Learning Case-based Knowledge for Disambiguating Chinese Word Segmentation: A preliminary study

Chunyu Kit[†] **Haihua Pan**[†]
Dept. of Chinese, Translation & Linguistics[†]
City University of Hong Kong
{ctckit, cthpan}@cityu.edu.hk

Hongbiao Chen[†‡]
Dept. of Foreign Trade & Economic[‡]
Cooperation of Guangdong Province, China
drhbchen@21cn.com

Abstract

Just like other NLP applications, a serious problem with Chinese word segmentation lies in the ambiguities involved. Disambiguation methods fall into different categories, e.g., rule-based, statistical-based and example-based approaches, each of which may involve a variety of machine learning techniques. In this paper we report our current progress within the example-based approach, including its framework, example representation and collection, example matching and application. Experimental results show that this effective approach resolves more than 90% of ambiguities found. Hence, if it is integrated effectively with a segmentation method of the precision $P > 95\%$, the resulting segmentation accuracy can reach, theoretically, beyond 99.5%.

1 Introduction

It has been nearly two decades since the early work of Chinese word segmentation (Liang, 1984; Liang and Liu, 1985; Liu and Liang, 1986; Liang, 1986) . Tokenization has been recognized as a widespread problem, rather than being unique to Chinese and other oriental languages. It is an initial or prerequisite phase of NLP for all languages, although the obscurity of the problem varies from language to language (Webster and Kit, 1992a; Palmer, 2000). Recent work on tokenization for European languages such as English is reported in (Grefenstette and Tapanainen, 1994; Grefenstette, 1999; Grefenstette et al., 2000), adopting a finite-state approach. However, identification of multi-word units such as proper names and technical terms in these languages is highly comparable to that of multi-character Chinese words: there are no delimiters available.

So far, a great variety of segmentation strategies for Chinese with various linguistic resources have been explored, yielding a large volume of literature on both linguistic and computational sides, as listed in (Liu et al., 1994; Guo, 1997), among many others. In general, these strategies can be divided into two camps, namely, dictionary-based and statistical-based approaches. Nevertheless, the former can be understood as a restricted instance of the latter, with an equi-probability for each word in a given dictionary[1].

Most, if not all, dictionary-based strategies are built upon a few basic "mechanical" segmentation methods based on string matching (Kit et al., 1989), among which the most applicable, thus widely used since the very beginning, are the two *maximum matching* methods (MMs), one scanning *forward* (FMM) and the other *backward* (BMM). Interestingly, their performance, frequently used as the baseline for evaluation, is never too far away from the state-of-the-art approaches in terms of segmentation accuracy. Although performing little statistical computation, the MMs comply, in general, with the essential principle of the statistical-based approaches: select a segmentation as probable as possible among all choices. This *ad hoc* way of choosing the segmentation with fewest words usually leads to, by coincidence, a more probable output than most other choices with more words[2].

[1] A dictionary is actually a restricted form of language model, in this sense.

[2] The coincidence of fewer words with a greater probability can be illustrated as follows: given a string s, the probability of its most probable segmentation $\mathrm{seg}(s)$ in terms of a given language model is

$$\mathrm{prob}(\mathrm{seg}(s)) = \max_{s=w_1 w_2 \cdots w_n} \prod_i^n \mathrm{prob}(w_i|\cdot)$$

where $\mathrm{prob}(w_i|\cdot)$ is some conditional probability in the model. Since all $\mathrm{prob}(w_i|\cdot) < 1.0$, this probability becomes smaller for a greater n. Clearly, it looks more straightforward in an equi-probability setting.

Statistical approaches involve language models, mostly finite-state ones, trained on some large-scale corpora, as showed in Fan and Tsai (1988), Chang et al. (1991), Chiang et al. (1992), Sproat et al. (1996), Pont and Croft (1996) and Ng and Lua (forthcoming). These approaches do not provide any explicit strategy for disambiguation, but they get more ambiguous chunks correctly segmented than MMs by virtue of probability. Other linguistic resources or computational processes can also be integrated for further improvement, e.g., Lai et al. (1991) attempts to integrate POS tagging with word segmentation for the enhancement of accuracy and Gan et al. (1997) integrates word boundary disambiguation into sentence processing within a probabilistic emergent model. There are also other approaches that incorporate various techniques of statistical NLP and machine learning, e.g., transformation-based error-driven learning (Palmer, 1997; Hockenmaier and Brew, 1998) and compression-based algorithm (Teahan et al., 2000).

