairs. All numbers are in percentage.

| model | E → F | F → E | combined |
|---|---|---|---|
| baseline HMM | 85.2 | 83.1 | 91.7 |
| WDHMM ($\tau = 1000$) | 86.1 | 83.8 | 93.3 |
| IBM model 4 (GIZA++) | 87.2 | 86.2 | 91.6 |

Table 2: Comparison of test set **Precision** between various models trained on 500K sentence pairs. All numbers are in percentage.

| model | E → F | F → E | combined |
|---|---|---|---|
| baseline HMM | 90.6 | 91.4 | 88.3 |
| WDHMM ($\tau = 1000$) | 91.9 | 92.6 | 89.1 |
| IBM model 4 (GIZA++) | 91.1 | 90.8 | 88.4 |

Table 3: Comparison of test set **Recall** between various models trained on 500K sentence pairs. All numbers are in percentage.

### 5.2 Machine translation on Europarl corpus

We further tested our WDHMM on a phrase-based machine translation system to see whether our improvement on word alignment can also improve MT accuracy measured by BLEU score (Papineni et al., 2002). The machine translation experiment was conducted on the English-to-French track of NAACL 2006 Europarl evaluation workshop. The supplied training corpus contains 688K sentence pairs. Text data are already tokenized. In our experiment, we first lower-cased all text, then word clustering was performed to cluster words of English and French into 32 word classes respectively using the tool provided by (J. Goodman). Then word alignment was performed. Both baseline HMM and IBM model 4 use word-class based transition models, and in WDHMM the word-class based transition model was used for prior distribution. The IBM model 4 is trained by GIZA++ with a regimen of 5 iterations of Model 1, 5 iterations of HMM, and 5 iterations of Model 4. Alignments of both directions are generated and then are combined by heuristic rules described in (Och and Ney 2000b). Then phrase table was extracted from the word aligned bilingual texts. The maximum phrase length was set to 7. In the phrase-based MT system, there are four channel models. They are direct maximum likelihood estimate of the probability of target phrase given source phrase, and the same estimate of source given target; we also compute the lexicon weighting features for source given target and target given source, respectively. Other models include word count and phrase count, and a 3-gram language model provided by the workshop. These models are combined in a log-linear framework with different weights (Och and Ney, 2002). The model weight vector is trained on a dev set with 2000 English sentences, each of which has one French translation reference. In the experiment, only the first 500 sentences were used to train the log-linear model weight vector, where minimum error rate (MER) training was used (Och, 2003). After MER training, the weight vector that gives the best accuracy on the development set was selected. We then applied it to tests. There are 2000 sentences in the development-test set *devtest*, 2000 sentences in a test set *test*, and 1064 out-of-domain sentences called *nc-test*. The Pharaoh phrase-based decoder (Koehn 2004b) was used for decoding. The maximum re-ordering limit for decoding was set to 7. We used default settings for all other parameters.

We present BLEU scores of MT systems using different word alignments on all three test sets, where Fig 2 shows BLEU scores of the two in-domain tests, and Fig 3 shows MT results on the out-of-domain test set. In testing, the prior parameter $\tau$ of WDHMM was varied in the range of [20, 5000].

In Fig. 2, the horizontal dash line and the horizontal dot line represent BLEU scores of the baseline HMM on *devtest* set and *test* set, respectively. The dash-line curve and dot-line curve represent the BLEU scores of WDHMM on these two tests. It is shown in the figure that WDHMM can achieve the best BLEU scores on both *devtest* and *test* when the prior parameter $\tau$ is set to 100. Furthermore, WDHMM also gives considerable improvement on BLEU score over the baseline HMM in a broad range of $\tau$ from 50 to 1000, which indicates that WDHMM works pretty stable within a reasonable range of prior distributions.

In Fig. 3, the horizontal dash line represents the BLEU score of baseline HMM on *nc-test* set and the dash-line curve represents BLEU scores of WDHMM on the out-of-domain test. The best BLEU is obtained at $\tau = 500$. It is interesting to see that the best $\tau$ for the out-of-domain test is larger than that of an in-domain test. One possible explanation is that for out-of-domain data, we need more robust modeling for outliers other than more accurate (in-domain) modeling. However, since the difference between $\tau = 500$ and $\tau = 100$ are very small, further experiments are needed before we can draw a conclusion.

