Incorporate Web Search Technology to Solve Out-of-Vocabulary Words in Chinese Word Segmentation *

Wei Qiao and Maosong Sun

State Key Laboratory of Intelligent Technology and Systems,
Tsinghua National Laboratory for Information Science and Technology,
Department of Computer Sci. & Tech., Tsinghua University,Beijing, 100084, China
qiaow04@mails.tsinghua.edu.cn, sms@mail.tsinghua.edu.cn

Abstract. Chinese word segmentation (CWS) is the fundamental technology for many NLP-related applications. It is reported that more than 60% of segmentation errors is caused by the out-of-vocabulary (OOV) words. Recent studies in CWS show that, statistical machine learning method is, to some extent, effective on solving OOV words. But labeled data is limited in size and unbalanced in content which makes it impossible to obtain all the required knowledge to recognize OOV words. In this paper, large scaled web data is incorporated as knowledge supplement. A framework which combines using web search technology and machine learning method is proposed. For each sentence, basic segmentation is performed using linear-chain Conditional Random Fields (CRF) model. Substrings which CRF model gives low confidence decisions are extracted and sent to search engine to perform web search based word segmentation. Final decision is made by considering both CRF model based segmentation result and that of web search based result. Evaluations are conducted on SIGHAN Bakeoff 2005 and 2006 datasets, showing the effectiveness of the proposed framework on dealing with OOV words.

Keywords: Chinese word segmentation, Conditional Random Fields, Machine learning, Out-of-Vocabulary words, Web data.

1 Introduction

Chinese word segmentation plays an important role in many Chinese language processing tasks. In the past decade it has drawn a large body of research in Chinese language processing community. A variety of methods have been exploited ranging from rule-based (Palmer, 1997; Cheng *et al.*, 1999) to statistics-based (Sproat *et al.*, 1996), word-based (Sun *et al.*, 1998) to character-based (Xue, 2003), supervised learning-based (Peng *et al.*, 2004; Low *et al.*, 2005) to unsupervised learning-based (Goldwater *et al.*, 2006; Zhao and Kit, 2008), as well as their hybrid (Gao *et al.*, 2005). It is reported in SIGHAN Bakeoff-2005 (Emerson, 2005) and SIGHAN Bakeoff-2006 (Levow, 2006) that the highest F1-measure achieved on open tracks is 97.9% while the OOV recall rate is only 84%. This performance is achieved on the test sets of which OOV rates only ranging from 2% to 8%. When facing Chinese running text with much higher OOV rate, the performance will drop dramatically. It is reported that performance loss caused by out-of-vocabulary (OOV) words is at least five times greater than that of segmentation ambiguities (Huang and Zhao, 2007). So, OOV problem is the main factor which extremely influences the performance of CWS system and there still has some room to improve.

* This work is supported by the National Science Foundation of China under Grant No. 60873174, the National 863 High-Tech Project of China under Grant No. 2007AA01Z148 and the Tsinghua-Sogou Joint Research Center for Search Engine Technologies.

Recent studies in CWS focus on statistic machine learning methods. Regarding CWS task as sequence labeling problem (Xue, 2003; Goh *et al.*, 2005), various machine learning methods can be adopted to do this task. Features derived from labeled corpora are taken to train the model. The performance of this kind of method much depends on the size and the quality of the training data. As labeled corpus is usually limited in size and unbalanced in content, it can not provide enough knowledge to train a model which is robust enough when facing large scaled running text which contains large majority of OOV words.

Nowadays the number of web pages grows very fast. The web text can be considered as a very large scaled knowledge database which seldom has OOV problem. So, one way that can supplement the knowledge is to incorporate web knowledge database. There already have some works which are motivated by this idea. The most related one is (Wang *et al.*, 2007), they proposed a search-based CWS method which is entirely unsupervised. They perform word segmentation as a search procedure by using search engine to directly find answer on web. First, sub-sentences are extracted from sentences using punctuation as delimiters. Second, these sub-sentences are directly sent to search engine as user queries. At last, the highlight parts in the returned snippets are used to construct the final word segmentation. Experimental result shows performance improvement on OOV recall rate but the reported F-measure is only about 87% which is much worse than supervised machine learning method. Motivated by taking both advantages of web-search method and supervised machine learning method, a new framework combines using web search and CRF model is proposed. For every sentence, segmentation candidates are collected and organized as lattice. Instead of sending sub-sentences as queries, specific small segments derived from the lattice are sent to the search engine. Search based segmentation is constructed using the highlighted parts of returned snippets. Final decision is made by measuring the distance of the search-based segmentation with the CRF segmentation candidates.

