Inf Syst Front (2011) 13:115–125
DOI 10.1007/s10796-010-9278-5

Domain-specific Chinese word segmentation using suffix tree and mutual information

Daniel Zeng · Donghua Wei · Michael Chau · Feiyue Wang

Published online: 17 October 2010

Abstract As the amount of online Chinese contents grows, there is a critical need for effective Chinese word segmentation approaches to facilitate Web computing applications in a range of domains including terrorism informatics. Most existing Chinese word segmentation approaches are either statistics-based or dictionary-based. The pure statistical method has lower precision, while the pure dictionary-based method cannot deal with new words beyond the dictionary. In this paper, we propose a hybrid method that is able to avoid the limitations of both types of approaches. Through the use of suffix tree and mutual information (MI) with the dictionary, our segmenter, called IASeg, achieves high accuracy in word segmentation when domain training is available. It can also identify new words through MI-based token merging and dictionary updating. In addition, with the proposed Improved Bigram method IASeg can process N-grams. To evaluate the performance of our segmenter, we compare it with two well-known systems, the Hylanda segmenter and the ICTCLAS segmenter, using a terrorism-centric corpus and a general corpus. The experiment results show that IASeg performs better than the benchmarks in both precision and recall for the domain-specific corpus and achieves comparable performance for the general corpus.

Keywords Mutual information · Chinese segmentation · N-gram · Suffix tree · Ukkonen algorithm · Heuristic rules

A preliminary and shorter version of this paper appeared in the *Proceedings of the 2008 Intelligence and Security Informatics Workshops* (Springer LNCS #5075).

D. Zeng (✉) · D. Wei · F. Wang
Chinese Academy of Sciences, Institute of Automation, Beijing, China
e-mail: zengdaniel@gmail.com

D. Wei
e-mail: donghuawei@gmail.com

F. Wang
e-mail: feiyue.wang@ia.ac.cn

D. Zeng · F. Wang
The University of Arizona, Tucson, AZ, USA

M. Chau
The University of Hong Kong, Hong Kong, China
e-mail: mchau@business.hku.hk

1 Introduction

Chinese word segmentation is a problem that has been widely studied in the Chinese language processing literature. The Chinese language consists of characters and a word is formed by one or more characters. The main problem in Chinese word segmentation is that, unlike English, there is no space between words in Chinese. Word segmentation is the first step of most analyses for Chinese texts. If words are not segmented correctly, one cannot extract meaningful Chinese words from documents and any subsequent text analyses could not be correctly conducted.

Most existing Chinese word segmentation techniques have been designed for general corpus. These techniques often suffer from being "too general" and do not perform well when processing documents in a new domain, especially for domain-specific collections that contain a lot of new words. In this paper, we aim to develop a domain-specific Chinese word segmentation approach with application to terrorism informatics.

From an application standpoint, China faces terrorism threats from a number of groups, many of them separatism-driven. As these groups are increasingly using the Web as a recruiting, mobilization, public relations, and even (to some

extent) operational platform, there is an urgent need for terrorism researchers and law enforcement officials to develop Web computing tools to analyze terrorism-related contents and extract useful patterns for both retrospective analysis and prospective early warning purposes. We are developing a large-scale terrorism informatics research platform along those lines with the Chinese Academy of Sciences as the main sponsor. The Web contents in Chinese are the primary focus and the reported work is a technical component of this research initiative. Technically, there are a number of practical considerations driving our research in domain-specific Chinese word segmentation. First, terrorism-related Web contents are relatively dynamic and new words and phrases are commonplace. Pure statistical methods do not deliver the needed performance as to accuracy. Pure dictionary-based methods are problematic as well because of the extensive manual efforts and the time delay. Second, the existing approaches perform poorly when dealing with text with mixed Chinese and English sentences. Such mixed text is common in the terrorism domain.

To meet these challenges, we propose in this paper a hybrid method that combines mutual information and suffix tree to address the word segmentation problem in domain-specific Chinese document analysis with application in terrorism informatics. The rest of the paper is structured as follows. Section 2 reviews related work in Chinese word segmentation. In Section 3 we describe our proposed algorithm based on mutual information and suffix tree. Section 4 reports the results of our evaluation study, in which we tested our proposed algorithm using a domain-specific corpus and a general corpus. In Section 5, we discuss our findings and finally in Section 6 we conclude our study with some discussions and suggestions for future research.

2 Related work

Chinese word segmentation has been studied for many years, but two problems in word segmentation, namely unknown word identification and ambiguity parsing, are still not completely solved. Studies on Chinese word segmentation can be roughly divided into two categories: heuristic dictionary-based methods, and statistical machine learning methods. Readers are referred to Wu and Tseng (1993) for a more detailed survey. In the following, we briefly review previous research in each category.

2.1 Dictionary-based methods

Dictionary-based methods mainly employ a predefined dictionary and some hand-generated rules for segmenting input sequence. These rules can be generally classified based on the scanning direction and the prior matching length. The Forward Matching Method (FMM), the input string will be scanned from the beginning to the end and matched against dictionary entries. In the Reverse Matching Method (RMM), the input string will be scanned from the end to the beginning. The Bidirectional Matching Method (BMM) scans the input string from both directions. The matching length can be based on maximum matching or minimum matching. Most popular dictionary-based segmenters use a hybrid matching method (Wong and Chan 1996). The main disadvantage of dictionary-based methods is that their performance depends on the coverage of the lexicon, which unfortunately may never be complete because new words appear constantly. Consequently, these methods cannot deal with the unknown words (sometimes called Out-Of-Vocabulary or OOV) identification and may result in wrong segmentation.

