# Character based Chinese-English 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

State-of-the-art machine translation is based on words. However, Chinese sentences are written in the form of a sequence of characters. Usually, word boundaries are detected using an off-the-shelf segmentation method before translation. Thus, optimal words may be lost as the Chinese word segmentation and the translation process are separated.

Therefore, we build a translation system based on Chinese characters directly. Chinese words are identified implicitly in translation models. Word segmentations and alignments are learned simultaneously in the model training. New words are invented using the Dirichlet process, and word distributions are calculated by Gibbs sampling. In search of the best translation, segmentation alternatives are represented as a lattice so that the final decision is integrated into the processing of decoding.

The experimental results show improvements over state-of-the-art Chinese-English translation systems on the GALE and the IWSLT tasks. Moreover, the proposed algorithms can be applied to statistical machine translation from Chinese to any other languages.

## 1 Introduction

In Chinese texts, sentences are written in the form of a sequence of Chinese characters, and words composed of single or multiple characters are not separated by delimiters naturally. This is different from most European languages and poses a challenge in natural language processing tasks, such as machine translation. The conventional way is to segment the Chinese character sequence into Chinese 'words'. Finding proper word boundaries in a sequence of Chinese characters is the so-called *Chinese word segmentation* (CWS) problem.

The Chinese word segmentation performance is usually evaluated by precision and recall, which is calculated based on how well the segmented text matches the reference text. However, we observed that the correlation of word segmentation qualities and translation error rates are not close enough in our experiments, so that we abandoned the previous one and only take the translation error rates as final evaluation criteria to measure the Chinese word segmentation performance. For machine translation, intuitively, the best Chinese words should be the units that provide the best word alignment and lead to the best translation performance.

The common solution has been to segment the Chinese text using an off-the-shelf CWS method and to perform a standard training and translation process once the segmentation is fixed. There are many ways to recognize word boundary. The simplest method is to use the **maximum matching**. Characters in a sequence are checked whether they matched the words in the dictionary from left to right; firstly to the longest words then to the shorter words. An **inverse maximum matching** matches words from right to left. This method is rather naive, because words with equal lengths are treated in the same way. Therefore, statistical methods are introduced to estimate model parameters based on the training data. The advantages of statistical approaches are robustness and generalization. By assigning a probability to each word entry high frequency and low frequency words are distinguished. Under various statistical methods, the **N-gram** based Chinese word segmentation is widely applied, such as in (Chen et al., 2005), (Wang and Liu, 2005), (LDC, 2003). We will present this method in Section 3 in detail and use the unigram word segmentation as one of the baselines in experiments, which requires a manual lexicon containing a list of Chinese words and their frequencies. The lexicon and frequencies are obtained using manually annotated data. **HMM** is another statistical model to perform word segmentation. In (Zhang and Malik, 2003), name entities and so on are taken as classes besides the segmentation lexicon. Bigram class dependency and word conditional probability are taken into account in the HMM based framework to search for optimal segmentation boundaries. A conditional random field (**CRF**) model is employed in (Andrew, 2006) for sequence segmentation.

Nonetheless, significant improvements in translation performance have not yet been shown to result from using these more sophisticated CWS methods, due to the following two reasons: 1. The segmentations may be erroneous, because the context varies. 2. The best segmentation for a given character sequence also depends on its translation. For a destined character sequence the 'correct' segmentation is not universal, but we need to consider the contexts and the language to be translated into. In standard approaches word segmentation is performed previously and independently on the translation system. Segmentation and alignment of words are two separate processes, though they actually influence each other.

The main characteristic of our Chinese word segmentation method is that our segmentation model is designed for and integrated into the machine translation system. We renovate the trivial approach segmenting words in the preprocessing but put the word segmentation into the word alignment training as well as decoding for the best translation. Translation on Chinese characters is feasible therefore as segmentation process is pushed to the translation.

There are two major problems to be solved for Chinese word segmentation: the first one is how to train the word segmentation model; the second one is how to perform the segmentation on a test corpus using the trained model.

For the model training, we will present two different approaches, both employing the bilingual information: the learned segmentation from alignment in Section 4 and the semi-supervised CWS in Section 5. The previous one is easier to implement with an initialization of the alignment between Chinese characters and English words; the latter one is more refined with an initialization of a unigram word segmentation with LDC lexicon. These two methods are parallel. Although we can initialize the latter one with the previous one, we did not apply it for simplicity. In learned segmentation from alignment a monolingual lexicon is extracted from single-best alignments of Chinese characters and English words, where bilingual and context information are employed. However, the single-best alignments may contain errors, thus we further refined the model into an semi-supervised method. Word segmentation and alignment are trained jointly considering both monolingual and bilingual information, in order to derive a segmentation suitable for machine translation, see (Xu et al., 2008). New word entries and their distributions are introduced automatically using Dirichlet process. These machine translation driven word segmentation modeling is expected to learn more adequate word distributions considering the current bilingual training corpus. Improved word distributions in the segmentation model may reduce the unknown words in the test corpus. For instance: '保证金' and '保证' both occur in the LDC lexicon. A unigram segmenter takes '保证金' as one Chinese word in the test corpus. However, no '保证金' but only '保证's appear in the training corpus. Therefore '保证金' in the test corpus is considered as an unknown word that cannot be translated. In contrast, the learned segmentation and the semi-supervised segmentation will leave out the word '保证金' from the automatic lexicon because it does not appear in the training corpus. In this way, '保证金' that are separated into two words '保证' and '金' can be translated. In addition to the unknown word problem, the learned segmentation and semi-supervised CWS also solve the problem of word segmentation ambiguities better, because the estimation of word entries and distributions are adapted to the bilingual training corpus. Our experiments on both large (GALE) and small (IWSLT) data tracks show improvements over the state-of-the-art machine translation systems with respect to translation performance.

Once the segmentation model, more precisely the lexicon is obtained, we need to solve the second problem, finding best Chinese word segmentation on a test corpus. We can use either a unigram word segmenter (as standard approach) or the word segmentation lattice in Section 6 to perform Chinese word segmentation on the test corpus. With the lattice segmentation, translation is carried out on character level in decoding, and multiple segmentations are represented as a lattice so that segmentation decisions are integrated into the search for the best translation. Similar approaches were applied in speech translation as described in (Ney, 1999), where speech recognition and text translation are combined by using the recognition lattices. We also weight the different segmentations with a language model trained on the Chinese corpus at the word level. Weighting the word segmentation by language model cost was introduced in (Luo and Roukos, 1996). Our experiments on Chinese-to-English translation show that our method improves the performance of a state-of-the-art machine translation system.

In the following context, we will first introduce the definition of Chinese word segmentation in Section 2 along with the notations for later sections. Then we will describe a standard $n$-gram segmentation method in Section 3. As major contents of this work, the learned segmentation from alignment, the semi-supervised CWS and the integrated CWS are described in Section 4, 5 and 6 respectively. Finally, translation experiments are shown in Section 7.

## 2 Definition

In statistical machine translation we are given a Chinese sentence in characters $c_1^K = c_1 c_2 ... c_k ... c_K$ ($k \in 1, 2, .., K$), which is to be translated into an English sentence $e_1^I = e_1 e_2 ... e_i ... e_I$ ($i \in 1, 2, .., I$), where $K$ and $I$ is the length of the Chinese sentence in characters and the English sentence in words, respectively.

In order to obtain a more adequate mapping between Chinese and English translation units, $c_1^K$ is usually segmented into words. The positions of *Chinese word boundaries* on a character sequence $c_1^K$ is indicated with $k_0 \equiv 0$ and $k_1^J = k_1 k_2 ... k_j ... k_J$ ($j \in 1, 2, .., J$), where $k_J \equiv K$, $k_j \in \{1, 2, ...K\}$ and $k_{j-1} < k_j$, and $J$ is the number of Chinese words for $c_1^K$. The $k_1^J$ store the information where the Chinese words are delimited in a sentence. We use $k_j$ to indicate the $j$-th word segmentation boundary is placed after (on the right side of ) the Chinese character $c_{k_j}$ and before character $c_{k_j+1}$, where $1 \leq j < J$. $k_0$ is a boundary on the left side of the first Chinese character $c_1$, which is defined as 0, and $k_J$ is always placed after the last Chinese character $c_K$ and therefore equals $K$.

Given a sequence of Chinese characters $c_1^K$ and its segmentation boundaries $k_1^J$, a sentence can also be represented in the form of a sequence of Chinese words $f_1^J = f_1 f_2 ... f_j ... f_J$ ($j \in 1, 2, .., J$), and each individual Chinese word $f_j$ is defined as

$$f_j = c_{k_{j-1}+1} .. c_{k_j} = c_{k_{j-1}+1}^{k_j} \tag{1}$$

$f_1^J$ is composed of two sources of information, the character sequence and its word segmentation.

An example is illustrated in Table 1. The sentence 小孩玩纸牌 contains five Chinese characters ($K = 5$), where $c_1$ denotes 小, $c_2$ denotes 孩, etc. The first word segmentation boundary is placed after the second Chinese character ($k_1 = 2$), the second boundary is after the third character ($k_2 = 3$) and the third boundary is after the fifth character ($k_3 = 5$). That means 小($c_1$) and 孩($c_2$) together compose the word 小孩($f_1$), 玩($f_2$) is a single character word, and 纸牌($c_4 c_5$) is the third word ($f_3$) of this sentence.