Recent research shifts its focus onto the following aspects, resorting to a variety of resources and techniques, in particular, machine learning techniques:

1. Lexical resource acquisition, including compilation and automatic detection of high-tech terms and unknown words like names, to complement a never-big-enough dictionary (Chang et al., 1995; Pont and Croft, 1996; Chang and Su, 1997);
2. Investigation into the nature and statistics of ambiguities (Sun and Zhou, 1998);
3. Unsupervised learning of words (Ge et al., 1999; Peng and Schuurmans, 2001)[3];
4. Disambiguation with different approaches (Liang, 1989; Jin, 1994; Sun and T'sou, 1995)

The work reported in this paper belongs to the last category, taking an instance-based learning approach, aimed to examine its prospects of disambiguation.

The rest of the paper is organized as follows. Section 2 briefly introduces the ambiguity problem and existing ambiguity detection strategies. Section 3 defines the notion and representation of examples, and formulates a similarity measure between an ambiguous input and an example. We present our disambiguation algorithm in Section 4 and experimental results and evaluation in Section 5, together with some discussion on the remaining errors, before concluding the paper in Section 6.

2 Ambiguity

Conceptually there are two essential types of ambiguity in Chinese word segmentation, which are conventionally termed as *overlapping* and *combinational* ambiguities. They can be formally defined as follows, given a dictionary $\mathcal{D}$:

Overlapping ambiguity A given string $\alpha\beta\gamma$ involves an overlapping ambiguity, if the set of sub-strings $\{\alpha\beta,\ \beta\gamma\} \subset \mathcal{D}$.

Combinational ambiguity A given string $\alpha\beta$ involves a combinational ambiguity, if the set of sub-strings $\{\alpha,\ \beta,\ \alpha\beta\} \subset \mathcal{D}$.

In practice the first type commonly co-occurs with the second, because almost all Chinese characters can be mono-character words. For the same reason, almost every multi-character word involves a combinational ambiguity. Fortunately, however, most of them are "resolved", characteristically, in a sense, by a MM strategy. Therefore, the focus of disambiguation is unsurprisingly put on the unresolved ones as well as the overlapping ambiguities.

2.1 Ambiguity detection

Conventionally, a straightforward strategy is exploited to detect ambiguities with the aid of FMM and BMM: the discrepancies of their outputs signal ambiguous strings. It appears adequately efficient, because only a forward and a backward scanning of the input will do.

However, its reliability remains a question, although it has been taken for granted for a long time that there would be few ambiguities left out, which is at odds with our observation that there are ambiguous strings for which both MMs output an identical segmentation. E.g.,

[3] Recent research in this direction appears to be closely related to the studies on computational lexical acquisition of other languages such as English (de Marcken, 1996; Brent, 1999; Kit and Wilks, 1999; Kit, 2000; Venkataraman, 2001) and to language modeling technology (Jelinek, 1997), typically involving a version of the EM algorithm (Dempster et al., 1977).

given a string abcde with {a, ab, bcd, c, de, e} $\in \mathcal{D}$, it is conceivable that both MMs output $\cdots$ab c de$\cdots$, and consequently the embedded ambiguity is unseen. So far we haven't seen any report on the incompleteness of ambiguity detection via this strategy.

A more comprehensive strategy would be that we first locate the boundaries of all *possible* words in terms of a given dictionary[4] are first located, and then, the common sub-strings among these words are detected: any common substring indicates an ambiguity.

Since our current work is intended to examine the effectiveness of an example-based learning approach to resolve found ambiguities, its merits do not rely on the completeness of ambiguity detection. The conventional strategy would suffice for the purpose of identifying an adequate number of ambiguities for our experiments.

