Information Sciences 181 (2011) 163–183

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Integrating unsupervised and supervised word segmentation: The role of goodness measures ☆

Hai Zhao [a,b,1], Chunyu Kit [a,*]

[a] Department of Chinese, Translation and Linguistics, City University of Hong Kong, 83 Tat Chee Ave., Kowloon, Hong Kong SAR, PR China
[b] Department of Computer Science and Engineering, Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai, PR China

ARTICLE INFO

Article history:
Received 21 October 2008
Received in revised form 28 April 2010
Accepted 3 September 2010

Keywords:
Chinese word segmentation
Unsupervised segmentation
Unknown word detection
Conditional random fields
Character tagging
Description length gain
Accessor variety
Boundary entropy

ABSTRACT

This study explores the feasibility of integrating unsupervised and supervised segmentation of Chinese texts for enhancing performance beyond the present state-of-the art, focusing on the critical role of the former in enhancing the latter. Following only a pre-defined goodness measure, unsupervised segmentation has the advantage of discovering many new words in raw texts, but it has the disadvantage of inevitably corrupting many known. By contrast, supervised segmentation conventionally trained only on a pre-segmented corpus is particularly good at identifying known words but possesses little intrinsic mechanism to deal with unseen ones until it is formulated as character tagging. To combine their strengths, we empirically evaluate a set of goodness measures, among which description length gain excels in word discovery, but simple strategies like word candidate pruning and assemble segmentation can further improve it. Interestingly, however, accessor variety and boundary entropy, two other goodness measures, are found more effective in enhancing the supervised learning of character tagging with the conditional random fields model. All goodness scores are discretized into feature values to enrich this model. The success of this approach has been verified by our experiments on the benchmark data sets of the last two Bakeoffs: on average, it achieves an error reduction of 6.39% over the best performance of closed test in Bakeoff-3 and ranks first in all five closed test tracks in Bakeoff-4, outperforming other participants significantly and consistently by an error reduction of 8.96%.

1. Introduction

Word segmentation is a primary task for computer processing of many East Asian languages including Chinese, because in the written forms of these languages word boundaries are not explicitly marked with any overt delimiters such as white space. Written Chinese appears in the form of sequence of characters rather than words. Thus, the task of word segmentation is to convert a text as a sequence of consecutive characters into a sequence of correctly delimited words. It is a special case of tokenization for language processing that involves more complicated problems than one would think at the first glance

☆ The research described in this paper was partially supported by the City University of Hong Kong through the Strategic Research Grants (SRG) 7002037, 7002388 and 7008003 and also by the National Natural Science Foundation of China (NSFC) through the Grand 60903119. The progressive results obtained at various stages of this research were presented disjointedly in a number of conferences and workshops [22,54–57]. Heartfelt thanks are given to the two anonymous reviewers for their insightful comments and advice that have helped improve this article significantly, and also to Olivia Kwong and Lisa Raphals for their helps.

* Corresponding author. Tel.: +852 27889310; fax: +852 27887320.
E-mail addresses: zhaohai@cs.sjtu.edu.cn (H. Zhao), ctckit@cityu.edu.hk (C. Kit).

[1] Supported by a postdoc research fellowship in the Department of Chinese, Translation and Linguistics, City University of Hong Kong.

0020-0255/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2010.09.008

[47,11,30,29]. Various approaches have been explored by scholars to tackle the two main causes of segmentation errors, namely, segmentation ambiguities and unknown (or out-of-vocabulary, OOV) words. Numerous efforts have also been devoted to examine and maximize the effectiveness of various kinds of language resource and linguistics information. For example, the critical role that morphemes and part-of-speeches can play is examined via morpheme-based lexical chunking in a recent study [9]. However, not until recent years have machine learning techniques been successfully applied to this particular task [49,31]. The rapid growth of large scale segmented corpora has brought in essential training data to enable the application of these techniques and the Bakeoffs,[2] in particular, have provided an international forum for examining and comparing their effectiveness in a comprehensive way [37,5,24], in addition to providing indispensable benchmark data sets for verifying and comparing the effectiveness of different supervised and unsupervised learning models.

Conventionally, supervised segmentation assumes a pre-segmented (or labeled) corpus for training a statistical model that can infer the optimal segmentation for an input sentence. Minimally, it assumes a bare vocabulary as a set of known words, either pre-defined or extracted from the pre-segmented training corpus. These words form the structural backbone of the model, and the optimal parameters of the model are to be obtained via training. However, regardless of whether the training is conducted on labeled or unlabeled data, or even both, its purpose is to determine the parameters in association with word sequences, not to determine the words. In principle, this kind of supervised training for a word-based model can hardly bring in any intrinsic mechanism for inferring OOV words. Unfortunately, among the two main causes of segmentation error, OOV words account for several times more errors than ambiguities [15], because a word-based statistical model trained on a labeled corpus of only known words has a strong power of disambiguation via probability estimation but little means to deal with unknown words.

In contrast, unsupervised segmentation does not rely on any given language resource such as a pre-defined vocabulary or a pre-segmented corpus. It is assumed to perform without any labeled data for training. In fact, it is intended to derive a vocabulary from scratch from unsegmented texts, in a way to estimate the likelihood of a substring being a true word by virtue of some pre-defined heuristics or a goodness measure. Usually, a goodness measure for this purpose is formulated in terms of statistical theory to capture our empirical observations of language characteristics in real data, rather than to express theoretical linguistic insights. Thus, it is not a surprise that linguistically-driven heuristic rules are often applied to remedy some idiosyncratic shortcomings of a statistics-based goodness measure for performance enhancement. Furthermore, it is worth noting that unsupervised segmentation is different from, and more complex than, word extraction. The former aims to carry out the segmentation task for a text, for which a decoding algorithm is indispensable; whereas the latter only needs to derive as the final output a list of word candidates from an unsegmented corpus [3,50,6], and hence may or may not involve segmenting the whole corpus into individual words.

Several studies have explored unsupervised segmentation of Chinese texts into words by various means and for various purposes [38,34,10,8,33,40,19]. These studies were formulated in very diverse ways, involving many kinds of heuristic rules. To our knowledge, however, there has not yet been any comprehensive evaluation to examine and compare the performance of different approaches in a consistent way using authoritative large scale "gold standard" data sets, such as the multi-standard ones for the Bakeoffs. Certainly, it is also more than interesting to take a close look at how these approaches correspond with different segmentation standards.

Considering that a statistical goodness measure represents, to a great extent, human observations of the global distributional characteristics of substrings throughout a given corpus of raw texts, one may proceed to exploit such characteristics to facilitate supervised learning of word segmentation. So far, supervised learning has only made use of local information about individual characters and/or substrings within the scope of a sentence, resorting to little global information derived from a whole large scale corpus.

This study explores the role that such global information, derived by various goodness measures, can play in both unsupervised and supervised segmentation. In particular, it focuses on examining four representative goodness measures for word discovery, namely, frequency [27], description length gain [21,20], accessor variety [6,7], and boundary entropy [44,3,16,19], aiming at exploring an effective way of applying them to enhance supervised segmentation.

Each of these measures is integrated into two baseline segmentation frameworks for a comprehensive evaluation. One is a generalized decoding algorithm to realize unsupervised segmentation with a goodness measure as objective function. The other is the conditional random fields (CRFs) model [23] for supervised segmentation via character tagging, conventionally trained only on a pre-segmented corpus. The latter is a state-of-the-art approach that has set new performance records in the field, as illustrated in [52,55], although its efficiency is yet to be further enhanced by various means [58,59]. All scores given by the goodness measures are discretized in the same way for use as feature values in the CRFs model. No other heuristic rules or prior knowledge are involved in this framework, so as to ensure a fair way of comparing the effectiveness of these goodness measures in enhancing the performance of the CRFs model. In this situation, among all evaluation measures in use, the error reduction rates indicating any further improvement over the existing performance records are particularly worth highlighting. All evaluations, for both supervised and unsupervised segmentation, are conducted on the benchmark data sets for Bakeoff-3 [24]. The main reason is that they are significantly larger than others and hence can provide technically more reliable evaluation results.

[2] The International Chinese Word Segmentation Bakeoffs, at http://www.sighan.org/{bakeoff2003,bakeoff2005,bakeoff2006} and http://www.chinalanguage.gov.cn/bakeoff 08/bakeoff-08_basic.html, conventionally referred to as Bakeoff -1, -2, -3 and -4, respectively.

The article is organized as follows. The next section presents the four representative goodness measures for unsupervised segmentation (including unknown word extraction) that are to be integrated into the supervised segmentation using the CRFs model. The experimental settings for their evaluation are given in Section 3. The evaluation results for unsupervised segmentation and its enhancement of supervised segmentation are reported in Sections 4 and 5, respectively. Conclusions are presented in Section 6, including a summary of our major contributions and an outlook for future work.

2. Goodness measure

In principle, both word discovery and unsupervised segmentation without any pre-segmented data for training or any prior knowledge about word form have to resort to some pre-defined criterion for estimating the likelihood of a candidate being a true word, resulting in a goodness score assigned to the candidate. There are also other types of goodness measure for unsupervised segmentation which do not quantify the word likelihood of a candidate. Instead, they indicate how good it is to segment a substring (or join two) at a certain point, e.g., mutual information (MI) [38,28,4,39,50], *t*-test [40], and Ando–Lee criterion [1], to name but a few. Inevitably, however, an *ad hoc* threshold is needed for a decision by any of these measures. The threshold may be manually specified according to one's experience or empirically determined in terms of available data, segmented or unsegmented. Other disadvantages of these measures include: (1) setting a threshold means imposing a brute-force heuristic rule upon a criterion for unsupervised processing; (2) contextual local information has no role to play in determining a segmentation; and (3) more critically, such a measure can only support a binary decision about a possible breaking point, not an optimization over a whole sentence searching for its best segmentation.

This study focuses on a number of representative goodness measures for unsupervised segmentation that allow both local and global information to interact with each other in the optimization process to derive the best segmentation for a sentence without any *ad hoc* threshold setting. Given a goodness measure M, we have a set of word candidates $W = \{\langle w_i, g_M(w_i)\rangle | i = 1, \ldots, n\}$ from an input corpus C of unlabeled texts, where w_i is a substring in C and $g_M(w_i)$ the scoring function by virtue of M. Working on an input sentence, the generalized decoding algorithms to be given in Section 4.1 will assume this setting for each goodness measure given below in this section.

2.1. Frequency

Frequency alone is hardly a sound estimator for how likely a substring is to be a true word, although one may feel that a more frequently occurring substring seems to have a better chance. Statistical substring reduction [27] is an attempt to turn frequency into a workable word-hood criterion. Its underlying assumption is that among two overlapping substrings of the same frequency, the shorter one is redundant and hence can be discarded as a word candidate.

For the purpose of adapting this kind of frequency after substring reduction (FSR) to a decoding algorithm, a goodness scoring function is defined as follows:

$$g_{\mathrm{FSR}}(s) = \log p(s), \tag{1}$$

where $p(s)$ is the frequency of a substring s. That is, the log value of the frequency is assigned to s as its goodness score. The purpose here is certainly not to approximate the number of bits (by log base 2) needed for encoding s, but to fit the frequency to a decoding algorithm in a similar way as the scoring functions for other goodness measures.

