The Left and Right Context of a Word: Overlapping Chinese Syllable Word Segmentation with Minimal Context[*]

MIKE TIAN-JIAN JIANG[*], National Tsing Hua University; Academia Sinica
TSUNG-HSIEN LEE[*], Academia Sinica; University of Texas at Austin
WEN-LIAN HSU, Academia Sinica; National Tsing Hua University

Since a Chinese syllable can correspond to many characters (homophones), the syllable-to-character conversion task is quite challenging for Chinese phonetic input methods (CPIM). There are usually two stages in a CPIM: 1. segment the syllable sequence into syllable words; 2. select the most likely character words for each syllable word. A CPIM usually assumes that the input is a complete sentence, and evaluates the performance based on a well-formed corpus. However, in practice, most Pinyin users prefer progressive text entry in several short chunks, mainly in one or two words each (most Chinese words consist of two or more characters). Short chunks do not provide enough contexts to perform the best possible syllable-to-character conversion, especially when a chunk consists of overlapping syllable words. In such cases, a conversion system often selects the boundary of a word with the highest frequency. Short chunk input is even more popular on platforms with limited computing power, such as mobile phones. Based on the observation that the relative strength of a word can be quite different when calculated leftwards or rightwards, we propose a simple division of the word context into the left context and the right context. Furthermore, we design a double ranking strategy for each word to reduce the number of errors in Step 1. Our strategy is modeled as the minimum feedback arc set problem on bipartite tournament with approximate solutions derived from genetic algorithm. Experiments show that, compared to the frequency-based method (FBM) (low memory and fast) and the conditional random fields (CRF) model (larger memory and slower), our double ranking strategy has the benefits of less memory, low power requirement with competitive performance. We believe a similar strategy could also be adopted to disambiguate conflicting linguistic patterns effectively.

Categories and Subject Descriptors: **I.2.7 [Natural Language Processing]**: Language Models

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Chinese phonetic input methods, syllable-to-word conversion

ACM Reference Format:

Jiang, M. T.-J., Lee, T.-H., and Hsu, W.-L. 2012. The Left and Right Context of a Word: Overlapping Chinese Syllable Word Segmentation with Minimal Context. ACM Trans. Asian Lang. Inform. Process. 0, 0, Article 0 (Month 2012), 25 pages.

DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

Most ideograph-based Asian languages consist of thousands of characters, making it impractical to create keyboards along the same style as alphabetic languages. In response, most modern systems come with built-in tools called input methods (IMs) for transforming multiple keystrokes into single ideographs. IMs are often categorized into “radical-based” or “phonetic-based” methods. With radical-based IMs, users construct characters by typing the composing radicals or strokes. Alternatively, phonetic-based IMs rely on phonetic transcriptions of ideographs, where users create characters by typing in the approximate spellings of their

[*] This work is supported in part by the National Science Foundation under grant NSC-100-2631-S-001-001, and the research center for Humanities and Social Sciences under grant IIS-50-23.

M. T.-J. Jiang and T.-H. Lee contributed equally to this work.

Author’s addresses: M. T.-J. Jiang, Computer Science Department, National Tsing Hua University; T.-H. Lee, Institute of Information Science, Academia Sinica; W.-L. Hsu, Institute of Information Science, Academia Sinica

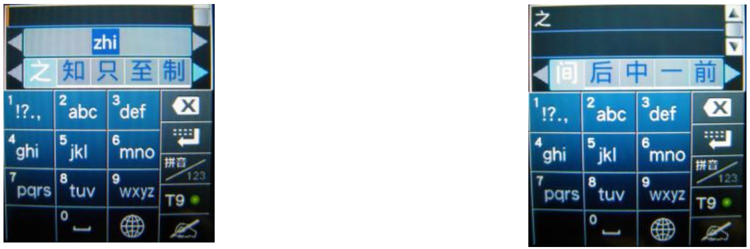
DOI10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

syllables. In the case of homographs or homophones, users are given a choice, and the proper character is selected and entered. While various types of IM can be used with a keyboard, this work specifically examines the context of predictive Chinese phonetic input method (CPIM). CPIM not only facilitates word prediction and word or phrase completion, but also disambiguates homophones of syllables into characters. To date, most natural language processing (NLP) research on Chinese IMs has focused on these predictive phonetic-based approaches, since Pinyin input is one of the most popular methods for Chinese typing, and homophone disambiguation, which can be regarded as a simplified version of speech recognition, is a major problem in Pinyin input. There are usually two steps in a CPIM: 1. *syllable word segmentation (SWS)*: segment the syllable sequence into syllable words; 2. *character word selection*: select the most likely character words for each syllable word. In this paper we shall focus on the SWS problem in Step 1. This paper attempts to balance the tradeoff between the cost of computing resource and the performance of homophone disambiguation based on an algorithmic study on short syllable sequence segmentation. With minimal context it is often difficult for a system to determine the most appropriate boundaries. Markovian based Pinyin input methods usually apply N-grams and dynamic programming to resolve ambiguities from both syllable and word sides [Chen and Lee 2000; Li et al. 2009; Gao et al. 2002; Wang et al. 2006; Xiao et al. 2007], we shall demonstrate that it is inefficient for mobile devices with limited resources [Liu and Wang 2002; Wu and Zheng 2003] and there are better alternatives.

One may argue that since character-based model is the-state-of-the-art of Chinese word segmentation, the most common way of CPIM is not to process syllables as words but to process them as characters. However, most previous works of Pinyin-to-character conversion prefer word-based model. For example, Chen and Lee [2000] and Gao et al. [2002] applied word-based tri-gram language model (with maximum matching word segmentation initially) for all possible word strings that match typed Pinyin to select the word string with the highest language model probability, because Yang et al. [1998] suggested that bypassing the issue of word boundaries did not yield good Pinyin-to-character conversion results. Since Gao et al. [2006] further elaborated that they assumed a unique mapping from word string to Pinyin string to make the decision problem depend solely upon probability of words, we may see their works as SWS. Similarly, Liu and Wang [2002] used unigram model with fewest words segmentation to implement their CPIM. Moreover, Wen et al. [2008] conducted a SWS specific work and shown that better SWS yield better CPIM. As a matter of fact, even sequential labeling models that were usually applied in fashion of characters, such as linear-chain Conditional Random Fields or Maximum Entropy Markov Models, were modified to use word-based features [Li et al. 2009; Xiao et al. 2007] to be tractable for CPIM in practice, which can be seen as a joint schema of SWS and character word selection that essentially does SWS in situ.

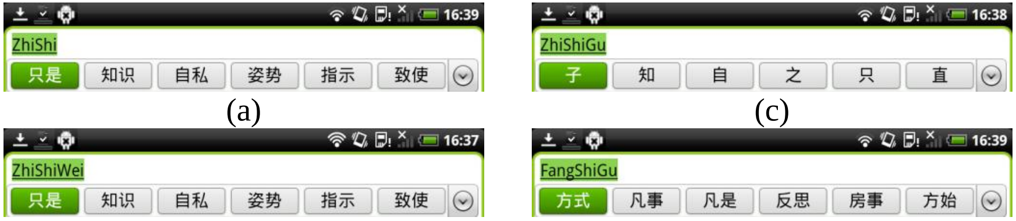
1.1 Motivation

Most studies on Chinese phonetic input method (CPIM) assume that the input is a complete sentence, which would provide sufficient context for language models to optimize their conversion performance. However, a phenomenon of short syllable input without delimiters is prevalent on mobile phones. Different keyboard layouts and/or computing power lead to different approaches. Some older system such as Dasher [Ward et al 2000] can just suggest character by character. Recent platforms may support so-called phrasal text entries that consider different depths of context [Liu and Wang 2007]. T9-alike methods utilize a context of only one word in European languages or of one character in Asian languages [Mackenzie and Soukoreff 2002]. Figure 1 gives examples of common T9 Pinyin usage. In Figure 1(a), typing a Pinyin syllable “zhi” gets a candidate list of frequent characters. Choosing the first candidate 之 (this/that) brings a candidate list of succeeding characters as Figure 1(b) demonstrates. With software QWERTY keyboard, larger memory and faster CPU, some OS manages to extend context to multi-character words. Figure 2 lists some typical cases. For a disyllabic input without explicit boundary marks, a system attempts to recognize word boundaries as in Figure 2(a). When the third syllable is pushed in, this system either keeps the candidate list unchanged as Figure 2(b) shows, or generates another candidate list as Figure 2(c) does. These cases indicate that, for the Pinyin syllables “zhi-shi-wei,” there is a bi-syllabic word hypothesis “zhi-shi” on the left. For the Pinyin syllables “zhi-shi-gu,” however, the preferred word boundary on the left becomes monosyllabic as “zhi.” Another interesting case here is Figure 2(d) with the Pinyin syllable “fang-shi-gu,” where the candidate list comes from “fang-shi” on the left. Similar boundary ambiguities of word hypotheses surrounding the syllable “shi” can be found on phrasal Pinyin input methods with longer context. Figure 3(a) presents the same behavior of Figure 2(a) and Figure 2(b), whereas Figure 3(b) matches Figure 2(c).



(a) (b)

Figure 1. (a) T9 Pinyin for a syllable “zhi” and its candidates of Chinese character; (b) A selected Character 之 (this/that) and suggested succeeding characters.