The rest of the paper is organized as follows. We introduce our specific implementation of linear-chain CRF model based word segmenter in Section 2. In Section 3, we propose the new segmentation framework which combines using search technology and supervised machine learning method. Experimental results are given in Section 4 and in Section 5, we conclude our work.

2 CRF-based Chinese word segmenter

Recent studies show that linear-chain structured CRF model(Lafferty *et al.*, 2001), which was first applied to CWS task in the year 2004 (Peng *et al.*, 2004), has been proved to be the most effective one for sequence labeling problem. In this paper, CRF-based word segmenter is selected as our basic word segmenter. In subsection 2.1, we introduce the specific implementation of our CRF-based Chinese word segmenter. Error analysis and performance evaluation is done in subsection 2.2 and 2.3 separately.

2.1 Implementation of CRF-based word segmenter

In this paper, the specific implementation of CRF-based word segmenter uses the CRF++ toolkit version 0.53 provided by Taku Kudo[1]. Four tags, denoted as S(single-character word), L(the most left character of a word), M(middle character of a word) and R(the most right character of a word), are used to distinguish the position of a character in a word. The window size is set as five to extract features to train the model. This means when we consider current character, the adjacent four characters (the two ahead of it and the two after it) are taken as local features. The basic feature template adopted from (Low *et al.*, 2005) is used, here we restate them to make the paper self-contained:

(a) $C_n, n = -2, -1, 0, 1, 2$

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

(b) $C_nC_{n+1}, n = -2, -1, 0, 1$

(c) $C_{-1}C_1$

(d) $Pu(C_0)$

(e) $T(C_{-2})T(C_{-1})T(C_0)T(C_1)T(C_2)$

Where C_n refers to a Chinese character, here, n indicates the relative distance to current character C_0. For example, C_1 indicates the character next to C_0 while C_1 refers to the character previous to C_0. $Pu(C_0)$ represents whether current character is a punctuation. $T(C_n)$ represents what type the character C_n belongs to. Here, four types are defined as Numbers, Dates (the Chinese characters for "day", "month", "year", respectively), English letters and Others. See more detailed illustration in (Low *et al.*, 2005).

2.2 Error analysis

The second International Chinese Word Segmentation Bakeoff (SIGHAN-bakeoff 2005) provides four datasets for Chinese word segmentation competition. Every set has training set and corresponding test set in it. In this paper, the one constructed by Microsoft Research center in Asia, denoted as MSRA05, is used to do error analysis.

A CRF-based word segmenter is trained on the training set of MSRA05. Segmentation is performed on the corresponding test set. Compared with gold standard, totally 2,908 segmentation errors are found. They are manually classified into four groups according to their error types: the segmentation error caused by OOV words fall into type A; those caused by ambiguity problem are classified into type B; for strings whose segmentation way provided by gold-standard and by CRF-based segmenter are both acceptable, fell under type C; errors caused by inaccurate Gold-standard fall into type D. Table 1 shows the error distribution.

Table 1: Error distribution of CRF-based word segmenter tested on MSRA05.

Error Type	A	B	C	D
# Errors	1,581	897	357	73
Percentage	54.4%	30.8%	12.3%	2.5%

From Table 1 we can see that, when using CRF-based word segmenter, OOV words causes over 54% segmentation errors. So, how to appropriately process OOV words is the key point to improve the entire performance.