2.2. Description length gain (DLG)

This goodness measure is formulated in [21,20] for word discovery via unsupervised segmentation of an input sentence into word candidates that give the optimal sum of compression effect. A compression-based approach to supervised segmentation of Chinese can be found in [42], which identifies words with the greatest compression effect according to a compression model trained, instead, on a pre-segmented corpus.

The DLG resulted from extracting all occurrences of a substring $x_i x_{i+1} \ldots x_j$ (also denoted as $x_{i\ldots j}$) as a word candidate from a corpus $X = x_1 x_2 \ldots x_n$ is defined as:

$$DLG(x_{i\ldots j}) = L(X) - L(X[r \rightarrow x_{i\ldots j}] \oplus x_{i\ldots j}), \tag{2}$$

where $X[r \rightarrow x_{i\ldots j}]$ represents the resultant corpus from replacing all instances of $x_{i\ldots j}$ with a new symbol r throughout X and $\oplus$ denotes the concatenation of two substrings. $L(\cdot)$ is the empirical description length of a corpus in bits. Following the classic information theory [36], it can be estimated in terms of the Shannon–Fano or Huffman coding as:

$$L(X) \doteq -|X| \sum_{x \in V_X} \hat{p}(x) \log_2 \hat{p}(x), \tag{3}$$

where $|\cdot|$ denotes the length of a string in number of characters, V_X is the character vocabulary of X and $\hat{p}(x)$ the frequency of x in X. For a given word candidate w, we define $g_{\mathrm{DLG}}(w) = DLG(w)$.

2.3. Accessor variety (AV)

It is proposed in [6] as a statistical criterion to measure the word likelihood of a substring for word extraction from raw texts of Chinese. The AV of a substring s is defined as:

$$\mathrm{AV}(s) = \min\{L_{av}(s), R_{av}(s)\}, \tag{4}$$

where the left and right accessor variety $L_{av}(s)$ and $R_{av}(s)$ are, respectively, the number of distinct predecessor and successor characters.

This method clearly follows Harris's [13,14] idea for identifying morphemes in an unfamiliar language that a string of phonemes is more likely to be a true morpheme if it appears in a larger number of distinct contexts. It is reported to have particular strength in dealing with low-frequency words. In [6], however, a threshold is set to filter out substrings whose AV values are so low that they are otherwise quantified as potentially meaningful word candidates, and a number of heuristic rules have to be applied to handle candidates consisting of a word and adhesive characters in a strong collocation that the AV is powerless to differentiate from true words. Various ways of combining AV value and word length into an effective target function for unsupervised segmentation are explored in [7].

In order not to diverge from our focus on examining the capacity of AV alone in unsupervised segmentation, all heuristics of these kinds as used for word extraction in [6] are discarded. Similar to the treatment for FSR, the log AV value of a word candidate is used in the decoding over an input sentence for identifying the best segmentation for it. That is, we define $g_{\mathrm{AV}}(w) = \log \mathrm{AV}(w)$ for a word candidate w.

2.4. Boundary entropy (BE)

This criterion is applied to unsupervised segmentation in a number of previous works [44,3,16,19]. The local entropy for a given substring $x_{i..j}$ is defined as:

$$h(x_{i..j}) = -\sum_{x \in V} p(x|x_{i.j}) \log p(x|x_{i.j}), \tag{5}$$

to quantify the average branching uncertainty after (or before) $x_{i..j}$, where $p(x|x_{i..j})$ is the co-occurrence probability of $x_{i..j}$ with character x. There are two types of $h(x_{i..j})$, namely $h_L(x_{i..j})$ and $h_R(x_{i..j})$, corresponding to the left and right directions to extend the substring $x_{i..j}$ [44].

In a similar way to (4), we have $h_{\min}(\cdot) = \min\{h_R(\cdot), h_L(\cdot)\}$. Accordingly, we define $g_{\mathrm{BE}}(w) = h_{\min}(w)$ for a word candidate w involved in the decoding for deriving the optimal segmentation of a sentence in terms of BE.

In principle, Harris's idea underlies both AV and BE; that is, the branching uncertainty is higher at a boundary in between language units such as morphemes or words than at other places. In this sense, it is justifiable to consider these two measures the discrete and continuous formulations of the same idea.

3. Experimental settings

The experiments reported below are conducted on all four data sets from Bakeoff-3 [24], whose corpus sizes in number of both characters and words are summarized in Table 1. Each set consists of a training and a test corpus, for use to train and test a supervised segmentation model respectively. For unsupervised segmentation, however, we will use the raw texts of the training corpora for training, and their annotated versions as the gold standard for evaluation. The main reason for not using the correspond ing test corpora is that the training corpora are much larger in size and hence can produce more reliable statistics and experimental results.

Segmentation performance is evaluated by word F-measure conventionally defined as $F = 2RP/(R + P)$, where the recall R and precision P are, respectively, the proportions of the correctly identified words to all words in the gold standard in question and to those in a segmenter's output. All evaluation results will be presented as F-scores, if not otherwise specified. A scoring tool for calculating these scores is available from the official website of SIGHAN.[3] However, in order to compare with related works by others, we will also use word boundary F-measure $F_b = 2R_bP_b/(R_b + P_b)$ as an auxiliary evaluation metric, where the word boundary recall R_b and precision P_b are defined, respectively, as the proportions of the correctly recognized word boundaries to all boundaries in the gold standard and a segmenter's output [21,1].

The recall of OOV words, R_{OOV}, was used in all previous Bakeoffs as one of the most important performance indicators for supervised segmentation, revealing the capacity and effectiveness of a supervised segmentation strategy in OOV word detection. Nevertheless, it is widely recognized that the F-measure for OOV words, F_{OOV}, defined in a similar way as F and F_b above, is a more comprehensive and less biased metric for assessing the performance of OOV detection. It is hence adopted for this research to supersede the conventional R_{OOV}.[4]

[3] http://www.sighan.org/bakeoff2003/score.

[4] We owe thanks to an anonymous reviewer for pushing hard for this supersedure.

Table 1
Bakeoff-3 corpora.

Corpus		AS	CityU	CTB	MSRA
Training (M)	Character	8.42	2.71	0.83	2.17
	Word	5.45	1.64	0.5	1.26
Test (K)	Character	146	364	256	173
	Word	91	220	154	100

Table 2
Performance comparison.

Max Length	Goodness Measure	Training corpus			
		AS	CityU	CTB	MSRA
2	FSR	0.400	0.454	0.462	0.432
	DLG/d	**0.592**	**0.610**	**0.604**	**0.603**
	AV	0.568	0.595	0.596	0.577
	BE	0.559	0.587	0.592	0.572
7	FSR	0.193	0.251	0.268	0.235
	DLG/d	0.331	0.397	0.409	0.379
	AV	**0.399**	**0.423**	**0.430**	**0.407**
	BE	0.390	0.419	0.428	0.403

4. Unsupervised segmentation

4.1. Decoding algorithms

Two generalized decoding algorithms are formulated for inferring the optimal segmentation for a given plain text, which is usually a sentence at a time. The first one, henceforth referred to as algorithm I, implements a Viterbi search for the best segmentation S^* for a text T as follows:

$$S^* = \underset{w_1 \cdots w_i \cdots w_n = T}{\operatorname{argmax}} \sum_{i=1}^{n} g(w_i), \tag{6}$$

with respect to all $w_i \in T$ and $\langle w_i, g(w_i)\rangle \in W$.

Another one, henceforth algorithm II, mimics the so-called "maximal match" segmentation but is guided by a goodness scoring function instead of word length. It works as follows to carry out a direction-constrained best-first search that repeatedly outputs the best current word w^* at the beginning of T and keeps its remainder $T = t^*$ for the next round of search,

$$\{w^*, t^*\} = \underset{wt=T}{\operatorname{argmax}}\, g(w), \tag{7}$$

with each $\langle w, g(w)\rangle \in W$. This algorithm will back off to the forward maximal match algorithm if the goodness function is set to word length. In this sense, it is a generalization of the latter. Symmetrically, it has a backward version that works the other way around.

An unsupervised segmentation method, henceforth algorithm III, to work with BE is provided in [19]. It works as follows: if $g(x_{i\cdots j+1}) > g(x_{i\cdots j})$ for any two overlapping substrings $x_{i\cdots j}$ and $x_{i\cdots j+1}$, then a segmentation point is located right after $x_{i\cdots j+1}$. This algorithm also has both a forward and a backward version. The union of the segmentation outputs from both versions is taken as the final output of the algorithm. Among the three algorithms given in Jin and Tanaka-Ishii [19], this one proves to give the best performance.[5]

4.2. Performance comparison

Note that a decoding algorithm requires a goodness score for each word candidate involved, including single-character ones. There are two ways to obtain this score, one by a goodness measure in use, if applicable, and the other by default value setting, usually to 0, for special treatment of some cases. For example, all single-character candidates having a negative DLG score will be assigned a default value for the decoding. We will use "/d" to indicate the experiments involving default value setting. This setting can speed up the decoding significantly at the cost of no significant difference in segmentation performance, thanks to its (side-) effect of discarding all substrings that occur only once in the training corpus in question.

[5] We owe thanks to Zhihui Jin and Kumiko Tanaka-Ishii for providing the technical details of their algorithms through personal communications.

Table 3
Performance comparison: AV vs. BE.

Max Length	Goodness Measure	Training corpus			
		AS	CityU	CTB	MSRA
2	$AV_{(I)}$	**0.568**	**0.595**	**0.596**	**0.577**
	$AV_{(II)}$/d	0.485	0.489	0.508	0.471
	$AV_{(II)}$	0.445	0.366	0.367	0.387
	$BE_{(I)}$	**0.559**	**0.587**	**0.592**	**0.572**
	$BE_{(II)}$/d	0.485	0.489	0.508	0.471
	$BE_{(II)}$	0.504	0.428	0.446	0.446
7	$AV_{(I)}$	0.399	0.423	0.430	0.407
	$AV_{(II)}$/d	**0.570**	**0.581**	**0.588**	**0.572**
	$AV_{(II)}$	0.445	0.366	0.368	0.387
	$BE_{(I)}$	0.390	0.419	0.428	0.403
	$BE_{(II)}$/d	**0.597**	**0.604**	**0.605**	**0.593**
	$BE_{(II)}$	0.508	0.431	0.449	0.446

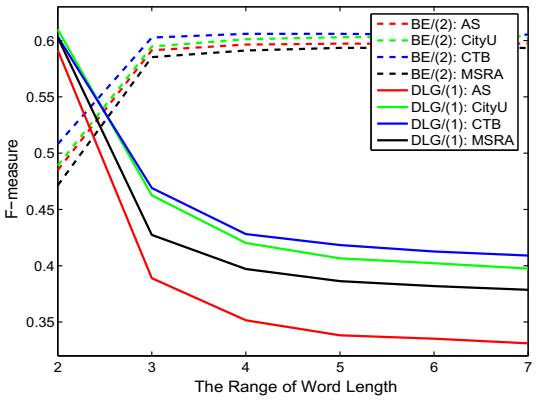

Fig. 1. Performance vs. word length.

