Adaptive Chinese Word Segmentation[1]

Jianfeng Gao[*], **Andi Wu**[*], **Mu Li**[*], **Chang-Ning Huang**[*], **Hongqiao Li**[**], **Xinsong Xia**[$], **Haowei Qin**[&]

[*] Microsoft Research. {jfgao, andiwu, muli, cnhuang}@microsoft.com

[**] Beijing Institute of Technology, Beijing. lhqtxm@bit.edu.cn

[$] Peking University, Beijing. xia_xinsong@founder.com

[&] Shanghai Jiaotong university, Shanghai. haoweiqin@sjtu.edu.cn

Abstract

This paper presents a Chinese word segmentation system which can adapt to different domains and standards. We first present a statistical framework where domain-specific words are identified in a unified approach to word segmentation based on linear models. We explore several features and describe how to create training data by sampling. We then describe a transformation-based learning method used to adapt our system to different word segmentation standards. Evaluation of the proposed system on five test sets with different standards shows that the system achieves state- of-the-art performance on all of them.

1 Introduction

Chinese word segmentation has been a long-standing research topic in Chinese language processing. Recent development in this field shows that, in addition to ambiguity resolution and unknown word detection, the usefulness of a Chinese word segmenter also depends crucially on its ability to adapt to different domains of texts and different segmentation standards.

The need of adaptation involves two research issues that we will address in this paper. The first is new word detection. Different domains/applications may have different vocabularies which contain new words/terms that are not available in a general dictionary. In this paper, new words refer to OOV words other than named entities, factoids and morphologically derived words. These words are mostly domain specific terms (e.g. 蜂窝式 'cellular') and time-sensitive political, social or cultural terms (e.g. 三通'Three Links', 非典 'SARS').

The second issue concerns the customizable display of word segmentation. Different Chinese NLP-enabled applications may have different requirements that call for different granularities of word segmentation. For example, speech recognition systems prefer "longer words" to achieve higher accuracy whereas information retrieval systems prefer "shorter words" to obtain higher recall rates, etc. (Wu, 2003). Given a word segmentation specification (or standard) and/or some application data used as training data, a segmenter with customizable display should be able to provide alternative segmentation units according to the specification which is either pre-defined or implied in the data.

In this paper, we first present a statistical framework for Chinese word segmentation, where various problems of word segmentation are solved simultaneously in a unified approach. Our approach is based on linear models where component models are inspired by the source-channel models of Chinese sentence generation. We then describe in detail how the new word identification (NWI) problem is handled in this framework. We explore several features and describe how to create training data by sampling. We evaluate the performance of our segmentation system using an annotated test set, where new words are simulated by sampling. We then describe a transformation-based learning (TBL, Brill, 1995) method that is used to adapt our system to different segmentation standards. We compare the adaptive system to other state-of-the-art systems using four test sets in the SIGHAN's First International Chinese Word Segmentation Bakeoff, each of which is constructed according to a different segmentation standard. The performance of our system is comparable to the best systems reported on all four test sets. It demonstrates the possibility of having a single adaptive Chinese word segmenter that is capable of supporting multiple user applications.

[1] This work was done while Hongqiao Li, Xinsong Xia and Haowei Qin were visiting Microsoft Research (MSR) Asia. We thank Xiaodan Zhu for his early contribution, and the three reviewers, one of whom alerted us the related work of (Uchimoto *et al*., 2001).

Word Class[2]	Model	Feature Functions, $f(S,W)$
Context Model	Word class based trigram, $P(W)$.	-log($P(W)$)
Lexical Word (LW)	---	1 if S forms a word lexicon entry, 0 otherwise.
Morphological Word (MW)	---	1 if S forms a morph lexicon entry, 0 otherwise.
Named Entity (NE)	Character/word bigram, $P(S\|NE)$.	-log($P(S\|NE)$)
Factoid (FT)	---	1 if S can be parsed using a factoid grammar, 0 otherwise
New Word (NW)	---	Score of SVM classifier

Figure 1: Context model, word classes, and class models, and feature functions.