2.3 The improvement potential when consider 10-best segmentation candidates

The conclusion obtained above is under the case that the one with the highest probability is selected as final segmentation result. What if we consider more candidates? Is there any possibility that the best answer be ranked behind or, in other words, with lower probability? In order to answer these questions, top 10 candidates are recorded, according to gold standard the best one is chosen from 10 candidates as final segmentation result. This constructs the upper bound of our improvement.

Word segmentation is performed on all the four datasets provided by the SIGHAN-bakeoff 2005, denoted as MSRA05, PKU05, CITYU05 and AS05 respectively[2]. The strategy using 10-best candidates are compared with the one-best strategy. Table 2 shows the comparison results which shows the improvement potential.

[2] http://sighan.cs.uchicago.edu/bakeoff2005/

Table 2: Improvement potential testing on SIGHAN-bakeoff 2005 dataset.

	OOV Rate	R-OOV 1 best	R-OOV 10 best	Improvement on R-OOV	F1 1 best	F1 10 best	Improvement on F1
MSRA05	2.6	75.6	89.1	13.5	96.6	98.9	2.3
PKU05	5.8	76.8	84.1	7.3	94.4	96.4	2.0
CITYU05	7.4	78.5	91.1	12.6	95.4	98.4	3.0
AS05	4.3	71.3	80.2	8.9	95.0	97.1	2.1

From Table 2 we can see that the improvement on F1-measure is about 2% while it is ranging from 7.3% to 13.5% on OOV recall. The improvement is statistical significant, thus is worthy of further investigation.

Experiment is done to see how many sentences have the case that the final selected segmentation is not the one with the highest probability. Figure 1 shows the distribution. The horizontal axis represents the Nth (N=0,. . . ,9) candidate while the vertical axis represents the number of sentences whose segmentation are provided by the Nth candidate:

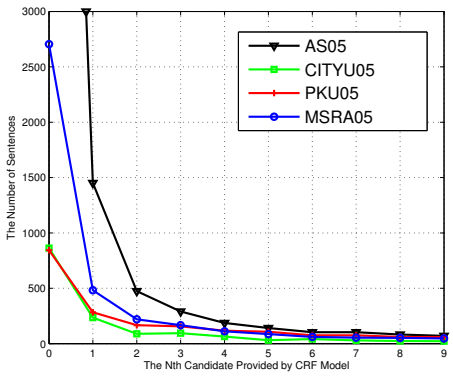

Figure 1: Distribution of N best results.

From Figure 1 we can see, there are still some exceptions that the best answer falls into the candidates with relatively lower confidence. And the amount of this kind of cases can't be ignored.

Experiment result shows that considering more candidates (20-best or more) doesn't provide significant improvement. So, we consider maximal top 10 candidates. We will illustrate how to decide the exact number of candidates automatically according to different cases of sentences.

3 The proposed framework

In this section we propose the new framework for Chinese word segmentation. The basic idea is mining information from web to perform web search based word segmentation. By using search based segmentation result, we re-rank the candidates provided by basic word segmenter.

What we want to benefit from web is it seldom has OOV problem, and OOV words are often given low confidence by CRF model. This motivates us to distinguish high confidence segmentation and low confidence segmentation. For substrings in a sentence, the segmentation way will be adopted if CRF model provides high confidence on them. Otherwise they will be recorded for further processing. In order to do this task, lattice is constructed based on segmentation candidates. Substrings with lower confidence can be easily extracted from the lattice and sent to search engine as queries. Search-based segmentation is performed on these substrings. Final segmentation is reconstructed through similarity measurement between the search-based segmentation and the

candidates. Figure 2 shows the whole flow chart of the proposed framework. It consists of three modules which will be introduced one by one in the following subsections.

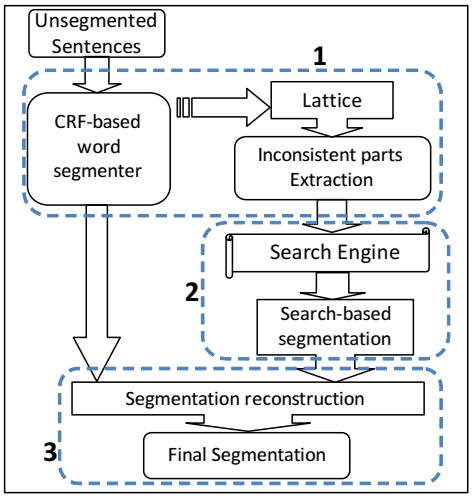

Figure 2: The flow chart of new proposed framework.

