A New Unsupervised Approach to Word Segmentation

Hanshi Wang*
Beijing Institute of Technology

Jian Zhu**
Beijing Institute of Technology

Shiping Tang†
Beijing Institute of Technology

Xiaozhong Fan‡
Beijing Institute of Technology

This article proposes ESA, a new unsupervised approach to word segmentation. ESA is an iterative process consisting of three phases: Evaluation, Selection, and Adjustment. In Evaluation, both the certainty and uncertainty of character sequence co-occurrence in corpora are considered as statistical evidence supporting goodness measurement. Additionally, the statistical data of character sequences with various lengths become comparable with each other by using a simple process called Balancing. In Selection, a local maximum strategy is adopted without thresholds, and the strategy can be implemented with dynamic programming. In Adjustment, a part of the statistical data is updated to improve successive results. In our experiment, ESA was evaluated on the SIGHAN Bakeoff-2 data set. The results suggest that ESA is effective on Chinese corpora. It is noteworthy that the F-measures of the results are basically monotone increasing and can rapidly converge to relatively high values. Furthermore, empirical formulae based on the results can be used to predict the parameter in ESA to avoid parameter estimation that is usually time-consuming.

1. Introduction

Word segmentation is an important task in natural language processing (NLP) for languages without word delimiters (e.g., Chinese). To date, most existing approaches to Chinese word segmentation (CWS) are supervised. Although supervised approaches

* School of Computer Science and Technology, Beijing Institute of Technology, Beijing, 100081 China. E-mail: necrostone@gmail.com.
** School of Computer Science and Technology, Beijing Institute of Technology, Beijing, 100081 China.
† School of Computer Science and Technology, Beijing Institute of Technology, Beijing, 100081 China.
‡ School of Computer Science and Technology, Beijing Institute of Technology, Beijing, 100081 China. E-mail: fxz@bit.edu.cn.

Submission received: 8 December 2009; revised submission received: 14 October 2010; accepted for publication: 18 November 2010.

reach higher accuracy than unsupervised ones in many cases, they involve much more human effort. Furthermore, unsupervised approaches are more adaptive to relatively unfamiliar languages for which we do not have enough linguistic knowledge. In addition, unsupervised approaches can cooperate with supervised ones to overcome drawbacks of both.

Since Sproat and Shih (1990) introduced mutual information (MI) to word segmentation, some researchers have conducted research on unsupervised approaches to word segmentation (Chang and Su 1997). Peng and Schuurmans (2001) proposed an unsupervised approach based on an improved expectation maximum (EM) learning algorithm and a pruning algorithm based on MI. Their approach outperforms soft-counting (Ge, Pratt, and Smyth 1999) that is also based on EM and MI. Non-parametric Bayesian techniques—for example, the Pitman-Yor process (PYP, a generalization of the Dirichlet process, DP) (Pitman and Yor 1997), hierarchical DP (HDP) (Teh et al. 2006; Goldwater, Griffiths, and Johnson 2006), hierarchical PYP (HPYP) (Teh 2006a, 2006b), and hierarchical HPYP (HHPYP) (Wood and Teh 2008)—have been introduced to word segmentation. Mochihashi, Yamada, and Ueda (2009) proposed a novel unsupervised approach based on HPYP, and evaluated it on a part of SIGHAN Bakeoff-2 data set (Emerson 2005). Their evaluation results suggested that their approach outperformed the previous ones.

Some approaches, such as TONGO (Ando and Lee 2000, 2003) and Voting Experts (Cohen, Heeringa, and Adams 2002; Cohen, Adams, and Heeringa 2007), are based on relatively simple ideas. In most cases, an unsupervised approach can be viewed as a kind of goodness measurement to find boundaries between words or filter words from candidates or both. There are four goodness algorithms reviewed by Zhao and Kit (2008a). The algorithms, including Description Length Gain (DLG) (Kit and Wilks 1999), Accessor Variety (Feng et al. 2004a, 2004b), and Branching Entropy (Tanaka-Ishii 2005; Jin and Tanaka-Ishii 2006), were evaluated on SIGHAN Bakeoff-3 data set (Levow 2006).

In this article, we propose ESA, a new unsupervised approach to word segmentation, and demonstrate its effectiveness on Chinese corpora. The approach was motivated by the following considerations:

1. In contrast to the semi-supervised or supervised approaches, we want to find an approach which produces acceptable results under harsh conditions. The harsh conditions are lack of prior knowledge, namely, no lexicons, annotated corpora, or linguistic rules. The acceptability involves comparison with the gold standards, which usually means the manually segmented results.
2. In contrast to existing unsupervised approaches, we want to explore the potential of completely unsupervised approaches. Therefore, we try to avoid any manual interference.

To avoid manual interference, we need to consider the following issues:

1. Unsupervised approaches usually rely on a maximization strategy or thresholds or both. Approaches adopting the maximization strategy alone can be easily adapted to various contexts with few manual adjustments, whereas approaches using thresholds may have a higher accuracy in some cases.

2. Many approaches have constraints on maximum word length. Furthermore, some approaches adopt different strategies for processing words of different lengths. The constraints and the different strategies may improve results on specific languages, however they reduce the generality of the approaches.
3. Many approaches process characters with different strategies according to different character types. The character types are usually identified by encoding information. Because encoding information is prior knowledge of specific languages, a completely unsupervised approach should avoid it as much as possible in order to be applicable under any conditions.

ESA is based on a new goodness algorithm that adopts a local maximum strategy and avoids thresholds. ESA has no constraints on maximum word length. In practice, this kind of constraint can have a negative impact on ESA's segmentation. A simple process called Balancing is introduced to uniformly process words of different lengths. Moreover, ESA uses a self-revision mechanism to improve segmentation accuracy and guarantees convergence after a small number of iterations. In practice, ESA has only one parameter that needs to be configured, and the parameter can be predicted by empirical formulae proposed in this article.

In Section 2, we describe ESA in detail. The SIGHAN Bakeoff-2 archives are available for research on the official Web site and therefore we can easily test ESA on that data. We provide our experimental results and discuss them in Section 3. In Section 4, we compare ESA with other approaches. Finally, we draw our conclusions in Section 5.

2. ESA

ESA consists of Evaluation, Selection, and Adjustment as shown in Figure 1, and it is based on two simple ideas:

1. A better result can be produced by combining certainty and uncertainty. The key is how to combine them.
2. A better result can be produced by adopting the self-revised pattern based on an iterative process.

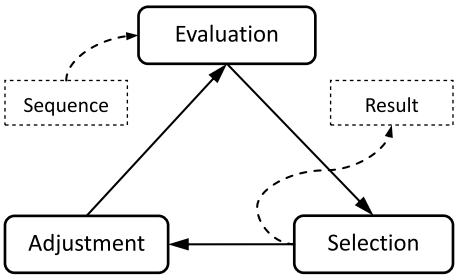


Figure 1
ESA and input/output data.

The input text can be viewed as a character sequence. A character sequence can be divided into two adjacent subsequences. The certainty mentioned previously means certainty of co-occurrence of adjacent subsequences. And the uncertainty means uncertainty of co-occurrence of adjacent subsequences. For example, suppose there are two character sequences, AB and AC. The occurrence of AB represents the certainty of co-occurrence of A and B, whereas the occurrence of AC represents the uncertainty of co-occurrence of A and B, and vice versa. The two kinds of information are combined to evaluate the segmentation. In other words, the decision of whether to segment a character sequence into two adjacent subsequences or not depends on both certainty and uncertainty. An iterative process can produce better results than a non-iterative scheme (Chang and Su 1997). In fact, the current result can be viewed as prior knowledge to adjust the next one.

Devising an unsupervised approach is similar to clarifying how infants segment words without explicit instructions. In particular, infants are able to learn words from various kinds of information such as familiar names (Bortfeld et al. 2005), edges of utterances (Seidl and Johnson 2006), meaning maps (Estes et al. 2007), and auditory forms of words (Swingley 2008). There are a few notable issues:

1. Familiarity (Bortfeld et al. 2005) can be represented by high frequency. Frequent character sequences provide more credibility than infrequent ones. The appearance frequencies are the most important information for word segmentation.
2. The edges of utterances (Seidl and Johnson 2006) can be viewed as natural boundaries. In practice, the boundaries given by punctuation can improve the accuracy of segmentation. However, we think that punctuation should be ignored by completely unsupervised approaches in order to avoid relying on encoding information.
3. Both word lists and statistical data as prior knowledge enable human infants to segment words (Estes et al. 2007). For a completely unsupervised approach, the prior knowledge can be the approach itself and the previous results produced by the approach.
4. The early vocabularies of human infants are based on the sounds of words (Swingley 2008). Some research (Goldwater, Griffiths, and Johnson 2006) is based on phonemes, but the input data are still text.

Before completely clarifying the mechanism of human learning, we tend to believe that machines can understand symbol sequences with simple logic.

2.1 Evaluation

Evaluation is the phase that gives a character sequence or a pair of adjacent subsequences a goodness value according to statistical information. There are three issues to be settled:

1. What are the character sequence and the pair of adjacent subsequences that can be evaluated?
2. What is the necessary statistical information and how do we get it?
3. How do we calculate the goodness?

2.1.1 The Target of Evaluation. A character sequence contains $\frac{(N+1)\times N}{2}$ subsequences, where N is the number of characters in the character sequence. These subsequences are the targets to be evaluated.

A character sequence can be divided into various pairs of adjacent subsequences as shown in Figure 2. A pair of adjacent subsequences contains a gap that is the potential boundary between the two subsequences. Every character sequence has an individual goodness value (IV). Every pair of adjacent subsequences has a combined goodness value (CV) based on the IV of each subsequence and the goodness value of the gap (LRV). IV and LRV indicate certainty and uncertainty of co-occurrence, respectively. Therefore, CV is the combination of certainty and uncertainty. IV is the base of CV, and LRV serves as the modifier of the base.

2.1.2 The Information Needed. The information mentioned here is the statistical information that can be extracted from corpora. There are two basic quantities to be directly measured:

1. The frequency of a character sequence. For example, if the character sequence is ABAB: the frequencies of A, B, and AB are all 2; and the frequencies of ABAB, ABA, BAB, and BA are all 1.
2. The number of character sequences of the same length. For example, if the character sequence is ABC: the sequences of length 1 are A, B, and C; the sequences of length 2 are AB and BC; and the sequence of length 3 is ABC itself. Therefore, the number of the sequences of length 1, 2, and 3 are 3, 2, and 1, respectively.

Furthermore, there are three other quantities that can be calculated according to those just mentioned:

1. The average frequency of character sequences of the same length. For example, there are only two sequences of length 1: A and B. The frequencies of A and B are 2 and 8, respectively. Therefore, the arithmetic mean of frequencies of A and B is 5. In other words, the average frequency of character sequences of length 1 is 5.
2. The entropy of Sequence Plus One (SP1) of a character sequence. For example, consider the character sequence X. The SP1 of X is a set, and each member of the SP1 contains X and one character. In detail, the entropy

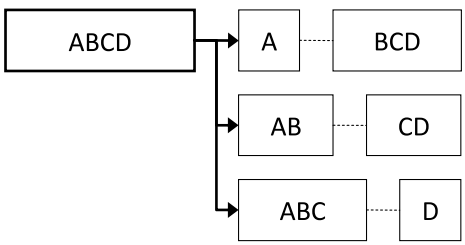


Figure 2
A character sequence and its subsequence pairs.

mentioned here is denoted by H(SP1L(X)) and H(SP1R(X)), where SP1L (SP1 Left) and SP1R (SP1 Right) are two subsets of SP1. In other words, SP1L(X) and SP1R(X) mean that the left side of X is attached with one character and the right side of X is attached with one character, respectively. For example, there are several character sequences: BC, ABC, BBC, BCD, and BCB. ABC and BBC are the members of SP1L(BC), whereas BCD and BCB are the members of SP1R(BC). Therefore, H(SP1L(BC)) is calculated according to the frequencies of ABC, BBC, and other members of SP1L(BC), whereas H(SP1R(BC)) is calculated according to the frequencies of BCD, BCB, and other members of SP1R(BC). The formal descriptions of SP1L and SP1R are

$$SP1L(x) = \{s|s = c \cdot x, s \in S, x \in S, c \in \Sigma\} \quad (1)$$

and

$$SP1R(x) = \{s|s = x \cdot c, s \in S, x \in S, c \in \Sigma\} \quad (2)$$

respectively. The symbol $\cdot$ denotes the attachment operator; s denotes a subsequence of the character sequence S; c denotes a character in the alphabet Σ. In fact, x is the largest proper subsequence of s.