16:39
ZhiShi
只是 知识 自私 姿势 指示 致使
16:38
ZhiShiGu
子 知 自 之 只 直
16:37
ZhiShiWei
只是 知识 自私 姿势 指示 致使
16:39
FangShiGu
方式 凡事 凡是 反思 房事 方始

(a) (c)

(b) (d)

Figure 2 (a) Pinyin for syllables “zhi-shi” and candidates of Chinese word;
(b) Pinyin for syllables “zhi-shi-wei” and candidates remaining the same”
(c) Pinyin for syllables “zhi-shi-gu” and candidates from “zhi” only;
(d) Pinyin for syllables “fang-shi-gu” and candidates from “fang-shi.”

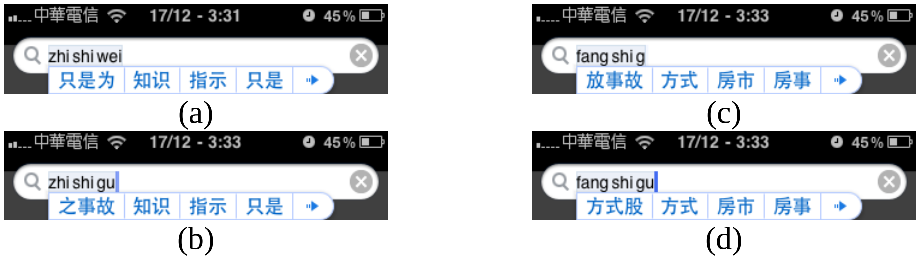


Figure 3 (a) Pinyin for syllables “zhi-shi-wei” and candidates of Chinese phrase;
(b) Pinyin for syllables “zhi-shi-gu” and candidates of Chinese phrase;
(c) Pinyin for incomplete syllables “fang-shi-g” and candidates of Chinese phrase;
(d) Pinyin for syllables “fang-shi-gu” and candidates of Chinese phrase.

Also, for desktop input methods, Pinyin users in China prefer to input short chunks that comprise relatively less words than complete sentences have, in order to obtain conversion result as soon as possible. This preference reflects on 首选词正确率 (first chosen word accuracy), which is emphasized by major Pinyin input method manufacturers since 2006[12345] and becomes one of China media’s favorite evaluation metrics of Pinyin input methods[679]. Table 1 gives examples extracted from a recent product comparison chart of popular Pinyin input methods in China[8].

Table 1. Short Pinyin syllables without boundary hints that cause non-unique conversion results in Chinese

Pinyin without boundaries	Common Result	Alternative Result
Xiangfengshi	相逢是 (to meet is…)	像风湿 (like rheumatism)
Qianfu	潜伏 (lurk)	其安抚 (its pacification)
Yonghengzhita	永恒之塔 (eternal tower)	永恒致他 (eternity to him)
Yibujieshouyuding	已不接受预订 (reservation has been closed)	一部接受预订 (one unit can be reserved)

1.2 Ambiguity on Syllable Words

Overlapping ambiguity on Chinese word segmentation (CWS) has been widely studied [Li et al. 2001; Li et al. 2003; Liang 1987; Qiao et al. 2008; Sun et al. 1998]. It is reported that, over 90% of overlapping ambiguity of words can be resolved in a context-free way [Li et al. 2001; Qiao et al. 2008; Sun et al. 1998]. According to Li et al. [2003], 47.98% of overlapping words have the same results suggested by forward maximum matching and backward maximum matching. On syllable string segmentation, however, similar phenomenon does not occur. As Zheng [1999] mentioned, there are just about 400 toneless monosyllables and

[1] http://fuwu.sogou.com/agent/market/20060612.html (in Chinese, retrieved on 2011/10)
[2] http://news.ccidnet.com/art/1032/20070409/1056141_1.html (in Chinese, retrieved on 2011/10)
[3] http://bbs.jjol.cn/showthread.php?t=1227 (in Chinese, retrieved on 2011/10)
[4] http://www.google.com/intl/zh-CN/ime/pinyin/privacy.html (in Chinese, retrieved on 2011/10)
[5] http://www.google.com/support/pinyin/bin/answer.py?hl=zh-Hans&answer=62636 (in Chinese, retrieved on 2011/10)
[6] http://pcedu.pconline.com.cn/pingce/pingcenormal/0909/1794426_1.html (in Chinese, retrieved on 2011/10)
[7] http://soft.zol.com.cn/103/1036808.html (in Chinese, retrieved on 2011/10)
[8] http://soft.zol.com.cn/132/1320458.html (in Chinese, retrieved on 2011/10)

about 1,300 tonal monosyllables, but they represent pronunciations for at least 6,700 Chinese characters. On the average, 17 Chinese characters share one toneless syllable and 5 Chinese characters share one tonal syllable. This situation implies that syllable string segmentation involves more ambiguity than word segmentation, and the boundary determination is even harder. For example, consider a Chinese character string 知识为 (knowledge is…), which is easy to segment into two words, namely 知识(knowledge) and 为 (is). By contrast, "zhi-shi-wei" as the toneless Pinyin syllable string of 知识为, is not that easy to find an unique choice for SWS such as "zhi-shi/wei", because the same syllable string can represent another Chinese character string 之侍卫 (someone's guard), which is segmented into 之 (someone's) and 侍卫 (guard) that suggest corresponding SWS as "zhi/shi-wei". Figure 4 illustrates the increased ambiguities from a character string 各国有企业 to its tonal syllable string "ge4-guo2-you3-qi4-ye4" and toneless syllable string "ge-guo-you-qi-ye". The first row of the character string in Figure 4 can be segmented into either 各 (every) / 国有 (state own) / 企业 (enterprise) or 各国 (each country) / 有(has) / 企业 (enterprise), where the direction of arrows indicate that 国.(country) is overlapped. The second, the third, and the fourth rows that are grouped in a cyan plate represent that the character string's tonal syllable string "ge4-guo2-you3-qi4-ye4" introduces an additional ambiguity because of the homophone 有气 (being angry) of "you3-qi4", while the following plate draws a

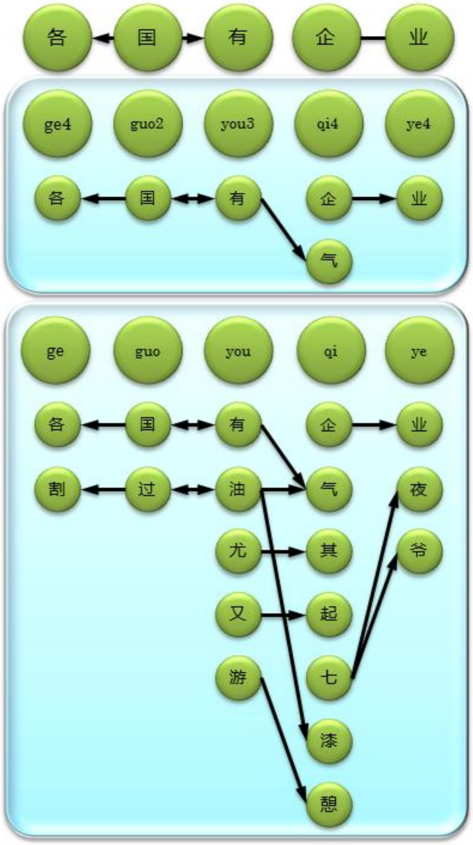


Figure 4. A character string 各国有企业 along with its tonal and toneless syllable strings form different degrees of ambiguities, representing by lattice.

lattice of the toneless syllable string “ge-guo-you-qi-ye” involving more homophones, such as 尤其 (especially) , 油漆 (paint) , and 油气 (oil and gas), 游憩 (recreation) of “you-qi”.

Since the study is about short syllable strings, we focus on syllable strings consisting of 3 to 6 syllables overlapping on their syllable words. For ambiguity on syllable words, formal definitions of ambiguities in word segmentation [Liang 1987] are adopted as

- A syllable string “XYZ” is an overlap ambiguity string (OAS) if it can be segmented into two syllable words either as “XY/Z” or “X/YZ”, depending on context.
- A syllable string “XY” is a combination ambiguity string (CAS) if X, Y, and XY are syllable words.

where a “syllable word” means a syllable substring that has one or more corresponding Chinese words according to certain word segmentation standard. For example as mentioned above, “zhi-shi-wei” is an OAS involving four toneless syllable words “zhi-shi”, “shi-wei”, “zhi”, and “wei” for 知识, 侍卫, 之, and 为, respectively.

Speaking of ambiguity string, two additional definitions involved. One is called longest OAS (LOAS). The LOAS is an OAS that is not a substring of any other OAS in a given chunk. For example, both 任何时候 (anytime) and 任何时 (anytime) are OASs, but only 任何时候 is a LOAS. The LOAS has been introduced for word segmentation study in sentence level [Sun et al. 1998], which is not feasible in this study of short syllable strings.

Another additional definition of ambiguity string is about pseudo ambiguity (PA) vs. true ambiguity (TA) [Sun et al. 1998]. The PA indicates that, despite the multiple segmentation possibilities (according to certain dictionary), there is only one way to segment the given string in reality (of certain corpus). For example, a given string 市政府 (city government) can be segmented into either 市 (city) / 政府 (government) or 市政 (city policy) / 府 (the seat of government) since 市政, 政府, 市, and 府 are all registered in the given dictionary, but the latter segmentation is not found in the given corpus. On the contrary, the TA means that the string can be segmented in more than one way in practice. For example, both 从小 (from one’s childhood) / 学 (learn) and 从 (from) / 小学 (elementary school) are usually easily recognized from a given corpus. In this study, the appearance of true overlap ambiguity string (TOAS) is one of the criteria for choosing corpus, but corpora have no TOAS are still useful as open test data, since PA strings of a small corpus may actually be unseen TA strings.

