Bayesian Semi-Supervised Chinese Word Segmentation for Statistical Machine Translation

Jia Xu[†], **Jianfeng Gao**[*], **Kristina Toutanova**[*], **Hermann Ney**[†]

Computer Science 6[†]
RWTH Aachen University
D-52056 Aachen, Germany
{xujia,ney}@cs.rwth-aachen.de

Microsoft Corporation[*]
One Microsoft Way
Redmond, WA 98052, USA
{jfgao,kristout}@microsoft.com

Abstract

Words in Chinese text are not naturally separated by delimiters, which poses a challenge to standard machine translation (MT) systems. In MT, the widely used approach is to apply a Chinese word segmenter trained from manually annotated data, using a fixed lexicon. Such word segmentation is not necessarily optimal for translation. We propose a Bayesian semi-supervised Chinese word segmentation model which uses both monolingual and bilingual information to derive a segmentation suitable for MT. Experiments show that our method improves a state-of-the-art MT system in a small and a large data environment.

1 Introduction

Chinese sentences are written in the form of a sequence of Chinese characters, and words are not separated by white spaces. This is different from most European languages and poses difficulty in many natural language processing tasks, such as machine translation.

It is difficult to define “correct” Chinese word segmentation (CWS) and various definitions have been proposed. In this work, we explore the idea that the best segmentation depends on the task, and concentrate on developing a CWS method for MT, which leads to better translation performance.

The common solution in Chinese-to-English translation has been to segment the Chinese text using an off-the-shelf CWS method, and to apply a standard translation model given the fixed segmentation. The most widely applied method for MT is unigram segmentation, such as segmentation using the LDC (LDC, 2003) tool, which requires a manual lexicon containing a list of Chinese words and their frequencies. The lexicon and frequencies are obtained using manually annotated data. This method is sub-optimal for MT. For example, 纸(paper) and 牌(card) can be two words or composed into one word 纸牌(cards). Since 纸牌does not exist in the manual lexicon, it cannot be generated by this method.

In addition to unigram segmentation, other methods have been proposed. For example, (Gao et al., 2005) described an adaptive CWS system, and (Andrew, 2006) employed a conditional random field model for sequence segmentation. However, these methods are not specifically developed for the MT application, and significant improvements in translation performance need to be shown.

In (Xu et al., 2004) and (Xu et al., 2005), word segmentations are integrated into MT systems during model training and translation. We refine the method in training using a Bayesian semi-supervised CWS approach motivated by (Goldwater et al., 2006). We describe a generative model which consists of a word model and two alignment models, representing the monolingual and bilingual information, respectively. In our methods, we first segment Chinese text using a unigram segmenter, and then learn new word types and word distributions, which are suitable for MT.

Our experiments on both large (NIST) and small (IWSLT) data tracks of Chinese-to-English translation show that our method improves the performance of a state-of-the-art machine translation system.

2 Review of the Baseline System

2.1 Word segmentation

In statistical machine translation, we are given a Chinese sentence in characters $c_1^K = c_1 \dots c_K$ which is to be translated into an English sentence $e_1^I = e_1 \dots e_I$. In order to obtain a more adequate mapping between Chinese and English words, c_1^K is usually segmented into words $f_1^J = f_1 \dots f_J$ in preprocessing.

In our baseline system, we apply the commonly

Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 1017–1024
Manchester, August 2008

used **unigram model** to generate the segmentation. Given a manually compiled lexicon containing words and their relative frequencies $P_s(f'_j)$, the best segmentation f_1^J is the one that maximizes the joint probability of all words in the sentence, with the assumption that words are independent of each other[1]:

$$\begin{aligned} f_1^J &= \operatorname*{argmax}_{f'^{J'}_1} Pr(f'^{J'}_1 | c_1^K) \\ &\approx \operatorname*{argmax}_{f'^{J'}_1} \prod_{j=1}^{J'} P_s(f'_j), \end{aligned}$$

where the maximization is taken over Chinese word sequences whose character sequence is c_1^K.

2.2 Translation system

Once we have segmented the Chinese sentences into words, we train standard alignment models in both directions with GIZA++ (Och and Ney, 2002) using models of IBM-1 (Brown et al., 1993), HMM (Vogel et al., 1996) and IBM-4 (Brown et al., 1993).