## 3 $n$-gram segmentation

The simplest and widely applied automatic segmentation tool is based on a *unigram segmentation*, which requires a lexicon containing a list of Chinese words and their frequencies, as shown in Table 2. Such a lexicon can be obtained using manually annotated data, e.g. the LDC lexicon (LDC, 2003) or generated automatically as described later in Section 4 and Section 5. We need to maximize the probability of a sentence considering all word segmentation alternatives. Assuming each Chinese word in the sentence is distributed independently, we are interested in knowing how to put the word delimiters properly so that the product of probabilities of all words is maximized:

$$\begin{aligned} \hat{k}_1^{\hat{J}}(c_1^K) &= \underset{k_1^J, J}{\operatorname{argmax}} \prod_{j=1}^{J} Pr(f_j = c_{k_{j-1}+1}^{k_j}) \\ &= \underset{k_1^J, J}{\operatorname{argmax}} \prod_{j=1}^{J} Pr(c_{k_{j-1}+1}^{k_j}) \end{aligned} \tag{2}$$

$Pr(f_j = c_{k_{j-1}+1}^{k_j})$ is the probability of a word $f_j$ which is a sequence of characters $c_{k_{j-1}+1}^{k_j}$ with a boundary after the $k_{j-1}$-th character and before the $k_j$-th character in the sentence. Taking the word dependency into account we extend Equation 2 into Equation 3, if we use the concept of an n-gram language model, where $n$ is the order of the n-gram model:

$$\begin{aligned} &\hat{k}_1^{\hat{J}}(c_1^K) \\ &= \underset{k_1^J, J}{\operatorname{argmax}} \prod_{j=1}^{J} Pr(c_{k_{j-1}+1}^{k_j} | c_{k_{j-2-n}+1}^{k_{j-1-n}}, .., c_{k_{j-2}+1}^{k_{j-1}}) \end{aligned} \tag{3}$$

Table 2: Manually generated Chinese word lexicon.

| Word | 小 | 孩 | 玩 | 纸 | 牌 | 小孩 | ... |
|---|---|---|---|---|---|---|---|
| Frequency | 3465 | 22 | 588 | 146 | 361 | 367 | ... |

The dynamic programming algorithm is used to maximize the product of the relative frequencies of all words in one sentence. The segmenter finds the path which has the highest product of word probabilities.

Another instance of this type of segmenter is the LDC tool, which is also based on unigram segmentation but with additional text normalizations. The LDC segmenter finds the path which has the highest product of word probability and the next word is selected from the longest phrase. More details can be found on the LDC web site (LDC, 2003).

The unigram Chinese word segmentation method is so far the most commonly applied method in machine translation, but it has several problems: First, maximizing the product of single word probabilities does not guarantee that the context information is taken into account; hence, the

Table 1: An example for the definition of Chinese words and word segmentations.

| | | | | | |
|---|---|---|---|---|---|
| Characters | 小 | 孩 | 玩 | 纸 | 牌 |
| Glosses of characters | small | child/children | play | paper | card/cards |
| $c_1^K$ | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ |
| Segmentation boundaries | 小 | 孩 \| | 玩 \| | 纸 | 牌 \| |
| $k_1^J$ | | $k_1 = 2$ | $k_2 = 3$ | | $k_3 = 5$ |
| Characters into words | | $c_1c_2$ | $c_3$ | | $c_4c_5$ |
| Words | | 小孩 | 玩 | | 纸牌 |
| Glosses of words | | child/children | play | | card/cards |
| $f_1^J$ | | $f_1$ | $f_2$ | | $f_3$ |

segmentation may contain errors. Second, a more accurate word segmentation does not always lead to a great improvement in translation performance. The "correct" segmentation for one character sequence is not universal but depends on the Chinese context and the destination language. This method is sub-optimal for MT. For example, 纸(paper) and 牌(card) can be separated 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.

## 4 Alignment derived segmentation

Table 3: An example of an alignment matrix between Chinese characters and English words. A gray box indicates a single-best (Viterbi) word alignment.

| | 小 | 孩 | 玩 | 纸 | 牌 |
|---|---|---|---|---|---|
| cards | • | • | • | ■ | ■ |
| play | • | • | ■ | • | • |
| Children | ■ | ■ | • | • | • |

We will introduce the first word segmentation approach, namely learned segmentation. A similar approach is presented in (Ma et al., 2007). They pack several consecutive words together which are considered corresponding to a single word in the opposite language and then bootstrap word alignment training using the modified segmentation of the sentences. We start from characters instead of from words so that the initially incorrect word segmentations can be avoided. As described in (Xu et al., 2004), the consecutive Chinese characters aligning to the same English word are concatenated into one word for a bilingual information derived word segmentation.

We will describe the algorithm in (Xu et al., 2004) in this section. In statistical machine translation we have a bilingual corpus available. This corpus is used to obtain a segmentation of the Chinese part in the following way: First, we train the statistical translation models with the bilingual corpus. There is no word segmentation performed on Chinese texts, and each Chinese character is interpreted a word. As a result of this word alignment training, we obtain for each sentence pair a mapping of Chinese characters to the corresponding English words i.e. the single-best word alignment between Chinese characters and English words. Such an alignment is represented as a binary matrix with $K \cdot I$ elements. An example is shown in Table 3, where a Chinese training sentence in characters is plotted along the horizontal axes and its English translation sentence in words along the vertical axes. The black boxes show the best alignment for this sentence pair after word alignment training. In this example the first two Chinese tokens are aligned to "Children", the next one is aligned to "play", and the last two tokens are aligned to "cards".

Based on this information, we can generate a Chinese word list with each entry composed of one or more Chinese characters, which are aligned to one English word in the word alignment matrix. In the experiments we train the alignments in both directions with GIZA++ tool and combine them. The methods of alignment combination are described in (Och, 2002). Chinese word entries are extracted based on this combined alignment. Lexicon entries learned from the alignment matrix in Table 3 are '小孩', '玩' and '纸牌'; each of them has a frequency of one. With the help of this self-learned lexicon we use a segmentation tool, such as a unigram segmenter in Section 3 to obtain a segmented Chinese text. If we calculate the frequencies for every word, the distribution can be obtained, too. As in Table 3, in the upper line are the words and in the lower line are the word frequencies. Finally, we retrain the translation system with the segmented corpus.

This lexicon shows the distribution of Chinese words in the training corpus. The extraction method differs from other self-learned methods because it uses the bilingual training corpus instead of the monolingual corpus such as in (Sproat and Shih, 1990). Since we are more interested

in the relationship between the languages, this method is more suitable for the machine translation application.

The central idea of the proposed lexicon learning method is: several Chinese characters constitute a Chinese word, if they are aligned to the same English word. Using this idea and the bilingual corpus, we can generate a Chinese lexicon automatically. As a conclusion, the 'learned translation with learned segmentation' consists of three steps:

1. The input is a sequence of Chinese characters without segmentation. After the training using GIZA++ , we extract a monolingual Chinese dictionary from the alignment.

2. Using this learned dictionary we segment the sequence of Chinese characters into words. In other words, the unigram method is used, but the manual lexicon is replaced by the learned lexicon.

3. Based on this word segmentation, we perform another training using GIZA++ . Then, after training the models IBM model 1, HMM and IBM model 4, we extract bilingual word groups, which are referred to as phrase-based translation.

# 5 Semi-supervised Chinese word segmentation

The learned segmentation from alignment models the word distributions in the lexicon using word alignment information. But the erroneous alignment can result in the incorrect word segmentation, which may lead to sub-optimal translation results. Therefore, we further propose a refined Chinese word segmentation model that learns both Chinese word entries and their distributions to generate a dynamic lexicon. New words are introduced with a prior distribution using Bayesian learning. Chinese word segmentation and word alignment, which have an influence and effect on each other, are trained simultaneously.

This method is semi-supervised, namely, Chinese word segmentation is jointly trained with word alignment given an initialized word segmentation and alignment. We employ linear interpolation to introduce new words to the lexicon with a prior distribution. We describe a generative model which consists of a word model and two alignment models, representing the monolingual and bilingual information respectively. We first segment the Chinese text using a unigram segmenter and then learn new word types and word distributions, which are suitable for MT.

In the following text, we will first introduce the semi-supervised word segmentation approaches in Section 5.1, along with its main characteristics that are different from the other approaches. Then we will describe the generative model in Section 5.2 in detail, which is the central idea of this approach. Furthermore, in Section **??** we will extend the generative model to a final model, similar to a maximum entropy model in which most features are derived from the sub-models of the generative model. Finally, we will discuss the training procedures using Gibbs sampling algorithm and the realignment alternatives in Section 5.3, 5.5 and 5.6.

## 5.1 Approaches

The training and translation processes for semi-supervised Chinese word segmentation are illustrated in Figure 1. The inputs to the system are the bilingual training data, including Chinese sentences in characters and its English translations in words, a manual Chinese word lexicon, such as LDC lexicon, as well as a test corpus on the character level. First, we segment the Chinese training corpus in characters with a unigram word segmentation using the manual lexicon, then the word alignment and Chinese word segmentation are jointly trained using Gibbs sampling. After that, by counting the Chinese word frequencies of the training corpus, we easily obtain a Gibbs sampling (GS) generated lexicon, in the same format of the manual lexicon as shown in Table 2. We take the relative frequencies after normalization as probabilities for word entries in the GS lexicon and the manual lexicon, then we linearly interpolate the probability in the GS lexicon with the probability in the manual lexicon of each word entry. The weight of probabilities in the manual lexicon is defined by user. The weight of probabilities in the manual lexicon and the weight of probabilities in the GS lexicon sum to one. The combined lexicon with the GS and manual lexicons is applied to perform a unigram segmentation on the test corpus. The Chinese word segmentation on the test corpus is another output from the proposed segmentation system.