1.3 Double Ranking Strategy

We first describe the following important observation on the context of words: when two syllable words overlap (or compete for a boundary) in a short syllable string, their relative positions (left or right) play a crucial role in determining which one should be selected. For example, “jun-shi” as 军事 (military) is a high frequency disyllable whose left and right strengths are quite different. It is very strong when competing with polysyllables on its right, as shown by the tri-syllable “jun-shi-jie,” which is usually segmented into “jun-shi/jie”, or 军事 (military) / 界 (area). However, it becomes relatively weak when competing with other polysyllables on its left, as shown by the tri-syllable “lu-jun-shi,” which is usually segmented into “lu-jun/shi,” or 陆军(army) / 是 (is). Table 2 provides more examples.

Table 2. Short Pinyin syllables have different strengths when competing with others on the left or right

Pinyin Syllable	Common Chinese Words	Alternative Chinese Words
ji-bing	罹患 (being affected) / 疾病 (disease) li-huan / ji-bing	及 (and) / 病人 (patient) ji / bing-ren
zhi-li	智力 (zhi-li, intelligence) / 测验 (test) zhi-li / ce-yian	自制 (self-restraint) / 力 (will power) zi-zhi / li
qi-zhong	其中 (among) / 有 (have) qi-zhong / you	蜜月期 (honeymoon) / 中 (in) mi-yue-qi / zhong
ji-hui	有 (get) / 机会 (chance) you / ji-hui	自由基 (free radicals) / 会 (will) zi-you-ji / hui
jia-shi	家事 (house-keeping) / 是 (is) jia-shi / shi	科学家 (scientist) / 逝世 (pass away) ke-xue-jia /shi-shi
guan-xi	关西 (Kansai) / 机场 (airport) guan-xi / ji-chang	检察官 (prosecutor) / 希望 (hope) jian-cha-guan / xi-wang
xing-li	行李 (luggage) / 箱 (case) xing-li / xiang	造型 (modeling) / 里 (inside) zao-xing / li
su-qiu	诉求 (demand) / 一 (one) su-qui / yi	火速 (at top speed) / 求医 (seek medical help) huo-su / qiu-yi
qiu-yi	火速 (at top speed) / 求医 (seek medical help) huo-su / qiu-yi	求 (beg) / 医师 (doctor) qiu / yi-shi
yi-shi	求 (beg) / 医师 (doctor) qiu / yi-shi	好球 (strike) / 一 (one) /失误 (error) hao-qiu / yi /shi-wu

Examples mentioned above clearly demonstrate that a polysyllable’s frequency is not necessarily representative of its strength in segmentation, and some of them can lead to complicated relationships, such as the Chinese character strings the row of “ji-hui”, namely 有(get) / 机会(chance) and 自由基 (free radicals) / 会 (will), contain another overlapping syllables of “you-ji”, while the last three rows of “su-qiu”, “qiu-yi”, “yi-shi” may form a chain as “su-qui-yi-shi”. Since Table 2 lists Chinese character words based on tonal Pinyin syllables, one may imagine that the situation of toneless Pinyin syllables can be even more complicated. Therefore, we propose a simple division of a word context into its left context and right context. Furthermore, we design a double ranking strategy for each word to reduce the number of errors in syllable word segmentation of Step 1: for each polysyllable *w*, we assign an integer as its left rank and another as its right rank. When a polysyllable *u* overlaps with another polysyllable *v* to its right, we compare the right rank of *u* to the left rank of *v* in order to determine whether the segmentation should follow that of *u* or *v* based on the following rules: If the right rank of *u* is bigger than the left rank of *v*, then *u* is selected. Conversely, if the right rank of *u* is smaller than the left rank of *v*, then *v* is selected; and if the right rank of *u* is equal to the left rank of *v*, the one with the higher frequency is selected. The left and right ranks of a polysyllable can be considered as the relative strength of the polysyllable in each direction. In some cases, the left and right ranks can differ substantially.

The remainder of this paper is organized as follows. Section 2 provides the preliminaries of problem formulation. In Section 3, we describe the proposed algorithms. The experiment results are detailed in Section 4. We then summarize our conclusions in Section 5.

2. PROBLEM FORMULATION

2.1 Double Rank Assignment Problem (DRAP)

In DRAP, we assign left and right ranks to each syllable to help us perform syllable word segmentation in Chinese. For the initial assignment, we consider only pairs of syllables overlapping in a single phoneme. Ranks obtained this way will be applied to segment syllables in general short texts. We have conducted closed tests of toneless Pinyin with ranks 10, 20, 40, and 80. We then decide to limit the total number of ranks to 20 for further experiments (on open tests on toneless Pinyin and closed tests on tonal Pinyin), since the 20 rank version can be implemented quite efficiently and does not seem to affect the performance.

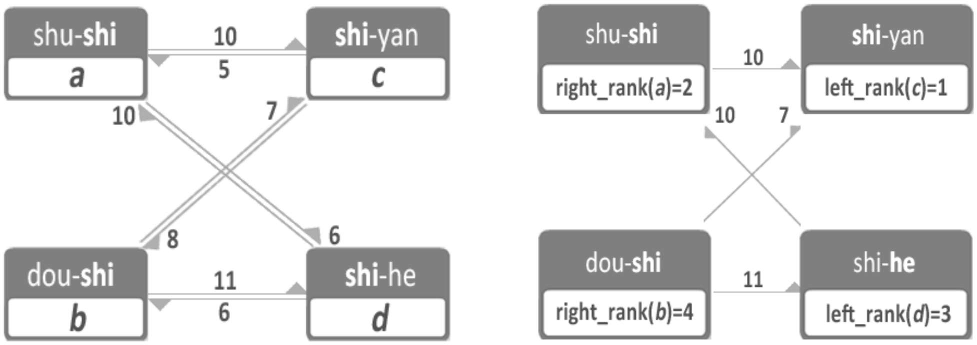
2.2 Problem Definition

We now formally define the DRAP for Chinese short syllable word segmentation. For each syllable t, consider the following competition graph $G = (L(V) \cup R(V), E)$, in which each vertex in $L(V)$ denotes a polysyllable that ends in the syllable t; each vertex in $R(V)$ denotes a polysyllable that begins with t; and each arc $e_{r,s}$ in E directed from a vertex r in $L(V) \cup R(V)$ to a vertex s with weight $w(e_{r,s})$ represents the number of times syllable word r is selected over s in a desirable segmentation. Assign a left rank $left\_rank(v)$ to each vertex v in $R(V)$ and a right rank $right\_rank(u)$ to each vertex u in $L(V)$. A syllable's left and right ranks are both positive integers no larger than a pre-defined limit *rank_limit*. After rank assignment, we determine the relation between any two connected vertices u and v, where u is in $L(V)$ and v is in $R(V)$, as follows.

I. If $right\_rank(u) > left\_rank(v)$, then u is selected.
II. If $right\_rank(u) < left\_rank(v)$, then v is selected.
III. If $right\_rank(u) = left\_rank(v)$, then the polysyllable with the higher frequency is selected.

Next, let E' be the set of arcs $e_{u,v}$ in which u is selected. Our objective is to maximize the total score $\Sigma\{w(e_{u,v}) \mid e_{u,v} \in E'\}$, which represents the number of times this rank assignment correctly chooses segmented syllables. On the other hand, for each arc $e_{u,v}$ not in E', $w(e_{u,v})$ represents the penalty incurred for the incorrect segmentation. Hence, an equivalent objective of DRAP is to minimize the total penalty $\Sigma\{w(e_{u,v}) \mid e_{u,v} \notin E'\}$. Figure 5 provides an overview of our problem. The competition graph in Figure 5(a) contains four vertices, a, b, c, and d, which are Pinyin polysyllables that begin or end with the Pinyin monosyllable "shi." The number on each edge represents its weight. Figure 5(b) shows the optimal solution if we assign $right\_rank(a) = 2$, $right\_rank(b) = 4$, $left\_rank(c) =$

1, and *left_rank*(*d*) = 3. The total weight of all the edges in Figure 5(b) is 38, which is the score of the double rank assignment.



(a) (b)

Figure 5. (a) A sample competition graph formed by the syllable “shi;” (b) An optimal solution.

2.3 The Feedback Arc Set Problem (FASP)

A problem closely related to the DRAP is the following Feedback Arc Set Problem (FASP). Given a bipartite graph $G = (V_1 \text{ x } V_2, E)$ with arcs directed between V_1 and V_2, the FASP is to delete a set of arcs E_1 with minimum total weight such that the remaining graph is acyclic. If one ignores the rank-limit and assign only distinct ranks, then the FASP can be reduced to the DRAP as follows. Given a bipartite graph, regarded as a competition graph with $L(V) = V_1$, $R(V) = V_2$, solve the DRAP by assigning distinct ranks to vertices in V_1 and V_2 to minimize the total penalty. Let $E_1 = \{e_{u,v} \mid \text{rank}(u) < \text{rank}(v)\}$. Then E_1 would be a solution to the FASP. Because if there is another set E'' with smaller weight whose deletion would also make the graph acyclic, then perform a topological sort on the graph $G = (V_1 \text{ x } V_2, E\text{-}E'')$. The order obtained would serve as the rank assignment for the DRAP. So E'' would be a solution to the DRAP better than E_1.

