A Local Generative Model for Chinese Word Segmentation

Kaixu Zhang[1], Maosong Sun[1], and Ping Xue[2]

[1] State Key Lab of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology
Tsinghua University, Beijing, 100084, China P.R.
zhangkx03@mails.thu.edu.cn, sms@thu.edu.cn
[2] The Boeing Company
ping.xue@boeing.com

Abstract. This paper presents a local generative model for Chinese word segmentation, which has faster learning process than discriminative models and can do unsupervised learning. It has the ability to make use of larger resources. In this model, four successive characters are used to determine whether a character interval should be a word boundary or not. The Gibbs sampling algorithm, as well as three additional rules, is applied for the unsupervised learning. Besides words, the word candidates that are generated by our model can improve the performance of Chinese information retrieval. The experiments show that in supervised learning our method outperforms a language model based method. And the performance on one corpus is better than the best one reported in SIGHAN bakeoff 05. In unsupervised learning, our method achieves the comparable performance compared to the state-of-the-art method.

Keywords: probability model, natural language processing, Chinese word segmentation.

1 Introduction

Being different from English, there are no delimiters in Chinese text to indicate words. The words are potentially existent, in a certain sense, and important for further NLP tasks or information retrieval (IR). The fundamental task to segment characters in Chinese text into words is named as Chinese word segmentation (CWS).

Xue [1] presented a maximum entropy (ME) based model for CWS as character tagging using local context as evidence. To avoid the weakness of these local models, conditional random field (CRF) models [2], perceptrons [3] and other global discriminative models [4] were introduced. Global models, namely sentence models, deal with each sentence or paragraph as a whole. Indirectly, long-distance relations can be taken into account. These models get good performance in the literature. However, the learning process of these models are time-consuming.

P.-J. Cheng et al. (Eds.): AIRS 2010, LNCS 6458, pp. 420–431, 2010.

Language models, which estimate the generating probability of each sentence, are used for CWS. As generative models, language models can learn from corpora much faster than discriminative models, and can be easily adapted for unsupervised learning [5], [6].

In this paper, we present a local generative model, which has not been explored before. The relationship between our model and existent models is shown in Table 1.

Table 1. The relationship between our model and other models for CWS

	local model	global model
generative	our model	language models
discriminative	ME based	CRF based

Following the notation of [7] and [8], the *interval* between two successive characters in Chinese can be classified into two types, namely *separated* and *combined*. A separated interval indicates that those two characters belong to two separated words, while a combined interval indicates that those two characters belong to the same words. The separated intervals are therefore recognized as word boundaries.

Our local generative model estimates the probability of four successive characters, called the *context*, given the type of the interval in the middle of these characters. And the model predicts the interval type separately and only based on the corresponding context.

We also provide a method to generate *word candidates* based on our model. The word candidates are the words that an input sentence could contain at some segmentation granularity. Comparing to using words, using word candidates can improve the performance of IR.

Notice that the traditional output of CWS may not fit the purpose of Chinese IR. Due to the lack of explicit word boundaries in Chinese text, there is no commonly accepted definition of words among Chinese speakers and even linguists. The difference is mainly about the segmentation granularity in some situations. That causes the granularity of the words in queries may differ from the ones in the CWS. Indexing only segmented Chinese words can not get a good performance for Chinese IR.

A binary tree based segmentation representation for Chinese IR was introduced by [9]. The result shows that indexing all word candidates at different granularity improves the performance of IR. However, the method proposed by [9] needs a specific annotated corpus and a ranking-based model.

Using the output of our model, we can get all word candidates without any specific annotation on the training corpus. All the word candidates of a sentence correspond to all the nodes in the binary tree of the corresponding sentence. So indexing word candidates benefits Chinese IR as well as the binary tree based method does.

The experiments are on SIGHAN bakeoff 05 [10] corpora in both supervised and unsupervised CWS.

