# A Unified Character-Based Tagging Framework for Chinese Word Segmentation

HAI ZHAO
Shanghai Jiao Tong University & Soochow University
CHANG-NING HUANG and MU LI
Microsoft Research Asia
BAO-LIANG LU
Shanghai Jiao Tong University

Chinese word segmentation is an active area in Chinese language processing though it is suffering from the argument about what precisely is a word in Chinese. Based on corpus-based segmentation standard, we launched this study. In detail, we regard Chinese word segmentation as a character-based tagging problem. We show that there has been a potent trend of using a character-based tagging approach in this field. In particular, learning from segmented corpus with or without additional linguistic resources is treated in a unified way in which the only difference depends on how the feature template set is selected. It differs from existing work in that both feature template selection and tag set selection are considered in our approach, instead of the previous feature template focus only technique. We show that there is a significant performance difference as different tag sets are selected. This is especially applied to a six-tag set, which is good enough for most current segmented corpora. The linguistic meaning of a tag set is also discussed. Our results show that a simple learning system with six $n$-gram feature templates and a six-tag set can obtain competitive performance in the cases of learning only from a training corpus. In cases when additional linguistic resources are available, an ensemble learning technique, assistant segmenter, is proposed and its effectiveness is verified. Assistant segmenter is also proven to be an

effective method as segmentation standard adaptation that outperforms existing ones. Based on the proposed approach, our system provides state-of-the-art performance in all 12 corpora of three international Chinese word segmentation bakeoffs.

## 1. INTRODUCTION

Chinese text is written without natural delimiters such as white spaces, so word segmentation is often an essential first step in Chinese language processing. Though it seems simple, Chinese word segmentation is actually not a trivial problem, and has been an active research area in computational linguistics for more than 20 years [Fan and Tsai 1988; Sproat and Shih 1990; Sproat et al. 1996; Sun et al. 1998; Sun and Tsou 2001; Sproat and Shih 2002; Gao et al. 2005].

Chinese word segmentation is challenging partially due to the difficulty in defining what encompasses a word in Chinese. In the literature, there are various linguistic criteria in the theoretical linguistic community [Packard 2000], each of which provides valuable insight to Chinese "word-hood." Though these definitions or standards are in general agreement with each other, there are specific instances in which they are not. Fortunately, the rapid development of corpus linguistics has allowed a segmentation standard to be effectively represented by a segmented corpus. This brings an obvious convenience in explicitly or implicitly avoiding those ambiguities or conflicts inside segmentation standards. In addition, this provides broader word description than a guideline manual. The drawbacks of a rule-based method, used in a corpus-based method, can be overcome by enlarging the corpus [Sproat and Shih 2002].

Chinese word segmentation can be classified into two categories: dictionary-based methods and statistical methods.

The most successful dictionary-based methods are variations of the maximum matching algorithm, which greedily searches through a sentence in an attempt to find the longest subsequence of Chinese characters that matches a word entry in a pre-compiled dictionary [Nie et al. 1994]. Typically, a dictionary-based approach addresses the ambiguity problem with some heuristics. There exist two kinds of ambiguities in Chinese word segmentation when using the dictionary approach. One is overlapping ambiguity, which can be roughly detected by a mismatch from forward maximum matching (FMM) and backward maximum matching (BMM). The other is combination ambiguity, which can be defined by an uncertain decision to split a character sequence

when both the whole character sequence and all its members exist in the dictionary [Tsai 2006]. To solve the ambiguity problem well, many techniques have been developed, including various kinds of statistical learning methods [Luo et al. 2002; Li et al. 2003; Sun et al. 2006].

The performance of dictionary-based methods largely depends upon the coverage of the dictionary. However, it is difficult to compile a complete dictionary due to the appearance of out-of-vocabulary (OOV) words (namely, unknown words). Thus, researchers turn to statistic-based methods to better deal with OOV words, and in particular, OOV detection. There are two strategies that handle OOV word detection. While Wu and Jiang [2000] and Chen [2003] handle it separately, Sproat et al. [1996], Xue [2003], and Gao et al. [2005] treat it as part of the segmentation procedure. In this article, we unify OOV word detection and Chinese word segmentation.

The current trend in OOV word detection is to employ character-based methods. Since all Chinese words are formed by a closed set of characters of about 6,500 or about 3,500 in most cases,[1] it is most straightforward to regard OOV word detection as combining these successive single characters within a focused scope[2] Yuan [1997] and Fu and Wang [1999] tackled OOV word detection based on four word-formation patterns and head-middle-tail structures. Xue [2003] pioneered character-based tagging method via maximum entropy (MaxEnt) modeling. Since then, much attention has been paid to character-based methods, and various learning models such as support vector machines (SVMs) and conditional random fields (CRFs) have been employed within this framework [Peng et al. 2004; Tseng et al. 2005; Goh et al. 2005; Zhu et al. 2006].

In this study, we focus on a corpus-based learning method for Chinese word segmentation. With the availability of a large segmented corpus, the acquisition of word segmentation information is regarded as a supervised learning of the segmented corpus.

Motivated by linguistic facts, we classify characters in Chinese text into more categories according to their positions in words than those in previous work [Xue 2003]. Such a categorization extension is transformed into an extension of the tag set that is essentially used in learning procedure, which brings a general trend of performance enhancement. In addition, we explore ensemble learning for effectively integrating linguistic resources with CRF itself adopted as the ensemble scheme and an assistant segmenter approach proposed to deal with various additional linguistic resources (outside the training corpus and including lexicons, named entity information, and segmented corpora with different standards). All these construct a general scheme for properly integrating various kinds of linguistic information into Chinese word segmentation.

---

[1]Here, the term "character" indicates all types of characters that could appear in a real Chinese text, including Chinese characters (namely, Hanzi), Arabic numerals, letters, etc.

[2]Though modern Chinese character sets normally include approximately 10,000-20,000 characters, much fewer are really used in everyday life. Typically, 2,500 most widely Chinese characters can cover 97.97% text, while 3,500 characters can cover 99.48% text.

Table I. Corpora Statistics of Bakeoff 2003, 2005, and 2006

| Provider | Corpus | Encoding | #Training words | #Test words | OOV rate |
|---|---|---|---|---|---|
| Academia | AS2003 | Big5 | 5.8M | 12K | 0.022 |
| Sinica | AS2005 | Big5 | 5.45M | 122K | 0.043 |
| | AS2006 | Big5 | 5.45M | 91K | 0.042 |
| City University of | CityU2003 | Big5 | 240K | 35K | 0.071 |
| Hong Kong | CityU2005 | Big5 | 1.46M | 41K | 0.074 |
| | CityU2006 | Big5 | 1.64M | 220K | 0.040 |
| University of | CTB2003 | GB | 250K | 40K | 0.181 |
| Pennsylvania | CTB2006 | GB | 508K | 154K | 0.088 |
| Microsoft | MSRA2005 | GB | 2.37M | 107K | 0.026 |
| Research Asia | MSRA2006 | GB | 1.26M | 100K | 0.034 |
| Peking | PKU2003 | GB | 1.1M | 17K | 0.069 |
| University | PKU2005 | GB | 1.1M | 104K | 0.058 |

In summary, we will describe a suite of complete techniques for Chinese word segmentation and show that Chinese word segmentation can be properly tackled using a unified supervised learning framework, given a segmented corpus, with possible consideration of additional linguistic information.

The remainder of the article is organized as follows. Section 2 discusses technical trends revealed by international Chinese word segmentation bakeoffs. Section 3 briefly introduces conditional random fields. Sections 3.1 and 3.2 describe basic feature template setting and tag set selection, respectively, with their experimental results given in Section 4. Section 5 presents and evaluates an assistant segmenter method for integrating additional linguistic knowledge. Sections 6 and 7 compare and discuss our system with the state-of-the-art ones, respectively. Finally, Section 8 summarizes our contributions in this article.

## 2. THE CORPUS

For a comprehensive comparison of Chinese word segmentation in a common test corpora, SIGHAN held three International Chinese Word Segmentation Bakeoffs in 2003, 2005, and 2006,[3] and attracted 12, 23, and 24 participants, respectively [Sproat and Emerson 2003; Emerson 2005; Levow 2006].

Each bakeoff specifies two types of test tracks: open test, which has no limitation on additional resources, and closed test, which only allows the corresponding training data.

We regard the closed test as a standard supervised learning task from training data alone, and the open test as common characteristics of human language, which is the point in which natural language learning differs from other data learning paradigms.

All the corpora since Bakeoff-2003[4] Table I) are taken as our evaluation data sets.

---

[3]In 2006, the third bakeoff was renamed the International Chinese Language Processing Bakeoff due to the introduction of the named entity recognition task.
[4]The organizers of three bakeoffs adopted different names for the same convention. For example, CityU2003 was noted as HK in Sproat and Emerson [2003], MSRA2005 was noted as MSR in Emerson [2005], and CTB2006 was noted as UPUC and AS2006 as CKIP in Levow [2006].

Table II. Performance Comparison of Topline and Baseline Systems on Different Corpora of Bakeoff-2003

| Corpus | AS2003 | CityU2003 | CTB2003 | PKU2003 |
|---|---|---|---|---|
| Topline | 0.992 | 0.989 | 0.985 | 0.995 |
| Baseline | 0.915 | 0.867 | 0.725 | 0.867 |
| Topline minus Baseline | 0.077 | 0.122 | 0.260 | 0.128 |
| 1 minus Topline | 0.008 | 0.011 | 0.015 | 0.005 |
| Ratio: OOV vs. ambiguity | 9.6 | 11.1 | 17.3 | 25.6 |

Table III. Performance Comparison of Topline and Baseline Systems on Different Corpora of Bakeoff-2005

| Corpus | AS2005 | CityU2005 | MSRA2005 | PKU2005 |
|---|---|---|---|---|
| Topline | 0.982 | 0.989 | 0.991 | 0.987 |
| Baseline | 0.882 | 0.833 | 0.933 | 0.869 |
| Topline minus Baseline | 0.100 | 0.156 | 0.058 | 0.118 |
| 1 minus Topline | 0.018 | 0.011 | 0.009 | 0.013 |
| Ratio: OOV vs. ambiguity | 5.6 | 14.2 | 6.4 | 9.1 |

Table IV. Performance Comparison of Topline and Baseline Systems on Different Corpora of Bakeoff-2006

| Corpus | AS2006 | CityU2006 | CTB2006 | MSRA2006 |
|---|---|---|---|---|
| Topline | 0.983 | 0.984 | 0.976 | 0.993 |
| Baseline | 0.892 | 0.906 | 0.790 | 0.900 |
| Topline minus Baseline | 0.091 | 0.078 | 0.186 | 0.093 |
| 1 minus Topline | 0.017 | 0.016 | 0.024 | 0.007 |
| Ratio: OOV vs. ambiguity | 5.4 | 4.9 | 7.8 | 13.3 |

For evaluation, we adopt the commonly-used recall ($R$), precision ($P$), and $F_1$-measure ( $2RP/(R + P)$ ), where recall is the proportion of correctly segmented words in the gold-standard segmentation, precision is the proportion of correctly segmented words output by the segmenter, and $F_1$-measure is the harmonic mean of recall and precision.

To check which makes the most performance loss in Chinese word segmentation on different corpora, each bakeoff [Emerson 2005; Sproat and Emerson 2003; Levow 2006] adopts the FMM algorithm to generate topline and baseline performance figures. This is done by generating a dictionary based only on the vocabulary in each test (topline) and training (baseline) corpus and segmenting the respective test corpus. In addition, the performance gaps between a perfect system (100%) and the topline, and between topline and baseline,[5] are also presented in Tables II, III, and IV, respectively. If we take the value of 1 minus topline as a metric of ambiguity loss, and topline minus baseline the metric of OOV loss, then from the ratio between OOV loss and ambiguity loss that is given in the bottom row of each table, we see that OOV loss is far more than ambiguity loss (from 4.9 to 25.6 times).