3.1 Module 1: Lattice construction and low confidence substrings extraction

In this subsection, in order to extract low confidence parts, lattice is constructed according to segmentation candidates provided by CRF model. This corresponds to module 1 in the flow chart shown in Figure 2.

Here we give out the formal description of the construction process of lattice:

Given a Chinese string S to be segmented, with length l, then there exists $l+1$ segmentation positions denoted as $p_0, p_1, .., p_l$. Each specific segmentation way s for S can be represented by a sequence of positions $\{p_{s_0}, p_{s_1}, ..., p_{s_l}\}$ which satisfies:

(a) $s_0 = 0, s_l = l$

(b) $s_i \in \{0, 1, ..., l\}$

(c) $s_i < s_{i+1}$

Also s can be viewed as a total order, and different segmentations can be combined into a partial order relationship, with p_0 as source and p_l as destination. Take Figure 3 for example, the position 0 is the source and 13 is the destination. If a segmentation position p_i is unanimous point, then in the partial order they define, unanimous positions can be viewed as joint nodes, just like the position 1, 8 and 10 which are emphasized in Figure 3.

A sequence of unanimous positions can be retrieved from the partial order, recorded as p_u. If the sub graph between two positions in p_u , i.e., p_{u_i} and p_{u_j}, is inherently a total order, then the substring between p_{u_i} and p_{u_j} contains unanimous segmentations. For example, substring "前(ex-)" between position 0 and 1, "担任(take on)" between position 8 and 10 and "模特儿(model)" between position 10 and 13 in Figure 3. If not, then there exists several possible segmentations and the substring defined by the pair of positions, such as "港姐嘉碧仪应邀(Miss HongKong Biyi Jia accepts the invitation)" defined by the pair of positions 1 and 8 in Figure 3, will be delivered for further diagnosis by search engines.

By now the low confidence strings are extracted which compose a set denoted as S_u. In subsection 3.2 we will describe the procedure of segmentation reconstruction for strings in S_u.

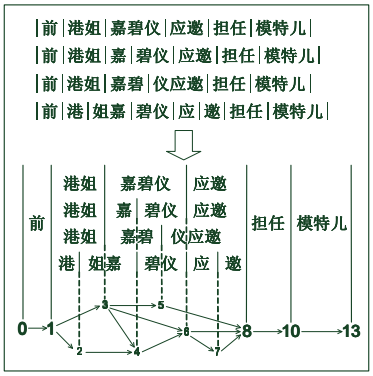

Figure 3: The illustration of Lattice construction.

3.2 Module 2: Search-based segmentation for low confidence substrings

The search-based segmentation in this paper is an unsupervised one. The idea is motivated by the work of (Wang *et al.*, 2007) while the method is a very different one. Briefly, the segmentation is implemented by using the highlighted parts in the snippets returned by search engine. Specifically in this paper, we use Sogou [3] search engine to do this task. The detailed implementation of our search-based segmentation can be divided into two parts. In the followings, we will introduce them one by one.

The first part is segments collection.

String in S_u are automatically submitted to search engine as user queries. The returned snippets are collected for further processing. Figure 4 shows an example of the returned snippet when using string "港姐嘉碧仪应邀(Miss HongKong Biyi Jia accepts invitation)" as query. Every highlighted (red) part in the snippet is said as a segment.

Figure 4: The search result of "港姐嘉碧仪应邀(Miss HongKong Biyi Jia accepts the invitation)" using Sogou search engine.

For each substring, in total one hundred snippets are used to do segments collecting and at meanwhile, the frequency of each segment is recorded. We then rank the segments in descending order of their frequency and organize the data in the form of shown in Figure 5.

The second part is segmentation reconstructed.