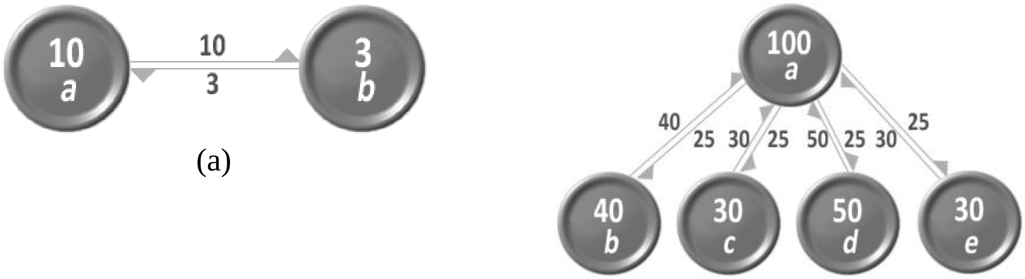
FASP is NP-hard [Guo et al. 2007]. Hence, DRAP is also NP-hard. There are several good approximation algorithms for FASP [Even et al. 1998; Gupta, 2008]. Generally, they only focus on the un-weighted case. We propose an algorithm in Section 3 for the weighted DRAP with prescribed rank-limit.

3. ALGORITHMS

In this section, we describe our double ranking algorithm (DRA) for DRAP in Chinese syllable word segmentation. As the size of graph G is too large, DRA first reduces the vertex set by pre-assigning ranks to low-frequency syllables. Then, it employs a genetic algorithm to assign ranks to the remaining syllables. A 10-rank DRA is denoted by DRA_{10}, a 20-rank one is DRA_{20}, and so on. The data set used in the experiment is described next.

3.1 Rank Pre-assignment

The number of vertices (syllables) in a competition graph can sometimes be more than 10,000, which is too large to handle. To reduce the problem size, we set a syllable’s left and right ranks as its frequency if the frequency is less than or equal to *rank_limit*/2. This rank pre-assignment step reduces the size of the original problem by approximately 70%. For the remaining high frequency syllables, we use a genetic algorithm to determine their ranks. This pre-assignment can effectively reduce the problem size and still maintain the quality of our solution. Figure 6 explains why we pre-assign ranks for low-frequency syllables. In Figure 6, the number on each vertex represents its frequency, and the number on each edge represents its weight.



(a)

(b)

Figure 6. (a) A competition graph with low frequency vertices.
(b) A competition graph with high frequency vertices.

Figure 6(a) shows a competition graph with low-frequency vertices. Note that a vertex’s frequency is equal to the aggregated weight of all its outgoing arcs, and a low-frequency vertex usually has a small number of such arcs. Hence, for a low-frequency vertex, the weight of each of its outgoing arcs and the vertex’s frequency are usually similar. In this situation, it is reasonable to assume that a vertex’s frequency can represent its strength in syllable word segmentation. For example, in Figure 6(a), if we set the right rank of *a* and the left rank of *b* as each vertex’s frequency, we obtain an optimal solution in this case (with the number of segmentation errors = 3). On the other hand, Figure 6(b) shows a competition graph that contains high frequency vertices. Since a high frequency vertex usually has many outgoing arcs that could share its frequency, in most cases, the vertex’s frequency cannot represent its true strength in syllable word segmentation. For example, in Figure 6(b), if we set each vertex’s left and right ranks based on the vertex’s frequency we might overestimate the right rank of *a* and underestimate the left ranks of *b*, *c*, *d*, and *e*. As a result, the solution would generate a rank assignment with 150 segmentation errors compared with an optimal solution, which only has 100 such errors. These two examples indicate that, in syllable word segmentation, a vertex’s strength and its frequency are likely to be similar if its frequency is low. Our rank pre-assignment method uses the lower part of the ranking (1 to *rank_limit*/2) for low frequency vertices, and full ranking (1 to *rank_limit*) for high frequency vertices whose ranks will be assigned by genetic algorithm later. Therefore, by pre-assigning ranks for low frequency vertices, our method can reduce the problem size; at the same time, by maintaining the full ranking for high frequency vertices, we can maintain the flexibility of solution candidates.

3.2 Genetic Algorithm for the DRAP

In this section we describe the genetic algorithm (GA) for assigning ranks to high frequency syllables. The encoding part of our GA transforms each non-assigned vertex, whose frequency is higher than *rank_limit*/2, into a single bit of a chromosome after rank pre-assignment. Each bit represents the left (resp. right) rank of a vertex in *R*(*V*) (resp. *L*(*V*)) ranges from 1 ~ *rank_limit*.

A key to the success of GA is the creation of the initial population. Generating a set of initial populations with *quality* and *diversity* is most important for our GA. In our initial population, there are two groups of chromosomes. One group is generated by randomly assigning each vertex's rank from 1 ~ *rank_limit*. The aim here is to ensure diversity; whereas the objective of the second group (as described below) is to maintain the quality of the population. Since we use the results of frequency-based method (FBM) as the baseline to evaluate the feasibility of our solution, a set of populations with the quality of FBM would be an appropriate reference for us to generate the initial population. However, FBM assigns each syllable a unique rank (namely, its frequency), which violates the ranking constraint of our problem. Moreover, since we have already pre-assigned ranks to low frequency syllables to represent their frequency (1 ~ *rank_limit*/2), it is a little challenging to build a set of initial populations with the quality of FBM under the ranking constraint. To resolve this issue, we adopt the following method:

I. Sort the chromosomes according to each vertex's frequency in non-decreasing order.
II. For each chromosome, randomly select *rank_limit*/2 disjoint intervals in the chromosome order.
III. For each chromosome, assign all bits in the first interval with value (*rank_limit*/2 + 1), assign all bits in the second interval with value (*rank_limit*/2 + 2), and so on.

Since our problem uses syllable frequency as the tie-breaker for two syllables' relations if the syllables have identical ranks, and all syllable ranks in a chromosome are higher than the ranks of pre-assigned syllables, this method can generate a set of chromosomes with similar quality as FBM. Figure 7 shows the steps of the proposed method.

Figure 7(a) shows a chromosome sorted in non-decreasing order of each syllable's frequency; and Figure 7(b) shows a chromosome in which each vertex's rank is also its frequency. Although this chromosome has the same quality as FBM, its ranking could exceed *rank_limit*. Therefore, we need to use another method to generate chromosomes with the quality of FBM. In Figure 7(c), for each chromosome, we randomly generate *rank_limit*/2 disjoint intervals and assign the same rank value to all syllables in an interval. As long as the ranks of all the syllables in a chromosome are in non-decreasing order, all the syllables and the pre-assigned syllables will yield a double rank assignment result with the quality of FBM. Therefore, we can apply this method to generate as many different chromosomes with the baseline quality as the initial population of our GA.

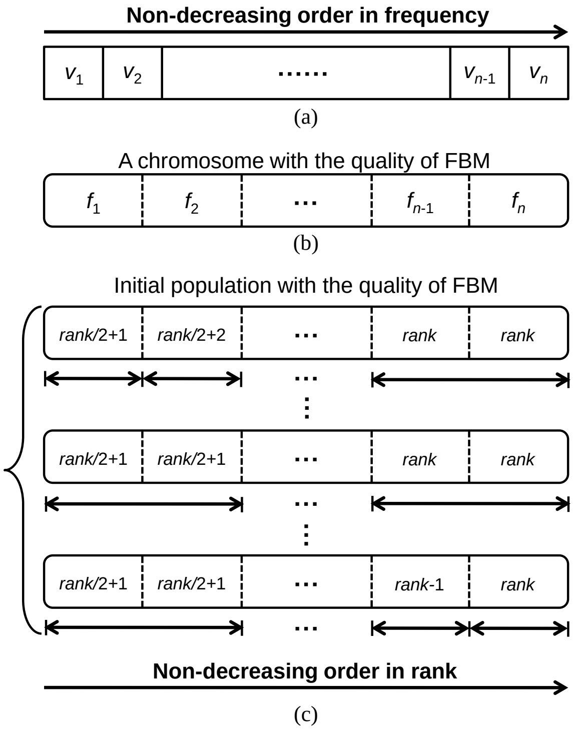

Non-decreasing order in frequency
v_1
v_2
……
v_{n-1}
v_n
(a)
A chromosome with the quality of FBM
f_1
f_2
…
f_{n-1}
f_n
(b)
Initial population with the quality of FBM
rank/2+1
rank/2+2
rank
rank
rank/2+1
rank/2+1
rank
rank
rank/2+1
rank/2+1
rank-1
rank
Non-decreasing order in rank
(c)

Figure 7. (a) A chromosome sorted according to each syllable's frequency.
(b) A chromosome in which each bit has its own rank (frequency).
(c) A set of chromosomes with the quality of FBM.

Crossover has a significant effect on the final result of the GA. Since the topology of our problem is a graph, traditional crossover approaches (one-point crossover, two-point crossover, and uniform crossover) in the GA may not be suitable. Therefore, we adopted a new crossover method called CrossNet, proposed by [Stonedahl et al. 2008]. Cross Net provides an effective crossover method for graph-like problems and it helped us generate better double rank assignment results.

The fitness function of our GA calculates the number of syllable word segmentation errors of a chromosome. A chromosome's fitness represents the quality of its double rank assignment and guides the reproduction process of our GA. We use the tournament selection scheme, which randomly picks two chromosomes and selects the one with lower fitness as well as fewer errors in the reproduction process.