2 Chinese Word Segmentation with Linear Models

Let S be a Chinese sentence which is a character string. For all possible word segmentations W, we will choose the most likely one W^* which achieves the highest conditional probability $P(W|S)$: $W^* = \text{argmax}_W P(W|S)$. According to Bayes' decision rule and dropping the constant denominator, we can equivalently perform the following maximization:

$$W^* = \arg\max_W P(W)P(S \mid W) . \quad (1)$$

Equation (1) represents a source-channel approach to Chinese word segmentation. This approach models the generation process of a Chinese sentence: first, the speaker selects a sequence of concepts W to output, according to the probability distribution $P(W)$; then he attempts to express each concept by choosing a sequence of characters, according to the probability distribution $P(S|W)$.

We define *word class* as a group of words that are supposed to be generated according to the same distribution (or in the same manner). For instance, all Chinese person names form a word class. We then have multiple channel models, each for one word class. Since a channel model estimates the likelihood that a character string is generated given a word class, it is also referred to as *class model*. Similarly, source model is referred to as *context model* because it indicates the likelihood that a word class occurs in a context. We have only one context model which is a word-class-based trigram model. Figure 1 shows word classes and class models that we used in our system. We notice that different class models are constructed in different ways (e.g. name entity models are *n*-gram models trained on corpora whereas factoid models use derivation rules and have binary values). The dynamic value ranges of different class models can be so different that it is improper to combine all models through simple multiplication as Equation (1).

In this study we use linear models. The method is derived from linear discriminant functions widely used for pattern classification (Duda *et al.*, 2001), and has been recently introduced into NLP tasks by Collins and Duffy (2001). It is also related to log-linear models for machine translation (Och, 2003).

In this framework, we have a set of $M+1$ feature functions $f_i(S,W)$, $i = 0,\ldots,M$. They are derived from the context model (i.e. $f_0(W)$) and M class models, each for one word class, as shown in Figure 1: For probabilistic models such as the context model or person name model, the feature functions are defined as the negative logarithm of the corresponding probabilistic models. For each feature function, there is a model parameter λ_i. The best word segmentation W^* is determined by the decision rule as

$$W^* = \arg\max_W Score(\lambda_0^M, S, W) = \arg\max_W \sum_{i=0}^{M} \lambda_i f_i(S,W) \quad (2)$$

Below we describe how to optimize λs. Our method is a discriminative approach inspired by the Minimum Error Rate Training method proposed in Och (2003). Assume that we can measure the number of segmentation errors in W by comparing it with a reference segmentation R using a function $Er(R,W)$. The training criterion is to minimize the count of errors over the training data as

$$\hat{\lambda}_1^M = \arg\min_{\lambda_1^M} \sum_{S,W,R} Er(R, W(S, \lambda_1^M)) , \quad (3)$$

where W is detected by Equation (2). However, we cannot apply standard gradient descent to optimize

[2] In our system, we define three types of named entity: person name (PN), location name (LN), organization (ON) and transliteration name (TN); ten types of factoid: date, time (TIME), percentage, money, number (NUM), measure, e-mail, phone number, and WWW; and five types of morphologically derived words (MDW): affixation, reduplication, merging, head particle and split.

Initialization: $\lambda_0=\alpha$, $\lambda_i=1$, $i = 1,\ldots,M$.
For $t = 1 \ldots T$, $j = 1 \ldots N$
$W_j = \text{argmax} \sum \lambda_i f_i(S_j,W)$
For $i = 1 \ldots M$
$\lambda_i = \lambda_i + \eta(Score(\boldsymbol{\lambda},S,W)-Score(\boldsymbol{\lambda},S,R))(f_i(R) - f_i(W))$,
where $\boldsymbol{\lambda}=\{\lambda_0, \lambda_1,\ldots,\lambda_M\}$ and η =0.001.