The contributions of this paper are two-fold. (1) We introduce a local generative model. This simple model has a competitive performance in both supervised and unsupervised CWS, while it enjoys a faster learning process and has the ability to make use of larger resources. (2) We find a general way to generate all the possible segmented word at any segmentation granularity, which may benefit IR and other applications.

2 The Local Generative Model

2.1 The Model

We denote the type of interval I by y, which can be either a separated interval s or a combined interval c, and denote the corresponding context consisting of four successive characters $l_2l_1r_1r_2$ by x. Each pair of interval type and its context is treated as a sample (x, y). Examples can be found in Table 2.

Table 2. Part of samples from the sentence '材料利用率低' (The utility rate of the material is low) segmented as '材料 | 利用率 | 低' (material | utility rate | low). The '(-1)' and the '(-2)' are two additional pseudo characters

x (context)	y (type of the interval)
(-2)(-1)材料	s (separated)
材料利用	s
料利用率	c (combined)

The aim of our model is to estimate the generative probability of the samples $p(x, y) = p(y)p(x|y)$. For x is only based on local information and the probability we use is of the generative form, we call our model local and generative.

The priori $p(y)$ is easy to be estimated appropriately by the maximum likelihood estimation. And there could be various way to estimate the likelihood $p(x|y)$, which is the more complicated part. The choice needs to show the relationship between the interval type and the context, and to be dependent on the size of the corpus.

In supervised learning, the estimations we used for both separated interval and combined interval are in the same form

$$\begin{cases} p(x|s) = p_{s1}(l_1, r_1)p_{s2}(l_2|l_1, r_1)p_{s3}(r_2|l_1, r_1) \\ p(x|c) = p_{c1}(l_1, r_1)p_{c2}(l_2|l_1, r_1)p_{c3}(r_2|l_1, r_1) \end{cases} \tag{1}$$

The probabilistic graphical model representation of this model is shown in Figure 1. These two characters, l_1 and r_1, in the middle are first generated together according to the interval type, for they are strongly dependent on the interval type. Then two distinct trigram models are used to generate the outer two characters l_2 and r_2 respectively.

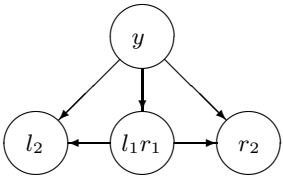

Fig. 1. The probabilistic graphical model representation of the local generative model

Unlike the n-gram language models, these probabilities in the right side of Equation 1 are distinct and estimated individually.

2.2 Predicting

In the learning phrase, the probabilities of the model, $p(y)$, $p(x|s)$, and $p(x|c)$, are estimated. In the predicting phrase, which can be roughly regarded as a binary classification task, for each new context x^* in the training data, a predicted interval y^* can be obtained by the following approach based on Bayesian decision theory. This approach is for both supervised and unsupervised learning.

We define a discriminate function $g(x^*)$ as:

$$g(x^*) = \log \frac{p(s|x^*)}{p(c|x^*)} = \log \frac{p(x^*|s)}{p(x^*|c)} \frac{p(s)}{p(c)} \tag{2}$$

The interval y^* to be predicted can be got by:

$$y^* = \begin{cases} s \text{ , if } g(x^*) > 0 \\ c \text{ , if } g(x^*) < 0 \end{cases} \tag{3}$$

2.3 Unsupervised Learning and the Rules

Following Mochihashi et al. ([6]), we use Gibbs sampling method in the unsupervised learning to learn the model from the raw text.

With slight changes, the local generative model is suitable for the unsupervised Chinese word segmentation, which is usually not feasible for most discriminative models. We also slightly modify the Gibbs sampling method. Details are in the next section.

Three additional rules used in the sampling are introduced below. Wherever these rules are applied, the type of intervals can be determined without any other processes.

The first rule is that any interval is the separated interval, if one of the two immediate characters of this interval is a Chinese character, and the other is a punctuation mark. This idea is from [11], where punctuation marks in raw Chinese text were used as the implicit annotations for unsupervised CWS. So we can say that the punctuation marks make unsupervised CWS semi-supervised. For example, the interval in ', 按' is a separated interval.