Algorithm I is first applied to carry out unsupervised segmentation of the raw texts of all Bakeoff-3 training corpora with the aid of the above goodness measures. Both word candidates and their goodness scores are derived from the raw texts. The performance of these measures is presented in Table 2, where "Max Length" is the maximal length allowable for the word candidates in an experiment and the best performance in a set of experiments on each corpus for comparison purpose is marked in boldface (likewise in other tables hereafter). In this way we examine the effectiveness of a goodness measure in identifying true words of various lengths. The majority of Chinese words are of 1 character or 2, and very few exceed 7. From the table we can see that (1) FSR underperforms significantly, (2) DLG has the strongest performance on short words whereas AV outperforms all others as the max length gets larger, and (3) BE has a performance closely and consistently comparable to that of AV.

Algorithm II performs forward and backward segmentation with the corresponding version of AV and BE measure, i.e., L_{av}/h_L for the backward and R_{av}/h_R for the forward respectively, and then their outputs are merged into one. That is, all segmentation points by either of them count towards the final. A further performance comparison of AV and BE is conducted with this algorithm, and the results are presented in Table 3. We can see that the former has a better performance on shorter words with algorithm I and the latter performs better on longer ones with algorithm II. Interestingly, AV/d and BE/d achieve exactly the same F-scores on all four corpora. How segmentation performance varies along with word length is plotted into curves as in Fig. 1, with DLG and BE as examples. It shows that DLG performs better on words of length 2 and BE on longer ones.

4.3. Word candidate pruning

Although a good number of word candidates are given up by the default goodness threshold 0, the total number of them is very large for any of the four goodness measures, as presented in Table 4. It shows that FSR generates the largest set of word candidates and DLG the smallest. Interestingly, again, AV and BE generate exactly the same candidate lists for all corpora.

Besides word length, the number of word candidates is another crucial factor to affect segmentation performance. The initial candidate set is simply all substrings in the input corpus not exceeding the word length constraint. In fact, it is of little help to include substrings of length >7 in the candidate set, for no more than 0.0001% of Chinese words are of 7 characters or

Table 4
The number of word candidates by default setting.

Goodness Measure	Training Corpus			
	AS	CityU	CTB	MSRA
FSR	2009 K	832 K	294 K	661 K
DLG	543 K	265 K	96 K	232 K
AV	1153 K	443 K	160 K	337 K
BE	1153 K	443 K	160 K	337 K

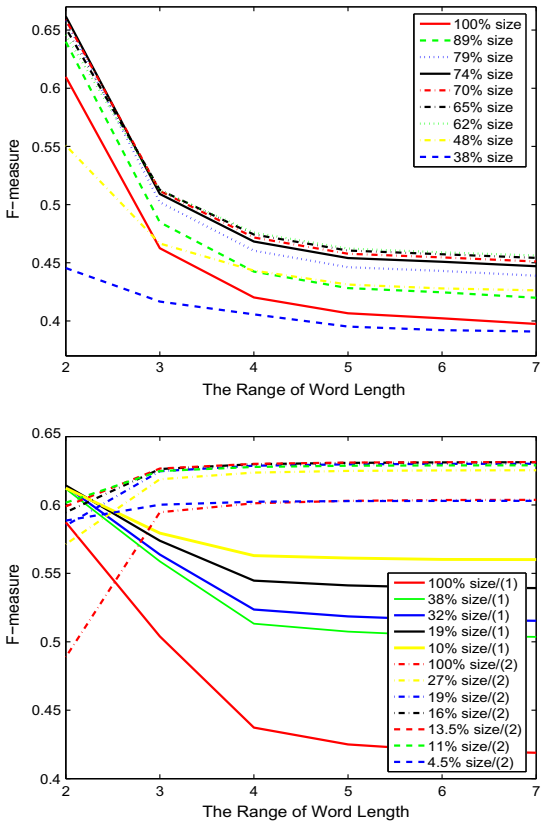

Fig. 2. Performance vs. candidate pruning rate: DLG and BE.

more, according to our statistics from all Bakeoff corpora. Reducing the number of candidates is one of the plausible ways to enhance the quality of the candidate set as a whole. Since each candidate has a goodness score to indicate how good it is, a straightforward way to do so is to prune off those whose scores are too low. To examine how segmentation performance is affected by such pruning and then decide the optimal pruning rate, we conduct a series of experiments with each goodness measure. Fig. 2 presents two of them, one with DLG (on the left) and the other with BE, both on CityU training corpus. It indicates that appropriate pruning does lead to significant improvement with little change of these measures' respective advantages on two-character words and longer ones. Interestingly, each goodness measure has a stable and similar performance in a range of pruning rates around the optimal one, e.g., 62–79% for DLG and 11–19% for BE, as illustrated in Fig. 2.

The optimal pruning rates found through experiments for the four goodness measures are presented in Table 5 and their corresponding performance in Table 6, showing a remarkable improvement over the default setting. Again, the pruning does not affect the advantage of DLG on two-character words and AV/BE on longer ones. Among the four measures, DLG achieves the best overall performance because of its particular strength in dealing with short words. Without doubt, it is the overwhelming number of two-character words in Chinese that allows it to triumph.

4.4. Ensemble unsupervised segmentation

Although word candidate pruning by a proper percentage cut does bring about promising performance improvement, it is infeasible to determine the optimal pruning rate in practice for an arbitrary unlabeled corpus without knowing its gold

Table 5
Optimal rates (%) of candidate pruning.

Decoding algorithm	Goodness measure			
	FSR	DLG	AV	BE
(I)	1.8	70.0	12.5	20.0
(II)	–	–	8.0	12.5

Table 6
Performance via candidate pruning by optimal percentage cut.

Max Length	Goodness Measure	Training corpus			
		AS	CityU	CTB	MSRA
2	$FSR_{(I)}$	0.501	0.525	0.513	0.522
	$DLG_{(I)}/d$	**0.710**	**0.650**	**0.664**	**0.638**
	$AV_{(I)}$	0.616	0.625	0.609	0.618
	$BE_{(I)}$	0.613	0.614	0.605	0.611
	$AV_{(II)}/d$	0.585	0.602	0.589	0.599
	$BE_{(II)}/d$	0.591	0.599	0.596	0.593
7	$FSR_{(I)}$	0.444	0.491	0.486	0.486
	$DLG_{(I)}/d$	0.420	0.447	0.460	0.423
	$AV_{(I)}$	0.517	0.568	0.549	0.544
	$BE_{(I)}$	0.501	0.539	0.510	0.519
	$AV_{(II)}/d$	0.623	0.624	0.604	0.615
	$BE_{(II)}/d$	**0.630**	**0.631**	**0.620**	**0.622**

Table 7
Performance of ensemble segmentation.

Max Length	Goodness Measure	Training corpus			
		AS	CityU	CTB	MSRA
2	$FSR_{(I)}$	0.629	0.635	0.624	0.623
	$DLG_{(I)}/d$	**0.664**	**0.653**	**0.643**	**0.650**
	$AV_{(I)}$	0.641	0.644	0.631	0.634
7	$BE_{(I)}$	0.640	0.643	0.632	0.634
	$AV_{(II)}/d$	0.595	0.637	0.624	0.610
	$BE_{(II)}/d$	0.593	0.635	0.620	0.609
	$DLG_{(I)}/d + AV_{(II)}/d$	**0.672**	**0.684**	**0.663**	**0.665**
	$DLG_{(I)}/d + BE_{(II)}/d$	0.660	0.681	0.656	0.653

standard segmentation in advance. A parameter-free method to tackle this problem is to let the available goodness measures to do the pruning for each other, so as to obtain a better candidate set than they can individually.

A straightforward method for this is to have the intersection of individual candidate sets as pruning result. An underlying assumption here is that the agreement of several sound criteria may come up with a more reliable decision than any individual one of them. As the above four goodness measures are concerned, only DLG and AV (or BE) are needed in order to get this result, because under the default setting both AV and BE result in the same set of candidates and DLG generates only a subset of what FSR does, according to our observations.

Whether this intersection set of word candidates is really better than those before the reciprocal pruning can be verified by putting it into experiments of optimal segmentation with each of these goodness measures, to see if any further improvement can be achieved. The results of our experiments are presented in Table 7, showing that algorithm I achieves a marvelous improvement on short words with all goodness measures except DLG, as compared to Table 6. Interestingly, however, DLG still keeps its performance at the top despite some slip-back.

More importantly, however, is it possible to push for any further improvement by virtue of this intersection set, the best candidate set obtainable so far in an unsupervised manner? A possibility for this certainly lies in how well the strengths of various goodness measures can be put together so as to remedy their weaknesses reciprocally. Since DLG and AV/BE are known to be particularly strong in recognizing two- character words and longer ones respectively, but relatively weak the other way around, a simple strategy to combine them is to enforce all words of length >2 in the AV/BE segmentation upon the corresponding parts in the DLG segmentation. As expected, this ensemble method gives a better overall performance than all others that have been tried so far, as presented at the bottom of Table 7.

Table 8
Performance comparison of decoding algorithms on the CTB, MSRA and PKU corpus.

Max Length	Goodness Measure	CTB		MSRA		PKU	
		/d	–	/d	–	/d	–
2	$AV_{(I)}$	0.535	0.546	0.519	0.529	0.529	0.538
	$AV^{*}_{(I)}$	**0.607**	0.607	**0.612**	0.612	**0.616**	0.616
	$AV_{(II)}$	0.440	0.381	0.417	0.361	0.429	0.374
	$AV^{*}_{(II)}$	0.538	0.538	0.522	0.522	0.531	0.531
	$AV_{(III)}$	0.382	0.084	0.362	0.084	0.375	0.085
	$AV^{*}_{(III)}$	0.458	0.458	0.431	0.431	0.441	0.441
	$BE_{(I)}$	0.533	0.544	0.516	0.526	0.526	0.534
	$BE^{*}_{(I)}$	**0.607**	0.607	**0.610**	0.610	**0.615**	0.615
	$BE_{(II)}$	0.440	0.476	0.417	0.475	0.429	0.488
	$BE^{*}_{(II)}$	0.538	0.538	0.522	0.522	0.531	0.531
	$BE_{(III)}$	0.382	0.440	0.362	0.465	0.375	0.469
	$BE^{*}_{(III)}$	0.458	0.458	0.431	0.431	0.441	0.441
6	$AV_{(I)}$	0.360	0.368	0.331	0.345	0.313	0.325
	$AV^{*}_{(I)}$	0.407	0.407	0.393	0.393	0.372	0.372
	$AV_{(II)}$	0.571	0.381	0.570	0.361	0.588	0.373
	$AV^{*}_{(II)}$	**0.645**	0.645	**0.651**	0.651	**0.663**	0.663
	$AV_{(III)}$	0.382	0.377	0.363	0.432	0.376	0.453
	$AV^{*}_{(III)}$	0.461	0.461	0.435	0.435	0.445	0.445
	$BE_{(I)}$	0.357	0.363	0.327	0.337	0.309	0.319
	$BE^{*}_{(I)}$	0.405	0.405	0.391	0.391	0.370	0.370
	$BE_{(II)}$	0.602	0.485	0.611	0.488	0.624	0.501
	$BE^{*}_{(II)}$	**0.651**	0.651	**0.667**	0.667	**0.676**	0.676
	$BE_{(III)}$	0.382	0.572	0.363	0.614	0.376	0.624
	$BE^{*}_{(III)}$	0.463	0.463	0.437	0.437	0.447	0.447

4.5. Comparison of algorithms

In [19], algorithm III works with BE and is evaluated with the PKU Corpus[6] of 1.1 M words as gold standard, but its word candidate set is extracted from the 200 MB Contemporary Chinese Corpus of several years of the People's Daily.[7] For the purpose of comparison, we carry out an evaluation with a similar setting to compare this algorithm with the other two, using the raw texts of three training corpora of Bakeoff-3 in GB code, including the PKU corpus, with word candidates extracted from the unlabeled texts of People's Daily (1993–1997), of 213 MB and about 100 M characters. With the aid of the AV/BE criteria, a total of 4.42 M candidates up to 6-characters[8] long are extracted.