3 Examples and similarity measure

We intend to disambiguate Chinese word segmentation ambiguities within the framework of *case*-based learning. This supervised learning approach, also labeled as *memory*-based, *instance*-based or *example*-based learning, has been popular for various NLP applications in recent years, e.g., the TiMBL learner (Daelemans et al., 2001). TiMBL is developed as a general memory-based learning environment to integrate a set of learning algorithms. It has been widely applied to disambiguating a variety of NLP tasks, including PP attachment (Zavrel et al., 1998), shallow parsing (Daelemans et al., 1999) and WSD (Veenstra et al., 2000; Stevenson and Wilks, 2001). In this paper, the general principle of case-based learning is followed but the formulation below is nevertheless specific to our problem.

An *example* here is defined as a quadruple $< C^l,\ e,\ C^r,\ S >$, where the strings C^l and C^r are the left and right *contexts* within which the ambiguous string e appears, and S is the correct segmentation of e within the particular context. If denoting the quadruple as E, we also refer to S as seg(E) or seg(e), interchangeably.

The *distance*, or *similarity*, between an example E and a given triple $A =< C_a^l,\ a,\ C_a^r >$ with the ambiguous string a is defined as

$$\Delta(A, E) = \delta(a, e)(1 + \sum_{i}^{\{l,r\}} \delta^i(C_a^i, C^i)) \qquad (1)$$

where $\delta(\cdot,\cdot)$ indicates the identity of two ambiguous strings, defined as

$$\delta(a, e) = \begin{cases} 1, & \text{if } a{=}e \\ 0, & \text{otherwise} \end{cases} \qquad (2)$$

and $\delta^i(\cdot,\cdot)$ (for $i \in \{l,\ r\}$) is the similarity of the corresponding contexts, measured in terms of the length of their common prefix (for the right contexts) or suffix (for the left contexts) in number of words[5]. For two given strings if we denote their common suffix (i.e., affix from the *right*) and prefix (i.e., affix from the *lefty*) respectively as[6] $f^r(\cdot,\cdot)$ and $f^l(\cdot,\cdot)$, we have $\delta^i(\cdot,\cdot) = |f^{\bar{i}}(\cdot,\cdot)|$. Thus, we can rewrite (1) into (3) below.

$$\Delta(A, E) = \delta(a, e)(1 + \sum_{i}^{\{l,r\}} |f^{\bar{i}}(C_a^i, C^i)|) \qquad (3)$$

Actually, the idea behind this equation is more straightforward than it looks. Basically, we measure the similarity of a given triple (i.e., an ambiguous string and its contexts) and an example in terms of the similarity of their contexts. However, this similarity is meaningful if and only if the strings in question are identical. This is why we define $\delta(a, e)$.

Given a triple $A =< C_a^l,\ a,\ C_a^r >$ and a collection $\mathcal{E}$ of examples, known as *example base* (EB), the strategy we undertake to determine a segmentation for the ambiguous string a can be formulated as follows, for $\Delta(A, E) \geq 1$:

$$\text{seg}(a \in A) = \text{seg}\left(\arg\max_{E \in \mathcal{E}} \Delta(A, E)\right) \qquad (4)$$

where seg($\cdot$) denotes the segmentation of a given string or example. Straightforwardly, Equation (4) can be read off as the following: segment a in the same way as its most similar example in the example base.

[4] Notice that ambiguities are dictionary-dependent.

[5] Obviously, measuring the length in number of characters is an alternative to explore in our future research.

[6] For example,

f^l(abc, dbc) = null, f^r(abc, dbc) = bc
f^l(abc, abd) = ab, f^r(abc, abd) = null

4 Algorithms

In order to test the effectiveness of the disambiguation strategy formulated above, we need to collect examples from a large-scale unrestricted corpus via a sound ambiguity detection program, and apply the examples to ambiguous strings in a test corpus via an example application program. In this section we present the algorithms for these purposes.

4.1 Ambiguity detection

We take a conventional approach to ambiguity detection, by detecting the discrepancies of the outputs from the FMM and BMM segmentations. Given an input corpus $\mathcal{C}$, it can be realized, plainly, by the following algorithm:

Ambiguity detection algorithm: `ambd`($\mathcal{C}$)

1. $\mathcal{F} =$ `FMM`($\mathcal{C}$) and $\mathcal{B} =$ `BMM`($\mathcal{C}$)
2. Return `diff`($\mathcal{F}, \mathcal{B}$)

where `FMM`($\cdot$) and `FMM`($\cdot$) return the FMM and BMM segmentations of $\mathcal{C}$, and `diff`($\cdot, \cdot$) returns the discrepancies of the two segmentations.