For every substring, we iteratively select the segment with currently the highest frequency as a segmentation unit, and tag the corresponding characters in the original string until all the characters in the substring is tagged. For example in Figure5, the segment "港姐 (Miss HongKong)" will first be selected as a segmentation unit. Thus the two characters "港姐(Miss HongKong)" in substring "港姐嘉碧仪应邀(Miss HongKong Biyi Jia accepts the invitation)" will be tagged as L and R respectively.

One would suspect of using the local segmenter of search engine. Here, we argue that although search engines generally have their own local segmenters, the returned highlighted parts which we

[3] Sogou is a Chinese search engine which is owned by Sohu, Inc., and is one of the fastest growing search engines in China.

Segments of the string	Frequency
港姐 (Miss HongKong)	30
嘉碧仪 (Biyi Jia)	23
应邀 (Accept invitation)	8
港姐应邀 (Miss HongKong accepts invitation)	7

Figure 5: Segments of query string: "港姐嘉碧仪应邀"(Miss HongKong Biyi Jia accepts the invitation).

use are quite different from that generated by the segmenters. We have investigated the segmentation strategy of Sogou search engine, it is a "reduplicate" one. In other words, they make redundant segmentation. Take segmentation result of "港姐嘉碧仪应邀(Miss HongKong Biyi Jia accepts the invitation)" as an example, by checking the HTML source code, we could find that both "港姐嘉碧仪(Miss HongKong Biyi Jia)", "港姐(Miss HongKong)" and "嘉碧仪(Biyi Jia)" are taken as segmentation units. So, our search-based segmentation results are generally independent to the local segmenters of the search engine.

3.3 Module 3: Segmentation reconstruction

Now, we have search-based segmentation result and the segmentation candidates. There are two ways to reconstruct the final segmentation:

1. For low confidence part, directly take place the segmentation provided by basic word segmenter by the search-based segmentation result, thus we can reconstruct segmentation result;

2. After we finish 1, do similarity measurement between reconstructed one and the candidates. The candidate which has highest score will be taken as final segmentation.

In experiment part, we will compare the two strategies (with and without similarity measurement).

In the following, we introduce the similarity measurement method used in this paper. Inspired by Edit distance, Segmentation Distance (SD) is proposed here to measure the similarity between two segmented strings: For two segmented strings S_1 and S_2 the segmentation distance is defined as the minimum number of boundary insertions and boundary deletions to transform one segmentation way to the other which can be represented as:

$SD(S_1, S_2) = \min\{\Sigma(Insertion(S_1 \rightarrow S_2) + Deletion(S_1 \rightarrow S_2))\}$

Dynamic programming algorithm is used to calculate SD value.

4 Experimental results

In experiment part, we firstly determine the number of candidates and rules for lattice construction. Then, performance evaluation for the proposed framework will be performed on five corpora provided by SIGHAN-bakeoff 2005 (Emerson, 2005) and SIGHAN-bakeoff 2006 (Levow, 2006). Statistics of the five datasets[4] is listed in Table 3. The performance is evaluated by F1-Measure and OOV recall rate (R_OOV).

4.1 Determine the number of candidates for lattice construction

If all the 10-best candidates are used to construct lattice, there will be many small segments in S_u (see section 3.1) which only contains one character. In order to bring down the complexity of lattice, experiment is done to determine the number of candidates we adopted to construct the

[4] Here, PKU05 dataset is not included the representation of personal names is different from others: family name and given name are segmented as two words

Table 3: Statistics of five datasets of SIGHAN-bakeoff 2005 and 2006.

Corpora	Encoding	Training(MB)	Test(KB)	OOV Rate(%)
MSRA05	GB	2.37	107	2.6
AS05	BIG5	5.45	122	4.3
CITYU05	BIG5	1.46	41	7.4
MSRA06	GB	1.26	100	3.4
CTB06	BIG5	0.5	154	8.8

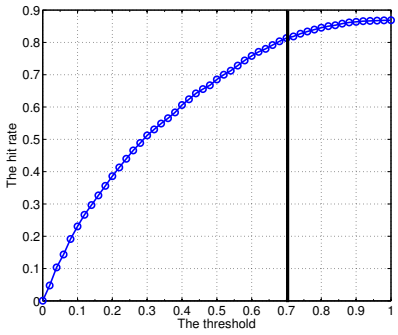

Figure 6: The relation of the threshold and the hit rate

lattice. Candidates are selected out one by one according to the priority of high probability until the accumulate probability value reaches the threshold. During this process, we aim to have hit rate as high as possible. Here "hit rate" is defined as: The number of gold-standard segmentation candidates which are selected out divided by the total number of candidates we selected out.

