Word Extraction Based on Semantic Constraints in Chinese Word-Formation

Maosong Sun[1], Shengfen Luo[1], and Benjamin K T'sou[2]

[1] National Lab. of Intelligent Tech. & Systems,
Tsinghua University, Beijing 100084, China
sms@mail.tsinghua.edu.cn

[2] Language Information Sciences Research Centre, City University of Hong Kong
rlbtsou@cityu.edu.hk

Abstract. This paper presents a novel approach to Chinese word extraction based on semantic information of characters. A thesaurus of Chinese characters is conducted. A Chinese lexicon with 63,738 two-character words, together with the thesaurus of characters, are explored to learn semantic constraints between characters in Chinese word-formation, forming a semantic-tag-based HMM. The Baum-Welch re-estimation scheme is then chosen to train parameters of the HMM in the way of unsupervised learning. Various statistical measures for estimating the likelihood of a character string being a word are further tested. Large-scale experiments show that the results are promising: the F-score of this word extraction method can reach 68.5% whereas its counterpart, the character-based mutual information method, can only reach 47.5%.

1 Introduction

Processing of unknown words is important for Chinese word identification in running texts. New words are generated quite often with the rapid development of Chinese society. In experience, the accuracy of a word identification system will decrease about 10% if unknown words are not treated properly [12].

Chinese is an isolating language. Methods for processing of unknown words in inflective languages, like, for example [5], may not be appropriate for Chinese because of its different morphological structure. A Chinese word is composed of either single or multiple Chinese characters. In most cases, a Chinese character has at least one sense, and can stand independently at the morphological level. The task of extracting Chinese words with multi-characters from texts is quite similar to that of extracting phrases (e.g., compound nouns) in English, if we regard Chinese characters as English words.

Researches in this field have been done extensively. Generally, there are two kinds of methods for word/phrase extraction, i.e., rule-based and statistic-based. The latter has become the mainstream of the state-of-the-art. In statistic-based approaches, the soundness of an extracted item being a word/phrase is usually estimated by the associative strength between constituents of it. Two widely used statistical measures for

A. Gelbukh (Ed.): CICLing 2005, LNCS 3406, pp. 202–213, 2005.

quantifying the associative strength are frequency and mutual information [1, 2, 7, 10]. Some variations/derivations of these two basic types, log-likelihood for instance, are also exploited [3, 4, 9, 11].

All work so far on Chinese word extraction has depended directly on characters involved in extracted items to measure the associative strength. These approaches ignored an important characteristic of Chinese words: each character of a word is usually meaningful, thus the sense sequence of the involved characters may reflect the semantic constraint 'hidden' in the word to some extent. Consequently, all sense sequences over a lexicon would constitute complete semantic constraints underlying Chinese word-formation. This suggests that semantic constraints in the lexicon implicitly may be helpful for validating Chinese words. The biggest advantage achieved by taking the semantic information into consideration is that we can make certain degree of inference in word extraction. For example, suppose '美军' (American army), '日军' (Japanese army) and '苏军' (Soviet army) are contained in the lexicon, whereas '俄军' (Russian army) is not. We find that all these four words bear the same sense sequence 'country+army' ('美' for the United States, '日' for Japan, '苏' for Soviet Union, and '俄' for Russia), so a hypothesis comes: '俄军' is possibly a word. The idea is simple and straightforward, but it is radically different from previous ones: word extraction will depend heavily on senses of characters, rather than on characters. Furthermore, the associative strength can also be determined statistically using senses of characters. A side effect of doing so is that the data sparseness problem in word extraction may be better settled.

The paper will focus on this novel approach. Section 2 introduces the key linguistic resources used, Section 3 describes the proposed method in detail, and Section 4 gives experimental results and analyses. We conclude in Section 5.

2 Key Linguistic Resources Used

Two key linguistic resources are mainly used in this research: *THSCS*, a thesaurus of Chinese characters, and, *THW2*, a Chinese lexicon.

