Unsupervised Joint Monolingual Character Alignment and Word Segmentation

Zhiyang Teng[1,2], Hao Xiong[2,3], and Qun Liu[2,4]

[1] University of Chinese Academy of Sciences
[2] Institute of Computing Technology, Chinese Academy of Sciences
[3] Torangetek Information Technology (Beijing) Ltd.
[4] Centre for Next Generation Localisation
Faculty of Engineering and Computing, Dublin City University
{tengzhiyang,xionghao,liuqun}@ict.ac.cn

Abstract. We propose a novel Bayesian model for fully unsupervised word segmentation based on monolingual character alignment. Adapted bilingual word alignment models and a Bayesian language model are combined through product of experts to estimate the joint posterior distribution of a monolingual character alignment and the corresponding segmentation. Our approach enhances the performance of conventional hierarchical Pitman-Yor language models with richer character-level features. In the conducted experiments, our model achieves an 88.6% word token f-score on the standard Brent version of the Bernstein-Ratner corpora. Moreover, on standard Chinese segmentation datasets, our method outperforms a baseline model by 1.9-2.9 f-score points.

Keywords: unsupervised word segmentation, word alignment, Gibbs sampling; Pitman-Yor process.

1 Introduction

Many advanced natural language processing applications, such as dependency parsing and machine translation, use words as basic units. Unlike English, there are no white spaces between words in many Asian languages. Therefore, identifying word boundaries is a fundamental task of processing these languages.

There are two major categories of machine learning approaches: supervised methods and unsupervised methods. Supervised methods rely heavily on labeled data of a given language, thus they require much manual work. On the other hand, unsupervised methods have become increasingly important in research due to their independence of human efforts, as well as adaptability to any domains. In addition, the unsupervised learning process shows insights on how human beings acquire lexical knowledge.

In general, many previous unsupervised methods can be classified into two categories. The first category focuses on making use of heuristic rules based on local statistics such as the cohesion and the separating degree of resulting units [1] [2]. The second category evaluates probability of a segmentation of a given string based on explicit probabilistic models via nonparametric Bayesian inference [3–5]. Bayesian methods become popular because of its simplicity, interpretability and high accuracy. While a challenge

M. Sun et al. (Eds.): CCL and NLP-NABD 2014, LNAI 8801, pp. 1–12, 2014.

for Bayesian unsupervised word segmentation is how to model contextual dependencies. Contextual information plays a significant role in evaluating segmentation scores. Contextual dependencies include word-level dependencies and character-level dependencies. Several hierarchical Bayesian models are capable to capture continuous word-level dependencies [3–5]. Besides, [4] considered continuous character dependencies and [5] characterized a wider range of inter-word dependencies by adaptor grammars which is the state-of-the-art model. But adaptor grammars for segmentation is dependent on language. Different grammars need to be carefully designed for different languages. It is still expensive to apply adaptor grammar on natural text corpora due to high computational cost.

In addition to normal word-level dependencies, our approach utilizes character-level dependencies from three perspectives. Firstly, we try to explore not only continuous character groups but also gappy character patterns among different words. For example, we intend to learn the extremely meaningful gappy pattern "h...t" among words such as "**h**a**t**", "**h**i**t**","**h**o**t**" and "**h**ur**t**". Similar patterns also can be easily found in Chinese. Pattern "**计**...**器**" appears in words such as "**计**算**器** (calculator)", "**计**时**器** (timer)", "**计**分**器** (scoring indicator)" and "**计**程**器** (taximeter)". When we come to a plausible word of this pattern, it might be reasonable to assign this word high probability. Secondly, We pay direct attention to the location of a character. The location of a character in a string have great impacts on whether the character should be merged into left, right or as a separate word. For example, given an English string "**a**sm**a**rtboy", the first letter "a" tends to be a separate word, but the fourth letter "a" tends to be combined with other characters. Thirdly, we show emphasis on the fertility of a character. Fertility means that how many characters a character usually related to. It has an implicit influence on word length which is believed to be an important factor for unsupervised word segmentation.

Word alignment models for SMT are very good at inducing lexical association, locality and fertility parameters. [6] exploited monolingual word alignments to extract collocations. [7] demonstrated that these factors were surprisingly effective for the unsupervised dependency parsing under a monolingual alignment model. We are inspired to treat the word segmentation as a problem of monolingual character alignment. By taking the source side and the target side as the same sequence of monolingual characters, we can produce an alignment inside a string. When we produce a character alignment, we simultaneously obtain a segmentation that each word is consistent with the character alignment by a mapping algorithm. A Gibbs sampler samples every candidate alignment position for each character. The posterior distribution is product of experts of IBM Models 1-3 [8], hidden markov alignment model [9], as well as a hierarchical Pitman-Yor language model [10]. After several iterations, most frequent samples are selected to be final segmentation results.