We apply Gibbs sampling for model training. The Gibbs sampling algorithm generates an instance from the distribution of each variable in turn, depending on the current values of the other variables. This characteristic is particular interesting if the categories are unknown and to be learned. In this chapter the categories are the Chinese words, namely word entries themselves. There are approximately 90K Chinese characters and 7000 are commonly in use; any of these characters can be an element of a word. Usually, a Chinese word is composed of one to four characters. The number of Chinese words is calculated as $7000 + 7000^2 + 7000^3 + 7000^4$, which is difficult to be fixed before translation. If there is no previously defined lexicon, the number of all possible segmentations for a sequence of characters $c_1^K$ is

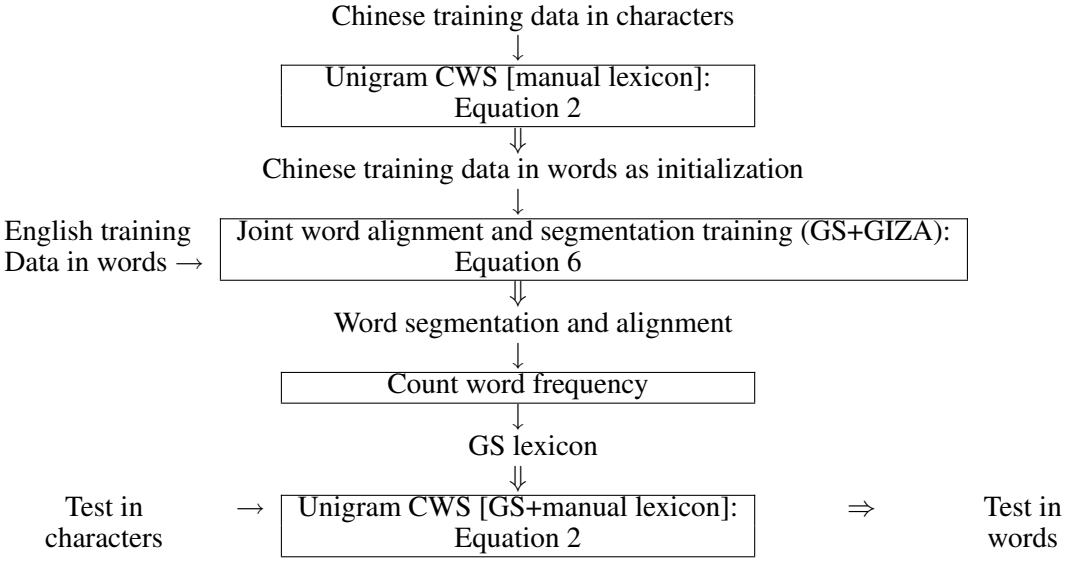


Figure 1: Workflow of semi-supervised CWS. Model training is performed using Gibbs sampling (GS).

$2^{K-1}$. That means that the complexity is exponential in the order of character sequence length. Therefore, we have to approximate the space of all possible derivations in some way: we can define that a word contains at most four characters. With such a constraint, the complexity becomes polynomial, but it is still a high order polynomial. We can perform a standard beam search to prune low cost paths. As an alternative to draw a space with all segmentation derivations for one sentence, we apply the Gibbs sampling algorithm, which learns each parameter value depending on all other parameter values in turn. It is shown in (Blunsom and Osborne, 2008) that in the decoding the search spaces produced by the sampling approach occupied roughly half the disc space as those produced by the beam search with similar results.

### 5.2 Generative and log-linear model

We apply a generative classifier to learn a model of the joint probability of observations. As shown in Table 4, the generative model assumes 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.

The joint probability of the observations $({c_1}^K, e_1^I)$ can be obtained by summing up all possible values of the hidden variables $k_1^J$ and $b_1^I$. The model probability $Pr(c_1^K, e_1^I)$ can be seen as the sum of all possible Chinese word segmentations $k_1^J$ of the character sequence $c_1^K$:

$$Pr(c_1^K, e_1^I) = \sum_{k_1^J} \sum_{b_1^I} Pr(c_1^K, e_1^I, k_1^J, b_1^I) \quad (4)$$

Given a sequence of Chinese characters and its word segmentation boundaries, the corresponding sequence of Chinese words is determined, and vice versa. Given a sequence of Chinese words, we can determine its sequence of Chinese characters and its word segmentation boundaries. Therefore, we use $f_1^J$ to represent the information of $c_1^K$ and $k_1^J$. 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:

$$Pr(e_1^I, f_1^J, b_1^I) = Pr(f_1^J) Pr(e_1^I, b_1^I | f_1^J) \quad (5)$$

This suggests a monolingual Chinese sentence model $Pr(f_1^J)$ and a bilingual translation model $Pr(e_1^I, b_1^I | f_1^J)$. In the following paragraphs we will describe the modeling assumptions behind the monolingual Chinese sentence model and the translation model, respectively.

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 optimize the weights of the sub-models on a development set by maximizing the BLEU score of the final translation. We used three features derived from Equation 7 and equations in Section 5.2.3.

Table 4: Observations and hidden variables of the generative model for Chinese word segmentation.

| | Symbol | Abb. | Example |
|---|---|---|---|
| Observations | | | |
| Chinese sequence in characters | $c_1^K$ | C | 小 孩 玩 纸 牌 |
| English sentence | $e_1^I$ | E | Children play cards |
| Hidden variables | | | |
| Alignment normal | $a_1^J$ | A | e.g. cards → 牌 |
| Alignment inverse | $b_1^I$ | B | e.g. 牌→ cards |
| Segmentation (Chinese sequence in words) | $k_1^J$ ($f_1^J$) | ($F$) | e.g. 小孩 玩 纸牌 |

The maximum entropy model can be viewed as a weighted linear combination of the log probabilities of the sub-models. The weights that are optimized on development datasets have empirical justifications. Since different sub-models have been trained on different datasets, their dynamic value ranges can be so different that it is inappropriate to combine their log probabilities through simple addition. Moreover, for instance, some models may be poorly estimated due to the lack of a large amount of training data. Therefore, empirical results have demonstrated that the use of scaling factors that reflect the relative contribution of different sub-models often improves the performance. Similar approaches have been used very successfully before, for example in the IBM models 3–6 (Och, 1999). The final model applied in the experiments is

$$\begin{aligned} & Pr(c_1^K, e_1^I, f_1^J, a_1^J, b_1^I) \qquad (6) \\ \approx\ & \frac{1}{Z} Pr(f_1^J)^{\lambda_1} \cdot Pr(e_1^I, b_1^I | f_1^J)^{\lambda_2} \\ & \cdot Pr(f_1^J, a_1^J | e_1^I)^{\lambda_3}, \end{aligned}$$

where $a$ is the alignment for Chinese-to-English translation, and $b$ is the alignment for English-to-Chinese translation.

#### 5.2.1 Monolingual Chinese sentence model

We use the unigram model to estimate the sentence probability using monolingual information where words are generated independently. The probability of a sequence of Chinese words in a sentence is thus:

$$Pr(f_1^J) \ = \ \prod_{j=1}^{J} P_G(f_j), \qquad (7)$$

where $P_G(f_j)$ is further explained by the word generation model in the following section.

#### 5.2.2 Word generation by Bayesian learning

The conventional Chinese word segmentation approach applies a manual lexicon containing fixed Chinese words and their frequencies as distributions. Different from a standard method, the semi-supervised segmentation model can introduce new Chinese word types and learn word distributions automatically from unlabeled data.

We never estimate a word distribution explicitly but instead integrate over its possible values and perform a Bayesian inference. It is easy to compute the probability of a Chinese word given a set of already generated words, while integrating over word distributions. Each word in a Chinese corpus is generated using linear interpolation:

$$P_G(f) = (1 - \alpha_1)\frac{N(f)}{N} + \alpha_1 P_0(f) \qquad (8)$$

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 model with linear interpolation in Equation 8 is equivalent to a word model based on Chinese restaurant process, such as the word model in (Goldwater et al., 2006). Each random variable $f$ is drawn independently and identically from G , where G is a distribution over words drawn from a prior with base measure $P_0$ and a concentration parameter $\alpha_1$.

$N(f)$ is the number of Chinese words $f$ in the previous context: In the first training iteration, $N(f)$ is the frequency of word $f$ appearing from the beginning of the text to the current position; after the first iteration, it is the frequency of word $f$ in the text counted in the last iteration. the $N$ is the total number of Chinese words, $P_0$ is the base probability over words, and $\alpha_1$ influences the probability of introducing a new word at each step and controls the size of the lexicon. The probability of generating a word increases as more instances of that word are seen. $\alpha_1$ controls the number of word types, i.e. size of the lexicon. It is the

total probability to generate any new words. $P_0$ defines a probability distribution over new words i.e. how likely a sequence of Chinese characters forms a word.

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

$$P_0(f) = \sum_L P(L)P(f|L), \quad (9)$$

where $L$ is the number of Chinese characters of word $f$. We decompose the spelling model into a word length model $P(L)$ and a word model depending on its length $P(f|L)$. The length model follows a Poisson distribution:

$$P(L) = \frac{\kappa^L}{L!}e^{-\kappa}, \quad (10)$$

where $\sum_L P(L) = \sum_L \frac{\kappa^L}{L!}e^{-\kappa} = 1$ according to Poisson.

The word model is defined based on two cases: a uniform distribution on words with a length equals to a given length and zero otherwise. The probability of a word $f$ given a word length $L$ is defined as

$$P(f|L) = \begin{cases} (\frac{1}{|c|})^L : |f| = L \\ 0 \quad\quad : |f| \neq L \end{cases} \quad (11)$$

$|f|$ is the length of the word $f$, and $|c|$ is the character vocabulary size i.e. the number of different characters in the document. The normalization constraint is proved for Equation 11 as follows: $\sum_f P(f|L) = |c|^L \cdot (\frac{1}{|c|})^L + 0 = 1$, because there are $|c|^L$ possible words with a length of $L$ and each has a probability of $(\frac{1}{|c|})^L$, the other words with a length not equals to $L$ have probabilities of 0.