3. The average entropies of SP1s of character sequences of the same length. The SP1s refer to both SP1Ls and SP1Rs. For example, there are only three character sequences of length 2: AB, BC, and CD. Therefore, the sum of H(SP1L(AB)), H(SP1L(BC)), and H(SP1L(CD)) is the numerator of the arithmetic mean, and the denominator is 3. The arithmetic mean is the average entropies of SP1Ls of AB, BC, and CD.

We directly use the prefix tree (trie) (Fredkin 1960) to record the information. Some other data structures can also be used (Morrison 1968; McCreight 1976; Manber and Myers 1990).

2.1.3 The Calculation of Goodness. The *IV* of a character sequence is formulated as

$$IV(x) = (\frac{F_x}{FM_L})^L \quad (3)$$

The superscript L is the exponent; x is the character sequence to be evaluated; L is the length of x. F denotes the frequency of a character sequence, and therefore F_x is that of x; FM denotes the average frequency of character sequences of the same length, and therefore FM_L is that of length L. F can be viewed as a local variable, and FM brings global effects to the formula. By the division in IV, the character sequences of different lengths become comparable with each other. The division is based on a pattern called Balancing, which means keeping balance between the local and global effects. Furthermore, FM is formulated as

$$FM_L = \frac{1}{N} \sum_N F_L \quad (4)$$

The entropies of SP1L and SP1R are formulated as

$$H(SP1L(x)) = -\sum_{i=1}^{n} p(F_{Lx_i}) \ln p(F_{Lx_i}) \tag{5}$$

and

$$H(SP1R(x)) = -\sum_{i=1}^{n} p(F_{Rx_i}) \ln p(F_{Rx_i}) \tag{6}$$

respectively. Lx_i and Rx_i denote the ith members in *SP1L(x)* and *SP1R(x)*, respectively; F_{Lx} and F_{Rx} denote the frequencies of members in *SP1L(x)* and *SP1R(x)*, respectively. The average entropies of SP1Ls and SP1Rs of character sequences of the same length are formulated as

$$HLM_L = \frac{1}{N} \sum_{N} HL_L \tag{7}$$

and

$$HRM_L = \frac{1}{N} \sum_{N} HR_L \tag{8}$$

respectively. *HLM* and *HRM* denote the average entropies of SP1Ls and SP1Rs of character sequences of certain length, respectively; *HL* and *HR* denote the entropies of SP1L and SP1R of a character sequence of a certain length, respectively; *N* is the number of *HL* or *HR*; *L* is the length of character sequences.

The *CV* of a pair of adjacent subsequences is formulated as

$$CV(S_{left} \cdot S_{right}) = IV(S_{left}) \times IV(S_{right}) \times LRV(S_{left} \cdot S_{right}) \tag{9}$$

where the *LRV* is formulated as

$$LRV(S_{left} \cdot S_{right}) = \left(\frac{H(SP1R(S_{left})) \times H(SP1L(S_{right}))}{HRM_{L1} \times HLM_{L2}}\right)^{x} \tag{10}$$

The superscript x is the exponent; The symbol $\cdot$ denotes that S_{left} and S_{right} are two adjacent sequences; $L1$ and $L2$ denote the lengths of character sequences in SP1R(S_{left}) and SP1L(S_{right}), respectively. Actually, $L1$ and $L2$ are equal to the lengths of $S_{\text{left}} + 1$ and $S_{\text{right}} + 1$, respectively. The division in *LRV* represents a Balancing process. As before, the entropies of SP1s of character sequences of different lengths become comparable with each other by using Balancing.

IV and *LRV* represent certainty and uncertainty of co-occurrence of two adjacent character sequences, respectively. Their combination is *CV*. The exponent in *LRV* represents the weight of *LRV* in *CV*.

2.2 Selection

Selection is the phase in which a maximization strategy is used to determine the best segmentation of a character sequence.

For example, the character sequence ABC has six subsequences: A, B, C, AB, BC, and ABC. Therefore, ABC can be divided into various pairs of adjacent subsequences, as shown in Figure 3.

The process of Selection can be viewed as the sets of comparisons. Each set consists of all comparisons between a character sequence itself and all pairs of adjacent subsequences of the character sequence. In Figure 3, the character sequence BC is compared with the pair of adjacent subsequences B · C. If the CV of B · C is higher than the IV of BC, BC will be segmented into B and C.

Processing the longer sequence is done in a similar manner. For example, ABC has two adjacent subsequence pairs: A · BC and AB · C. Both BC and AB have their own adjacent subsequence pairs B · C and A · B, respectively. Therefore, Selection must process BC and AB before processing ABC. The IV of ABC, the CV of A · BC, and the CV of AB · C are compared with each other after Selection finishes processing both BC and AB. The one with the highest goodness value is the final choice.

We adopt a dynamic programming technique to limit the computational complexity of the algorithm in Selection. Therefore, the algorithm takes polynomial time over the length of the character sequence being processed.

2.2.1 The Primary Criterion and Secondary Criteria. The task of Selection is to maximize the goodness value, which can be defined as

$$seg^* = \underset{seg}{\operatorname{argmax}} E(seg) \tag{11}$$

where *seg* denotes all possible segmentations of a character sequence, *seg** denotes the finally selected segmentation, and *E* denotes an evaluation.

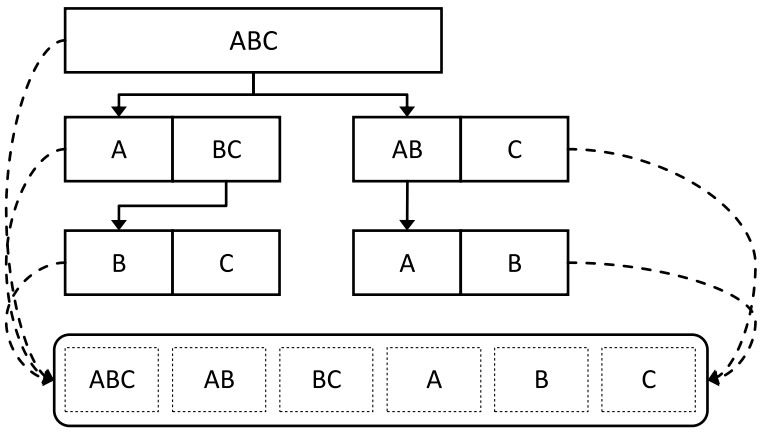


Figure 3
The pairs of adjacent subsequences.

For a character sequence and its pairs of adjacent subsequences, Selection is further formulated as

$$S^* = \underset{0 \le i \le N}{\operatorname{argmax}} E(S_i) \tag{12}$$

where

$$E(S_i) = \begin{cases} IV(S_0) & i = 0, \\ CV(S_i) & 1 \le i \le N-1. \end{cases} \tag{13}$$

S_0 is the character sequence and N is its length; $S_i(1 \le i \le N-1)$ is the pair of adjacent subsequences in S_0, where the second subsequence starts at i. The evaluations for a character sequence and a pair of adjacent subsequences are IV and CV, respectively. The combination of IV and CV is the primary criterion in Selection.

Besides the primary criterion, it is necessary to provide a mechanism for making the decision when the primary criterion cannot resolve it. For instance, when the *IVs* and *CVs* being compared are equal to each other, other criteria are needed. *LRV* is used as a secondary criterion in Selection:

$$E(S_i) = \begin{cases} LRV(S_0) = 1 & i = 0, \\ LRV(S_i) = LRV(S_{i.left} \cdot S_{i.right}) & 1 \le i \le N-1. \end{cases} \tag{14}$$

Although the secondary criteria exist in Selection, their effects are not significant in practice.

2.2.2 The Path of Selection. ESA records the paths of processes of Selection as shown in Figure 4. The result of Selection can be viewed as a binary tree as shown in Figure 5.

2.3 Adjustment

Adjustment is the phase that updates the data. Specifically, it uses the result produced by the previous Selection to update the data that will be used by the next Evaluation. There are two issues to be settled:

1. What data can be updated after the previous Selection?
2. How can we use the updated data for the next Evaluation?

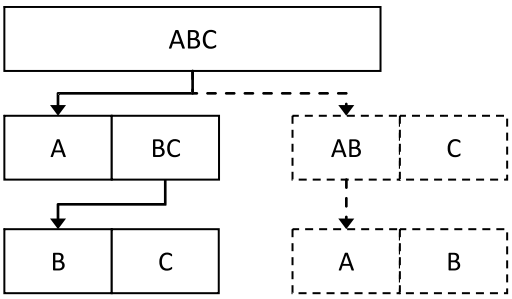


Figure 4
The path of Selection.

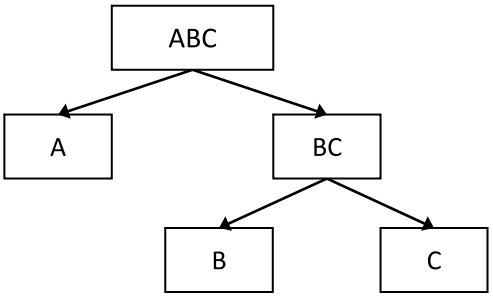


Figure 5
The binary tree of Selection.

2.3.1 Updatable Data. Our idea is that the result of the previous Selection is based on the overestimation of frequencies of some character sequences. For example, when the character sequence X is selected as a word after Selection, the original frequencies of X's proper subsequences are considered as being overestimated because all of them were regarded as potential words before the Selection. In other words, if X is selected as a word, its proper subsequences will not continue to be regarded as potential words. Therefore, the frequencies of these subsequences are reduced, and Adjustment can be viewed as the corrections to the overestimated frequencies.

In our example, the character sequence ABC occurs twice in an input, as shown in Figure 6. After Selection, one of the ABCs is selected as a word but not the other. Therefore, the frequencies of all proper subsequences of the selected ABC are reduced by 1 and those of the other ABC are not changed, as shown in Figure 7.

This process reminds us of Statistical Substring Reduction (SSR) (Zhang et al. 2003; Lü, Zhang, and Hu 2004), although the idea of SSR is not similar to ours. SSR implies that the existence of a character sequence usually has a negative impact on the independent existence of subsequences of this character sequence. If the frequency of a subsequence is near to that of its supersequence, the subsequence will be removed. Frequency of Substring with Reduction (FSR) (Zhao and Kit 2008a) is derived from SSR.

2.3.2 Using Updated Data. The frequency of a character sequence (the *F* in *IV*) is the only quantity to be changed. *FM* in *IV*, *LRV*, and others are not changed.

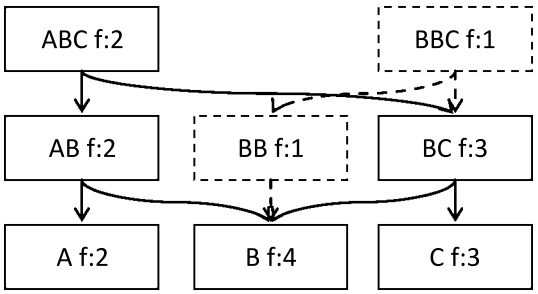


Figure 6
The initial frequencies of character sequences.