The evaluation results are presented in Table 8, with "*" indicating reciprocal pruning of word candidates, in contrast to the default pruning by goodness score >0 as before. A performance comparison in terms of boundary F-measure on the PKU corpus is also given in Table 9.[9] Unfortunately, both tables provide no evidence in favor of algorithm III. Even more undesirable is that this algorithm gains no significant performance improvement via candidate pruning.

4.6. Comparison with supervised segmentation

Empirical evidence is provided in [15] for an estimation of the degree to which the four segmentation standards involved in Bakeoff-3 differ from each other. As quoted in Table 10, a consistency rate above 84.8% is found among the four Bakeoff-3 corpora. It seems realistic not to expect unsupervised segmentation to surpass the agreement of these gold standards, and thus reasonable to take this figure as the topline of its performance. On the other hand, it is shown in [53] that words up to 2 characters long account for more than 90% of all words in Chinese texts and single-character words alone for about 50%. Thus, it is reasonable to take as the baseline the performance of the brute-force guess of every single-character as a word.

Furthermore, unsupervised segmentation needs to be conducted in a comparable manner in order to ensure a fair comparison with supervised segmentation. The latter usually involves training on a pre-segmented training corpus (optionally joined with the raw texts of a test corpus) and then evaluation on the test corpus. For each test track of Bakeoff-3, we first extract word candidates from the raw texts of both the training and test corpora with the aid of the above reciprocal pruning,

[6] http://icl.pku.edu.cn/icl_groups/corpus/dwldform1.asp.
[7] http://ccl.pku.edu.cn:8080/ccl_corpus/jsearch/index.jsp.
[8] This is to keep consistent with [19], where the maximum *n*-gram length is set to 6.
[9] The best boundary precision, recall and F-score reported in [19] are 0.88, 0.79 and 0.833 respectively, whereas our re-implementation of the same algorithm gives a slightly higher F-score 0.837 under a similar experimental setting.

Table 9
Performance comparison by boundary F-measure on the PKU corpus.

Max Length	Goodness Measure	Algorithm					
		I/d	I	II/d	II	III/d	III
2	AV_*	0.758	0.763	0.747	0.762	0.762	0.382
	AV^*	**0.821**	0.821	0.803	0.803	0.781	0.781
	BE_*	0.756	0.760	0.747	0.803	0.762	0.750
	BE^*	**0.820**	0.820	0.803	0.803	0.781	0.781
6	AV_*	0.695	0.700	0.830	0.762	0.762	0.728
	AV^*	0.728	0.728	**0.865**	0.865	0.783	0.783
	BE_*	0.696	0.699	0.849	0.810	0.762	0.837
	BE^*	0.728	0.728	**0.872**	0.872	0.784	0.784

Table 10
Consistency rate among Bakeoff-3 segmentation standards.

Test corpus	Training corpus			
	AS	CityU	CTB	MSRA
AS	1.000	0.926	0.959	0.858
CityU	0.932	1.000	0.935	0.849
CTB	0.942	0.910	1.000	0.877
MSRA	0.857	0.848	0.887	1.000

Table 11
Comparison of performance against supervised segmentation.

Max Length	Goodness Measure	Test corpus			
		AS	CityU	CTB	MSRA
Baseline		0.389	0.345	0.337	0.353
2	$DLG_{(I)}/d$	0.597	0.616	0.601	0.602
	$DLG^*_{(I)}/d$	0.655	0.659	0.632	0.655
	$AV_{(I)}$	0.577	0.603	0.597	0.583
	$AV^*_{(I)}$	0.630	0.650	0.618	0.638
	$BE_{(I)}$	0.570	0.598	0.594	0.580
	$BE^*_{(I)}$	0.629	0.649	0.618	0.638
7	$AV_{(II)}/d$	0.512	0.551	0.543	0.526
	$AV^*_{(II)}/d$	0.591	0.644	0.618	0.604
	$BE_{(II)}/d$	0.518	0.554	0.546	0.533
	$BE^*_{(II)}/d$	0.587	0.641	0.614	0.605
	$DLG^*_{(I)}/d + AV^*_{(II)}/d$	**0.663**	**0.692**	**0.658**	**0.667**
	$DLG^*_{(I)}/d + BE^*_{(II)}/d$	0.650	0.689	0.650	0.656
	Worst closed	0.710	0.589	0.818	0.819
	Best closed	0.958	0.972	0.933	0.963

and then perform unsupervised segmentation of the test corpus as guided by each goodness measure. The segmentation outputs are evaluated with reference to the gold standard of each track respectively.

The evaluation results are presented in Table 11, together with the best and worst official results of Bakeoff-3 closed test for a comparison. Unsurprisingly, the unsupervised segmentation is far from being capable of competing against its supervised counterpart, according to the difference of their performance. However, these experiments confirm a positive effect that the best combination of goodness measures achieve an F-score in the range of 0.65–0.7 on all test corpora involved, without using any prior knowledge other than the pre-defined goodness measures for extracting word candidates from unlabeled texts.

4.7. Discussion

Note that the DLG criterion is applied to perform segmentation so as to maximize the overall compression effect concerning all substrings involved, which is a global effect throughout the entire corpus of texts in question. Thus it works well when

incorporated into a framework for probability maximization, favoring highly frequent but independent substrings, in particular, the long ones. Many unsupervised segmentation criteria are prone to the issue of long word bias, i.e., over preferring long substrings; so is DLG. This explains why its performance on long words is not as good as on short ones.

Both the AV and BE measures estimate the branching uncertainty at an end of a given substring. They are more concerned with local uncertainty information about a substring, instead of any kind of global effect on a whole corpus resulted from having the substring as a word. This may explain why greedy search via maximal matching can benefit more from these two measures than the Viterbi search can.

Clearly, these two measures share a similar underlying idea. In our view, they actually present different formulations, one discrete and the other continuous, of the same phenomenon, to which we refer as branching uncertainty. The fact that both AV and BE derive an identical set of word candidates from the same corpus by the default pruning certainly provides a piece of empirical evidence to support this view.

5. Integration of unsupervised and supervised segmentation

5.1. Baseline system

The baseline system of supervised learning for Chinese word segmentation that we attempt to further enhance with the aid of unsupervised segmentation is a state-of-the-art one that achieved the best performance in all closed tests in the last two Bakeoffs [53,55]. It is a successful application of the conditional random fields (CRFs) model [23] to formulate the problem of word segmentation as a supervised learning task through character tagging.

CRFs are a statistical sequence modeling framework that is reported to outperform other popular learning models, including MaxEnt (maximum entropy), in a number of natural language processing applications [35]. CRFs modeling is first applied to Chinese word segmentation in [31], treating it as a binary decision task to determine whether a character is the beginning of a word in a sentence. Accordingly, to segment a sentence is to infer the optimal sequence of such decisions, via tagging (or labeling), for all characters in the sentence in terms of a CRFs model that has been trained on an available pre-segmented corpus.

A CRFs model assigns the following probability to a label sequence for a sequence of characters, which is usually a sentence,

$$P_\lambda(y|s) = \frac{1}{Z}\exp\left(\sum_{c\in C}\sum_{k}\lambda_k f_k(y_c, y_{c-1}, s, c)\right), \tag{8}$$

where y is a label sequence, s a character sequence, Z a normalization term, f_k a feature function with λ_k as its weight, C the tag (or label) set in use, and c indexes into a character in the character sequence to be tagged. For CRFs learning, we use the CRF++ package,[10] with necessary modifications to speed up its training on large scale pre-segmented corpora.

It is shown in [53] that the CRFs learning achieves a better performance of word segmentation for Chinese with a 6-tag set than with any others. Hence this tag set, in company with six selected feature templates of character n-gram, is adopted in our current work for establishing the baseline system. The six tags are B, B_2, B_3, M, E and S, to mark the first, second, third, other middle, ending and sole character in a word respectively. Accordingly, we have the following tag sequences S, BE, BB_2E, BB_2B_3E, BB_2B_3ME and $BB_2B_3M\cdots ME$ for characters in a word of length $1,2,\ldots,5$ and 6 (or above) respectively. The six feature templates are C_{-1}, C_0, C_1, $C_{-1}C_0$, C_0C_1 and $C_{-1}C_1$, where the current, previous and next character are indexed by 0, -1 and 1 respectively. A template is a string pattern in relation to the current character for selecting a substring as a feature for CRFs training. This combination of tags and templates is empirically selected through a series of experiments on Bakeoff data sets [53].

5.2. Goodness scores as features

The basic idea of integrating a goodness measure for unsupervised segmentation into a supervised learner such as our baseline system is to inform the learner of the goodness score assigned by the measure to a substring about how likely it is to be a true word. A feasible way to do so is to discretize such goodness scores into integral feature values for integration into the learner, aimed at heightening the learner's ability to identify OOV words at the cost of sacrificing as few in-vocabulary (IV) words as possible.

To make use of such information in an efficient way, it is necessary to filter out those redundant word candidates with insufficient chance to be true words, by means of a proper word length constraint in addition to the default pruning by means of a goodness score threshold. It is reported in [53] that less than 1% of words are longer than 6 characters in Chinese texts, according to the statistics from the Bakeoff-3 data sets. Thus, for the sake of efficiency, we opt to consider only character n-grams under the constraint of this length[11] for the purpose of feature generation. Note that the feature generation is to derive CRFs features from relevant n-grams for a current character in question, not to identify true words among them. The true

[10] http://crfpp.sourceforge.net/.

[11] In fact, allowing longer n-grams leads to little difference in performance, according to our experiments.

words are to be identified via character tagging. Our attempt here is to derive more effective features from these n-grams to enrich the baseline system in the hope of enhancing its tagging. An implicit assumption underlying the word length constraint is that anything six characters away from the current character is assumed to have no role to play in determining its tag.

The CRFs are used as an ensemble model to integrate these features into those extracted from pre-segmented training corpora by the six feature templates. Note that given a character within a particular context, there is a group of relevant features and their values to interact with one another in determining the best choice for its tag. In our current work, a goodness measure brings about two types of new features for a character. The first type is on the n-grams containing that character, with a feature function defined as:

$$f_n(s) = \begin{cases} 1, & \text{if } s \in L, \\ 0, & \text{otherwise}, \end{cases} \tag{9}$$

where L is the word candidate set given by the goodness measure under the length constraint. In the case of more than one goodness measure is involved in generating the candidate set, the default reciprocal pruning is applied, that is, no candidate with any goodness score $\leqslant 0$ is allowed.