Our MT system uses a phrase-based decoder and the log-linear model described in (Zens and Ney, 2004). Features in the log-linear model include translation models in two directions, a language model, a distortion model and a sentence length penalty. The feature weights are tuned on the development set using a downhill simplex algorithm (Press et al., 2002). The language model is a statistical ngram model estimated using modified Kneser-Ney smoothing.

3 Unigram Dirichlet Process Model for CWS

The simplest version of our model is based on a unigram Dirichlet Process (DP) model, using only monolingual information. Different from a standard unigram model for CWS, our model can introduce new Chinese word types and learn word distributions automatically from unlabeled data.

According to this model, a corpus of Chinese words $f_1, \ldots f_m, \ldots, f_M$ is generated via:

$$G|\alpha, P_0 \sim DP(\alpha, P_0)$$

$$f_m|G \sim G$$

where G is a distribution over words drawn from a Dirichlet Process prior with base measure P_0 and concentration parameter α.

We never explicitly estimate G but instead integrate over its possible values and perform Bayesian inference. It is easy to compute the probability of a Chinese word given a set of already generated words, while integrating over G. This is done by casting Chinese word generation as a Chinese restaurant process (CRP) (Aldous, 1985), i.e. a restaurant with an infinite number of tables (approximately corresponding to Chinese word types), each table with infinite number of seats (approximately corresponding to Chinese word frequencies).

The Dirichlet Process model can be viewed intuitively as a cache model (Goldwater et al., 2006). Each word f_j in the corpus is either retrieved from a cache or generated anew given the previously observed words f_{-j}:

$$P(f_j|f_{-j}) = \frac{N(f_j){}_{+}\alpha P_0(f_j)}{N + \alpha}, \tag{1}$$

where $N(f_j)$ is the number of Chinese words f_j in the previous context. N is the total number of Chinese words, P_0 is the base probability over words, and α influences the probability of introducing a new word at each step and controls the size of the lexicon. The probability of generating a word from the cache increases as more instances of that word are seen.

For the base distribution P_0, which governs the generation of new words, we use the following distribution (called the **spelling model**):

$$\begin{aligned} P_0(f) &= P(L)P_0(f|L) \\ &= \frac{\lambda^L}{L!} e^{-\lambda} u^L \end{aligned} \tag{2}$$

where $\frac{1}{u}$ is the number of characters in the document, i.e. character vocabulary size, and L is the number of Chinese characters of word f. We note that this is a Poisson distribution on word length and a unigram distribution on characters given the length. We used $\lambda = 2$ and $\alpha = 0.3$ in our experiments.

4 CWS Model for MT

As a solution to the problems with the conventional approach to CWS mentioned in Section 1, we propose a generative model for CWS in Section 4.1, and then extend the model to a more general but deficient model, similar to a maximum entropy model in which most features are derived from the submodels of the generative model.

4.1 Generative Model

The generative model assume that a corpus of parallel sentences $({c_1}^K, {e_1}^I)$ is generated along with a hidden sequence of Chinese words ${f_1}^J$ and a hidden word alignment ${b_1}^I$ for every sentence. The alignment indicates the aligned Chinese word f_{b_i} for each English word e_i, where f_0 indicates a special *null* word as in the IBM models.

[1] The notational convention will be as follows: we use the symbol $Pr(\cdot)$ to denote general probability distributions with (nearly) no specific assumptions. In contrast, for model-based probability distributions, we use the generic symbol $P(\cdot)$.

Without assuming any special form for the probability of a sentence pair along with hidden variables, we can factor it into a monolingual Chinese sentence probability and a bilingual translation probability as follows:

$$\begin{aligned} & Pr({c_1}^K, {e_1}^I, {f_1}^J, {b_1}^I) \\ & \quad = Pr(c_1^K, f_1^J) Pr(e_1^I, b_1^I | f_1^J) \\ & \quad = Pr(f_1^J) \delta({f_1}^J, {c_1}^K) Pr(e_1^I, b_1^I | f_1^J), \end{aligned}$$

where $\delta(f_1^J, c_1^K)$ is 1 if the characters of the sequence of words ${f_1}^J$ are ${c_1}^K$, and to 0 otherwise. We can drop the conditioning on ${c_1}^K$ in $Pr(e_1^I, b_1^I | f_1^J)$, because the characters are deterministic given the words.

The joint probability of the observations $({c_1}^K, e_1^I)$ can be obtained by summing over all possible values of the hidden variables f_1^J and b_1^I.