Besides, we create the second rule based on Arabic numerals, that any interval is the combined interval, if two immediate characters of this interval are both Arabic numerals. For example, the interval in '1 0' of '又要忙活 1 0 多亩责任田' is a combined interval.

Finally, The third rule is that any intervals at the beginning or at the end of the sequences are all separated intervals.

Table 3 gives us an overview of how the rules work on the raw Chinese text. We found that nearly 1/5 intervals in the raw Chinese text can be determined by the three simple rules in the first three corpora.

Table 3. The statistical information of the SIGHAN bakeoff 05 corpora about the intervals that rules can be applied on

	MSR	CTU	PKU	AS
total intervals	4,325,733	2,525,555	2,020,181	9,289,116
rules can be applied	850,870	492,357	394,889	2,411,779
rate	19.7%	19.5%	19.5%	26.0%

3 Algorithm Detail

3.1 Interpolated Kneser-Ney Smoothing

Unlike the n-gram language model, where we estimate different probabilities for the characters in different positions in the context x, similar smoothing technique is needed in the probability estimation process since the size of corpus is limited. The final performance of this model is sensitive with the smoothing technique.

Here we only discuss the smoothing method for supervised learning. The smoothing method for unsupervised learning is similar. There are two kinds of smoothing needed for the probabilities for supervised learning. One is for the probability of the bigram $p(l_1, r_1)$ in the middle, the other is for the probabilities of the unigram given a bigram, $p(l_2|r_1l_1)$ and $p(r_2|l_1r_1)$. We employ an n-gram language model smoothing method called interpolated Kneser-Ney smoothing to smoothen the probabilities. The notation is mainly based on the technical report by [12].

The probability of x with the interpolated Kneser-Ney smoothing is

$$p(x) = P_{\text{bi}}^{\text{IKN}}(l_1, r_1) P_{r_1,l_1}^{\text{IKN}}(l_2) P_{l_1,r_1}^{\text{IKN}}(r_2) \tag{4}$$

This probability is for both separated and combined intervals, only with different contexts to be estimated.

First, we introduce the smoothed probabilities of the unigram given a bigram, $P_{r_1,l_1}^{\text{IKN}}(l_2)$ and $P_{l_1,r_1}^{\text{IKN}}(r_2)$.

The probability of character unigram c given the context n-gram $\mathbf{u}$ is

$$P_{\mathbf{u}}^{\text{IKN}}(c) = \frac{\max(0, C_{\mathbf{u}c} - d_{|\mathbf{u}|})}{C_{\mathbf{u}\cdot}} + \frac{d_{|\mathbf{u}|} t_{\mathbf{u}\cdot}}{C_{\mathbf{u}\cdot}} P_{\pi(\mathbf{u})}^{\text{IKN}}(c) \tag{5}$$

where $C_{\mathbf{u}'}$ is the count of certain sequence $\mathbf{u}'$ appearing in the training data, $t_{\mathbf{u}\cdot} = |\{c'|c_{\mathbf{u}c'} > 0\}|$ is the number of different characters following the context $\mathbf{u}$, d_i is the discount, $\pi(\mathbf{u})$ is the postfix of the context $\mathbf{u}$ which has length of $|\mathbf{u}| - 1$, and $P^{\mathrm{IKN}}_{\pi(\mathbf{u})}(c)$ is the back-off probability.