We firstly developed *THSCS* (short for the Semantic Coding System for Chinese Characters), a thesaurus of Chinese characters. It covers all 6,763 Chinese characters defined in GB-2312, a National Standard of China for Chinese character set in information exchange. In *THSCS*, each character is assigned its possible semantic categories (semantic tags) manually. The principle in designing *THSCS* is that its semantic hierarchy is as compatible as possible with that of *TYCCL*, a well-known thesaurus of Chinese words [8].

There are totally 1,380 semantic categories in *THSCS*. Their distributions are not balanced. As shown in Fig. 1, the most frequent category occurs 927 times, but a majority of categories occur only a few times: 36.4% no more than 5 times, and 87.0% no more than 20 times.

About 54.12% of the 6,763 characters are polysemous according to *THSCS*. Table 1 gives the distributions of these polysemous characters. Note that these polysemous characters are more active than those with single category. An observation over all 32,624 two-character words in *TYCCL* shows that only 1.40% of them do not contain any polysemous characters.

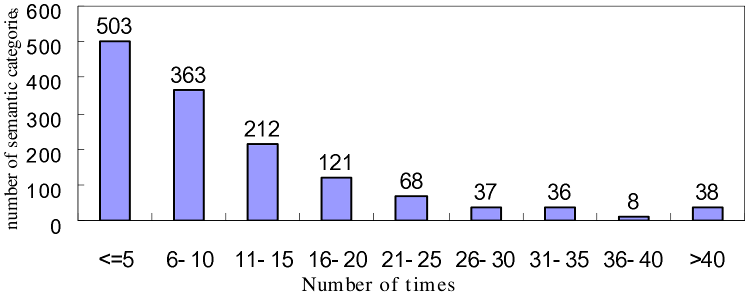

Fig. 1. Distribution of semantic categories in THSCS

Table 1. Distribution of polysemous characters

# of senses per character	# of characters	Percentage in polysemous characters
2	1,556	42.7%
3	787	21.6%
4	457	12.5%
5	285	7.8%
6	181	5.0%
7	124	3.4%
8	79	2.17%
9	51	0.85%
More than 9	123	3.9%

THW2, a lexicon with 63,378 two-character words, is used to learn semantic constraints underlying Chinese word-formation. The reason for choosing two-character words is that they comprise the largest proportion in a Chinese lexicon and represent the most popular word-formation of Chinese.

3 The Proposed Method

3.1 Representing Semantic Constraints in Word-Formation by Hidden Markov Model

Let *C* be the set of Chinese characters, *T* be a thesaurus over *C*, *S* be the set of semantic tags derived from *T*, *W* be the set of Chinese wordlist of two-character words, and *WS* be the set of pairs <word, semantic tags> over *W* in which every character in a word is assigned a unique semantic tag (though the character may possibly have multiple semantic tags in terms of *T*). Then we can construct a Hidden Markov Model (HMM) accordingly as a five-tuple $WF_{sem} = (S, S_0, C, P_S, P_C)$:

S serves as the set of states; $S_0 \in S$ is the set of initial states associated with the initials of semantic tag sequences of *W*; C serves as the set of output alphabet; $P_S = \{p(s_j \mid s_i)\}$ ($s_i \in S$, $s_j \in S$) is the set of transition probabilities among states; $P_C = \{p(c_k \mid s_i, s_j)\}$ ($s_i \in S$, $s_j \in S$, $c_k \in C$) is the set of observation probabilities.

Both P_S and P_C will be trained using *WS*.

This five-tuple (S, S_0, C, P_S, P_C) describes the semantic constraints in word formation underlying *W* statistically and systematically.

