PAPER

Bayesian Word Alignment and Phrase Table Training for Statistical Machine Translation

Zezhong LI[†a)], *Member*, Hideto IKEDA[†], *Nonmember*, *and* Junichi FUKUMOTO[†], *Member*

SUMMARY In most phrase-based statistical machine translation (SMT) systems, the translation model relies on word alignment, which serves as a constraint for the subsequent building of a phrase table. Word alignment is usually inferred by GIZA++, which implements all the IBM models and HMM model in the framework of Expectation Maximum (EM). In this paper, we present a fully Bayesian inference for word alignment. Different from the EM approach, the Bayesian inference makes use of all possible parameter values rather than estimating a single parameter value, from which we expect a more robust inference. After inferring the word alignment, current SMT systems usually train the phrase table from Viterbi word alignment, which is prone to learn incorrect phrases due to the word alignment mistakes. To overcome this drawback, a new phrase extraction method is proposed based on multiple Gibbs samples from Bayesian inference for word alignment. Empirical results show promising improvements over baselines in alignment quality as well as the translation performance.

key words: *Bayesian inference, word alignment, phrase extraction, reordering, statistical machine translation*

1. Introduction

In phrase-based statistical machine translation, a source sentence is translated by the decoder concatenating translation options from an inventory called a phrase table, which is the key component that contributes to the success of the final translation performance. Approaches to building such a phrase table can be classified into two groups: the first approach is called two-staged approach, in which phrases are collected by fixing a word alignment and applying phrase extraction heuristics, and a typical representation of such an idea is described in [1]; the second approach is called direct phrase alignment, in which phrases can be learned directly [2]. A great deal of literature has compared the above two approaches. In theory, the direct phrase alignment is more theoretically sound and elegant over the heuristic-based two-staged approach, but it also has challenges, primarily including computing complexity that arises from the exponentially large number of decompositions of a bilingual sentence pair into phrase pairs and degenerate behavior (explaining the training corpus with one phrase pair per sentence). Currently the two-staged approach has already been more broadly adopted by phrase-based SMT systems, such as Moses [4] and Phrasal [5], and the reason might be attributed to the massive amounts of effort on word alignment and phrase extraction, which provides more reliable baselines for the translation performance. We also conjecture that phrase is a better translation granularity than word, but not necessarily means it is also an appropriate granularity for learning, which is determined by the property of data sparseness in natural language processing (henceforth NLP). Therefore, we will follow the two-staged approach.

In the first step of our two-staged approach, we propose a Bayesian inference for word alignment, which has been broadly applied to various unsupervised learning of latent linguistic structure [6]. Two theoretical factors contribute to its superiority. One factor is that taking advantage of all possible parameter values leads to greater robustness in decision, thus mitigating over-fitting manifested as “garbage collector” (the phenomenon that low-frequency words in the target language aligning to too many words in the source language) [7]. The other factor is that the use of priors can lead to sparse distributions, which is more consistent with the nature of natural language. Another practical advantage of Bayesian approach is that the implementation can be much easier [8], whereas GIZA++ is usually treated as a black box, which is hard to understand and to improve [9].

In the second step of our two-staged approach, we extract phrase pairs from all the generated alignment samples in the previous step. This is very similar with phrase extraction from N-best alignments, which can overcome some disadvantages of extraction from 1-best alignment, and has proven to be effective in improving translation performance [13], [14].

In the following sections, we describe our Bayesian model and Gibbs sampling in Sect. 2, and show our new approach to extracting phrases in Sect. 3. Section 4 reports the result of experiment. Section 5 compares the related research, and Sect. 6 draws the conclusions.

2. Word Alignment Models

2.1 Alignment

In statistical machine translation, one core task is to model the translation probability $P(F|E)$, which describes the relationship between a source sentence $F = f_1, f_2 \dots f_J$ and a target sentence $E = e_1, e_2 \dots e_I$. In alignment model $P(F, A|E)$, a hidden variable $A = a_1, a_2 \dots a_J$ is introduced to describe a mapping between words in E and F, and the value a_j is defined as the index of the word in E to which f_j is aligned. The relationship between the translation model and the alignment model is given by:

Manuscript received October 18, 2012.
Manuscript revised January 22, 2013.

[†]The authors are with the Department of Media Technology, Ritsumeikan University, Kusatsu-shi, 525–8577 Japan.
a) E-mail: lizezhonglaile@163.com

DOI: 10.1587/transinf.E96.D.1536

$$P(F|E) = \sum_{A} P(F, A|E)$$

A special case in the alignment model is $a_j = 0$, which means f_j is aligned with spurious word e_0 (i. e. not aligned with any target word). $P(F, A|E)$ can be written as a product of conditional probabilities. Each such product corresponds in a natural way to a generative process of developing (F, A) from E, and varying the generative process leads to different alignment models, such as the series of IBM models [10].