Notice that Xue and Shen [2003] proposed a character-based tagging method using a maximum entropy model and achieved the second rank in closed test of

---

[5]In the case of topline, all words are assumed to be known, that is, without OOV words and only segmentation ambiguities may cause performance loss. Therefore, we adopt 1 minus topline to metric the ambiguity loss.

Table V. The Official Results of Xue and Shen's System in the AS2003 Corpus

| Participant(Site ID) | R | P | F | $R_{OOV}$ | $R_{IV}$ |
|---|---|---|---|---|---|
| Chen [2003](S09) | 0.966 | 0.956 | 0.961 | 0.364 | 0.980 |
| Xue and Shen [2003](S12) | 0.961 | 0.958 | 0.959 | 0.729 | 0.966 |
| Asahara et al. [2003](S06) | 0.944 | 0.945 | 0.945 | 0.574 | 0.952 |

AS2003 corpus among the official results of Bakeoff-2003 (see Table V, where recalls of in-vocabulary (IV) word are also given), which has the highest recall of OOV ($R_{OOV}$) in this track [Sproat and Emerson 2003; Xue and Shen 2003; Asahara et al. 2003; Chen 2003]. Xue and Shen [2003] also obtained the third best result in the CityU2003 closed test with the highest $R_{OOV}$, 0.670. These results imply that the character-based tagging method is effective for OOV word detection. As OOV word detection is much more serious than segmentation ambiguities, it is possible for researchers to get performance enhancement only if suitable techniques are adopted to strengthen this method. This assessment did come true in Bakeoff-2005. In all of proposed methods of Bakeoff-2005, the character-based tagging method quickly rose as the most remarkable in obtaining the best results in almost all test corpora [Low et al. 2005; Tseng et al. 2005].

Based on above investigation, we adopt the character-based tagging framework for Chinese word segmentation in this study.

## 3. THE LEARNING FRAMEWORK

Peng et al. [2004] first used CRFs for Chinese word segmentation by treating it as a binary decision task, such that each Chinese character is labeled either as the beginning of a word or not. Following Peng et al. [2004], we employ the state-of-the-art linear-chain conditional random fields [Lafferty et al. 2001] with tunable Gaussian prior.[6] CRF often outperforms MaxEnt model [Rosenfeld et al. 2006], another popular machine-learning method in NLP. The main reason is that CRF suffers less from the label bias problem when compared to MaxEnt and other conditional Markov models [Lafferty et al. 2001]. So far, CRF has been very successful in a good number of NLP applications [Sha and Pereira 2003].

Given a particular sequence of characters, a CRF computes the probability of its corresponding hidden label sequence as

$$p_\lambda(Y|W) = \frac{1}{Z(W)} \exp\left( \sum_{t \in T} \sum_k \lambda_k f_k(y_{t-1}, y_t, W, t) \right) \qquad (1)$$

where $Y = \{y_t\}$ is the label sequence for the sentence, $f_k$ is a feature function, $\lambda_k$ is the weight value for the corresponding feature function $f_k$, $W$ is the sequence of unsegmented characters, $Z(W)$ is a normalization term, $T$ is the tag set, and

---

[6]Gaussian prior is the only parameter that we should handle for this article, and it is set to 100 throughout the whole article.

Table VI. Basic Feature Templates Set

| Code | Type | Feature | Function |
|------|------|---------|----------|
| (a) | Unigram | $C_{-1}, C_0, C_1$ | Previous, current, or next character |
| (b) | Bigram | $C_{-1}C_0$ | Previous and current characters |
| | | $C_0C_1$ | Current and next characters |
| | | $C_{-1}C_1$ | Previous and next characters |
| (c) | Character type | $T_{-1}T_0T_1$ | Types of previous, current and next character |

Table VII. Four Feature Template Sets

| ID of Feature template set | Description |
|------|------|
| TMPT-1 | Defined in Table VI. |
| TMPT-2 | Add $C_{-2}, C_2$ and remove $T_{-1}T_0T_1$ in TMPT-1 |
| TMPT-3 | Add $C_{-2}$ in TMPT-1 |
| TMPT-4 | Remove $T_{-1}T_0T_1$ in TMPT-1 |

$t$ reflects the position of current character. In particular, we employ the CRF++ package version 0.42 developed by Taku Kudo.[7]

Given the learning framework, we need to define the corresponding features. Similar to MaxEnt, our CRF-based learning framework regards Chinese word segmentation as a character-based tagging task. For details about feature generation, please refer to the work in Ratnaparkhi [1996]. Here we treat all features as functions derived from various feature templates and tag sets. So, our method is about two key issues: feature template settings and tag set selection.

## 3.1 Feature Template Settings

The probability model and corresponding feature function are defined over the set $H \times S$, where $H$ is the set of possible contexts (or any predefined condition) and $S$ is the set of possible tags. Generally, a feature function can be defined as

$$f(h, t) = \begin{cases} 1, & \text{if } h = h_i \text{ and } t = t_j \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

where $h_i \in H$ and $t_j \in S$.

For simplicity, features are generally organized into groups by called feature templates. For example, a unigram template $C_1$ stands for the next character occurring in the corpus after the current character $C_0$.

Table VI shows basic feature templates while Table VII lists four widely-used sets for experimental comparison.

Here we give an explanation to feature template (c) in Table VI. Feature template (c) is slightly improved from the counterparts in Low et al. [2005]. $T_n$, for $n = -1, 0, 1$, stands for some predefined class (type) of previous, current, or next character. There are five defined classes: numbers or characters that stand for numbers represent class 1, those characters whose meanings are date and time represent class 2, English letters represent class 3, punctuation

---

[7]http://chasen.org/ taku/software/CRF++/

Table VIII. Character Classes Employed in Feature Template (c)

| Class | Description | Character set |
|---|---|---|
| 1 | Digit | 0,1,2,3,4,5,6,7,8,9 |
| | | 一, 二, 三, 四, 五, 六, 七, 八, 九, 十, 百, 千, 万, 零, |
| | | ０,１,２,３,４,５,６,７,８,９ |
| 2 | Date/time | 年, 月, 日, 时, 分, 秒 |
| 3 | English letters | a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z |
| | | A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z |
| 4 | Punctuation | ，, 。! – – … |
| 5 | Other characters | 人, 丁, 可, 民, … |

Table IX. Tag Sets Employed in the State-of-the-Art Chinese Word Segmentation Systems

| Four-tag set Low/(Xue) | | Three-tag set Zhang | | Two-tag set Peng/Tseng | |
|---|---|---|---|---|---|
| Function | Tag | Function | Tag | Function | Tag |
| Begin | B(LL) | Begin | B | Start | Start |
| Middle | M(MM) | Middle | I | Continue | NoStart |
| End | E(RR) | or end | | | |
| Single | S(LR) | Single | O | | |

labels represent class 4 and other characters represent class 5. The character set for each class is shown in Table VIII.[8]

## 3.2 Tag Set Selection

Various sets of tags have been proposed in the literature, which are shown in Table IX. For example, Xue [2003] and Low et al. [2005] used the four-tag set in the MaxEnt model. Peng et al. [2004] and Tseng et al. [2005] used the two-tag set in the CRF model, and Zhang et al. [2006] used the three-tag set.

Though features are determined by both feature template and tag sets, in the literature, the tag set is often specified beforehand and paid much less attention than the feature template. However, we will show that tag set selection is as important as feature template set selection.

To better model longer words, we extend the four-tag set as adopted in Xue [2003] and Low et al. [2005]. In particular, the tag $B_2$ is added into the four-tag set to form a five-tag set, which stands for the second character position in a multi-character word only if it is not the last character in the word. Similarly, the tag $B_3$ is added into the five-tag set to form a six-tag set, which stands for the third character position in a multi-character word only if it is not the last

---

[8]As mentioned, only the training data are allowed for the particular task in the closed test of Bakeoff. The criteria to determine if the test is subjected to the closed test is how feature templates are defined in the learning framework. $n$-gram feature templates are often recognized as standard ones for the closed test without question. Thus most disputes occur in feature template (c) in Table VI. According to guidelines given by the organizer of bakeoff, using the information of character types was not allowed in the closed test. This was obeyed by Ng and Low [2004] and Tseng et al. [2005], while it was not followed in Li [2005], Lau and King [2005], Zhou [2005], and Zhu et al. [2006]. Tsai et al. [2006] developed a clustering algorithm to find different character types. For a comprehensive comparison, we will give both experimental results with and without feature template (c).

Table X.  Definitions of Different Tag Sets

| Tag set | Tags | Tag sequences for words of different lengths |
|---------|------|----------------------------------------------|
| 2-tag | $B, E$ | $B, BE, BEE, ...$ |
| 3-tag/a | $B, E, S$ | $S, BE, BEE, ...$ |
| 3-tag/b | $B, M, E$ | $B, BE, BME, BMME, ...$ |
| 4-tag | $B, M, E, S$ | $S, BE, BME, BMME, ...$ |
| 5-tag | $B, B_2, M, E, S$ | $S, BE, BB_2E, BB_2ME, BB_2MME, ...$ |
| 6-tag | $B, B_2, B_3, M, E, S$ | $S, BE, BB_2E, BB_2B_3E, BB_2B_3ME, ...$ |
| 7-tag | $B, B_2, B_3, B_4, M, E, S$ | $S, BE, BB_2E, BB_2B_3E, BB_2B_3B_4E, BB_2B_3B_4ME, ...$ |

Table XI. The Distribution of Words with Different Lengths in the Training Data of Different Corpora (Big5)

| wordLen | AS2003 | AS2005 | AS2006 | CityU2003 | CityU2005 | CityU2006 |
|---------|--------|--------|--------|-----------|-----------|-----------|
| 1 | 0.5447 | 0.5712 | 0.5494 | 0.4940 | 0.4689 | 0.4727 |
| 2 | 0.3938 | 0.3787 | 0.3900 | 0.4271 | 0.4554 | 0.4509 |
| 3 | 0.0463 | 0.0358 | 0.0463 | 0.0587 | 0.0597 | 0.0613 |
| 4 | 0.0107 | 0.0099 | 0.0105 | 0.0159 | 0.0134 | 0.0126 |
| 5 | 0.0018 | 0.0019 | 0.0017 | 0.0024 | 0.0016 | 0.0015 |
| $\leq 5$ | 0.9973 | 0.9974 | 0.9980 | 0.9981 | 0.9990 | 0.9990 |
| 6 | 0.0008 | 0.0007 | 0.0006 | 0.0010 | 0.0005 | 0.0004 |
| $\leq 7$ | 0.9987 | 0.9986 | 0.9990 | 0.9996 | 0.9998 | 0.9998 |

Table XII. The Distribution of Words with Different Lengths in the Training Data of Different Corpora (GB)

| wordLen | CTB2003 | CTB2006 | MSRA2005 | MSRA2006 | PKU2003 | PKU2005 |
|---------|---------|---------|----------|----------|---------|---------|
| 1 | 0.4367 | 0.4813 | 0.4715 | 0.4690 | 0.4721 | 0.4727 |
| 2 | 0.4719 | 0.4411 | 0.4387 | 0.4379 | 0.4508 | 0.4499 |
| 3 | 0.0672 | 0.0596 | 0.0475 | 0.0493 | 0.0495 | 0.0495 |
| 4 | 0.0116 | 0.0106 | 0.0242 | 0.0243 | 0.0204 | 0.0205 |
| 5 | 0.0076 | 0.0047 | 0.0089 | 0.0094 | 0.0057 | 0.0056 |
| $\leq 5$ | 0.9950 | 0.9973 | 0.9899 | 0.9899 | 0.9985 | 0.9983 |
| 6 | 0.0024 | 0.0014 | 0.0037 | 0.0038 | 0.0007 | 0.0007 |
| $\leq 7$ | 0.9984 | 0.9992 | 0.9962 | 0.9963 | 0.9997 | 0.9995 |

character. For systematic evaluation, Table X lists various tag sets explored in this study.