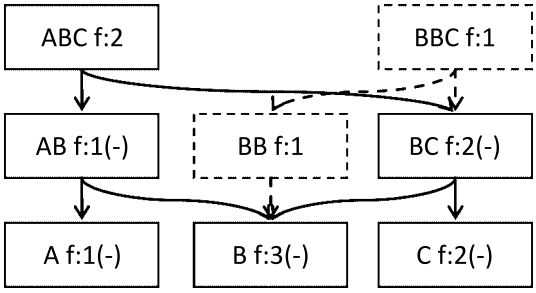


Figure 7
The adjusted frequencies of character sequences.

For example, there are two character sequences: ABC and BBC. The initial records are A(1), B(3), C(2), AB(1), BB(1), BC(2), ABC(1), and BBC(1), where the number in parentheses is the F in IV. After Selection, ABC and BBC are segmented into AB_C and BB_C, respectively. The subsequences of AB are A and B, and the subsequences of BB are two Bs. Because of the idea of Adjustment, the Fs of these subsequences need to be reduced. Therefore, the records become A($1 - 1 = 0$), B($3 - 1 - 2 = 0$), C(2), AB(1), BB(1), BC(1), ABC(1), and BBC(1) after Adjustment.

2.4 Preprocessing Input Data

Sentences are the input data directly accepted by many approaches. The difference between a sentence and a character sequence as discussed in this article is whether or not punctuation is used as prior knowledge to segment text. Some approaches further utilize encoding information. For example, non-Chinese characters such as digits and Latin letters can be separated from Chinese characters so that they can be separately processed.

In ESA, the length of a character sequence is limited for computational complexity. Limiting the length of a character sequence is different from limiting that of a word. In other words, ESA limits the length of the input but not that of the output. In practice, the limitation has a negative impact on the segmentation accuracy of ESA.

2.4.1 Maximum Sequence Length. The time complexity of ESA is polynomial over the length of the character sequence being processed. The preprocessing segments the whole character sequence into multiple sequences within a length limit, and line breaks are regarded as the only natural delimiters. If a character sequence is still longer than the limited length, it will be further divided into two shorter sequences. The location of segmentation (LoS) is determined by the formula

$$LoS(S) = \underset{0<i<L}{\operatorname{argmax}} LRV(s_i^0 \cdot s_{L-i}^i) \tag{15}$$

where S denotes the character sequence to be divided, L is the length of S, i is the index of the character in S, and s denotes the sequence after division. The superscript and the subscript of s denote the start index and the length of s, respectively. If the length of s exceeds the limit, s will be further divided.

The algorithm of *LoS* uses *LRV* alone to segment character sequences before the execution of the main algorithm of ESA. *LRV* alone is inferior to the main algorithm. Therefore, limiting the input character sequence to a short length reduces the effectiveness of the main algorithm.

One-character and two-character words are the most common for Chinese, but words of more than five characters are very rare (Teahan et al. 2000). Therefore, Zhao and Kit (2008a) limited the word length to two or seven in their test. In practice, limiting the maximum word length usually has a positive impact on many approaches.

2.4.2 Encoding Information. There are two levels of encoding information that can be used to improve the results:

1. Punctuation can be used to divide a character sequence into natural sentences.
2. Different types of characters can be separately processed.

2.5 The Result Form

ESA can output the segmentation results in a hierarchical format. Figure 8 shows an example.

The result in the hierarchical format cannot directly be evaluated by the Bakeoff score script. Therefore, the hierarchical format will be changed to a format that looks like A_B_C, where the symbol _ denotes a space character.

2.6 Summary

In this section, we briefly describe the whole algorithm of ESA and provide some thoughts about it.

2.6.1 The Whole Algorithm in Brief. The implementation of ESA consists of four steps:

Step 1: Preprocessing.

Step 2: Evaluation and Selection.

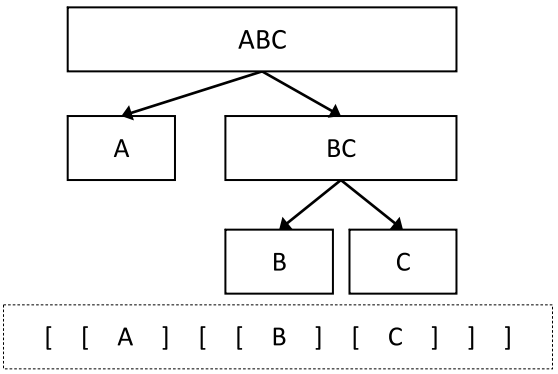


Figure 8
The hierarchical form of a result.

Step 3: Adjustment.

Step 4: Completion. When the current result of segmentation is the same as the previous result, or ESA reaches a given number of iterations, ESA stops. Otherwise, ESA repeats the process from Step 1.

Preprocessing has two steps:

Step 1: Segment a corpus into multiple character sequences. No character sequence exceeds the limit of maximum length.

Step 2: The frequencies (F in IV) of all subsequences in every character sequence are recorded. According to the frequencies, FM in IV and the other quantities in LRV are calculated.

ESA integrates Evaluation and Selection into a recursive algorithm Segment:

Algorithm: Segment.

Input: An entry of data structure. The entry represents a character sequence S.

Output: None.

Comment: L denotes the length of S; FV denotes the final goodness value; FS denotes the final segmentation; $s[i][j]$ denotes the subsequence of S, where i and j denote the start and end indices of s in S, respectively; _ and · denote the delimiter and linkage of character sequences, respectively.

```
01:  If S's FV = 0 then
02:      If L = 1 then
03:          S's FV ← IV(S)
04:          S's FS ← s[1][L]
05:      Else
06:          Max ← IV(S)
07:          Seg ← s[1][L]
08:          For i ← 1 to L
09:              Segment (s[1][i])
10:              Segment (s[i][L])
11:              CV ← s[1][i]'s FV × s[i][L]'s FV × LRV(s[1][i]· s[i][L])
12:              If Max < CV then
13:                  Max ← CV
14:                  Seg ← s[1][i]'s FS·_· s[i][L]'s FS
15:              End if
16:          End for
17:          S's FV ← Max
18:          S's FS ← Seg
19:      End if
20:  End if
```

2.6.2 Discussion. IV is an exponential formula. The base is a ratio, which represents the influence of a segment on the goodness of segmentation. The exponent is the length of the segment, which means that each character in the segment has an equal influence on the goodness.

LRV is also an exponential formula. The base is the product of two ratios, which represents the influence of the gap between two adjacent segments on the goodness of segmentation. The exponent is the weight needed to harmonize the influences of *IV* and *LRV* in *CV*.

CV is the combination of *IV* and *LRV*. The whole goodness of segmentation can be viewed as nested *CVs*. For example, the character sequence ABC is segmented into A_B_C. The whole goodness is equivalent to

$$CV(A \cdot B \cdot C) = IV(A) \times CV(B \cdot C) \times LRV(A \cdot BC) \tag{16}$$

or

$$CV(A \cdot B \cdot C) = CV(A \cdot B) \times IV(C) \times LRV(AB \cdot C) \tag{17}$$

Although the character sequence can possibly be segmented in a different order, the whole goodness is equivalent to

$$CV(A \cdot B \cdot C) = IV(A) \times IV(B) \times IV(C) \times LRV(A \cdot BC) \times LRV(AB \cdot C) \tag{18}$$

Therefore, the design of *CV* ensures the consistency of the goodness measurement across different orders of segmentation.

By using the length as the exponent of *IV*, *CV* has a feature. For example, the character sequence ABC can be segmented into A_B_C, AB_C, A_BC, and ABC. The products of *IVs* in the nested *CVs* are

$$\frac{F_A}{FM_1} \times \frac{F_B}{FM_1} \times \frac{F_C}{FM_1} \tag{19}$$

$$\frac{F_{AB}}{FM_2} \times \frac{F_{AB}}{FM_2} \times \frac{F_C}{FM_1} \tag{20}$$

$$\frac{F_A}{FM_1} \times \frac{F_{BC}}{FM_2} \times \frac{F_{BC}}{FM_2} \tag{21}$$

$$\frac{F_{ABC}}{FM_3} \times \frac{F_{ABC}}{FM_3} \times \frac{F_{ABC}}{FM_3} \tag{22}$$

No matter how it is segmented, the influence of each character is equally considered.

The idea of ESA is to introduce few manually assigned parameters when using the unannotated corpora alone. The only parameter in ESA is the exponent in *LRV*, which can be predicted with the empirical formulae.

We think that a completely unsupervised approach to word segmentation should be tested with the closed criterion in Bakeoff. Furthermore, there are three reasons why the approach should not depend on punctuation to segment sentences:

1. Although punctuation is easily identified with encoding information, it cannot be identified when lacking such information.

2. Segmenting sentences with punctuation is based on the assumption that a language must have punctuation. In fact, some languages, such as ancient Chinese, have no such symbols. This phenomenon even partly exists in modern Chinese, which implies a lack of delimiters between words.

3. The completely unsupervised approach should have more abilities to process unfamiliar languages, because this kind of approach is similar to the infant language learner without prior knowledge such as lexicons, annotated corpora, and character information. The completely unsupervised approach should discover new knowledge instead of just using it.

3. Experiment

The experiment uses the SIGHAN Bakeoff-2 data set that is publicly available on the SIGHAN Web site (www.sighan.org). We tested ESA with various settings.

In this section, we describe the test settings, report experimental results, and discuss those results. According to the experiment, we establish the empirical formulae to predict the exponent in *LRV*.

3.1 Settings

We used four different settings to test ESA:

1. Punctuation and other encoding information are not used, and the maximum length of character sequences is 30. The result with this setting can be viewed as a baseline.

2. Punctuation and other encoding information are not used, and the maximum length of character sequences is 10. The result with this setting demonstrates that the limitation to the maximum length of character sequences has a negative impact on ESA.

3. Punctuation is used to segment character sequences into sentences, and the maximum length of character sequences is 30. The result with this setting demonstrates that punctuation can significantly improve the segmentation accuracy of ESA.

4. Both punctuation and other encoding information are used, and the maximum length of character sequences is 30. The result with this setting demonstrates that discriminating non-Chinese characters from Chinese ones can further improve the accuracy.

A simple algorithm called character-as-word (CAW) (Palmer 1997) was used for the baseline in the paper of Zhao and Kit (2008a). We believe there are two reasons why CAW might be viewed as evidence of the effectiveness of unsupervised approaches:

1. The unsupervised approaches are not comparable with supervised ones in general, because the conventional criteria are the manual segmentations known as gold standards. Gold standards, however,

cannot be unified into a single standard (Fung and Wu 1994; Sproat et al. 1996).

2. The unsupervised approaches are not comparable with each other to some extent. This is not only because the researchers carried out their experiments on different corpora and with test settings, but also because the different approaches may be adapted to different applications (Sproat and Shih 2001; Sproat and Emerson 2003; Wu 2003; Gao et al. 2005).

Zhao (2009) even suggested that character-level analysis could replace word-level analysis for Chinese. However, we think that making comparisons based on similar corpora and settings with other approaches is necessary when regarding word segmentation as an independent task.

3.2 Targets

There are eight corpora consisting of four training and four test corpora in the Bakeoff-2 data set. Because of the different sizes of the corpora and the different settings in the experiment, the number of character sequences (N1) and nodes in trie (N2) produced by ESA are also different, as shown in Table 1.

3.3 Results

The experimental results produced by ESA with the four different settings are shown in Tables 2, 3, 4, and 5, respectively. X denotes the exponent in *LRV*; the numbers in the column headings of the tables are the numbers of iterations; the bold F-measure is almost the best; the asterisked number is the proper exponent.

Table 1
The scales of corpora.