2.2 Review of IBM Model 4

IBM Model 4 is a fertility-based alignment model, where fertility ϕ_i is defined as the number of source words aligned with target word e_i. As is shown in (1), the alignment model is decomposed into fertility model $P(\phi_0^I|E)$, lexical model $P(\tau_0^I|\phi_0^I, E)$ and distortion model $P(\pi_0^I|\phi_0^I, \tau_0^I, E)$, n, t and d are their parameter sets respectively. This decomposition corresponds to such a generative process: given E, Model 4 first generates ϕ_i for each e_i with a probability of $n(\phi_i|e_i)$ and a list of source words to connect to it; this list is called tablet τ_i, and the k^{th} word τ_{ik} in the list τ_i is produced with a probability of $t(\tau_{ik}|e_i)$; after producing all the source words, Model 4 performs a distortion, i. e. places word τ_{ik} into position π_{ik} with a probability of $p_{ik}(\pi_{ik})$, and finally (F, A) is generated.

$$\begin{aligned} & P(F, A|E; n, t, d) \qquad (1) \\ & = P(\phi_0^I|E; n)P(\tau_0^I|\phi_0^I, E; t)P(\pi_0^I|\phi_0^I, \tau_0^I, E; d) \\ & = n_0\left(\phi_0|\sum_{i=1}^{I}\phi_i\right)\prod_{i=1}^{I} n(\phi_i|e_i)\prod_{i=0}^{I}\prod_{k=1}^{\phi_i} t(\tau_{ik}|e_i) \\ & \quad \frac{1}{\phi_0!}\prod_{i=1}^{I}\prod_{k=1}^{\phi_i} p_{ik}(\pi_{ik}) \end{aligned}$$

where

$$n_0\left(\phi_0|\sum_{i=1}^{I}\phi_i\right) = \left(\begin{array}{c} \sum_{i=1}^{I}\phi_i \\ \phi_0 \end{array}\right) {p_0}^{\sum_{i=1}^{I}\phi_i - 2\phi_0} {p_1}^{\phi_0} \qquad (2)$$

$$p_{ik}(\pi_{ik}) = \begin{cases} d_1(j - c_{\rho_i}|\mathfrak{A}(e_{\rho_i}), \mathfrak{B}(\tau_{i1})) & if\, k = 1 \\ d_{>1}(j - \pi_{ik-1}|\mathfrak{B}(\tau_{ik})) & if\, k > 1 \end{cases} \qquad (3)$$

The distortion and fertility probabilities for e_0 are treated differently. As is seen in Eq. (2), ϕ_0 follows a binomial distribution with auxiliary parameters p_0 and p_1. Equation (3) shows the distortion distribution, where d_1 is the probability of placing τ_{i1} into position π_{i1}, and $d_{>1}$ is the probability of placing τ_{ik} into position π_{ik}. Words aligned to e_0 are placed only after all the other words have been placed, i. e. the words are permuted in the left ϕ_0 vacancies, thus the probability of the permutation is $1/\phi_0!$. As for the other variables, ρ_i is the first position to the left of i for which $\phi_{\rho_i} > 0$, and c_ρ is the ceiling of the average position of the words of tablet τ_ρ, $\mathfrak{A}(e)$ is the target word class and $\mathfrak{B}(f)$ is the source word class (more details can be seen in Brown's paper) [10].

2.3 Bayesian Model

Our Bayesian model almost repeats the same generative scenarios shown in the previous section, but puts appropriate priors for the parameters in the model and changes a simplified distortion model. As the same in Model 4, both the fertility and translation for each target word follow a Multinomial distribution, but in our proposed Bayesian setting, all the fertility and translation parameters will be treated as random variables with priors, and Dirichlet distribution [11] seems to be a natural choice for them, since it is conjugate to the Multinomial distribution so that inference will be tractable. Note that we can't identify how many source words will be as the possible translations of a target word e, or how many kinds of fertility a word can have. In other words, we can't decide on the dimensionality for the distributions beforehand. Fortunately, Dirichlet Process (DP) [11] can solve this problem, which can be seen as an infinite-dimensional analogue of Dirichlet distribution:

$$n_e \sim DP(\alpha, N_0(\phi|e)) \qquad (4)$$

$$t_e \sim DP(\beta, T_0(f|e)) \qquad (5)$$

n_e and t_e denote all the fertility and lexical parameters for the target word e, α and β are concentration parameters that determine n_e and t_e's variances. We set base distributions N_0 as Poisson distribution with parameter λ_e to encode our prior knowledge that high fertility value should be discouraged, and λ_e denotes the expected fertility for e, and we assign 1 for all λ_e for simplicity. Formula (4) doesn't include the fertility parameters for e_0, and we still use Formula (2) to model it. As for base distribution T_0, shown as:

$$T_0(f|e) = \sum_{et,ft} p(et|e)p(ft|et)p(f|ft) \qquad (6)$$

where et (ft) denotes e's (f's) word class, $p(ft|et)$ is a class translation model, $p(et|e)$ is a transition probability from word to class, and $p(f|ft)$ is a uniform distribution (over word types included in ft) for each class ft. In practice, the word class can be replaced with Part-of-Speech (POS), and in this case, Eq. (6) encodes such a prior knowledge: POS provides clues for the alignment. Especially for Named Entity, words that share same Named Entity type tend to be aligned.

As for the distortion model, we abandon the condition on the word class to decrease model's complexity (as is shown in Eq. (3)), and here we simply adopt a distance penalty (except words aligned to e_0) shown as follows

$$p_\pi(A) \propto \frac{1}{\phi_0!}\prod_{j=1, a_j \neq 0}^{J} b^{|j - prev(j)|} \qquad (7)$$

$$prev(j) = \begin{cases} \pi_{\rho_i\phi_{\rho_i}} & if\, k = 1 \\ \pi_{ik-1} & if\, k > 1 \end{cases} \qquad (8)$$

where b is a fixed value less than 1, $prev(j)$ means position of f_j's predecessor along the j-axis whose coordinate represents the index of words in F. In the first part of (7), the reciprocal of $\phi_0!$ models the distortion procedure for words generated by e_0, which uses the same strategy as Model 4 (seen in Eq. (1)). In the second part, the continued product means a distance penalty. Due to the above simplification for distortion model, we will see a more direct and concise inference in the following section. Another theoretical reason is that we don't expect a skewed distribution for the above parameters the same as the lexical models. Therefore, it is unnecessary to put a sparse prior for these parameters. In the Bayesian framework, all the parameters will be treated as variables. With the given E_1^S and fixed hyper-parameters θ, the joint distribution of F_1^S, A_1^S and parameters can be noted as

$$P(F_1^S, A_1^S, n, t, d | E_1^S; \theta) \tag{9}$$

where θ represents all the hyper-parameters including α and β, n and t include all the n_e and t_e for each target word e, and d actually just contains b. Unlike Eq. (1), we write E_1^S (F_1^S) instead of E (F), which refers to all the S target (source) sentences included in the parallel corpus, since in practice, the inference is performed on the whole corpus rather than a single sentence.