#### 5.2.3 Translation model

We employ the inverse IBM model 1 to generate English words and alignments given the Chinese words. In For every Chinese word $f$ (including the *null* word), a distribution over English words $G_f$ is first drawn from a prior $P_0(e)$, which is estimated by 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

$$P(e_1^I, b_1^I|f_1^J) = \prod_{i=1}^{I} \frac{1}{J+1} P_{G_{f_{b_i}}}(e_i|f_{b_i}), \quad (12)$$

where $e_i$ is distributed according to $G_{f_{b_i}}$. This is the same model form as the inverse IBM model 1. We place priors on the Chinese-word specific distributions over English words. [1]

In practice, we observed that using a word-alignment model in one direction is not sufficient. We then added a factor to the final model which includes word alignment in the other direction. Such combinations of models in both directions are widely used for phrase extraction such as discussed in (Och and Ney, 2004).

Therefore, we also used a translation model in the other direction, the IBM model 1. We ignore the detailed description here, because the calculation is the same as that of the inverse IBM model 1. For every English word $e$ (including the *null* word), a distribution over Chinese words $G_e$ is first drawn from a prior $P_G(f)$ derived from Equation 8. 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:

$$P(f_1^J, a_1^J|e_1^I) = \prod_{j=1}^{J} \frac{1}{I+1} P_{G_{e_{a_j}}}(f_j|e_{a_j}) \quad (13)$$

### 5.3 Gibbs sampling training

Using the log-linear model we would like to choose the most likely word segmentation given the observed pairs of Chinese-English sentences.It is generally impossible to find the most likely segmentation according to the proposed 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 produces 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 Markov chain result from sampling each component of the state conditioned on the current value of all other variables.

Gibbs sampling is an iterative procedure that samples variables given the current values of all other variables. The Gibbs sampler for Chinese word segmentation works as follows: For each step, we take a single possible boundary point by fixing other segmentations and alignments, then we compare hypotheses considering this boundary and the related alignments. After sampling by using the posterior probabilities of each candidate,

[1] $f_{b_i}$ 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 $e_{a_j}$.

we choose one of these candidates and perform the same operation for the next position. To perform Gibbs sampling we start with an initial word segmentation and with initial word alignments. We re-sample iteratively the word segmentation and alignments according to Equation 6. For instance, we are interested in determining the word boundary after 纸 in the sentence '小孩玩纸牌'. We only show the example with the monolingual model for convenience. We suppose that 纸 牌 are two words from the initialization. $N$ is the number of words in Chinese corpus. First, we decrease the related counts $N$, $N$(纸), $N$(牌), $N$(纸, Children), .. with one. After that, we calculate the probabilities $P$(纸 牌|..), $P$(纸牌|..), .. again. Now, we compare $P$(纸 牌|..) and $P$(纸牌|..) using sampling i.e. after the normalization on the probabilities so that $P'$(纸 牌|..) and $P'$(纸牌|..) sum to one, we select a random number between zero and one, if this random number is smaller than $P'$(纸牌|..), we choose 纸牌, otherwise, we choose 纸牌. That means that a higher probability segmentation is more likely to be chosen. Finally, we increase the associate counts of the chosen segmentation $N$, $N$(纸牌), $N$(纸牌, Children), $N$(纸牌, play), $N$(纸牌, card),.. with one. This is an iterative process going over all positions in a document until the segmentation results converge.

### 5.4 Determining the set of alternatives

Table 5: General algorithm of Gibbs sampling for CWS.

| |
|---|
| Input: initial segmentation and alignments |
| Output: sampled segmentation and alignments |
| for each sentence $s = 1$ to $S$ |
| for each character position $k = 1$ to $K$ that $c_k \in d_s$ |
| Create $\|e\| + 1$ candidates, $cba^+_{k,e_}$ and $cba^-_k$ |
| $cba^+_{k,e_}$: there is a word boundary after $c_k$ |
| $cba^-_k$: there is no word boundary after $c_k$ |
| Compute probabilities of alternatives |
| as illustrated in Figure 2, 3, 4 and 5 |
| Sample boundary and relevant alignments |
| Update counts |

Sampling on word segmentation can change the Chinese word and its alignment. For reasons of efficiency, we only allow alternative alignments which deviate minimally from the current alignments for comparing different word segmentations and not for large-scale modification of the word alignment. Thus, we only use models derived from IBM model 1 satisfying the one-to-many constraints in both directions.

Hence, we organize the sampling process around possible word boundaries. We consider a set of alternatives for the boundary after each character and relevant alignment links at each step in the Gibbs sampler keeping all other hidden variables fixed. We need to compute the probability of each of the alternatives at each step given the fixed values of the other hidden variables.

We introduce some notation to make the presentation easier. The observations are $D = (d_1, ..d_s, .., d_S)$, where $d_s$=$(c_1^K, e_1^I)$ 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 $b_1^I$. For each character $c_k$ in each sentence, we consider two alternative segmentations: ${c_k}^+$ indicates the segmentation where a boundary exists after $c_k$, and ${c_k}^-$ indicates the segmentation where no boundary exists after $c_k$ keeping all other boundaries fixed. Let $f$ denote the single word spanning character $c_k$ if 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 $2^{|e|}$ new possible alignments in the E-to-C direction $b^+_{k1}, \ldots, b^+_{k|e|}$, such as in Figure 2. $|e|$ is defined as the total number of English words aligned to $f'$ or $f''$, previously to $f$. Due to reasons of brevity, we denote these alternatives by ${cba_{k,e_}}^+$ and ${cba_k}^-$, where $e_$ is the $e_$-th candidate after re-alignment.

For every position $k$ in sentence pair $s$, we denote by $dh_{sk}$ the observations and hidden variables for all other sentences than sentence $s$, and the observations and hidden variables inside sentence $s$, 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. ${dh_{sk}}^+$ denotes that there is a word boundary after $c_k$ in sentence $s$, and ${dh_{sk}}^-$ denotes that there is no word boundary after $c_k$ in sentence $s$ given the observations.

Together with the boundary versus no-boundary state at each character position, we re-sample a set of alignment links between English words and any of the Chinese words $f$ or $f'$ and $f''$ keeping all other word alignments in the sentence pair fixed. 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 $c_k$ previously (see Figure 2 and 3).

If there was 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 $f$ in the E-to-C direction and by $e_*$ the English word aligned to $f$ in the C-to-E direction in this case. Due to

the fact that we consider the IBM model 1 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 previous. (see Table 5 for an overview of the alternative hypotheses.)

We consider $2^{|e|}$ different hypotheses which include a boundary at $k$ in this case, where $2^{|e|}$ depends on the number of words aligned to $f$ previously. As 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. These alternatives arise by considering that each of the words in $\{e\}$ needs to align to either $f'$ or $f''$, and there are $2^{|e|}$ combinations of these alignments. For example, if $\{e\} = \{e_\prime, e_{\prime\prime}\}$ as in Figure 2, after splitting the word $f$ there are four possible alignments illustrated in Figure 2: I. $(f', e_\prime)$ and $(f'', e_{\prime\prime})$, II. $(f', e_{\prime\prime})$ and $(f'', e_\prime)$, III. $(f', e_\prime)$ and $(f', e_\prime)$, IV. $(f'', e_{\prime\prime})$ and $(f'', e_{\prime\prime})$. For the alignment $a_k$ in the C-to-E direction, we only consider one option, in which both resulting words $f'$ and $f''$ align to $e_*$. These alternatives form ${cba_{k,e_}}^+$ in Table 5.

Case 2. There was a boundary at $c_k$ previously (see Figure 4 and 5).

In this case, for the hypothesis $c_k^+$ we only consider one alternative, which is exactly the same as the assignment of segmentation and alignments as previous.

Let $f'$ and $f''$ denote the two words at position $k$ previously. As in Figure 4, $e'$ and $e''$ denote the English words aligned to them in the E-to-C direction, respectively, and as in Figure 5, ${e'}_*$ and ${e''}_*$ denote the English words aligned to $f'$ and $f''$ in the C-to-E direction.

We only consider one hypothesis ${cba_k}^-$ where there is no boundary at $c_k$. There is a single word $f$ spanning position $k$ in this hypothesis, and all words align to $f$ in the E-to-C direction. For the C-to-E direction we approximately consider the 'better' one 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.

### 5.5 Computing probabilities of alternatives

For the Gibbs sampling algorithm in Table 5, we need to compute the probability of each alternative segmentation/alignments given the fixed values of the rest of the data $dh_{sk}$. 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.

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 the proposed model is

$$P(cba_k|dh_{sk}) \propto P_m(cba_k|dh_{sk})^{\lambda_1} \cdot P_{ef}(cba_k|dh_{sk})^{\lambda_2} \cdot P_{fe}(cba_k|dh_{sk})^{\lambda_3}, \quad (14)$$

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

Now we describe the computations of each of the component probabilities.

#### 5.5.1 Word model probability

The word model probability $P_m(cab_k|dh_{sk})$ in Equation 14 is derived from Equation 7. There are two cases: If the hypothesis specifies that there is a boundary after character $c_k$, we need the probabilities of the two resulting words $f'$, and $f''$; otherwise, we need the probability of the single word $f$.