2.2 Statistical and machine learning methods

Statistical methods rely on different measures to decide on the segmentation boundary in Chinese. Sun et al. (2004) use a liner function of mutual information (MI) and difference of *t*-test to perform text segmentation. Many researchers also concentrate on two-character words (bigrams), because two is the most common length in Chinese words. Dai et al. (1999) use contextual and positional information, and found that contextual information is the most important factor for bigram extraction. They found that positional frequency is not helpful in determining words. Yu et al. (2006) proposed a cascaded Hidden Markov Model (HMM) for location and organization identification. Other researchers, such as Jia and Zhang (2007), Xue et al. (2002), Xue (2003), and Low et al. (2005) focus on the Maximum Entropy (ME) models. Li and Zhang (2002) use Expectation Maximization and Maximum Likelihood Prediction to deal with Chinese word segmentation. Zhou and Liu (2002) construct state chart using the information of whether several characters can compose one word and propose an algorithm to generate candidate words. Sproat et al. (1996) propose a Stochastic Finite-State Word-Segmentation method by combining character states with dictionary-based heuristic rules. There are several others: Hockenmaier and Brew (1998) present an algorithm, based on Palmer's (1997) experiments, that applies a symbolic machine learning technique to the problem of Chinese word segmentation. Many other statistic-based machine learning methods have been used in Chinese word segmentation, such as SVM-based segmentation (Li et al. 2001), the CRF method segmentation (Peng et al. 2004), unsupervised models (Creutz and Lagus 2007).

In sum, all methods have to rely on either character-level information indicated by the co-occurrence probability, conditional probability, position status, or word-level

information provided by the dictionary or the language knowledge, such as the part-of-speech, morphological, syntactic and semantic knowledge (Cui et al. 2006). It is often difficult for statistical methods to segment words when no such information is available, e.g., words that have low frequencies of occurrences. Many researchers combine the available information and achieved better performance in both unsupervised learning (Peng and Schuurmans 2001) and supervised learning (Teahan et al. 2000).

3 A hybrid approach using mutual information and suffix tree

In this paper, we propose to use Mutual Information (MI) and Suffix Tree to perform Chinese word segmentation. These two techniques allow us to combine the dictionary-based method and the statistics-based method in word segmentation, mitigating the problems associated with applying either method alone. While Ponte and Croft (1996) just deal with bigrams, we focus more on segmentation of trigrams and longer words. We first use a training corpus to train the bigram model and use a lexicon to establish the improved bigram model. We then use MI and the improved bigram model combining with the Suffix Tree to parse the given text. Suffix tree is used here because it has the advantage of storing partial texts while allowing fast matching.

In this section, we describe our proposed algorithm, called IASeg. The overall algorithm is shown in Fig. 1. IASeg has two phases: the training phase and the test phase. The target of the training phase is to construct the dictionary and the N-grams, which include the Unigram, Bigram and the Improved Bigram. Both the dictionary and the N-grams contain the general and the domain-specific words or phrases. As shown in Fig. 1, suffix tree is used to represent and store the training corpus which is already properly segmented. The words extracted from these segmentation results constitute the dictionary. The unigrams and simple bigrams are calculated using standard methods and kept in a hash table. The calculation of the N-grams with $n>2$ is done through our proposed Improved Bigram approach that makes heavy use of MI. The details about MI calculations, the use of suffix tree as a data structure, and the Improved Bigram algorithm, are provided in the subsequent subsections.

In the test phase, we first split the input string into tokens using dictionary-based FMM heuristic. Then we calculate the strings' Mutual Information to predict unknown words and decide whether we should merge the two adjacent tokens as one new word. If the formation of the new word is supported, we update the dictionary dynamically. Finally, we output the segmentation results.

One key aspect of our proposed method is that whenever available, a domain-specific dictionary is used for word segmentation. However, unlike in a pure dictionary-based approach, this dictionary is automatically updated through MI calculations whenever new documents are encountered.

3.1 N-gram construction

The N-gram word model is based on the Markovian assumption. If the N-th character is related only with its preceding (N-1) characters of context, and without any other correlations, we call it the N-gram word model or (N-1)-order Markov model.

Given a string $w_1w_2...w_n$, with its length being n, we have the following equations.In unigram, we have:

$$p(w_1w_2...w_n) = p(w_1)p(w_2)...p(w_n)$$

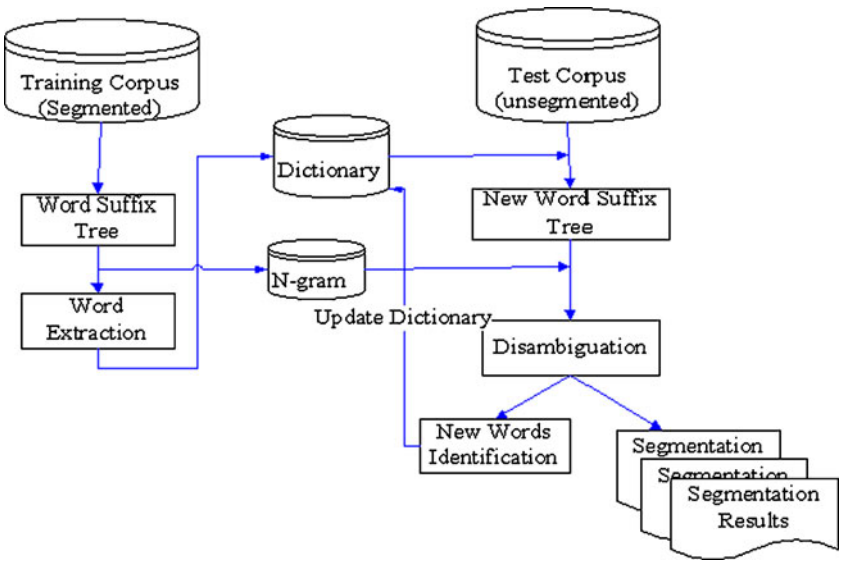

Fig. 1 Overall architecture of our system

in which *p(w)* indicates the probability of a character or string *w* appearing in a given corpus.

Using bigram (1-order Markov model), we have:

$$p(w_1w_2...w_n) = p(w_1)p(w_2/w_1)p(w_3/w_2)...p(w_n/w_{n-1})$$

in which $p(w_i|w_j)$ indicates the probability of a character or string w_i following w_j given that w_j appears in the given corpus.