The other type is intended to further differentiate the role (or significance) of the above n-gram features in deciding a character's tag. A feature function for a feature template over an n-gram substring s with a goodness score $g(s)$ is defined as follows for this purpose,

$$f_n(s, g(s)) = \begin{cases} t, & \text{if } t \leqslant g(w) < t + 1, \\ 0, & \text{otherwise}, \end{cases} \tag{10}$$

where t is an integer to represent a goodness score as a feature value in the baseline learner. Besides this simple way to discretize (or, equivalently, categorize) continuous goodness scores into integral feature values, there are other ways of doing it for various purposes, e.g., transferring the scores into a fixed number of integers. Our experiments show that this simple approach serves our purpose very well for all goodness measures involved.

But, still, a technical detail needs to be clarified. For a current character involved in several word candidate n-grams simultaneously, which is usually the case, only the one with the greatest goodness score is used to activate the above feature functions for that character. This simple treatment makes the feature representation robust enough to cope with many infrequent n-grams. Moreover, the above feature values do not have to do with the word candidate pruning by goodness threshold or length constraint. Simply applying the above feature functions to all substrings in an input corpus does not cause any difference in the performance of the CRFs learning. The pruning only has a role to play in efficiency enhancement, which is crucial to our current work, because the CRFs training takes an awfully long time. Also, note that the integer t in (10) is not a value for any parameter in the system. Instead, it is a symbolic feature value in the supervised learning. In this sense, our system is actually parameter-free on top of the baseline system.

5.3. Effectiveness of goodness measures

The system described in Section 5.1 is taken as the baseline to examine the effectiveness of the four goodness measures in enhancing the CRFs learning through the above feature generation. The results of our evaluation using the Bakeoff-3 closed test data sets are presented in Tables 12 and 13, specifying either feature generation by (9) or (10) is applied. All new features are derived from the unannotated texts of both the training and test corpora for each track of closed test. The CRFs model is then trained on the annotated training corpus and applied to segment the test corpus for performance evaluation.

Table 12 shows that every goodness measure brings in, consistently, a significant performance improvement over the baseline and that exploiting goodness scores via (10) leads to further improvement over (9), although slighter than that of (9) over the baseline. Note that an improvement by half a percentage point is remarkable at this performance level, for it means an error reduction rate around 10%. The best error reduction rates so achieved range from 6.39% to 16.83% over the baseline. Among the four goodness measures, AV and BE give the most competitive performance while DLG the least, indicating that the best goodness measure for unsupervised segmentation is not necessarily the best choice to enhance the supervised segmentation by CRFs learning of character tagging. It is also worth noting that AV and BE give highly similar results in performance enhancement, revealing their intrinsic similarity.

To evaluate the performance of OOV detection, we resort to F_{OOV}, a more comprehensive evaluation metric than R_{OOV}. Table 12 shows in its right column that all goodness measures help, at their best, the baseline to improve its F_{OOV} on all test corpora by an increment of 2.76–7.0% points, corresponding to a growth of 4.55–9.82% and an error reduction of 7.4–24.18%. The least increment is on MSRA, on which the best R_{OOV} goes down, exceptionally, by -0.3 percentage point, according to our experimental records. Nevertheless, this little impairment of R_{OOV} does not hinder its F-score from going up, indicating a greater positive effect on improving the corresponding precision of OOV words in this case.

Table 13 presents the performance enhancement by the best combinations of the goodness measures, in comparison to their best individual performance. Surprisingly, however, none of the combinations leads to a better F-score than the best of AV and BE, giving a sharp contrast to the ensemble unsupervised segmentation that the goodness measures in combination overwhelmingly surpass their best individual. This result suggests that although each goodness measure does provide the

Table 12
Performance enhancement by new features from different goodness measures: F-score and F_{OOV}.

Goodness Measure	F-score				F_{OOV}			
	AS	CityU	CTB	MSRA	AS	CityU	CTB	MSRA
Baseline	0.9539	0.9691	0.9321	0.9609	0.6419	0.7379	0.7105	0.6065
$AV_{(9)}$	0.9566	0.9721	0.9373	0.9630	0.6678	0.7691	0.7645	0.6273
$AV_{(10)}$	0.9573	0.9740	**0.9428**	**0.9634**	0.6778	0.7896	**0.7805**	**0.6341**
$BE_{(9)}$	0.9566	0.9721	0.9373	0.9630	0.6678	0.7691	0.7645	0.6273
$BE_{(10)}$	**0.9584**	**0.9743**	0.9421	0.9633	**0.6951**	**0.7922**	0.7760	0.6313
$FSR_{(9)}$	0.9565	0.9715	0.9367	0.9621	0.6620	0.7603	0.7548	0.6156
$FSR_{(10)}$	0.9575	0.9735	0.9415	0.9630	0.6808	0.7840	0.7747	0.6238
$DLG_{(9)}$	0.9554	0.9708	0.9395	0.9616	0.6567	0.7547	0.7580	0.6154
$DLG_{(10)}$	0.9560	0.9718	0.9401	0.9617	0.6685	0.7820	0.7605	0.6233
Error Reduction (%)	**9.76**	**16.83**	**15.76**	**6.39**	**14.86**	**20.72**	**24.18**	**7.40**

Table 13
Performance enhancement by goodness measures using feature function (10).

Goodness Measure				F-score				F_{OOV}			
AV	BE	FSR	DLG	AS	CityU	CTB	MSRA	AS	CityU	CTB	MSRA
Baseline				0.9539	0.9691	0.9321	0.9609	0.6419	0.7379	0.7105	0.6065
+				0.9573	0.9740	**0.9428**	**0.9634**	0.6778	0.7896	**0.7805**	**0.6341**
	+			**0.9584**	**0.9743**	0.9421	0.9633	**0.6951**	**0.7922**	0.7760	0.6313
+	+			0.9570	0.9726	0.9421	**0.9635**	0.6702	0.7753	0.7701	0.6357
+		+		0.9574	**0.9739**	0.9425	0.9633	0.6797	**0.7899**	0.7742	**0.6366**
+			+	0.9569	0.9733	0.9425	0.9629	0.6727	0.7824	0.7776	0.6272
	+	+		**0.9575**	0.9725	0.9423	0.9631	0.6787	0.7717	0.7703	0.6285
	+		+	0.9570	0.9734	**0.9428**	0.9627	0.6764	0.7856	**0.7787**	0.6262
		+	+	0.9573	0.9732	0.9416	0.9632	**0.6817**	0.7826	0.7679	0.6269
+	+	+	+	**0.9575**	0.9729	0.9413	0.9630	0.6801	0.7780	0.7685	0.6262

Table 14
Feature templates for various tag sets.

Tags	Source	Feature templates
2	Tsai et al. [43]	C_{-2}, C_{-1}, C_0, C_1, $C_{-3}C_{-1}$, $C_{-2}C_0$, $C_{-2}C_{-1}$, $C_{-1}C_0$, $C_{-1}C_1$, C_0C_1
4	Xue [49]	C_{-2}, C_{-1}, C_0, C_1, C_2, $C_{-2}C_{-1}$, $C_{-1}C_0$, $C_{-1}C_1$, C_0C_1, C_1C_2
6	Zhao et al. [52]	C_{-1}, C_0, C_1, $C_{-1}C_0$, C_0C_1, $C_{-1}C_1$

baseline with more information about word boundaries, such information provided by each of them does not necessarily complement each other in a reciprocally beneficial way.

5.4. Computational cost

As reported above, two technical advancements are integrated into the CRFs learning of character tagging for further improvement upon the state-of-the art of Chinese word segmentation. One is to use a larger tag set than before, of 6 tags as in [52]. The other is to derive additional features from unlabeled texts, to complement those extracted from labeled training data. This subsection examines their cost-effectiveness in terms of the extra computational cost needed to accommodate them in the CRFs model.

For this purpose, we compare the computational cost of using the 2-, 4- and 6-tag sets on the Bakeoff-3 data sets. The best known *n*-gram feature templates for these tag sets are adopted from the previous works that have proved their effectiveness, as illustrated in Table 14. The comparison in Table 15 in terms of training time, the number of features and memory cost shows that the 6-tag set needs a much longer time but only a slightly larger memory space for training on the same data sets.

The number of additional features derived by the four goodness measures via feature functions (9) and (10) are presented in Table 16, where the "baseline" is the numbers of features in the baseline system extracted from the training data by the six *n*-gram feature templates for the 6-tag set. In contrast to these many of features, the additional features are so few in

Table 15
Comparison of computational cost.

Tags	Template	AS	CityU	CTB	MSRA
Training time (minutes)					
2	Tsai	112	52	16	35
4	Xue	206	79	28	73
6	Zhao	402	146	47	117
Numer of features ($\times 10^6$)					
2	Tsai	13.2	7.3	3.1	5.5
4	Xue	16.1	9.0	3.9	6.8
6	Zhao	15.6	8.8	3.8	6.6
Memory cost (Giga bytes)					
2	Tsai	5.4	2.4	0.9	1.8
4	Xue	6.6	2.8	1.1	2.2
6	Zhao	6.4	2.7	1.0	2.1

Table 16
Numbers of additional features.

Goodness	Feature function (9)				Feature function (10)			
Measure	AS	CityU	CTB	MSRA	AS	CityU	CTB	MSRA
Baseline (M)	15.6	8.8	3.8	6.6	15.6	8.8	3.8	6.6
AV	120	120	120	120	672	666	540	660
Incr. (10^{-6})	8	14	32	18	43	76	142	100
BE	120	120	120	120	756	720	726	720
Incr. (10^{-6})	8	14	32	18	48	82	191	109
FSR	120	120	120	120	1014	936	846	936
Incr. (10^{-6})	8	14	32	18	65	107	222	142
DLG	114	114	114	114	**1692**	1482	1350	1476
Incr. (10^{-6})	7	13	30	17	108	169	**355**	223

number, causing only an increment of no more than 0.17% and, accordingly, a negligible addition to the training time, e.g., an addition of tens of seconds to hours on our machine,[12] according to our observation. Such efficiency in computation is certainly an advantage of the proposed method to integrate the unsupervised segmentation.

The high computational complexity of CRFs training is its notorious disadvantage for practical application, especially when it is applied to Chinese word segmentation for which a huge-scaled feature set is inevitable. The success of the 6-tag baseline system certainly owes much to tailoring, via fewer feature templates than others, the feature set into a size accommodable by our machine. But more noticeable is that our key work to integrate unsupervised goodness measures into this baseline leads to a significant rise of performance at a negligible increase of computational cost.

5.5. Comparison to the best

The improved performance by integrating unsupervised and supervised segmentation, as reported above, can be compared to the best in the closed test of Bakeoff-3. The rule for the closed test is that no additional information about words beyond the provided training corpus is allowed, in contrast to the open test that any kind of resources may be used. A summary of the best official performance scores in the closed test of Bakeoff-3 is presented in Table 17 for a comparison with ours. All six participants with at least a third best performance score in any track of the closed test are included in this table [2,43,45,51,52,60]. Our results are obtained by integrating BE features into the baseline system.