The dictionary used to support the MMs is a merger of the word lists from Liu et al. (1994) and Yu (1998), consisting of 53K entries. It is a medium-sized dictionary. With regards to the dictionary size and the weakness of the ambiguity detection algorithm, we keep ourselves alert of the fact that there are a certain number of ambiguities that are not detected by our program. And the resolutions for the ambiguous strings so detected are manually prepared, by selecting an answer from the outputs of the MMs in use.

4.2 Disambiguation

Given an example base $\mathcal{E}$ and a text corpus $\mathcal{C}$ as testing data, the disambiguation algorithm works along the following steps:

Disambiguation algorithm: `disamb`($\mathcal{C}$, $\mathcal{E}$)

1. Ambiguity detection: $\mathcal{A} =$ `ambd`($\mathcal{C}$)
2. For every $\alpha a \beta \in \mathcal{C}$ such that $a \in \mathcal{A}$, let $A = <\alpha,\ a,\ \beta>$
   - 2.1 Search for $E = \arg\max_{e \in \mathcal{E}} \Delta(A, e)$
   - 2.2 If $\Delta(A, E) > 1$, seg(a) = seg(E)
   - 2.3 Else seg(a) = $\arg\max_{s \in \{\texttt{FMM}(a),\ \texttt{BMM}(a)\}} q(s)$

where $q(\cdot)$ gives a probability-like score for a segmentation, by which we hope to get a better result than a random or brute-force choice between the FMM and BMM outputs (that we could have made). We refer to $q(\cdot)$ as a *solidness* function that is defined as the following, mainly for the simplicity of implementation:

$$q(w_1 w_2 \cdots w_n) = \prod_i^n p_w(w_i) \qquad (5)$$

where $p_w(\cdot)$ is the probability of a given string being a word. It is defined as

$$p_w(w_i) = \frac{f_w(w_i)}{f(w_i)}$$

where $f_w(\cdot)$ and $f(\cdot)$ are, respectively, the frequencies of a given item occurring as a word and as a string in the training corpus. Since it is an approximation, we can count the word frequencies based on the FMM output.

5 Experiment and evaluation

A number of experiments were conducted on unrestricted texts for the purpose of testing the effectiveness of the above disambiguation approach. In this section we present the data for training (i.e., example collection) and testing, experimental results and evaluation.

5.1 Data

The data we used for the experiments are news texts collected from mainland China, Hong Kong and Taiwan. The corpus size is of 778K words and 1.5M characters in total, in 1534 text files. About 3/4 of the data, of 1.16M characters in 1.1K files, are used for training and the remaining 1/4, of 360K characters in about 0.4K files, for testing. The statistics about the ambiguous strings found in the training and testing data is given in Table 1. From the ambiguity-word (EW) ratio, we can see that the ambiguity distribution among the two data sets is approximately even.

5.2 Results and evaluation

Theoretically, disambiguation accuracy on the training data should be 100%, because all found ambiguities are manually resolved. In contrast, the accuracy on the test set is more indicative of the effectiveness of the disambiguation strategy.

Training Data			EW Ratio
Number of cases	Total	5401	0.91%
	Unique	3018	0.51%
Testing Data			EW Ratio
Number of cases	Total	1648	0.90%
	Unique	995	0.54%

Table 1: Ambiguities in training & testing data

Our experimental results show that among 1648 ambiguities found in the test set, 1488 are properly resolved, in terms of our manual checking of the disambiguation outputs. Accordingly, the disambiguation accuracy is 90.29%.