Our model achieves an 88.6% word token F-score on English phonetic transcripts corpora [11], which outperforms the best model in [4] by more than 16.5% in F-score and approaches the state-of-art model [5]. On standard Chinese text datasets, we also improve the segmentation accuracy by 1.9 to 2.9 F-score points compared to [1].

The rest of the paper is organized as follows. After introducing background and related works, we describe the joint model. Then we explain the Gibbs sampling algorithm. In the last two sections, we show the experimental results and draw conclusions.

2 Background and Related Work

2.1 Word Alignment

Given a foreign sentence $\mathbf{f} = (f_1, ..., f_J)$ and an English sentence $\mathbf{e} = (e_1, ..., e_I)$, to model the translation probability from **e** to **f** , a hidden alignment variable a is introduced, $Pr(\mathbf{f}|\mathbf{e}) = \sum_a Pr(\mathbf{f}, a|\mathbf{e})$, where $a = (a_1, ..., a_J)$ and $a_j \in \{0, ..., I\}$. IBM model 1 only considers lexical translation probability $t(f_j|e_{a_j})$. Model 2 adds an explicit alignment model $a(a_j|j, I, J)$, which considers the impact of location. Model 3 adds a fertility model $n(\phi_i|e_i)$ to indicate how many words e usually translates to. In Hidden Markov alignment model, an alignment is dependent of the previous one.

The word alignment problem was joint inference with segmentation learning in [12], [13] and [14]. But all these works rely on bilingual information.

2.2 Introduction to Pitman-Yor Process

A Pitman-Yor Process (PYP) [15] is a stochastic process that generates power-law distributions. It is governed by a discount parameter $0 \leq d < 1$, a strength parameter $a > -d$ and a base distribution G_0. The generated distribution G is marked as $G \sim PYP(a, d, G_0)$. The discount parameter d is responsible for probability smoothing while the strength parameter a controls the similarity between G_0 and G.

The generative procedure of PYP can be represented by a variant of the Chinese Restaurant Process (CRP). CRP can be described using the analogy of a restaurant has an infinite number of tables, each of which has an infinite capacity for customers. Customers enter the restaurant one by one and each chooses to sit at a table. The first customer always sits at the first table. Suppose after a time, a restaurant already has n customers and m occupied tables. The next customer either selects an occupied table ($1 \leq k \leq m$) or an empty table ($k = m + 1$) to seat. Let z_i be the table index of the i-th customer. Then the table choosing probability

$$p(z_{n+1} = k|z_1, ..., z_n) = \begin{cases} \dfrac{c_{t_k} - d}{n + a} & 1 \leq k \leq m \\ \dfrac{d * m + a}{n + a} & k = m + 1 \end{cases} \tag{1}$$

where c_{t_k} is the number of customers seated at table k in $z_1, ..., z_n$.

For the labeled PYP, we can regard "label" as a dish served to customers. When a customer seat an occupied table, he can share the dish labeling that table with others. Otherwise, the table he takes will be labeled by a dish h with probability $G_0(h)$. Let l_i denote the label of table i. Given previous label and table assignments, we can sum over all the tables labeled with h,

$$p(l_{n+1} = h|z_1, ..., z_n, l_1, ..., l_n) = \frac{c_h - d * m_h + (d * m + a) * G_0(h)}{n + a} \tag{2}$$

where m_h is the number of tables served with the dish h and c_h is the number of customers who are served with the dish h in the previous table assignments.

Labeled PYP is used by many bayesian unsupervised word segmentation models. The hierarchical Pitman-Yor language model describes the n-gram language model in

a way that the $(n-1)-$gram probability distribution is used as the base distribution to generate n-gram probability distribution. The unigram model $G_1 = \{P_1(\cdot)\}$ is generated as $G_1 \propto PYP(a_1, d_1, G_0)$, the bigram model $G_2 = \{P_2(\cdot|w)\}$ is generated as $G_2 \propto PYP(a_2, d_2, G_1)$ and so on. G_0 is a probability distribution of words in the corpus vocabulary. [4] employs a nested hierarchical Pitman-Yor language model. The base distribution $G_0 = \{P_0(\cdot)\}$ is also generated as $G_0 \propto PYP(a_0, d_0, G_c)$, where G_c is a character-level n-gram language model and G_c is generated in the same way of word-level distributions. Dirichlet Process and Hierarchical Dirichlet Process are also used for segmentation in [3][16]. Dirichlet Process is a special case of PYP with discount parameter equals to zero. Adaptor grammars also uses PYP to describe the probability distribution of a parsing rule [5].