Figure 2: The training algorithm for model parameters

model parameters according to Equation (3) because the gradient cannot be computed explicitly (i.e., *Er* is not differentiable), and there are many local minima in the error surface. We then use a variation called stochastic gradient descent (or *unthresholded perceptron*, Mitchell, 1997). As shown in Figure 2, the algorithm takes *T* passes over the training set (i.e. *N* sentences). All parameters are initially set to be 1, except for the context model parameter λ_0 which is set to be a constant α during training, and is estimated separately on held-out data. Class model parameters are updated in a simple additive fashion. Notice that $Score(\boldsymbol{\lambda},S,W)$ is not less than $Score(\boldsymbol{\lambda},S,R)$. Intuitively the updated rule increases the parameter values for word classes whose models were "underestimated" (i.e. expected feature value $f(W)$ is less than observed feature value $f(R)$), and decreases the parameter values whose models were "overestimated" (i.e. $f(W)$ is larger than $f(R)$). Although the method cannot guarantee a global optimal solution, it is chosen for our modeling because of its efficiency and the best results achieved in our experiments.

Given the linear models, the procedure of word segmentation in our system is as follows: First, all word candidates (lexical words and OOV words of certain types) are generated, each with its word class tag and class model score. Second, Viterbi search is used to select the best W according to Equation (2). Since the resulting W^* is a sequence of segmented words that are either lexical words or OOV words with certain types (e.g. person name, morphological words, new words) we then have a system that can perform word segmentation and OOV word detection simultaneously in a unified approach. Most previous works treat OOV word detection as a separate step after word segmentation. Compared to these approaches, our method avoids the error propagation problem and can incorporate a variety of knowledge to achieve a globally optimal solution. The superiority of the unified approach has been demonstrated empirically in Gao *et al.* (2003), and will also be discussed in Section 5.

3 New Word Identification

New words in this section refer to OOV words that are neither recognized as named entities or factoids nor derived by morphological rules. These words are mostly domain specific and/or time-sensitive. The identification of such new words has not been studied extensively before. It is an important issue that would have substantial impact on the performance of word segmentation. For example, approximately 30% of OOV words in the SIGHAN's **PK** corpus (see Table 1) are new words of this type. There has been previous work on detecting Chinese new words from a large corpus in an off-line manner and updating the dictionary before word segmentation. However, our approach is able to detect new words on-line, i.e. to spot new words in a sentence on the fly during the process of word segmentation where widely-used statistical features such as mutual information or term frequency are not available.

For brevity of discussion, we will focus on the identification of 2-character new words, denoted as NW_11. Other types of new words such as NW_21 (a 2-character word followed with a character) and NW_12 can be detected similarly (e.g. by viewing the 2-character word as an inseparable unit, like a character). Below, we shall describe the class model and context model for NWI, and the creation of training data by sampling.

3.1 Class Model

We use a classifier (SVM in our experiments) to estimate the likelihood of two adjacent characters to form a new word. Of the great number of features we experimented, three linguistically-motivated features are chosen due to their effectiveness and availability for on-line detection. They are *Independent Word Probability* (IWP), *Anti-Word Pair* (AWP), and *Word Formation Analogy* (WFA). Below we describe each feature in turn. In Section 3.2, we shall describe the way the training data (new word list) for the classifier is created by sampling.

IWP is a real valued feature. Most Chinese characters can be used either as independent words or component parts of multi-character words, or both. The IWP of a single character is the likelihood for this character to appear as an independent word in texts (Wu and Jiang, 2000):

$$IWP(x) = \frac{C(x,W)}{C(x)}. \quad (4)$$

where $C(x, W)$ is the number of occurrences of the character x as an independent word in training data, and $C(x)$ is the total number of x in training data. We assume that the IWP of a character string is the product of the IWPs of the component characters. Intuitively, the lower the IWP value, the more likely the character string forms a new word. In our implementation, the training data is word-segmented.

AWP is a binary feature derived from IWP. For example, the value of AWP of an NW_11 candidate *ab* is defined as: $AWP(ab)=1$ if $IWP(a)>\theta$ or $IWP(b)>\theta$, 0 otherwise. $\theta \in [0, 1]$ is a pre-set threshold. Intuitively, if one of the component characters is very likely to be an independent word, it is unlikely to be able to form a word with any other characters. While IWP considers all component characters in a new word candidate, AWP only considers the one with the maximal IWP value.