We do not report the overall segmentation accuracy here for a number of reasons. Firstly, almost every paper in recent years reports a segmentation accuracy that nearly reaches the ceiling. This fact suggests that such figures seem to have carried less and less academic significance, in the sense that they do not measure any significant advance in tackling the major remaining problems in Chinese word segmentation, such as unknown words and segmentation ambiguities. Instead, all these figures seem to indicate a similar performance, which is, more interestingly, even similar to the performance reported a decade ago. Secondly, we have not had much ground to compare different systems' performance, not only because they were tested with different sets of data but also because the ways of calculating the segmentation accuracy are observed to be different from one another. On the contrary, the disambiguation accuracy is more specific, revealing exactly the capacity of a disambiguation strategy to resolve particular ambiguities found. It is reasonable to assume that everyone can get the unambiguous part correct in word segmentation, so we need not bother taking this part into account for the evaluation of disambiguation performance. Instead, we choose to concentrate on the problematic part, reporting only the disambiguation accuracy for the purpose of evaluation.

5.3 Discussions

As pointed out before, the conventional strategy for ambiguity detection that we have adopted is known to be incomplete. Many remaining ambiguities in the data are still to be brought to light. It is certainly a research direction that deserves more effort. Discovering more such missing cases can no doubt enlarge the example base significantly, and consequently enhance the strength of this case-based learning approach to disambiguation.

This problem is also related to the intrinsic disambiguation ability of the rudimental MMs: they segment many ambiguous strings correctly because of their own characteristics rather than by chance. Thus, it is worth digging out these uncovered ambiguities as examples so that they can be correctly handled when they show up elsewhere that would puzzle the MMs.

A more detailed analysis of experimental results is also expected, e.g., how many cases are resolved by existing examples and how many others by chance, i.e., by the $q(\cdot)$ function, which was designed to alleviate, rather than resolve, the problem. Also, a careful analysis of unseen cases in the testing data is also critical for a more thorough evaluation of the merits of the case-based learning approach. It will reveal the *coverage* of the EB and severity of the sparse data problem. A conceivable solution for the moment is that we construct all possible ambiguities based on a given dictionary and assign to them proper resolutions, so as to produce an EB with greater coverage.

6 Conclusions

In this paper we have presented a case-based learning approach to resolving Chinese word segmentation ambiguities. We adopted a simple representation for the examples, each consisting of an ambiguous string and its contexts, and also formulated a similarity measure for matching an ambiguity and an example from the example base. The effectiveness of this learning approach was tested on a set of unrestricted news texts of 1.5M characters, and a disambiguation accuracy of 90% was achieved

With this promising result, what we can expect is that if this approach could be effectively integrated with a segmentation algorithm that has a segmentation performance of the accuracy P, the overall segmentation accuracy one can expect would be

$$P' = P + (1 - P)90\% = (90 + 10P)\%$$

From this formula, we can see that if $P > 90\%$,

then $P' > 99\%$, and if $P > 95\%$, then $P' > 99.5\%$. Therefore, a bright future seems to be promised, because most Chinese word segmenters were reported to have achieved an accuracy over 95%, according to the literature.

However, the problems we still have with this case-based learning approach include, mainly, the incompleteness of ambiguity detection and the unknown coverage of the example base collected from unrestricted texts. All these remaining problems, that we will tackle in our future research, would have certain effect on the effectiveness of integrating it into any Chinese word segmenter.

References

M. R. Brent. 1999. An efficient, probabilistically sound algorithm for segmentation and word discovery. *Machine Learning*, 34:71–106.

Jing-Shin Chang and Keh-Yih Su. 1997. An unsupervised iterative method for Chinese new lexicon extraction. *International Journal of Computational Linguistics & Chinese Language Processing*, 1(1):101–157.

Jyun-Sheng Chang, C.-D. Chen, and S.-D. Chen. 1991. Chinese word segmentation through constraint satisfaction and statistical optimization. In *ROCLING-IV*, pages 147–165, National Chiao-Tung University, Hsinchu, Taiwan.

Jing-Shin Chang, Yi-Chung Lin, and Keh-Yih Su. 1995. Automatic construction of a Chinese electronic dictionary. In David Yarovsky and Kenneth Church, editors, *WVLC-3*, pages 107–120, Somerset, New Jersey, June.

Keh-Jiann Chen and Shing-Huan Liu. 1992. Word identification for mandarin Chinese sentences. In *COLING'92*, volume I, pages 101–107, Nantes, France, July 23-28.

Tung-Hui Chiang, Ming-Yu Lin, and Keh-Yih Su. 1992. Statistical models for word segmentation and unknown word resolution. In *ROCLING-V*, pages 121–146, Taiwan.