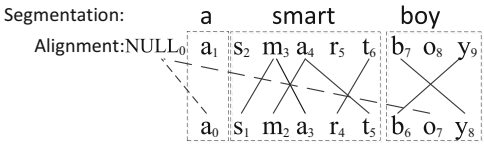
3 Joint Model of Character Alignment and Word Segmentation

3.1 Monolingual Character Alignment (MCA)

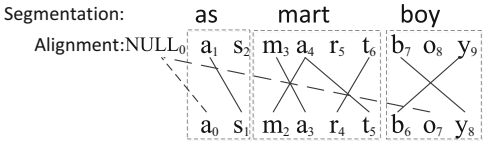
Given a string $\mathbf{s} = c_1...c_n$, the character alignment $a^* = \{(j, a_j)|j \in [1...n], a_j \in [0...n]\}$ is computed by equation (4).

$$a^* = \arg\max_a P(a|\mathbf{s}) \tag{3}$$

$a_j = i$ means character c_j aligns to character c_i. $a_j = 0$ means character c_j aligns to "NULL". Monolingual character alignment (MCA) prevents each character is aligned to itself at the same position. Without this constraint, each character will tend to align to itself. That kind of alignments make no sense for segmentation. Figure 1 shows two MCA examples of string "asmartboy".



(a) A correct segmentation example



(b) A wrong segmentation example

Fig. 1. Two alignment examples of string "asmartboy". The subscript number stands for the corresponding position in a string. (a) has a good derivation for correct segmentation , "a smart boy". (b) leads to a bad result,"as mart boy".

3.2 Generative Story

Given a string s, our model maximizes the probability of a character alignment a and a word segmentation w by equation (5).

$$(a^*, w^*) = \arg\max_{a,w} p(a, w|\mathbf{s}, \Theta) \tag{4}$$

Θ is hyperparameter. We apply generative models to decompose $p(a, w|\mathbf{s}, \Theta)$.

Two-Stage Model: The simplest way is to employ a two-stage model. Firstly, we generate a character alignment inside the given string. Then, we deduce a word segmentation result from the character alignment. The decomposing procedure is shown in equation (6).

$$p(a, w|\mathbf{s}, \Theta) = p(a|\mathbf{s}, \Theta) * p(w|a, \mathbf{s}, \Theta) \tag{5}$$

$p(a|\mathbf{s}, \Theta)$ is the alignment model and $p(w|a, \mathbf{s}, \Theta)$ is the segmentation model. But the two-stage model has two disadvantages. First, the alignment a is very sparse. The segmentation model has little effect on the alignment model. The purpose of MCA is to infer a good segmentation, rather than to capture translation clues. We suppose character alignment and word segmentation benefit from each other. A good character alignment could lead to a good word segmentation and vice versa. Second, the computational cost of $p(w|a, \mathbf{s}, \Theta)$ is relatively expensive since every alignment has the probability to be mapped to several segmentations according to different heuristic rules. The segmentation selection procedure is relatively slow.

One-Step Model: In order to overcome defects of the two-stage model, the one-step model produce word segmentation and character alignment simultaneously. The generating procedure is shown in equation (7).

$$p(a, w|\mathbf{s}, \Theta) = p(a|\mathbf{s}, \Theta) * p(w_a|\mathbf{s}, \Theta) \tag{6}$$

w_a denotes the corresponding segmentation according to character alignment a. This model generates a character alignment a first. At the same time, it converts a to a unique segmentation w_a. The segmentation result is the side product of character alignment. One advantage of this model is that it makes a tighter connection between character alignment and word segmentation. When generating a character alignment, the probability of the corresponding segmentation must be considered. For example, if we want to compare the two alignments in Figure 1, we need to consider the plausibility of related segmentations, "a smart boy" and "as mart boy". Another advantage is that it is efficient because it performs a unique mapping from character alignment to segmentation and only requires a single step computing.