WFA is a binary feature. Given a character pair (x, y), a character (or a multi-character string) z is called the *common stem* of (x, y) if at least one of the following two conditions hold: (1) character strings xz and yz are lexical words (i.e. x and y as prefixes); and (2) character strings zx and zy are lexical words (i.e. x and y as suffixes). We then collect a list of such character pairs, called *affix pairs*, of which the number of common stems is larger than a pre-set threshold. The value of WFA for a given NW_11 candidate *ab* is defined as: $WFA(ab) = 1$ if there exist an affix pair (a, x) (or (b, x)) and the string xb (or ax) is a lexical word, 0 otherwise. For example, given an NW_11 candidate 下岗 (xia4-gang3, 'out of work'), we have $WFA(下岗) = 1$ because (上, 下) is an affix pair (they have 32 common stems such as _任, _游, _台, _车, _面, _午, _班) and 上岗 (shang4-gang3, 'take over a shift') is a lexical word.

3.2 Context Model

The motivations of using context model for NWI are two-fold. The first is to capture useful contextual information. For example, new words are more likely to be nouns than pronouns, and the POS tagging is context-sensitive. The second is more important. As described in Section 2, with a context model, NWI can be performed simultaneously with other word segmentation tasks (e.g.: word break, named entity recognition and morphological analysis) in a unified approach.

However, it is difficult to develop a training corpus where new words are annotated because "we usually do not know what we don't know". Our solution is Monte Carlo simulation. We sample a set of new words from our dictionary according to the distribution – the probability that any lexical word w would be a new word $P(NW|w)$. We then generate a new-word-annotated corpus from a word-segmented text corpus.

Now we describe the way $P(NW|w)$ is estimated. It is reasonable to assume that new words are those words whose probability to appear in a new document is lower than general lexical words. Let $P_i(k)$ be the probability of word w_i that occurs k times in a document. In our experiments, we assume that $P(NW|w_i)$ can be approximated by the probability of w_i occurring less than K times in a new document:

$$P(NW \mid w_i) \approx \sum_{k=0}^{K-1} P_i(k), \tag{5}$$

where the constant K is dependent on the size of the document: The larger the document, the larger the value. $P_i(k)$ can be estimated using several term distribution models (see Chapter 15.3 in Manning and Schütze, 1999). Following the empirical study in (Gao and Lee, 2000), we use K-Mixture (Katz, 1996) which estimate $P_i(k)$ as

$$P_i(k) = (1-\alpha)\delta_{k,0} + \frac{\alpha}{\beta+1}\left(\frac{\beta}{\beta+1}\right)^k, \tag{6}$$

where $\delta_{k,0}=1$ if $k=0$, 0 otherwise. α and β are parameters that can be fit using the observed mean λ and the observed inverse document frequency IDF as follow:

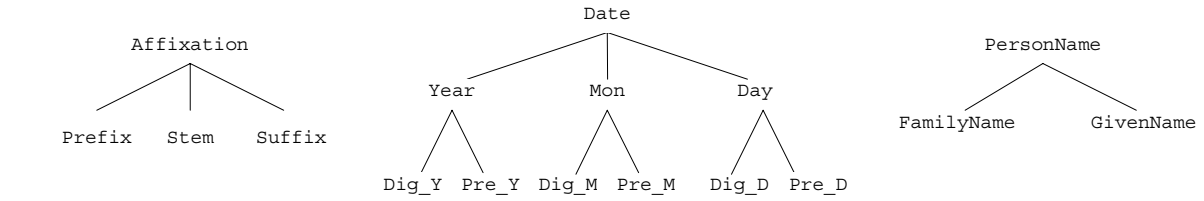
$$\lambda = \frac{cf}{N},\ IDF = \log\frac{N}{df},$$

$$\beta = \lambda \times 2^{IDF} - 1 = \frac{cf - df}{df}, \text{ and } \alpha = \frac{\lambda}{\beta},$$

where cf is the total number of occurrence of word w_i in training data, df is the number of documents in training data that w_i occurs in, and N is the total number of documents. In our implementation, the training data contain approximately 40 thousand documents that have been balanced among domain, style and time.