Given any character string c_1c_2 ($c_1 \in C$, $c_2 \in C$), the following derivations hold for $LW(c_1c_2)$, the likelihood of this string being a word, according to properties of HMM and Bayes theorem:

$$\begin{aligned} LW(c_1c_2) &= \sum_{s_1s_2} p(s_1)p(s_2 \mid s_1)p(c_1 \mid s_1, s_2)p(c_2 \mid s_1, s_2) \\ &\approx \sum_{s_1s_2} p(s_1)p(s_2 \mid s_1)p(c_1 \mid s_1)p(c_2 \mid s_2) \\ &= \sum_{s_1s_2} p(s_1, s_2)\frac{p(s_1 \mid c_1)p(c_1)}{p(s_1)}\frac{p(s_2 \mid c_2)p(c_2)}{p(s_2)} \\ &= \sum_{s_1s_2} \frac{p(s_1, s_2)}{p(s_1)p(s_2)} p(s_1 \mid c_1)p(s_2 \mid c_2)p(c_1)p(c_2) \cdot \end{aligned} \tag{1}$$

where s_1s_2 ($s_1 \in S, s_2 \in S$) is any semantic tag sequence generated by c_1c_2 in a combinatorial way.

We ignore $p(c_1)$ and $p(c_2)$ in (1) in order to increase the generalization power of $LW(c_1c_2)$:

$$LW(c_1c_2) = \sum_{s_1s_2} \frac{p(s_1, s_2)}{p(s_1)p(s_2)} p(s_1 \mid c_1)p(s_2 \mid c_2) \cdot \tag{2}$$

For sake of clarity, let:

$$MI^*(s_1, s_2) = \frac{p(s_1, s_2)}{p(s_1)p(s_2)} \cdot \tag{3}$$

then an equivalent of formula (2), denoted $LW_{MI^*}(c_1c_2)$, is obtained consequently:

$$LW_{MI^*}(c_1c_2) = \sum_{s_1s_2} MI^*(s_1, s_2)p(s_1 \mid c_1)p(s_2 \mid c_2) \cdot \tag{4}$$

Note that $MI^*(s_1, s_2)$ is exactly the inner part of $MI(s_1, s_2)$:

$$MI(s_1,s_2)=\log_2\frac{p(s_1,s_2)}{p(s_1)p(s_2)}\cdot \tag{5}$$

We thus put forward a variation of formula (4), denoted $LW_{MI}(c_1c_2)$, as an alternative of the likelihood, though the derivation from formula 4 to 6 does not hold mathematically:

$$LW_{MI}(c_1c_2)=\sum_{s_1s_2}MI(s_1,s_2)p(s_1\mid c_1)p(s_2\mid c_2)\cdot \tag{6}$$

And, another alternative $LW_P(c_1c_2)$ is presented for the purpose of comparisons:

$$LW_P(c_1c_2)=\sum_{s_1s_2}p(s_1,s_2)p(s_1\mid c_1)p(s_2\mid c_2)\cdot \tag{7}$$

Now we have three alternatives for measuring the likelihood of c_1c_2 being a word: $LW_{MI^*}(c_1c_2)$, $LW_{MI}(c_1c_2)$ and $LW_P(c_1c_2)$. We shall choose the most appropriate one in Section 4.

3.2 Estimation of HMM Parameters

If we already have a manually annotated *WS*, the training of WF_{sem} will be easy. Unfortunately, we do not have it yet. In fact, we only have *C*, *T*, *S* and *W*. It is very tough to handcraft such a *WS* because the related linguistic study is poor, resulting in a lack of theoretical preparations necessary to do so. We have to seek for strategies to make some degree of approximations in parameter estimation. We try three schemes.

3.2.1 The Mean Scheme

For any word $w=c_1c_2\in W$, suppose c_i has n_i possible semantic tags { $s_{i,1},\ldots,s_{i,n_i}$ } according to *T* (*i*=1,2, $n_i\geq 1$):

$$\begin{array}{lll} w= & c_1 & c_2 \\ & s_{1,1} & s_{2,1} \\ & s_{1,2} & s_{2,2} \\ & \cdots\cdots\cdots\cdots & \\ & s_{1,n_1} & s_{2,n_2} \end{array}$$