We gives a detailed BLEU-wise comparison between baseline HMM and WDHMM in Table 4, where for WDHMM, $\tau$ =100 is used since it gives the best performance on the development-test set *devtest*. In the same table, we also provide BLEU results of using IBM model 4. Compared to baseline HMM alignment model, WDHMM can improve the BLEU score nearly 1% on in-domain test sets, and the improvement reduces to about 0.5% on the out-of-domain test. When compared to IBM model 4, WDHMM still gives higher BLEU scores, and outperform model 4 by about 0.8% on the *test* set. However the gain is reduced to 0.3% on *devtest* and 0.5% on the out-of-domain *nc-test*.

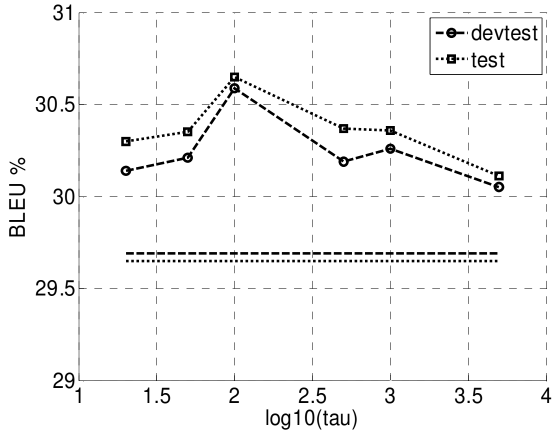


Figure 2: Machine translation results on Europarl, English to French track, devtest and test sets. The BLEU score of HMM baseline and the BLEU score of WDHMM as the prior parameter $\tau$ is varied from 20 to 5000. Note that the $x$ axis is in log scale.

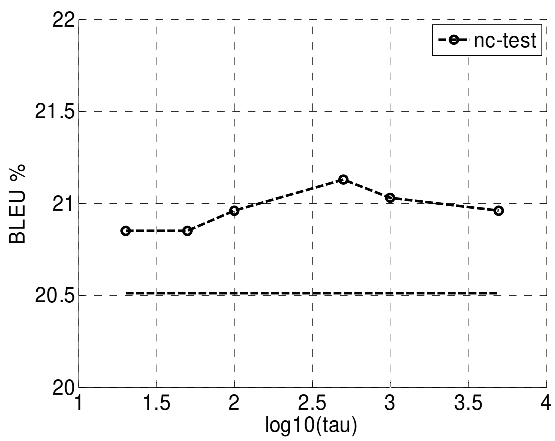


Figure 3: Machine translation results on Europarl, English to French track, out-of-domain test sets. The BLEU score of HMM baseline and the BLEU score of WDHMM as the prior parameter $\tau$ is varied from 20 to 5000. Note that the $x$ axis is in log scale.

| model | *devtest* | *test* | *nc-test* |
|---|---|---|---|
| baseline HMM | 29.69 | 29.65 | 20.51 |
| WDHMM ($\tau$ = 100) | 30.59 | 30.65 | 20.96 |
| IBM model 4 | 30.29 | 29.86 | 20.51 |

Table 4: Comparison of BLEU scores on devtest, test, and nc-test set between various word alignment models. All numbers are in percentage.

In order to verify whether these gains from WDHMM are statistically significant, we implemented paired bootstrap resampling method proposed by Koehn (2004b) to compute statistical significance of the above test results. In table 5, it is shown that BLEU gains of WDHMM over HMM and IBM-4 on different test sets, except the gain over IBM model 4 on the *devtest* set, are statistically significant with a significance level > 95%.

| significance level | *devtest* | *test* | *nc-test* |
|---|---|---|---|
| WDHMM ($\tau$=100) vs. HMM | 99.9% | 99.9% | 99.5% |
| WDHMM ($\tau$=100) vs. IBM model 4 | 93.7% | 99.9% | 99.3% |

Table 5: Statistical significance test of the BLEU improvement of WDHMM ($\tau$ = 100) vs. HMM baseline, and WDHMM ($\tau$ = 100) vs. IBM model 4 on devtest, test, and nc-test sets.

### 5.3 Runtime performance of WDHMM

WDHMM runs as fast as a normal HMM, and the extra memory needed for the word dependent transition model is proportional to the vocabulary size of the source language given that the distortion sets of $\{c(d;e)\}$ are bucketed. Runtime speed of WDHMM and IBM-model 4 using GIZA++ is tabulated in table 6. The results are based on Europarl English to French alignment and these tests were conducted on a fast PC with 3.0GHz CPU and 16GB memory. In Table 6, WDHMM includes 5 iterations of model 1 training followed by 5 iterations of WDHMM, while "IBM model 4" includes 5 iterations for model 1, 5 iterations for HMM, and 5 iterations for model 4. It is shown in Table 6 that WDHMM is more than four times faster to produce the end-to-end word alignment.

| model | runtime (min) |
|---|---|
| WDHMM | 121 |
| IBM model 4 | 537 |

Table 6: comparison of runtime performance bewteen WDHMM training and IBM model 4 training using GIZA++.