In Sections 4.1.1 and 4.1.2, we will describe the modeling assumptions behind the monolingual Chinese sentence model and the translation model, respectively.

4.1.1 Monolingual Chinese sentence model

We use the Dirichlet Process unigram word model introduced in section 3. In this model, the parameters of a distribution over words G are first drawn from the Dirichlet prior $DP(\alpha, P_0)$. Words are then independently generated according to G. The probability of a sequence of Chinese words in a sentence is thus:

$$Pr(f_1^J) \approx \prod_{j=1}^{J} P(f_j | G) \tag{3}$$

4.1.2 Translation model

We employ the Dirichlet Process inverse IBM model 1 to generate English words and alignment given the Chinese words. In this model, for every Chinese word f (including the *null* word), a distribution over English words G_f is first drawn from a Dirichlet Process prior $DP(\alpha, P_0(e))$, where $P_0(e)$ we used the empirical distribution over English words in the parallel data. Then, given these parameters, the probability of an English sentence and alignment given a Chinese sentence (sequence of words) is given by:

$$P(e_1^I, b_1^I | f_1^J, G_f) = \prod_{i=1}^{I} \frac{1}{J+1} P(e_i | G_{f_{b_i}})$$

This is the same model form as inverse IBM model 1, except we have placed Dirichlet Process priors on the Chinese-word specific distributions over English words. [2]

In practice, we observed that using a word-alignment model in one direction is not sufficient. We then added a factor to our model which includes word alignment in the other direction , i.e. a Dirichlet Process IBM model 1. We ignore the detailed description here, because the calculation is the same as that of the inverse IBM model 1. According to this model, for every English word e (including the *null* word), a distribution over Chinese words G_e is first drawn from a Dirichlet Process prior $DP(\alpha, P_0(f))$. Here, for the base distribution $P_0(f)$ we used the same spelling model as for the monolingual unigram Dirichlet Process prior. The probability of a sequence of Chinese words f_1^J and a word alignment a_1^J given a sequence of English words e_1^I is then:

$$P(f_1^J, a_1^J | e_1^I, G_e) = \prod_{j=1}^{J} \frac{1}{I+1} P(f_j | G_{e_{a_j}})$$

4.2 Final Model

We put the monolingual model and the translation models in both directions together into a single model, where each of the component models is weighted by a scaling factor. This is similar to a maximum entropy model. We fit the weights of the sub-models on a development set by maximizing the BLEU score of the final translation.

$$\begin{aligned} & P(c_1^K, e_1^I, f_1^J, a_1^J, b_1^I) \\ & \approx \frac{1}{Z} P(f_1^J)^{\lambda_1} \cdot P(e_1^I, b_1^I | f_1^J)^{\lambda_2} \\ & \quad \cdot P(f_1^J, a_1^J | e_1^I)^{\lambda_3}, \end{aligned} \tag{4}$$

where Z is the normalization factor.

In practice we do not re-normalize the probabilities and our model is thus deficient because it does not sum to 1 over valid observations. However, we found the model work very good in our experiments. Similar deficient models have been used very successfully before, for example, in the IBM models 3–6 and in the unsupervised grammar induction model of (Klein and Manning, 2002).

5 Gibbs Sampling Training

It is generally impossible to find the most likely segmentation according to our Bayesian model using exact inference, because the hidden variables do not allow exact computation of the integrals. Nonetheless, it is possible to define algorithms using Markov chain Monte Carlo (MCMC) that produce a stream of samples from the posterior distribution of the hidden variables given the observations. We applied the Gibbs sampler (Geman and Geman, 1984) — one of the simplest MCMC methods, in which transitions between states of the

[2] f_{b_i} is the Chinese word aligned to e_i and $G_{f_{b_i}}$ is the distribution over English words conditioned on the word f_{b_i}. Similarly, e_{a_j} is the English word aligned to f_j in the other direction and $G_{e_{a_j}}$ is the distribution over Chinese words conditioned on e_{a_j}.

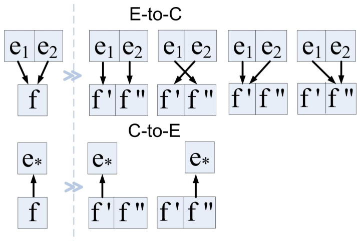

Figure 1: Case I, transition from a no-boundary to a boundary state, f to $f'f''$.