4 Adaptation to Different Standards

The word segmentation standard (or standard for brevity) varies from system to system because there is no commonly accepted definition of Chinese



Condition: 'Affixation'
Actions: Insert a boundary between 'Prefix' and 'Stem'…

Condition: 'Date'
Actions: Insert a boundary between 'Year' and 'Mon' …

Condition: 'PersonName'
Actions: Insert a boundary between 'FamilyName' and 'GivenName'…

Figure 3: Word internal structure and class-type transformation templates.

words and different applications may have different requirements that call for different granularities of word segmentation.

It is ideal to develop a single word segmentation system that is able to adapt to different standards. We consider the following standard adaptation paradigm. Suppose we have a 'general' standard pre-defined by ourselves. We have also created a large amount of training data which are segmented according to this general standard. We then develop a *generic* word segmenter, i.e. the system described in Sections 2 and 3. Whenever we deploy the segmenter for any application, we need to customize the output of the segmenter according to an application-specific standard, which is not always explicitly defined. However, it is often implicitly defined in a given amount of application data (called adaptation data) from which the specific standard can be partially learned.

In our system, the standard adaptation is conducted by a postprocessor which performs an ordered list of transformations on the output of the generic segmenter – removing extraneous word boundaries, and inserting new boundaries – to obtain a word segmentation that meets a different standard.

The method we use is transformation-based learning (Brill, 1995), which requires an initial segmentation, a goal segmentation into which we wish to transform the initial segmentation and a space of allowable transformations (i.e. transformation templates). Under the abovementioned adaptation paradigm, the initial segmentation is the output of the generic segmenter. The goal segmentation is adaptation data. The transformation templates can make reference to words (i.e. lexicalized templates) as well as some pre-defined types (i.e. class-type based templates), as described below.

We notice that most variability in word segmentation across different standards comes from those words that are not typically stored in the dictionary. Those words are dynamic in nature and are usually formed through productive morphological processes. In this study, we focus on three categories: morphologically derived words (MDW), named entities (NE) and factoids.

For each word class that belongs to these categories[2], we define an internal structure similar to (Wu, 2003). The structure is a tree with 'word class' as the root, and 'component types' as the other nodes. There are 30 component types. As shown in Figure 3, the word class `Affixation` has three component types: `Prefix`, `Stem` and `Suffix`. Similarly, `PersonName` has two component types and `Date` has nine – 3 as non-terminals and 6 as terminals. These internal structures are assigned to words by the generic segmenter at run time.

The transformation templates for words of the above three categories are of the form:

Condition: word class
Actions:
- Insert – place a new boundary between two component types.
- Delete – remove an existing boundary between two component types.

Since the application of the transformations derived from the above templates are conditioned on word class and make reference to component types, we call the templates *class-type transformation templates*. Some examples are shown in Figure 3.

In addition, we also use *lexicalized transformation templates* as:

- Insert – place a new boundary between two lemmas.

- `Delete – remove an existing boundary between two lemmas.`

Here, lemmas refer to those basic lexical words that cannot be formed by any productive morphological process. They are mostly single characters, bi-character words, and 4-character idioms.

In short, our adaptive Chinese word segmenter consists of two components: (1) a generic segmenter that is capable of adapting to the vocabularies of different domains and (2) a set of output adaptors, learned from application data, for adapting to different "application-specific" standards

5 Evaluation

We evaluated the proposed adaptive word segmentation system (henceforth **AWS**) using five different standards. The training and test corpora of these standards are detailed in Table 1, where **MSR** is defined by ourselves, and the other four are standards used in SIGHAN's First International Chinese Word Segmentation Bakeoff (Bakeoff test sets for brevity, see Sproat and Emperson (2003) for details).

Corpus	Abbrev.	# Tr. Word	# Te. Word
'General' standard	**MSR**	20M	226K
Beijing University	**PK**	1.1M	17K
U. Penn Chinese Treebank	**CTB**	250K	40K
Hong Kong City U.	**HK**	240K	35K
Academia Sinica	**AS**	5.8M	12K

Table 1: standards and corpora.