For the back-off probabilities, modified counts are used and defined as follows:

$$t_{c'\mathbf{u}c} = \begin{cases} 1 \text{ if} c_{c'\mathbf{u}c} > 0 \\ 0 \text{ if} c_{c'\mathbf{u}c} = 0 \end{cases} \tag{6}$$

$$C_{\mathbf{u}'c} = t_{\cdot\mathbf{u}'c} = \sum_{c'} t_{c'\mathbf{u}c} \tag{7}$$

The back-off probabilities can be defined recursively until the context $\mathbf{u}$ is $\emptyset$. When the context $\mathbf{u}$ is $\emptyset$, we use a maximum entropy estimation as the final back-off probability:

$$P^{\mathrm{IKN}}_{\emptyset}(c) = \frac{\max(0, C_c - d_0)}{C_\cdot} + \frac{d_0 t_\cdot}{C_\cdot} P^{\mathrm{IKN}}_{\mathrm{ME}}(c) \tag{8}$$

The maximum entropy probability is

$$P^{\mathrm{IKN}}_{\mathrm{ME}}(c) = \frac{1}{N} \tag{9}$$

where N is the size of the set of all possible characters.

Then we introduce the smoothed probability of the bigram, $P^{\mathrm{IKN}}_{\mathrm{bi}}(l_1, r_1)$.

When apply the interpolated Kneser-Ney smoothing method to estimate the probability of bigram (l_1, r_1) in the middle, an adapted version is used as:

$$\begin{array}{l} P^{\mathrm{IKN}}_{\mathrm{bi}}(l_1, r_1) = \frac{\max(0, c_{l_1,r_1} - d_2)}{c_\cdot} \\ + \frac{d_2}{t_{l_1,r_1}} c_{l_1,r_1} P^{\mathrm{IKN}}_{\emptyset}(l_1) P^{\mathrm{IKN}}_{\emptyset}(r_1) \end{array} \tag{10}$$

where the back-off probability, which is a product of $P^{\mathrm{IKN}}_{\emptyset}(l_1)$ and $P^{\mathrm{IKN}}_{\emptyset}(r_1)$, assumes that l_1 and r_1 are generated individually. This is quite different from the ordinary n-gram language model back-off method.

The discount d_i can be optimized by holding out a development set or be assigned empirically. The whole learning process for the generative model is to process the samples of contexts and intervals once, without any iteration to optimize any coefficients of the model as the maximum entropy models, CRFs or perceptron methods do. This makes the learning process of our model much faster than those of the discriminative models.

3.2 Adapted Gibbs Sampling

In unsupervised learning, we modify our model and the Gibbs sampling method to learn a model from the raw corpus.

First of all, we slightly change the generative model when we apply this model to unsupervised learning for CWS, for the convergence will be slow and difficult if we use the original model.

The probabilities of the model we use for unsupervised CWS are

$$\begin{cases} p(x|s) = p_s(l_2, l_1)p_s(r_2, r_1) \\ p(x|c) = p_c(l_1, r_1)p_c(l_2|l_1, r_1)p_c(r_2|l_1, r_1) \end{cases} \tag{11}$$

where the probability of the combined intervals is the same as the one for the supervised learning, but the probability of the separated intervals ignores the relationship between these two characters from different words, for this relationship is hard to learn from the raw corpus.

The adapted algorithm of Gibbs sampling [13] is described in Figure 2. There are two changes from the original Gibbs sampling algorithm. The function add() remove() and sample() are the operations for the estimation of the $p(x, y)$ in Gibbs sampling.

Adapted Gibbs sampling algorithm

input x_i, λ
$k := 1000$
$n := 0$
$n_s := 0$
for $t := 1 \ldots T$
  for $i := 1 \ldots I$
    remove(x_i, y_i^{t-1})
    if rule-can-be-applied(x_i) **then**
      $y_i^t :=$ rule(x_i)
    else
      $y_i^t :=$ sample($p(x_i|s)e^{k(n_s - n\lambda)}$,$p(x_i|c)$)
    add(x_i, y_i^t)
    $n := n + 1$
    if $y_i^t = s$ **then** $n_s := n_s + 1$
output $p(y)$,$p(x|y)$

Fig. 2. The adapted Gibbs sampling algorithm

First, three rules introduced in Section 2 are used in this algorithm. The interval types of the samples that those three rules can be applied is not sampled but assigned directly according to those rules.