3.3 Mapping Alignment to Segmentation

We design a mapping algorithm for extracting segmentation $\mathbf{w} = w_1...w_2$ from character alignment **a** according to **alignment consistency**. Statistical machine translation models often make use of alignment consistency to extract bilingual phrase pairs. A span $c_i...c_j$ is consistent with character alignment if their alignments satisfy two conditions. (1) $\forall k$, if $i \leq k \leq j$, then $a_k \in 0 \cup [i, j]$; (2) $\forall k$, if $k < i$ or $k > j$, then $a_k \notin [i, j]$; We assume if a span consistent with character alignment then this span is a high plausible word. Considering length factors, we choose **smallest non-crossed span**

as a word. In figure 1(a), span "asmart" is a valid span of alignment consistency, but it is too long. Instead, we regard two smaller spanes "a" and "smart" as words. In monolingual character alignment, a span consistent with alignment may cover another one. In figure 1(a), "o" is a span consistent with alignment, but links "b-y" and "y-b" cross this span, therefore we choose the longer span "boy" as a word. The smallest non-crossed strategy try to control the length of a word as well as to bond connected characters as many as possible into a word. The mapping algorithm is shown in Algorithm 1.

In Algorithm 1, we do a greedy search to find smallest non-crossed span set w. Line 1-3 is initiation. f stands for the start point of the next smallest non-crossed span. P stands for the alignment boundary of the previous character. C stands for the alignment boundary of the current character. The alignment boundary of a character means the minimum value and maximum value of the set of its position, the position it aligned to and all positions aligned to it. For example, in figure 1(a), the alignment boundary of m_3 is $< 2, 4 >$. Line 4-14 traverse remained characters one by one. According to the relationship of P and C, word boundaries are determined. Line 6-10 means if P and C are not intersected, then we find a target span $[f, i-1]$ and update the value of P and f. Line 11-14 means if P and C are intersected, then we merge P and C. $P \oplus C = [\min(P.l, C.l), \max(P.r, C.r)]$. At last, we add the tail span to w at line 15. According to Algorithm 1, two character alignments and their corresponding segmentations are shown in Figure 1.

3.4 Product of Experts

We decompose $p(\mathbf{a}|\mathbf{s}, \Theta)$ in equation (7) into four sub-models according to IBM Models and Hidden Markov alignment model. Those models are used to exploit literal, position and fertility factors for segmentation. Considering character alignment and word segmentation could have effect on each other in the one-step model, we use a product of experts (PoE) to combine them. PoE multiplies several probability distribution together and has bias toward samples which have high probability in all sub-models. Let random variable aw be the pair of a and w. The probability distribution $P(aw|\mathbf{s}, \Theta)$ under PoE is shown in Equation (8).

$$P(aw|s) = \frac{\prod_i P_i(aw|s)}{\sum_{aw'} \prod_i P_i(aw'|s)}, \quad i \in \{m_1, m_2, m_3, m_h, m_s\} \tag{7}$$

m_1, m_2, m_3 and m_h are adapted from alignment models IBM Model 1-3 and HMM model respectively. m_s is a model for segmentation. Distributions over these five sub-models and their Pitman-Yor priors are shown in Table 1.

Character Association Model: Characters with high co-occurrence will tend to be bound together. IBM Model 1 is changed to model the character association probability in MCA.

$$P_{m_1}(aw|s) = P_{m_1}(a|s) = \prod_{i=1}^{l} P_{m_1}(c_i|c_{a_i}) \tag{8}$$

l is the length of s. $P_{m_1}(c_j|c_i)$ describes the probability c_j is connected to c_i. In Table 1, $|V|$ is the number of character types in the corpus.

Table 1. Distributions over five sub-models and their Pitman-Yor priors.

model	distribution
m_1	$c_j \vert c_i; G_1^{m_1}(c_i) \sim PYP(a^{m_1}, d^{m_1}, G_0^{m_1}),\ G_0^{m_1} \sim U(\frac{1}{\vert V \vert})$
m_2	$m_i \vert n_i; G_1^{m_2}(n_i) \sim PYP(a^{m_2}, d^{m_2}, G_0^{m_2}),\ G_0^{m_2} \sim U(\frac{1}{D})$
m_3	$\phi_i \vert c_i, G_1^{m_3}(c_i) \sim PYP(a^{m_3}, d^{m_3}, G_0^{m_3}),\ G_0^{m_3} \sim U(\frac{1}{F})$
m_h	$hd_i \vert hc_i, G_1^{m_h}(hc_i) \sim PYP(a^{m_h}, d^{m_h}, G_0^{m_h}),\ G_0^{m_h} \sim U(\frac{1}{T})$
m_s	$w_i \vert w_{i-1}; G_2^{m_s} \sim PYP(a_2^{m_s}, d_2^{m_s}, G_1^{m_s})$ $w_i; G_1^{m_s} \sim PYP(a_1^{m_s}, d_1^{m_s}, G_0^{m_s})$

Location Model: We use a same alternate distance model described in [7].