2.4 Bayesian Inference

Instead of sampling the parameters explicitly, we adopt a collapsed sampler with all the parameters marginalized out

$$P(F_1^S, A_1^S | E_1^S; \theta) = \int_{n,t} P(F_1^S, A_1^S, n, t, d | E_1^S; \theta) \tag{10}$$

where d doesn't need integral since we don't treat this parameter as a random variable, and will be replaced by constant b in the left part of Formula 10. Using $P(F_1^S, A_1^S | E_1^S; \theta)$, we can easily estimate by normalization the distribution of alignment given E_1^S and F_1^S, which is the objective of our inference. Apparently, we can't get the analytic solution to the above posterior distribution in Formula 10, so a frequent strategy is Gibbs sampling [12], an instance of Markov Chain Monte Carlo technique. Since F is observable, the only hidden variable that needs sampling is A (here F and A belong to the current (E, F) that is being sampled), then we sample each a_j alternatively. The probability for a new component value when the other values are fixed is

$$\begin{aligned} P(a_j | A_{\neg j}, F_1^S, E_1^S; \theta) \propto\ & P_\phi(a_j | A_{\neg j}, F_1^S, E_1^S; \theta) \\ & P_\tau(a_j | A_{\neg j}, F_1^S, E_1^S; \theta) P_\pi(a_j | A_{\neg j}, F_1^S, E_1^S; \theta) \end{aligned} \tag{11}$$

where $A_{\neg j}$ denotes the alignment exclude a_j. As is shown in Formula 11, the probability of new sample can be calculated according to the three sub-models (fertility model P_ϕ, lexical model P_τ and distortion model P_π), which is very similar to the E step in the IBM models training [10], but in a way that can be metaphorized as Chinese Restaurant Process (CRP) or *cache model* instead of using fixed parameters.

First, we investigate the lexical model which is responsible for generating the appropriate translation word:

$$P_\tau(a_j | A_{\neg j}, F_1^S, E_1^S; \theta) \propto \frac{N(e_{a_j}, f_j) + \beta T_0(f_j | e_{a_j})}{\Sigma_f N(e_{a_j}, f) + \beta} \tag{12}$$

where $N(e, f)$ is the number of links between (e, f) in the other part of this sentence pair and other sentence pairs in the training corpus, and can be called *cache count*. This formula is deduced by the integral for t_e. One way of understanding this prediction that DP model makes is through cache model, where f can be generated either by drawing from the T_0 with a probability proportional to $\beta T_0(f|e)$, or by drawing from the cache of previous translation events with a probability proportional to $N(e, f)$.

Second, the probability of fertility model is

$$P_\phi(a_j | A_{\neg j}, F_1^S, E_1^S; \theta) \propto \frac{N(e_{a_j}, \phi_{a_j} + 1) + \alpha N_0(\phi_{a_j} + 1 | e_{a_j})}{N(e_{a_j}, \phi_{a_j}) + \alpha N_0(\phi_{a_j} | e_{a_j})} \tag{13}$$

where $N(e, \phi)$ is the frequency of cases that word e has a fertility ϕ, and the denominator encodes the fact that the new prediction will cause an instance of word-fertility to be removed from the cache as the new word-fertility is added.

As for the distortion model, it's unnecessary to consider the cache model for calculating the distortion model, we have

$$P_\pi(a_j | A_{\neg j}, F_1^S, E_1^S; \theta) \propto b^{|j - prev(j)| + |next(j) - j| - |next(j) - prev(j)|} \tag{14}$$

where the exponent means 3 distortions are changed (in this exponent, the first two terms are the new distortions, and the third term is the old distortion), and $next(j)$ is an inverse function of $prev(j)$, which means successor's position of f_j.

As in the Model 4, the fertility and distortion for e_0 are treated differently. If $a_j = 0$, Formula (13) and (14) should be replaced by (15) and (16) respectively, which can be derived from Formula (1) and (2).

$$\begin{aligned} P_\phi(a_j = 0 | A_{\neg j}, F_1^S, E_1^S; \theta) &\propto \frac{n_0(\phi_0 + 1 | \sum_{i=1}^I \phi_i)}{n_0(\phi_0 | \sum_{i=1}^I \phi_i)} \\ &= \frac{(\sum_{i=1}^I \phi_i - \phi_0) p_1}{(\phi_0 + 1) p_0} \end{aligned} \tag{15}$$

$$P_\pi(a_j = 0 | A_{\neg j}, F_1^S, E_1^S; \theta) \propto \frac{\phi_0!}{(\phi_0 + 1)!} = \frac{1}{\phi_0 + 1} \tag{16}$$

The whole procedure for sampling is described in Table 1: lines 1-3 initialize each bilingual pair with an HMM Viterbi alignment. Although initialization can be arbitrary in theory, HMM alignment can speed up the convergence; for-loops in line 4 and 5 denote sample each sentence in the corpus for each sampling iteration; in lines 6-13, alignment is sampled by sampling each a_j alternatively; line 7 deletes the old value of a_j, and accordingly update the cache count $N(e, f)$ and $N(e, \phi)$; line 9 calculates the probability of each