Another change is for the priori $p(y)$ when we apply the sampling. The Gibbs sampling may cause the priori probability we learn from the data set to be quite different from the real value that we can get from the gold standard segmented corpus, even if we set a priori to the $p(y)$. Thus, we adapt the Gibbs sampling in order to fix the $p(y)$ that the Gibbs sampling could converge to. The total number of intervals n and the number of sampled separated intervals n_s are recorded. And this algorithm uses a term $e^{k(n_s - n\lambda)}$, which can restrict the priori $p(s)$ strongly close to a given value λ, instead of $p(y)$ as the priori in sampling. The strength is controlled by k.

3.3 Word Candidates at Different Granularity

Using the output of our model, we can get all word candidates, at different granularity without any specific annotation on the corpus. The aim of using word candidates is to improve the performance of IR.

For an input sentence $c_1c_2 \dots c_n$, we can calculate the degree of confidence $q(c_i)$ that there is a word boundary after c_i:

$$q(c_i) = g(c_{i-1}, c_i, c_{i+1}, c_{i+2}) \tag{12}$$

A substring $c_a \dots c_b$ of the sentence is a word candidate if and only if:

$$\min\left(q(c_{a-1}), q(c_b)\right) > \max\left(q(c_a), q(c_{a+1}), \dots, q(c_{b-1})\right) \tag{13}$$

That is, if two intervals right before and after a substring are more likely to be the word boundary than any inner intervals in the substring, this substring can be identified as a word at some granularity.

We can define the word as a substring $c_a \dots c_b$ such that

$$\min\left(q(c_{a-1}), q(c_b)\right) > 0 > \max\left(q(c_a), q(c_{a+1}), \dots, q(c_{b-1})\right) \tag{14}$$

The only difference is that there is a zero in the middle of the inequations. This means that words are only generated at a certain granularity which the training corpus has. Interestingly, we can say that the definition of word candidates is the generalization of the definition of words.

For example, given a input sequence '年轻人有学问' (The youngsters have a lot of knowledge) , we have q(年) $= -2.517$, q(轻) $= -2.194$, q(人) $= 2.027$, q(有) $= 1.791$ and q(学) $= -1.644$. For this sequence, all the word candidates are '年', '轻', '人', '有', '学', '问', '年轻', '学问', '年轻人', '有学问' and '年轻人有学问'.

Generally, all the word candidates of a sentence correspond to all the nodes in the binary tree [9] of the same sentence. So indexing word candidates benefits Chinese IR as well as the binary tree based method does.

4 Experiments

4.1 Experiment Setup

All our experiments of supervised and unsupervised learning are on four corpora in SIGHAN bakeoff 05. Two corpora provided by Academia Sinica (AS) and City University of Hong Kong (CTU) are in traditional Chinese, while the other two corpora provided by Peking University (PKU) and Microsoft Research (MSR) are in simplified Chinese. The overview of these corpora are in Table 4. The OOV (out of vocabulary) rate is the rate of the words in the test set that do not appear in the training set.

The discounts in the interpolated Kneser-Ney smoothing need to be assigned. Empirically, the discounts d_0, d_1 and d_2 are assigned to 0.25 0.85 and 0.95

Table 4. Overview of the four corpora in SIGHAN bakeoff 05

	MSR	CTU	PKU	AS
Training set size (words)	2,368,391	1,455,630	1,109,947	5,449,581
test set size (words)	106,873	40,936	104,372	122,610
OOV rate	0.026	0.074	0.058	0.043

respectively for the smoothing of the probabilities for unigram given bigram, and d_0 and d_1 are assigned to 0.4 and 0.75 respectively for the smoothing of the probabilities for bigram. Those values are for all corpora and both supervised and unsupervised learning. Notice that the discounts could also be optimized by holding out a development set.

The f1 measure based on the precision and recall [10] is the commonly used measurement for the evaluation for CWS. We use the f1 measure for our evaluation and comparison.