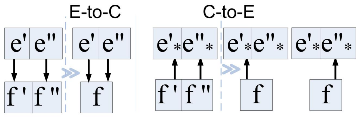

Figure 2: Case II, transition from a boundary to a no-boundary state, $f'f''$ to f.

Markov chain result from sampling each component of the state conditioned on the current value of all other variables.

In our problem, the observations are $D = (d_1, ..d_n, .., d_N)$, where d_n=(c_1^K, e_1^I) indicates a bilingual sentence pair, the hidden variables are the word segmentations f_1^J and the alignments in two directions a_1^J and b_1^I.

To perform Gibbs sampling, we start with an initial word segmentation and initial word alignments, and iteratively re-sample the word-segmentation and alignments according to our model of Equation 4.

Note that for efficiency, we only allow limited modifications to the initial word alignments. Thus we only use models derived from IBM-1 (instead of IBM-4) for comparing different word segmentations. On the other hand, re-sampling the segmentation causes re-linking alignment points to parts or groups of the original words.

Hence, we organize our sampling process around possible word boundaries. For each character c_k in each sentence, we consider two alternative segmentations: ${c_k}^+$ indicates the segmentation where there is a boundary after c_k and ${c_k}^-$ indicates the segmentation where there is no boundary after c_k, keeping all other boundaries fixed. Let f denote the single word spanning character c_k when there is no boundary after it, and f',f'' denote the two adjacent words resulting if there is a boundary: f' includes c_k and f'' starts just to the right, with character c_{k+1}. The introduction of f' and f'' leads to M new possible alignments in the E-to-C direction $b_{k1}^+, \ldots, b_{kM}^+$, such as in Figure 1. Together with the boundary vs no-boundary state at each character position, we re-sample a set of alignment links between English words and any of the Chinese words f,f', and f'', keeping all other word alignments in the sentence pair fixed. (See Figures 1 and 2.)

Table 1: General Algorithm of GS for CWS.

```
Input: $D$ with an initial segmentation and alignments
Output: $D$ with sampled segmentation and alignments
for $n = 1$ to $\hat{N}$
    for $k = 1$ to $K$ that $c_k \in d_n$
        Create $M$+1 candidates, $cba_{k,m}^+$ and $cba_k^-$, where
            $cba_{k,m}^+$: there is a word boundary after $c_k$
            $cba_k^-$: there is no word boundary after $c_k$
        Compute probabilities
            $P(cba_{k,m}^+|{dh_{nk}}^-)$
            $P(cba_k^-|{dh_{nk}}^-)$
        Sample boundary and relevant alignments
        Update counts
```

Thus at each step in the Gibbs sampler, we consider a set of alternatives for the boundary after c_k and relevant alignment links, keeping all other hidden variables fixed. At each step, we need to compute the probability of each of the alternatives, given the fixed values of the other hidden variables.

We introduce some notation to make the presentation easier. For every position k in sentence pair n, we denote by ${dh_{nk}}^-$ the observations and hidden variables for all sentences other than sentence n, and the observations and hidden variables inside sentence n, not involving character position c_k. The fixed variables inside the sentence are the words not neighboring position k, and the alignments in both directions to these words.

In the process of sampling, we consider a set of alternatives: segmentation ${c_k}^+$ along with the product space of relevant alignments in both directions $b_{k1}^+, \ldots, b_{kM}^+$, and a_k^+, and segmentation c_k^- along with relevant alignments ${b_k}^-$ and a_k^-. For brevity, we denote these alternatives by ${cba_{k,m}}^+$ and ${cba_k}^-$.

We describe how we derive the set of alternatives in section 5.2 and how we compute their probabilities in section 5.1.

Table 1 shows schematically one iteration of Gibbs sampling through the whole training corpus of parallel sentences, where $\hat{N}$ is the number of parallel sentences.

5.1 Computing probabilities of alternatives

For the Gibbs sampling algorithm in Table 1, we need to compute the probability of each alternative segmentation/alignments, given the fixed values of the rest of the data ${dh_{nk}}^-$. The probability of the hidden variables in the alternatives is proportional to the joint probability of the hidden variables and observations, and thus it is sufficient to compute the probability of the latter. We compute these probabilities using the Chinese restaurant process sampling scheme for the Dirichlet Process, thus integrating over all of the possible values of the distributions G, G_f and G_e.