From Table 17, we can see that the state-of-the-art performance as in Bakeoff-3 goes far beyond that of our baseline as given in Table 12, but with the aid of the new features generated by AV/BE, our system competes favorably against all best ones in Bakeoff-3. The error reduction we have achieved over these best ranges from 1.08% to 14.37%, giving an average of 6.39%. Particularly worth noting is that this is a comparison disadvantageous to us, although we excel, in that all these best scores were achieved with the aid of critical extra heuristics or resources only allowed in the open test. For example, although officially prohibited, a manually prepared character type list was used as features in both [52,60], and an empirical optimal value for a key parameter in the system of [45] was estimated in advance using an external segmented corpus. In contrast, our system is free of any such extra resource.

[12] Dell PowerEdge SC1435, of two dual-core AMD CPUs, 2.8 GHz, & 24 G memory.

Table 17
Comparisons of the best F-scores in Bakeoff-3 and ours on the same data sets.

Participant	(Site ID)	AS	CityU	CTB	MSRA
Zhu	(1)	0.944	0.968	0.927	0.956
Carpenter	(9)	0.943	0.961	0.907	0.957
Tsai	(15)	0.957	**0.972**	–	0.955
Zhao[a]	(20)	**0.958**	0.971	**0.933**	–
Zhang	(26)	0.949	0.965	0.926	0.957
Wang	(32)	0.953	0.970	0.930	**0.963**
Best of Bakeoff-3		0.9576	0.972	0.9332	0.963
Ours		0.9584	0.9743	0.9428	0.9634
Error reduction (%)		**1.89**	**8.21**	**14.37**	**1.08**

[a] If not rounded up to Bakeoff-3 official format, the F-scores in this row are 0.957635, 0.971140 and 0.933155, respectively. The F-scores for the best of Bakeoff-3 are adjusted accordingly.

Table 18
Statistical significance: comparison of the best closed test results of Bakeoff-3 and ours.

Corpus	#Word	Best	R	C_r	P	C_p	F
AS	91 K	Bakeoff-3	0.961	0.001280	0.955	0.001371	0.958
		Ours	0.964	0.001235	0.953	0.001403	0.958
CityU	220 K	Bakeoff-3	0.973	0.000691	0.972	0.000703	0.972
		Ours	0.974	0.000679	0.974	0.000679	0.974
CTB	154 K	Bakeoff-3	0.940	0.001207	0.926	0.001330	0.933
		Ours	0.947	0.001142	0.937	0.001238	0.943
MSRA	100 K	Bakeoff-3	0.964	0.001176	0.961	0.001222	0.963
		Ours	0.960	0.001239	0.967	0.001130	0.963

Table 19
The best performance in Bakeoff-4 closed test.

Corpus		CityU	CKIP	CTB	NCC	SXU
#Word	Training	1093	722	642	913	528
	Test	236	91	81	152	114
F-score	Ours (Site: 2)	0.9510	0.9470	0.9589	0.9405	0.9623
	Site: 26	0.9430	0.9429	0.9533	0.9386	0.9588
	Site: 5	0.9471	0.9413	0.9447	0.9365	0.9555
	Site: 31	–	–	0.9517	0.9344	0.9543
Error reduction (%)		**14.04**	**7.18**	**11.99**	**3.09**	**8.50**

To further check the significance of the difference of our performance from the others, we have performed a few statistical significance tests by comparing our recalls, precisions and F-scores to the best ones in the same closed test. Following the previous work in [37] to assume a binomial distribution for such scores, we compute the 95% confidence interval as $\pm 2\sqrt{p(1-p)/n}$ in light of the Central Limit Theorem for Bernoulli trials [12], where n is the number of trials (i.e., words in our case). It is also assumed that the recall R may approximate the probability of a true word being identified correctly and the precision P the probability that an identified word is really a word. Accordingly, two types of interval, C_R and C_P, are computed by setting p to R and P respectively. One can verify whether two experimental results are significantly different at the 95% confidence level by checking whether their confidence intervals overlap. The values of C_R and C_P for the best Bakeoff-3 results and ours are presented in Table 18, where the rows of Bakeoff-3 are quoted from [24].

A more convincing piece of evidence than the above to re-confirm the decent advantage of integrating unsupervised and supervised segmentation comes from the best two systems in the closed test of Bakeoff-4, whose official performance scores are presented in Table 19 in comparison to a few others in the third places. It is reported in [55] that the AV features derived from the unannotated training and test corpora help a CRFs model, very much like the one for our baseline here, to rank first throughout all closed test tracks. Interestingly, another CRFs model [46] utilizing a raw text feature similar to BE, referred to as contextual entropy though, ranks second in all but one closed track, illustrating an overall performance significantly better than the others in lower ranks. With the aid of the AV features, our model achieves an average error reduction by 8.96% over the runner-up ones in terms of their F-scores.

Table 20
Comparison of performance on Bakeoff-2 data sets.

Corpus	AS	CityU	MSRA	PKU
Xiong et al. [48]	–	0.956	0.972	0.952
Baseline	0.9534	0.9476	0.9735	0.9515
+AV	0.9570	0.9610	0.9758	0.9540
Error reduction (%)	–	**11.36**	**13.57**	**4.17**

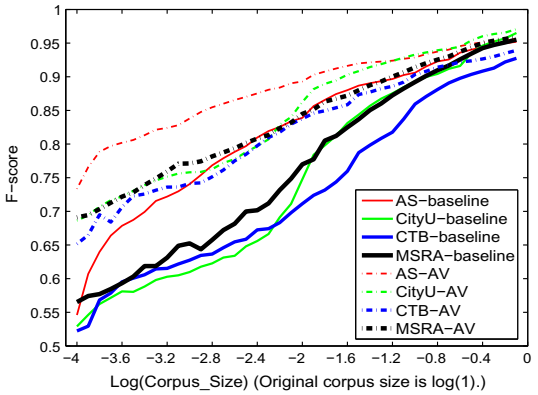

Fig. 3. Performance along training corpus size.

Recently, a novel approach to CRFs training is proposed in [48] with minimum tag error as criterion, using three of the four Bakeoff-2 data sets for evaluation. It is validated via a favorable performance comparison to our baseline on two of the three data sets, termed as achieving a state-of-the-art performance. A comprehensive comparison of our performance to theirs is presented in Table 20, showing again the effectiveness of the AV features as in achieving an error reduction nearly 10% on average.

With regard to the fact that Bakeoff-4 data sets are, in general, significantly smaller than those for Bakeoff-3, there is a reasonable guess that the unsupervised segmentation may help the CRFs model to enhance its performance more significantly on small data sets. This is particularly valuable, because it is so costly to prepare a large scale high-quality segmented corpus for training. To test this guess, we have carried out a series of experiments with Bakeoff-3 corpora. For each data set, the training corpus is exponentially reduced in size by random sampling while the evaluation is performed on the respective original test corpus. The performance curves of the CRFs model trained with and without AV features along the size change of the training corpus are presented in Fig. 3 for a comparison, showing that the AV features become more effective in enhancing the CRFs model's performance as the training corpus becomes smaller. Remarkably, the AV features can lead to a performance enhancement by 20–35% over the baseline as a training corpus is reduced to $1/10^4$ of its original size.

5.6. Using open resources

So far only the plain texts of the training and test corpora in a given data set have been used for extracting unsupervised goodness features to enhance supervised segmentation. Interestingly, however, it is reported that large scale external resources such as segmented corpora can be used to facilitate supervised learning for various segmentation tasks for further improvement over using only a given training corpus, e.g., as illustrated in [26,52]. With regard to the fact that a large scale segmented corpus is developed at an expensive cost, it is particularly meaningful to ask whether our approach can benefit from using external unlabeled texts that can be obtained at very little cost.

Also, note that extracting unsupervised features from unlabeled texts is to provide a set of word candidates to facilitate CRFs training on a segmented corpus. We know that an external dictionary of known true words can provide more reliable lexical information than this kind of word candidates. Can we merge these two kinds of lexical information, one extracted from unlabeled texts and the other from an external dictionary, to achieve a better performance than using only one of them?

To answer these questions, we have conducted a number of experiments on CTB and MSRA corpora of Bakeoff-3, using the corpus of People's Daily (1993–97), of about 100 M characters, as external unlabeled texts and a Chinese word list from Peiking University,[13] of 108 K words of one to four characters long, as an external dictionary. Both the evaluation corpora and the

[13] Available at: http://ccl.pku.edu.cn/doubtfire/Course/Chinese%20Information%20Processing/Source_Code/Chapter_8/Lexicon_full_2000.zip.

Table 21
Performance using external resources.

Corpus	Metric	Text	Dict	Text + Dict	Best open
CTB	F-score	0.9401	0.9412	0.9443	0.944
	R_{OOV}	0.7382	0.7412	0.7565	0.768
	F_{OOV}	0.7651	0.7680	0.7871	–
MSRA	F-score	0.9674	0.9681	0.9716	0.979
	R_{OOV}	0.6905	0.6905	0.7140	0.839
	F_{OOV}	0.6737	0.6786	0.7141	–

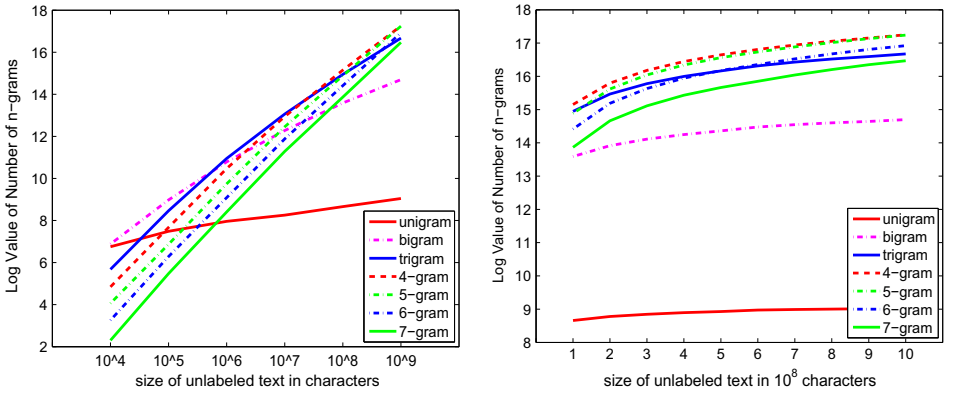

Fig. 4. Number of *n*-grams along corpus size growth: exponential vs. linear.

external resources are in GB code. The same feature representation as in [26] is adopted for the external dictionary. AV is used as the criterion for extracting unsupervised features from the unlabeled texts. The experimental results are presented in Table 21 in comparison to the best performance in the same tracks of Bakeoff-3 open test, with "Text" to indicate using People's Daily as external texts and "Dict" using lexical features from the external dictionary.

These results show that the two kinds of lexical information do strengthen each other, achieving a better performance than any of them alone. Furthermore, this performance is highly competitive against the best in the Bakeoff-3 open test, which was obtained through exploiting large scale external segmented corpora, lexicons and named entity information [17,25]. In contrast, we only use external unlabeled texts and an external word list. All these give evidence that our approach is also effective in exploiting external resources, including unlabeled texts and known words.