Using trigram (2-order Markov model), we have:

$$p(w_1w_2...w_n) = p(w_1)p(w_2/w_1)p(w_3/w_1w_2)p(w_4/w_2w_3)...p(w_n/w_{n-2}w_{n-1})$$

For an N-gram model, we have:

$$p(w_1w_2...w_n) = p(w_1)p(w_2/w_1)p(w_3/w_1w_2)$$
$$p(w_4/w_1w_2w_3)...p(w_n/w_1w_2...w_{n-1})$$

The above equations can be expressed by one equation:

$$p(w_1w_2...w_n) = \prod_{i=1}^{n} p(w_i/w_{i-(N-1)}w_{i-(N-1)+1}w_{i-(N-1)+2}...w_{i-1})$$

where $n=length(w_1w_2...w_n)$, in which w_i denotes a characters. *N* is the length of *N*-grams to be considered. Table 1 gives a summary of the expressions used.

To make calculation simpler we often use the following parameter evaluation equations, based on their relative frequencies, using a Maximum Likelihood Estimation (MLE) method.

$$p_{MLE}(w_n/w_{n-1}) = \frac{count(w_{n-1}w_n)}{count(w_{n-1})}$$
$$p_{MLE}(w_n/w_1^{n-1}) = \frac{count(w_1w_2w_3...w_n)}{count(w_1w_2w_3...w_{n-1})}$$
$$p_{MLE}(w_n/w_{n-N+1}^{n-1}) = \frac{count(w_{n-N+1}^{n-1}w_n)}{count(w_{n-N+1}^{n-1})}$$

For an N-gram model a large number of parameters need to be estimated. In this study, we develop an Improved Bigram approach to deal with longer grams. More details will be discussed in the following subsection.

3.2 MI measure

3.2.1 Basic concepts and MI calculation

The concept of Mutual Information comes from Information Theory, which measures two events' dependence (compactness) using:

$$MI(w_i, w_j) = \log_2 \frac{p(w_i, w_j)}{p(w_i)p(w_j)},$$

In Natural Language Processing (NLP), $MI(w_i,w_j)$ is used to estimate the *compactness* of two characters: w_i, w_j. If $MI(w_i,w_j)$ is higher than a given threshold value (usually estimated through experiments, denoted by μ), we can regard them as one word. To simplify calculation, we define $p_{MLE}(w_i)=f(w_i)=count(w_i)/M$, where M denotes the number of characters in the given corpus. As such, we can rewrite the MI equation as follows:

$$MI(w_i, w_j) = \log_2 \frac{count(w_i, w_j)M}{count(w_i)count(w_j)}$$

Researchers (Fang and Yang 2005) also have used the following equation to calculate the *MI* of two bigrams *"w_iw_j"* and *"w_pw_q"*, each of them being a bigram string, and achieved better performance in such cases where a four-character word is composed of two bigrams, e.g., "高音喇叭 (high pitch loudspeaker)", in which both "高音 (high pitch)" and "喇叭 (loudspeaker)" are words.

$$MI(w_iw_j, w_pw_q) = \log_2 \frac{Gf(w_iw_j, w_pw_q)}{K \times f(w_iw_j)f(w_pw_q)}$$

where *G* is the total number of characters in the corpus, *K* is the total number of the tokens in the corpus.

In our approach, however, we do not adopt this method because we have different classes of n-grams but just use one threshold. In order to come up with one measure standard, we use *MLE* calculation equations together with

Table 1 Expressions in equations

Expression	Meaning
w_i	A character
length(*str*)	Number of characters in a *str*
n	Length of the given string
w_i^j	Simple expression of string of $w_iw_{i+1}...w_j$, $i<j$
p (*w*)	Probability of string *w* in a given corpus
$count(w_1w_2...w_n)$	Frequency of n-gram $w_1w_2...w_n$ in a given corpus
p(*x y*)	Probability of co-occurrence of *x*, *y*, where *x*, *y* can be a character or a string
f(*x*)	Frequency estimate of *x*
N	Length of *N*-grams to be considered
M	Number of training instances

the *Lidstone* flatness algorithm to avoid the sparseness of the co-occurrences.

3.2.2 Flatness algorithm

Most of the probabilities involved in the MI calculation are very small and can result in *zero* probability. To avoid numerical problems associated with these zero probabilities, we use the *Lidstone* flatness function:

$$P_{Lid}(w_1 \ldots w_n) = \frac{count(w_1 \ldots w_n) + \lambda}{A + \lambda * B}$$

where $\lambda = 0.5$, B is the number of bins that the training instances are divided into, and A is the corpus size (number of tokens).

For the forecast, we use

$$p(w_1^j / w_1^i) = \frac{count(w_1^i, w_1^j)}{count(w_1^i) + \lambda B}, i \geq 1, j \geq 1, 0 < \lambda < 1$$

Especially, to deal with the probability of single word w_1^i, using following:

$$p(w_1^i) = \frac{count(w_1^i) + \lambda}{A + \lambda B}, 0 < \lambda < 1, i \geq 1$$

on condition that w_1^i, w_1^j, are non-empty strings.

In our research, we store the tokens and their frequencies. If we need to calculate their MI, we first retrieve the tokens and their frequencies, then calculate the MI using the equations described earlier.

3.2.3 Improved bigram

In this subsection we describe our improved bigram model used to deal with N-gram words with $n \geq 3$. This "Improved Bigram" model, when coupled with Suffix Tree, can be viewed as an efficient implementation of an N-gram model. A key motivation behind this less direct N-gram approach is that the Mutual Information calculations (and filtering) can take place much more naturally in our approach.

We store patterns using a hash table for MI computation with the unigram and simple bigram. This approach allows us to process multi-gram words, such as 4-gram words and 5-gram words, and even more parameters prediction.

We show an example of our improved Bigram in Fig. 2. Consider the words: "东/土耳其/斯坦/信息/中心 (East Turkistan Information Center)". The MI of "土耳其" and "斯坦" will be calculated. As this is greater than the threshold, the frequency of the term "土耳其斯坦 (Turkistan)" will be stored and the MI of "东" and "土耳其斯坦" will be further calculated to obtain the correct term "东土耳其斯坦(East Turkistan)".