The mean scheme will simply set:

$$p(s_{i,j}\mid c_i)=\frac{1}{n_i}\quad (i=1,2,j=1,\ldots,n_i). \tag{8}$$

Let *f(x)* and *f(x, y)* stand for the number of times *x* occurs and *xy* co-occurs over *W* respectively, then the contribution of semantic tag $s_{i,j}$ of character c_i of this *w* to the frequency counting of P_C would be:

$$f(s_{i,j}) = f(s_{i,j}) + \frac{1}{n_i} \quad (i=1,2). \tag{9}$$

and the contribution of semantic tag sequence $s_{1,j}s_{2,k}$ of this w to the frequency counting of P_S would be:

$$f(s_{1,j}, s_{2,k}) = f(s_{1,j}, s_{2,k}) + \frac{1}{n_1 n_2} \quad (j=1, \ldots, n_1, k=1,\ldots,n_2). \tag{10}$$

We shall obtain P_S and P_C after the above process has been done over all w in W.

3.2.2 The Bias Scheme

The bias scheme will apply the mean scheme first, and then adjust $p(s_{i,j} \mid c_i)$ by the resulting $f(s_{i,j})$:

$$p(s_{i,j} \mid c_i) = \frac{f(s_{i,j})}{\sum_j f(s_{i,j})}. \tag{11}$$

3.2.3 The Baum-Welch Re-estimation Scheme

Baum-Welch re-estimation algorithm is often used in unsupervised learning of HMM parameters [6]. The algorithm is re-paraphrased to fit the need here:

Step 1. Initialize P_S and P_C with the mean scheme.

Step 2. Apply Baum-Welch algorithm one pass through W based on P_S and P_C.

Step 3. Calculate new $P_S^{'}$ and $P_C^{'}$ according to the results of step 2.

Step 4. Let:

$$Q_W^{'} = \sum_{c_1c_2 \in W} \sum_{s_1s_2 \in c_1c_2} p^{'}(s_1, s_2) p^{'}(s_1 \mid c_1) p^{'}(s_2 \mid c_2)$$

$$Q_W = \sum_{c_1c_2 \in W} \sum_{s_1s_2 \in c_1c_2} p(s_1, s_2) p(s_1 \mid c_1) p(s_2 \mid c_2)$$

calculate:

$$\delta_Q = Q_W^{'} - Q_W$$

If $\delta_Q \le \delta_0$ then return $P_S^{'}$ and $P_C^{'}$ as the final solution;

else do $P_S \leftarrow P_S^{'}$, $P_C \leftarrow P_C^{'}$, go to step 2.

where δ_0 is the desired convergence limit to be determined experimentally.

3.3 Static Versus Dynamic Training

Static training refers to the strategy, as described in Section 3.2, that every word w in W is treated equally in estimating P_S and P_C, while dynamic training refers to another strategy that w in W is weighted by its frequency in a large corpus. In dynamic training, a frequent word in usage will be given a higher weight, and its corresponding semantic tag sequence will play more important role in word-formation. For example, the character '全' belongs to 'the state of fullness or partialness' with semantic tag 'Eb02' in *THSCS*. There exist only two words, '全国'(the whole country) and '全省'(the whole province), with semantic tag sequence 'Eb02+Di02', in *THW2*(Di02 for 'countries or administrative districts'), thus the importance of sequence 'Eb02+Di02' in word-formation is very low in static training. But these two words appear frequently in a corpus, indicating that the word-building ability of this sequence may be under-estimated. Obviously, its importance will be raised a lot in dynamic training.

All formulae in Section 3.2 still hold in dynamic training.

4 Experiments

A series of experiments are carried out to fix the factors of the framework proposed in Section 3. In static training, *THW2* is used as the training data. In dynamic training, all words in *THW2* are weighted by their string frequencies derived from *RCC*, a very huge raw corpus composed of about 1,000M Chinese characters. The open test is performed on *PDA98J*, a manually word-segmented corpus composed of the People Daily of January 1998 with about 1.3M Chinese characters, developed by the Institute of Computational Linguistics, Peking University: all distinct character bigrams excluding proper nouns in *PDA98J* are exhaustively collected, – in total, we obtain 238,946 such bigrams, among which 23,725 are two-character words. These 238,946 character bigrams form the test set, denoted *TS238946*, of experiments.