4. EXPERIMENTS

To compare with our method, we consider the frequency-based method (FBM) and the conditional random fields (CRF) [Lafferty et al. 2001] approach. The latter is state-of-the-art technique on modern CWS [Zhao et al. 2010]. We conducted two sets of experiments: toneless and tonal. For each syllable, we apply DRA to determine the effect of our double ranking strategy compared with that of FBM and CRF.

4.1 Data Set

In this study, we use the Xinhua Agency part of Chinese Gigaword Third Edition (Gigaword in short, hereafter) [Graff 2007] as training set and closed test set, and segment it automatically into approximately 279,500,000 words in simplified Chinese, by Peking University's segmenter (PKU segmenter) [Duan et al. 2003]. We then apply the HowNet dictionary to convert Gigaword into a Pinyin corpus. Two Pinyin polysyllables are said to be in conflict if they overlap. Since Gigaword is already segmented, a Pinyin polysyllable is deemed selected if it conforms to the word boundary of the corpus. Thus, for any two overlapping Pinyin polysyllables *u* and *v*, we can calculate the number of times *u* is selected over *v* and vice versa in word segmentation. Based on the Pinyin corpus, we select a Pinyin syllable *t*, and build a competition graph $G = (V, E)$. There are 390 types of toneless and 1,198 types of tonal monosyllables converted from Gigaword, and they consequently produce overlapping polysyllables in 116,681 types without tone and 28,533 types with tone, respectively.

The corpora for independent (open) test are from the Third International CWS Bakeoff of the Special Interest Group of the Association for Computational Linguistics (SIGHAN) [Levow 2006]. For a study on Pinyin syllable, four corpora in simplified Chinese are chosen: the training set and test set from Microsoft Research (denoted by "MSR-training" and "MSR-test", respectively), and the training set and test set from Peking University (denoted by "PKU-training" and "PKU-test", respectively). The Pinyin conversion gets 391, 197, 197, and 199 types of toneless monosyllables in MSR-training, MSR-test, PKU-training, and PKU-test, respectively. For the coverage of monosyllable types, MSR-test, PKU-training, and PKU-test are not large enough. Even worse, TOAS of syllables are not found in these four corpora, which is the main reason of using Gigaword as training corpus instead of SIGHAN corpora.

Consequently, issues about segmentation standards arise. PKU-training and PKU-test may share a certain level of consistency with Gigaword that was segmented by PKU segmenter, but MSR-training and MSR-test may not. However, although segmentation criterions in MSR and PKU corpora are different, these differences are mostly in CAS rather than in OAS, which means that experiments of overlapping syllables will not be much affected. For example, once some standard treats 军事界 (military area) as a whole segment instead of two segments as 军事 (military) and 界 (area), the corresponding tri-syllable "jun-shi-jie" may disagree with previous outcome of competition via training set for its substring "jun-shi" to "陆军(army) / 是 (is)" for "lu-jun-shi", and this situation could be a test for robustness with the problem definition remaining intact. For a similar concern, although we are aware of the existence of Tagged Chinese Gigaword Version 2.0 [Huang 2009], it could introduce a more complicated relationship between word segmentation standards, since it is based on heterogeneous CKIP and ICTCLAS tagging systems [Huang 2008]. Nevertheless, according to Wen et al. [2008], errors in SWS caused by CAS mostly will not have any influence on the final CPIM results.

The choice of using large and (semi-)automatic segmented corpora instead of small and manually segmented ones is a common compromise between the ideal and the reality of CPIM studies. Most previous works of Pinyin-to-character conversion involved in-house corpora. A series of language model studies involving CPIM is based on large, balanced, yet not publicly available corpora as training set and an independent open test set that is from different sources of the training set [Gao et al. 2002; Gao et al. 2006], similar to this paper, to ensure the experiment is pragmatic. In fact, while CPIM studies are usually not evaluated with SIGHAN's or other manually segmented corpora in terms of precision and recall, there is simply no way to make a comparative study of CPIM for the time being. Instead of splitting a relative small and manually segmented corpus into training set, development (held-out) set, and test set for common parameter tuning scheme such as cross-validation or held-out estimation, CPIM researchers tend to investigate performance in a pragmatic way that data sets comprise texts in different domains, styles, and time.

4.2 Experiments on Pinyin Syllables

For each toneless syllable in our corpus, we generate a competition graph and solve DRAP on the graph. For each such syllable, we test FBM, CRF, and DRA and count the number of syllable word segmentation errors as Penalty for each method. Common evaluation metrics of CWS, such as word-based precision, recall, and their harmonic average F_1-score, do not fit SWS, because overlapping syllables do not have unique choices of segmentations, as mentioned in Section 1.2, to be gold standard for precision/recall/F_1-score calculations.

A series of experiments are conducted by linear-chain CRF since it appears to be very effective on sequential labeling problems including CWS. The character-based tagging scheme [Xue 2003; Zhao et al. 2010] is adopted in CRF as monosyllable-based one for short syllable word segmentation. Configurations of tag set and feature set are similar to works for SIGHAN Bakeoffs [Low et al. 2005; Peng et al. 2004; Tsai et al. 2005; Tseng et al. 2005; Xue and Shen 2003; Zhang et al. 2006; Zhao et al. 2010]. Table 3 and Table 4 in the following provide the feature templates in the format of CRF++[†] and samples of annotated training data for each configuration, respectively. Specifically, the configuration CRF$_6$ listed in Table 4 is the-state-of-the-art of CWS [Zhao et al. 2010], therefore its CRF parameter of Gaussian prior (c=100 in the usage of CRF++) is applied. In the interest of brevity and clarity, we do not draw huge tables or charts of preliminary experiments for hyper-parameter (*i.e.* the Gaussian prior) tuning or for the context window size of feature templates and additional features. The preliminary experiments use 5-fold cross-validation to tune parameters or to select features. Like related works, context window sizes larger than 3 monosyllables do not help much, especially when the tag set applies more monosyllable-position types [Zhao et al. 2010]. Therefore feature templates in Table 3 exclude tri-syllabic compounds and do not exceed the position -2 or 2. Additional features such as accessor variety substrings [Feng et al. 2005; Zhao and Kit 2011] or word type indicators [Tseng et al. 2005] are not employed since interactions of their combinations can be complicated and may be beyond the scope of this work,

[†] Taku Kudo. 2005. CRF++, version 0.54. http://crfpp.sourceforge.net/ (Retrieved on 2011/10)

which has no intentions to elaborate upon the feature engineering for CRF. Although there’s always a certain chance that sophisticated features make CRF invincible, the price those features paid could still be a weakness when this work would like to highlight applications on resource limited devices. Hence CRF experiments in this work are not for competitions but for relative benchmarks. It is worth noting that specific word type indicators those informing CRF where the overlapping occurred had been considered, but in our experiences, their contributions to character-based segmentation models would be negative sometimes. That’s also one of the reasons why this work categorizes CRF experiments by tag set, to make the relative benchmarks purely based on monosyllable for modeling overlapping ambiguities.

Table 3. Feature templates for CRF configurations.

CRF++ template	Meaning
U0:%x[0,0] U1:%x[-1,0] U2:%x[1,0] U3:%x[-2,0] U4:%x[2,0]	The monosyllable at the position relative to current monosyllable, where a positive value indicates next/right monosyllable and a negative value means previous/left one, is going to be associated with the output tag of current monosyllable as unigram.
U5:%x[-1,0]/%x[0,0] U6:%x[0,0]/%x[1,0] U7:%x[-2,0]/%x[-1,0] U8:%x[1,0]/%x[2,0] U9:%x[-1,0]/%x[1,0]	The monosyllable pair at positions to relative current monosyllable is going to be associated with the output tag of current monosyllable as unigram.
B	The current monosyllable is going to be associated with the output tags of current monosyllable and previous monosyllable as bigram.

Table 4. Samples of annotated training data for CRF configurations.

Tag Set (subscription *n* indexes the number of tag type)	Sample of Annotated Training Data					
	Bai	Fen	Zhi	Wu	shi	wei
CRF$_2$ [Peng et al. 2004]	B	I	I	I	I	B
CRF$_3$ [Zhang et al. 2006]	B	I	I	I	I	S
CRF$_4$ [Xue and Shen 2003]	B	I	I	I	E	S
CRF$_5$ [Zhao et al. 2010]	B	1	I	I	E	S
CRF$_6$ [Zhao et al. 2010]	B	1	2	I	E	S

The syllable sequence “bai-fen-zhi-wu-shi-wei” could be segmented as “bai-fen-zhi-wu-shi/wei” (百分之五十 (fifty percent) / 为 (is)), or “bai-fen-zhi-wu/shi-wei” (百分之五 (five percent) / 视为 (seen as)). Instead of tagging every occurrence of these conflicting patterns in the whole corpus, only the more frequent one (the former), is annotated to be a training sample as Table 4 illustrated. This is a necessary procedure of feature selection, because the training corpus is too large to compute for CRF pragmatically.

In Table 5, we list the experiment results for the top-10 most frequent toneless Pinyin monosyllables. This table details the following results: Penalty generated by FBM as baseline; Penalty generated by DRA$_{10}$, DRA$_{20}$, DRA$_{40}$, and DRA$_{80}$; Penalty generated by each CRF configurations (denoted by “CRF” with the number of tag type of corresponding tag set indexed the same way as in Table 4). Boldface indicates the best case of each row in Penalty, and bold-italic style represents the best performance that CRF control group can reach. The results show that FBM and CRF$_6$ increase total Penalties by 106.3% and 43.7% than DRA$_{20}$, respectively.