$$P_{m_2}(aw|s) = P_{m_2}(a|s) = \prod_{i=1}^{l} P_{m_2}(a_i - i|c_i, l) \tag{9}$$

In Table 1, $m_i = a_i - i$ and $n_i = (c_i, l)$, $P_{m_2}(m_i|n_i)$ describes the probability of the alignment offset of n_i is m_i. D is the types of possible values of m_i. We use $D = 10$ in this paper. The valid value of m_i is restricted to $[-5, 5]$. We prevent too long alignments because it will cause the under-segmentation problem. When $a_i = 0$, we use a similar method used in [17] by viewing every character is preceded by a NULL token. It means that if $a_i = 0$ then $a_i - i$ is set to be 1. Location model tries to make the same character behave differently in different positions.

Fertility Model: IBM Model 3 introduces two kinds of probability, NULL insertion probability and fertility probability. NULL insertion can not be applied to MCA. Because source side and target side are the same in MCA, so we do not need to insert NULL tokens in target side. Also in MCA all characters are allowed to be aligned to NULL, Thus $l - \phi_0$ might be zero. ϕ_0 denotes the number of characters aligned to NULL. Instead, we directly handle the alignment probability from NULL by

$$P_{m_3}(aw|s) = P_{m_3}(a|s) = \binom{l}{\phi_0} p_1^{\phi_0} p_0^{l-\phi_0} \prod_{i=1}^{l} \phi_i! n(\phi_i|c_i) \tag{10}$$

$n(\phi_i|c_i)$ indicates the probability of ϕ_i characters are aligned to c_i. In Table 1, F is types of character fertilities. We use $F = 5$ in this paper. p_1 is the probability of a character linked to NULL, $p_0 = 1 - p_1$. A character which prefers to distribution over greater value of ϕ has a tendency of forming multi-character words with neighboring aligned characters. We use $p_1 = 0.2$ in this paper.

Transition Model: [9] proposed a HMM-based alignment model. Similar to the Location Model, we reformulate the transition probability as:

$$P_{m_h}(aw|s) = P_{m_h}(a|s) = \prod_{i=1}^{l} P_{m_h}(a_i - a_{i-1}|c_{a_{i-1}}, l) \tag{11}$$

In Table 1, $hd_i = a_i - a_{i-1}$ and $hc_i = (c_i, l)$. T is the number of distance types. We make $T = 5$ in this paper. This model depicts the first order dependence of jump over characters. Jump distance is usually small inside a word but large between word

boundaries. When $a_i = 0$, the same method mentioned in Location Model is used to calculate the distance.

Segmentation Model: A bigram Pitman-Yor language model is adopted as:

$$P_{m_s}(aw|s) = P_{m_s}(w_a|s) = \prod_{i=1}^{l+1} P_{m_s}(w_i|w_{i-1}) \tag{12}$$

A special marker \$ is added to both the start and the end of the word sequence. In Table 1, the spelling model $G_0^{m_s}(w)$ is the same as [12]:

$$G_0^{m_s}(w) = \frac{e^{-\lambda_0}{\lambda_0}^k}{k!}\frac{1}{|V|^k} \tag{13}$$

where k is the length of w. Different from [12], we use a method proposed in [4] to estimate λ_0 by a Gamma Prior during each iteration instead of leaving it as a constant.

Algorithm 1: Converting Alignment to Segmentation

Input: string s, alignment boundary b
Output: word spanes w

$w \leftarrow \phi$
$f \leftarrow 1$
$P \leftarrow < b[1].l, b[1].r >$
for $i \leftarrow 2...|s|$ **do**
  $C \leftarrow < b[1].l, b[1].r >$
  if $!P \cap C$ **then**
    $w \leftarrow w \cup [f, i-1]$
    $f \leftarrow i$
    $P \leftarrow C$
  end
  else
    $P \leftarrow P \oplus C$
  end
end
$w \leftarrow w \cup [f, |s|]$

Algorithm 2: Gibbs sampler of MCA

Input: S, B
Output: $\mathbf{\Theta}$

Initialize segmentations **w** and alignments A;
for $m = 1$ *to* B **do**
  for a *in* A **do**
    Remove customers of w_a from $\mathbf{\Theta}$;
    for $i = 1$ *to* $|a|$ **do**
      Remove (i, a_i) from $\mathbf{\Theta}$;
      Draw a_i according to equation (14);
      Add (i, a_i) to $\mathbf{\Theta}$;
    end
    Add customers of w_a to $\mathbf{\Theta}$
  end
  Sample Hyperparameters of $\mathbf{\Theta}$;
end