This method is useful for segmenting terms that are named entities, such as combining ″邓/小平″ to ″邓小平(Deng Xiaopeng)", and ″中国/人民/银行″ to ″中国人民银行 (The People's Bank of China) ", etc. It can also deal with some ambiguous pairs, such as "新西兰花". For instance, if the training corpus are mostly about vegetables (i.e., having a lot of individual occurrences of "西兰花"), then it would split into "新/西兰花 (new broccoli)"; but if the corpus are about the country (i.e., having a lot of individual occurrences of "新西兰 (New Zealand)"), then it would be segmented as "新西兰/花 (New Zealand flowers)".

3.3 Suffix tree

Given a string $S \in \Sigma^n$, the suffix tree T_S of S is the Compact Trie of all the suffixes of $S\$$, $\$ \notin \Sigma$. The suffix tree is the basic data structure in combinatorial pattern matching because of its many elegant uses. Furthermore, it has a compact $O(n)$ space representation that can be constructed in $O(n)$ optimal time for a constant-size alphabet (Weiner 1973). The original construction and its analysis are nontrivial. Some efforts have been spent on producing simplified linear time algorithms (Chen and Seiferas 1985; McCreight 1976), though all such efforts have been variants of the original approach of Weiner. Due to limited space, readers are referred to other papers for the details of the model and implementation of suffix tree (e.g., Chan et al. 2005; Chen and Seiferas 1985; Giegerich and Kurtz 1997; Maaß 1999; McCreight 1976; Weiner 1973; Zhang et al. 2004).

In this study we use the *Ukkonen* algorithm to construct the Suffix Tree (Ukkonen 1992; Ukkonen 1995). The algorithm is an online arithmetic that builds suffix tree from left to right, i.e. adding t_{i+1} (a new character or label edge) to the current suffix tree $STree(T_n)$ and forms another new suffix tree $STree(T_{n+1})$, $0 < n \leq |T|$, where $|T|$ is the length of the string.

For example, the term "中国伊斯兰伊斯兰教会 (Chinese Islamic Shariah Council)" has the following suffix strings: "中国伊斯兰伊斯兰教会", "国伊斯兰伊斯兰教会", "伊斯兰伊斯兰教会", "斯兰伊斯兰教会", "兰伊斯兰教会", "伊斯兰教会", "斯兰教会", "兰教会", "教会", and "会". With the Ukkonen algorithm, these strings are constructed as the Suffix Trie as shown in the top of the Fig. 3, and then compressed to the tree at the bottom of Fig. 3.

3.4 Lexicon construction

Our algorithm uses the known words (dictionary) generated from the previous stage (training processing) to segment the test corpus through the FMM heuristic rule. In order to improve the efficiency of matching, our system stores the dictionary using a Trie structure, and sorts the items

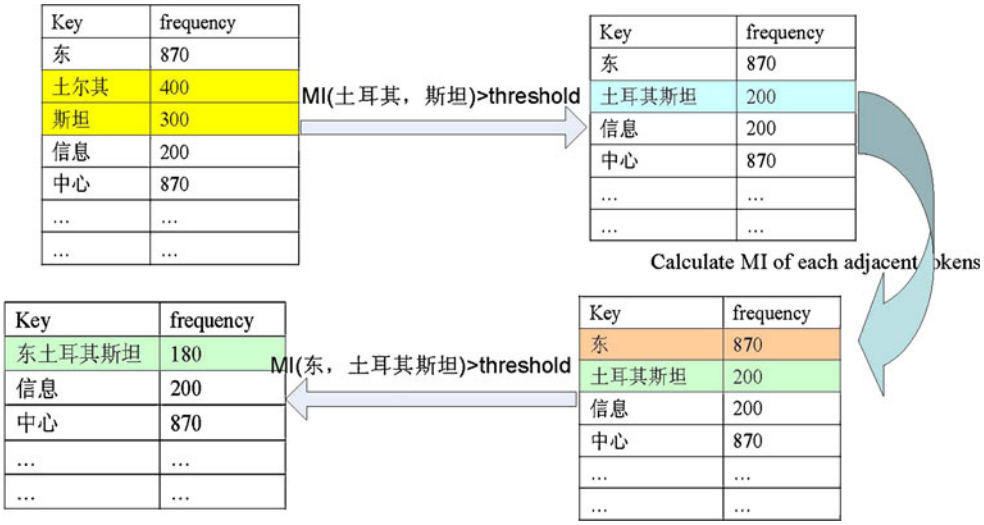

Fig. 2 Example of improved bigrams

according to the order of the Chinese characters based on their Pinyin Romanization.

The construction steps are as follows:

1. The entire dictionary is stored as a Forest;
2. Each tree contains all the words which have the same first character;
3. All the second characters of these words in the same tree are children of the root node;
4. Other characters follow the same token.

3.5 Comparison with existing methods

Previous research has used mutual information for Chinese word segmentation. For example, both Chien (1997) and Ong and Chen (1999), utilize MI in their key phrase extraction. Our proposed algorithm is different from these existing studies. First, MI is used in different ways and different stages in our segmenter. Chien (1997) first split a given string into tokens with different lengths and use MI to filter out the strings with an MI value lower than the threshold. Ong and Chen (1999) extend Chien's work by suggesting an updateable PAT-tree that allows the update of string frequencies dynamically. Different from their methods, we first split a given string coarsely, then compute MI of the neighbor tokens, and compare the MI value with the threshold. If the MI value is higher, then we merge the tokens and add the new word into the dictionary. Otherwise, we keep them unmerged. Another major difference is that we use a hybrid approach. In the first stage, we perform coarse splitting using a dictionary-based method to split the given texts, while the other two methods directly compound the characters according to their compositions.