Given various possibilities, it will be interesting to select an effective tag set in the segmentation task. Since our task is to segment a sequence of characters into a sequence of words with various lengths (in characters), it is natural to consider the word length distribution in a corpus. Tables XI and XII show the distribution of words with different lengths in all the 12 training corpora of the three bakeoffs.

Two factors exist in determining the importance of words with different lengths, especially long ones. The first factor is the percentage of the corpus covered by the words no less than a certain length, $k$, in characters and can be calculated by

$$r_k = \frac{\sum_{i=k}^{K} iN_i}{\sum_{i=1}^{K} iN_i}, \tag{3}$$

Table XIII. Distribution of Average Weighted Word Length in the Training Data of Different Corpora (Big5)

| Word length | AS2003 | AS2005 | AS2006 | CityU2003 | CityU2005 | CityU2006 |
|---|---|---|---|---|---|---|
| Total | 1.546 | 1.509 | 1.536 | 1.613 | 1.628 | 1.624 |
| $\geq 2$ | 1.001 | 0.938 | 0.987 | 1.119 | 1.159 | 1.151 |
| $\geq 3$ | 0.214 | 0.181 | 0.207 | 0.265 | 0.248 | 0.249 |
| $\geq 4$ | 0.075 | 0.073 | 0.068 | 0.089 | 0.069 | 0.065 |
| $\geq 5$ | 0.032 | 0.033 | 0.026 | 0.025 | 0.015 | 0.015 |
| $\geq 6$ | 0.023 | 0.024 | 0.017 | 0.013 | 0.007 | 0.007 |
| $\geq 7$ | 0.018 | 0.020 | 0.014 | 0.007 | 0.004 | 0.005 |
| $\geq 8$ | 0.014 | 0.017 | 0.011 | 0.004 | 0.002 | 0.002 |

Table XIV. Distribution of Average Weighted Word Length in the Training Data of Different Corpora (GB)

| Word length | CTB2003 | CTB2006 | MSRA2005 | MSRA2006 | PKU2003 | PKU2005 |
|---|---|---|---|---|---|---|
| Total | 1.702 | 1.627 | 1.710 | 1.714 | 1.643 | 1.646 |
| $\geq 2$ | 1.265 | 1.146 | 1.240 | 1.245 | 1.171 | 1.173 |
| $\geq 3$ | 0.321 | 0.264 | 0.362 | 0.370 | 0.269 | 0.273 |
| $\geq 4$ | 0.120 | 0.085 | 0.219 | 0.222 | 0.121 | 0.124 |
| $\geq 5$ | 0.073 | 0.042 | 0.122 | 0.124 | 0.039 | 0.042 |
| $\geq 6$ | 0.035 | 0.019 | 0.078 | 0.077 | 0.011 | 0.014 |
| $\geq 7$ | 0.021 | 0.011 | 0.055 | 0.055 | 0.006 | 0.010 |
| $\geq 8$ | 0.013 | 0.007 | 0.037 | 0.037 | 0.003 | 0.007 |

where $N_i$ is the number of those words whose lengths are $i$, and $K$ is the largest length of a word in the corpus. The second factor is related to the average length of all words and can be calculated by

$$l_{avg} = \frac{\sum_{i=1}^{K} iN_i}{N}, \tag{4}$$

where $N = \sum_{i=1}^{K} N_i$ is the number of all words in the corpus.

By leveraging both factors, the average weighted word length is computed in this study as

$$L_k = l_{avg} r_k = \frac{\sum_{i=1}^{K} iN_i}{N} \left( \frac{\sum_{i=k}^{K} iN_i}{\sum_{i=1}^{K} iN_i} \right) = \frac{1}{N} \sum_{i=k}^{K} iN_i, \tag{5}$$

where $L_k$ is the average weighted word length for $i \geq k$. In particular, if $k = 1$, then $L_1 = l_{avg}$. By involving $l_{avg}$, we can fairly compare different segmented corpora using $L_k$.

Tables XIII and XIV are the distribution of average weighted word length of 12 training corpora in the three bakeoffs. It is observed that average word length of MSRA2005, MSRA2006,[9] and CTB2003 are the longest. As for CityU2005 and CityU2006, though they are not the shortest ones, their average weighted lengths are the shortest if we only consider the words longer than four characters.

Note that a six-tag set can label a five-character or shorter word without repeating its tags. For example,"给"(give) is tagged by "S", "和平" (peace) is

---

[9]For the training corpus, MSRA2006 is a subset of MSRA2005 and CTB2003 is a subset of CTB2006.

tagged by "$BE$", "天安门广场" (Tian'anmen Square) is tagged by "$BB_2B_3ME$," "航天飞机" (space shuttle) is tagged by "$BB_2B_3E$," and so on. The fact that all the characters in a word can be tagged by different tags allows precise character discrimination in different positions of a word.[10] Considering that five-character or shorter words have covered around 99% of any corpus with different segmentation standards (Tables XI and XII), employing the six-tag set is enough to capture discriminative information. In addition, the capability of labeling a five-character word without repeating tags means that the learner actually works within a five-character window context even only with unigram feature templates. According to values in Tables XIII and XIV, average word lengths of 12 corpora ranges from 1.5 to 1.7 characters. Therefore, a three-word window may effectively cover 4.5 to 5.1 characters. That is, learning from a five-character window is equally doing that both from previous and next two characters and previous and next words. Note that the six-tag set is the smallest one that can fully tag a five-character word without repeating its tags.

We employ $L_5$ as our empirical criteria to determine if the six-tag set should be taken. If $L_5$ is larger than the pre-defined threshold, then we adopt the six-tag set. Otherwise, we consider a tag set with five or less tags. We will see that the threshold can be empirically set to 0.02 from experimental results in those training corpora of Bakeoff-2003 and 2005.

We don't consider a tag set with more than six tags until now due to two reasons. The first reason is that word length statistics. Tables XI and XII show that more than 99% of words in all corpora are less than six characters. This low bound will be 99.50% if MSRA2005 and MSRA2006 are excluded. This will also explain why the six-tag set works well in most cases. The second reason is related to computational cost. CRF learning is quite sensitive to the number of tags and too many tags will cause a dramatic increase in computational cost. In spite of these issues, we will still consider some tag sets with more than six tags to explore possible performance improvement.

## 4. EXPERIMENTS WITH DIFFERENT FEATURE TEMPLATES AND TAG SETS

Twelve corpora are available from Bakeoff-2003, 2005, and 2006, and all of them are selected to perform the evaluation. Table I gives a summary of these corpora. All experimental results in the rest of this article will be evaluated by $F_1$-measure unless specified.

### 4.1 Experimental Results on Bakeoff 2003 and 2005

Tables XV and XVI compare different tag sets and feature template sets on CTB2006[11] (GB encoding) and CityU2003 (Big5 encoding), respectively. They show that TMPT-1 performs best when combined with the six-tag set. Though both TMPT-2 and TEMT-4 are simple $n$-gram template sets, it can be also observed that TMPT-2 or TMPT-4 yields substantial performance improvement

---

[10]For example, tags $B_2$ and $B_3$ in the six-tag set cannot be differentiated in the four-tag set, as both of these two tags are noted as $M$ in the latter.
[11]Note that CTB2003 is a subset of CTB2006 for training corpus.

Table XV. Experimental Results on CityU2003 with Different Feature Template Sets and Tag Sets

|         | TMPT-1 | TMPT-2 | TMPT-3 | TMPT-4 |
|---------|--------|--------|--------|--------|
| 2-tag   | 0.9361 | 0.9302 | 0.9340 | 0.9335 |
| 3-tag/a | 0.9475 | 0.9417 | 0.9455 | 0.9443 |
| 3-tag/b | 0.9458 | 0.9429 | 0.9458 | 0.9431 |
| 4-tag   | 0.9500 | 0.9450 | 0.9494 | 0.9471 |
| 5-tag   | 0.9509 | 0.9461 | **0.9495** | 0.9474 |
| 6-tag   | **0.9512** | **0.9462** | 0.9494 | **0.9475** |
| 7-tag   | **0.9512** | 0.9458 | 0.9494 | **0.9475** |

Table XVI. Experimental Results on CTB2006 with Different Feature Template Sets and Tag Sets

|         | TMPT-1 | TMPT-2 | TMPT-3 | TMPT-4 |
|---------|--------|--------|--------|--------|
| 2-tag   | 0.9131 | 0.9074 | 0.9116 | 0.9118 |
| 3-tag/a | 0.9295 | 0.9233 | 0.9277 | 0.9279 |
| 3-tag/b | 0.9260 | 0.9195 | 0.9239 | 0.9241 |
| 4-tag   | 0.9322 | 0.9247 | 0.9288 | 0.9307 |
| 5-tag   | 0.9338 | 0.9259 | **0.9312** | 0.9321 |
| 6-tag   | **0.9340** | **0.9261** | 0.9309 | **0.9322** |
| 7-tag   | 0.9336 | 0.9259 | 0.9307 | 0.9317 |

Table XVII. Relationship Between Tag Set Selection and Average Weighted Word Length

| Participant | CityU2003 | CityU2005 | CTB2003 | MSRA2005 | PKU2003 | PKU2005 |
|-------------|-----------|-----------|---------|----------|---------|---------|
| 6-tag       | 0.9512 | 0.9563 | 0.8753 | 0.9738 | 0.9555 | 0.9530 |
| 5-tag       | 0.9509 | 0.9565 | 0.8744 | 0.9724 | 0.9549 | 0.9528 |
| Difference  | 0.0003 | *-0.0002* | **0.0009** | **0.0014** | 0.0006 | 0.0002 |
| $L_5$       | 0.0252 | *0.0150* | **0.0732** | **0.1223** | 0.0390 | 0.0423 |

when combined with a larger tag set. Meanwhile, TMPT-1 loses its top performance place when combined with the two-tag set.

Table XVII demonstrates the relationship between tag set selection and average weighted word length using the feature template set TMPT-4 by default. Though the difference is slight, the CityU2005 corpus with the smallest $L_5$ value (the criterion to choose the six-tag set) among six corpora and gets the better performance at the five-tag set instead of the six-tag set, while MSRA2005 and CTB2003, the two corpora with the largest $L_5$, obtain the most performance increase from the five-tag to the six-tag set.