To better verify the effectiveness of our semantic-tag-based word extraction method, some typical methods based directly on characters rather than semantic tags are also tested in parallel for comparisons. *PDR9596*, a raw corpus composed of the People Daily of 1995 and 1996 with about 50M Chinese characters is used to train character bigrams on these occasions.

4.1 Determining the Most Appropriate $LW(c_1c_2)$

We need to decide which of $LW_{MI^*}(c_1c_2)$, $LW_{MI}(c_1c_2)$ and $LW_P(c_1c_2)$ is most appropriate for measuring the likelihood of a character string c_1c_2 being a word. Here, we use the Baum-Welch re-estimation scheme to estimate WF_{sem}, because the scheme sounds more refined than the other two, the mean and the bias (experimental results in Sections 4.2 and 4.3 will support this assumption). Then we compare the performance of $LW_{MI^*}(c_1c_2)$, $LW_{MI}(c_1c_2)$ and $LW_P(c_1c_2)$ in word extraction on *TS238946* in the context of static training. As shown in Fig. 2, $LW_{MI}(c_1c_2)$ is the best among the three, achieving a slightly better performance than $LW_{MI^*}(c_1c_2)$, though the latter is most rational mathematically. The performance of $LW_P(c_1c_2)$ is the worst, far away from that of $LW_{MI}(c_1c_2)$ and $LW_{MI^*}(c_1c_2)$. We therefore choose $LW_{MI}(c_1c_2)$.

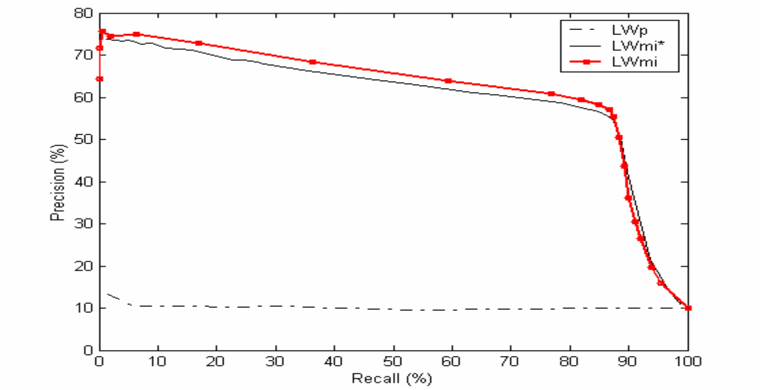

Fig. 2. Performance of $LW_{MI^*}(c_1c_2)$, $LW_{MI}(c_1c_2)$ and $LW_P(c_1c_2)$ in word extraction

4.2 Performance Comparisons Among Various Methods

We experimented with seven candidate methods carefully designed in various settings (the former four are semantic-tag-based, and the latter three are character-based):

- SMean: $LW_{MI}(c_1c_2)$, the mean scheme, static training;
- SBias: $LW_{MI}(c_1c_2)$, the bias scheme, static training;
- SBW: $LW_{MI}(c_1c_2)$, the Baum-Welch re-estimation scheme, static training;
- SDBW: $LW_{MI}(c_1c_2)$, the Baum-Welch re-estimation scheme, dynamic training;
- CP: $p(c_1, c_2)$;
- CMI: $mi(c_1, c_2)$;
- CLL: $\log - likelihood(c_1, c_2)$

Experimental results are given in Fig.3 and Table 2.

The following comparisons can be made based on experimental results from three perspectives:

(1) Comparison among the three schemes for parameter estimation of HMM:

The highest F-measure of SBias, SMean and SBW is 45.0% (at 50.0% recall), 62.0% (at 70.0% recall) and 68.0% (at 80.0% recall), and the 10-point average F-measure is 33.5%, 44.9% and 46.2%, respectively. The fact that SBW increases about 23.0% in the highest F-measure and 12.7% in the average F-measure compared to SBias indicates that the impact of the scheme of HMM parameter estimation on word extraction is obvious. In addition, it is a bit surprise that SBias is much weaker than SMean.