Let $N$ denote the total number of word tokens in the rest of the corpus $dh_{sk}$, and $N(f)$ denote the number of instances of word $f$ in $dh_{sk}$. The probabilities in the two cases $P_m(cab_k^+|dh_{sk})$ and $P_m(cab_k^-|dh_{sk})$ are computed using $P_G(f') \cdot P_G(f'')$ and $P_G(f)$ respectively.

#### 5.5.2 Translation model probability

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

In the first case, we assume that there is a word boundary in $cab_k$, but previously there was no word boundary in $cab_k$, and we assume 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 $e_*$ or $f''$ is aligned to $e_*$ in the C-to-E direction according to the alignment $a_k$ (see Figure 4 and 5). Here, we overload 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 $J$ denote the number of Chinese words according to this segmentation. We consider the *null* words to calculate IBM model 1 in both directions. We also denote the total number of English words aligned to either $f'$ or $f''$ in the E-to-C direction by $|e|$.

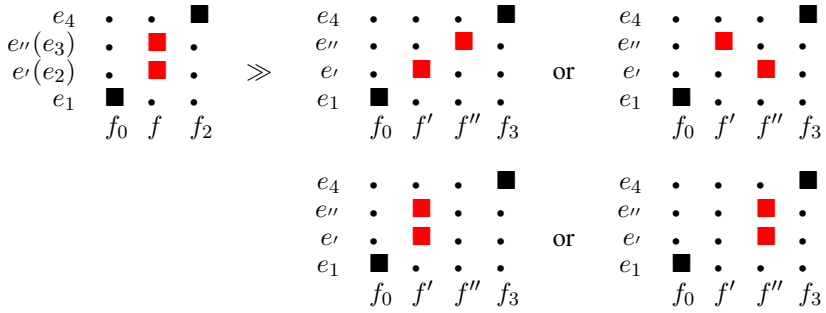


Figure 2: Transition from a no-boundary($-$) to a boundary($+$): The monolingual probability $P_m(cba_k^+|dh_{sk}{}^-)$ is estimated as $P_G(f')P_G(f'')$, and the translation probability in the E-to-C direction $P_{ef}(cba_k^+|dh_{sk}{}^-)$ is estimated as $\frac{1}{J+2}{}^{|e|}P(e_{\prime}|f')P(e_{\prime\prime}|f'')$ or $\frac{1}{J+2}{}^{|e|}P(e_{\prime\prime}|f')P(e_{\prime}|f'')$ or $\frac{1}{J+2}{}^{|e|}P(e_{\prime}|f')P(e_{\prime}|f'')$ or $\frac{1}{J+2}{}^{|e|}P(e_{\prime\prime}|f')P(e_{\prime\prime}|f'')$, here $|e|=2$ and $J=2$.

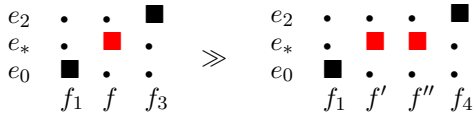


Figure 3: Transition from a no-boundary($-$) to a boundary($+$): The monolingual probability $P_m(cba_k^+|dh_{sk}{}^-)$ is estimated as $P_G(f')P_G(f'')$, and the translation probability in the C-to-E direction $P_{fe}(cba_k^+|dh_{sk}{}^-)$ is estimated as $\frac{1}{I+1}{}^2P_G(f'|e_*)P_G(f''|e_*)$, here $|e|=2$ and $I=2$.

The translation model probability $P_{ef}(cba_k^+|dh_{sk}{}^-)$ from no word boundary to a word boundary in the E-to-C direction if $f'$ aligns to $e'$ and $f''$ aligns to $e''$ as in Figure 2 is thus

$$\left(\frac{1}{J+2}\right)^{|e|} P_G(e'|f')P_G(e''|f'').$$

The translation model probabilities with other re-alignments in Figure 2 are calculated in the same way. Here we compute $P_G(e'|f')$ and $P_G(e''|f'')$ as

$$P_G(e|f) \quad = \quad (1-\alpha_3)\frac{N(e,f)}{N(f)} + \alpha_3 P(e),$$

where the counts are computed over the fixed assignments $dh_{nk}{}^-$, and $P(e)$ is the relative frequency among all English words in the corpus.

The translation probability in the other direction $P_{fe}(cba_k^+|dh_{sk}{}^-)$ is similarly computed as

$$\left(\frac{1}{I+1}\right)^2 P_G(f'|e_*)P_G(f''|e_*),$$

where $P_G(f'|e_*)$ and $P_G(f''|e_*)$ are computed as

$$P_G(f|e) \quad = \quad (1-\alpha_2)\frac{N(f,e)}{N(e)} + \alpha_2 P_G(f).$$

In the second case, if the hypothesis in the evaluation does not have a word boundary at position $k$, the total number of Chinese words would be one less i.e. $J$ instead 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 (see Figure 4), and a single word $e'_*$ or $e''_*$ aligned to $f$ in the C-to-E direction (see Figure 5). The probability of this hypothesis is computed analogously.

The parameters $\theta$ are estimated on-the-fly, which means that updating $\theta$ indicates updating the counts $N(f,e)$, $N$, $N(e)$ and $N(f)$ according to the proposed model. The probabilities are computed when it is called in the sampling.

### 5.6 Complete segmentation algorithm

So far, we have described how we re-sample word segmentation and alignments, starting from an initial segmentation and alignments from GIZA++. Putting these pieces together, we get the algorithm that is summarized in Table 5.

We further improve performance by repeatedly aligning the corpus using GIZA++ for a more adequate re-alignment. We do so after deriving a new segmentation. The complete algorithm which includes this step is shown in Table 6, where $F_t$ indicates the word segmentation at iteration $t$ and $A_t$

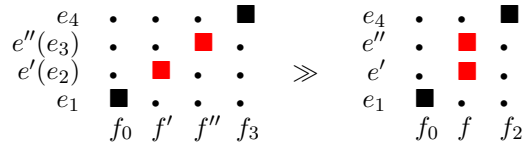


Figure 4: Transition from a boundary(+) to a no-boundary(−): The monolingual probability $P_m(cba_k^-|dh_{sk}{}^+)$ is estimated as $P_G(f)$, and the translation probability in the E-to-C direction $P_{ef}(cba_k^-|dh_{sk}{}^+)$ is estimated as $\frac{1}{J}{}^{|e|}P_G(e'|f)P_G(e''|f)$, here $|e| = 2$,$J = 3$.

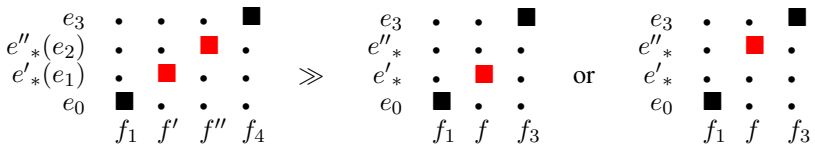


Figure 5: Transition from a boundary(+) to a no-boundary(−): The monolingual probability $P_m(cba_k^-|dh_{sk}{}^+)$ is estimated as $P_G(f)$, and the translation probability in the C-to-E direction $P_{fe}(cba_k^-|dh_{sk}{}^+)$ is estimated as $\frac{1}{I+1}P_G(f|e'_*)$ or $\frac{1}{I+1}P_G(f|e''_*)$, here $|e| = 2$ and $I = 3$.

denotes the GIZA++ corpus alignment in both directions. The GS re-segmentation step is done according to the algorithm in Table 5.

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 of the final segmentation of the training corpus $F_T$.

## 6 Integrated word segmentation in search

We described the learned and semi-supervised Chinese word segmentation methods in Section 4 and 5, where the word segmentation is learned during the training of the word alignment. However, a test corpus is still processed using a unigram segmenter, which does not guarantee optimal segmentations as addressed in Section 1. For instance, 小孩 and 孩 can both mean 'children'. The first one is used more often. Therefore, a segmenter usually puts both characters together rather than separating them. But if only 孩but not 小孩 appears in the training corpus, 小孩 should be broken into two words in the test corpus so that 孩 can be recognized and translated into 'children'.

Whenever inconsistencies appear between the expressions in training and test data, it is a good way to consider segmentation alternatives in order to adapt the writing style of the test text to that of the training text. Hence, a so-called 'integrated word segmentation' will be described in detail in this section as introduced in (Xu et al., 2005a). The algorithm works as follows: Given a set of character sequences as the input text, we take all possible segmentations of one sentence into account and integrate the segmentation decision into the search for the translation. Different segmentation possibilities represented as a lattice instead of a single segmentation are translated, and the segmentation decision is only taken during the search for the best translation.

In the conventional method a single-best word segmentation is only employed into the search for the best translation. This approach is not ideal because the segmentation may not be optimal for these translations given the training data segmentation. Making hard decisions in word segmentation may lead to loosing Chinese words that can contribute to find the correct translations. Hence, for one input sentence, we take all possible segmentations into account and represent them as a lattice. The input to the translation system is then a set of lattices instead of the segmented text. In the integrated segmentation the search decision of the word segmentation is combined with the translation decision as a global decision. The best segmentation of a sentence is only selected while the translation is generated. Using this method, we are able to translate any character-based Chinese text given a segmented training corpus, and it even outperforms a standard approach with manually segmented input text.

### 6.1 Segmentation lattice

We take the notations in Section 2. In the conventional approach the best translation of $c_1^K$ can be performed by first finding the best segmentation as in Equation 2, then searching for the best translation given fixed word segmentation:

$$\hat{e}_1^{\hat{I}} = \underset{e_1^I}{\operatorname{argmax}} \left\{ Pr(e_1^I|c_1^K, \hat{k}_1^{\hat{J}}) \right\} \quad (15)$$

Table 6: Complete algorithm of Gibbs sampler for CWS including alignment models. The observations are $D = (d_1, ..d_s, .., d_S)$, where $d_s$=$(c_1^K, e_1^I)$ indicates a bilingual sentence pair. Hidden variables $F_t$ and $A_t$ indicate the word segmentation and word alignment of the corpus in the $t$-th iteration respectively.