In this study, we propose a hybrid method to avoid the limitations of the pure dictionary-based methods or the pure statistic methods. Through the Suffix Tree and Mutual Information with dictionary, we achieve the accurate segmentation when domain training is possible. Compared with the early methods, it can reach the goal of new words identification through MI-threshold and the token merge algorithm, and update the dictionary. In addition, with the Improved Bigram and Suffix Tree, it can also process N-grams efficiently.

4 Evaluation

In order to evaluate the performance of our IASeg system, we compare it with the Hylanda segmenter (www.hylanda.com) and the ICTCLAS segmenter (Zhang et al. 2003). The Hylanda segmenter is a dictionary-based segmenter that has been widely used in practice (e.g., the search engine Zhongsou). ICTCLAS is an HMM-based segmenter. Both segmenters were chosen because they have shown very good performance in previous studies (Leydesdorff and Zhou 2008; Zhang et al. 2003).

We use precision, recall, and F-measure to evaluate the performance of the segmenters. The calculations are as follows:

$$precision = \frac{correctNum}{autoTotalNum}$$
$$recall = \frac{correctNum}{manualTotalNum}$$
$$F - measure = \frac{2 \times recall \times precision}{recall + precision}$$

where *correctNum* is the number of words correctly identified by the automatic method, *autoTotalNum* is the total number of words identified by the automatic method, and *manualTotalNum* is the number of words identified in the manual segmentation. A perfect segmenter will have a recall and precision of 100%. All these measures can be calculated automatically from a machine-segmented text, along with the human-segmented gold standard.

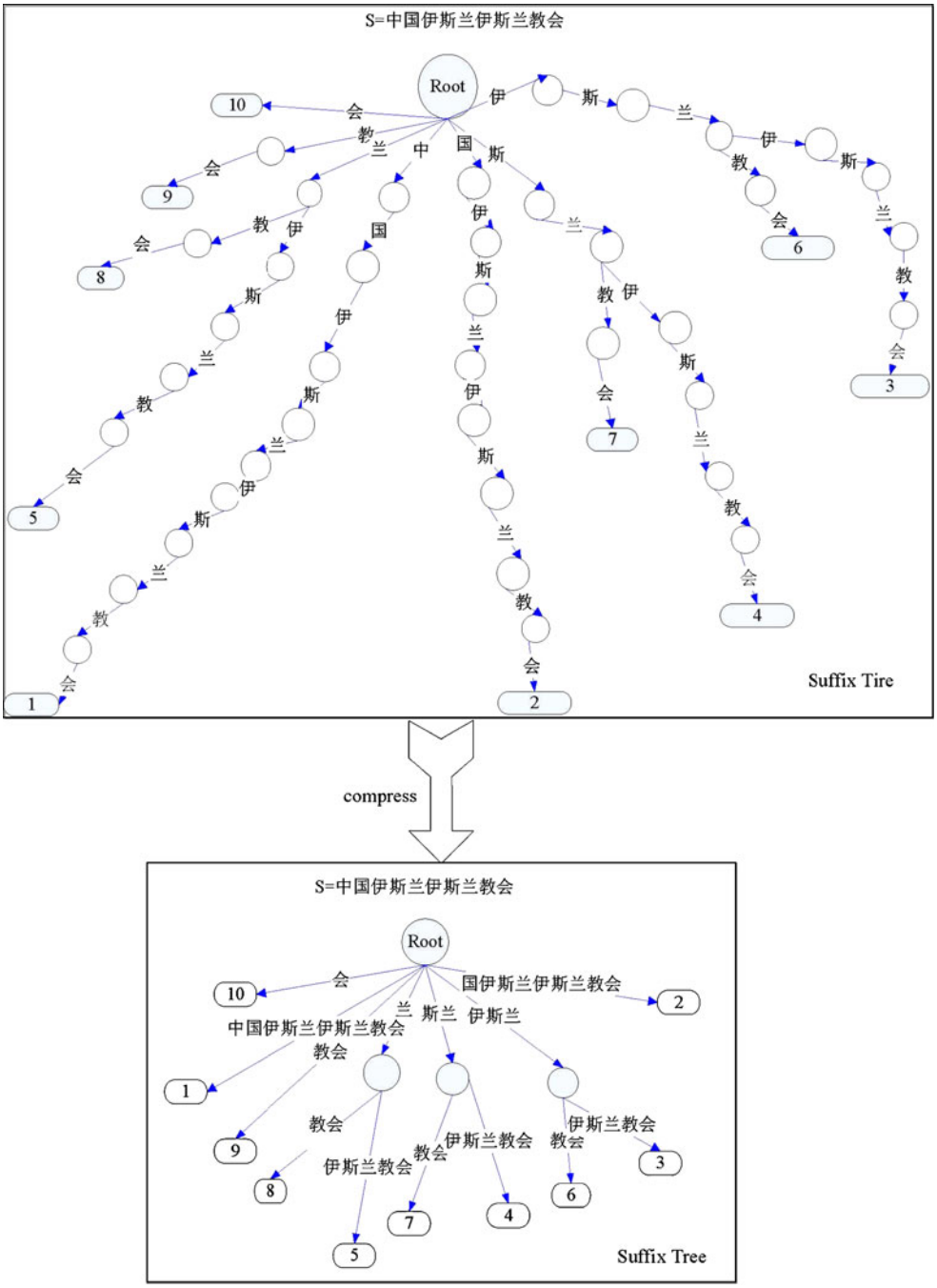

Fig. 3 Example of suffix trie compression

4.1 Experiment 1

To test our segmenter in domain-specific Chinese word segmentation, we chose to evaluate its performance on terrorism-related contents. Extremists and terrorists have been using the Internet to spread their ideology and recruit new members (Chen 2006; Raghu and Chen 2007). It has been shown possible to extract useful extremist or terrorist information and identify their relationships from these documents (Chen and Xu 2006, Chau and Xu 2007).