MSR is used as the general standard in our experiments, on the basis of which the generic segmenter has been developed. The training and test corpora were annotated manually, where there is only one allowable word segmentation for each sentence. The training corpus contains approximately 35 million Chinese characters from various domains of text such as newspapers, novels, magazines etc. 90% of the training corpus are used for context model training, and 10% are held-out data for model parameter training as shown in Figure 2. The NE class models, as shown in Figure 1, were trained on the corresponding NE lists that were collected separately. The test set contains a total of 225,734 tokens, including 205,162 lexicon/morph-lexicon words, 3,703 PNs, 5,287 LNs, 3,822 ONs, and 4,152 factoids. In Section 5.1, we will describe some simulated test sets that are derived from the **MSR** test set by sampling NWs from a 98,686-entry dictionary.

The four Bakeoff standards are used as 'specific' standards into which we wish to adapt the general standard. We notice in Table 1 that the sizes of adaptation data sets (i.e. training corpora of the four Bakeoff standards) are much smaller than that of the **MSR** training set. The experimental setting turns out to be a good simulation of the adaptation paradigm described in Section 4.

The performance of word segmentation is measured through test precision (P), test recall (R), F score (which is defined as 2PR/(P+R)), the OOV rate for the test corpus (on Bakeoff corpora, OOV is defined as the set of words in the test corpus not occurring in the training corpus.), the recall on OOV words (Roov), and the recall on in-vocabulary (Riv) words. We also tested the statistical significance of results, using the criterion proposed by Sproat and Emperson (2003), and all results reported in this section are significantly different from each other.

5.1 NWI Results

This section discusses two factors that we believe have the most impact on the performance of NWI. First, we compare methods where we use the NWI component (i.e. an SVM classifier) as a post-processor versus as a feature function in the linear models of Equation (2). Second, we compare different sampling methods of creating simulated training data for context model. Which sampling method is best depends on the nature of $P(NW|w)$. As described in Section 3.2, $P(NW|w)$ is unknown and has to be approximated by $P_i(k)$ in our study, so it is expected that the closer $P(NW|w)$ and $P_i(k)$ are, the better the resulting context model. We compare three estimates of $P_i(k)$ in Equation (5) using term models based on Uniform, Possion, and K- Mixture distributions, respectively.

Table 2 shows the results of the generic segmenter on three test sets that are derived from the **MSR** test set using the above three different sampling methods, respectively. For all three distributions, unified approaches (i.e. using NWI component as a feature function) outperform consecutive approaches (i.e. using NWI component as a post-processor). This demonstrates empirically the benefits of using context model for NWI and the unified approach to Chinese word segmentation, as described in 3.2. We also perform NWI on Bakeoff

	# of NW	**AWS** w/o NW		**AWS** w/ NW (post-processor)				**AWS** w/ NW (unified approach)			
		word segmentation		word segmentation		NW		word segmentation		NW	
		P%	R%	P%	R%	P%	R%	P%	R%	P%	R%
Uniform	5,682	92.6	94.5	94.7	95.2	64.1	66.8	95.1	95.5	68.1	78.4
Poisson	3,862	93.4	95.6	94.5	95.9	61.4	45.6	95.0	95.7	57.2	60.6
K-Mixture	2,915	94.7	96.4	95.1	96.2	44.1	41.5	95.6	96.2	46.2	60.4

Table 2: NWI results on **MSR** test set, NWI as post-processor versus unified approach

	PK						**CTB**					
	P	R	F	OOV	Roov	Riv	P	R	F	OOV	Roov	Riv
1. AWS w/o adaptation	.824	.854	.839	.069	.320	.861	.799	.818	.809	.181	.624	.861
2. AWS	**.952**	**.959**	**.955**	**.069**	**.781**	**.972**	**.895**	**.914**	**.904**	**.181**	**.746**	**.950**
3. AWS w/o NWI	.949	.963	.956	.069	.741	.980	.875	.910	.892	.181	.690	.959
4. FMM w/ adaptation	.913	.946	.929	.069	.524	.977	.805	.874	.838	.181	.521	.952
5. Rank 1 in Bakeoff	.956	.963	**.959**	.069	.799	.975	.907	.916	**.912**	.181	.766	.949
6. Rank 2 in Bakeoff	.943	.963	.953	.069	.743	.980	.891	.911	.901	.181	.736	.949

Table 3: Comparison scores for **PK** open and **CTB** open.