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

However, in the transfer of the single-best segmentation from Equation 2 to Equation 15 some segmentations that are potentially optimal for the translation may be lost. Therefore, we combine the two steps. The search is then rewritten as:

$$\hat{e}_1^{\hat{I}} = \underset{I, e_1^I}{\operatorname{argmax}} \left\{ Pr(e_1^I | c_1^K) \right\} \qquad (16)$$

We optimize the segmentation boundaries $k_1^J$ to achieve the best translation directly. In this way, the segmentation model and the translation model are combined into one model. The global decision on Chinese word segmentation and translation are performed together.

### 6.2 Constructing segmentation lattices

To perform the lattices translation we introduce the weighted finite-state acceptor (Kanthak and Ney, 2004). There are many approaches to build a segmentation lattice. The aim of the lattice construction is on one hand allowing word segmentation alternatives as candidates for translation and on the other hand avoiding too many ambiguities so that segmentations leading to optimal translations can always be preferred.

We generate the segmentation lattice with the following steps: First, we make a word list from the vocabulary of the segmented Chinese training corpus where each word in the list is mapped by its character sequences. Then we convert the mapping of the word list into a finite-state transducer for segmentation so that the input labels are the character sequences, and the output labels are Chinese words. After that, the input character sequence in the test set is represented as a linear acceptor, and this acceptor is composed with the segmentation transducer. Finally, we get a lattice which represents all possible segmentations of this sentence. We only need to read the segmentation lattice to have an integrated word segmentation in the translation. Note that the vocabulary in the test set is a subset of the character vocabulary of the word list because the unknown characters are added as single words in the word list.

### 6.3 Weighting segmentation lattices

The method for introducing segmentation alternatives is based on the assumption that the decoder is robust enough to choose the right segmentation using the translation model costs. However, if there are ambiguities, the decoder might only prefer a path with lower translation costs without considering any context information. As a result, translations differ to a great extent in comparison to the original content. Therefore, we discuss possible features to evaluate different segmentations in this section. Paths in a lattice are weighted by feature costs. Infrequent word segmentations are penalized and frequent word segmentations gain priority. In this way features in the segmentation lattice and in the decoder both contribute to find the best segmentation results.

We can weight lattice paths in many ways. For example, we can apply a language model estimated on the segmented Chinese training text. A unigram language model gives priorities to frequent words used in the training data, and a higher order language model also captures the source context information for decisions. We trained the language model on the Chinese training corpus with the SRILM toolkit (Stolcke, 2002) and used the modified Kneser-Ney discounting.

To combine the segmentation lattice with the word-based language model we simply transform the language model into a finite-state transducer and compose the lattice with it. Note that after inserting the weights the number of states and arcs in a lattice may increase for different language model histories.

## 7 Translation experiments

In this work experiments are performed on two types of datasets: a small data track, where the training corpus is rather clean and efficient to test the translation algorithms on sparse data; a large data track, where the best translation system is expected. The bilingual training corpus contains hundreds of million words, and the monolingual English data can obtain trillions of words.

We take the IWSLT (International Workshop on Spoken Language Translation) (IWSLT, 2007)

task for the small data track. The IWSLT organization holds an annual evaluation campaign that is carried out using a multilingual speech corpus on a small data track. The provided *Basic Travel Expression Corpus* (BTEC) (Takezawa et al., 2002) is a multilingual speech corpus which contains tourism-related sentences similar to those that are found in phrase books.

After the tokenization and automatic sentence segmentation, the training corpus nearly contains 43K bilingual sentences for each language as shown in Table 7. We calculated the number of words and the vocabulary size as well as the number of singletons of the corpus. We used three test sets from IWSLT translation evaluations: the Dev2, Dev3 and Eval in 2007. Running words (R.W.), OOVs of running words i.e. OOVs (R.W.) and OOVs of vocabulary i.e. OOVs (Voc.) are listed, too. Each of them contains 16 references respectively. For convenience, we only list the statistics of the first reference translation after the tokenization. The Dev2 is selected as the development corpus, Dev3 and Eval are taken as evaluation corpora.

In Table 7, we show the corpus statistics using different Chinese word segmentations: ICTCLAS (ICT, 2009); LDC (LDC, 2003); the unigram segmentation (Unigram) as described in Equation 2; the learned segmentation from alignment as discussed in Section 4 with the alignment combination IU (Och, 2002) (Learned-IU); the Gibbs sampling generated CWS (GS) presented in Section 5. In addition, we also performed training and decoding directly on Chinese characters where each Chinese character is treated as a single Chinese word (Chars).

Table 8: Corpus statistics of task GALE 2008.

| | | Chinese | English |
|---|---|---|---|
| Train | Sentences[M] | 19.5 | |
| | R. W.[M] | 242.9 | 264.3 |
| | Vocabulary[K] | 295.0 | 545.1 |
| Dev | Sentences | 480 | |
| | R. W.[K] | 13.8 | 17.2 |
| | Vocabulary[K] | 3.3 | 3.3 |
| | OOVs (R. W.) | 2 | 71 |
| | OOVs (Voc.) | 2 | 50 |
| Test | Sentences | 485 | |
| | R. W.[K] | 13.7 | 17.1 |
| | Vocabulary[K] | 3.3 | 3.3 |
| | OOVs (R. W.) | 4 | 85 |
| | OOVs (Voc.) | 3 | 53 |

For the large data track, we experiment on the GALE (Global Autonomous Language Exploitation) (GALE, 2006) 2008 task. The training corpora for GALE are a set of individual corpora collected from different sources provided by the Linguistic Data Consortium (LDC, 2005) and GALE. The domains of most sub-corpora are news articles. Some sub-corpora contain documents from other domains such as transcriptions of broadcast conversation, web text and newsgroups.

The corpus statistics of the bilingual training data and the test sets are shown in Table 8. The preprocessing step includes the tokenization and the categorization of the numbers and dates. Long sentences are segmented into short sentences using the sentence segmentation method that we introduced in (Xu et al., 2005b) to reduce the training time. After the preprocessing and segmentation, the parallel training data contains more than 19.5 million sentences and more than 240 million words in each language.

Our baseline systems are the official submission systems by the RWTH-Aachen university in the evaluation of IWSLT 2007 and in the evaluation of GALE 2008 for system combination. The training corpus (Train) is used to train the word alignment and segmentation models. The feature weights of different translation models in the decoding are optimized on the development corpus (Dev) using the downhill simplex (Press et al., 2002) algorithm with respect to the BLEU (Papineni et al., 2002) score. The resulting systems are evaluated on the evaluation (Eval) corpora. For convenience we evaluate hypotheses without case information.

### 7.1 Statistics of word Length in dictionary

The central idea of the learned and semi-supervised CWS methods is to generate the lexicon automatically using bilingual information so that the segmentation is task- and domain- oriented. As there is no unique definition of a 'correct' lexicon, we will compare the statistics on the word lengths in the lexicon of the learned segmentation from alignment (Learned lexicon) and the word lengths in the lexicon generated by the semi-supervised CWS using Gibbs sampling (GS lexicon) to that in the manual lexicon provided by LDC (LDC, 2003).

Table 9 shows the statistics of the word lengths in these three lexicons. We calculate the number of word entries, whose length is from 1 to 7 and larger than 7. For example, there are 2 334 words consisting of a single character in the LDC lexicon, 2 582 words in the learned lexicon and 1 941 words in the GS lexicon. These single character words represent 18.6% of the total number of entries in the LDC lexicon, 16.5% in the learned lexicon and 29.3% in the GS lexicon.

From Table 9 we see that in the manual LDC lexicon more than 60% of the words consist of two characters and only about 15% of the words consist of three or four characters. Longer words with more than four characters are used seldomly. Ev-

Table 7: Corpus statistics of task IWSLT 2007.

| | | Chinese | | | | | | English |
|---|---|---|---|---|---|---|---|---|
| | | Chars | ICT-CLAS | LDC | Uni-gram | Learned-IU | GS | |
| Train | Sentences[K] | 42.9 | | | | | | |
| | R.W.[K] | 519.9 | 380.3 | 385.4 | 393.8 | 343.7 | 396.8 | 420.4 |
| | Vocabulary[K] | 2.8 | 11.8 | 9.4 | 8.8 | 13.3 | 6.2 | 9.9 |
| | Singletons | 364 | 4637 | 2841 | 2629 | 4755 | 727 | 3937 |
| Dev2 | Sentences | 500 | | | | | | |
| | R.W.[K] | 4.8 | 3.6 | 3607 | 3.7 | 3.3 | 3.7 | 3.9 |
| | Vocabulary | 823 | 950 | 1021 | 987 | 1078 | 1004 | 834 |
| | OOVs (R.W.) | 7 | 75 | 52 | 49 | 17 | 16 | 216 |
| | OOVs (Voc.) | 6 | 73 | 50 | 47 | 15 | 14 | 44 |
| Dev3 | Sentences | 506 | | | | | | |
| | R.W.[K] | 5.2 | 3.8 | 3.8 | 3.9 | 3.6 | 4.0 | 4.0 |
| | Vocabulary | 837 | 936 | 996 | 969 | 1081 | 980 | 831 |
| | OOVs (R.W.) | 242 | 72 | 51 | 51 | 18 | 19 | 194 |
| | OOVs (Voc.) | 20 | 69 | 48 | 48 | 16 | 15 | 45 |
| Eval | Sentences | 489 | | | | | | |
| | R.W.[K] | 4.3 | 3.2 | 3.3 | 3.3 | 3.0 | 3.4 | 3.8 |
| | Vocabulary | 762 | 885 | 944 | 915 | 1008 | 904 | 819 |
| | OOVs (R.W.) | 5 | 60 | 37 | 33 | 9 | 13 | 205 |
| | OOVs (Voc.) | 5 | 59 | 36 | 32 | 9 | 12 | 33 |

idently, there are too many words with more than two characters in the learned dictionary. In the GS lexicon, the length distribution is similar to that in the LDC lexicon. There are about 15% word entries containing more than two characters. Figure 6 visualizes the statistics in Table 9. The horizontal axes show the word lengths and the vertical axes show the percentages of the word entries in the lexicon with a given length.