(2) Comparison between static and dynamic training:

The highest F-measure of SDBW is 68.5% (at 80.0% recall), and its 10-point average F-measure is 47.9%. SDBW increases about 0.5% in the highest F-measure and 1.7% in the average F-measure compared to SBW.

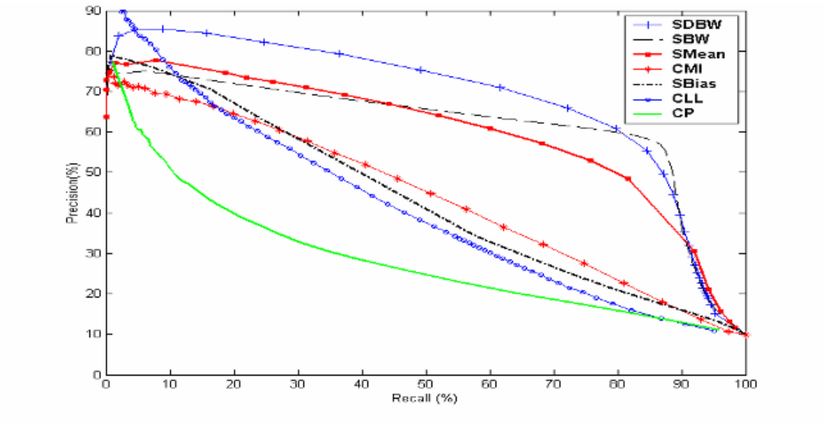

Fig. 3. Performance of various semantic-tag-based and character-based methods

Table 2. 10-point F-measure of various semantic-tag-based and character-based methods

Recall(%)	F-Measure(%)						
	CP	CLL	CMI	SBias	SMean	SBW	SDBW
10	16.7	17.5	17.5	17.5	17.5	17.5	18.0
20	26.6	30.4	30.5	30.5	31.0	30.5	32.0
30	31.5	38.8	39.8	39.0	42.0	40.5	43.0
40	33.3	42.8	45.3	44.0	50.5	50.0	52.5
50	33.0	43.0	47.5	45.0	57.0	56.0	60.0
60	31.5	40.1	46.7	42.3	60.5	62.0	65.0
70	29.4	34.8	43.0	38.5	62.0	65.0	68.0
80	26.5	28.4	36.2	33.0	61.5	68.0	68.5
90	22.5	22.1	26.6	27.2	48.5	54.0	54.0
100	18.1	18.1	18.1	18.1	18.1	18.1	18.1
Average	26.9	31.6	35.1	33.5	44.9	46.2	47.9

Observe a word candidate '全州'(the whole state): its semantic-tag sequence is 'Eb02+Di02'. 'Eb02' is productive in word-formation, resulting in that LW_{MI}(全州) is quite low (-3.26) and therefore rejected to be a word by SBW. However, as stated in Section 3.3, the sequence 'Eb02+Di02' is frequent in dynamic training (because of the presence of '全国' and '全省' in *THW2*), leading to an increasing of LW_{MI}(全州) to 1.04, – '全州' is thus successfully recognized by SDBW.

(3) Comparison between semantic-tag-based and character-based approaches:

The highest F-measure of CP, CLL and CMI is 33.3% (at 40.0% recall), 43.0% (at 50.0% recall) and 47.5% (at 50.0% recall), and the 10-point average F-measure is 26.9%, 31.6% and 35.1%, respectively. CMI outperforms the other two in character-based approaches. Further notice that the performance is improved very significantly as we move from CMI to SDBW: SDBW increases about 21.0% in the highest F-measure and 12.8% in the average F-measure compared to CMI!