Table 1 Gibbs sampling for word alignment.

```
1   For each sentence pair $(E, F)$ in $(E_1^S, F_1^S)$
2      Initialize alignment
3   End for
4   For each iteration
5      For each sentence pair $(E, F)$ in $(E_1^S, F_1^S)$
6         For $j := 1$ to $J$
7            Delete $a_j$; update the cache count
8            For $i := 0$ to $I$
9               Calculate $p(a_j = i|A_{\neg j}, F, E; \theta)$
10           End for
11           Normalize $p(a_j|A_{\neg j}, F, E; \theta)$
12           Sample a new value for $a_j$; update the cache count
13        End for
14        If (Current iteration $\geq$ Burn-in)
15           Collect alignments for $(E, F)$
16     End for
17  End for
```

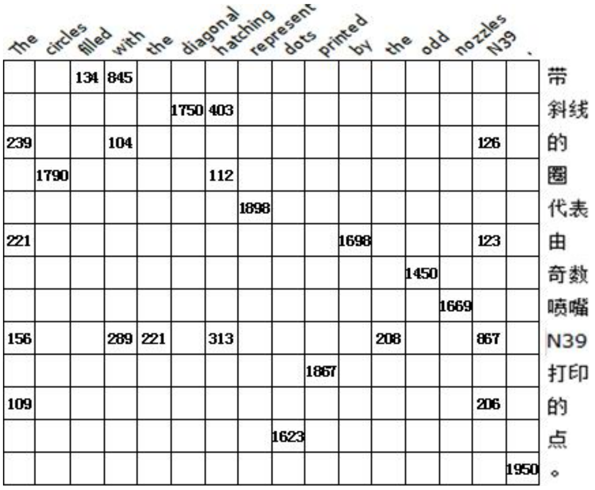
possible value for a_j, i. e. we get the distribution of a_j; and line 12 selects a possible value for a_j in terms of the distribution; and lines 14-15 collect the samples after burn-in which will be used in the succeeding training.

3. Phrase Table Training

We adopt the two-staged approach to phrase table training. Since previous research has pointed out that phrase extraction from Viterbi alignment will lead to inaccurate and insufficient phrase extraction, we extend the ideas proposed in [13], [14], where N-best alignments are exploited which helps reducing the propagation of errors to downstream estimation in the MT system.

Liu et al. [14] proposed a new structure called weighted alignment matrix (WAM) to encode all possible alignments for a sentence pair compactly. In this matrix, each element that corresponds to a word pair is assigned a probability to measure the confidence of aligning the two words, and the confidence score is calculated from N-best alignments. A similar matrix called alignment sample matrix (SM) is used to store our output of Gibbs sampling for word alignment, which consists of multiple samples of word alignment, but leads to a more direct phrase extraction. We will see that, it's more convenient to compute the confidence score.

Suppose we already get the output of Bayesian inference for word alignment that contains N samples bidirectionally, that is, we have $2N$ samples, then construct a $I \times J$ matrix SM, in which each element $SM(i, j)$ denotes the number of alignment points between (i, j) among all the samples (see Fig. 1). Our objective is to build a phrase table and associate each phrase pair $(e_{i_1}^{i_2}, f_{j_1}^{j_2})$ in the table with 4 scores, including relative frequency $tran(e_{i_1}^{i_2}|f_{j_1}^{j_2})$, $tran(f_{j_1}^{j_2}|e_{i_1}^{i_2})$, lexical weight $lex(e_{i_1}^{i_2}|f_{j_1}^{j_2})$ and $lex(f_{j_1}^{j_2}|e_{i_1}^{i_2})$, all of which will be used as features in the decoder of translation system. The overall framework is shown in Table 2: line 1 initializes an empty phrase table; in lines 2-10, we extract all the possible phrase pairs that meet some constraints; a local pruning is performed in lines 7-8; lines 11-16 compute the phrase count; and the left part calculates the 4 scores for each phase pair.



	The	circles	filled	with	the	diagonal	hatching	represent	dots	printed	by	the	odd	nozzles	N39	.
带			134	845												
斜线						1750	403									
的	239			104											126	
圈		1790					112									
代表								1898								
由	221										1698				123	
奇数													1450			
喷嘴														1669		
N39	156			289	221		313					208			867	
打印										1867						
的	109														206	
点									1623							
。																1950

Fig. 1 An example for SM, each element value denotes the number of alignment points between word pairs marked by its x coordinate and y coordinate, we omit the element that has a value less than 100.

Table 2 Building phrase table.

```
1   Initialize the phrase table $T = \{\}$
2   For each sentence pair $(E_s, F_s)$ , do
3      Construct the alignment sample matrix SM
4      $P_s = \{(e_{i_1}^{i_2}, f_{j_1}^{j_2})$ : satisfies some constraints$\}$
5      For each $(e_{i_1}^{i_2}, f_{j_1}^{j_2})$ in $P_s$
6         Calculate $f_C(e_{i_1}^{i_2}, f_{j_1}^{j_2})$
7         If($f_C(e_{i_1}^{i_2}, f_{j_1}^{j_2}) < \sigma$)
8            Discard $(e_{i_1}^{i_2}, f_{j_1}^{j_2})$ from $P_s$
9      End for
10  End for
11  For each $P_s$
12     For each $(\widetilde{e}, \widetilde{f})$ in $P_s$
13        Add $(\widetilde{e}, \widetilde{f})$ to $T$
14        Accumulate $f_C(e_{i_1}^{i_2}, f_{j_1}^{j_2})$ to $Count(\widetilde{e}, \widetilde{f})$
15     End for
16  End for
17  For each $(\widetilde{e}, \widetilde{f})$ in $T$
18     Calculate $tran(e_{i_1}^{i_2}|f_{j_1}^{j_2})$, $tran(f_{j_1}^{j_2}|e_{i_1}^{i_2})$, $lex(e_{i_1}^{i_2}|f_{j_1}^{j_2})$
    and $lex(f_{j_1}^{j_2}|e_{i_1}^{i_2})$
19  End for
```