We test syllable ranks generated from our corpus on open (independent) test corpora to assess the feasibility of DRA. Since Table 5 shows that performances of DRAs are not sensitive to ranks, DRA_{20} is selected for the rest of experiments. For comparison, we also apply FBM using each syllable's frequency in our training corpus on the same test corpora.

Table 5. The experiment results for the top-10 most frequent toneless Pinyin monosyllables.

Syllable	FBM	DRA_{10}	DRA_{20}	DRA_{40}	DRA_{80}	CRF_{6}	CRF_{5}	CRF_{4}	CRF_{3}	CRF_{2}
de	39302	**21659**	22396	21741	22743	45833	***45747***	46361	82946	325797
Shi	471542	**281907**	287214	297810	286976	***316311***	323665	357803	622725	1986781
Yi	291534	147251	**130554**	132848	140551	***181982***	256288	276184	449684	1227310
Ji	292688	173814	169710	**169448**	175147	***169635***	173056	204114	426371	1432339
Guo	133336	60193	57329	**54536**	57794	132065	159731	***123031***	316715	1262814
Zhi	235900	124084	**115495**	124363	127584	***129397***	137277	163045	310392	851270
zhong	114712	58224	**57680**	58227	63579	***114286***	176187	154103	327565	530140
Li	164808	89833	**89302**	93090	101007	***102011***	100815	111927	251443	1126689
He	51938	**23688**	25433	26378	28493	70858	***67878***	83408	214723	974709
Wei	98907	58065	**55041**	56662	59555	59164	***57903***	63150	112060	461408
Top-10 Subtotal Penalty	1894667	1038718	**1010154**	1035103	1063429	***1321542***	1498547	1583126	3114624	10179257
Top-10 Subtotal Ratio	1.876	1.028	1.000	1.025	1.053	1.308	1.483	1.567	3.083	10.077
Total Penalty	7502828	3700967	**3637509**	3778074	3961623	***5227524***	5476084	5720357	11251366	50430942
Total Ratio	2.063	1.017	1.000	1.039	1.089	1.437	1.505	1.573	3.093	13.864

Since the test corpora contain the correct segmentation of each un-segmented Pinyin syllable, we could compare DRA with other methodologies such as FBM by segmenting the un-segmented Pinyin syllable in the corpora. Figure 8 illustrates how we use the generated double ranks in our corpus for syllable word segmentation in the test corpora. In Figure 8(a), we show the correct segmentation of an un-segmented Pinyin syllable "*a-b-c-d*," where "*a*," "*b*," "*c*," and "*d*" are

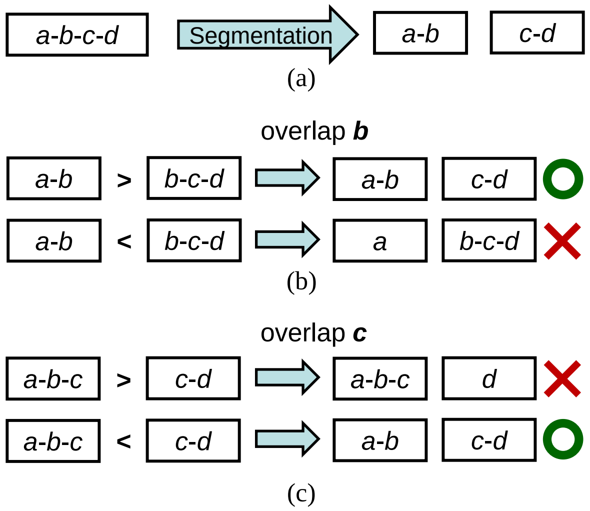


Figure 8. (a) The correct segmentation of "*a-b-c-d*."
(b) Segmentations of overlapping Pinyin syllable "*b*."
(c) Segmentations of overlapping Pinyin syllable "*c*."

Pinyin monosyllables, and "*a-b-c-d*" is segmented into "*a-b*" and "*c-d*." As the segmentation point is located between "*b*" and "*c*," we only consider the cases where two Pinyin polysyllables overlap "*b*" or "*c*." Figure 8(b) shows the case where two Pinyin polysyllables, "*a-b*" and "*b-c-d*," overlap "*b*." In this case, DRA compares the right rank of "*a-b*" and the left rank of "*b-c-d*" to determine which syllable is selected. In contrast, FBM compared the frequency of "*a-b*" and the frequency of "*b-c-d*" to determine which syllable is selected. After a syllable has been selected, we can assess whether the segmentation is correct. For example, in Figure 8(b), the upper segmentation that selects the syllable "*a-b*" and matches the correct segmentation is correct; and the lower segmentation that selects the syllable "*b-c-d*" and mismatches the correct segmentation is wrong. Therefore, we could count Penalty for DRA and FBM. Figure 8(c) shows another case where two Pinyin polysyllables, "*a-b-c*" and "*c-d*," overlap "*c*." Similarly, we count Penalty for DRA and FBM. Then, we test models that are trained from Gigaword in the previous experiments by FBM, CRF and DRA on the four simplified Chinese corpora from SIGHAN's 3rd CWS Bakeoff. The results are listed in Table 6.

Table 6. The experiment results of FBM and DRA for syllable word segmentation of four corpora from the 3rd CWS Bakeoff of SIGHAN 2006**.**

Corpus	FBM	DRA$_{20}$	CRF$_6$	CRF$_5$	CRF$_4$	CRF$_3$	CRF$_2$
MSR-training	248092	77945	**71029**	***73040***	73393	106869	281592
MSR-test	20133	6215	***5595***	5663	**5570**	8686	22755
PKU-training	96319	33034	**30407**	***30901***	31703	43947	111267
PKU-test	26505	**9116**	***9308***	9567	9650	13676	35223
Total	391049	126310	**116339**	***119171***	120316	173178	450837
Ratio	3.061	1.000	0.921	0.943	0.953	1.371	3.569

The experiment results show that, in the four corpora, DRA could reduce Penalty of FBM by 206.1%. Since the corpus we used to derive each syllable's double ranks (*i.e.*, Gigaword) is totally independent of the four test corpora, the experiment results clearly demonstrate the robustness of the double ranking strategy.

To see the effects of DRA on tonal syllables, we conduct similar experiments. We still used Gigaword as our test corpus with tonal syllables. The experiment results for the top 10 most frequent tonal Pinyin monosyllables are listed in Table 7.

Table 7. The experiment results for the top-10 most frequent tonal Pinyin monosyllables.

Syllable	FBM	DRA$_{20}$	CRF$_6$	CRF$_5$	CRF$_4$	CRF$_3$	CRF$_2$
de5	2581	2203	**131**	***206***	***206***	613	33980
shi4	93818	**47642**	64916	***58365***	87294	126013	845398
zai4	***6163***	**2453**	6727	6920	6721	6548	122709
he2	***20886***	**12767**	58014	32461	30493	38531	263519
guo2	***36895***	**18153**	74882	82702	80681	241411	1045328
yi1	***60952***	**16313**	85387	64307	85126	121211	402644
le5	0	0	0	0	1	51	27416
bu4	64021	**27252**	***28023***	30920	31730	47359	428127
zhong1	***29810***	**14748**	86946	55287	54010	112139	280140
dui4	13710	***6087***	8525	**5370**	9135	9337	216565
Top-10 Subtotal Penalty	***328836***	**147618**	413551	336538	385397	703213	3665826
Top-10 Subtotal Ratio	2.228	1.000	2.801	2.280	2.611	4.764	24.833
Total Penalty	***2651488***	**1112975**	3383088	2924905	3169910	5729006	46986907
Total Ratio	2.382	1.000	3.040	2.628	2.848	5.147	42.217

The results show that DRA is still effective on the tonal Pinyin syllable word segmentations according to Penalty.

4.3 Discussions

Note that in Table 7, the row “le5” of 了 (an expletive), every method conduct 0 Penalty. This is because “le5” of 了 can only be used as an expletive and is always segmented into an single Pinyin word in Chinese, there is no cycle in the competition graph formed by “le5” of 了. CRF$_4$, CRF$_3$, and CRF$_2$, however, suffer for monosyllable-based modeling that may synthesize syllables into segmentations that are unseen in the training data. This fact implies that lower Penalty indicates higher performance of SWS, and subsequently implies better user experiences of CPIM.

One may be curious why DRA does not perform better than CRF on the independent tests shown in Table 6. This phenomenon actually indicates one of the main differences between DRA and CRF, *i.e.* polysyllable (word) based matching method vs. monosyllable (character) based discriminative model, where the former does not solve out-of-vocabulary (OOV) problem directly while the later usually concatenates unseen compounds that happen to be unknown words conveniently. DRA, as the proposed method of this work, is not designed to resolve overlapping ambiguities and recognize unknown words simultaneously. To make a clearer picture of overlapping ambiguity resolutions on comparisons between DRA and CRF, Table 8 lists in-vocabulary (IV) penalties on the independent tests, and it turns out that DRA is a lot better.

Table 8. The experiment results in terms of Penalty$_{IV}$ on DRA$_{20}$ and CRF$_6$ for syllable word segmentation of four corpora from the 3rd CWS Bakeoff of SIGHAN 2006**.**

Corpus	DRA$_{20}$	CRF$_6$
MSR-training	**29116**	33437
MSR-test	**2364**	2909
PKU-training	**13076**	13954
PKU-test	**3780**	4066
Total	**48336**	54366
Ratio	1.000	1.125

The OOV rates in terms of overlapping polysyllable pairs according to the training data Gigaword version 3 of MSR-training, MSR-test, PKU-training, and PKU-test are 26.37%, 23.64%, 28.47%, and 29.21%, respectively. For CRF models, advantages and disadvantages both come from monosyllable concatenations that can be either unknown polysyllables luckily or artificial ones unfortunately, so statistics and examples are provided in Table 9.