Recall the word candidate ‘俄军’ (Russian army) in Section 1: ‘俄军’ occurs only 3 times in *PDR9596*, while its involved characters ‘俄’ and ‘军’ occurs pretty frequently, making CMI(俄军) under 1.00 and rejected to be a word by CMI. In this case, CMI in fact suffers from the data sparseness problem. Our semantic-tag-based approach can resolve this problem in some degree: there exist a number of words with the same semantic-tag sequence ‘country+army’ in *THW2*, such as ‘美军’(American army), ‘日军’(Japanese army) and ‘苏军’(Soviet army), and those words occur in the corpus quite often, – as a consequence, LW_{MI}(俄军) raises to 4.24 while using SDBW, and ‘俄军’ is accepted as a word.

Summarizing the experimental results, SDBW and SBW outperforms all the other five methods, and SDBW is the best among the all.

4.3 Further Observations on the Baum-Welch Re-estimation Scheme

As said in Section 4.2, both SDBW and SBW explore the Baum-Welch re-estimation scheme to acquire more adequate HMM parameters. Let’s have a more detailed look at it.

One look is that the scheme converges after 95 times iteration.

Another look is about why the scheme is quite effective? We tend to partially answer this question from the angle of sense tagging, under an assumption that strong ability in sense disambiguation may lead to good performance in measuring word likelihood. Similar to part-of-speech tagging, we apply Viterbi algorithm to any word $w = c_1c_2$, finding the most likely semantic-tag sequence $s_1^{'}s_2^{'}$ for it, according to the HMM obtained from the Baum-Welch re-estimation scheme:

$$\begin{aligned} s_1^{'}s_2^{'} &= \arg\max_{s_1s_2}\ p(s_1s_2 \mid c_1c_2) \\ &= \arg\max_{s_1s_2}\ \frac{p(s_1, s2)}{p(c_1c_2)} p(c_1c_2 \mid s_1s_2) \\ &= \arg\max_{s_1s_2}\ p(s_1, s_2)p(c_1c_2 \mid s_1s_2) \\ &\approx \arg\max_{s_1s_2}\ p(s_1, s_2)p(c_1 \mid s_1)p(c_2 \mid s_2) \\ &= \arg\max_{s_1s_2}\ p(s_1, s_2)\frac{p(s_1 \mid c_1)p(c_1)}{p(s_1)}\frac{p(s_2 \mid c_2)p(c_2)}{p(s_2)} \\ &= \arg\max_{s_1s_2}\ \frac{p(s_1, s_2)}{p(s_1)p(s_2)} p(s_1 \mid c_1)p(s_2 \mid c_2) \\ &= \arg\max_{s_1s_2}\ MI^{*}(s_1, s_2)p(s_1 \mid c_1)p(s_2 \mid c_2)\ . \end{aligned} \tag{12}$$

Note that the inner parts of formulae (4) and (12) are identical.

We randomly extract 2,027 two-character words from *THW2*, and manually annotate those words with a unique semantic-tag sequence each, constituting the test set of the sense tagging experiment. In the test set, there are totally 4,054 characters, out of them, 3,054 are polysemous. The accuracy of sense tagging is defined as:

$$Accuracy\,1 = \frac{number - of - characters - correctly - tagged}{total - number - of - characters}$$

$$Accuracy\,2 = \frac{number - of - polysemous - characters - correctly - tagged}{total - number - of - polysemous - characters}$$

We take SBias as a baseline of comparison. SBias and SBW will correspond to two classical computational models in part-of-speech tagging, i.e., the unigram model and the bigram model, if we relate sense tagging to part-of-speech tagging. The results are listed in Table 3.

Table 3. SBW and SBias in sense tagging

	Number	SBias	SBW
	Total number of characters	4054	4054
1	Number of correctly tagged characters	1999	2395
	Accuracy1 (%)	49.3	59.1
	Total number of polysemous characters	3054	3054
2	Number of correctly tagged polysemous characters	999	1395
	Accuracy2 (%)	32.7	45.7

The disambiguation ability of SBW is more powerful than that of SBias. This may provide an evidence of why the word extraction performance of the former is much better than the latter. The results also indicate that the difficulty of sense tagging would be larger than that of part-of-speech tagging in Chinese: the bigram models usually achieve over 90% accuracy in part-of-speech tagging, if counted on the total number of words in texts, whereas SBW here can only achieve 59.1% accuracy in sense tagging.