	HK						**AS**					
	P	R	F	OOV	Roov	Riv	P	R	F	OOV	Roov	Riv
1. AWS w/o adaptation	.819	.822	.820	.071	.593	.840	.832	.838	.835	.021	.405	.847
2. AWS	**.948**	**.960**	**.954**	**.071**	**.746**	**.977**	**.955**	**.961**	**.958**	**.021**	**.584**	**.969**
3. AWS w/o NWI	.937	.958	.947	.071	.694	.978	.958	.943	.951	.021	.436	.969
4. FMM w/ adaptation	.818	.823	.821	.071	.591	.841	.930	.947	.939	.021	.160	.964
5. Rank 1 in Bakeoff	.954	.958	**.956**	.071	.788	.971	.894	.915	.904	.021	.426	.926
6. Rank 2 in Bakeoff	.863	.909	.886	.071	.579	.935	.853	.892	.872	.021	.236	.906

Table 4: Comparison scores for **HK** open and **AS** open.

test sets. As shown in Tables 3 and 4 (Rows 2 and 3), the use of NW functions (via the unified approach) substantially improves the word segmentation performance.

We find in our experiments that NWs sampled by Possion and K-Mixture are mostly specific and time-sensitive terms, in agreement with our intuition, while NWs sampled by Uniform include more common words and lemmas that are easier to detect. Consequently, by Uniform sampling, the P/R of NWI is the highest but the P/R of the overall word segmentation is the lowest, as shown in Table 2. Notice that the three sampling methods are not comparable in terms of P/R of NWI in Table 2 because of different sampling result in different sets of new words in the test set. We then perform NWI on Bakeoff test sets where the sets of new words are less dependent on specific sampling methods. The results however do not give a clear indication which sampling method is the best because the test sets are too small to show the difference. We then leave it to future work a thorough empirical comparison among different sampling methods.

5.2 Standard Adaptation Results

The results of standard adaptation on four Bakeoff test sets are shown in Tables 3 and 4. A set of transformations for each standard is learnt using TBL from the corresponding Bakeoff training set. For each test set, we report results using our system with and without standard adaptation (Rows 1 and 2). It turns out that performance improves dramatically across the board in all four test sets.

For comparison, we also include in each table the results of using the forward maximum matching (FMM) greedy segmenter as a generic segmenter (Row 4), and the top 2 scores (sorted by F) that are reported in SIGHAN's First International Chinese Word Segmentation Bakeoff (Rows 5 and 6). We can see that with adaptation, our generic segmenter can achieve state-of-the-art performance on different standards, showing its superiority over other systems. For example, there is no single segmenter in SIGHAN's Bakeoff, which achieved top-2 ranks in all four test sets (Sproat and Emperson, 2003).

We notice in Table 3 and 4 that the quality of adaptation seems to depend largely upon the size of adaptation data: we outperformed the best bakeoff systems in the **AS** set because the size of the adaptation data is big while we are worse in the **CTB** set because of the small size of the adaptation data. To verify our speculation, we evaluated the adaptation results using subsets of the **AS** training set of different sizes, and observed the same trend. However, even with a much smaller adaptation data set (e.g. 250K), we still outperform the best bakeoff results.

6 Related Work

Many methods of Chinese word segmentation have been proposed (See Wu and Tseng, 1993; Sproat and Shih, 2001 for reviews). However, it is difficult to compare systems due to the fact that there is no widely accepted standard. There has been less work on dealing with NWI and standard adaptation.