From Figure 6 we can see, distinguished by the threshold value of "0.7", the curve grows steeply before that while it turns flat after that. The hit rate reaches 80% at 0.7. In the latter part of this paper we take 0.7 as the threshold to determine number of candidates we adopted.

4.2 Performance Evaluation

Firstly, we compare the performance between baseline word segmenter (CRF-based word segmenter) and our proposed framework including with and without similarity measurement. The comparison result is given in Table 4. It shows that, compared with baseline, the proposed scheme without similarity measurement achieves improvement on both OOV recall rate (3.2% to 8.4%) and F1 (0.4% to 1.9%). Further more, the strategy with similarity measurement shows even better performance on OOV recall rate.

Table 4: Performance comparison with baseline system (%).

Corpora	R-OOV CRF	F1 CRF	R-OOV CRF+Search	F1 CRF+Search	R-OOV CRF+Search +similarity	F1-Measure CRF+Search +similarity
MSRA05	75.6	96.6	79.6	97.0	80.9	97.1
AS05	71.3	95.0	75.2	95.5	75.8	95.8
CITYU05	78.5	95.4	81.7	96.0	82.6	96.5
MSRA06	67.3	95.3	75.7	97.2	76.5	97.3
CTB06	71.2	93.0	78.3	94.6	79.5	94.7

Secondly, the best performance we achieved is taken to do comparison with the best reported result of SIGHAN. Table 5 gives out the result. We can see, in most cases our proposed scheme

achieves improvement on R_OOV. Since OOV rate of SIGHAN datasets ranging only from 2.6% to 8.8%, although R_OOV is significantly improved the F-measure does not improve much.

Table 5: Compare with the best reported results (%).

Corpora	Participant	R-OOV Open best	F1-Measure Open best	R-OOV CRF+Search +similarity	F1-Measure CRF+Search +similarity
MSRA05	Wei Jiang	59.0	**97.2**	**80.9**	97.1
	Hwee Tou Ng	73.6	96.8	**80.9**	**97.1**
AS05	Hwee Tou Ng	68.4	**95.6**	**75.8**	**95.8**
	Yaoyong Li	68.6	94.8	**75.8**	**95.8**
CITYU05	Hwee Tou Ng	80.6	**96.2**	**82.6**	**96.5**
MSRA06	France Telecom	**83.9**	**97.9**	76.5	97.3
	France Telecom	**84.0**	**97.7**	76.5	97.3
CTB06	Univ. Texas Austin	76.8	94.4	**79.5**	**94.7**

Here, we manually construct a small test set (denoted as C_Web) which is extracted from web pages (about 100 pages). It includes various topics such as Sports, Medical science, Mechanical, etc., with 4,000 words in it. MSRA05 training set is taken to train CRF model. Table 6 gives out the performance comparison between baseline and our proposed method. It shows, for high OOV test set, that the proposed method achieves significant improvement.

Table 6: Performance comparison with baseline system.

Corpora	OOV rate(%)	R-OOV CRF (%)	F1 CRF (%)	R-OOV CRF+Search +similarity (%)	F1 CRF+Search +similarity (%)
C_Web	21.5	74.8	92.6	90.3	97.2

We investigate the segmentation result and select some typical sentences to do test. Table 7 shows three sentences which are wrongly (the underlined parts) segmented by using ICTCLAS1.0[5] and MSRSeg1.0[6]. It shows that our proposed scheme is effective on particular OOV types such as new words ("万人迷"(Mac daddy star)), loanwords ("的士高"(Disco)) and name entity ("嘉碧仪"(Biyi Jia)) while well known high-quality word segmenter, such as MSRSeg1.0 and ICTCLAS1.0 fail on processing these kinds of OOV words.