Corpus	Character	Type	Setting 1	Setting 2	Setting 3	Setting 4
CITYU test	67,689	N1	1,428,763	609,921	463,756	406,312
		N2	1,308,964	491,328	363,120	312,151
PKU test	172,733	N1	4,431,621	1,640,688	1,248,317	1,093,046
		N2	3,953,008	1,214,480	896,741	769,674
MSR test	184,355	N1	3,910,003	1,665,511	1,313,351	1,269,858
		N2	3,462,762	1,227,640	946,220	907,226
AS test	197,681	N1	1,840,266	1,353,924	1,305,937	1,210,572
		N2	1,477,908	993,669	992,100	912,221
PKU train	1,826,448	N1	47,565,891	17,430,107	13,227,039	12,721,709
		N2	41,906,011	12,011,275	8,837,184	8,414,698
CITYU train	2,403,354	N1	50,633,595	21,654,982	17,152,484	15,065,762
		N2	43,831,426	15,074,334	11,714,172	9,987,122
MSR train	4,050,469	N1	84,599,783	36,594,096	30,639,574	29,616,260
		N2	71,293,182	23,936,279	20,164,516	19,183,186
AS train	8,368,050	N1	69,454,846	53,877,473	51,986,335	49,455,126
		N2	46,782,170	31,499,015	32,641,248	30,788,602

The results are monotone increasing and rapidly converging in most cases, unless the exponent considerably diverges from the proper value. The larger exponent leads to more insertion errors, whereas the smaller one leads to more deletion errors. On one hand, ESA segments a character sequence into more parts when we increase the exponent in *LRV* (the weight of *LRV* in *CV*), which can produce fewer deletion errors. On the other hand, the character sequence is segmented into fewer parts

Table 2
The results of setting 1 (Punctuation and other encoding information are not used; the maximum length is 30).

	X	1	2	3	4	5	6	7	8	9	10
CITYU test	0.5*	.613	.679	.702	.713	.718	.720	.721	.722	.723	**.723**
	1.0	.586	.647	.669	.683	.688	.693	.695	.696	.697	.697
	1.5	.557	.613	.629	.637	.643	.645	.647	.649	.649	.650
	2.0	.532	.572	.585	.590	.592	.594	.594	.595	.595	.595
PKU test	0.5	.666	.722	.737	.743	.745	.744	.744	.744	.743	.742
	1.0*	.648	.713	.732	.740	.745	.747	.748	.749	.750	**.750**
	1.5	.625	.687	.713	.724	.725	.727	.728	.729	.730	.730
	2.0	.607	.665	.678	.687	.691	.693	.693	.694	.694	.695
MSR test	0.5	.672	.739	.755	.759	.759	.758	.758	.758	.758	.757
	0.9*	.658	.725	.749	.757	.762	.764	.763	.764	.764	**.764**
	1.0	.654	.721	.743	.754	.758	.760	.761	.761	.762	.762
	1.5	.629	.697	.716	.723	.726	.729	.731	.732	.732	.732
	2.0	.606	.668	.686	.694	.697	.699	.699	.700	.700	.700
AS test	0.5	.664	.725	.747	.757	.758	.760	.761	.762	.761	.762
	0.6*	.661	.725	.743	.754	.759	.761	.762	.763	.764	**.764**
	1.0	.640	.704	.729	.739	.744	.747	.748	.749	.749	.749
	1.5	.604	.663	.690	.699	.705	.709	.710	.710	.711	.711
	2.0	.579	.623	.643	.650	.653	.655	.656	.656	.656	.656
PKU train	0.5	.686	.714	.712	.710	.706	.705	.703	.703	.702	.702
	1.0	.690	.729	.733	.734	.734	.733	.733	.732	.732	.732
	1.5	.688	.736	.747	.748	.750	.751	.752	.752	.753	.753
	1.8*	.685	.740	.755	.758	.760	.760	.760	.761	.761	**.761**
	2.0	.682	.734	.750	.755	.758	.759	.759	.759	.759	.760
CITYU train	0.5	.681	.718	.719	.716	.713	.713	.711	.712	.711	.711
	1.0	.683	.728	.737	.739	.739	.739	.738	.738	.738	.738
	1.5	.678	.732	.746	.751	.754	.754	.755	.756	.757	.757
	1.7*	.676	.732	.747	.756	.759	.761	.761	.763	.763	**.764**
	2.0	.666	.727	.745	.753	.756	.758	.759	.761	.761	.762
MSR Train	0.5	.687	.701	.690	.684	.681	.680	.678	.678	.677	.678
	1.0	.696	.722	.720	.717	.714	.713	.712	.712	.711	.711
	1.5	.695	.735	.740	.742	.742	.742	.742	.742	.742	.742
	2.0	.689	.740	.753	.757	.758	.759	.760	.760	.760	.761
	2.5*	.683	.736	.752	.758	.760	.761	.762	.763	.763	**.763**
AS Train	0.5	.670	.659	.647	.641	.639	.638	.637	.637	.637	.637
	1.0	.685	.688	.680	.676	.673	.673	.672	.672	.672	.672
	1.5	.698	.713	.709	.706	.704	.704	.704	.703	.703	.703
	2.0	.701	.726	.731	.733	.734	.734	.734	.735	.734	.734
	2.8*	.699	.739	.750	.753	.755	.754	.755	.755	.755	**.756**

Column headings represent the number of iterations. Boldfaced results represent almost the best F-measure.
* = the proper exponent.

Table 3
The results of setting 2 (Punctuation and other encoding information are not used; the maximum length is 10).

	X	1	2	3	4	5	6	7	8	9	10
CITYU test	0.35*	.614	.676	.694	.701	.705	.704	.705	.706	.706	**.706**
	0.5	.607	.667	.690	.697	.699	.701	.702	.703	.702	.703
	1.0	.577	.636	.656	.669	.674	.678	.679	.680	.680	.681
	1.5	.552	.601	.617	.624	.629	.630	.632	.632	.633	.633
	2.0	.525	.564	.574	.577	.578	.579	.580	.580	.580	.581
PKU test	0.5	.662	.715	.731	.737	.738	.739	.739	.738	.738	.738
	0.65*	.655	.714	.730	.734	.737	.737	.739	.738	.739	**.739**
	1.0	.642	.702	.721	.728	.732	.734	.735	.736	.737	.737
	1.5	.619	.674	.696	.707	.711	.713	.714	.715	.715	.715
	2.0	.598	.648	.663	.667	.671	.673	.674	.675	.676	.676
MSR test	0.5	.668	.733	.748	.753	.753	.754	.754	.753	.753	.753
	0.65*	.662	.729	.745	.751	.753	.754	.754	.754	.754	**.755**
	1.0	.649	.712	.731	.739	.744	.746	.747	.748	.749	.749
	1.5	.622	.686	.706	.712	.716	.718	.719	.719	.719	.720
	2.0	.598	.656	.674	.682	.684	.687	.687	.688	.688	.688
AS test	0.45*	.659	.718	.736	.744	.747	.749	.750	.751	.751	**.752**
	0.5	.656	.716	.734	.742	.745	.747	.748	.749	.749	.750
	1.0	.633	.696	.717	.725	.729	.731	.731	.732	.732	.732
	1.5	.598	.652	.676	.686	.690	.693	.694	.695	.695	.695
	2.0	.574	.615	.631	.639	.642	.642	.643	.643	.643	.643
PKU train	0.5	.700	.734	.737	.737	.734	.735	.733	.734	.733	.734
	1.0	.696	.737	.747	.749	.749	.750	.750	.750	.750	.750
	1.5*	.686	.736	.748	.75	.754	.753	.754	.754	.754	**.754**
	2.0	.677	.728	.743	.747	.750	.749	.751	.750	.752	.750
CITYU train	0.5	.695	.741	.744	.746	.743	.745	.742	.744	.742	.744
	1.0	.689	.739	.750	.755	.755	.756	.755	.757	.755	.757
	1.5*	.679	.733	.746	.755	.756	.760	.759	.761	.760	**.761**
	2.0	.663	.723	.740	.747	.748	.752	.751	.753	.752	.753
MSR Train	0.5	.708	.735	.731	.729	.726	.726	.724	.725	.724	.725
	1.0	.706	.743	.747	.748	.746	.747	.746	.747	.746	.747
	1.5	.698	.744	.753	.756	.756	.757	.756	.758	.757	.758
	1.9*	.690	.740	.751	.757	.757	.760	.759	.760	.759	**.760**
	2.0	.687	.738	.750	.756	.756	.759	.758	.759	.758	.759
AS Train	0.5	.694	.697	.691	.688	.686	.686	.685	.685	.685	.685
	1.0	.702	.716	.714	.713	.712	.712	.712	.712	.712	.712
	1.5	.706	.730	.732	.733	.733	.733	.733	.733	.733	.733
	2.0	.704	.732	.739	.742	.743	.744	.744	.745	.745	.745
	2.8*	.694	.733	.745	.748	.751	.752	.752	.752	.752	**.752**

Column headings represent the number of iterations. Boldfaced results represent almost the best F-measure.
* = the proper exponent.

after a number of iterations, which can produce more deletion errors. ESA produced too many deletion errors with an excessively low weight of *LRV* (such as 0.5 in the test of the AS training corpus), which is why the iteration finally produced a worse result.

Increasing the maximum length of input character sequences magnifies the effect of the main algorithm (*CV*) and reduces that of preprocessing (*LRV* alone). In practice, the results are improved when increasing the maximum length from 10 to 30

Table 4
The results of setting 3 (Punctuation is used; the maximum length is 30).

	X	1	2	3	4	5	6	7	8	9	10
CITYU test	0.3*	.627	.701	.726	.739	.744	.747	.748	.749	.748	**.749**
	0.5	.615	.688	.714	.723	.729	.734	.737	.738	.738	.739
	1.0	.587	.649	.673	.683	.689	.692	.696	.697	.698	.699
	1.5	.553	.606	.629	.639	.642	.644	.645	.645	.645	.646
	2.0	.528	.570	.583	.589	.590	.591	.591	.591	.591	.591
PKU test	0.5*	.673	.739	.759	.767	.771	.774	.774	.774	.774	**.774**
	1.0	.651	.719	.741	.751	.756	.759	.760	.761	.761	.762
	1.5	.623	.687	.713	.724	.731	.735	.736	.737	.737	.737
	2.0	.602	.657	.672	.678	.680	.681	.682	.683	.683	.683
MSR test	0.5*	.680	.753	.773	.779	.781	.782	.783	.783	.783	**.784**
	1.0	.656	.726	.749	.759	.766	.769	.771	.772	.772	.772
	1.5	.627	.693	.716	.724	.728	.730	.731	.732	.732	.732
	2.0	.600	.658	.681	.686	.690	.692	.694	.695	.695	.695
AS test	0.3*	.673	.739	.762	.773	.776	.778	.778	.779	.779	**.779**
	0.5	.664	.732	.753	.764	.770	.773	.775	.776	.776	.777
	1.0	.632	.699	.725	.737	.742	.745	.746	.747	.748	.748
	1.5	.598	.658	.678	.690	.695	.696	.697	.697	.697	.698
	2.0	.575	.619	.633	.641	.644	.645	.646	.646	.646	.646
PKU train	0.5	.736	.777	.783	.783	.782	.782	.781	.781	.781	.781
	1.0	.725	.776	.788	.791	.793	.794	.794	.794	.794	.794
	1.1*	.721	.775	.787	.792	.794	.794	.794	.795	.795	**.795**
	1.5	.710	.767	.782	.788	.791	.792	.792	.792	.793	.793
	2.0	.694	.754	.772	.779	.781	.783	.783	.784	.784	.784
CITYU train	0.5	.740	.790	.797	.797	.796	.796	.795	.795	.795	.795
	1.0	.732	.788	.802	.806	.808	.808	.808	.808	.808	.808
	1.4*	.719	.782	.800	.808	.812	.814	.815	.815	.816	**.816**
	1.5	.714	.778	.797	.806	.810	.812	.813	.814	.815	.815
	2.0	.693	.759	.781	.790	.794	.796	.797	.798	.798	.798
MSR Train	0.5	.744	.775	.774	.771	.770	.769	.768	.768	.768	.768
	1.0	.741	.782	.788	.788	.788	.788	.788	.787	.787	.787
	1.5	.728	.781	.793	.796	.797	.798	.798	.799	.799	.799
	1.6*	.728	.782	.795	.799	.800	.801	.801	.802	.802	**.802**
	2.0	.709	.770	.786	.792	.794	.796	.796	.797	.797	.797
AS Train	0.5	.728	.738	.735	.732	.730	.730	.729	.729	.729	.729
	1.0	.732	.752	.753	.752	.752	.751	.751	.751	.751	.751
	1.5	.732	.762	.766	.768	.769	.769	.769	.769	.769	.769
	2.0	.729	.764	.774	.776	.778	.779	.779	.780	.780	.780
	2.2*	.726	.763	.772	.777	.779	.781	.782	.782	.782	**.782**

Column headings represent the number of iterations. Boldfaced results represent almost the best F-measure.
* = the proper exponent.

as shown in Figure 9. Therefore, the limitation on the maximum length really makes a negative impact on ESA. However, when the maximum length of input character sequences is further increased to 50 or 100, the results are not very different, as shown in Table 6.