3.1 Initial Phrase Extraction

This subsection discusses how to get initial set P_s in line 4 of Table 2. Apparently, it's not practical to enumerate all possible phrase pairs, which will result in slow decoding and translation noise. Therefore, constraints for extraction should be introduced. Different from the extraction from Viterbi alignment, we can't give a consistence constraint [1] since multiple alignments are combined into a matrix. Before we describe our constraint, we need define a concept called *Link probability* $p_m(i, j)$ to show how good the correspondence for (e_i, f_j) is in this sentence pair.

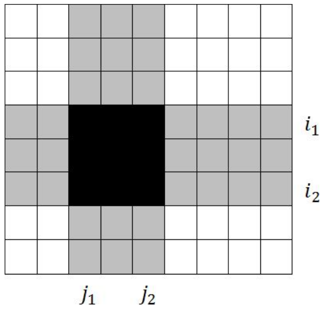


Fig. 2 Two regions of a phrase pair in SM.

$$p_m(i,j) = \frac{SM(i,j)}{2N} \tag{17}$$

Now our constraints can be described as: at least one $p_m(i,j)$ inside the phrase pair (the points in the black region of Fig. 2) is greater than σ_1, and none of $p_m(i,j)$ outside (the points in the gray region of Fig. 2) is greater than σ_2; the length limitations are set to be 6 both for source and target phrases, and prune phrase pairs whose difference in phrase length are higher than 4.

3.2 Relative Frequency

Using relative frequency as a feature is a common practice to measure the goodness of a bilingual phrase pair. And it's easy to get the traditional estimation for relative frequency $tran(e_{i_1}^{i_2}|f_{j_1}^{j_2})$, since we just need to count how often it occurs in the corpus. But for our extraction from multiple samples (see lines 5-19 in Table 2), we resort to fractional count.

In order to define the fractional count, we follow the approach in [14], and the difference is that we get phrases from SM instead of WAM. Given a SM and a phrase pair $(e_{i_1}^{i_2}, f_{j_1}^{j_2})$, two regions are identified: $in(i_1, j_1, i_2, j_2)$ (the black part in Fig. 2) and $out(i_1, j_1, i_2, j_2)$ (the gray part in Fig. 2). First, we define inside probability $\alpha(i_1, j_1, i_2, j_2)$ and outside probability $\beta(i_1, j_1, i_2, j_2)$ respectively.

$$\alpha(i_1, j_1, i_2, j_2) = 1 - \prod_{(i,j)\in in(i_1,j_1,i_2,j_2)} 1 - p_m(i,j)$$
$$\beta(i_1, j_1, i_2, j_2) = \prod_{(i,j)\in out(i_1,j_1,i_2,j_2)} 1 - p_m(i,j)$$

Then we can define the fractional count of phrase pair in region (i_1, j_1, i_2, j_2) as:

$$f_C(e_{i_1}^{i_2}, f_{j_1}^{j_2}) = \alpha(i_1, j_1, i_2, j_2)\beta(i_1, j_1, i_2, j_2)$$

We should notice that the above formula just means a local fractional count, which is estimated from a single region (i_1, j_1, i_2, j_2) in a sentence pair (E, F), and the global fractional count of phrase pair $(\widetilde{e}, \widetilde{f})$ can be calculated by collecting f_C from the entire corpus.

$$count(\widetilde{e}, \widetilde{f}) = \sum_{E,F} \sum_{i_1,j_1,i_2,j_2} f_C(e_{i_1}^{i_2}, f_{j_1}^{j_2})\delta(e_{i_1}^{i_2}, \widetilde{e})\delta(f_{j_1}^{j_2}, \widetilde{f})$$

where δ is the Kronecker delta function, equal to 1 when both of its arguments are the same and equal to 0 otherwise. With the aid of fractional count, we can get the relative frequency of phrase pair $(\widetilde{e}, \widetilde{f})$:

$$tran(\widetilde{e}|\widetilde{f}) = \frac{count(\widetilde{e}, \widetilde{f})}{\sum_{\widetilde{e}'} count(\widetilde{e}', \widetilde{f})}$$

3.3 Lexical Weight

Since most phrase pairs appear only a few times in the training corpus, lexical weight is introduced as another measurement for goodness of a phrase pair and has the advantage of avoiding overestimation [1]. The calculation for lexical weight can be written as

$$lex(e_{i_1}^{i_2}|f_{j_1}^{j_2}) = \prod_{i=i_1}^{i_2} \max_j t(e_i|f_j)$$

where $j \in \{j|t(e_i|f_j) > 0\, and\, (j_1 \leqslant j \leqslant j_2\, or\, j = 0)\}$, and $t(e|f)$ denotes lexical probability, so lexical weight measures the phrase quality on the level of words in the phrase pair. Recall the Bayesian inference in Sect. 2, and we just treat the lexical parameter as a hidden variable and integrate out it, thus we can't get it from the Bayesian inference directly. But through counting the number of alignment links in the Gibbs samples, we can get the expected values for the lexical parameters as follows

$$t(e|f) = \frac{\sum_{E,F}\sum_{i,j} SM(i,j)\delta(e_i, e)\delta(f_j, f)}{\sum_{E,F}\sum_j(\delta(f_j, f)\sum_i SM(i,j))}$$

3.4 Phrase Pruning

Low-quality phrase pairs should be discarded to boost efficiency of phrase training and alleviate decoding errors as well. As is seen in Table 2, the procedure described in lines 7 and 8 is called Phrase Pruning, and we discard any phrase pair that has a local fractional count lower than threshold σ.