Let cba_k denote an alternative hypothesis including boundary or no boundary at position k, and relevant alignments to English words in both directions of the one or two Chinese words resulting from the segmentation at k. The probability of this configuration given by our model is:

$$P(cba_k|dh_{nk}{}^{-}) \propto P_m(cba_k|dh_{nk}{}^{-})^{\lambda_1} \cdot P_{ef}(cba_k|dh_{nk}{}^{-})^{\lambda_2} \cdot P_{fe}(cba_k|dh_{nk}{}^{-})^{\lambda_3}, \quad (5)$$

where $P_m(cba_k|dh_{nk}{}^{-})$ is the monolingual word probability, and $P_{fe}(cba_k|dh_{nk}{}^{-})$ and $P_{ef}(cba_k|dh_{nk}{}^{-})$ are the translation probabilities in the two directions.

We now describe the computations of each of the component probabilities.

5.1.1 Word model probability

The word model probability $P_m(cab_k|dh_{nk}{}^{-})$ in Equation 5 is derived from Equations 3 and 1:

There are two cases, depending on whether the hypothesis specifies that there is a boundary after character c_k, in which case we need the probabilities of the two resulting words f', and f'', or there is no boundary, in which case we need the probability of the single word f. (See the initial states in Figures 1 and 2, respectively.)

Let N denote the total number of word tokens in the rest of the corpus $dh_{nk}{}^{-}$, and $N(f)$ denote the number of instances of word f in $dh_{nk}{}^{-}$. The probabilities in the two cases are

$$P_m(c_k^+|dh_{nk}{}^{-}) \propto \frac{N(f') + \alpha P_0(f')}{N + \alpha} \cdot \frac{N(f'') + \alpha P_0(f'')}{N + \alpha}$$

$$P_m(c_k^-|dh_{nk}{}^{-}) \propto \frac{N(f) + \alpha P_0(f)}{N + \alpha}$$

Here $P_0(f)$ is computed using Equation 2.

5.1.2 Translation model probability

The translation model probabilities depend on whether or not there is a segmentation boundary at c_k and which English words are aligned to the relevant Chinese words.

In the first case, assume that there is a word boundary in cab_k, and that English words $\{e'\}$ are aligned to f' and words $\{e''\}$ are aligned to f'' in the E-to-C direction according to the alignment b_k, and that f' is aligned to $e_*{}'$ and f'' is aligned to $e_*{}''$ in the C-to-E direction according to the alignment a_k (see the initial state in Figure 1). Here we overloaded notations and use b_k and a_k to indicate the alignments of the relevant Chinese words at position k to any English words. Let I denote the total number of English words in the sentence, and $J+1$ denote the number of Chinese words according to this segmentation. We also denote the total number of English words aligned to either f' or f'' in the E-to-C direction by P.

The translation model probability in the E-to-C direction is thus:

$$P_{ef}(c_k^+, b_k, a_k|dh_{nk}{}^{-}) \propto \frac{1}{(J+2})^P \prod_{e'} P(e'|f', dh_{nk}{}^{-}) \prod_{e''} P(e''|f'', dh_{nk}{}^{-})$$

Here we compute $P(e|f, dh_{nk}{}^{-})$ as:

$$P(e|f, dh_{nk}{}^{-}) = \frac{N(e,f) + \alpha P_0(e)}{N(f) + \alpha},$$

where the counts are computed over the fixed assignments $dh_{nk}{}^{-}$.

The translation probability in the other direction is similarly computed as:

$$P_{fe}(c_k^+, b_k, a_k|dh_{nk}{}^{-}) \propto \left(\frac{1}{I+1}\right)^2 P(f'|e_*, dh_{nk}{}^{-}) P(f''|e_*, dh_{nk}{}^{-})$$

And $P(f|e, dh_{nk}{}^{-})$ is computed as:

$$P(f|e, dh_{nk}{}^{-}) = \frac{N(f,e) + \alpha P_0(f)}{N(e) + \alpha},$$

where the counts are computed over the fixed assignments $dh_{nk}{}^{-}$.

In the second case, if the hypothesis in evaluation does not have a word boundary at position k, the total number of Chinese words would be one less, i.e. J instead of $J+1$ in the equations above, and there would be a single set of English words aligned to the word f in the E-to-C direction, and a single word e_* aligned to f in the C-to-E direction (see the initial state in Figure 2. The probability of this hypothesis is computed analogously.

5.2 Determining the set of alternative hypotheses

As mentioned earlier, we consider alternative alignments which deviate minimally from the current alignments, and which satisfy the constraints of the IBM model 1 in both directions. In order to describe the set of alternatives, we consider two cases, depending on whether there is a boundary at the current character before sampling at position k.