The past terrorism informatics research has primarily focused on English and Arabic contents. Our research focuses on terrorism-related contents that are mostly Chinese but also frequently contain English phrases (e.g., names of organizations), a characteristic of the real-world data. To the best of our knowledge, there is no publically-available terrorism-related Chinese corpus available for experimentation. In our experiment, we collected a set of terrorism-related Chinese documents from the Web using a home-grown Web crawler as follows. Using as the seeds a manually-constructed list of 82 URLs which contain terrorism-related contents, our crawler retrieved about 500 Web pages, mostly terrorism-related news articles. A pre-processing module was used to discard irrelevant contents such as advertising anchor text and promotional Web links. Two

graduate students experienced in terrorism informatics studies then went through all these cleansed documents and selected 342 news articles with direct terrorism relevance. An automated module using a coarse-grained open-source Chinese word segmentation subroutine was then used to heuristically go through all these identified articles to identify duplications. After removing 12 duplicated articles, we finally obtained our Chinese terrorism dataset consisting of 330 news articles. For each of these articles, the same two graduate students manually produced segmentation results, using the results from the open-source Chinese word segmentation subroutine obtained earlier as the starting points. Their segmentation results were independently validated by another graduate student whose research area is Chinese language processing. The gold sets and training sets used in our experiments were all sampled from these 330 segmented articles.

We ran the three segmenters under comparison using this terrorism corpus. In our algorithm, the dictionary trie is updated dynamically after the first run of the training corpus. Finally we obtained 57,339 words through the terrorism corpus, including some highly frequent general words.

The segmentation results of these three segmenters are shown in Table 2. Overall we observe that our segmenter achieves the best performance among the three segmenters in terms of precision, recall, and F-measure. To provide for better validity of the results, we also performed cross-validation testing on IASeg. The corpus was manually divided into 20 portions of roughly-equal sizes with the size measured by the total number of characters each portion contains. We have partitioned the documents to roughly equal-sized portions to reduce the impact of the size of an individual portion on the performance of the methods compared. The average precision, recall, and F-measure obtained are 0.959, 0.950, and 0.955, respectively. We suggest that the higher performance in cross validation is possibly due to the availability of a larger training corpus than in the hold-out test.

4.2 Experiment 2

In addition to the terrorism-specific corpus, we also compare the three segmenters using a general corpus in order to demonstrate their performances for general texts. In this experiment, we use the corpus provided by the Second International Chinese Word Segmentation Bakeoff. This competition was held in 2005 and the results were presented at the 4th SIGHAN Workshop (http://sighan.cs.uchicago.edu/bakeoff2005/). In particular, we use the corpus provided by Microsoft Research (MSR corpus) at the bakeoff. The MSR corpus contains email addresses, URLs, etc., and adopts its own evaluation standard. The segmentation results of the three segmenters are shown in Table 3.

Since Hylanda and ICT are general-purpose segmenters and IASeg is a domain-specific segmenter, we expect that the Hylanda and ICT segmenters perform better on the MSR corpus than on the terrorism corpus, while the IASeg performs better on the terrorism corpus. The results from Tables 2 and 3 show that the IASeg performs better on the terrorism content but also has comparable performance for the general corpus. In terms of F-measure, the performance of the Hylanda segementer improves from 0.887 to 0.905, and ICT improves from 0.822 to 0.875. On the other hand, the performance of IASeg drops from 0.948 to 0.908. Nonetheless, it is still the highest among the three segmenters being evaluated.

4.3 Experiment 3

In the third experiment, we want to evaluate the performance of the IASeg sgementer when we do not have a training corpus in the given domain. In this experiment we train the IASeg segmenter using the MSR corpus, and test the segmenter using the terrorism content used in Experiment 1. The segmenter achieves a precision, recall, and F-measure of 0.746, 0.853, and 0.796, respectively. As expected, these values are not as good as the results obtained when the segmenter has access to domain-specific training corpus. The results show that the segmenter is more effective when domain-specific training corpuses are available.

5 Discussion

Overall, our experiments demonstrate the following:

1. When using terrorism-related content for training, IASeg performs better than the two benchmark systems in segmenting terrorism-related content.

Table 2 Results of the three segmenters on terrorism-related content

	Hylanda	ICT	IASeg
Precision	0.860	0.776	0.948
Recall	0.916	0.866	0.948
F-measure	0.887	0.822	0.948

Table 3 Results of the three segmenters on the general content

	Hylanda	ICT	IASeg
Precision	0.899	0.850	0.902
Recall	0.910	0.902	0.914
F-measure	0.905	0.875	0.908

2. When using a general corpus for training, the performance of IASeg drops, but it still performs better than the two benchmark systems in segmenting general content.

Based on our testing of the segmenter on the terrorism-related corpus and the general corpus, we found that two aspects of the training data have a profound influence on the model's accuracy. First, some errors are obviously caused by deficiencies in the training data, such as improperly segmented common words and name entities. Second, some errors stem from the split of the training and testing data.

We observe that our algorithm has the following characteristics:

1. Different thresholds in mutual information will achieve different results. For example, a threshold of *20* may just keep all the tokens in their original form, while a threshold of *9* will result in merging some high co-occurrence adjacent tokens as one word. In general, we found that a lower threshold will make the segmenter to prefer longer words, thus resembling more closely with named entity extraction tools.
2. By using suffix tree, we can do searching and matching more easily and efficiently. Using the *Ukkonen* algorithm, we can construct the suffix tree in *O(n)* time complexity and *O(n)* space complexity.
3. Through our improved bigram structure, we can filter the low MI token-pairs, which greatly improves the boundary forecast accuracy.

We also found that different segmentation standards assumed by the corpus will greatly affect the segmentation results. For example, it is not easy even for humans to judge whether the term "中国人民 (Chinese people)" should be treated as one word or segmented into two. Such different standards used in different corpuses have make it difficult, if not impossible, to have a perfect segmenter.