4.2 Supervised Learning

The learning process for generative models is much faster than the one for discriminative models. The whole learning and predicting process for each corpus terminated in minutes.

Results are shown in Table 5. Here we only concern about the f1 measure for each corpus. It shows that the performances of our model are competitive with the best performances reported by SIGHAN [10], all except one of which are based on discriminative models. Especially on CTU corpus, our performance is better than the best one reported by SIGHAN. And the performances of our model are better than those of the language model [6], which is also a generative model.

There is an interesting observation on the types of segmentation errors. We can adjust the output by replacing some words with corresponding word candidates at finer or coarser granularity according to the gold standard result. The performance could be significantly improved, which is shown in the parentheses of Table 5.

Table 5. Results of supervised learning on SIGHAN bakeoff 05 corpora. The values in the parentheses are the f1 measures that we adjust the result by replacing some words with corresponding word candidates according to the gold standard result to maximize the recall. Results of the language model are from the paper by [6]

method	MSR	CTU	PKU	AS
our model	0.961 (0.995)	0.947 (0.986)	0.945 (0.991)	0.946 (0.990)
language model	0.945	0.941	-	-
SIGHAN reported best	0.964	0.943	0.950	0.952

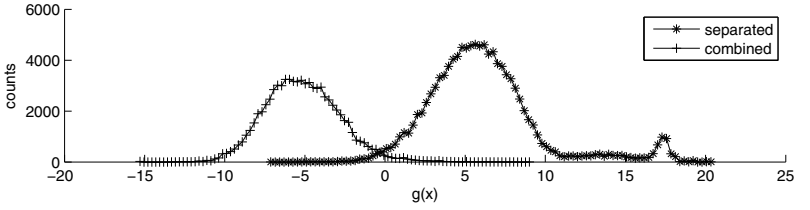

Fig. 3. Distributions of $g(x)$ of the separated and combined intervals on the MSR corpus in supervised learning

Fig. 3 shows the distributions of $g(x)$ for the separated and combined intervals on the MSR corpus in supervised learning. This figure indicates that the $g(x)$ of misclassified samples centers near 0. For the predicting approach is based on the Bayesian decision theory, the value $g(x)$ can be recognized as the degree of confidence. A majority of the types of the intervals could be predicted with high confidence.

4.3 Unsupervised Learning

For unsupervised learning, we directly assign λ in the adapted Gibbs sampling to the value calculated based on the training corpus. In fact, empirical experiments show that this value is not the best estimation for λ.

Table 6 shows the results of experiments on unsupervised learning. The models are learned in two ways, one is to normally use training data, the other is to use test data only for the Gibbs sampling. Figure 4 shows two learning curves of both ways on MSR corpus. The performances of our model are comparable with the state-of-the-art unsupervised method reported by [6].

Similar phenomena occur in the unsupervised learning if we adjust the output using word candidates. The f1 can significantly increase if the result is at a good granularity.

Table 6. Result of unsupervised learning on SIGHAN bakeoff 05 corpora. The values in the parentheses are the f1 measures that we adjust the result by replacing some words with corresponding word candidates according to the gold standard result to maximize the recall. Results of the language model are from the paper by [6].

method	MSR	CTU	PKU	AS
test set only	0.801 (0.941)	0.723 (0.908)	0.774 (0.935)	0.761 (0.937)
training set only	0.819 (0.957)	0.778 (0.951)	0.800 (0.950)	0.796 (0.961)
language model	0.807	0.824	-	-

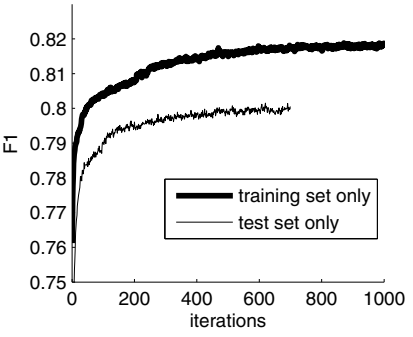

Fig. 4. The learning curve of adapted Gibbs sampling on MSR corpus

5 Discussion and Conclusion

The experiments on SIGHAN bakeoff 05 data show that our model are better than the language model in the supervised learning, and with much faster learning process comparing to discriminative models like CRFs and perceptron. The f1 measure in one of four corpora is better than the best one reported in SIGHAN bakeoff 05. The performance of unsupervised learning of this model is comparable with the state-of-the-art.