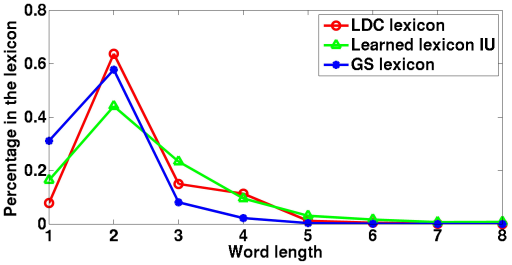


Figure 6: Curves of number of words given a length in the LDC lexicon, learned lexicon with alignment combination IU and GS lexicon using Gibbs sampling.

There are frequencies for each word entry in the lexicon. If we take these frequencies into account, we obtain the total number of words given a length in the lexicon. The percentages are shown in Figure 7. For example, if the frequency of one two-character word entry is ten, we add ten to the two-character words count instead of one. Figure 7 shows that the word distribution in the manual LDC lexicon is closer to that in the GS lexicon than to that in the learned lexicon.

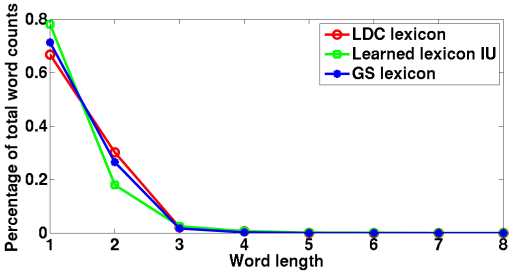


Figure 7: Curves of total number of words given a length in the LDC lexicon, learned lexicon with alignment combination IU and GS lexicon using Gibbs sampling.

### 7.2 Evaluation results

In Table 7, we show the corpus statistics using different Chinese word segmentations: ICT-CLAS (ICT, 2009); LDC (LDC, 2003); the unigram segmentation (Unigram) as described in Equation 2; the learned segmentation from alignment as discussed in Section 4 with the alignment combination IU (Och, 2002) (Learned-IU); the Gibbs sampling generated CWS (GS) presented in Section 5. In addition, we also performed training and decoding directly on Chinese characters where each Chinese character is treated as a single Chi-

Table 9: Statistics of word lengths in the vocabulary of the LDC lexicon, learned lexicon with alignment combination IU (Och, 2002) (Learned lexicon-IU) and lexicon generated by semi-supervised CWS using Gibbs sampling (GS lexicon).

| Word length | LDC lexicon | | Learned lexicon-IU | | GS lexicon | |
|---|---|---|---|---|---|---|
| | Count | [%] | Count | [%] | Count | [%] |
| 1 | 2 334 | 18.6 | 2 582 | 16.5 | 1 941 | 29.3 |
| 2 | 8 149 | 65.1 | 6 926 | 44.1 | 3 599 | 54.3 |
| 3 | 1 188 | 9.5 | 3 670 | 23.4 | 508 | 7.7 |
| 4 | 759 | 6.1 | 1 507 | 9.6 | 141 | 2.1 |
| 5 | 70 | 0.6 | 490 | 3.1 | 24 | 3.6 |
| 6 | 20 | 0.2 | 267 | 1.7 | 9 | 1.4 |
| 7 | 6 | 0.0 | 118 | 0.8 | 3 | 0.5 |
| $> 7$ | 11 | 0.0 | 130 | 0.8 | 1 | 0.0 |
| total | 12 527 | 100 | 15 690 | 100 | 6226 | 100 |

nese word (Chars).

We show the translation results of CWS methods in Table 10. This includes performing training and decoding directly on Chinese characters where each Chinese character is treated as a single Chinese word (Chars), LDC (LDC, 2003), CRF (Andrew, 2006), ICTCLAS (Zhang and Malik, 2003), the unigram segmentation (Unigram) and 9-gram segmentation (9-gram) as described in Section 3, the learned segmentation from alignment with three alignment combination approaches, right,intersect and IU, as discussed in Section 4 and the Gibbs sampling generated CWS (GS) as in Section 5 as well as the integrated word segmentation in Section 6. The evaluations are performed under automatic criteria WER, PER, BLEU and TER (Snover et al., 2006) scores. The average sentence length of each hypothesis (Length) is shown in Table 10, too.

From Table 10, we see IU combination performs the best with respect to the translation result among all the alignment combination methods: right, intersect and IU as described in (Och, 2002). The integrated Chinese word segmentation performed on the unigram segmented system leads to a consistent improvement in the translation performance with respect to the TER score. The word segmentation lattice is not weighted because we did not observe an improvement using the language model weighting as described in Section 6.3. The GS is evaluated using our full model with both monolingual and bilingual information according to Section **??**. The model weights $\lambda$ in Equation 6 are 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++ in Table 6. We set $\kappa = 2$ in the Poisson model and $\alpha_1 = \alpha_2 = \alpha_3 = 0.3$ for new word generation for simplicity. The unigram method is implied to initialize the GS method and to segment the test corpus using a lexicon learned by GS. From the corpus statistics in Table 7 we observe that the vocabulary size of the Chinese training corpus is smaller in GS than in the unigram method, even though the number of running words is similar in both corpora. This shows that the distribution of Chinese words is more concentrated if using GS. In the final translation results, under all test sets and evaluation criteria, GS outperforms the other methods. The absolute WER decreases with 1.2% on Dev3 and with 1.1% on Eval data over unigram method.

In the semi-supervised CWS we take the unigram method to initialize the Gibbs sampling and to segment the test corpus using a combined lexicon as illustrated in Figure 1. The probability of each word entry from the manual lexicon is linearly interpolated with that in the Gibbs sampling generated lexicon with a weight of 0.4. The integrated Chinese word segmentation does not perform better than the baseline in the large data track, because we are not able to put corpora using different Chinese word segmentations into word alignment training due to limited computational resources and a large amount of training data currently.

Table 11 shows the translation performance on the GALE 2007 task. The LDC, unigram and GS segmentation methods are compared. Because of the computational complexity, we only optimized the feature weights of different translation models in the decoding for the LDC system. There is still room for improvements, if we can optimize the feature weights of translation models in the decoding for the other systems. 'Semi-supervised (LDC+GS-0.5)' indicates the semi-supervised CWS method described in Section 5.1 where we combine the LDC lexicon and the Gibbs sampling generated lexicon using linear interpolation with weights of 0.5 and 0.5, re-

Table 10: Translation results for various Chinese word segmentation methods studied in this work on task IWSLT 2007[%]. The experiments marked by an asterik are based on the software downloaded from (LDC, 2003) and (ICT, 2009). The baseline system using ICTCLAS is an improved RWTH system after the final submission in the IWSLT 2007.

| Test | CWS method | WER | PER | BLEU | TER | Length |
|---|---|---|---|---|---|---|
| Dev2 | Unigram | 38.2 | 31.2 | 55.4 | 37.0 | 7.49 |
| | GS | 36.8 | 30.0 | 56.6 | 35.5 | 7.35 |
| | Integrated | 37.4 | 30.5 | 55.9 | 36.1 | 7.38 |
| Dev3 | Unigram | 33.5 | 27.5 | 60.4 | 32.1 | 7.60 |
| | GS | 32.3 | 26.6 | 61.0 | 31.4 | 7.61 |
| | Integrated | 32.8 | 26.3 | 60.1 | 31.0 | 7.38 |
| Eval | Chars | 49.3 | 41.8 | 35.4 | 47.5 | 6.98 |
| | LDC | 46.2 | 40.0 | 39.2 | 45.0 | 6.69 |
| | CRF | 47.0 | 41.5 | 37.2 | 46.0 | 6.34 |
| | ICTCLAS | 45.9 | 40.4 | 40.1 | 44.9 | 6.75 |
| | Unigram | 46.8 | 40.2 | 41.6 | 45.6 | 7.11 |
| | 9-gram | 46.9 | 40.4 | 40.1 | 45.4 | 6.94 |
| | Learned-right | 47.4 | 42.2 | 38.5 | 46.5 | 6.64 |
| | Learned-intersect | 46.5 | 40.1 | 39.5 | 45.3 | 6.98 |
| | Learned-IU | 46.4 | 40.5 | 40.0 | 45.3 | 6.94 |
| | GS | 45.9 | 40.0 | 41.6 | 44.8 | 6.93 |
| | Integrated | 44.9 | 39.4 | 41.0 | 43.5 | 6.72 |

Table 11: Translation results for various Chinese word segmentation methods studied in this work on task GALE 2008[%].

| Test | CWS Method | WER | PER | BLEU | TER |
|---|---|---|---|---|---|
| Eval | LDC | 73.0 | 49.5 | 28.2 | 67.1 |
| | Unigram | 73.0 | 49.7 | 28.4 | 67.2 |
| | GS | 73.0 | 49.5 | 28.4 | 67.1 |
| | Semi-supervised (LDC+GS-0.5) | 72.6 | 48.7 | 28.5 | 66.3 |
| | Semi-supervised (LDC+GS-0.6) | 72.5 | 48.6 | 28.7 | 66.3 |

spectively. 'Semi-supervised (LDC+GS-0.5)' indicates semi-supervised CWS method where the weight of the LDC lexicon is 0.6 and the weight of the Gibbs sampling generated lexicon is 0.4. The semi-supervised method improve the performance with 0.5% in the BLEU score compared to the baseline method.