Although we insist that a completely unsupervised approach should not rely on encoding information, punctuation and other encoding information are effective in improving segmentation results, as shown in Figure 9.

Table 5
The results of setting 4 (Punctuation and other encoding information are used; the maximum length is 30).

	X	1	2	3	4	5	6	7	8	9	10
CITYU test	0.3*	.635	.709	.735	.748	.754	.757	.758	.759	.760	**.760**
	0.5	.623	.695	.722	.732	.738	.743	.744	.745	.746	.747
	1.0	.596	.656	.682	.693	.699	.702	.705	.706	.707	.707
	1.5	.561	.615	.637	.644	.649	.650	.650	.650	.650	.650
	2.0	.538	.579	.591	.597	.600	.600	.601	.601	.601	.601
PKU test	0.5*	.682	.746	.766	.773	.776	.778	.778	.778	.778	**.778**
	1.0	.659	.728	.749	.759	.764	.766	.768	.769	.770	.770
	1.5	.632	.696	.720	.732	.737	.740	.741	.742	.742	.742
	2.0	.610	.666	.683	.688	.691	.693	.693	.694	.694	.694
MSR test	0.5*	.693	.770	.790	.797	.800	.800	.800	.800	.801	**.801**
	1.0	.670	.741	.767	.777	.783	.786	.788	.789	.789	.789
	1.5	.640	.708	.731	.740	.745	.748	.750	.750	.751	.751
	2.0	.614	.675	.699	.704	.708	.710	.712	.712	.712	.713
AS test	0.3*	.682	.747	.769	.779	.783	.783	.784	.785	.785	**.785**
	0.5	.673	.740	.759	.771	.777	.780	.782	.784	.784	.784
	1.0	.641	.708	.733	.744	.749	.751	.753	.754	.754	.754
	1.5	.606	.666	.686	.698	.703	.704	.705	.705	.705	.705
	2.0	.584	.627	.642	.649	.652	.653	.654	.654	.654	.655
PKU train	0.5	.743	.782	.788	.787	.786	.785	.785	.785	.784	.784
	1.0	.732	.781	.792	.795	.797	.798	.798	.797	.798	.798
	1.2*	.726	.777	.791	.795	.797	.798	.799	.799	.800	**.800**
	1.5	.718	.771	.786	.792	.795	.796	.797	.797	.798	.798
	2.0	.702	.760	.777	.783	.785	.786	.787	.787	.787	.787
CITYU train	0.5	.750	.803	.812	.813	.812	.811	.811	.811	.811	.811
	1.0	.740	.799	.814	.819	.821	.822	.822	.822	.822	.822
	1.4*	.728	.792	.811	.820	.824	.826	.827	.828	.829	**.829**
	1.5	.723	.787	.807	.816	.821	.823	.824	.825	.825	.826
	2.0	.701	.768	.791	.800	.804	.807	.808	.809	.809	.809
MSR Train	0.5	.760	.792	.790	.787	.785	.784	.784	.784	.784	.784
	1.0	.758	.799	.805	.805	.805	.805	.804	.804	.804	.804
	1.5	.745	.799	.811	.814	.815	.815	.816	.816	.816	.816
	1.6*	.742	.797	.810	.814	.816	.817	.817	.818	.818	**.818**
	2.0	.725	.787	.803	.809	.812	.813	.814	.814	.814	.814
AS Train	0.5	.732	.742	.738	.735	.733	.733	.732	.732	.732	.732
	1.0	.736	.755	.756	.755	.755	.755	.755	.754	.754	.754
	1.5	.736	.765	.770	.771	.772	.772	.772	.772	.772	.772
	2.0	.734	.767	.777	.780	.781	.782	.783	.783	.784	.784
	2.2*	.730	.767	.776	.781	.783	.785	.785	.785	.785	**.786**

Column headings represent the number of iterations. Boldfaced results represent almost the best F-measure.
* = the proper exponent.

It is noteworthy that there is a strong correlation between the proper exponent in LRV and the scale of the corpus (N1 and N2), as shown in Figure 10. According to the simple regression analysis, the proper exponents can be approximately predicted. The empirical formulae for the prediction are shown in Table 7. The predictions cannot perfectly fit the proper exponents, but they can be used to produce acceptable results.

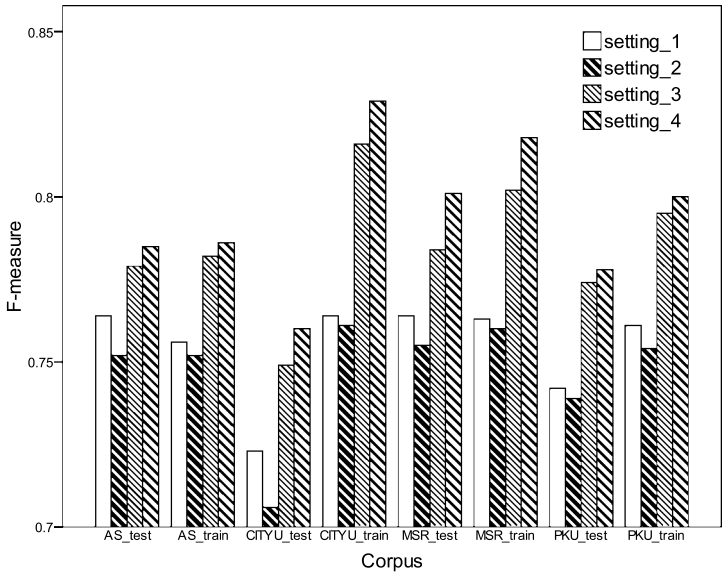


Figure 9
The difference between the results of four settings.

3.4 Discussion

In this section, we discuss the convergence of segmentation results and computational complexity of ESA.

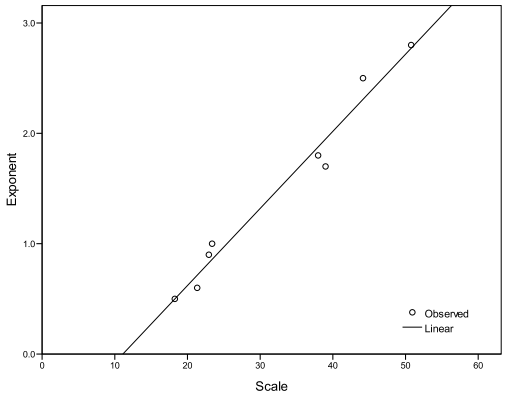
3.4.1 Convergence. In this section, we discuss the convergence of ESA.

Definition 1
For a given character sequence A of length N, there is a set I of all possible segmentations. The *seg* denotes the member of I (i.e., $seg \in I$). There is a sequence S of which each item s_i is a subset of I: $I = \bigcup_0^{N-1} s_i$, $s_i \cap s_j = \varnothing$, $i \in [0,N-1]$, $j \in [0,N-1]$, and $i \neq j$. Therefore,

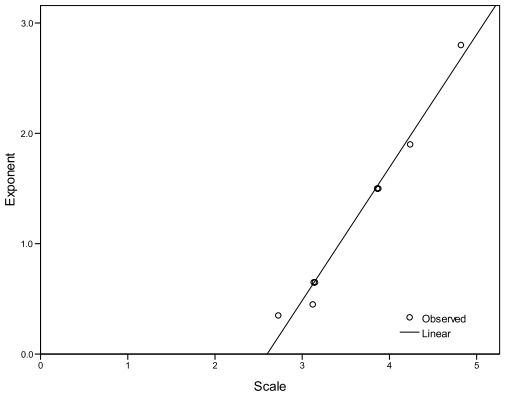
$$s_i = \{seg_k | \textit{The amount of delimitors of } seg_k \textit{ is } i; \; seg_k \in I, k \in [1, \binom{N-1}{i}]\}, i \in [0, N-1] \tag{23}$$

Table 6
The results brought by different maximum lengths.

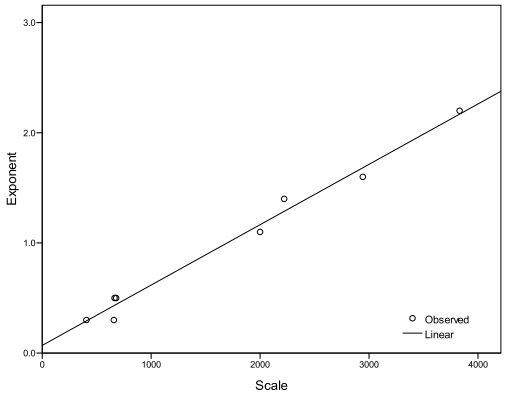
Corpus	10	30	50	100
CITYU test	.706	.723	.726	.726
PKU test	.739	.750	.750	.749
MSR test	.755	.764	.765	.764
AS test	.752	.764	.765	.765



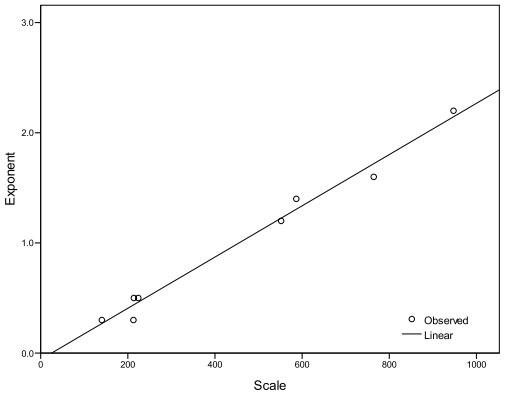
(a) Setting 1 (8 samples. Scale = $N1 \times N2^{-0.8}$)



(b) Setting 2 (8 samples. Scale = $N1 \times N2^{-0.94}$)



(c) Setting 3 (8 samples. Scale = $N1 \times N2^{-0.55}$)



(d) Setting 4 (8 samples. Scale = $N1 \times N2^{-0.63}$)

Figure 10
The correlation between the scales and the proper exponents.

For example, the set I consists of all possible segmentations of the character sequence ABC, namely I = {A_B_C, AB_C, A_BC, ABC}, where _ is a delimiter. There are three items s_0, s_1, and s_2 in the sequence S. Specifically, s_0 = {ABC}, s_1 = {AB_C, A_BC}, and s_2 = {A_B_C}, where the subscript is the number of delimiters contained by the

Table 7
The empirical formulae for the prediction (linear model).