Table 9. Errors of Artificial Segmentations (according to Gigaword v3) that CRF$_{6}$ synthesized against overlapping syllable word segmentations.

Corpus	Proportion of Penalty on Artificial Segmentations	Example of Artificial Segmentations	Real Segment
Gigaword v3	10.74%	shi-shi-jie	shi / shi-jie, shi-shi / jie
MSR-training	16.15%	dang-zhong-yang	dang / zhong-yang
MSR-test	20.39%	she-hui-zhu-yi	she-hui / zhu-yi
PKU-training	15.97%	gao-ke-ji	gao / ke-ji
PKU-test	10.60%	de-fu-ze-ren	de / fu-ze-ren

For instance, the existent Pinyin segmentation “de / fu-ze-ren” is likely to be “的 (of) / 负责人 (a person in charge)” that can never be a whole syllable word no matter which segmentation standard is applied, since “de” for “的 (of)” is almost always a bound morpheme. Our training data of Gigaword v3 have been filtered to consist of segmentations have overlapping ambiguities only, which means each segmentation comprise exactly two syllable words. However, linear-chain CRF models sometimes tend to synthesize consecutive high frequency unigrams and/or bigrams of monosyllables into a single syllable word, no matter that word is an artificial one or not according to the training data.

Besides the OOV issue, we further speculate that the reason DRA outperforms CRF is due to the differentiation of the left and right context. Although linear-chain CRF is able to learn context via expanding the window size of feature templates and increasing the variety of prediction label, the weights still come from undirected relationship of context, *i.e.* frequency. For the example mentioned in Section 1.3, a linear-chain CRF may need longer chunks to get better results, while short chunks with directed graph of left and right context is good enough for DRA. On overlapping short syllable word segmentation as this work defined, CRF has no choice but memorize unigram and bigram mechanically. Table 10 lists some high-Penalty cases that DRA predicted better than CRF.

Table 10. High Penalty cases that DRA$_{20}$ outperforms CRF$_{6}$

CRF$_{6}$ Segmentation	CRF$_{6}$ Segmentation Count	DRA$_{20}$ Segmentation	DRA$_{20}$ Segmentation Count	DRA$_{20}$ Double t Ranks
ge-guo / jia	255	ge / guo-jia	40716	ge-guo:18 < 19:guo-jia
you / qi-shi	282	you-qi / shi	20424	you-qi:19 > 16:qi-shi
deng-fang / mian	58	deng / fang-mian	37576	deng-fang:6 < 14:fang-mian
yi-gong / jin	22	yi / gong-jin	8057	yi-gong:17 < 19:gong-jin
bu-fu / he	127	bu / fu-he	5407	bu-fu:16 < 19:fu-he

For example, the polysyllable “you-qi-shi” can be either “you-qi / shi” for “尤其 (especially) / 是 (is)” or “you / qi-shi” for “有 (have) / 启示 (inspiration)” that DRA ranked right hand side strength of “you-qi” higher than left hand side strength of “qi-shi” while CRF chose “you / qi-shi” as the segmentation because “qi-shi” can be a high frequency segment for “其实(actually)” if the left and right contexts were not evaluated.

The fact that DRA performs better than CRF on in-vocabulary SWS of short strings does not imply DRA would perform worse than CRF on longer strings. For example, a recent CPIM evaluation showed that context length and performance do not necessarily have positive correlations [Jiang et al. 2011].

4.4 Space and Time Requirements

Since this study attempts to strike a balance between the cost of computing resource and the benefit of SWS performance, space requirement could be one of evaluation criterion. However, space requirement may vary from system to system, depending on implementation. Especially for monosyllable-based CRF and polysyllable-based FBM and DRA, the scales of input units are quite different. To make a fair comparison, we calculate the sizes of raw files for each model, without compressions, as the conceptual benchmark of space requirement. For toneless Pinyin, the file size of FBM is about 4.97MB. DRA consumes 5.41MB for identical syllables within FBM and additional small space for double ranks. CRF models as the control groups in the experiment require about 24.2MB – 35MB, depending on the label combinations of character, tag set and feature template. Zhao et al. [2010] reasons that the space requirement of CRF using L-BFGS algorithm is in the same scale with the time complexity of a single CRF training iteration, which is shown by Cohn et al. [2005] as $O(n^2)$ where n is the number of label combinations. For tonal Pinyin, while file sizes of FBM and DRA model both increase slightly to about 5.92MB and 6.53MB, respectively. However, the file sizes of CRF model inflate dramatically to 100MB – 154MB. These facts, as listed in Table 11, indicate that DRA effectively outperforms FBM with similar scale of space complexity while maintains competitive performance to CRF in a much more efficient way.

Table 11. Model sizes of FBM, DRA and CRF.

	FBM	DRA	CRF$_6$	CRF$_5$	CRF$_4$	CRF$_3$	CRF$_2$
Toneless Model Size (MB)	4.97	5.41	35.0	32.3	29.6	26.9	24.2
Tonal Model Size (MB)	5.92	6.53	154	140	127	113	100

While CRF-based SWS spends $O(n^2)$ for Viterbi algorithm, where n stands for the number of monosyllables (characters) of a given input string, DRA and FBM based SWS need only $O(n)$ roughly for a proper greedy algorithm (e.g. forward maximum matching) scanning syllable words from the input string in situations similar to Figure 8.

4.5 Intelligent Memorization

Based on the observation on error cases of CRF and the efficiency on space requirement of DRA, this work further suggests that DRA can use spare storage to keep track of error cases according to the training data, to memorize high-Penalty errors intelligently rather than cram up all combinations of monosyllable n-grams. By balancing the trade-off between space and performance, one may decide how many cases are sufficient to load in resource limited devices. One of the most intuitive ways to do so is, first, sorting error cases proportionally by Penalty-byte rate, and then record preferred segmentations one by one until the limitation of space or the expectation of reduced Penalty is reached. Figure 9 and Figure 10 illustrate the trend of accumulated Penalty reduction and space requirement, respectively, on the top 10,000 error cases. According to these two charts, one may get a highly fitted exemplar set, which can potentially reduce Penalty almost to zero in the closed test, by recalling top 10,000 preferred segmentations as errata that only spend extra 140 Kbytes to store! To provide a more informative analysis, Table 12 lists top-10 error cases of DRA$_{20}$ proportionally by the Penalty-byte rate. Clearly, most errors are caused by high double ranks and relatively high frequencies of competing syllables while strong preferences for one of segmentations on overlapping ambiguities are still there.

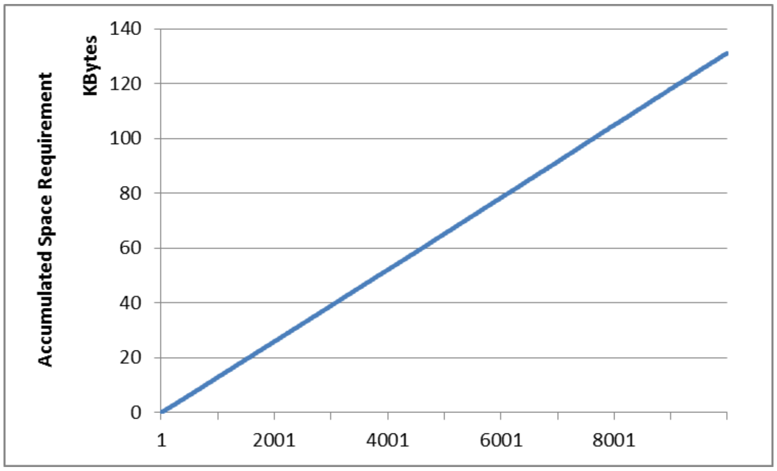


Figure 10. Accumulated space requirement of top 10,000 error cases sorted by Penalty-byte rate according to the closed test on Gigaword v3

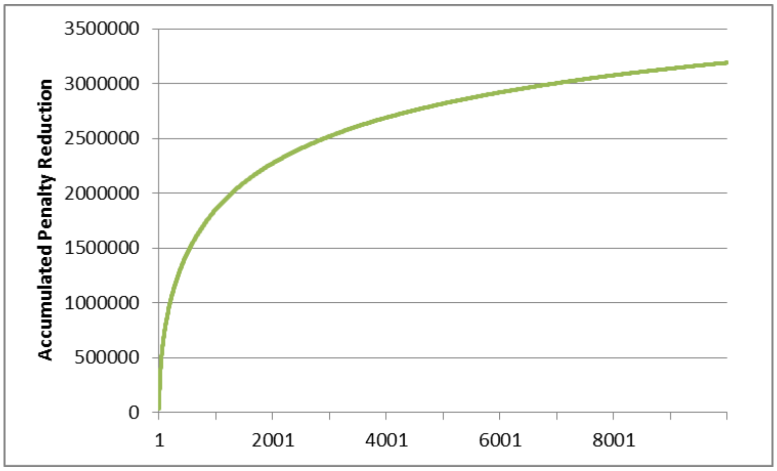


Figure 9. Accumulated penalty reduction of top 10,000 error cases sorted by Penalty-byte rate according to the closed test on Gigaword v3

Table 12. Top 10 Penalty-byte rate error cases of DRA$_{20}$ according to the closed test on Gigaword v3