To measure the diversity of translation systems generated based on different Chinese word segmentations, we performed leaving-one-system-out experiment. In Table 12 we show the translation performance by leaving a single system out of system combination and re-optimizing the weights on the Test 08 data. The Dev 08 data was used as a blind test set. The experiment was performed on the newswire documents.

The result on the blind test by the best single system at RWTH official submission is evaluated with a BLEU score of 32.1%. Several systems are included in the system combination: standard phrase-based systems, syntax-based systems and systems with different Chinese word segmentations. After the system combination, the BLEU score is enhanced to 34.9%. By leaving out the phrase-based system with LDC segmentation, the BLEU score decreases 0.1%, while by leaving out the phrase-based system with semi-supervised CWS, the BLEU score decreases 0.4%, which indicates that the system with semi-supervised CWS contributes 0.4% in the BLEU score on Dev 08 in the RWTH system combination. The semi-supervised CWS is applied into the system combination for the final submission of RWTH Aachen University.

### 7.3 Effect of segmentation on translation results

We present some examples of translation outputs to show that the segmentation has an effect on the translation quality in Table 13. Three examples are selected from the experiments on the Eval corpus of IWSLT 2007. For each of them we show the Chinese source sentence segmented using the unigram, learned segmentation and semi-supervised

Table 12: Final translation submission results of leaving-one-system-out experiment in system combination by RWTH on GALE 2008 for newswire data[%].

| Newswire | Test 08 | | Dev 08 | |
|---|---|---|---|---|
| | BLEU | TER | BLEU | TER |
| Best single system | 31.0 | 61.6 | 32.1 | 61.8 |
| System combination | 34.9 | 57.8 | 35.7 | 57.5 |
| Leaving one out: without pbt-LDC | 35.2 | 57.7 | 35.6 | 57.6 |
| Leaving one out: without pbt-semi-supervised-CWS | 34.6 | 58.0 | 35.3 | 57.8 |

Table 13: Examples of segmentation and translation outputs with baseline, learned segmentation and semi-supervised CWS method.

| | | |
|---|---|---|
| a) | Baseline | 我 会 晚 些 到 , .. |
| | | i will arrive late , .. |
| | Learned-IU | 我 会 晚些 到 , .. |
| | | i will call later , .. |
| | GS | 我 会 晚 些 到 , .. |
| | CWS | i will arrive late , .. |
| | REF | I'll arrive late , .. |
| b) | Baseline | 请 告诉 我 总 金额 。 |
| | | please show me the in . |
| | Learned-IU | 请 告诉 我 总 金 额 。 |
| | | please show me the total price . |
| | GS | 请 告诉 我 总 金 额 。 |
| | CWS | please show me the total price . |
| | REF | can you tell me the total amount ? |
| c) | Baseline | 请 给 我 可口可乐 。 |
| | | please give me . |
| | Learned-IU | 请给 我 可口 可乐 。 |
| | | please give me a good coke . |
| | GS | 请 给 我 可 口 可乐 。 |
| | CWS | please give me a coke . |
| | REF | coke , please . |

segmentation method, as well as their corresponding translation and the human reference translation.

In the first example, both semi-supervised CWS and baseline methods lead to correct segmentations, while the learned segmentation results in an error, because '晚些' should be separated into two words. '晚' means 'late' and '些' means 'a little'.

In the second example, the translation results in the learned and semi-supervised CWS segmentation are closer to the reference translation. As '金' or '额' occurs more often than they are combined in the training corpus, it is easier to recognize the single character word in the evaluation text.

In the third example, the segmentation with both learned method and baseline method made mistakes. For the learned method '请给' should be in two words, where '请' means 'please' and '给' means 'give'. Though the baseline segmentation is reasonable for a human evaluation, the translation result is still erroneous because the sequence of characters '可口可乐' never appears in the training corpus, but '可乐' can be found many times. As both of them mean 'coke', we only need the word '可乐' to obtain the correct translation.

We compared all translation outputs on the Eval set using semi-supervised CWS with the baseline method. 196 sentences are different out of 489 lines whereby 64 sentences from semi-supervised CWS are better, 33 sentences are worse, and the remaining sentences have similar translation qualities.

### 7.4 Conclusions

We have successfully developed novel Chinese word segmentation methods for statistical machine translation. In the training process, Chinese word boundaries are learned jointly with the word alignments. Both monolingual and bilingual information are employed to derive a segmentation suitable for machine translation. New Chinese words and distributions are generated automatically. In the translation time multiple segmentation alternatives instead of the single-best segmentation are considered, and the segmentation decision is taken during the search for the best translation. Not only in a small, but also in a large data environment, our method outperformed the standard Chinese word segmentation approach in terms of the final Chinese to English translation quality.

## References


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

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

Blunsom, Phil and Miles Osborne. 2008. Probabilistic inference for machine translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natu-*

*ral Language Processing*, pages 215–223, Honolulu, Hawaii, October.

Chen, Aitao, Yiping Zhou, Anne Zhang, and Gordon Sun. 2005. Unigram language model for Chinese word segmentation. *Proceedings of the 4th SIGHAN Workshop on Chinese Language Processing*, pages 138–141.

GALE. 2006. Nist machine translation evaluation for gale home page. http://www.nist.gov/speech/tests/gale.

Geman, S. and D. Geman. 1984. Stochastic relaxation, gibbs distribution and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741.

Goldwater, Sharon, Thomas L. Griffiths, and Mark Johnson. 2006. Contextual dependencies in unsupervised word segmentation. In *Proceedings of Coling/ACL*, Sydney, July.

ICT. 2009. ICTCLAS Chinese word segmentation system home page. http://www.ictclas.org.

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

Kanthak, Stephan and Hermann Ney. 2004. FSA: An efficient and flexible C++ toolkit for finite state automata using on-demand computation. pages 510–517, Barcelona, Spain, July.

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

LDC. 2005. Linguistic data consortium resource home page. http://www.ldc.upenn.edu/Projects/TIDES.

Luo, Xiaoqiang and Salim Roukos. 1996. An iterative algorithm to build Chinese language models. In *Proceedings of the 34th annual meeting of the Associaton for Computational Linguistics*, pages 139–143, Santa Cruz, California.

Ma, Yanjun, Nicolas Stroppa, and Andy Way. 2007. Bootstrapping word alignment via word packing. In *Proceedings the 45th Annual Meeting of the Association for Computational Linguistics*, pages 304–311, Prague, Czech Republic, June.

Ney, Hermann. 1999. Speech translation: Coupling of recognition and translation. In *Proceedings International Conference on Acoustics, Speech, and Signal Processing*, pages 517–520, Phoenix, AR, March.

Och, Franz J. and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):135–244, December.

Och, Franz J. 1999. An efficient method for determining bilingual word classes. In *EACL '99: Ninth Conference of the Europ. Chapter of the Association for Computational Linguistics*, pages 71–76, Bergen, Norway, June.

Och, Franz J. 2002. *Statistical Machine Translation: From Single Word Models to Alignment Templates*. Ph.D. thesis, Computer Science Department, RWTH Aachen, Germany, October.

Papineni, Kishore A., Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, July.

Press, William H., Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. 2002. *Numerical Recipes in C++*. Cambridge University Press, Cambridge, UK.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of AMTA*, pages 223–231, Cambridge, MA, August.

Sproat, Richard and Chilin Shih. 1990. A statistical method for finding word boundaries in Chinese text. *Computer Processing of Chinese and Oriental Languages*, 4:336–351, April.

Stolcke, Andreas. 2002. SRILM - an extensible language modeling toolkit. In *Proceedings of the International Conference On Spoken Language Processing*, pages 901–904, Denver, Colorado, September.

Takezawa, Toshiyuki, Eiichiro Sumita, Fumiaki Sugaya, Hirofumi Yamamoto, and Seiichi Yamamoto. 2002. Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC)*, pages 147–152, Las Palmas, Spain, May.

Wang, Zhuoran and Ting Liu. 2005. Chinese unknown word identification based on local bigram model. *International journal of computer processing of oriental languages*, 18(3):185–196.

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

Xu, Jia, Evgeny Matusov, Richard Zens, and Hermann Ney. 2005a. Integrated Chinese word segmentation in statistical machine translation. In *Proceedings of IWSLT 2005 (International Workshop on Spoken Language Translation)*, pages 141–147, Pittsburgh, PA, October.

Xu, Jia, Richard Zens, and Hermann Ney. 2005b. Sentence segmentation using IBM word alignment model 1. In *Proceedings of EAMT 2005 (10th Annual Conference of the European Association for Machine Translation)*, pages 280–287, Budapest, Hungary, May.

Xu, Jia, Jianfeng Gao, Kristina Toutanova, and Hermann Ney. 2008. Bayesian semi-supervised chinese word segmentation for statistical machine translation. In *Proceedings of COLING*, pages 1017–1024, August.

Zhang, Hao and Jitendra Malik. 2003. Learning a discriminative classifier using shape context distances. In *CVPR 2003, International Conference on Computer Vision and Pattern Recognition*, volume I, pages 242–247, Madison, WI, June.