Setting	Equation	df	R^2
1	$P = 0.0699 \times N1 \times N2^{-0.8} - 0.7766$	6	.973
2	$P = 1.2056 \times N1 \times N2^{-0.94} - 3.132$	6	.982
3	$P = 0.0005 \times N1 \times N2^{-0.55} + 0.0696$	6	.986
4	$P = 0.0023 \times N1 \times N2^{-0.63} - 0.0586$	6	.985

In all cases, significance = .000 (i.e., $p < .001$)
P denotes the proper exponent in LRV.
R^2 is the coefficient of determination.
df is the degree of freedom.

segmentation in the item. The subscripts are non-negative integers and they are never equal to each other. Therefore, S can also be viewed as a finite sequence of non-negative integers (i.e., the subscripts 0, 1, 2).

Definition 2
According to Definition 1, there are four types of changes from the current segmentation to the next one, as shown in Figure 11:

> Change 1 denotes that the number of delimiters in the next segmentation is smaller than that in the current one.
>
> Change 2 denotes that the number of delimiters in the next segmentation is larger than that in the current one.
>
> Change 3a denotes that the number of delimiters in the next segmentation is equal to that in the current one, and the locations of delimiters in the next segmentation are identical to those in the current one.
>
> Change 3b denotes that the number of delimiters in the next segmentation is equal to that in the current one, but the locations of delimiters in the next segmentation are different from those in the current one.

Theorem 1
The main algorithm of ESA is a monotone increasing function of F (the F in IV).

Proof
In ESA, the goodness evaluation E of segmentations of a character sequence X is

$$E = \prod CV = \prod_{M1} IV \times \prod_{M2} LRV \tag{24}$$

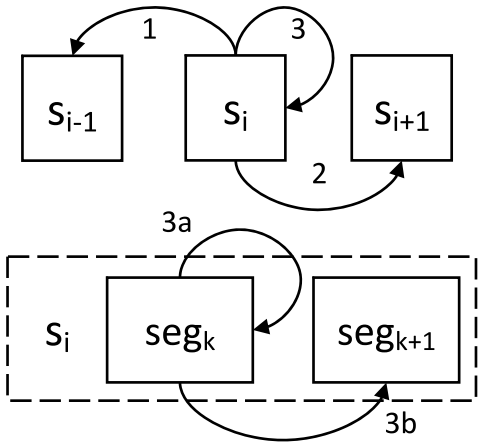


Figure 11
The four types of changes.

where

$$\prod_{M1} IV = \left(\frac{F}{FM}\right)^N \tag{25}$$

$M1$ and $M2$ are the numbers of segments and delimiters in the segmentation, respectively. N is the length of X. LRV, FM, and N are constant for a given X. $M1$ and $M2$ are also constant for a given segmentation of X. Therefore, F is the only variable and E can be viewed as a function of F, that is,

$$E = C \times F^N \tag{26}$$

Both C and N are constant; meanwhile, C, N, and F are positive integers. Consequently, E is a monotone increasing function of F. ■

Theorem 2
For an input character sequence of finite length, the segmentation results produced by ESA converge.

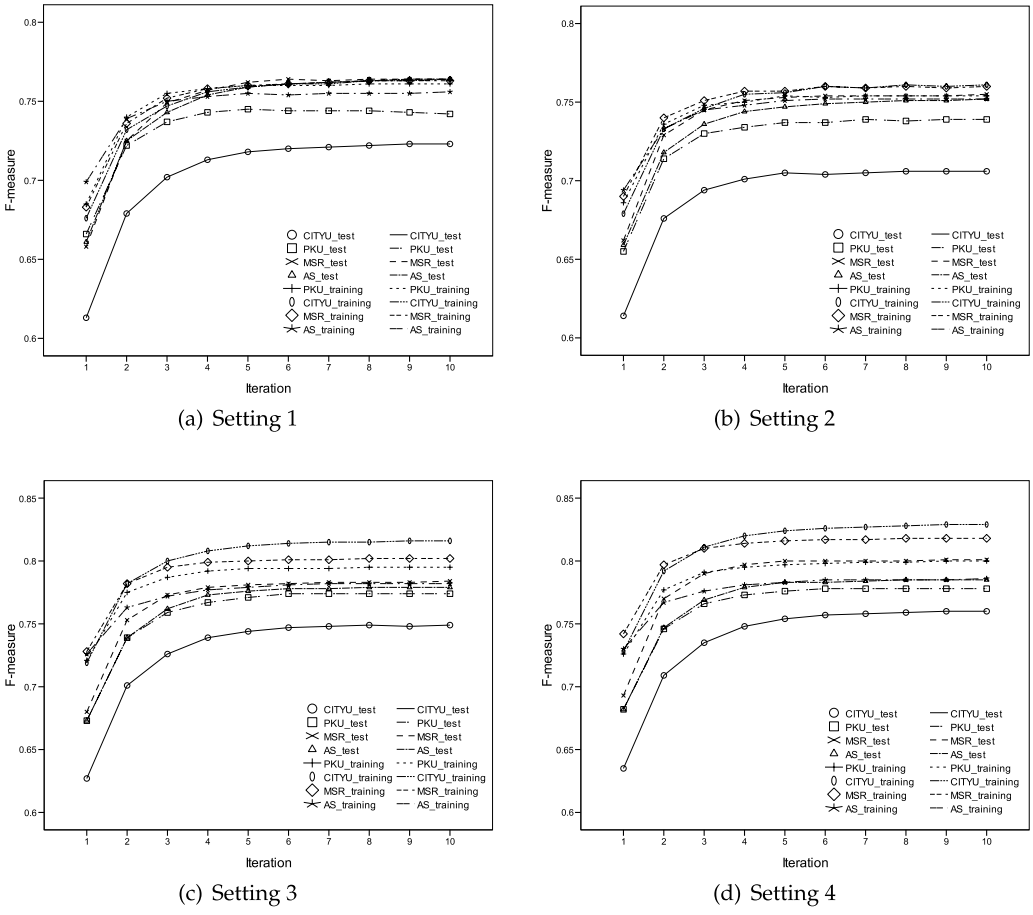
Proof
When s_i and s_{i+1} are the selected result and the discarded result in the current segmentation of X respectively, the goodness value of s_i is larger than that of s_{i+1}, that is, $E(s_i) > E(s_{i+1})$. Additionally, the number of delimiters in s_{i+1} is larger than that in s_i. Adjustment in ESA ensures that the frequencies of segments in the selected result are not changed, and the frequencies of all proper subsequences of the segments are reduced. According to Theorem 1, $E(s_i) > E(s_{i+1})$ is true in the next segmentation. Therefore, change 2 in Definition 2 cannot exist.

For example, the selected result in the current segmentation of a character sequence ABC is AB_C. Therefore, E(AB_C) $>$ E(A_B_C) is true in the current segmentation. The frequencies of subsequences A and B are reduced by 1, whereas those of segments AB and C are not changed. According to Theorem 1, E(AB_C) $>$ E(A_B_C) is true in the next segmentation. Therefore, the selected result in the next segmentation cannot be A_B_C.

The selected results that have the same number of delimiters are regarded as the same segmentation, which means that the results with changes 3a or 3b are viewed as unchanged results by ESA. Therefore, the results produced by ESA are monotone decreasing on the sequence S defined by Definition 1, which means that there are only changes 1 and 3 in ESA. Additionally, S can be viewed as a finite sequence of nonnegative integers, which means that S can also be viewed as a finite sequence of real numbers with a lower bound. According to the monotone convergence theorem, the successive results produced by ESA converge. ■

The experimental results support the conclusion of convergence. ESA approximately converged after five iterations in all cases as shown in Figure 12. Although we cannot prove that the results always converge to the optimum F-measure, ESA ensures that the F-measure is monotone increasing in most cases.

Otherwise, we could head in another direction to explain the convergence: If the iterative process of ESA can be viewed as an EM type, the property of EM will be borne



(a) Setting 1 (b) Setting 2

(c) Setting 3 (d) Setting 4

Figure 12
Convergence of results.

by ESA, which means that ESA theoretically can converge (Dempster, Laird, and Rubin 1977; Wu 1983).

3.4.2 Complexity. The core algorithm, Segment, is implemented with dynamic programming. Each character sequence is only processed once. The total number of processes is $\frac{N\times(N+1)}{2}$, where N is the number of characters in the character sequence. In detail, each character sequence is compared $N-1$ times and is calculated N times including 1 *IV* and $N-1$ *CVs*, which means that the growth rate is O(N^2). Further analyzing the algorithm, the total number of comparisons is

$$\begin{aligned}\sum_{k=1}^{N-1}(N-k)\times k &= N\sum_{k=1}^{N-1}k-\sum_{k=1}^{N-1}k^2\\ &= \frac{N^2(N-1)}{2}-\frac{N(N-1)(2N-1)}{6}\\ &= \frac{N^3-N}{6}\end{aligned} \tag{27}$$

and the total number of calculations is

$$\sum_{k=1}^{N}(N-k+1)\times k = (N+1)\sum_{k=1}^{N}k - \sum_{k=1}^{N}k^2$$
$$= \frac{N(N+1)^2}{2} - \frac{N(N+1)(2N+1)}{6}$$
$$= \frac{N^3+3N^2+2N}{6} \quad (28)$$

Therefore, the time complexity is $O(N^3)$ in the worst case.

For an iterative process, the total complexity is the product of the number of iterations and the complexity per iteration. For example, the complexity of Nested Pitman-Yor Language Model (NPYLM) (Mochihashi, Yamada, and Ueda 2009) is $O(N\times L^2)$ for bigrams, where N is the length of a character sequence and L is the maximum word length accepted. Therefore, the time complexity of NPYLM is $O(N^3)$ when the length of the character sequence and the maximum word length are equal to each other. In addition, NPYLM approximately converges around 50 iterations in the experiment reported, whereas ESA converges around five iterations in our experiment. Therefore, ESA seems to be faster.

In practice, the time complexity of ESA is not greater than $O(N^2)$, as shown in Figure 13, where N is the maximum length of input character sequences. The ESA implementation is a single thread program, and we run it on an AMD Athlon64 system at 2.61GHz.

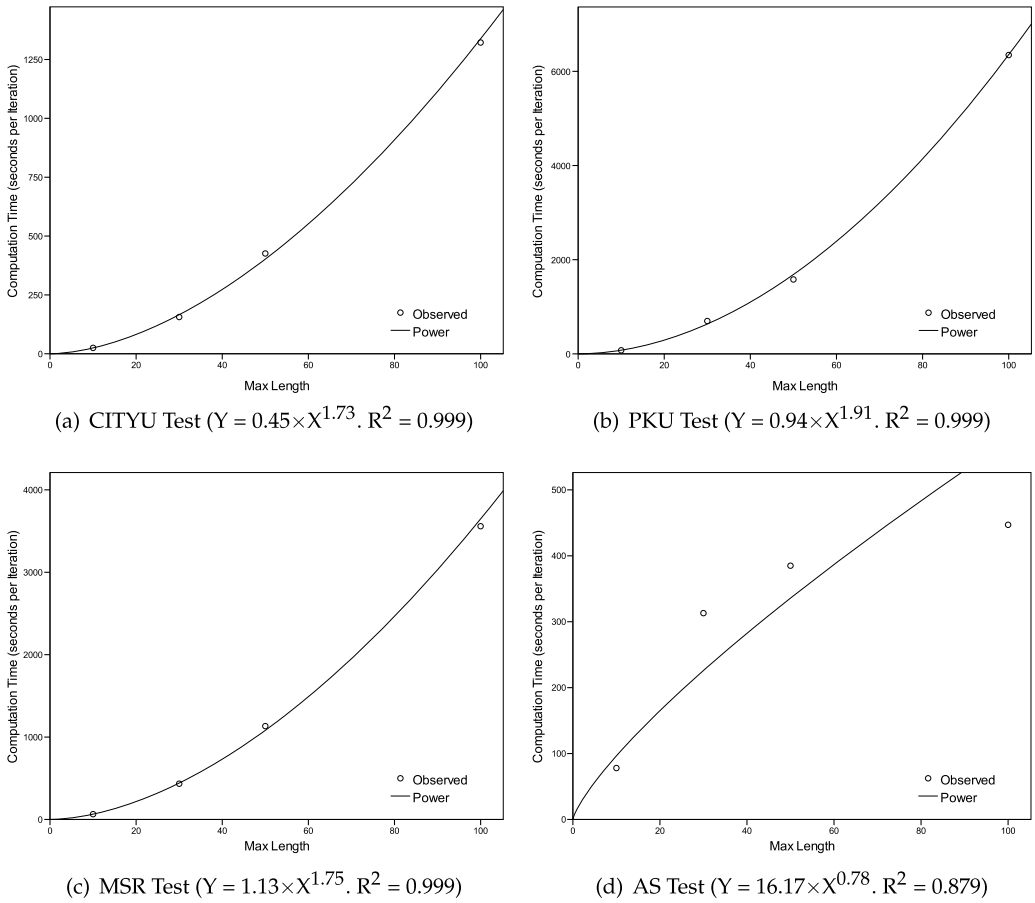
4. Comparison

In this section, we briefly describe some approaches to word segmentation and compare them with ESA. We mainly concentrate on unsupervised approaches because of the motivation for our study.