Chinese word segmentation research reported in this paper is a technical component of a major terrorism informatics research project initiated by the Chinese Academy of Sciences, focusing on analyzing open-source (mostly Chinese) information from the Web and developing a large-scale computing platform to enable such analyses. The particular application-domain emphasis has been on terrorist and separatist groups that have engaged in violent acts in China. As a critical and necessary component of this terrorism informatics platform, the reported Chinese word segmentation module has been used to pre-process all the incoming Web data streams to prepare data for the follow-up analyses ranging from entity-extraction, sentiment information detection and polarity assessment, relation extraction, text summarization and visualization, to domain event extraction and cultural-computing based group behavior prediction.

6 Conclusion

In this paper, we propose a method on Chinese word segmentation based on suffix tree and mutual information. We integrate character-level information and word-level information and achieve encouraging results in segmenting a terrorism-related corpus. Our algorithm uses a two-stage statistical word segmentation. In the first stage, word suffix tree are used to generate a dynamic dictionary and N-gram model on input text, and then a hybrid approach is employed in the second stage to incorporate word N-gram probabilities, and mutual information with word-formation patterns to detect Out-Of-Vocabulary words. From our experiment, we found that the proposed segmenter was able to achieve good performance in Chinese word segmentation, particularly for domain-specific corpuses.

Our future work includes the following:

- Improve our strategies by adding more words' position information and part-of-speech to develop an integrated segmenter which can perform known word segmentation and unknown word identification at the same time.
- Address the OAS (overlap ambiguity string) problem using syntax rules and address the "CAS" (combination ambiguity string) problem using SVM classifier.
- Study the possibility of performing Chinese named entity recognition using the HMM-based tagger and its integration with this Chinese analyzer.
- Investigate the problem of event information extraction based on syntax structure.

Acknowledgments The reported work was supported in part by the following grants: NNSFC #90924302 and #60921061, MOST #2006AA010106, CAS #2F07C01, NSF #IIS-0428241, and HKU #10207565. We thank our team member Mr. Qingyang Xu for his help with the experiments. We also thank Ms. Fenglin Li and Ms. Shufang Tang for their help with data preparation and processing.

References

Chan, H. L., Hon, W. K., Lam, T. W., Sadakane, K. (2005) Dynamic dictionary matching and compressed suffix trees. *Proceedings of the sixteenth annual ACM-SIAM symposium on discrete algorithms*, Society for Industrial and Applied Mathematics. ISBN: 0-89871-585-7.

Chau, M., & Xu, J. (2007). Mining communities and their relationships in blogs: a study of online hate groups. *International Journal of Human-Computer Studies, 65*(1), 57–70.

Chen, H. (2006). Intelligence and security informatics: information systems perspective. *Decision Support Systems, 41*(3), 555–559.

Chen, M. T., Seiferas, J. (1985). Efficient and elegant subword-tree construction. *Combinatorial Algorithm on Words* (pp 97–107). NATO Advanced Science Institutes, Series F, vol. 12, Springer, Berlin.

Chen, H., & Xu, J. (2006). Intelligence and security informatics. *Annual Review of Information Science and Technology, 40*, 229–289.

Chien, L. F. (1997). PAT-tree based keyword extraction for chinese information retrieval. *ACM SIGIR*

Creutz, M., Lagus, K. (2007). Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing*, Volume 4, Issue 1.

Cui, S. Q., Liu, Q., Meng, Y., Yu, H., & Nishino, F. (2006). New word detection based on large-scale corpus. *Journal of Computer Research and Development, 43*(05), 927–932.

Dai, Y. B., Khoo, S. G. T., Loh, T. E. (1999). A new statistical formula for Chinese word segmentation incorporating contextual information. In: *Proc. of the 22nd ACM SIGIR Conf. on Research and Development in Information Retrieval* (pp 82–89).

Fang, Y., Yang, H. E. H. (2005). The algorithm design and realization to calculate the mutual information of four-word-string in large scale corpus. *Computer Development & Applications*, Vol.1.

Giegerich, R., & Kurtz, S. (1997). From Ukkonen to McCreight and Weiner: a unifying view to linear-time suffix tree construction. *Algorithmica, 19*, 331–353.

Hockenmaier, J., & Brew, C. (1998). Error-driven segmentation of Chinese. *Communications of COLIPS, 1*(1), 69–84.

Jia, N., & Zhang, Q. (2007). Identification of Chinese names based on maximum entropy model. *Computer Engineering, 33*(9), 31–33.

Leydesdorff, L., & Zhou, P. (2008). Co-word analysis using the Chinese character set. *Journal of the American Society for Information Science and Technology, 59*(9), 1528–1530.

Li, J. F., & Zhang, Y. F. (2002). Segmenting Chinese by EM algorithm. *Journal of the China Society for Scientific and Technical Information, 03*, 13–16.

Li, R., Liu, S. H., Ye, S. W., & Shi, Z. Z. (2001). A method of crossing ambiguities in Chinese word segmentation based on SVM and k-NN. *Journal of Chinese Information Processing, 15* (6), 13–18 (in Chinese).

Low, J. K., Ng, H. T., Guo, W. (2005). A maximum entropy approach to Chinese word segmentation. *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing* (pp 161-164). Jeju Island, Korea.

Maaß, M. (1999). Suffix trees and their applications. *Ferienakademie* 1999 Kurs 2: *Bäume: Algorithmik and Kombinatorik*.

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

Ong, T. H., Chen, H. (1999). Updateable PAT-tree approach to chinese key phrase extraction using mutual information: a linguistic foundation for knowledge management. In *Proceedings of the Asian Digital Library Conference* (pp 63-84). Taipei, Taiwan.

Palmer, D. (1997). A trainable rule-based algorithm to word segmentation. In *Proceedings of the 35th Annual Meeting of the Association of Computational Linguistics*. Madrid, Spain.

Peng, F. C., Schuurmans D. (2001). Self-supervised Chinese word segmentation. *Proceedings of the 4th International Symposium of Intelligent Data Analysis* (pp 238–247).

Peng, F. C., Feng, F. F., McCallum, A. (2004). Chinese segmentation and new word detection using conditional random fields. *COLING 2004*, Geneva, Switzerland.

Ponte, J. M., Croft, W. B. (1996). Useg: a retargetable word segmentation procedure for information retrieval. In *Proceedings of SDAIR96*, Las Vegas, Nevada.