5.7. How much unlabeled text for a significant enhancement?

Our approach involves a trade-off between the performance enhancement and the amount of unlabeled text in use to achieve this enhancement. It is thus a critical issue to estimate the threshold amount that can ensure a significant enhancement. For this purpose, we conduct a series of experiments with AV as the unsupervised criterion. Since it is preferable to have a larger and more balanced corpus of unlabeled texts than the five-year corpus from People's Daily with a strong stylistic bias, we opt for the Chinese Gigaword (CGW) Third Edition[14] of one billion characters. We use a series of its sub-corpora of various sizes in the experiments for the purpose of examining how the scale of unlabeled text affects the performance. We let the size grow in two different ways, one exponential and the other linear, each covering a different range of size. All sub-corpora are extracted from the beginning of the original CGW corpus.

Following the default settings as before, only *n*-gram substrings that appear more than once are considered word candidates. Their numbers are depicted in Figs. 4 and 5 as logarithmic values along the growth of corpus size, with each curve for *n*-grams of a different length. These figures show that the candidates extracted by AV follow a relatively stable growth rate along the corpus size, except the unigram ones whose number only increases slightly even along the exponential growth of corpus size. An explanation for this is that Chinese characters are practically a closed set, in the sense that the frequently used ones are around 3000 and the double of this amount covers almost all Chinese texts. All others are practically inactive in Chinese morphology. This explains why the unigrams extracted by AV hardly exceeds 3000 even if the sub-corpus in question reaches one billion characters.

Then, the word segmentation is performed on the two GB code corpora of Bakeoff-3 as before, with the aid of the AV features derived from the CGW sub-corpora of various size to enhance the CRFs model. The segmentation performance is

[14] See http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2007T38.

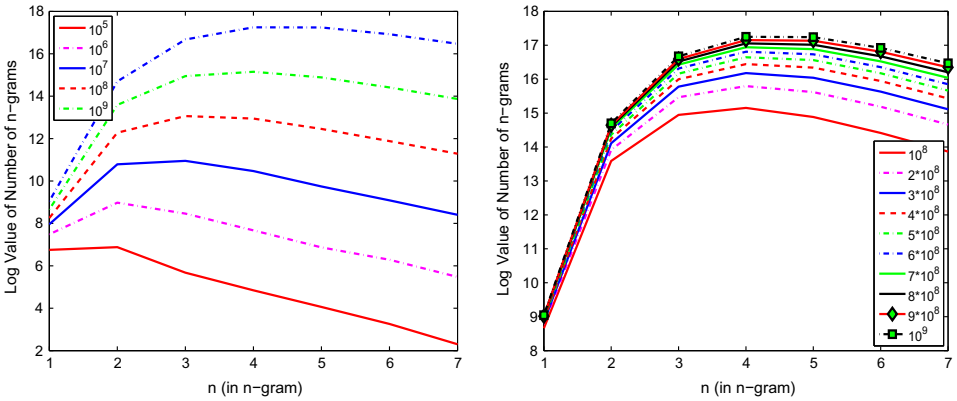

Fig. 5. Number of *n*-grams vs. *n* along corpus size growth: exponential vs. linear.

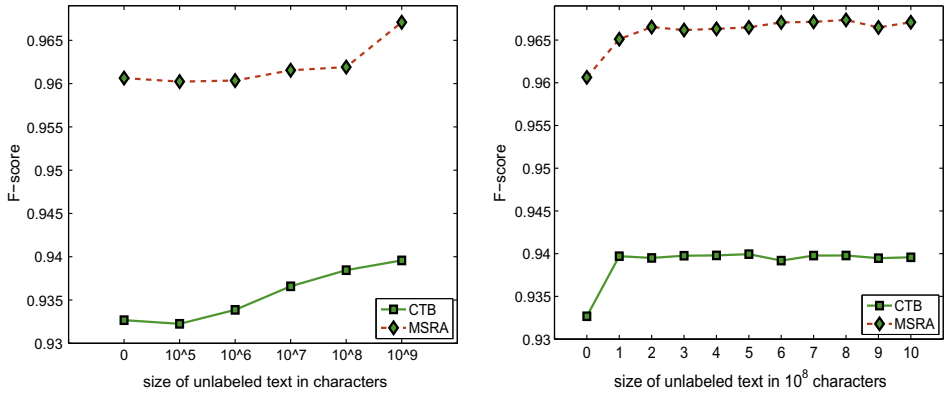

Fig. 6. Performance along size growth of unlabeled corpus: exponential vs. linear.

presented in Fig. 6 to show the effect of the size of unlabeled texts, in both exponential and linear growth at different scale levels. This empirical result suggests that a balanced corpus of unlabeled texts slightly more than 10^8 characters can effectively help the CRFs model obtain the best performance on word segmentation. This size gives a nice trade-off between performance enhancement and the volume of unlabeled texts for this particular task.

5.8. Semi-supervised learning

It is shown above that the unsupervised goodness measures, when applied on the same data set, including the raw texts of both the training and test corpora, do help to improve the supervised learning from the segmented training corpus. Interestingly, however, when unlabeled data is exploited this way to improve supervised learning, semi-supervised learning is actually involved.

As a supervised learning model for sequence labeling, CRFs have been extended for semi-supervised learning. Semi-supervised CRFs are proposed in [18] based on a minimum entropy regularizer, whose parameters are estimated to maximize the likelihood of labeled data and the negative conditional entropy of the unlabeled data. In [41], another semi-supervised learning approach is proposed based on a hybrid generative and discriminative model. By defining an objective function for the hybrid model in log-linear form, the discriminative structured predictor (i.e., the CRFs) on labeled data and the generative model on unlabeled data are integrated, in a way that the generative model utilizes the unlabeled data to increase the sum of the discriminant functions during parameter estimation.

The underlying idea in our current work is close to that in [41]. However, since the CRFs learning for word segmentation is a very heavy task in terms of the size of training data and the immense computation required, another round of parameter estimation using additional lexical information such as new word candidates with their goodness scores derived through unsupervised segmentation would inevitably worsen the situation. To get around this problem, we have adopted a direct course to merge the lexical information into the discriminant model for training. This approach has been proven efficient and effective by our experiments.

6. Conclusion

After a quarter century of exploration, Chinese word segmentation has becomes a mature language processing technology. The state-of-the art in this area has approached very closely to the ceiling point of performance, and we are not so sure any more if its error rate is still significantly higher than that of a test corpus manually prepared by experts. Most IV words can be identified correctly, for most ambiguities involving them can be properly resolved by statistical learning when large scale pre-segmented corpora are available for training. In contrast, OOV words remain a more prominent issue, which has become the major account for most segmentation errors. As the creation of new words for novel things in this rapidly changing world is unlikely to slow down in the information era, it is unrealistic to deny the strong call for a sound resolution for this unavoidable problem in the computer processing of any language, in particular, a language like Chinese without explicit word delimiters.

It is as a response to this call that we have explored the feasibility of integrating unsupervised segmentation into supervised segmentation for performance enhancement, in the hope of helping the CRFs learning of character tagging to recognize more OOV words. It is the first attempt of this kind in the field. Interestingly, however, our experimental results show that the global information derived from raw texts by a goodness measure of unsupervised segmentation helps the CRFs model to improve its performance not only on OOV words but also on IV words, resulting in, desirably, a significant improvement on its overall performance. This is evidenced by the further error reduction that we have achieved over the best existing performance.

It is for the purpose of fitting the global information into the CRFs character tagging that we have empirically compared a number of goodness measures for unsupervised segmentation, to examine their effectiveness in segmenting Chinese texts. Their performance is achieved under certain constraints (e.g., word length $\leqslant 6$) imposed by the existing settings of the best CRFs model that we have had for character tagging. The comparison exhibits not only their strengths in recognizing words of various lengths, achieving a performance far beyond the baseline, but also the possibility of improving their performance via word candidate pruning[15] and ensemble segmentation. Nevertheless, this is by no means to imply that our work under such constraints has exhausted their potentials for unsupervised segmentation. Many other criteria for word discovery, e.g., mutual information [38], should also be included in the comparison if we had come up with an effective way to fit them into the current settings of our CRFs learning model.

We also note that our research provides only a simple and straightforward way, among so many possibilities, to integrate unsupervised and supervised segmentation, by casting the goodness scores of word candidates into feature values for use in the CRFs model. It is by no means the sole or the best one. There are other possibly more effective ways for future exploration. The greatest contribution of our recent work is not to show the significantly and consistently better performance of our model against others', including the best ones in the Bakeoffs, as evidenced by the error reduction rates around 6.39–8.96% over the best scores in Bakeoff-3 and -4. Rather, it is that we have empirically proved that the utility of unlabeled raw texts in enhancing the best supervised learning model for segmentation conventionally trained only on large scale pre-segmented corpora. The raw texts proved to be helpful include not only the test corpus and the unsegmented version of the training corpus in the same evaluation data set but also other texts from open resources. It is not a surprise that unsupervised segmentation, which infers words from raw texts from scratch with few lexical resources, can never compete with its supervised counterpart, but it is surely a great success that its strengths can be integrated into the latter for such a substantial enhancement.

Even so, unsupervised segmentation never stops playing a critical role in word segmentation, no matter how advanced the latter has been, because new word creation never ends and the existing lexical resources are hence never sufficient. Therefore, OOV word detection remains the most prominent issue in the field for our future research. Also, learning from our experience in integrating supervised and unsupervised segmentation, we have observed the necessity to integrate the strengths of various word segmentation approaches, in particular, the conventional word-based statistical modeling and the current trend of character-based CRFs learning. If these technologies could be put together to complement each other effectively, we would certainly expect our future to bring up a more full-fledged and versatile segmentor with a stable and satisfactory performance on various kinds of Chinese open texts.

References

[1] R.K. Ando, L. Lee, Mostly-unsupervised statistical segmentation of Japanese: Applications to Kanji, in: Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL 2000), Seattle, Washington, 2000, pp. 241–248.
[2] B. Carpenter, Character language models for Chinese word segmentation and named entity recognition, in: Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing (SIGHAN-5), Sydney, Australia, 2006, pp. 169–172.
[3] J.-S. Chang, K.-Y. Su, An unsupervised iterative method for Chinese new lexicon extraction, International Journal of Computational Linguistics and Chinese Language Processing (CLCLP) 2 (2) (1997) 97–148.
[4] L.-F. Chien, PAT-tree-based keyword extraction for Chinese information retrieval, in: Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'97), Philadelphia, 1997, pp. 50–58.
[5] T. Emerson, The second international Chinese word segmentation bakeoff, in: Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing (SIGHAN-4), Jeju Island, Korea, 2005, pp. 123–133.

[15] This confirms again the observation by other researchers, e.g., [32], that the quality of word candidates plays a critical role in ensuring the quality of unsupervised segmentation output.