4.1 Descriptive Comparison

A statistical approach was proposed by Teahan et al. (2000) (TH), which is based on the partial matching (PPM) symbol-wise compression scheme. The approach consists of multiple order models and an escape strategy that is used to transition from the higher order model to the lower one. The approach calculates the escape probabilities and the probabilities of successive characters according to the training corpus that is manually segmented. Therefore, TH is a supervised approach.

Iterative Word Segmentation and Likelihood Ratio Ranking (IWSLRR) (Chang and Su 1997) is an iterative process that uses both certainty and uncertainty information (MI and entropy, respectively). The approach segments words according to an augmented dictionary and adds potential words to the dictionary. The program consists of a segmentation module and a filtering module. The augmented dictionary consists of a system dictionary and the potential words. The system dictionary stores the known words given as prior knowledge. The potential words are produced by merging and filtering. MI and entropy are combined by Gaussian mixture in the filtering algorithm. The filtering algorithm uses a threshold to make the choice for the potential words. In Chang and Su (1997), the system dictionary was the combination of two dictionaries.



(a) CITYU Test ($Y = 0.45 \times X^{1.73}$. $R^2 = 0.999$)

(b) PKU Test ($Y = 0.94 \times X^{1.91}$. $R^2 = 0.999$)

(c) MSR Test ($Y = 1.13 \times X^{1.75}$. $R^2 = 0.999$)

(d) AS Test ($Y = 16.17 \times X^{0.78}$. $R^2 = 0.879$)

Figure 13
The time complexity in practice (4 samples: 10, 30, 50, and 100).

In addition, their approach extracted the potential words from unannotated corpora. Therefore, IWSLRR is semi-supervised. Whereas ESA is completely unsupervised, both IWSLRR and ESA use the combination of certainty and uncertainty. MI and entropy are the measures of two kinds of information in IWSLRR, whereas ESA uses IV and LRV. ESA directly multiplies IV and LRV to combine them, unlike IWSLRR, which uses Gaussian mixture.

Description Length Gain (DLG) (Kit and Wilks 1999) can be viewed as a compression algorithm. The algorithm finds the best substitutes for certain sequences to reduce the description length calculated by the encoding algorithms. The DLG value is negative when replacing the one-character sequence. Therefore DLG cannot uniformly process words of different lengths. That is to say, DLG needs an additional strategy to process one-character words, whereas ESA uniformly processes words of different lengths.

TONGO (Threshold and maximum for *n*-grams that overlap) (Ando and Lee 2000, 2003) counts non-straddling strings on two sides of potential boundaries and straddling strings containing the potential boundaries. To process words of different lengths, the algorithm compares the two types of strings of the same length with each other. The sum of goodness values produced from the comparisons is called the "total vote." If the "total vote" is the local maximum or exceeds a threshold, the location of the boundary

will be determined. The algorithm uses held-out data sets to estimate the maximum order of *n*-grams and the threshold. The held-out data sets are manually annotated. Therefore, TONGO is semi-supervised. The non-straddling and straddling strings can be viewed as uncertainty and certainty, respectively. In addition, the "total vote" is the strategy used to combine the two kinds of information. Therefore, the similarities and differences between ESA and TONGO are similar to those between ESA and IWSLRR.

The Self-Supervised (SS) (Peng and Schuurmans 2001) approach has two parts:

1. Use EM to establish a core lexicon and a candidate lexicon. The selected word candidates move from the candidate lexicon to the core lexicon, which is termed forward selection. The selected word candidates move from the core lexicon to the candidate lexicon, which is termed backward selection. The two kinds of selection are made by a self-improving algorithm. The algorithm is based on the changes of F-measures evaluated by validation corpora.
2. Use MI to split long words in the lexicon.

The pruning algorithm uses two thresholds that are manually assigned, and the validation corpus is manually annotated. Therefore, SS is semi-supervised. The word candidates represent certainty and MI is used to measure uncertainty. These two kinds of information are independently considered by SS, whereas ESA uses a different method to combine them.

Voting Experts (VE) (Cohen, Heeringa, and Adams 2002; Cohen, Adams, and Heeringa 2007) uses logarithm frequency and boundary entropy. Two independent voting experts are based on the two kinds of information, respectively. VE uses a local maximum strategy in a window. The maximum window size limits the word length. To process words of different lengths, VE standardizes frequencies and boundary entropies. A threshold is used to make the final decision on segmentation. VE independently processes the two kinds of information, whereas ESA combines them in CV.

Accessor Variety (AV) (Feng et al. 2004a, 2004b) uses uncertainty between a string and its adjacent characters to assess the string's independence of its context. The counts of right and left adjacent characters are called right and left AV, respectively. The AV of the string is the minimum between the right and left AV. The algorithm uses a local maximum strategy. It uses a threshold to process short words, especially one-character words. Several functions are provided to balance words of different lengths. The AV value is similar to H(SP1) in ESA.

Branching Entropy (BE) (Tanaka-Ishii 2005; Jin and Tanaka-Ishii 2006) is based on a law, namely: The uncertainty of tokens coming after a long character sequence must be lower than that coming after a short character sequence when the long sequence is a supersequence of the short sequence. If the law is broken, there must be boundaries. Specifically, the algorithm has three rules to determine the location of boundaries:

1. The entropy of the location is the local maxima.
2. The entropy of the location is greater than that of the previous location in the same sequence and the difference of the two entropies is greater than a given threshold.
3. The entropy of the location is larger than a given threshold.

BE only uses uncertainty information.

Nested Pitman-Yor Language Model (NPYLM) (Mochihashi, Yamada, and Ueda 2009) is a Hierarchical Pitman-Yor Language Model (HPYLM) (Teh 2006a, 2006b). The base measure of the HPYLM is also a HPYLM. Specifically, NPYLM uses a character HPYLM as the base measure of a word HPYLM. To process words of different lengths, NPYLM uses Poisson distribution to correct the base measure of the character HPYLM. The parameter λ of the Poisson distribution is a variable determined by specific language and word types. In detail, the λ is estimated by a Gamma distribution with two hyperparameters assigned manually. It is noteworthy that the F-measures of this approach were higher than 0.8 on two corpora of Bakeoff-2.

Most approaches use certainty and uncertainty information. Some use only one of them, such as AV and BE. Others use both of them, such as IWSLRR, TONGO, SS, VE, and ESA. The combination of certainty and uncertainty is necessary for the latter approaches, though the specific strategies of combination are different in each approach.

Seeking the maxima and using thresholds are two strategies adopted to make a final decision on segmentation. All of the approaches use either the local maximum strategy, the global strategy, or both. Their difference lies in whether or not they use thresholds. Using thresholds involves more human effort and introduces random factors because of differences in each corpus. ESA avoids using thresholds so that it can be applied to different corpora without much adjustment.

Many approaches have their own strategies to process words of different lengths:

1. The Poisson distribution in NPYLM.
2. Pruning in SS.
3. The restriction on the order of *n*-gram in TONGO.
4. Standardizing in VE.
5. Several functions and thresholds in AV.
6. Ignoring one-character words in DLG.
7. Merging and filtering in IWSLRR.
8. Balancing in ESA.

Some approaches adopt an iterative process, such as NPYLM, SS, IWSLRR, and ESA. The iterative process can often improve the accuracy of an unsupervised approach (Chang and Su 1997). In practice, ESA greatly improves its segmentation results by using an iterative process.

4.2 Quantitative Comparison

In this section, we compare the performance of ESA with that of other approaches, which we can divide into two categories:

1. Unsupervised approaches. NPYLM, AV, BE, and DLG were tested on Bakeoff data sets, and therefore ESA can be directly compared with them. VE was not evaluated on a Bakeoff data set, and therefore we compare ESA with it on a similar scale and setting.
2. Semi-supervised and supervised approaches. We compare ESA with IWSLRR, SS, TONGO, and TH on similar scales and settings.

Mochihashi, Yamada, and Ueda (2009) trained NPYLM on Bakeoff-2 training data and tested it on Bakeoff-2 test data, which means that the statistical information of the training and test data was used together by NPYLM. Therefore, we evaluate ESA on the merged corpora to compare with NPYLM. Specifically, we merge the test corpora with the corresponding training ones. For example, the CITYU test corpus is added to the end of the CITYU training corpus as a single corpus. NPYLM considered specific language and word types. Therefore, we use setting 4 to test ESA, as shown in Table 8. In addition, we cite the best and worst F-measures of supervised approaches in the Bakeoff-2 closed test (Emerson 2005) for comparison in Table 8.

Zhao and Kit (2008a) tested AV, BE, and DLG with both the training and test data of Bakeoff-3 to extract word candidates. Therefore, we evaluate ESA on merged corpora. Zhao and Kit claimed that the approaches were tested without any prior knowledge. Therefore, we use setting 1 to test ESA, as shown in Table 9. In addition, we cite the best and worst F-measures of supervised approaches in the Bakeoff-3 closed test (Levow 2006) for comparison in Table 9.

In Tables 8 and 9, there are two different evaluations: E1 and E2. ESA segments the merged corpora in both of them. The part of a segmentation result belonging to the original test corpus is evaluated alone in E1, whereas the whole of the segmentation result is evaluated in E2.

VE was evaluated on Guo Jin's Mandarin Chinese PH corpus (Cohen, Heeringa, and Adams 2002; Cohen, Adams, and Heeringa 2007). When the average length of

Table 8
The comparison between NPYLM and ESA.

	CITYU	MSR
NPYLM bigram	**.824**	.802
NPYLM trigram	.817	**.807**
ESA E1	**.828**	.819
ESA E2	.804	**.831**
Worst Closed	.759	.896
Best Closed	.943	.966
The exponent in LRV	1.4	1.7

Table 9
The comparison between DLG, AV, BE, and ESA.

	CKIP(AS)	CITYU	UPUC(CTB)	MSRA
DLG	.655	.659	.632	.655
AV	.630	.650	.618	.638
BE	.629	.649	.618	.638
DLG&AV	**.663**	**.692**	**.658**	**.667**
DLG&BE	.650	.689	.650	.656
ESA E1	**.752**	.757	.770	.760
ESA E2	.748	**.764**	**.783**	**.779**
Worst Closed	.710	.589	.818	.819
Best Closed	.958	.972	.933	.963
The exponent in LRV	2.8	1.8	1.4	1.9

words (VE1) was given, the F-measure of VE was 0.77. However, the F-measure was 0.57 without a given length (VE2). Because the scale of the PH corpus is relatively small (19,163 characters), we use the arithmetic mean of F-measures of four Bakeoff-2 test corpora, whose scales are also relatively small, to compare with VE. The window size of VE was six and the punctuation of the corpus was removed in their experiment. Therefore, we use the result of setting 3 to compare with VE, as shown in Table 10.