4 Gibbs Sampling

It is hard to do exact inference due to the exponential alignments in equation (8). Therefore, we use Gibbs sampling to simulate the procedure of character alignment. Gibbs sampling is a special case of Monte Carlo Markov Chain method, and it is guaranteed to converge to the true posterior distribution. The denominator in equation (8) is expensive to track, therefore we ignore the denominator. Assume before a sampling iteration the segmentation of a string **s** is w. The distribution of candidate alignments of a position i conditioned on other values is:

$$P(a_i = j|\mathbf{A}_{-i}, \mathbf{S}; \mathbf{\Theta}) \propto P_{m_s}(w'_{a_i=j}|\mathbf{W}(\mathbf{A})_{-w_a}; \mathbf{\Theta}) \times \prod_k P_k(a_i = j|\mathbf{A}_{-i}, \mathbf{S}; \mathbf{\Theta}) \tag{14}$$

where $k \in \{m_1, m_2, m_3, m_h\}$, the subscript $-i$ denotes the exclusion of current position, $-w_a$ denotes the exclusion of current segmentation. $w'_{a_i=j}$ means the new segmentation after setting $a_i = j$.

The sampling algorithm is described in Algorithm 2. S is the monolingual corpus, B is the number of burn-in iterations. The Gibbs sampler first randomly initializes word boundaries of a string and then randomly assigns an alignment connected to characters in the same word for each character. After initialization, the Gibbs sampler repeatedly samples a reasonable alignment for each character conditioned on all other alignments and segmentations. A blocked computing is performed by moving an alignment from one position to another since each movement might result in different segmentations. An example of counts change during one movement is shown in Table 3. After B burn-in iterations, we collect K segmentation samples for each string s. The most frequent sample will be the final result. As for the hyper-parameter sampling, we use a slice sampler [5] by putting a flat beta prior $Beta(1, 1)$ on the discount parameter d and a vague prior $Gamma(10, 0.1)$ on the strength parameter a.

5 Experiments

To evaluate the efficiency of our model, we conducted experiments on two kinds of corpus. One of them is the public SIGHAN Bakeoff 2005 dataset [18]. This dataset contains four kinds of data, i.e. CITYU, MSR, PKU and AS. CITYU and AS are traditional Chinese text. MSR and PKU are simplified Chinese text. We only use the test set data for alignment. The other corpus consists of English phonetic scripts made by Brent of the Bernstein-Ratner corpus [11] in the CHILDES database [19]. A line of this corpus is "yuwanttusiD6bUk" and the corresponding English text is "you want to see the book". We need to segment the phonetic script into "yu want tu si D6 bUk". Details of all corpora are shown in Table 2.

For Chinese, punctuation and consecutive non-Chinese characters are recognized as a single character,such as English letters and Arabic numerals. We make use of them to segment a long string into several shorter strings. This preprocessing is beneficial for string alignment to overcome a part of sparsity of sentence length. We compare our result with models that also encode the information of punctuation or word types.

For each corpus, we simultaneously ran 4 chains. Each chain employs a Gibbs sampler for 501 iterations, including 250 burn-in iterations. In order to speed up convergence, we use a simulated annealing procedure, which cools down the Gibbs sampler from a high temperature $T_0 = 10$ to a final temperature $T_f = 1$ with geometric decline $(\frac{T_f}{T_0})^{\frac{1}{n}}$. n is the number of burn-in iterations. After each 10 iteration, we sample hyperparameters for 20 iterations.

Generally, We use word token precision(P), recall(R), F1-measure(F) to evaluate the performance. For phonetic scripts, we also calculate the same metrics(LP, LR, LF) over induced lexicons.

5.1 English phonetic transcripts

On English phonetic transcripts, we compared our model with Hierarchical Dirichlet Process based model (HDP), Nested Hierarchical Pitman-Yor Langauge model (NPY)

Table 2. Statistics of five corpora. W: words. C: characters. AC: the average count of a word. Word tokens divided by word types equals AC. Seg_sents: the number of sentences after preprocessing.

Corpus	W			C		Sents	Seg_sents
	AC	Types	Tokens	Types	Tokens		
CITYU	4.5	9001	40937	2953	66346	1492	11584
AS	6.5	18811	122613	3884	196299	14432	36392
PKU	7.9	13149	104373	3433	168975	1946	31128
MSR	8.3	12923	106873	3341	180987	3985	33189
Brent	25.3	1321	33399	50	95809	9791	9791

and Adaptor Grammar based model (AG). Table 4 shows the accuracy of segmentation results. Word token accuracy of our model has surpassed both HDP and NPY. It is surprised that the result of MCA is so close to AG, even MCA has a weaker ability to identify word-level collocations compared to a three layers of collocation-syllable structure in AG. One significant difference between AG and MCA is that AG models the relationship between characters in terms of a hierarchical syllable structure while MCA applies an alignment structure.