## 4.2  Experimental Results on Bakeoff 2006

In this section, we report experimental results on the corpora of Bakeoff-2006. Still, different combinations of tag sets and feature template sets are considered. Especially, we consider three effective feature template sets originally for a two-tag set [Peng et al. 2004; Tseng et al. 2005; Tsai et al. 2006], and refer them as T_Peng, T_Tseng and T_Tsai respectively in the following. As Tseng et al. [2005] and Tsai et al. [2006] used some additional features besides $n$-gram ones, the $n$-gram parts of their feature template sets are extracted to construct two feature template sets, T_TsengNgram and T_TsaiNgram, respectively. Those additional features that Tseng et al. [2005] used are listed in

Table XVIII. Additional Feature Templates Extracted from Self-Collected Lexicons

| Type | Feature | Function |
|---|---|---|
| Known Word | $C_{-1}C_0$ | The substring $C_{-1}C_0$ is in the known word vocabulary. |
| Single-character word | $C_{-1}, C_0, C_1$ | The previous, current or next character is in the single-character word list. |
| Prefix | $C_{-1}$ | The previous character is in the prefix character list. |
| Suffix | $C_0$ | The current character is in the suffix character list. |

Table XIX. Comparison of Different Feature Template Sets and Tag Sets Employed in the State-of-the-Art Systems on the Corpora of Bakeoff-2006

| Tag set | Feature template | | | F-measures on different corpora | | | |
|---|---|---|---|---|---|---|---|
| | $n$-gram | T_TsengAdd | PT3 | AS | CityU | CTB | MSRA |
| 2 | T_Peng | | | 0.940 | 0.955 | 0.909 | 0.940 |
| | | + | | 0.948 | 0.963 | 0.923 | 0.947 |
| | | | + | 0.946 | 0.958 | 0.910 | 0.942 |
| | | + | + | 0.951 | 0.966 | 0.924 | 0.948 |
| 2 | T_TsengNgram | | | 0.944 | 0.961 | 0.923 | 0.944 |
| | | + | | 0.946 | 0.963 | 0.923 | 0.948 |
| | | | + | 0.948 | 0.964 | 0.924 | 0.945 |
| | | + | + | 0.950 | 0.965 | 0.924 | 0.949 |
| 2 | T_TsaiNgram | | | 0.946 | 0.962 | 0.924 | 0.946 |
| | | + | | 0.948 | 0.964 | 0.925 | 0.949 |
| | | | + | 0.951 | 0.965 | 0.926 | 0.946 |
| | | + | + | 0.952 | 0.966 | 0.926 | 0.950 |
| 4 | Zhu*(SVM) | | | 0.932 | 0.955 | 0.914 | 0.939 |
| | +Postprocessing | | | 0.944 | 0.968 | 0.927 | 0.956 |
| | Wang*(MaxEnt) | | | / | / | / | 0.953 |
| | +LM | | | 0.953 | 0.970 | 0.930 | 0.963 |
| | TMPT-2 | | | 0.952 | 0.966 | 0.931 | 0.955 |
| | | | + | 0.957 | 0.969 | 0.933 | 0.955 |
| 6 | TMPT-4 | | | 0.954 | 0.969 | 0.932 | 0.961 |
| | | | + | 0.959 | 0.972 | 0.934 | 0.961 |
| Best results of Bakeoff2006 | | | | 0.958 | 0.972 | 0.933 | 0.963 |

Table XVIII and noted as T_TsengAdd. All feature templates in this set are related to self-collected lexicons extracted from the training corpus. Thus this set is also applied to strengthen T_Peng and T_TsaiNgram. For the four-tag set, the same $n$-gram set that was also adopted by Xue [2003] and Low et al. [2005] for MaxEnt learning, TMTP-2, is correspondingly used. For six-tag set, TMPT-4 is adopted. In addition, character type feature $T_{-1}T_0T_1$, noted as PT3, is also used as an extra feature for the two-tag set. Table XIX compares different combinations of feature template and tag sets employed in the state-of-the-art systems.

Some results using MaxEnt and SVM from Wang et al. [2006] and Zhu et al. [2006] are also given in Table XIX. An ensemble method was used in Wang et al. [2006], where MaxEnt and $n$-gram language model were integrated through a simple weighted method. In Zhu et al. [2006], SVM with the exact same feature template set as Ng and Low [2004] was adopted. In addition, some postprocessing rules were applied to further correct those incorrect segmentations by the SVM segmenter.

Table XX.  Significance Test Between Different Tag Sets on the Corpora of Bakeoff-2006

| Tag set | | Significant degree | | | |
|---|---|---|---|---|---|
| From | To | AS2006 | CityU2006 | CTB2006 | MSRA2006 |
| 2 | 4 | *** | ** | *** | *** |
| 4 | 6 | ** | ** | * | *** |
| 2 | 6 | *** | *** | *** | *** |

Table XXI.  Comparison of Feature Numbers on the Corpora of Bakeoff-2006

| Tag set | Feature template | Number of features in different corpora ($\times 10^6$) | | | |
|---|---|---|---|---|---|
| | | AS2006 | CityU2006 | CTB2006 | MSRA2006 |
| 2 | T_Peng | 12.5 | 6.1 | 2.4 | 4.5 |
| | T_TsengNgram | 8.6 | 4.8 | 2.1 | 3.6 |
| | T_TsaiNgram | 13.2 | 7.3 | 3.1 | 5.5 |
| 4 | TMPT-2 | 16.1 | 9.0 | 3.9 | 6.8 |
| 6 | TMPT-4 | 15.6 | 8.8 | 3.8 | 6.6 |

Table XIX suggests a general trend that the more tags in tag set, the better performance we obtain. It also shows that Tseng's additional feature templates work well for all $n$-gram feature template sets with a two-tag set. This is due to that this set of feature templates were designed to compensate the deficiency of the two-tag set.

To see whether an improvement is significant, we also conduct significance tests using paired $t$-test. In this article, ***, **, and * denote $p$-values of an improvement less than 0.01, in-between (0.01, 0,05] and greater than 0.05, which mean significantly better, moderately better, and slightly better, respectively. The significance tests are performed between the results of four-tag and two-tag sets and between those of six-tag and two/four-tag sets, as shown in Table XX. Three feature template sets, T_TsaiNgram+T_TsengAdd, TMTP-2 and TMPT-4, are applied to two, four, and six-tag sets, respectively. Table XX shows that most performance improvements are significant as the tag set is continuously enlarged.

Notice that most researchers who adopted CRF as their learning model used the two-tag set, though some subsequent work has shown that four or six tags were more effective. This is largely due to computational cost.

For example, CRF training using L-BFGS algorithm often requires hundreds or thousands of iterations [Malouf 2002], each of which involves calculating the log-likelihood and its derivative. Cohn et al. [2005] shows that the time complexity of a single iteration is $O(n_l^2)$, where $n_l$ is the number of labels (tags) and it is still an issue to state the precise bound on the number of iterations. Moreover, efficient CRF implementations normally cache the feature values for every possible clique labeling for the training data. This leads to a space requirement of $O(n_l^2)$, too.

The preceeding theoretical analysis means that a CRF learner with a six-tag set will cost nine times as much memory or time as that with a two-tag set for the same task. This is not acceptable in most practical applications. Please refer to Tables XXI, XXII, and XXIII for detailed comparison in practice.

These tables show that training cost does increase as the tag set is enlarged. However, the actual increase is not so much as the prediction by the theoretical

Table XXII.  Comparison of Memory Cost on the Corpora of Bakeoff-2006

| Tag set | Feature template | Memory cost in different corpora (Mega bytes) | | | |
|---|---|---|---|---|---|
| | | AS2006 | CityU2006 | CTB2006 | MSRA2006 |
| 2 | T_Peng | 5,220 | 2,101 | 790 | 1,637 |
| | T_TsengNgram | 4,490 | 1,857 | 723 | 1,464 |
| | T_TsaiNgram | 5,428 | 2,362 | 933 | 1,847 |
| 4 | TMPT-2 | 6,590 | 2,787 | 1,084 | 2,180 |
| 6 | TMPT-4 | 6,379 | 2,662 | 984 | 2,090 |

Table XXIII.  Comparison of Training Time on the Corpora of Bakeoff-2006

| Tag set | Feature template | Training time in different corpora (Minutes) | | | |
|---|---|---|---|---|---|
| | | AS2006 | CityU2006 | CTB2006 | MSRA2006 |
| 2 | T_Peng | 110 | 42 | 12 | 32 |
| | T_TsengNgram | 98 | 33 | 10 | 32 |
| | T_TsaiNgram | 112 | 52 | 16 | 35 |
| 4 | TMPT-2 | 206 | 79 | 28 | 73 |
| 6 | TMPT-4 | 402 | 146 | 47 | 117 |

analysis. The experimental results show that the six-tag set costs nearly twice as much time as the four-tag set and about three times as the two-tag set. Fortunately, the memory cost with the six $n$-gram feature templates, TMTP-4, remains very close to that of the two- and four-tag sets with the $n$-gram feature template sets [Peng et al. 2004; Tseng et al. 2005; Tsai et al. 2006; Xue 2003]. This may be attributed to two factors. The first is the imbalance of tag distribution. We take the training corpus of CityU2006 with the six-tag set as an example. Tag $S$ covers 28.35% of the corpus, tag $B$ covers 32.36%, tag $E$ covers 32.36%, while other three tags, $B_2$, $B_3$, and $M$ as a whole only cover 6.93%. The second factor is that less feature templates prefer the larger tag set and vice versa. For example, TMPT-4 for the six-tag set includes six templates, while TMPT-2 for the four-tag set includes ten templates.

One reason that we finally choose the six-tag set as our standard tag set is that larger tag set can at most slightly improve the performance. Another reason is that the more tags mean the more training time and memory the system needs. Thus, the six-tag set is a good trade-off.

## 4.3  More Than Six Tags

This section evaluates the effectiveness of more than six tags in more detail. Only an even number of tags is taken into account to ensure the learning conducted in a sliding window that is symmetrical to the current character. For example, given unigram feature templates, the six-tag set ranges from previous two to next two characters, the eight-tag set ranges from previous three to next three characters, and the 14-tag set ranges from previous six to next six characters, and so on.

Considering how the six-tag set is extended from the four-tag set, we may continuously extend the six-tag set to the eight-tag set in the same way by including two more tags $B_4$ and $B_5$ to represent the fourth and fifth positions of characters in a multi-character word. Similarly two more tags $B_6$ and $B_7$

Table XXIV.  Comparison of 6-tag Set with Larger Ones

| Corpus | Tag Set | | | | |
|---|---|---|---|---|---|
| | 6 | 8 | 10 | 12 | 14 |
| AS2006 | 0.9538 | **0.9541** | 0.9540 | 0.9537 | 0.9539 |
| CityU2006 | **0.9691** | 0.9688 | 0.9688 | 0.9688 | 0.9686 |
| CTB2006 | 0.9320 | **0.9322** | **0.9322** | 0.9320 | **0.9322** |
| MSRA2006 | 0.9608 | **0.9617** | 0.9611 | 0.9611 | 0.9612 |

Table XXV. Performance Comparison of Different Feature Template Sets on the Corpora of Bakeoff-2006

| Feature Template | F-measure | | | |
|---|---|---|---|---|
| | AS2006 | CityU2006 | CTB2006 | MSRA2006 |
| TMPT-4 | 0.9538 | 0.9691 | 0.9320 | 0.9608 |
| TMPT-1 | 0.9586 | 0.9716 | 0.9339 | 0.9613 |

are added into the eight-tag set to form the ten-tag set, and so does for larger tag set.

Table XXIV compares different tag sets with the feature template set TMPT-4 by default, on the four corpora of Bakeoff-2006.

It should be noted that although the eight-tag set reaches its peak performance on three corpora, it only slightly outperforms the six-tag set. Thus, it is hard to recognize the performance differences as statistically significant. In addition, we cannot observe an obvious trend of performance increase when the tag set is continuously enlarged according to the results in Table XXIV, though much more time-consuming and memory-consuming learning with larger than the six-tag sets are expected. However, an obvious performance increase with the eight-tag set compared to the six-tag set can be found in MSRA2006 corpus. Note that $L_7$ value of MSRA2006 $(0.0546)$[12] is much more than that of CityU2006 (0.0045) or CTB2006 (0.0108). This explains why an obvious peak performance appears with the eight-tag set in MSRA2006 corpus. Even though the eight-tag set can achieve better performance than the six-tag set in most cases, considering that learning with it will cost twice as much time than that with the six-tag set, we still prefer the six-tag set in this study.