We use the result of setting 4 to compare both semi-supervised and supervised approaches: IWSLRR (Chang and Su 1997), SS (Peng and Schuurmans 2001), and TONGO (Ando and Lee 2000, 2003) are semi-supervised; TH (Teahan et al. 2000) is supervised. These four approaches used relatively large corpora to train and test, and therefore we use the arithmetic mean of F-measures of four Bakeoff-2 training corpora to compare with them, as shown in Table 11.

The system dictionary in the tests of IWSLRR was the combination of the Academia Sinica dictionary (CKIP 90) and the BDC electronic dictionary (BDC 93). The unannotated Chinese corpus used by IWSLRR contained 311,591 sentences (about 1,670,000 words, a relatively large scale), which came from the *China Times Daily News*. B, T, and Q denote bigrams, trigrams, and quadragrams (i.e., words of 2, 3, and 4 characters), respectively. The result of IWSLRR was achieved after 21 iterations.

SS used segmented text as validation data. The training corpus had 90M characters, which contained one year of the People's Daily news service stories. The test corpus was the Chinese Tree bank from LDC (1M characters), which contained 325 articles from the Xinhua newswire. The maximum length of words in the test of SS was four.

TONGO used segmented text as held-out data. The corpus had 79,326,406 characters, which came from the 1993 Nikkei Japanese newswire. The maximum order of *n*-grams in the test of TONGO was six. However, Ando and Lee (2000, 2003) did not directly present specific F-measures. The F-measure of TONGO was approximately 0.816 in their charts.

TH is typically supervised, and therefore we estimate the accuracy of TH on cross-corpora to compare with ESA. According to the results presented in Teahan et al. (2000), we establish the correlation between error rate and F-measure by using simple regression analysis. The corpora used by TH were Guo Jin's Mandarin Chinese PH corpus containing about 1M words and the Rocling Standard Segmentation Corpus containing about 2M words. L and P in the table denote the estimations of a linear regression model (R^2 = 0.905) and a second order polynomial model (R^2 = 0.95), respectively. T1 denotes training with the PH corpus and testing with the Rocling corpus, and T2 denotes training with the Rocling corpus and testing with the PH corpus.

Table 10
The comparison between VE and ESA.

	VE1	VE2	ESA
F-measure	.770	.570	.781

Table 11
The comparison between IWSLRR(I), SS(S), TONGO(O), TH(T), and ESA.

	I-B	I-T	I-Q	S	O	T1-L	T1-P	T2-L	T2-P	ESA
F-measure	.761	.536	.703	.742	.816	.805	.637	.750	.419	.808

5. Conclusion

This article proposes ESA, an unsupervised approach to word segmentation and demonstrates its effectiveness on Chinese corpora. ESA has no thresholds or parameters estimated. The only parameter (the exponent in LRV) can be predicted via empirical formulae. ESA can produce acceptable results without any encoding information except line breaks. When ESA utilizes prior knowledge such as punctuation and other encoding information, it performs much better.

In practice, unsupervised approaches can take a few steps to improve usability, including:

1. Combine with supervised approaches (Zhao and Kit 2008b) or become a supervised approach (Mochihashi, Yamada, and Ueda 2009).
2. Find more suitable applications for unsupervised approaches (Bod 2006).

Acknowledgments

We thank Dr. Gina-Anne Levow very much for providing the Bakeoff-3 corpora.

References

Ando, Rie Kubota and Lillian Lee. 2000. Mostly-unsupervised statistical segmentation of Japanese: Applications to Kanji. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL'2000)*, pages 241–248, Seattle, WA.

Ando, Rie Kubota and Lillian Lee. 2003. Mostly-unsupervised statistical segmentation of Japanese kanji sequences. *Natural Language Engineering*, 9(2):127–149.

Bod, Rens. 2006. An all-subtrees approach to unsupervised parsing. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL'2006)*, pages 865–872, Sydney.

Bortfeld, Heather, James L. Morgan, Roberta Michnick Golinkoff, and Karen Rathbun. 2005. Mommy and me: Familiar names help launch babies into speech-stream segmentation. *Psychological Science*, 16(4):298–304.

Chang, Jing-Shin and Keh-Yih Su. 1997. An unsupervised iterative method for Chinese new lexicon extraction. *International Journal of Computational Linguistics & Chinese Language Processing*, 1(1):101–157.

Cohen, Paul, Niall Adams, and Brent Heeringa. 2007. Voting experts: An unsupervised algorithm for segmenting sequences. *Intelligent Data Analysis*, 11(6):607–625.

Cohen, Paul, Brent Heeringa, and Niall Adams. 2002. Unsupervised segmentation of categorical time series into episodes. In *Proceedings of the 2nd IEEE International Conference on Data Mining (ICDM'2002)*, pages 99–106, Maebashi.

Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Emerson, Thomas. 2005. The second international Chinese word segmentation Bakeoff. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, pages 123–133, Jeju Island.

Estes, Katharine Graf, Julia L. Evans, Martha W. Alibali, and Jenny R. Saffran. 2007. Can infants map meaning to newly segmented words? Statistical segmentation and word learning. *Psychological Science*, 18(3):254–260.

Feng, Haodi, Kang Chen, Xiaotie Deng, and Weimin Zheng. 2004a. Accessor variety criteria for Chinese word extraction. *Computational Linguistics*, 30(1):75–93.

Feng, Haodi, Kang Chen, Chunyu Kit, and Xiaotie Deng. 2004b. Unsupervised segmentation of Chinese corpus using accessor variety. In *Proceedings of the 1st International Joint Conference on Natural Language Processing (IJCNLP'2004)*, pages 282–288, Sanya, Hainan Island.

Fredkin, Edward. 1960. Trie memory. *Communications of the ACM*, 3(9):490–499.

Fung, Pascale and Dekai Wu. 1994. Statistical augmentation of a Chinese machine-readable dictionary. In *Proceedings of the 2nd Workshop on Very Large Corpora (WVLC-2) at the*

15th International Conference on Computational Linguistics (COLING'1994), pages 69–85, Kyoto.

Gao, Jianfeng, Mu Li, Andi Wu, and Chang-Ning Huang. 2005. Chinese word segmentation and named entity recognition: A pragmatic approach. *Computational Linguistics*, 31(4):531–574.

Ge, Xianping, Wanda Pratt, and Padhraic Smyth. 1999. Discovering Chinese words from unsegmented text. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'1999)*, pages 271–272, Berkeley, CA.

Goldwater, Sharon, Thomas L. Griffiths, and Mark Johnson. 2006. Contextual dependencies in unsupervised word segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL'2006)*, pages 673–680, Sydney.

Jin, Zhihui and Kumiko Tanaka-Ishii. 2006. Unsupervised segmentation of Chinese text by use of branching entropy. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL'2006) Main Conference Poster Sessions*, pages 428–435, Sydney.

Kit, Chunyu and Yorick Wilks. 1999. Unsupervised learning of word boundary with description length gain. In *Proceedings of Ninth Conference of the European Chapter of the Association for Computational Linguistics (EACL'1999): Computational Natural Language Learning (CoNLL'1999)*, pages 1–6, Bergen.

Levow, Gina-Anne. 2006. The third international Chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 108–117, Sydney.

Lü, Xueqiang, Le Zhang, and Junfeng Hu. 2004. Statistical substring reduction in linear time. In *Proceedings of the 1st International Joint Conference on Natural Language Processing (IJCNLP'2004)*, pages 320–327, Sanya, Hainan Island.

Manber, Udi and Gene Myers. 1990. Suffix arrays: A new method for on-line string searches. In *Proceedings of the 1st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'1990)*, pages 319–327, San Francisco, CA.

McCreight, Edward M. 1976. A space-economical suffix tree construction algorithm. *Journal of the ACM*, 23(2): 262–272.

Mochihashi, Daichi, Takeshi Yamada, and Naonori Ueda. 2009. Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-IJCNLP'2009)*, pages 100–108, Suntec.

Morrison, Donald R. 1968. PATRICIA–practical algorithm to retrieve information coded in alphanumeric. *Journal of the ACM*, 15(4):514–534.

Palmer, David D. 1997. A trainable rule-based algorithm for word segmentation. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL'1997)*, pages 321–328, Madrid.

Peng, Fuchun and Dale Schuurmans. 2001. Self-supervised Chinese word segmentation. In *Proceedings of the Fourth International Symposium on Intelligent Data Analysis (IDA'2001)*, pages 238–247, Lisbon.

Pitman, Jim and Marc Yor. 1997. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, 25(2):855–900.

Seidl, Amanda and Elizabeth K. Johnson. 2006. Infant word segmentation revisited: Edge alignment facilitates target extraction. *Developmental Science*, 9(6):565–573.

Sproat, Richard and Thomas Emerson. 2003. The first international Chinese word segmentation bakeoff. In *Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing*, pages 133–143, Sapporo.

Sproat, Richard, William Gales, Chilin Shih, and Nancy Chang. 1996. A stochastic finite-state word-segmentation algorithm for Chinese. *Computational Linguistics*, 22(3):377–404.

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

Sproat, Richard and Chilin Shih. 2001. Corpus-based methods in Chinese morphology and phonology. Unpublished course notes, 2001 Summer Institute of the Linguistic Society of America, in the

Subinstitute on Chinese Corpus Linguistics at the University of California, Santa Barbara, CA.
Swingley, Daniel. 2008. The roots of the early vocabulary in infants' learning from speech. *Current Directions in Psychological Science*, 17(5):308–312.
Tanaka-Ishii, Kumiko. 2005. Entropy as an indicator of context boundaries: An experiment using a Web search engine. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP'2005)*, pages 93–105, Jeju Island.
Teahan, W. J., Rodger McNab, Yingying Wen, and Ian H. Witten. 2000. A compression-based algorithm for Chinese word segmentation. *Computational Linguistics*, 26(3):375–393.
Teh, Yee Whye. 2006a. A Bayesian interpretation of interpolated Kneser-Ney. Technical report TRA2/06, National University of Singapore, School of Computing.
Teh, Yee Whye. 2006b. A hierarchical Bayesian language model based on Pitman-Yor processes. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL'2006)*, pages 985–992, Sydney.
Teh, Yee Whye, Michael I. Jordan, Matthew J. Beal, and David M. Blei. 2006. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581.
Wood, Frank and Yee Whye Teh. 2008. *A hierarchical, hierarchical Pitman-Yor process language model*. In *The 25th International Conference on Machine Learning (ICML'2008) Workshop on Nonparametric Bayes*, Helsinki.
Wu, Andi. 2003. Customizable segmentation of morphologically derived words in Chinese. *International Journal of Computational Linguistics and Chinese Language Processing*, 8(1):1–27.
Wu, C. F. Jeff. 1983. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1):95–103.
Zhang, Le, Xueqiang Lü, Yanna Shen, and Tianshun Yao. 2003. A statistical approach to extract Chinese chunk candidates from large corpora. In *Proceedings of 20th International Conference on Computer Processing of Oriental Languages (ICCPOL'2003)*, pages 109–117, ShengYang.
Zhao, Hai. 2009. Character-level dependencies in Chinese: Usefulness and learning. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL'2009)*, pages 879–887, Athens.
Zhao, Hai and Chunyu Kit. 2008a. An empirical comparison of goodness measures for unsupervised Chinese word segmentation with a unified framework. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing (IJCNLP'2008) Volume-I*, pages 9–16, Hyderabad.
Zhao, Hai and Chunyu Kit. 2008b. Exploiting unlabeled text with different unsupervised segmentation criteria for Chinese word segmentation. *Research in Computing Science*, 33:93–104.