Raghu, T. S., & Chen, H. (2007). Cyberinfrastructure for homeland security: advances in information sharing, data mining, and collaboration systems. *Decision Support Systems, 43*(4), 1321–1323.

Sproat, R., Shih, C., Gale, W., & Chang, N. (1996). A stochastic finite-state word-segmentation algorithm for Chinese. *Computational Linguistics, 22*(3), 377–404.

Sun, M. S., Xiao, M., & Zou, J. Y. (2004). Chinese word segmentation without using dictionary based on unsupervised learning strategy. *Chinese Journal of Computers, 27*(6), 736–742.

Teahan, W. J., Wen, Y., McNab, R. J., & Witten, I. H. (2000). A compression-based algorithm for Chinese word segmentation. *Computational Linguistics, 26*, 375–393.

Ukkonen, E. (1992). Constructing suffix trees on-line in linear time. In Jv Leeuwen (ed), *Proc. IFIP 12th World Computer Congress on Algorithms, Software, Architecture* (pp 484–492) Madrid, Spain.

Ukkonen, E. (1995). On-line Construction of Suffix-Trees. *Algorithmica*, 14(3).

Weiner, P. (1973). Linear pattern matching algorithms. *Proc. 14th IEEE Annual Symp. on Switching and Automata Theory* (pp 1-11).

Wong, P.-k., Chan, C. (1996). Chinese word segmentation based on maximum matching and word binding force. *Proceedings of the 16th International Conference on Computational Linguistics* (pp 200–203).

Wu, Z., & Tseng, G. (1993). Chinese text segmentation for text retrieval achievements and problems. *Journal of the American Society for Information Science, 44*(9), 532–542.

Xue, N. W. (2003). Chinese word segmentation as character tagging. *International Journal of Computational Linguistics and Chinese Language Processing, 8*(1), 29–48.

Xue, N.W., Chiou, Fu-Dong, and Palmer, M. Building a large annotated Chinese corpus. In *Proceedings of the 19th International Conference on Computational Linguistics*. Taipei, Taiwan, 2002.

Yu, H. K., Zhang, H. P., Liu, Q., Lv, X. Q., & Shi, S. C. (2006). Chinese named entity identification using cascaded hidden Markov model. *Journal on Communications, 27*(2), 87–94.

Zhang, H. P., Yu, H. K., Xiong, D. Y., Liu Q. (2003). HMM-Based Chinese lexical analyzer ICTCLAS. In *Proc. of the 2nd SIGHAN Workshop* (pp 184–187).

Zhang, C. L., Hao, F. L., Wan, W. L. (2004). An automatic and dictionary-free Chinese word segmentation method based on suffix array. *Journal of Jilin University (Science Edition)*, Vol 4.

Zhou, L. X., Liu, Q. (2002). A Character-net Based Chinese Text Segmentation Method, *SEMANET: Building and Using Semantic Networks Workshop* at the 19th COLING (pp 101–106).

Daniel Zeng received the M.S. and Ph.D. degrees in industrial administration from Carnegie Mellon University and the B.S. degree in economics and operations research from the University of Science and Technology of China, Hefei, China. He is a Research Professor at the Institute of Automation in the Chinese Academy of Sciences and a Professor and Honeywell Fellow in the Department of Management Information Systems at the University of Arizona. Zeng's research interests include intelligence and security informatics, spatial-temporal data analysis, infectious disease informatics, social computing, recommender systems, software agents, and applied operations research and game theory with application in e-commerce and online advertising systems. He has published one monograph and more than 170 peer-reviewed articles. His research has been mainly funded by the U.S. National Science Foundation, the National Natural Science Foundation of China, the Chinese Academy of Sciences, the U.S. Department of Homeland Security, the Ministry of Science and Technology of China, and the Ministry of Health of China.

Donghua Wei is currently a banking and financial system manager in the Postal Savings Bank of China. She received her M.S. degree in Computer Application Technology from Beijing Normal University and her B.S. degree in management information systems from Beijing Technology and Business University. Her research interests include information retrieval and Web mining. She has published several conference papers at PAISI 2007, ICCSE 2007, and PAISI 2008.

Michael Chau is an Assistant Professor in the School of Business at the University of Hong Kong. He received his Ph.D. degree in management information systems from the University of Arizona and a bachelor degree in computer science and information systems from the University of Hong Kong. His current research interests include information retrieval, Web mining, data mining, knowledge management, electronic commerce, and security informatics. He has published more than 80 research articles in leading journals and conferences, including *IEEE Computer*, *Journal of the America Society for Information Science and Technology*, *Journal of the Association for Information Systems, Decision Support Systems*, and *Communications of the ACM*. He is a senior member of ACM, AIS, and IEEE. More information can be found at http://www.business.hku.hk/~mchau/.

Fei-Yue Wang received his Ph.D. in Computer and Systems Engineering, minor in Computer Science, from the Rensselaer Polytechnic Institute (RPI), Troy, New York, USA, in 1990. He joined the University of Arizona in 1990 and is the Professor of Systems and Industrial Engineering and Director of the Program for Advanced Research in Complex Systems. In 1998, he founded the Intelligent Control and Systems Engineering Center at the Institute of Automation, Chinese Academy of Sciences, Beijing, China. Since 2002, he has been the Director of the Key Laboratory of Complex Systems and Intelligence Science at the Chinese Academy of Sciences. His current research interests include intelligent control systems; social computing; modeling, analysis, and control mechanism of complex systems. He has published more than 200 books, book chapters, and papers in those areas since 1984 and received research funding from NSF, DOE, DOT, NNSF, CAS, MOST, Caterpillar, IBM, HP, AT&T, GM, BHP, RVSI, ABB, and Kelon. He is an elected Fellow of the Institute of Electrical and Electronics Engineers (IEEE), International Council of Systems Engineering (INCOSE), International Federation of Automatic Control (IFAC) and the American Association for the Advancement of Science (AAAS).