4. Experiments

4.1 Corpus Preprocess

The corpus we used is a Chinese-English corpus in domain of patent, which is released by NTCIR 9 [16]. Corpus preprocessing is necessary, such as Chinese segmentation, long sentence filtering and numeric character processing. Finally, we select 350000 sentence pairs as the training corpus (sentences longer than 35 words are filtered), 1000 pairs as the development set, and 1000 pairs as the test set (we prepare one referenced translation for each sentence) for translation.

4.2 Evaluation for Word Alignment

We adopt the standard AER (alignment error rate) as our evaluation metric [18]. We annotated 300 sentence pairs only with sure alignment for alignment evaluation.

Table 3 Performance of word alignment.

Method Description	AER
GIZA++ & grow-diag-final	16.12%
Proposed & δ=0.3 & POS	13.70%
Proposed & δ=0.4 & POS	12.45%
Proposed & δ=0.5 & POS	13.08%
Proposed & δ=0.3 & word cluster	15.23%
Proposed & δ=0.4 & word cluster	14.06%
Proposed & δ=0.5 & word cluster	14.78%

Table 4 Comparison of performance in terms of AER.

Method	AER	Training Time
Proposed	12.45%	40h
GIZA++ & grow-diag-final	16.12%	23h
VB	13.90%	28h
SHMM	13.58%	12h
PR	13.21%	32h
Bayesian Model 1	17.34%	5h

Table 5 Translation performance in English-Chinese with varying σ.

σ	0.01	0.05	0.1	0.2
BLEU%	31.40	31.49	31.28	31.10

We take Model 4 as a standard baseline, since it seems that Model 4 provides the most stable and widely-used baseline for many language pairs and domains. First, we run GIZA++ bidirectionally in the standard configuration (training scheme is abbreviated as $1^5H^53^34^3$) and have a symmetrization following the heuristics grow-diag-final [1].

Before running our Bayesian aligner, we should estimate the parameters in Eq. (6). We tagged the training corpus using some POS taggers, and replace each word by its POS to get a POS parallel corpus. Then, we ran IBM model 1 on the POS corpus to get the POS translation probabilities. Through dividing the number of occurrences of the word-tag pair (e, et) by the number of occurrences of e, we can get $p(et|e)$. For each POS ft, if word f is tagged with ft at least once in the training corpus, then $p(f|ft)$ is equal to the result of dividing 1 by the number of unique words tagged with ft; otherwise, $p(f|ft)$ is 0. Another way to tag the corpus is using some automatically induced word clusters. Moreover, we used mkcls (included in GIZA++) to get 50 word clusters both for English and Chinese, and the comparisons between them are shown in Table 3.

We set 1000 as the number of total iterations and 0 as the burn-in value, and set $\alpha = 1$, $\beta = 100$ and $b = 0.9$. After two unidirectional Bayesian models are trained simultaneously, we combine them using soft union [26], where an alignment link (i, j) is kept if $p_m(i, j) > \delta$. Apparently, varying this threshold offers a natural way to tradeoff precision and recall for alignment. Table 3 shows the comparison of AER between Model 4 and our Bayesian model with δ in several values. We can see that Bayesian model reveals a satisfying improvement for alignment quality when using POS and $\delta = 0.4$ (hereafter our proposed method will adopt this configuration without explicit illustrations), with a reduction of 3.67% over baseline in terms of AER, and we attribute this improvement to the superiority of Bayesian inference. Table 3 also compares the results between using POS and using word clusters generated by mkcls, and POS shows a better performance. We think this is due to the fact that POS tagger can get a more reliable word class than unsupervised word clustering, since the POS tagger is trained on a large annotated corpus in a supervised way.

With the aid of some open source toolkits, including Variational GIZA++ [23] which implements Variational Bayes (henceforth VB), Berkeley aligner [24], PostCat [25] and Mermer's Perl code for Bayesian inference [22], we had a comparison with state-of-the-art approaches in Table 4. All the experiments are run on a computer with 16 Intel Xeon CPUs (dual-cores, 3.00 GHz) and 16G memory. Here are some important configurations for the above toolkits. As for VB, we also trained in the bootstrapping regimen of $1^5H^53^34^3$ with hyper-parameter set to 0 and symmetrized using the grow-diag-final heuristic. Line 3 is Berkeley's symmetric HMM (henceforth SHMM) [24], where we set 0.3 as the threshold for the posterior decoding. In line 4, we use the Posterior Regularization (henceforth PR) with symmetry constraint which is proved better than bijectivity constraint [25]; we set 0.002 as the convergence stopping criteria which determines the length of training time sensitively, and set 0.3 as the threshold for soft union [26]. Line 5 is a fully Bayesian inference for Model 1, and we used the default parameters configured in the Perl toolkit. We refer the reader for the original papers for more detailed meanings of these configurations.

As is shown in Table 4, we can see that our model outperforms VB, joint HMM, PR and Bayesian Model 1. As for the reason why our approach is better than VB, we think Gibbs sampling is superior to Variational approach on inferring word alignment, although it is proved not like this for POS tagging [15]. As for the SHMM, PR and Bayesian Model 1, our model shows the advantages of fertility-based models over sequence-based models, and proves a similar perspective that fertility is an inherent cross-language property [9].