Case 1. There was no boundary at c_k in the previous state (see Figure 1).

If there is no boundary at c_k, there is a single word f spanning that position. We denote by $\{e\}$ the set of English words aligned to f at that state in the E-to-C direction and by e_* the English word aligned to f in the C-to-E direction. Since every state we consider satisfies the IBM one-to-many constraints, there is exactly one English word aligned to f in the C-to-E direction and the words $\{e\}$ have no other words aligned to them in the E-to-C direction.

In this case, we consider as hypothesis ${cba_k}^-$ the same segmentation and alignment as in the previous state. (see Table 1 for an overview of the alternative hypotheses.)

We consider M different hypotheses which include a boundary at k in this case, where M depends on the number of words $\{e\}$ aligned to f in the previous state. Because we are breaking the word f into two words f' and f'' by placing a boundary at c_k, we need to re-align the words $\{e\}$ to either f' or f''. Additionally we need to align f' and f'' to English words in the C-to-E direction. The number of different hypotheses is equal to 2^P where $P = |\{e\}|$. These alternatives arise by considering that each of the words in $\{e\}$ needs to align to either f' or f'', and there are 2^P combinations of these alignments. For example, if $\{e\} = \{e_1, e_2\}$, after splitting the word f there are four possible alignments, illustrated in Figure 1: I. (f', e_1) and (f'', e_2), II. (f', e_2) and (f'', e_1), III. (f', e_1) and (f', e_2), IV. (f'', e_1) and (f'', e_2). For the alignment a_k in the C-to-E direction, we consider only one option, in which both resulting words f' and f'' align to e_*. These alternatives form ${cba_{k,m}}^+$ in Table 1.

Case 2. There was a boundary at c_k in the previous state (see Figure 2).

In this case, for the hypotheses c_k^+ we consider only one alternative, which is exactly the same as the assignment of segmentation and alignments in the previous state. Thus we have $M = 1$ in Table 1.

Let f' and f'' denote the two words at position k in the previous state, $\{e'\}$ and $\{e''\}$ denote the sets of English words aligned to them in the E-to-C direction, respectively, and ${e_*}'$ and ${e_*}''$ denote the English words aligned to f' and f'' in the C-to-E direction.

We consider only one hypothesis ${cba_k}^-$ where there is no boundary at c_k. In this hypothesis, there is a single word $f = f'f''$ spanning position k, and all words $\{e'\} \cup \{e''\}$ align to f in the E-to-C direction. For the C-to-E direction we consider the "better" of the alignments (f, e'_*) and (f, e''_*) where the better alignment is defined as the one having higher probability according to the C-to-E word translation probabilities.

Table 2: Complete Algorithm of Gibbs Sampler for CWS including Alignment Models.

```
Input: $D$, $F_0$
Output: $A_T$, $F_T$
for $t = 1$ to $T$
    Run GIZA++ on $(D, F_{t-1})$ to obtain $A_t$
    Run GS on $(D, F_{t-1}, A_t)$ to obtain $F_t$
```

5.3 Complete segmentation algorithm

So far, we have described how we re-sample word segmentation and alignments according to our model, starting from an initial segmentation and alignments from GIZA++. Putting these pieces together, the algorithm is summarized in Table 1.

We found that we can further improve performance by repeatedly aligning the corpus using GIZA++, after deriving a new segmentation using our model. The complete algorithm which includes this step is shown in Table 2, where F_t indicates the word segmentation at iteration t and A_t denotes the GIZA++ corpus alignment in both directions. The GS re-segmentation step is done according to the algorithm in Table 1.

Using this algorithm, we obtain a new segmentation of the Chinese data and train the translation models using this segmentation as in the baseline MT system. To segment the test data for translation, we use a unigram model, trained with maximum likelihood estimation off of the final segmentation of the training corpus F_T.

6 Translation Experiments

We performed experiments using our models for CWS on a large and a small data track. We evaluated performance by measuring WER (word error rate), PER (position-independent word error rate), BLEU (Papineni et al., 2002) and TER (translation error rate) (Snover et al., 2006) using multiple references.

6.1 Translation Task: Large Track NIST

We first report the experiments using our monolingual unigram Dirichlet Process model for word segmentation on the NIST machine translation task (NIST, 2005). Because of the computational requirements, we only employed the monolingual word model for this large data track, i.e. the feature weights were $\lambda_1 = 1, \lambda_2 = 0, \lambda_3 = 0$. Therefore, no alignment information needs to be maintained in this case.