### 4.4 Experimental Results with Different FeatureTemplate Sets

To evaluate the contribution of different feature templates with or without character type feature, we perform this group of experiments with two template sets, TMPT-1 and TMPT-4 (Table XXV). It is observed that as demonstrated by Low et al. [2005] the character type feature is helpful though $n$-gram features still contribute most.

### 5. UNIFYING ASSISTANT SEGMENTERS INTO THE LEARNING FRAMEWORK

Since learning only from a training corpus is not always enough, it is sometimes beneficial for Chinese word segmentation to use additional linguistic

---

[12]$L_7$ is related to the eight-tag set selection since the eight-tag set is able to fully tag a seven-character word according to the discussion in Section 3.2.

Table XXVI.  Performance Comparison of Cross-Test

| Test | Training corpus | | | |
|---|---|---|---|---|
| corpus | AS2006 | CTB2006 | CityU2006 | MSRA2006 |
| AS2006 | 0.9538 | 0.8941 | 0.8971 | 0.8246 |
| CTB2006 | 0.8985 | 0.9320 | 0.8823 | 0.8430 |
| CityU2006 | 0.8891 | 0.8711 | 0.9691 | 0.8156 |
| MSRA2006 | 0.8174 | 0.8264 | 0.8221 | 0.9608 |

Table XXVII.  Relative Agreeable Rate Among Different Segmentation Standards

| Test | Training corpus | | | |
|---|---|---|---|---|
| corpus | AS2006 | CTB2006 | CityU2006 | MSRA2006 |
| AS2006 | 1.0000 | 0.9593 | 0.9256 | 0.8583 |
| CTB2006 | 0.9420 | 1.0000 | 0.9104 | 0.8774 |
| CityU2006 | 0.9321 | 0.9346 | 1.0000 | 0.8488 |
| MSRA2006 | 0.8570 | 0.8866 | 0.8483 | 1.0000 |

resources. Assistant segmenter is a feature method that represents additional linguistic knowledge. Here, two types of additional feature templates are adopted to improve the performance further.

## 5.1 Assistant Segmenter

Low et al. [2005] observed that though different segmentation standards are presented, segmentation differences only exist in a few words. In fact, most word segmenters trained on different corpora agree in most cases. To verify this observation, we demonstrate some results on cross-test among different segmentation standards on the four corpora of Bakeoff-2006.

Our method is straightforward in that the segmenter is trained on a training corpus and the test is performed on the corresponding test corpus and the other three test corpora as well. All of the systems employ the same feature template set TMPT-4 and the six-tag set. Tables XXVI and XXVII show that as expected, different standards agree in most of segmentation cases.

Tables XXVI and XXVII display a consistent rate of more than 84% among four segmentation standards of Bakeoff-2006. Note that the least rate of consistency occurs between MSRA2006 and other three corpora. This means that the MSRA segmentation standard adopts quite different guideline from the other standards. Though it is not directly comparable due to different evaluation circumstances, we are still reminded that the rate of agreement among human judgment was only 76% on average as reported in Sproat et al. [1996].

The consistency among different standards makes it feasible to customize a pre-defined standard into any other standards as reported by Gao et al. [2005]. And it also motivates us to incorporate different segmenters into one segmenter on the current standard. For convenience, we call the segmenter subjected to the current standard the main segmenter, and the others assistant segmenters.

A feature template will be added for an assistant segmenter:

$$t(C_0)$$

where $t(C_0)$ is the output tag of the assistant segmenter for the current character $C_0$. For example, consider character sequence, 他来自北京 (He comes from Beijing), an assistant segmenter gives the tag sequence "*SBEBE*" according to its output segmentation, then $t(C_0)$ by this assistant segmenter is *S*, *B*, *E*, *B*, and *E* for each current character, respectively.

Indeed, our study shows that the more assistant segmenters are used, the better performance we can achieve. However, not all assistant segmenters are helpful for segmentation tasks in some special cases. In addition to cross-validation in training corpus that could be a general method to select the useful assistant segmenter, we empirically adopt a principle that only those assistant segmenters that are different from main segmenter in either training corpora or features should be chosen. In detail, if assistant and main segmenters are trained with the same types of features, then the training corpus of the assistant segmenter should be quite different from that of the main segmenter. The word "quite" means that two training corpora should not overlap too much. For example, the former should be neither the same as the latter nor a superset of the latter. If two training corpora overlap, then the features used by the assistant segmenter and the main segmenter should be somewhat different, too. In one word, the assistant segmenter should be somewhat different from the main segmenter in prior knowledge about the training set. From the view of machine learning, the assistant segmenter approach here is to equally take CRF itself as an ensemble learning strategy to integrate selected assistant segmenters into the main segmenter. Thus, learning from the same corpus with the same way cannot bring more useful information for the ensemble goal. This issue has been summarized as a diversity requirement for learning component in ensemble learning community [Kuncheva and Whitaker 2003]. Therefore, our principle may be viewed as a special case deduced from this general integration principle in machine learning. We will find that the principle of using an assistant segmenter is useful in alleviating a great deal of computational cost by cross-validation.

The proposed assistant segmenter method is more convenient and tractable compared to the additional training corpus method [Low et al. 2005]. First, the assistant segmenter is a parameter-free approach, that is, no parameters are defined or required by our approach, while the additional corpus method is not. Second, the additional corpus method is only able to extract material from external corpus, but fails to take advantage of a well-trained segmenter if the external corpus cannot be accessed at all. Third, the assistant segmenter is more computationally efficient than the additional corpus method, especially for CRF learning. The reason is that the increase of training corpus size leads to a simultaneous increase in training cost. Meanwhile, assistant segmenters only slightly increase the training cost.

## 5.2 External Dictionary

The external dictionary method for character-based word segmentation was first introduced by Low et al. [2005]. We continue to adopt this technique in this study.

Assuming that a subsequence includes $C_0$ in the sentence, then the longest word $W$ in the dictionary that matches such a subsequence will be chosen. The following features derived from the dictionary are added:

$Lt_0$
$C_n t_0 (n = -1, 0, 1),$

where $t_0$ is the boundary tag of $C_0$ in $W$, and $L$ is the number of characters in $W$, namely, the length of $W$. Our empirical study shows that $t_0$ is more effective than $t_{-1}$ or $t_1$; that is why this tag is adopted.

In this study, we apply the same online dictionary from Peking University as employed in Low et al. [2005], which contains approximately 108,000 words of one to four characters in length.[13]

It is interesting that we may also regard the external dictionary method as a variant of assistant segmenter to some degree. An example is a maximal matching segmenter with the specified external dictionary (we will verify this assertion through experiments). Thus, all of our additional techniques for integrating additional linguistic information can be viewed as assistant segmenter ones.

## 5.3 Assistant Segmenter Feature as Standard Adaptation

Though most previous word segmenters were developed on some standard that assumes a single correct segmentation, there is still some work that attempts a segmentation standard adaptation, or development of customizable segmenters to make full use of existing work. The first adaptation method was introduced by Gao et al. [2004], which can be viewed as an improved version of that in Brill [1995], where the adaptation rules (or transformations) are acquired automatically from application data via the transformation-based learning (TBL) method. Though the use of TBL for Chinese word segmentation is not new [Palmer 1997; Hockenmaier and Brew 1998], none of them aim at standard adaptation, but error-driven word segmentation instead [Gao et al. 2004].

TBL method is an error-driven technique for corpus tagging task. Assuming that an initial annotator, some predefined transformation rules and an annotated corpus are available, TBL runs iteratively to find the best rule that causes the most error reduction in the whole corpus to determine the order of using rules until no rules can reduce errors. As for TBL-based standard adaptation, it requires an initial segmentation, a goal segmentation, and a space of allowable transformations. Under the adaptation paradigm proposed by Gao et al. [2004; 2005], the initial segmentation is the output of the generic segmenter that holds the original segmentation standard. The goal segmentation is represented by adaptation data. The transformation templates can adopt either words or some predefined types.

---

[13]The dictionary can be downloaded from http://ccl.pku.edu.cn/doubtfire/Course/ Chinese%20Information%20Processing/Source_Code/Chapter_8/Lexicon_full_2000.zip.

Some problems can be handled well using simple heuristic rules for special cases by TBL. However, defining the rules for special cases can be time-consuming, difficult, and prone to errors and omissions. Normally a different rule set is needed for each domain. TBL-based standard adaptation works well in most cases as reported in Gao et al. [2004; 2005]. However, it still has an obvious drawback in that its training is very expensive, that is, too much memory and time is required. In addition, we also expect more performance enhancement by considering improved technique.

Using the same technique as the assistant segmenter method, a novel standard adaptation may be conducted. Our proposed standard adaptation method works like the following. Suppose a segmenter $S$ follows the original segmentation standard $A$, and we wish to adapt the segmenter $S$ to segmentation standard $B$. In this case, we train the CRF segmenter with the following feature templates: (1) all CRF feature templates for the segmentation standard $B$ using a training data set segmented according to standard $B$, and (2) a feature template $t(C_0)$, where the tag $t(C_0)$ is output by segmenter $S$ based on segmentation standard $A$. In this novel adaptation framework, we just reappraise each feature template in our system with assistant segmenter features. Now, assistant segmenter based feature will become the main feature in training, and others are the assistant (transformation) ones as adaptation rules. The adaptation procedure will still be the training of CRF. Nothing is different in the segmentation system besides our viewpoint of it.

Of course, some assistant transformation rules (additional features) can still be used for performance enhancement of standard adaptation. Segmentation standard adaptation with assistant segmenter also allows us to make a customized tradeoff between training cost and adaptation performance as the TBL method did. For example, if we want to get the better adaptation performance, then we may put all basic feature templates defined in TMPT-1. Or, we only take some unigram feature templates for faster training.

To verify the effectiveness of the proposed approach, we perform some comparison experiments. In particular, we adopt the MSR standard as the original segmentation standard as described in MSRSeg of Gao et al. [2005].[14] This standard was also developed by Microsoft Research Asia in Beijing. As the builder of these segmented corpora, we recognize that MSRA standard, represented by MSRA2005 or MSRA2006 corpora of bakeoff, is related to MSR standard as they share some guidelines. Thus, the difference between them is actually slight according to our evaluation. However, this fact does not imply that the assistant segmenter as standard adaptation is constrained by the similarity or difference of standards. We will show that our approach is capable of adopting any existing standard.

Different additional feature template sets are used as adaptation rules in CTB2006 and MSRA2006 corpora to demonstrate the adaptation performance. Table XXVIII shows the experimental results, where unigram stands for the

---

[14]A public version of MSRSeg, S-MSRSeg, is available from from Microsoft's Web site http://research.microsoft.com/research/downloads/Details/47c06c94-e9c7-414c-9e22-c2ef49100d1e/Details.aspx.

Table XXVIII. Contributions of Different Additional Features for Segmentation Standards Adaptation from MSR Standard

| Corpus | Additional feature templates | | | |
|---|---|---|---|---|
| | None | Unigram | TMPT-4 | TMPT-1 |
| CTB2006 | 0.8335 | 0.9194 | 0.9407 | 0.9426 |
| MSRA2006 | 0.9700 | 0.9744 | 0.9807 | 0.9814 |

Table XXIX. Segmentation Standards Adaptation from MSR Standard: Ours vs. Gao et al. [2005]

| Participants | Corpora | | | |
|---|---|---|---|---|
| | AS2003 | CityU2003 | CTB2003 | PKU2003 |
| Gao et al. [2005] | 0.958 | 0.954 | 0.904 | 0.955 |
| Ours/w TMPT-4 | 0.9702 | 0.9556 | 0.9051 | 0.9597 |
| Ours/w TMPT-1 | 0.9701 | 0.9582 | 0.9063 | 0.9600 |

three unigram feature templates defined in Table VI. Still from a view of the assistant segmenter, the results in the leftmost column with title "None" in Table XXVIII are given by direct segmentation of MSRSeg, or by a CRF segmenter only with features as outputs of MSRSeg assistant segmenter. Both of them output the same segmentation for the same input sentence. The results in the other three columns are given by the CRF segmenter with corresponding adaptation feature template set incorporated with MSRSeg assistant segmenter. We observe that the more adaptation features are used, the higher performance we can obtain.