4.3 Evaluation for Translation

We used Moses as the decoder, SRI Language Model Toolkit [19] to train a 4-grams model on the target side of training corpus. We evaluated the translation quality using BLEU metric [21], which is the most popular method for automatic evaluation of machine translation.

Firstly, we configure our threshold values in our proposed approach. We set $\sigma_1 = 0.3$ and $\sigma_2 = 0.5$. As for the threshold of fractional count σ, we conducted the following experiments as is shown in Table 5, which shows the optimal value of σ is 0.05.

For the sake of comparison, we conducted 6 experiments in Table 6. All the experiments incorporated the default features in Moses: $tran(\tilde{e}|\tilde{f})$, $tran(\tilde{f}|\tilde{e})$, $lex(\tilde{e}|\tilde{f})$, $lex(\tilde{f}|\tilde{e})$, language model feature, word penalty, phrase penalty and linear distortion feature. Since lexicalized re-

Table 6 Performance of final translation (BLEU%).

No.	Method Description	English-Chinese
Experiment 1	Proposed	31.49
Experiment 2	GIZA++ & Viterbi	30.46
Experiment 3	VB & Viterbi	30.70
Experiment 4	GIZA++ & WAM	30.79
Experiment 5	SHMM & WAM	30.98
Experiment 6	PR & WAM	31.16

Table 7 Translation evaluation on the second data set (BLEU%).

No.	Method Description	English-Chinese
Experiment 1	Proposed	24.93
Experiment 2	GIZA++ & Viterbi	23.78
Experiment 3	VB & Viterbi	24.21
Experiment 4	GIZA++ & WAM	24.19
Experiment 5	SHMM & WAM	24.52
Experiment 6	PR & WAM	24.98

ordering features are trained from Viterbi alignment, we optionally added 6 lexicalized reordering features (in Experiment 2 and 3). In experiment 1, we implemented the whole framework introduced in this paper. Experiment 2-3 used 1-best alignment to extract phrases satisfying consistence constraint. Experiment 4 repeated the WAM approach [14] by using its accompanying toolkits. Experiment 5 used symmetric HMM alignment and Experiment 6 used alignment generated by PostCat, and both of them used toolkit Geppetto (http://code.google.com/p/geppetto/) for phrase extraction. In all the above experiments, we set 6 as the maximal phrase length both for Chinese and English, and phrase pairs whose difference in phrase length higher than 4 are filtered. Although all the last 3 approaches are called "WAM", it is worth noting that they get the link's posterior in different ways. Experiment 4 used n-best alignments to calculate an approximate posterior, whereas 5 and 6 used forward-backward algorithm to get an exact posterior estimation because HMM is tractable.

As is shown in Table 6, the improvement over Experiment 2-5 is statistically significant at $p < 0.05$ by using sign-test [27]. Although the improvements over Experiment 7 are not always statistically significant, our approach maintains consistent superiority in translation quality.

4.4 Evaluation on the Second Data Set

To further examine the effectiveness of our approach, we evaluate the translation performance on the second data set. We used FBIS newswire data (LDC2003E14), and selected 100K Chinese-English pairs as the training corpus, 1K as the develop set and 1K as the test set. We reuse the same configurations introduced in the last section, and evaluation results are listed in Table 7. We get similar results with the previous evaluation, a slight difference is that our approach is a little inferior to PR based approach, but still has a decent improvement over Experiment 2-5.

5. Related Work

The most prevailing method for inferring the parameters of a probabilistic model is based on the principle of MLE, and EM is a special case when there exist hidden variables. However, there is a trend that research in NLP is turning away from EM in favor of Bayesian methods, such as POS tagging [15], PCFG [20], word alignment [22] and phrase alignment [3].

Our approach to word alignment is similar to [22] in spirit to Bayesian inference, where it places a prior for the model parameters and adopts a collapsed sampler, but they take Model 1 as the inference objective, which we suppose somewhat simple and crude. [23] used variational Bayes which closely resembles the normal form of EM algorithm to improve the performance of GIZA++, as well as the translation performance. Together with the approach introduced in this paper, all these efforts aim to obviate the "garbage collector effect" which increases the likelihood of the training data [23] but obviously is unreasonable. Intuitively, this overfitting manifests as a high fertility value for the rare word, but mathematically, this overfitting shows up as a lexical distribution closer to uniform, i. e. the rare word spreads its probability mass broadly over too many target words. Whereas sparse prior can lead to a more skewed distribution to overcome the overfitting, which is the theoretical advantage of Bayesian approach.

Contrary to the Bayesian trend, some works adhere to the old-fashioned EM but in a modified way. Two typical works are done by [24] and [25], both of which modify the E-step. The former used the product of bidirectional posterior distribution (normalization is necessary) to replace the unidirectional posterior distribution for the latent alignment, which made the model symmetric, the latter estimated the posteriors in E-step with rich constraints. Their improvements over baselines can be seen in Table 4.

Zhao [9] proposed a brief fertility based HMM model, which also decreases the complexity of Model 4 but keeps the fertility as a component of modeling. They didn't place any prior on the parameters, which can be viewed as a stochastic EM, and they assumed fertility follows Poisson distribution, whereas we took Poisson distribution as the base distribution for fertility in the DP prior.

At the second phase of the two-staged training in SMT, phrase table training is usually performed on the basis of word alignment. The word aligner outputs multiple possible alignments along with corresponding probabilities. However, the common practice is using only optimal alignment, which obviously results in an information loss. Recent studies have shown phrase table training can benefit from multiple alignments. As is shown in [14], to construct the matrix, 550 alignments were obtained from 50 × 50 bidirectional alignment pairs for each sentence pair on average and renormalized to estimate the posterior probability. Nevertheless, in our approach, we simply need collect samples since they are generated from the posterior distribution, which is guaranteed by Gibbs sampling. Wang et al. [17] proposed a general and extensible framework for phrase extraction, and their main contribution is to provide a toolkit that can easily experiment any combination between various word alignments and phrase extraction heuristics. They also paid special attention to some heuristics related with punctuations and Chinese particles, and produced an improvement in spoken text.