5.2 Model Comparison

We design experiments to show the effectiveness of five sub-models. Results of multiple combinations are shown in Table 5. m_1 and m_s are essential for basic segmentation. We include them in all combinations. By comparison setting (1) to (2), (3) to (4) and (4) to (6), we can find F value increase by 25.2, 16.9 and 18.6 points respectively. It approves our hypothesis that location factor m_2 plays a crucial role in improving segmentation result. Another interesting phenomenon we observed is that model m_3 always helps improve the recall value. With analysis of setting (2) and (3) together with setting (5) and (6), we can infer that the precision value does not change too much, but the recall value both increase by 5.7 points. This result can explain our fertility factor m_3 is capable of overcoming the under-segment problem to some extent. Although m_h also is a location factor, it is less powerful than m_2. But combining these two overlapped factors still achieve a valuable improvement.

5.3 Chinese Word Segmentation

For each Chinese corpus, we only use the test set data. As far as we known, ESA [1] has reported results under the same experimental conditions. NPY model used additional training data. So we use ESA as our baseline model. Results are shown in Table 6.

Our model gains an accuracy improvement from 1.9 points to 2.9 points. Even when compared to NPY model, we outperform them on MSR data set with 1.7 points improvement. However, MCA loses to NPY on CITYU corpus. We can infer from Table 2 that CITYU is a smaller corpus compared to another three Chinese corpora, and the AC value is the smallest. It tends to have a positive correlation between AC and the accuracy improvements. The greater AC value is, the more times a word appears from a corpora. Characters inside general words will have stronger relationships. NPY has good performance on smaller corpus while MCA might show its potential on larger corpus. Some

examples of segmentation of MSR corpus are“ 但 做 一些 **力所能及** 的 小事 的 机会 却 很多 。 有些 大 学生 **眼高手低** ， 不 屑于 做 小 事情”. From these results, we can see that our method can recognize some words with complex character structures, such as “力所能及” and “眼高手低”. Some words are over-segmented. “大 学生” should be merged as a whole word. “ 大” can be regarded as affix characters, it often appears in the boundary of a word. Therefore, they are more probable to align to NULL. This phenomenon can lead to some fine-grained segmentation results. But fine-grained segmentation results might be more suitable for SMT, we will evaluate them in future.

Table 3. Counts change when the alignment of s_1 moves from m_3 to a_1, as shown from Figure 1(a) to Figure1(b)

model	decrement	increment
m_1	$(s\|m)$	$(s\|a)$
m_2	$(2\|s,9)$	$(0\|s,9)$
m_3	$(1\|m),(0\|a)$	$(1\|a),(0\|m)$
m_h	$(2\|s,9),(1\|m,9)$	$(0\|s,9),(3\|m,9)$
m_s	$ a smart boy $	$ as mart boy $

Table 4. Segmentation accuracies on Brent Corpus. NPY(n) denotes n-gram NPY language model. HDP refers to the result reported in [16].

Model	P	R	F	LP	LR	LF
HDP	75.2	69.6	72.3	**63.5**	55.2	**59.1**
NPY(2)	74.8	76.7	75.7	57.3	56.6	57.0
NPY(3)	74.8	75.2	75.0	47.8	**59.7**	53.1
AG	-	-	**89**	-	-	-
MCA	**87.0**	**90.4**	88.6	63.2	52.9	57.6

Table 5. F-measure comparsion using various model combinations on English phonetic transcripts

Model	P	R	F
$(1)m_1+m_s$	65.7	53.5	**59.0**
$(2)m_1+m_2+m_s$	85.7	**82.7**	**84.2**
$(3)m_1+m_2+m_3+m_s$	85.4	**88.4**	**86.9**
$(4)m_1+m_h+m_3+m_s$	72.7	62.0	**70.0**
$(5)m_1+m_2+m_h+m_s$	86.2	**84.7**	85.4
(6)All	87.0	**90.4**	**88.6**

Table 6. Segmentation accuracy on Chinese corpora. ESA_{best} denotes setting 4 in [1]. Marker ’+’ shows the increment from ESA_{best} to MCA.