All feature functions in Figure 1, except the NW function, are derived from models presented in (Gao *et al*., 2003). The linear models are similar to what was presented in Collins and Duffy (2001). An alternative to linear models is the log-linear models suggested by Och (2003). See Collins (2002) for a comparison of these approaches.

The features for NWI were studied in Wu & Jiang (2000) and Li *et al*. (2004). The use of sampling was proposed in Della Pietra *et al*. (1997) and Rosenfeld *et al*. (2001). There is also a related work on this line in Japanese (Uchimoto *et al*., 2001).

A detailed discussion on differences among the four Bakeoff standards is presented in Wu (2003), which also proposes an adaptive system where the display of the output can be customized by users. The method described in Section 4 can be viewed as an improved version in that the transformations are learnt automatically from adaptation data. The use of TBL for Chinese word segmentation was first suggested in Palmer (1997).

7 Conclusion

This paper presents a statistical approach to adaptive Chinese word segmentation based on linear models and TBL. The system has two components: A generic segmenter that can adapt to the vocabularies of different domains, and a set of output adaptors, learned from application data, for adapting to different "application-specific" standards. We evaluate our system on five test sets, each corresponding to a different standard. We achieve state-of-the-art performance on all test sets.

References

Brill, Eric. 1995. Transformation-based error-driven learning and natural language processing: a case study in Part-of-Speech tagging. In: *Computational Linguistics*, 21(4).

Collins, Michael and Nigel Duffy. 2001. Convolution kernels for natural language. In: *Advances in Neural Information Processing Systems (NLPS 14)*.

Collins, Michael. 2002. Parameter estimation for statistical parsing models: theory and practice of distribution-free methods. To appear.

Della Pietra, S., Della Pietra, V., and Lafferty, J. 1997. Inducing features of random fields. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, 380-393.

Duda, Richard O, Hart, Peter E. and Stork, David G. 2001. Pattern classification. John Wiley & Sons, Inc.

Gao, Jianfeng and Kai-Fu Lee. 2000. Distribution based pruning of backoff language models. In: *ACL2000*.

Gao, Jianfeng, Mu Li and Chang-Ning Huang. 2003. Improved source-channel model for Chinese word segmentation. In: *ACL2003*.

Katz, S. M. 1996. Distribution of content words and phrases in text and language modeling, In: *Natural Language Engineering*, 1996(2): 15-59

Li, Hongqiao, Chang-Ning Huang, Jianfeng Gao and Xiaozhong Fan. 2004. The use of SVM for Chinese new word identification. In: *IJCNLP2004*.

Manning, C. D. and H. Schütze, 1999. Foundations of Statistical Natural Language Processing. The MIT Press.

Mitchell, Tom M. 1997. Machine learning. The McGraw-Hill Companies, Inc.

Och, Franz. 2003. Minimum error rate training in statistical machine translation. In: *ACL2003*.

Palmer, D. 1997. A trainable rule-based algorithm for word segmentation. In: *ACL '97*.

Rosenfeld, R., S. F. Chen and X. Zhu. 2001. Whole sentence exponential language models: a vehicle for linguistic statistical integration. In: *Computer Speech and Language*, 15 (1).

Sproat, Richard and Chilin Shih. 2002. Corpus-based methods in Chinese morphology and phonology. In: *COLING 2002*.

Sproat, Richard and Tom Emerson. 2003. The first international Chinese word segmentation bakeoff. In: *SIGHAN 2003*.

Uchimoto, K., S. Sekine and H. Isahara. 2001. The unknown word problem: a morphological analysis of Japanese using maximum entropy aided by a dictionary. In: *EMNLP2001*.

Wu, Andi and Zixin Jiang. 2000. Statistically-enhanced new word identification in a rule-based Chinese system. In: *Proc of the 2rd ACL Chinese Processing Workshop.*

Wu, Andi. 2003. Customizable segmentation of morphologically derived words in Chinese. In: *International Journal of Computational Linguistics and Chinese Language Processing*, 8(1): 1-27.

Wu, Zimin and Gwyneth Tseng. 1993. Chinese text segmentation for text retrieval achievements and problems. In: *JASIS*, 44(9): 532-542.