6. Conclusions and Future Work

In this paper, we have proposed a Bayesian inference for word alignment, which currently is a promising replacement for EM and has already been broadly applied for various tasks in the field of NLP. To the best of our knowledge, it is the first attempt to adopt a fully Bayesian inference to a fertility-based alignment model. We also proposed a novel method for phrase table learning from Gibbs samples. Our experiments show a decent improvement for word alignment as well as a significant improvement on translation quality. As for our future work, we are trying to extract discontinuous phrases from the alignment samples, and further improve the translation quality.

References

[1] P. Koehn, F.J. Och, and D. Marcu, "Statistical phrase based translation," Proc. HLT-NAACL, pp.48–54, 2003.

[2] D. Marcu and D. Wong, "A phrase-based, joint probability model for statistical machine translation," Proc. EMNLP, pp.133–139, 2002.

[3] J. DeNero, et al., "Sampling alignment structure under a bayesian translation model," Proc. EMNLP, pp.314–323, 2008.

[4] P. Koehn, et al., "Moses: open source toolkit for statistical machine yranslation," Proc. ACL, pp.177–180, 2007.

[5] D. Cer, et al., "Phrasal: A toolkit for statistical machine translation with facilities for extraction and incorporation of arbitrary model features," Proc. NAACL, pp.177–180, 2010.

[6] S. Goldwater and T. Griffiths, "A fully bayesian approach to unsupervised Part-of-Speech tagging," Proc. ACL, pp.744–751, 2007.

[7] R.C. Moore, "Improving IBM word alignment model 1," Proc. ACL, pp.518–525, 2004.

[8] P. Resnik and E. Hardisty, "Gibbs sampling for the uninitiated," Technical Report, University of Maryland, 2010.

[9] S.J. Zhao and D. Gildea, "A fast fertility hidden markov model for word alignment using MCMC," Proc. EMNLP, pp.596–605, 2010.

[10] P.F. Brown, Stephen A.D. Pietra, V.J.D. Pietra, and R.L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," Computational Linguistics, vol.19, no.2, pp.263–311, 1993.

[11] T.S. Ferguson, "A Bayesian analysis of some nonparametric problems," Annals of Statistics, vol.1, no.2, pp.209–230, 1973.

[12] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," IEEE Trans. Pattern Anal. Mach. Intell., vol.6, no.6, pp.721–741, 1984.

[13] A. Venugopal, et al., "Wider pipelines: N-best alignments and parses in MT training," Proc. AMTA, pp.192–201, 2008.

[14] Y. Liu, T. Xia, X.Y. Xiao, and Q. Liu, "Weighted alignment matrices for statistical machine translation," Proc. EMNLP, pp.1017–1026, 2009.

[15] M. Johnson, "Why doesn't EM Find Good HMM POS-taggers," Proc. EMNLP-CoNLL, pp.296–305, 2007.

[16] T. Sakai, J. Hideo, "Overview of NTCIR-9," Workshop of NTCIR-9, pp.559–578, 2011.

[17] L. Wang, T. Lus, J. Graca, L. Coheur, and I. Trancoso, "Towards a general and extensible phrase-extraction algorithm," Proc. IWSLT, pp.313–320, 2005.

[18] F.J. Och and H. Ney, "A systematic comparison of various statistical alignment models," Computational Linguistics, vol.29, no.1, pp.19–51, 2003.

[19] A. Stolke, "SRILM: an extensible language modeling toolkit," Proc. ICSLP, pp.901–904, 2002.

[20] M. Johnson, T. Griffiths, and S. Goldwater, "Bayesian inference for PCFGs via Markov chain Monte Carlo," Proc. HLT-NAACL, pp.139–146, 2007.

[21] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "BLEU: A method for automatic evaluation of machine translation," Proc. ACL, pp.311–318, 2002.

[22] C. Mermer and M. Saraclar, "Bayesian word alignment for statistical machine translation," Proc. ACL, pp.182–187, USA, 2011.

[23] D. Riley and D. Gildea, "Improving the IBM alignment models using variational bayes," Proc. ACL, pp.306–310, 2012.

[24] P. Liang, B. Taskar, and D. Klein, "Alignment by agreement," Proc. HLT-NAACL, pp.104–111, 2006.

[25] J.V. Graca, K. Ganchev, and B. Taskar, "Learning tractable word alignment models with complex constraints," Computational Linguistics, vol.36, no.3, pp.481–504, 2010.

[26] J. DeNero and D. Klein, "Tailoring word alignments to statistical machine translation," Proc. ACL, pp.17–24, 2007.

[27] M. Collins, P. Koehn, and I. Kucerova, "Clause restructuring for statistical machine translation," Proc. ACL, pp.531–540, 2005.

Zezhong Li is currently a PhD candidate in Department of Computer Science, Ritsumeikan University, His main research interests include Machine Translation and Natural Language Processing.

Hideto Ikeda received the PhD from Hiroshima University. Dr. Ikeda is currently a professor at Ritsumeikan University. His main research interests include Database, eLearning, Machine Translation and Natural Language Processing.

Junichi Fukumoto received the PhD in Language and Linguistics from University of Manchester in 1999. Dr. Fukumoto is currently a professor at Ritsumeikan University. He is a member of the ACL and IEICE. His main research interests include Question Answering and Natural Language Processing.