5 Conclusions

This paper presents a semantic-tag-based approach to automatic extraction of two-character words of Chinese. The key feature of this approach is that it tries to capture Chinese word-formation using semantic constraints between characters in words, mainly based on a thesaurus of Chinese characters and a Chinese lexicon. The Baum-Welch re-estimation scheme is used to train parameters of semantic HMM in the way of unsupervised learning. No literature has reported on the similar work so far. The large-scale experiments demonstrate that the proposed method is effective: compared to the character-based methods, the F-measure of SDBW and SBW increases over 20.0%.

Further work will concern some unsolved issues. One issue is on how to minimize the possible negative effect of semantic-tag-based approach. For instance, we use SDBW and CMI to extract 30,000 words out of *TS238946* respectively. SDBW can recognize 18,568 words and CMI recognize 9,671 words successfully. CMI covers 46.4% of what SDBW has correctly recognized and SDBW covers 89.2% of what CMI has correctly recognized, but SDBW fails to correctly recognize 10.8% of what CMI has correctly recognized. Another issue is on how to expand the method to the task of extracting multiple-character words.

Acknowledgements. This research is supported by the National Natural Science Foundation of China under grant number 60321002 and the National 863 Project of China under grant number 2001AA114210－03.

References

1. Calzolari, N., Bindi, R.: Acquision of Lexical Information from a Large Textual Italian Corpus. In: Proc. of COLING'90, Helsinki, Finland, 1990, 54-59.
2. Chien, L.F.: PAT-tree-based Adaptive Keyphrase Extraction for Intelligent Chinese Information Retrieval. Information Processing and Management, special issue: Information Retrieval with Asian Language, 1998.
3. Daille, B.: Study and Implementation of Combined Techniques Automatic Extraction of Terminology. In: Proc. of the Balancing Act Workshop at 32nd Annual Meeting of the ACL, 1994, 29-36
4. Dunning, T.: Accurate Method for the Statistics of Surprise and Coincidence. Computational Linguistics, 1993, vol 19 no. 1 61-75.
5. Gelbukh, A., Sidorov, G.: Approach to Construction of Automatic Morphological Analysis Systems for Inflective Languages with Little Effort. Lecture Notes in Computer Science, N 2588, Springer-Verlag, (2003) 215-220.
6. Hajic, J.: HMM Parameters Estimation: The Baum-Welch Algorithm. www.cs.jhu.edu/~hajic, 2000.
7. Johansson, C.: Good Bigrams. In: Proc. of COLING'96. Copenhagen, Denmark, 1996
8. Mei, J.J.: Tong Yi Ci Ci Lin. Shanghai Cishu Press, 1983.
9. Merkel, M., Andersson, M.: Knowledge-lite Extraction of Multi-word Units with Language Filters and Entropy Thresholds. In: Proc. of RIAO'2000, Paris, France, 2000, 737-746.
10. Nie, J.Y., Hannan, M.L., Jin, W.: Unknown Word Detection and Segmentation of Chinese Using Statistical and Heuristic Knowledge. Communications of COLIPS, 199, vol 5 47-57
11. Sornlertlamvanich, V., Potipiti, T., Charoenporn, T.: Automatic Corpus-based Thai Word Extraction with the C4.5 Learning Algorithm. In: Proc. of COLING'2000, Saarbrucken, Germany, 2000, 802-807.
12. Sun, M.S., Shen, D.Y., Huang, C.N.: CSeg&Tag1.0: A Practical Word Segmenter and POS Tagger for Chinese Texts. In: Proc. of the 5th Int'l Conference on Applied Natural Language Processing, Washington DC, USA, 1997, 119-126.