To give a comparison with TBL-based adaptation [Gao et al. 2005], we also perform a group of experiments in four corpora of Bakeoff-2003. Without considering sophisticated handcrafted transformation rules as employed in existing work, we simply use two feature sets in training. Nevertheless, Table XXIX shows that our approach is superior to the existing TBL method.

## 5.4 Assistant Named Entity Recognizer

The idea to integrate named entity (NE) information into a segmenter is straightforward from the assistant segmenter framework. Intuitively, a sequence of words will be more convincingly segmented if it is also recognized as NE. Actually, NE is almost always segmented as a word in almost all segmentation standards.

A feature template will be added for an assistant NE recognizer:

$$t_{NE}(C_0),$$

where $t_{NE}(C_0)$ is the output tag of the assistant NE recognizer for the current character $C_0$. For example, consider a character sequence, 他来自北京 (He comes from Beijing), an assistant NE recognizer identifies 北京 (Beijing) as "LOCATION" NE according to its output, then $t_{NE}(C_0)$ will be "No", "No", "No", "*Location − Start*" and "*Location − End*" for each current character, respectively.

In this study, we use MSRSeg as our NE recognizer since it also outputs NE information. Besides person name, location name, and organization name,

Table XXX.　A List of Assistant Segmenters with IDs

| ID | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| Assistant segmenter | MSRSeg | MSRSegNE | MSRA2005 | PKU2003 | PKU2005 | CTB2006 |
| ID | G | H | I | J | K | L |
| Assistant segmenter | AS2003 | AS2005 | CityU2003 | CityU2005 | CityU2006 | AS2006 |

Table XXXI.　Contribution of Assistant Segmenters on the CTB2006 Corpus

| Segmenter | TMPT-1+Ext.Dict. | +C | +D+E | +G+H | +I+J+K | +A | +B(Final) |
|---|---|---|---|---|---|---|---|
| $F$-measure | 0.9423 | 0.9468 | 0.9475 | 0.9515 | 0.9518 | 0.9522 | 0.9531 |

Table XXXII.　Contribution of Assistant Segmenters on the MSRA2006 Corpus

| Segmenter | TMPT-1+Ext.Dict. | +E+G+H+K | +A | +B(Final) | +C |
|---|---|---|---|---|---|
| $F$-measure | 0.9694 | 0.9704 | 0.9823 | 0.9826 | 0.9702 |

NE outputs of MSRSeg also include ten types of factoid words such as date, duration, money, time, integer, e-mail, phone and so on. For details, please refer to Gao et al. [2005].

## 5.5 Evaluation

This group of experiments is to explore what will happen if we use many assistant segmenters in a task to integrate various linguistic resources. All the evaluations are done on the four corpora of Bakeoff-2006. With the help of the cross-validation method in training corpus and the proposed principle to select assistant segmenters, we integrate as many of the other segmenters as possible that are trained on all corpora from Bakeoff-2003, 2005, and 2006 with feature template set TMPT-4. The word segmenter and NE recognizer, MSRSeg, described in Gao et al. [2005], is also integrated.

Table XXX shows the IDs of those concerned assistant segmenters, where ID "MSRSeg" means that MSRSeg only works as word segmenter, ID "MSRSegNE" means that MSRSeg only works as NE recognizer, and ID "MSRA2005" stands for an assistant segmenter that is trained on MSRA2005 training corpus with feature template set TMPT-4, and so on. Experimental results with incremental assistant segmenter combination for CTB2006 and MSRA2006 are shown in Tables XXXI and XXXII, respectively. Note that assistant segmenter features are incrementally integrated into the system for performance enhancement from left to right in Table XXXI or XXXII. For example, the result of column with the title "+C" in Table XXXI indicates a feature template set of TMPT-1+Ext.Dict.+C, and the result of next column "+D+E" indicates a feature template set of TMPT-1+Ext.Dict.+C+D+E, and so on. The order to add assistant segmenters is simply according to alphabetical order of IDs except for MSRSeg and MSRSegNE. Our empirical results show that the final performance does not depend on such an order and the final selected assistant segmenter set is always the same.

Table XXXIII. Contribution of Different Types of Open Feature Templates on the Four
Corpora of Bakeoff-2006

| Corpora | Feature templates | | | | |
|---|---|---|---|---|---|
| | Basic Features | +External dictionary | +Assistant NE recognizer | +All assistant segmenters | All Features |
| AS2006 | 0.9586 | 0.9587 | 0.9591 | 0.9589 | 0.9607 |
| CityU2006 | 0.9716 | 0.9749 | 0.9724 | 0.9766 | 0.9778 |
| CTB2006 | 0.9339 | 0.9423 | 0.9360 | 0.9504 | 0.9531 |
| MSRA2006 | 0.9613 | 0.9694 | 0.9649 | 0.9812 | 0.9826 |

Our experimental results do show that the more assistant segmenters are used, the better performance we can achieve. However, according to the principle as mentioned in Section 5.1 that assistant segmenter should differ from the main segmenter to some degree, though we used all corpora of three Bake-offs to train assistant segmenters for tasks of Bakeoff-2006. Two assistant segmenters, AS2003 and AS2005, are not selected for AS2006 task, and the assistant segmenter MSRA2005 is not selected for MSRA2006. We also avoid using CTB2003 and MSRA2006 assistant segmenters in all tasks because the respective training corpus is a subset of that of CTB2006 and MSRA2005, respectively.

We may show one of the consequences when the principle of using assistant segmenter is violated. As we know the training corpus of MSRA2005 is a superset of that of MSRA2006, training for the main segmenter of MSRA2006 will be quickly prone to overfitting by the outputs of MSRA2005 assistant segmenter, and all effective other features are ignored by main segmenter through learning. Thus a dramatic decrease in performance occurs if we insist on using this assistant segmenter as shown in Table XXXII. In fact, such a low result is just what we can obtain when we train the segmenter in training corpus of MSRA2005 and test it in test corpus of MSRA2006 with TMPT-4 feature template set.[15]

Table XXXIII compares the contributions of different types of open feature templates. In Table XXXIII, basic features stand for feature template set TMPT-1, and MSRSegNE is still used as assistant NE recognizer. All assistant segmenters stand for all possible ones for the respective task except for those that are prohibited by the proposed principle of selecting assistant segmenter. Note that they may not be the same set for four tasks. Three columns whose titles contain "+" means that their feature template sets are the union set of TMPT-1 and the corresponding feature template set as titled, respectively. The title "All features" in the rightmost column means that all feature template sets in previous four columns should be used. Table XXXIII shows that assistant segmenter approach is robust for all kinds of linguistic resources. This verifies the effectiveness in integrating various linguistic information into a system.

---

[15]Both MaxEnt and CRF learning are capable of accommodating overlapped features. However, the learner will tend to overfit if a feature dominates during training. This is an explanation from machine learning theory.

Table XXXIV. Comparisons of Our Results with Best Reported Results in Closed Test
of Bakeoff-2003

| Participants | F-measures in different corpora | | | |
|---|---|---|---|---|
| | AS2003 | CityU2003 | CTB2003 | PKU2003 |
| Zhang and Liu [2003] | 0.938 | 0.901 | 0.881 | 0.951 |
| Peng et al. [2004] | 0.956 | 0.928 | 0.849 | 0.941 |
| Tseng et al. [2005] | 0.970 | 0.947 | 0.863 | 0.953 |
| Best results of Bakeoff-2003 | 0.961 | 0.940 | 0.881 | 0.951 |
| Goh et al. [2005] | 0.959 | 0.937 | 0.847 | 0.947 |
| Liang [2005]/w standard CRF | / | 0.937 | 0.879 | 0.941 |
| Liang [2005]/w Semi-CRF | / | 0.936 | 0.868 | 0.936 |
| Ours/TMPT-4 | 0.9727 | 0.9473 | 0.8720 | 0.9558 |
| Ours/TMPT-1 | 0.9737 | 0.9513 | 0.8753 | 0.9555 |

Table XXXV. Statistical Significance of the Results in Closed Test of Bakeoff-2003

| Corpus | #word | Participant | $R$ | $C_r$ | $P$ | $C_p$ | F-measure |
|---|---|---|---|---|---|---|---|
| AS2003 | 12K | Others | 0.966 | ±0.0033 | 0.956 | ±0.0037 | 0.961 |
| | | Ours | 0.9724 | ±0.00299 | 0.9732 | ±0.00295 | **0.9727** |
| CityU2003 | 35K | Others | 0.947 | ±0.0024 | 0.934 | ±0.0027 | 0.940 |
| | | Ours | 0.9472 | ±0.00239 | 0.9474 | ±0.00239 | **0.9473** |
| CTB2003 | 40K | Others | 0.886 | ±0.0032 | 0.875 | ±0.0033 | 0.881 |
| | | Ours | 0.8692 | ±0.00337 | 0.8746 | ±0.00332 | 0.8720 |
| PKU2003 | 17K | Others | 0.962 | ±0.0029 | 0.940 | ±0.0036 | 0.951 |
| | | Ours | 0.9548 | ±0.00319 | 0.9560 | ±0.00315 | **0.9558** |

# 6. PERFORMANCE COMPARISON WITH EXISTING WORK

## 6.1 Experimental Results in Corpora of Bakeoff-2003 and 2005

The comparison between our results and the best-reported results are shown
in Tables XXXIV through to XXXIX.[16] There are two types of reported results
for each corpus. One is the best official result in Bakeoff-2003 and 2005. The
other is presented by individual articles (i.e., post-evaluation) [Zhang and Liu
2003; Peng et al. 2004; Tseng et al. 2005; Low et al. 2005]. All of our results
are obtained with a six-tag set.

To check if performance improvement is statistically significant, we perform
some statistical significance tests in the results of closed test. Following pre-
vious work [Sproat and Emerson 2003] and assuming the binomial distrib-
ution, we may compute 95% confidence interval as $\pm 2\sqrt{p(1-p)/n}$ according
to the Central Limit Theorem for Bernoulli trials [Grinstead and Snell 1997],
where $n$ is the number of trials (words). We suppose that the recall represents
the probability of correct word identification, and the precision represents the
probability that a character string that has been identified as a word is really
a word. Thus two types of intervals, $C_r$ and $C_p$, can be computed, respectively.
One can determine if two results are significantly different at a 95% confidence
level by checking whether their confidence intervals overlap. The values of $C_r$
and $C_p$ for experimental results in Bakeoff-2003 and 2005 are given in Tables
XXXV and XXXVIII. Since many results given by individual literatures do

---

[16]We will cite the existing results not only from each Bakeoff but also from those post-evaluation
ones.