## 6 Discussion

Other works have been done to improve transition models in HMM based word alignment. Och and Ney (2000a) have suggested estimating word-class based transition models so as to provide more detailed transition probabilities. However, due to the sparse data problem, only a small number of word classes are usually used and the many words in the same class still have to share the same transition model. Toutanova et al. (2002) has proposed to estimate a word-dependent self-transition model **P**(**stay**|$e$) so that each word can have its own probability to decide whether to stay or jump to a different word. Later Deng and Byrne (2005) proposed a word dependent phrase length model to better model state occupancy. However, these model can only model the probability of self-jumping. Important knowledge of jumping from $e$ to a different position should also be word dependent but is not modeled.

Another interesting comparison is between WDHMM and the fertility-based models, e.g., IBM model 3-5. Compared to these models, a major disadvantage of HMM is the absence of a model of source word fertility. However, as discussed in (Toutanova et al. 2002),the word dependent self-transition model can be viewed as an approximation of fertility model. i.e., it models the number of consecutive target words generated by the source word with a geometric distribution. Therefore, with a well estimated word dependent transition model, this weakness of HMM is alleviated.

In this work, we proposed estimating a full word-dependent transition models in HMM based word alignment, and with Bayesian learning we can achieve robust model estimation under the sparse data condition. We have conducted a series of experiments to evaluate this method on word alignment and machine translation tests, and show significant improvement over baseline HMM in terms of AER and BLEU. It also performs better than the much more complicated IBM model 4 based word alignment model on various word alignment and machine translation tasks.

**Acknowledgments** The author is grateful to Chris Quirk and Arul Menezes for assistance with the MT system and for the valuable suggestions and discussions.

## References

C. M. Bishop, 2006. *Pattern Recognition and Machine Learning*. Springer.

P. Brown, S. D. Pietra, V. J. D. Pietra, and R. L. Mercer. 1994. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19:263–311.

Y. Deng and W. Byrne, 2005, HMM Word and Phrase Alignment For Statistical Machine Translation, in *Proceedings of HLT/EMNLP*.

J. Gauvain and C.-H. Lee, 1994, Maximum a Posteriori Estimation For Multivariate Gaussian Mixture Observations Of Markov Chains, *IEEE Trans on Speech and Audio Processing*.

J. Goodman, http://research.microsoft.com/~joshuago/

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of HLT-NAACL*.

P. Koehn, 2004a, Statistical Significance Tests for Machine Translation Evaluation, in *Proceedings of EMNLP*.

P. Koehn. 2004b. Pharaoh: A Beam Search Decoder For Phrase Based Statistical Machine Translation Models. In *Proceedings of AMTA*.

P. Liang, B. Taskar, and D. Klein, 2006, Alignment by Agreement, in *Proceedings of NAACL*.

R. Moore, W. Yih and A. Bode, 2006, Improved Discriminative Bilingual Word Alignment, In *Proceedings of COLING/ACL*.

F. J. Och and H. Ney. 2000a. A comparison of Alignment Models for Statistical Machine Translation. In *Proceedings of COLING*.

F. J. Och and H. Ney. 2000b. Improved Statistical Alignment Models. In *Proceedings of ACL*.

F. J. Och and H. Ney. 2000c. Giza++: Training of statistical translation models. http://www-i6.informatik.rwthaachen.de/och/software/GIZA++.html.

F. J. Och and H. Ney. 2002. Discriminative training and Maximum Entropy Models for Statistical Machine Translation, In *Proceedings of ACL*.

F. J. Och, 2003, Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of ACL*.

K. A. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 2002. Bleu: A Method For Automatic Evaluation Of Machine Translation. in *Proceedings of ACL*.

L. R. Rabiner, 1989 A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*.

K. Toutanova, H. T. Ilhan, and C. D. Manning. 2002. Extensions to HMM-based Statistical Word Alignment Models. In *Proceedings of EMNLP*.

S. Vogel, H. Ney, and C. Tillmann. 1996. HMM-based Word Alignment In Statistical Translation. In *Proceedings of COLING*.

H. Zhang and D. Gildea, 2005, Stochastic Lexicalized Inversion Transduction Grammar for Alignment, In *Proceedings of ACL*.