W. Daelemans, S. Buchholz, and J. Veenstra. 1999. Memory-based shallow parsing. In *CoNLL-99*, pages 53–60, Bergen, Norway.

Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. 2001. Timbl: Tilburg memory based learner, version 4.0, reference guide. Technical Report ILK Technical Report 01-04, Induction of Linguistic Knowledge, Tilburg University, The Netherlands.

C. de Marcken. 1996. *Unsupervised Language Acquisition*. Ph.D. thesis, MIT, Cambridge, Mass.y.

A. P. Dempster, N. M. Laird, and D. B.Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 34:1–38.

Charng-Kang Fan and Wen-Hsiang Tsai. 1988. Automatic word identification in Chinese sentences by the relaxation technique. *Computer Processing of Chinese and Oriental Languages*, 4(1):33–56.

Kok-Wee Gan, Martha Palmer, and Kim-Teng Lua. 1997. A statistically emergent approach for language processing: Application to modeling effects in ambiguous Chinese word boundary perception. *Computational Linguistics*, 22(4):531–553.

Xianping Ge, Wanda Pratt, and Padhraic Smyth. 1999. Discovering Chinese words from unsegmented text (poster abstract). In *SIGIR'99*, pages 271–272, Berkeley, August.

Gregory Grefenstette and P. Tapanainen. 1994. What is a word, what is a sentence? Problems of tokenization. In *3rd Conference on Computational Lexicography and Text Research, COMPLEX'94*, Budapest, July 7-10.

Gregory Grefenstette, Anne Schiller, and Salah Aït-Mokhtar. 2000. Recognizing lexical patterns in text. In F. van Eynde, D. Gibbon, and I. Schuurman, editors, *Lexicon Development for Speech and Language Processing*, pages 141–168. Kluwer, Dordrecht.

Gregory Grefenstette. 1999. Tokenization. In Hans van Halteren, editor, *Syntactic Wordclass Tagging*, pages 117–133. Kluwer, Dordrecht.

Yingchun Guan and Bei Qin. 1986. The design and implementation of a Chinese word statistical system. *Journal of Chinese Information Processing*, 1(1):26–32. (In Chinese).

Jin Guo. 1997. Critical tokenization and its properties. *Computational Linguistics*, 23(4):569–596.

J. Hockenmaier and C. Brew. 1998. Error-driven learning of Chinese word segmentation. In *PACLIC-12*, pages 218–229, Singapore. Chinese and Oriental Languages Processing Society.

F. Jelinek. 1997. *Statistical Methods for Speech Processing*. MIT Press, Cambridge, MA.

Wanying Jin. 1992. A case study: Chinese segmentation and its disambiguation. Technical Report MCCS-92-227, Computing Research Laboratory, New Mexico State University, Las Cruces.

Wanying Jin. 1994. Chinese segmentation disambiguation. In *COLING-94*, pages 1245–1249.

Chunyu Kit and Yorick Wilks. 1999. Unsupervised learning of word boundary with description length gain. In M. Osborne and E. T. K. Sang, editors, *CoNLL-99*, pages 1–6, Bergen, June.

Chunyu Kit, Yuan Liu, and Nanyuan Liang. 1989. On methods of Chinese automatic word segmentation. *Journal of Chinese Information Processing*, 3(1):1–32. (In Chinese).

Chunyu Kit. 2000. *Unsupervised Lexical Learning*

as Inductive Inference. Ph.D. thesis, University of Sheffield, UK.

Tom B. Y. Lai, Sun C. Lin, Chaofen Sun, and Maosong Sun. 1991. A maximal matching automatic Chinese word segmentation algorithm using corpus tagging for ambiguity resolution. In *ROCLING-IV*, pages 17–23.

Nanyuan Liang and Yuan Liu. 1985. The OM method of automatic word segmentation. *Chinese Information*, 1(2). (In Chinese).

Nanyuan Liang. 1984. Automatic word segmentation for written Chinese and the segmentation system CDWS. *Journal of Beijing University of Aeronautics and Astronautics*, ?(4). (In Chinese).

Nanyuan Liang. 1986. CDWS – an automatic word segmentation system for written Chinese. *Journal of Chinese Information Processing*, 1(2):44–52. (In Chinese).