In our model, comparing to existent linear discriminative models, information of character trigrams are used, when we calculate the probabilities of l_2 and r_2. And the smoothing technique makes our formulas nonlinear, although there's no coefficients optimization. This may be the reason for why our model can get better performance than the common expectation for the generative models. In addition, four successive characters as the context are proved to be enough for determine a interval type in most of the cases.

For the sake of IR, we define the word candidates for CWS. A previous study already showed that it benefits the performance of IR. The word candidates can be got by our model naturally and directly. Notice that they can also be got by global models such as CRFs, if we redefine the degree of confidence $q(c_i)$ as some marginal probability.

Another interesting result indicated by our experiment is that the errors of CWS are mainly caused by the segmentation granularity difference, without which the performance will increase significantly.

We believe that larger training data or even larger raw data will be helpful to improve the performance. Our model with the advantage shown in this paper is suitable for learning from larger resources. We will work on these issues.

Acknowledgments. This research is supported by the Boeing-Tsinghua Joint Research Project "Robust Chinese Word Segmentation and High Performance English-Chinese Bilingual Text Alignment".

References

1. Xue, N.: Chinese word segmentation as character tagging. Computational Linguistics and Chinese Language Processing 8, 29–48 (2003)
2. Peng, F., Feng, F., McCallum, A.: Chinese segmentation and new word detection using conditional random fields. In: COLING 2004, vol. 1, pp. 562–568 (2004)
3. Gao, J., Li, M., Wu, A., Huang, C.: Chinese word segmentation and named entity recognition: A pragmatic approach. Computational Linguistics 31, 531–574 (2005)
4. Kruengkrai, C., Uchimoto, K., Kazama, J., Wang, Y., Torisawa, K., Isahara, H.: An Error-Driven Word-Character Hybrid Model for Joint Chinese Word Segmentation and POS Tagging. In: 47th Annual Meeting of the ACL, vol. 1, pp. 513–521 (2009)
5. Goldwater, S., Griffiths, T., Johnson, M.: Contextual Dependencies in Unsupervised Word Segmentation. In: 21th Annual Meeting of the ACL, vol. 1, pp. 673–680 (2006)
6. Mochihashi, D., Yamada, T., Ueda, N.: Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling. In: 47th Annual Meeting of the ACL, vol. 1, pp. 100–108 (2009)
7. Sun, M., Shen, D., Tsou, B.: Chinese word segmentation without using lexicon and hand-crafted training data. In: Proceedings of the 17th International Conference on Computational Linguistics, vol. 2, pp. 1265–1271 (1998)
8. Huang, C., Šimon, P., Hsieh, S., Prévot, L.: Rethinking Chinese word segmentation: tokenization, character classification, or wordbreak identification. In: 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, vol. 1, pp. 69–72 (2007)
9. Liu, Y., Wang, B., Ding, F., Xu, S.: Information retrieval oriented word segmentation based on character associative strength ranking. In: The Conference on EMNLP, vol. 1, pp. 1061–1069 (2008)
10. Emerson, T.: The second international chinese word segmentation bakeoff. In: The Fourth SIGHAN Workshop on Chinese Language Processing, vol. 1, pp. 123–133 (2005)
11. Li, Z., Sun, M.: Punctuation as implicit annotations for Chinese word segmentation. Computational Linguistics 35, 505–512 (2009)
12. Teh, Y.: A Bayesian interpretation of interpolated Kneser-Ney. Technical Report (2006)
13. Bishop, C.: Pattern recognition and machine learning. Springer, Heidelberg (2006)