The bilingual training corpus is a superset of corpora in the news domain collected from different sources.

We took LDC (LDC, 2003) as a baseline CWS method (Base). As shown in Table 3, the training corpus in each language contains more than two million sentences. There are 56 million Chinese

Table 3: Statistics of corpora in task NIST.

Data	Sents.	Words[K]		Voc.[K]	
		Cn.	En.	Cn.	En.
Chars	2M	56M	49.5M	65.4	211
Base		39.2M		95.7	
GS		40.5M		95.4	
02	878	23.1	28.0	2.04	4.34
03	919	24.6	29.2	2.21	4.91
04	1788	49.8	60.7	2.61	6.71
05	1082	30.8	34.2	2.30	5.39

Table 4: Translation performance [% BLEU] with the baseline(LDC) and GS method on NIST.

MT-eval	LDC(Base)	GS
2005	32.85	33.26
2002	34.32	34.36
2003	33.41	33.75
2004	33.74	34.06

characters. The LDC and GS word segmentation methods generated 39.2 and 40.5 million running words, respectively.

The scaling factors of the translation models described in Section 2.2 were optimized on the development corpus, MT-eval 05 with 1082 sentences. The resulting systems were evaluated on the test corpora MT-eval 02-04. For convenience, we only list the statistics of the first English reference.

Starting from the baseline LDC output as initial word segmentation, we performed Gibbs sampling (GS) of word segmentations using 30 iterations over the Chinese training corpus.

Since BLEU is the official NIST measure of translation performance, we show the translation results measured in BLEU score only. As shown in Table 4, on the development data MT-eval 05, the BLEU score was improved by 0.4% absolute or more than 1% relative using GS. Similarly, the absolute BLEU scores are also improved on all other test sets, in the range of 0.04% to 0.4%.

We can see that even a monolingual semi-supervised word segmentation method can outperform a supervised one in MT, probably because the training/test corpora contain many unknown words and words have different frequencies in our MT data from they do in the manually labeled CWS data.

6.2 Translation Task: Small Track IWSLT

We evaluate our full model, using both monolingual and bilingual information, on the IWSLT data.

As shown in Table 5, the Chinese training corpus was segmented using the unigram segmenter (Base) described in Section 2.1 and our GS method. Since the unigram segmenter performs better in our experiments, we took it as the baseline and the method for initialization in later experiments. We see that the vocabulary size of the Chinese training corpus was reduced more significantly by GS than by the baseline method, even though they resulted in a similar number of running words. This shows that the distribution of Chinese words is more concentrated when using GS.

Table 5: Statistics of corpora in task IWSLT.

Test	Sents.	Words[K]		Voc.	
		Cn.	En.	Cn.	En.
Chars	42.9K	520	420	2780	9930
Base		394		8800	
GS		398		6230	
Dev2	500	3.74	3.82	1004	821
Dev3	506	4.01	3.90	980	820
Eval	489	3.39	3.72	904	810

Table 6: Translation performance with different CWS methods on IWSLT[%].

Test	Method	WER	PER	BLEU	TER
Dev2	Unigram (Base)	38.2	31.2	55.4	37.0
	GS	36.8	30.0	56.6	35.5
Dev3	Unigram (Base)	33.5	27.5	60.4	32.1
	GS	32.3	26.6	61.0	31.4
Eval	Characters	49.3	41.8	35.4	47.5
	LDC	46.2	40.0	39.2	45.0
	ICT	45.9	40.4	40.1	44.9
	Unigram (Base)	46.8	40.2	41.6	45.6
	9-gram	46.9	40.4	40.1	45.4
	GS	45.9	40.0	41.6	44.8

The parameter optimizations were performed on the Dev2 data with 500 sentences, and evaluations were done both on Dev3 and on Eval data, i.e. the evaluation corpus of (IWSLT, 2007).

The model weights λ of GS from Section 5.1.2 were optimized using the Powell (Press et al., 2002) algorithm with respect to the BLEU score. We obtained $\lambda_1 = 1.4$, $\lambda_2 = 1$ and $\lambda_3 = 0.8$ as optimal values and $T = 4$ as the optimal number of iterations of re-alignment with GIZA++.