Model	CITYU	PKU	MSR	AS
ESA_{best}	76.0	77.8	80.1	78.5
NPY(2)	**82.4**	-	80.2	-
NPY(3)	81.7	-	80.7	-
MCA	77.9 (+1.9)	**80.7** (+2.9)	**82.4** (+2.3)	**80.6** (+2.1)

6 Conclusion and Future Work

We present that it is beneficial to incorporate global character features into hierarchical bayesian models for unsupervised word segmentation. We adopt a joint model to produce monolingual character alignment and word segmentation at the same time. Through experiments, we show that this model plays a significant role in improving word segmentation accuracy on both phonetic scripts and Chinese natural text corpus. In the future, we will work out character to block alignment models instead of transforming character to character alignment models at hand.

Acknowledgements. Work supported by National Natural Science Foundation of China (Contract 61202216) and CAS Action Plan for the Development of Western China (No. KGZD- EW-501).

References

1. Wang, H., Zhu, J., Tang, S., Fan, X.: A new unsupervised approach to word segmentation. CL 37, 421–454 (2011)
2. Sun, M., Shen, D., Tsou, B.K.: Chinese word segmentation without using lexicon and hand-crafted training data. In: Proceedings of the Joint Conference of ACL and COLING, Montreal, Quebec, Canada, pp. 1265–1271. ACL (1998)
3. Goldwater, S., Griffiths, T.L., Johnson, M.: Contextual dependencies in unsupervised word segmentation. In: Proceedings of the Joint Conference of ACL and COLING, ACL-44, Stroudsburg, PA, USA, pp. 673–680 (2006)
4. Mochihashi, D., Yamada, T., Ueda, N.: Bayesian unsupervised word segmentation with nested pitman-yor language modeling. In: Proceedings of the Joint Conference of ACL and IJCNLP, ACL 2009, Stroudsburg, PA, USA, pp. 100–108 (2009)
5. Johnson, M., Goldwater, S.: Improving nonparameteric bayesian inference: experiments on unsupervised word segmentation with adaptor grammars. In: Proceedings of Human Language Technologies: The 2009 NAACL, NAACL 2009, Stroudsburg, PA, USA, pp. 317–325 (2009)
6. Liu, Z., Wang, H., Wu, H., Li, S.: Collocation extraction using monolingual word alignment method. In: Proceedings of EMNLP, Singapore, pp. 487–495 (2009)
7. Brody, S.: It depends on the translation: unsupervised dependency parsing via word alignment. In: Proceedings of EMNLP, EMNLP 2010, Stroudsburg, PA, USA, pp. 1214–1222 (2010)
8. Brown, P.F., Pietra, V.J.D., Pietra, S.A.D., Mercer, R.L.: The mathematics of statistical machine translation: Parameter estimation. Comput. Linguist. 19, 263–311 (1993)
9. Vogel, S., Ney, H., Tillmann, C.: Hmm-based word alignment in statistical translation. In: Proceedings of COLING, COLING 1996, Stroudsburg, PA, USA, pp. 836–841 (1996)
10. Teh, Y.W.: A hierarchical bayesian language model based on pitman-yor processes. In: Proceedings of the Joint Conference of ACL and COLING, ACL-44, Stroudsburg, PA, USA, pp. 985–992 (2006)
11. Bernstein-Ratner, N.: The phonology of parent-child speech. In: Nelson, K., van Kleeck, A. (eds.), vol. 6. Erlbaum, Hillsdale (1987)
12. Xu, J., Gao, J., Toutanova, K., Ney, H.: Bayesian semi-supervised Chinese word segmentation for statistical machine translation. In: Proceedings of COLING, COLING 2008, Stroudsburg, PA, USA, pp. 1017–1024 (2008)
13. Nguyen, T., Vogel, S., Smith, N.A.: Nonparametric word segmentation for machine translation. In: Proceedings of COLING, COLING 2010, Stroudsburg, PA, USA, pp. 815–823 (2010)
14. Chung, T., Gildea, D.: Unsupervised tokenization for machine translation. In: Proceedings of EMNLP, EMNLP 2009, Stroudsburg, PA, USA, pp. 718–726 (2009)
15. Pitman, J., Yor, M.: The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator (1995)
16. Goldwater, S., Griffiths, T.L., Johnson, M.: A bayesian framework for word segmentation: Exploring the effects of Context. Cognition 112, 21–54 (2009)
17. Och, F.J., Ney, H., Josef, F., Ney, O.H.: A systematic comparison of various statistical alignment models. Computational Linguistics 29 (2003)
18. Tom, E.: Second international Chinese word segmentation bakeoff (2005)
19. MacWhinney, B., Snow, C., et al.: The child language data exchange system. Journal of Child Language 12, 271–296 (1985)