Table XXXVI. Comparisons of Best Existing Results and Ours in Open Test of Bakeoff-2003

| Participants | *F*-measures in different corpora | | | |
|---|---|---|---|---|
| | AS2003 | CityU2003 | CTB2003 | PKU2003 |
| Best results of Bakeoff-2003 | 0.904 | 0.956 | 0.912 | 0.959 |
| Ours/TMPT-1 +Ext. dict. | **0.9732** | **0.9644** | 0.9028 | **0.9643** |
| Ours/TMPT-1 +Ext. dict. +MSRSeg | **0.9729** | **0.9660** | 0.9113 | **0.9673** |

Table XXXVII. Comparisons of Best Existing Results and Ours in Closed Test of Bakeoff-2005

| Participants | *F*-measures in different corpora | | | |
|---|---|---|---|---|
| | AS2005 | CityU2005 | MSRA2005 | PKU2005 |
| Low et al. [2005] | 0.953 | 0.950 | 0.960 | 0.948 |
| Tseng et al. [2005] | 0.947 | 0.943 | 0.964 | 0.950 |
| Best results of Bakeoff-2005 | 0.952 | 0.943 | 0.964 | 0.95 |
| Zhang et al. [2006]/Sub-word | 0.936 | 0.931 | 0.954 | 0.936 |
| Zhang et al. [2006]/Sub-word+Dict. | 0.951 | 0.951 | 0.971 | 0.951 |
| Ours/TMPT-4 | 0.9534 | 0.9476 | **0.9735** | 0.9515 |
| Ours/TMPT-1 | **0.9567** | **0.9563** | **0.9739** | **0.9530** |

Table XXXVIII. Statistical Significance of the Results in the Closed Test of Bakeoff-2005

| Corpus | #word | Participant | $R$ | $C_r$ | $P$ | $C_p$ | $F$ |
|---|---|---|---|---|---|---|---|
| AS2005 | 122K | Others | 0.952 | ±0.00122 | 0.951 | ±0.00123 | 0.952 |
| | | Ours | 0.9581 | ±0.00115 | 0.9471 | ±0.00128 | **0.9534** |
| CityU2005 | 41K | Others | 0.941 | ±0.00233 | 0.946 | ±0.00223 | 0.943 |
| | | Ours | 0.9468 | ±0.00222 | 0.9485 | ±0.00218 | **0.9476** |
| MSRA2005 | 107K | Others | 0.962 | ±0.00117 | 0.966 | ±0.00111 | 0.964 |
| | | Ours | 0.9718 | ±0.00101 | 0.9746 | ±0.00096 | **0.9735** |
| PKU2005 | 104K | Others | 0.953 | ±0.00131 | 0.946 | ±0.00140 | 0.95 |
| | | Ours | 0.9463 | ±0.00140 | 0.9568 | ±0.00126 | **0.9515** |

not give the values of $R$ and $P$, comparison in these two tables is performed between our results with TMPT-4 and the best official results of Bakeoff-2003 and 2005 [Sproat and Emerson 2003; Emerson 2005].

We find that the results of our system are much better than the best results of Bakeoff-2003 except for that in CTB2003 corpus. As we perform the experiments with the same system from Bakeoff-2003 to Bakeoff-2005, we may regard this as a consistent technical progress in Chinese word segmentation since 2003.

Some simplified results with assistant segmenter approach in open test are also demonstrated. Only one assistant segmenter is assigned for each task since our system has almost outperformed all the existing ones only with external dictionary technique. MSRSeg assistant segmenter is used for all tasks of Bakeoff-2003. As for each task of Bakeoff-2005, an assistant segmenter that is trained with TMPT-4 from the other training corpus in the same character encoding is used.

## 6.2 Experimental Results in Corpora of Bakeoff-2006

The comparison between our results and the best existing results in Bakeoff-2006 are shown in Tables XL-XLII. All features in Table XXXIII are used for

Table XXXIX. Comparisons of Best Existing Results and Our Results in the Open Test of Bakeoff-2005

| Participants | $F$-measures in different corpora | | | |
|---|---|---|---|---|
| | AS2005 | CityU2005 | MSRA2005 | PKU2005 |
| Low et al. [2005]/w Ext. dict | 0.955 | 0.960 | 0.968 | 0.965 |
| Low et al. [2005]/Final | 0.956 | 0.962 | 0.968 | 0.969 |
| Best results of Bakeoff-2005 | 0.956 | 0.962 | 0.972 | 0.969 |
| Ours/TMPT-1 +Ext. dict. | 0.9573 | 0.9650 | 0.9787 | 0.9648 |
| Ours/TMPT-1 +Ext. dict. | 0.9592 | 0.9657 | 0.9798 | 0.9665 |
| +(Assis.Seg.) | (CityU2005) | (AS2005) | (PKU2005) | (MSRA2005) |

Table XL. Comparisons of Best Existing Results and Ours in the Closed Test of Bakeoff-2006

| Participant (Site ID) | AS2006 | CityU2006 | CTB2006 | MSRA2006 |
|---|---|---|---|---|
| Wang et al. [2006](32) | 0.953 | 0.970 | 0.930 | 0.963 |
| Tsai et al. [2006](15) | 0.957 | 0.972 | / | 0.955 |
| Zhang et al. [2006](26) | 0.949 | 0.965 | 0.926 | 0.957 |
| Ours/w TMPT-4 | 0.9538 | 0.9691 | 0.9320 | 0.9608 |
| Ours/w TMPT-1 | 0.9586 | 0.9716 | 0.9339 | 0.9613 |

Table XLI. Statistical Significance of the Results in the Closed Test of Bakeoff-2006

| Corpus | #word | Participant | $R$ | $C_r$ | $P$ | $C_p$ | $F$ |
|---|---|---|---|---|---|---|---|
| AS2006 | 91K | Others | 0.961 | ±0.001280 | 0.953 | ±0.001400 | 0.957 |
| | | Ours | 0.9590 | ±0.00131 | 0.9489 | ±0.00146 | 0.9538 |
| CityU2006 | 220K | Others | 0.973 | ±0.000691 | 0.972 | ±0.000703 | 0.972 |
| | | Ours | 0.9688 | ±0.00074 | 0.9695 | ±0.00073 | 0.9691 |
| CTB2006 | 154K | Others | 0.936 | ±0.001244 | 0.923 | ±0.001355 | 0.930 |
| | | Ours | 0.9380 | ±0.00123 | 0.9263 | ±0.00133 | 0.9320 |
| MSRA2006 | 100K | Others | 0.964 | ±0.001176 | 0.961 | ±0.001222 | 0.963 |
| | | Ours | 0.9570 | ±0.00128 | 0.9647 | ±0.00117 | 0.9608 |

open test (the last row in Table XLII). The results of statistical significance tests are given in Table XLI [Levow 2006]. We see that our current system still obtains competitive performance in corpora of Bakeoff-2006.

## 6.3 Additional Metrics

To demonstrate some further difference between our results and the others, we also give a comparison of $F$-measure of OOV word ($F_{OOV}$). Fiv. The results of other participants except for us who ranked the first and the second in $F$-measures are given in Table XLIII.[17] We find that the order of $F_{OOV}$ is basically kept the same as $F$-measures of the whole performance. This further suggests that OOV word detection is very important for improvement of the whole segmentation performance, which is the point that our method pays great attention to.

Similar to Table XLIII, $R_{OOV}$ comparison is also given in Table XLIV. We find that though our system earns the highest $R_{OOV}$ in all four corpora, we do not get the same results in the word segmentation performance on the whole.

---

[17]The data of $F_{OOV}$ before Bakeoff-2006 are absent since the organizers of Bakeoff-2003 and 2005 did not provide $P_{OOV}$ or $F_{OOV}$.

Table XLII. Comparisons of Best Existing Results and Ours in the Open Test of Bakeoff-2006

| Participant | AS2006 | CityU2006 | CTB2006 | MSRA2006 |
|---|---|---|---|---|
| Other best results of Bakeoff | 0.954 | 0.977 | 0.944 | 0.979 |
| Ours | 0.9607 | 0.9778 | 0.9531 | 0.9826 |

Table XLIII. $F_{oov}$ Comparisons of Best Existing Results and Ours in the Closed Test of Bakeoff-2006

| Corpus | $F_{OOV}$ | | | |
|---|---|---|---|---|
| | Ours | | Others | |
| | /w TMPT-4 | / w TMPT-1 | Best results | Second best results |
| AS2006 | 0.6409 | 0.7216 | 0.715 | 0.713 |
| CityU2006 | 0.7379 | 0.7591 | 0.795 | 0.795 |
| CTB2006 | 0.7106 | 0.7211 | 0.722 | 0.709 |
| MSRA2006 | 0.6082 | 0.6129 | 0.635 | 0.559 |

Table XLIV. $R_{oov}$ Comparisons of Best Existing Results and Our Results in the Closed Test of the Corpora of Bakeoff-2006

| Corpus | $R_{OOV}$ | | | |
|---|---|---|---|---|
| | Ours | | Others | |
| | /w TMPT-4 | / w TMPT-1 | Best results | Second best results |
| AS2006 | 0.6687 | 0.7088 | 0.658 | 0.656 |
| CityU2006 | 0.7815 | 0.7927 | 0.787 | 0.787 |
| CTB2006 | 0.7094 | 0.7113 | 0.683 | 0.634 |
| MSRA2006 | 0.6672 | 0.6715 | 0.612 | 0.499 |

This partially suggests that $F_{OOV}$ is a better performance metric than $R_{OOV}$ to evaluate how OOV word identification affects the whole performance.

## 7. RELATED WORK

### 7.1 Feature Templates

Some existing systems with their feature templates, tag sets, and learning models are listed in Table XLV. There is another comparison between the system in Tseng et al. [2005] and ours: we select six $n$-gram feature templates in Table VI for the closed test, while there are 15 groups of feature templates in Tseng's system. However, with an appropriate tag set, our system performs better (see Table XXXVII).

We attribute the superiority of our system to an effective combination of tag sets and feature template sets. A joint selection for such a collocation has demonstrated through experimental results in Tables XV and XVI. Though all feature templates that are originally used to identify OOV words in Tseng et al. [2005] don't appear in our system, we do not lose such types of active features as our system is running, since additional tags with $n$-gram feature templates may still help to identify those consequent characters that finally form Chinese words.

The most significant difference from Peng et al. [2004] and Tseng et al. [2005] to our system is all $n$-gram feature templates that consist of $C_{-2}$ or $C_2$ are removed. It seems that our system only considers three-character window in context, but it is not the actual effect as discussed in Section 3.2. The

Table XLV.  Comparison of Feature Templates, Tag Sets, and Learning Models for the Closed Test

| Type | Xue [2003] | Low et al. [2005] | Peng et al. [2004] | Tseng et al. [2005] |
|---|---|---|---|---|
| Model | MaxEnt | | CRF | |
| Tag set | 4-tag | | 2-tag | |
| $n$-gram | $C_n, n = -2, -1, 0, 1, 2$ | | $C_n, n = -2, -1, 1, 2$ | $C_n, n = -2, -1, 0, 1$ |
| | $C_n C_{n+1}, n = -2, -1, 0, 1$ | | $C_n C_{n+1}, n = -2, -1, 0$ | |
| | $C_{-1} C_1$ | | $C_{-1} C_0 C_1$ | $C_2 C_0$ |
| Other closed feature templates | | | | $C_{-1} = C_0, C_{-1} = C_1$ |
| | | | | $C_{-1} + C_0$ |
| | | | | $C_n, n = -1, 0, 1$ as single-character word |
| | | | | $C_{-1}(C_0)$ as prefix(suffix) character |
| | | $T_{-2} T_{-1} T_0 T_1 T_2$ | | |
| | | $Pu(C_0)$ | | |

five-character window is still handled by our system with even more precise categorization information of characters.

It is possible to construct a state-of-the-art segmentation system still with a two-tag set or four-tag set, which has been verified in Tseng et al. [2005], Low et al. [2005], and Tsai et al. [2006]. However, it is often much more difficult to select useful feature templates than to select tag sets. Therefore, it will be more convenient to consider tag set selection for the first time. On the other hand, we may recognize that more feature templates in Tseng et al. [2005] than ours were used only because the two-tag set was adopted in their system.

In a word, if we may fully use tag set selection incorporated with feature template selection, then we will be able to develop a more effective and simplified system. In addition, this system can achieve competitive performance compared to existing systems.