Nanyuan Liang. 1989. Knowledge for Chinese word segmentation. *Journal of Chinese Information Processing*, 4(2):29–33. (In Chinese).

Yuan Liu and Nanyuan Liang. 1986. Basic engineering for Chinese processing – Modern Chinese word frequency counting. *Journal of Chinese Information Processing*, 1(1):17–23. (In Chinese).

Yuan Liu, Qiang Tan, and Xukun Shen. 1994. *Contemporary Chinese Word Segmentation Standard Used for Information Processing, and Automatic Word Segmentation Methods*. Tsinghua University Press, Bejing. (In Chinese).

Hong I Ng and Kim Teng Lua. (forthcoming). A word finding automation for Chinese sentence tokenization. Submitted to ACM Transaction of Asian Languages Processing.

David Palmer and J. Burger. 1997. Chinese word segmentation and information retrieval. In *AAAI Spring Symposium on Cross-Language Text and Speech Retrieval*.

David Palmer. 1997. A trainable rule-based algorithm for word segmentation. In *ACL-97*, pages 321–328, Madrid.

David D. Palmer. 2000. Tokenization and sentence segmentation. In R. Dale, H. Moisl, and H. Somers, editors, *Handbook of Natural Language Processing*, pages 11–35. Marcel Dekker, New York.

Fuchun Peng and Dale Schuurmans. 2001. Self-supervised Chinese word segmentation. In *4th International Symposium of Intelligent Data Analysis*, pages 238–247.

Jay M. Pont and W. Bruce Croft. 1996. USeg: A retargetable word segmentation procedure for information retrieval. In *Symposium on Document Analysis and Information Retrieval' 96 (SDAIR)*. UMass Technical Report TR96-2, Univ. of Mass., Amherst, MA.

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

Mark Stevenson and Yorick A. Wilks. 2001. The interaction of knowledge sources in word sense disambiguation. *Computational Linguistics*, 27(3):321–349.

Maosong Sun and Benjamin K. T'sou. 1995. Ambiguity resolution in Chinese word segmentation. In Benjamin K. T'sou and Tom B. Y. Lai, editors, *PACLIC-10*, Hong Kong, December 27-28.

Maosong Sun and Zhengping Zhou. 1998. Word segmentation ambiguity in Chinese texts. In Benjiamin K. T'sou, Tom B. Y. Lai, Samuel W. K. Chan, and Williams S-Y. Wang, editors, *Quantitative and Computational Studies on the Chinese Language*, pages 323–338. Language Information Sciences Research Centre, City University of Hong Kong.

W. J. Teahan, Yingying Wen, Rodger J. McNab, and Ian H. Witten. 2000. A compression-based algorithm for Chinese word segmentation. *Computational Linguistics*, 26(3):375–393.

J. Veenstra, A. Van den Bosch, S. Buchholz, W. Daelemans, and J. Zavrel. 2000. Memory-based word sense disambiguation. *Computing and the Humanities*, special issue on SENSEVAL, 34(1-2y).

Anand Venkataraman. 2001. A statistical model for word discovery in transcribed speech. *Computational Linguistics*, 27(3):353–372.

Jonathan J. Webster and Chunyu Kit. 1992a. Tokenization as the initial phase in NLP. In *COLING'92*, pages 1106–1110, Nantes, France, July 23-28.

Jonathan J. Webster and Chunyu Kit. 1992b. Tokenization for machine translation: What can be learned from Chinese word identification. In *Proc. of 3rd International Conference on Chinese Information Processing,*, Beijing.

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

Shiwen Yu. 1998. *Knowledge Base of Grammatical Information for Contemporary Chinese*. Tsinghua University Press, Bejing. (In Chinese).

Jakub Zavrel, Walter Daelemans, and Jorn Veenstra. 1998. Resolving PP attachment ambiguities with memory-based learning. In T. Mark Ellison, editor, *CoNLL97: Computational Natural Language Learning*, pages 136–144, Somerset, New Jersey.

Guodong Zhou and Kim Teng Lua. (forthcoming). A hybrid approach toward ambiguity resolution in segmenting Chinese sentences. Submitted to Computer Processing of Oriental Languages.