For a fair comparison, we evaluated on various CWS methods including translation on characters , LDC (LDC, 2003), ICT (Zhang et al., 2003), unigram, 9-gram and GS. Improvements using GS can be seen in Table 6. Under all test sets and evaluation criteria, GS outperforms the baseline method. The absolute WER decreases with 1.2% on Dev3 and with 1.1% on Eval data over baseline.

We compared the translation outputs using GS with the baseline method. On the Eval data, 196 sentences are different out of 489 lines, where 64 sentences from GS are better, 33 sentences are worse, and the rests have similar translation qualities. Table 7 shows two examples from the Eval corpus. We list segmentations produced by the baseline and GS methods, as well as the translations corresponding to these segmentations. The GS method generates better translation results than the baseline method in these cases.

Table 7: Segmentation and translation outputs with baseline and GS methods.

a)	Baseline	有 近路 吗？
		do you have a ?
	GS	有 近 路 吗？
		do you have a shorter way ?
	REF	is there a shorter route ?
b)	Baseline	请 告诉 我 总 金额。
		please show me the in .
	GS	请 告诉 我 总 金 额 。
		please show me the total price .
	REF	can you tell me the total amount ?

7 Conclusion and future work

We showed that it is possible to learn Chinese word boundaries such that the translation performance of Chinese-to-English MT systems is improved.

We presented a Bayesian generative model for parallel Chinese-English sentences which uses word segmentation and alignment as hidden variables, and incorporates both monolingual and bilingual information to derive a segmentation suitable for MT.

Starting with an initial word segmentation, our method learns both new Chinese words and distributions for these words. In a large and a small data environment, our method outperformed the standard Chinese word segmentation approach in terms of the Chinese to English translation quality. In future work, we plan to enrich our monolingual and bilingual models to better represent the true distribution of the data.

8 Acknowledgments

Jia Xu conducted this research during her internship at Microsoft Research. This material is also partly based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-06-C-0023.

References

Aldous, D. 1985. Exchangeability and related topics. In *École d'été de probabilités de Saint-Flour, XIII-1983*, pages 1–198, Springer, Berlin.

Andrew, G. 2006. A hybrid markov/semi-markov conditional random field for sequence segmentation. In *Proceedings of EMNLP*, Sydney, July.

Brown, P. F., S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, June.

Gao, J., M. Li, A. Wu, and C. Huang. 2005. Chinese word segmentation and named entity recognition: A pragmatic approach. *Computational Linguistics*, 31(4).

Goldwater, S., T. L. Griffiths, and M. Johnson. 2006. Contextual dependencies in unsupervised word segmentation. In *Proceedings of Coling/ACL*, Sydney, July.

IWSLT. 2007. International workshop on spoken language translation home page. http://www.slt.atr.jp/IWSLT2007.

Klein, D. and C. D. Manning. 2002. A generative constituent-context model for improved grammar induction. In *Proceedings of ACL*, pages 128–135.

LDC. 2003. Linguistic data consortium Chinese resource home page. http://www.ldc.upenn.edu/Projects/Chinese.

NIST. 2005. Machine translation home page. http://www.nist.gov/speech/tests/mt/index.htm.

Och, F. J. and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of ACL*, pages 295–302, Philadelphia, PA, July.

Papineni, K. A., S. Roukos, T. W., and W. J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318, Philadelphia, July.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. 2002. *Numerical Recipes in C++*. Cambridge University Press, Cambridge, UK.

Snover, M., B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of AMTA*, pages 223–231, Cambridge, MA, August.

Vogel, S., H. Ney, and C. Tillmann. 1996. HMM-based word alignment in statistical translation. In *Proceedings of COLING*.

Xu, J., R. Zens, and H. Ney. 2004. Do we need Chinese word segmentation for statistical machine translation? In *Proceedings of the SIGHAN Workshop on Chinese Language Learning*, pages 122–128, Barcelona, Spain, July.

Xu, J., E. Matusov, R. Zens, and H. Ney. 2005. Integrated Chinese word segmentation in statistical machine translation. In *Proceedings of IWSLT*, pages 141–147, Pittsburgh, PA, October.

Zens, R. and H. Ney. 2004. Improvements in phrase-based statistical machine translation. In *Proceedings of HLT/NAACL*, Boston, MA, May.

Zhang, H., H. Yu, D. Xiong, and Q. Liu. 2003. HHMM-based Chinese lexical analyzer ICTCLAS. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Learning*, pages 184–187, July.