Table 7: Three typical segmentation errors derived from MSRSeg1.0 and ICTCLAS1.0

Input sentences	Output of ICTCLAS1.0 and MSRSeg1.0
"万人迷再爆桃色" (Mac daddy star burst sex scandal again)	万 人 迷 再 爆 桃色 万 人 迷 再 爆 桃色
"港姐嘉碧仪应邀担任模特儿" (Miss HongKong Jiabi Yi accepts the invitation of being a model)	港 姐 嘉碧仪 应邀 担任 模特 儿 港 姐 嘉 碧 仪 应邀 担任 模特儿
"我们来到的士高劲歌" (We go to disco to sing songs)	我们 来到 的士 高 劲 歌 我们 来到 的 士 高劲歌

[5] ICTCLAS 1.0: http://www.nlp.org.cn
[6] MSRSeg.v1.:http://research.microsoft.com/-S-MSRSeg

5 Conclusion

Within this paper, a framework which combines supervised machine learning method and web search technology to do Chinese word segmentation is proposed. Experimental result shows that the proposed framework obtains improved segmentation performance and is especially effective on processing OOV words. There are still some future works left: first, we can construct a local search engine instead of using commercial ones. Second, well defined rules should be concluded to help us to reconstruct the search-based word segmentation.

References

Cheng, K.S., G.H. Young and K.F. Wong. 1999. A study on word-based and integral-bit Chinese text compression algorithms. *Journal of the American Society for Info. Sci.*, 50(3), 218-228.

Emerson, T. 2005. The second international Chinese word segmentation bakeoff. *Proceedings of the 4th SIGHAN Workshop*, pp.123-133.

Gao, J.F., M. Li, A. Wu and Chang-Ning Huang. 2005. Chinese word segmentation and named entity recognition: A pragmatic approach. *Computational Linguistics*, 31(4), 531-574.

Goh, C.L., Masayuku Asahara and Yuji Matsumoto. 2005. Chinese Word Segmentatin by Classification of Characters. *Computational Linguistics*, 10(3), 381-396.

Goldwater, S., T.L. Griffiths and M. Johnson. 2006. Contextual Dependencies in Unsupervised Word Segmentation. *Proceedings of COLING-ACL 2006* , pp.673-680.

Huang, C.N. and H. Zhao. 2007. Chinese Word Segmentation: A Decade Review *Journal of Chinese Information Processing*, 21(3), 8-20.

Lafferty, J., A.McCallum and F.Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proceedings of 18th International Conference on Machine Learning (ICML2001)*, pp.282-289.

Levow, G. 2006. The third international Chinese word segmentation bakeoff. *Proceedings of the 5th SIGHAN Workshop*, pp.108-117.

Low, J.K., Hwee Tou Ng and W.Y. Guo. 2005. A maximum entropy approach to Chinese word segmentation. *Proceedings of the 4th SIGHAN Workshop*, pp.161-164.

Palmer, David D. 1997. A Trainable Rule-based Algorithm for Word Segmentation. *Proceedings of Annual Meeting of the Association for Computational Linguistics 1997*, pp.321-328.

Peng, F., Fangfang Feng and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. *Proceedings of COLING 2004*, pp.562-568.

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

Sun M.S., D.Y. Shen and B. K. Tsou. 1998. Word segmentation without using lexicon and hand-crafted training data. *In Proceeding of COLING-ACL'98*, pp 1265-1271.

Wang, X.J., Y. Qin and W. Liu. 2007. A search-based Chinese word segmentation method. *Proceeding of 16th International Conf. of WWW*, pp.1129-1130.

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

Zhao, H. and Chunyu Kit. 2008. An Empirical Comparison of Goodness Measures for Unsupervised Chinese Word Segmentation with a Unified Framework. *Proceedings of 3th International Joint Conf. on Natural Language Processing (IJCNLP-2008)*, pp.9-16.