## 7.2 Postprocessing

There was a postprocessing issue in MaxEnt systems of Xue [2003] and Low et al. [2005]. If each character is just assigned the tag with the highest probability, then it is possible that the MaxEnt classifier sometimes produces an illegal sequence of tags (e.g., *M* is followed by *S*). To eliminate such possibilities, additional techniques were adopted in previous work. Typically, given an input character sequence, a decoding algorithm is demanded, running only within valid tag sequences.

It is fortunate that CRF is a sequence learner which can resist label-bias defined in Lafferty et al. [2001]. Thus, these types of invalid tag sequences never appear during our experiments, only if training corpus itself does not include invalid tag sequences. In detail, MaxEnt Markov model uses a per-state exponential model for the conditional probabilities of next states given the current state, while CRF has a single exponential model for the joint probability of the entire sequence of labels given the observation sequence. Thus, the normalizer $Z(W)$ in Equation (1) should be computed through the entire sequence for CRF model training. This can lead to an optimal model in the whole sequence which MaxEnt Morkov model cannot guarantee.

## 8. CONCLUSION

In this article, we address a traditional and basic issue in Chinese language processing, Chinese word segmentation. Our analysis shows that character-based tagging framework has been a powerful learning framework in this field. Then we adopt the framework with CRF for this study.

Our contribution on the learning framework of Chinese word segmentation are two-fold. 1) We consider both feature template selection and tag set selection instead of previous feature template selection only method. A comprehensive investigation of different tag sets is studied by analysis of average weighted word length computed from segmented corpus. We propose that average weighted word length of the corpus can be taken as the criteria to effectively choose tag set. We also show that a segmentation system with a six-tag set and six *n*-gram feature templates can achieve competitive performance in benchmark data sets from Bakeoffs. 2) As for integration of additional linguistic resources, an assistant segmenter approach is proposed, and its effectiveness is verified. The assistant segmenter method is easy to handle. We also show that this method can be generalized to integrate different types of linguistic resources, including corpora, dictionaries and trained segmenters. In addition, assistant segmenter method is also regarded as an effective standard adaptation method for different segmentation standards.

Based on the proposed method, our system provides state-of-the-art performance in all corpora of three bakeoffs.

## REFERENCES

ASAHARA, M., GOH, C. L., WANG, X., AND MATSUMOTO, Y. 2003. Combining segmenter and chunker for Chinese word segmentation. In *Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing (SIGHAN'03)*. 144–147.

BRILL, E. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Comput. Linguist. 21*, 4, 543–565.

CHEN, A. 2003. Chinese word segmentation using minimal linguistic knowledge. In *Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing (SIGHAN'03)*. 148–151.

COHN, T., SMITH, A., AND OSBORNE, M. 2005. Scaling conditional random fields using error-correcting codes. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. 10–17.

EMERSON, T. 2005. The second international Chinese word segmentation bakeoff. In *Proceedings of the 4th SIGHAN Workshop on Chinese Language Processing (SIGHAN'05)*. 123–133.

FAN, C.-K. AND TSAI, W.-H. 1988. Automatic word identification in Chinese sentences by the relaxation technique. *Comput. Proc. Chinese Oriental Lang. 4*, 1, 33–56.

FU, G.-H. AND WANG, X.-L. 1999. Unsupervised Chinese word segmentation and unknown word identification. In *Proceedings of the 5th Natural Language Processing Pacific Rim Symposium (NLPRS'99)*. 32–37.

GAO, J., LI, M., WU, A., AND HUANG, C.-N. 2005. Chinese word segmentation and named entity recognition: A pragmatic approach. *Comput. Linguist. 31*, 4, 531–574.

GAO, J., WU, A., LI, M., HUANG, C.-N., LI, H., XIA, X., AND QIN, H. 2004. Adaptive Chinese word segmentation. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04)*. 462–469.

GOH, C.-L., ASAHARA, M., AND MATSUMOTO, Y. 2005. Chinese word segmentatin by classification of characters. *Comput. Linguist. Chinese Lang. Proc. 10*, 3, 381–396.

GRINSTEAD, C. AND SNELL, J. L. 1997. *Introduction to Probability*. American Mathematical Society, Providence, RI.

HOCKENMAIER, J. AND BREW, C. 1998. Error driven segmentation of Chinese. *Comm. COLIPS 8*, 1, 69–84.

KUNCHEVA, L. I. AND WHITAKER, C. J. 2003. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Mach. Learn. 51*, 2, 181–207.

LAFFERTY, J. D., MCCALLUM, A., AND PEREIRA, F. C. N. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML'01)*. 282–289.

LAU, T. P. AND KING, I. 2005. Two-phase lmr-rc tagging for Chinese word segmentation. In *Proceedings of the 4th SIGHAN Workshop on Chinese Language Processing (SIGHAN'05)*. 183–186.

LEVOW, G.-A. 2006. The third international Chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the 5th SIGHAN Workshop on Chinese Language Processing (SIGHAN'06)*. 108–117.

LI, M., GAO, J., HUANG, C.-N., AND LI, J. 2003. Unsupervised training for overlapping ambiguity resolution in Chinese word segmentation. In *Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing (SIGHAN'03)*. 1–7.

LI, S. 2005. Chinese word segmentation in ICT-NLP. In *Proceedings of the 4th SIGHAN Workshop on Chinese Language Processing (SIGHAN'05)*. 187–188.

LIANG, P. 2005. Semi-supervised learning for natural language. M.S. thesis, Massachusetts Institute of Technology.

LOW, J. K., NG, H. T., AND GUO, W. 2005. A maximum entropy approach to Chinese word segmentation. In *Proceedings of the 4th SIGHAN Workshop on Chinese Language Processing (SIGHAN'05)*. 161–164.

LUO, X., SUN, M., AND TSOU, B. K. 2002. Covering ambiguity resolution in Chinese word segmentation based on contextual information. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING'02)*. 1–7.

MALOUF, R. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the Conference on Natural Language Learning (CoNLL'02)*. 49–55.

NG, H. T. AND LOW, J. K. 2004. Chinese part-of-speech tagging: One-at-a-time or all-at-once? Word-based or character-based? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'04)*. 277–284.

NIE, J.-Y., JIN, W., AND HANNAN, M.-L. 1994. A hybrid approach to unknown word detection and segmentation of Chinese. In *Proceedings of the International Conference on Chinese Computing (ICCC'94)*. 326–335.

PACKARD, J. 2000. *The Morphology of Chinese: A Linguistics and Cognitive Approach*. Cambridge University Press, Cambridge, UK.

PALMER, D. D. 1997. A trainable rule-based algorithm for word segmentation. In *Proceedings of the Association for Computational Linguistics (ACL'97)*. 321–328.

PENG, F., FENG, F., AND MCCALLUM, A. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING'04)*. 562–568.

RATNAPARKHI, A. 1996. A maximum entropy part-of-speech tagger. In *Proceedings of the Empirical Method in Natural Language Processing Conference (EMNLP'96)*. 133–142.

ROSENFELD, B., FELDMAN, R., AND FRESKO, M. 2006. A systematic cross-comparison of sequence classifiers. In *Proceedings of the SIAM International Conference on Data Mining (SDM'06)*. 563–567.

SHA, F. AND PEREIRA, F. 2003. Shallow parsing with conditional random fields. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (ACL-HLT'03)*. 134–141.

SPROAT, R. AND EMERSON, T. 2003. The first international Chinese word segmentation bakeoff. In *Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing (SIGHAN'03)*. 133–143.

SPROAT, R. AND SHIH, C. 1990. A statistical method for finding word boundaries in Chinese text. *Comput. Proc. Chinese Oriental Lang. 4*, 4, 336–351.

SPROAT, R. AND SHIH, C. 2002. Corpus-based methods in Chinese morphology and phonology. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING'02)*.

SPROAT, R., SHIH, C., GALE, W., AND CHANG, N. 1996. A stochastic finite-state word-segmentation algorithm for Chinese. *Comput. Linguist. 22*, 3, 377–404.

SUN, C., HUANG, C.-N., AND GUAN, Y. 2006. Combinative ambiguity string detection and resolution based on annotated corpus. In *Proceedings of the 3rd Student Workshop on Computational Linguistics (SWCL'06)*.

SUN, M., SHEN, D., AND TSOU, B. K. 1998. Chinese word segmentation without using lexicon and hand-crafted training data. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL'98)*. 1265–1271.

SUN, M. AND TSOU, B. K. 2001. A review and evaluation on automatic segmentation of Chinese (in Chinese) (汉语自动分词研究评述). *Contemporary Linguist. 3*, 1, 22–32.

TSAI, J.-L. 2006. BMM-based Chinese word segmentor with word support model for the SIGHAN bakeoff 2006. In *Proceedings of the 5th SIGHAN Workshop on Chinese Language Processing (SIGHAN'06)*. 130–133.

TSAI, R. T.-H., HUNG, H.-C., SUNG, C.-L., DAI, H.-J., AND HSU, W.-L. 2006. On closed task of Chinese word segmentation: An improved CRF model coupled with character clustering and automatically generated template matching. In *Proceedings of the 5th SIGHAN Workshop on Chinese Language Processing (SIGHAN'06)*. 108–117.

TSENG, H., CHANG, P., ANDREW, G., JURAFSKY, D., AND MANNING, C. 2005. A conditional random field word segmenter for SIGHAN bakeoff 2005. In *Proceedings of the 4th SIGHAN Workshop on Chinese Language Processing (SIGHAN'06)*. 168–171.

WANG, X., LIN, X., YU, D., TIAN, H., AND WU, X. 2006. Chinese word segmentation with maximum entropy and *N*-gram language model. In *Proceedings of the 5th SIGHAN Workshop on Chinese Language Processing (SIGHAN'06)*. 138–141.

WU, A. AND JIANG, Z. 2000. Statistically-enhanced new word identification in a rule-based Chinese system. In *Proceedings of the 2nd Chinese Processing Workshop (ACL'00)*. 46–51.

XUE, N. 2003. Chinese word segmentation as character tagging. *Comput. Linguist. Chinese Lang. Proc. 8*, 1, 29–48.

XUE, N. AND SHEN, L. 2003. Chinese word segmentation as LMR tagging. In *Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing (SIGHAN'03)*. 176–179.

YUAN, Y. 1997. Statistics based approaches towards Chinese language processing. Ph.D. thesis, National University of Singapore.

ZHANG, H.-P. AND LIU, Q. 2003. Chinese lexical analysis using hierarchical hidden markov model. In *Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing (SIGHAN'03)*. 63–70.

ZHANG, M., ZHOU, G.-D., YANG, L.-P., AND JI, D.-H. 2006. Chinese word segmentation and named entity recognition based on a context-dependent mutual information independence model. In *Proceedings of the 5th SIGHAN Workshop on Chinese Language Processing (SIGHAN'06)*. 154–157.

ZHANG, R., KIKUI, G., AND SUMITA, E. 2006. Subword-based tagging by conditional random fields for Chinese word segmentation. In *Proceedings of the Human Language Technology Conference/North American Chapter of the Association for Computational Linguistics (HLT/NAACL'06)*. 193–196.

ZHOU, G. D. 2005. A chunking strategy towards unknown word detection in Chinese word segmentation. In *Proceeding of the 2nd International Joint Conference on Natural Language Processing (IJCNLP'05)*. R. Dale, K.-F. Wong, J. Su, and O. Y. Kwong, Eds. Lecture Notes in Computer Science, vol. 3651, 530–541.

ZHU, M.-H., WANG, Y.-L., WANG, Z.-X., WANG, H.-Z., AND ZHU, J.-B. 2006. Designing special post-processing rules for SVM-based Chinese word segmentation. In *Proceedings of the 5th SIGHAN Workshop on Chinese Language Processing (SIGHAN'06)*. 217–220.