[6] H. Feng, K. Chen, X. Deng, W. Zheng, Accessor variety criteria for Chinese word extraction, Computational Linguistics 30 (1) (2004) 75–93.
[7] H. Feng, K. Chen, C. Kit, X. Deng, Unsupervised segmentation of Chinese corpus using accessor variety, in: K.-Y. Su, J. Tsujii, J.H. Lee, O.Y. Kwong (Eds.), Natural Language Processing – IJCNLP 2004, LNAI, vol. 3248, Springer, 2005, pp. 694–703.
[8] G.-H. Fu, X.-L. Wang, Unsupervised Chinese word segmentation and unknown word identification, in: The Fifth Natural Language Processing Pacific Rim Symposium 1999 (NLPRS'99), Closing the Millennium, Beijing, China, 1999, pp. 32–37.
[9] G.-H. Fu, C. Kit, J.J. Webster, Chinese word segmentation as morpheme-based lexical chunking, Information Sciences 178 (9) (2008) 2282–2296.
[10] X. Ge, W. Pratt, P. Smyth, Discovering Chinese words from unsegmented text, in: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99), ACM, Berkeley, CA, 1999, pp. 271–272.
[11] G. Grefenstette, Tokenisation, in: H. van Halteren (Ed.), Syntactic Wordclass Tagging, Kluwer, Dordrecht, 1999, pp. 117–133.
[12] C. Grinstead, J.L. Snell, Introduction to Probability, American Mathematical Society, Providence, RI, 1997.
[13] Z.S. Harris, From phoneme to morpheme, Language 31 (2) (1955) 90–222.
[14] Z.S. Harris, Morpheme boundaries within words, in: Papers in Structural and Transformational Linguistics, Reidel, Dordrecht, Holland, 1970, pp. 68–77.
[15] C.-N. Huang, H. Zhao, Chinese word segmentation: A decade review, Journal of Chinese Information Processing 21 (3) (2007) 8–20.
[16] J.H. Huang, D. Powers, Chinese word segmentation based on contextual entropy, in: D.H. Ji, K.-T. Lua (Eds.), Proceedings of the 17th Pacific Asian Conference on Language, Information and Computation (PACLIC 17), COLIPS Publication, Sentosa, Singapore, 2003, pp. 152–158.
[17] A.J. Jacobs, Y.W. Wong, Maximum entropy word segmentation of Chinese text, in: Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing (SIGHAN-5), Sydney, Australia, 2006, pp. 108–117.
[18] F. Jiao, S. Wang, C.-H. Lee, R. Greiner, D. Schuurmans, Semi-supervised conditional random fields for improved sequence segmentation and labeling, in: COLING/ACL-2006, Sydney, Australia, 2006, pp. 209–216.
[19] Z. Jin, K. Tanaka-Ishii, Unsupervised segmentation of Chinese text by use of branching entropy, in: COLING/ACL 2006, Sidney, Australia, 2006, pp. 428–435.
[20] C. Kit, Unsupervised lexical learning as inductive inference, Ph.D. Thesis, University of Sheffield, 2000.
[21] C. Kit, Y. Wilks, Unsupervised learning of word boundary with description length gain, in: Osborne, M., Sang, E.T.K. (Eds.), Computational Natural Language Learning (CoNLL-99), Bergen, Norway, 1999, pp. 1–6.
[22] C. Kit, H. Zhao, Improving Chinese word segmentation with description length gain, in: The 2007 International Conference on Artificial Intelligence (ICAI-2007), Las Vegas, Nevada, USA, 2007, pp. 846–851.
[23] J.D. Lafferty, A. McCallum, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: Proceedings of the Eighteenth International Conference on Machine Learning (ICML'01), Morgan Kaufmann, San Francisco, CA, USA, 2001, pp. 282–289.
[24] G.-A. Levow, The third international Chinese language processing bakeoff: Word segmentation and named entity recognition, in: Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing (SIGHAN-5), Sydney, Australia, 2006, pp. 108–117.
[25] W. Liu, H. Li, Y. Dong, N. He, H. Luo, H. Wang, France Telecom R& D Beijing word segmenter for SIGHAN bakeoff 2006, in: Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing (SIGHAN-5), Sydney, Australia, 2006, pp. 108–117.
[26] J.K. Low, H.T. Ng, W. Guo, A maximum entropy approach to Chinese word segmentation, in: Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing (SIGHAN-5), Jeju Island, Korea, 2005, pp. 161–164.
[27] X. Lü, L. Zhang, J. Hu, Statistical substring reduction in linear time, in: K.-Y. Su, J. Tsujii, J.H. Lee, O.Y. Kwong (Eds.), Natural Language Processing – IJCNLP 2004, LNAI, vol. 3248, Springer, 2005, pp. 320–327.
[28] K.-T. Lua, K.-W. Gan, An application of information theory in Chinese word segmentation, Computer Processing of Chinese and Oriental Languages 8 (1) (1994) 115–123.
[29] A. Mikheev, Text segmentation, in: R. Mitkov (Ed.), The Oxford Handbook of Computational Linguistics, Oxford University Press, 2003, pp. 201–218.
[30] D.D. Palmer, Tokenisation and sentence segmentation, in: R. Dale, H. Moisl, H. Somers (Eds.), Handbook of Natural Language Processing, Marcel Dekker, New York, 2000, pp. 11–36.
[31] F. Peng, F. Feng, A. McCallum, Chinese segmentation and new word detection using conditional random fields, in: Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004), Geneva, Switzerland, pp. 562–568.
[32] F. Peng, X. Huang, D. Schuurmans, N. Cercone, S. Robertson, Using self-supervised word segmentation in Chinese information retrieval, in: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01), Tampere, Finland, 2001, pp. 349–350.
[33] F. Peng, D. Schuurmans, Self-supervised Chinese word segmentation, in: The Fourth International Symposium on Intelligent Data Analysis (IDA-2001), Lisbon, Portugal, 2001, pp. 238–247.
[34] J.M. Ponte, W.B. Croft, USeg: A retargetable word segmentation procedure for information retrieval, Presented at the Symposium on Document Analysis and Information Retrieval'96 (SDAIR),Technical Report TR96-2, University of Massachusetts, Amherst, MA, 1996.
[35] B. Rosenfeld, R. Feldman, M. Fresko, A systematic cross-comparison of sequence classifiers, in: SDM 2006, Bethesda, Maryland, pp. 563–567.
[36] C.E. Shannon, A mathematical theory of communication, The Bell System Technical Journal 27 (1948) 379–423. 623–656.
[37] R. Sproat, T. Emerson, The first international Chinese word segmentation bakeoff, in: The Second SIGHAN Workshop on Chinese Language Processing (SIGHAN-2), Sapporo, Japan, 2003, pp. 133–143.
[38] R. Sproat, C. Shih, A statistical method for finding word boundaries in Chinese text, Computer Processing of Chinese and Oriental Languages 4 (4) (1990) 336–351.
[39] M. Sun, D. Shen, B.K. Tsou, Chinese word segmentation without using lexicon and hand-crafted training data, in: COLING-ACL'98, vol. 2, Montreal, Quebec, Canada, 1998, pp. 1265–1271.
[40] M. Sun, M. Xiao, B.K. Tsou, Chinese word segmentation without using dictionary based on unsupervised learning strategy, Chinese Journal of Computers 27 (6) (2004) 736–742.
[41] J. Suzuki, A. Fujino, H. Isozaki, Semi-supervised structured output learning based on a hybrid generative and discriminative approach, in: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL'07), Prague, Czech, 2007, pp. 791–800.
[42] W.J. Teahan, Y. Wen, R. McNab, I.H. Witten, A compression-based algorithm for Chinese word segmentation, Computational Linguistics 26 (3) (2000) 375–393.
[43] R.T.-H. Tsai, H.-C. Hung, C.-L. Sung, H.-J.Dai, W.-L. Hsu, On closed task of Chinese word segmentation: An improved CRF model coupled with character clustering and automatically generated template matching, in: Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing (SIGHAN-5), Sydney, Australia, 2006, pp. 108–117.
[44] C.-H. Tung, H.-J. Lee, Identification of unknown words from corpus, International Journal of Computer Processing of Chinese and Oriental Languages 8 (Suppl.) (1995) 131–146.
[45] X. Wang, X. Lin, D. Yu, H. Tian, X. Wu, Chinese word segmentation with maximum entropy and N-gram language model, in: Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing (SIGHAN-5), Sydney, Australia, 2006, pp. 138–141.
[46] Z. Wang, C. Huang, J. Zhu, The character-based CRF segmenter of MSRA & NEU for the 4th Bakeoff, in: Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing (SIGHAN-6), Hyderabad, India, 2008, pp.98–101.
[47] J.J. Webster, C. Kit, Tokenization as the initial phase in nlp, in: Proceedings of the 14th International Conference on Computational Linguistics (COLING-92), vol. IV, Nantes, France, 1992, pp. 1106–1110.
[48] Y. Xiong, J. Zhu, H. Huang, H. Xu, Minimum tag error for discriminative training of conditional random fields, Information Sciences 179 (1–2) (2009) 169–179.
[49] N. Xue, Chinese word segmentation as character tagging, International Journal of Computational Linguistics and Chinese Language Processing 8 (1) (2003) 29–48.

[50] J. Zhang, J. Gao, M. Zhou, Extraction of Chinese compound words – An experimental study on a very large corpus, in: Proceedings of the Second Chinese Language Processing Workshop, Hong Kong, China, 2000, pp. 132–139.
[51] M., Zhang, G.-D. Zhou, L.-P. Yang, D.-H. Ji, Chinese word segmentation and named entity recognition based on a context-dependent mutual information independence model, in: Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing (SIGHAN-5), Sydney, Australia, 2006, pp. 154–157.
[52] H. Zhao, Huang, C.-N., M. Li, An improved Chinese word segmentation system with conditional random field, in: Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing (SIGHAN-5), Sydney, Australia, 2006, pp. 162–165.
[53] H. Zhao, C.-N. Huang, Li, M., Lu, B.-L., Effective tag set selection in Chinese word segmentation via conditional random field modeling, in: Proceedings of the 20th Pacific Asian Conference on Language, Information and Computation (PACLIC 20), Wuhan, China, 2006, pp. 87–94.
[54] H. Zhao, C. Kit, Incorporating global information into supervised learning for Chinese word segmentation, in: Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics (PACLING 2007), Melbourne, Australia, 2007, pp. 66–74.
[55] H. Zhao, C. Kit, Unsupervised segmentation helps supervised learning of character tagging for word segmentation and named entity recognition, in: The Sixth SIGHAN Workshop on Chinese Language Processing (SIGHAN-6), Hyderabad, India, 2008, pp. 106–111.
[56] H. Zhao, C. Kit, An empirical comparison of goodness measures for unsupervised Chinese word segmentation with a unified framework, in: The Third International Joint Conference on Natural Language Processing (IJCNLP-2008), vol. 1, Hyderabad, India, 2008, pp. 9–16.
[57] H. Zhao, C. Kit, Exploiting unlabeled text with different unsupervised segmentation criteria for Chinese word segmentation, The 9th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2008), Haifa, Israel, 2008, Research in Computing Science 33 (2008) 93–104.
[58] H. Zhao, C. Kit, Scaling conditional random fields by one-against-the-other decomposition, Journal of Computer Science and Technology 23 (4) (2008) 612–619.
[59] H. Zhao, C. Kit, A simple and efficient model pruning method for conditional random fields, in: Proceedings of the 22nd International Conference on the Computer Processing of Oriental Languages (ICCPOL-2009), Hong Kong, China, 2009, pp. 149–159.
[60] M.-H. Zhu, Y.-L. Wang, Z.-X. Wang, H.-Z. Wang, J.-B. Zhu, Designing special post-processing rules for SVM-based Chinese word segmentation, in: Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing (SIGHAN-5), Sydney, Australia, 2006, pp. 217–220.