Penalty-	Preferred	Preferred	Alternative	Alternative	Left	Right	Left	Right	Left	Right
byte Rate	Segmentation	Count	Segmentation	Count	Segment	Rank	Count	Segment	Rank	Count
3093.455	de / li-shi	34028	de-li / shi	36	de-li	19	3859	li-shi	18	142877
2450.7	di-yi / ge	24507	di / yi-ge	72	di-yi	19	229393	yi-ge	19	366570
2189.769	zai / ci-jian	28467	zai-ci / jian	34	zai-ci	19	45333	ci-jian	18	45931
2163.333	zuo-chu / le	25960	zuo / chu-le	3	zuo-chu	18	115801	chu-le	19	24169
1740.286	cai-fang / shi	24364	cai / fang-shi	6	cai-fang	18	45548	fang-shi	19	72069
1558.429	ge / fang-mian	21818	ge-fang / mian	2	ge-fang	16	637	fang-mian	14	194397
1434	de / mu-de	14340	de-mu / de	5	de-mu	13	62	mu-de	8	26
1428.846	chan-pin / de	18575	chan / pin-de	2	chan-pin	18	162671	pin-de	19	1492
1225.385	de / li-chang	15930	de-li / chang	1	de-li	19	3859	li-chang	18	31716
1208.636	guo-qu / de	13295	guo / qu-de	37	guo-qu	19	71659	qu-de	19	143282

Although the cost of intelligent memorization is relatively low, its performance for the open (independent) test on IV still concerns us. Hence Figure 11 shows the utilization of top 10,000 errata, which demonstrates trends similar to the closed test set, while Table 13 lists the percentages of improvements for Penalty$_{IV}$ according to statistics in Table 8. Both of them suggest that intelligent memorization is effective and stable.

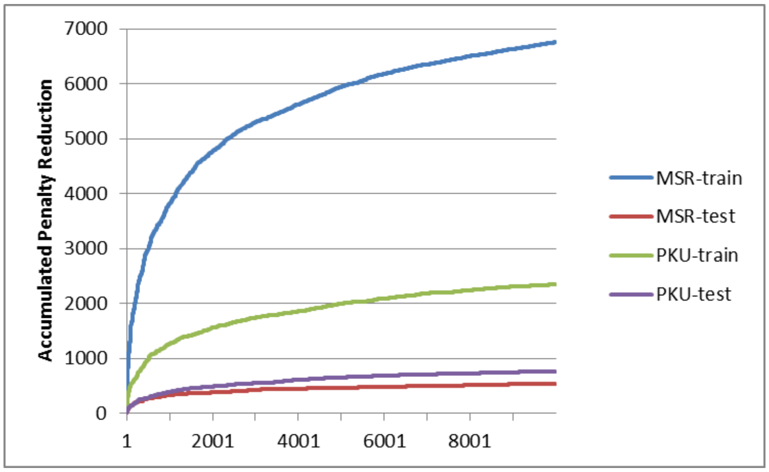


Figure 11. Accumulated penalty reduction of top 10,000 intelligent errata for syllable word segmentation on four corpora from the 3rd CWS Bakeoff of SIGHAN 2006

Table 13. The experiment results in terms of Penalty$_{IV}$ reduction according to Table 8 on the top 10,000 intelligent errata for syllable word segmentation of four corpora from the 3rd CWS Bakeoff of SIGHAN 2006**.**

Corpus	Penalty$_{IV}$ Reduction	Penalty$_{IV}$ Reduction Rate
MSR-training	**6755**	23.20%
MSR-test	**537**	22.72%
PKU-training	**2352**	17.99%
PKU-test	**768**	20.32%
Overall	**10412**	21.54%

5. CONCLUSIONS

In this paper, we propose a double ranking strategy for overlapping syllable word segmentation in short texts. The experiment results show that the strategy can reduce Penalty by 90.2% segmentation of toneless overlapping Pinyin polysyllables using FBM. In addition, the results of experiments on independent corpora and the segmentation of tonal syllables further demonstrate the feasibility and robustness of the double ranking strategy.

As we mentioned in the abstract, there are usually two stages in a CPIM: 1. segment the syllable sequence into syllable words; 2. select the most likely character words for each syllable word. Being able to do the SWS task in (1) well, the character word selection task in (2) only needs to deal with homophones with the same delimiters, which would make this two-stage approach much better and simpler than the alternative, namely, intermingle segmentation with word selection. By the way, the task in (2) is very similar to multi-stage part-of-speech (POS) tagging or word sense disambiguation. In fact, literature of Chinese POS tagging concluded that although "all-at-once" models may work a little better than multi-stage ones, the costs of the former are always much higher than those of the later [Zhang and Sun 2011]. Sometimes, a well-designed multi-stage system can be even more accurate than a joint model system since the joint model usually faces a large and complex search space that makes fine-tuning more difficult or even intractable [Sun 2011]. This is also why we do not process OOV and prefer to have separate stages for unknown word detection and named entity recognition.

We believe a similar strategy could also be adopted to disambiguate conflicting linguistic patterns effectively. Linguistic patterns are important features in natural language processing. In machine learning algorithms, it is customary to train a specific weight for each feature. Given a test sentence, the features' weights are aggregated to find an optimal combination. However, in some cases, the text could be short and incomplete, and therefore not amenable to full-fledged analysis. As [Sproat and Emerson 2003] pointed out, the handling of short strings with minimal context, such as queries submitted to a search engine, has only been studied indirectly. When two patterns in a short text overlap, disambiguation based on one fixed weight for each feature does not necessarily yield the best result, in which case the double ranking strategy could be considered.

There are many other extensions that the double ranking strategy can be considered, which will be the topics for future research.

6. ACKNOWLEDGEMENT

This research was supported in part by the National Science Council under grant NSC 100-2631-S-001-001, NSC 100-2319-B-010-002, and the research center for Humanities and Social Sciences under grant IIS-50-23.

REFERENCES

CHEN, Z. AND LEE, K.-F. 2000. A new statistical approach to Chinese Pinyin input. In Proceedings of ACL. 241-247.

DIETTERICH, T. G. 2002. Machine Learning for Sequential Data: A Review. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, Terry Caelli, Adnan Amin, Robert P. W. Duin, Mohamed S. Kamel, and Dick de Ridder (Eds.). Springer-Verlag, London, UK, UK, 15-30.

DONG, Z.- D. AND DONG, Q. 2006. HowNet and the Computation of Meaning. World Scientific Pub Co Inc.

DUAN, H.-M., BAI X.-J., CHANG, B.-B., AND YU, S.-W. 2003. Chinese word segmentation at Peking University. In *Proceedings of the second SIGHAN workshop on Chinese language processing - Volume 17* (SIGHAN '03), Vol. 17. Association for Computational Linguistics, Stroudsburg, PA, USA, 152-155. DOI=10.3115/1119250.1119272 http://dx.doi.org/10.3115/1119250.1119272

EVEN, G., NAOR, J., SCHIEBER, B., AND SUDAN, M. 1998. Approximating Minimum Feedback Sets and Multi-Cuts in Directed Graphs. *Algorithmica*, 20, 2, 151-174.

GAO, J.-F., GOODMAN, J., LI, M., AND LEE, K.-F. 2002. Toward a unified approach to statistical language modeling for Chinese. *ACM Trans. Asian Lang. Inform. Process.*, 1, 1, 3-33.

GAO, J.-F., SUZUKI, H., AND YUAN, W. 2006. An empirical study on language model adaptation. *ACM Trans. Asian Lang. Inform. Process*., 5, 3, 209-227.

GAO, J.-F. AND ZHANG, M. 2002. Improving language model size reduction using better pruning criteria. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (ACL '02). Association for Computational Linguistics, Stroudsburg, PA, USA, 176-182. DOI=10.3115/1073083.1073114

GOODMAN, J. AND GAO, J.-F. 2000. Language Model Size Reduction By Pruning And Clustering. In *Proceedings of ICSLP'00*, Beijing, China.

GRAFF, D. 2007. Chinese Gigaword Third Edition. *Linguistic Data Consortium, Philadelphia*. Catalog Number LDC2007T38.

GUO, J., HÜFFNER, F., AND MOSER, H. 2007. Feedback Arc Set in Bipartite Tournaments is NP-Complete. *Information Processing Letters*, 102, 2-3, 62-65.

GUPTA, S. 2008. Feedback Arc Set Problem in Bipartite Tournaments. *Information Processing Letters*, 105, 4, 150-154.

HUANG, C.-R. LEE, L. –H., QU, W.-G., AND YU, S.-W. 2008. Quality Assurance of Automatic Annotation of Very Large Corpora: a Study Based on Heterogeneous Tagging Systems. In *Proceedings of the 6th International Conference on Language Resources and Evaluation* (*LREC'08*), Marrakech, Morocco.

HUANG, C.-R. 2009. Tagged Chinese Gigaword Version 2.0. *Linguistic Data Consortium, Philadelphia*. Catalog Number LDC2009T14

JIANG, M. T.-J., LEE, C.-W., LIU, C., CHANG, Y.-C., AND HSU W.-L. 2011. Robustness Analysis of Adaptive Chinese Input Methods. In *Proceedings of the Workshop on Advances in Text Input Methods* (*WTIM 2011*), 53-61. Chiang Mai, Thailand.

KARP, R. M. 1972. Reducibility among Combinational Problems. *Complexity of Computer Computations*, Plenum Publishing Corporation.

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

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

LI, M., GAO, J.-F., HUANG, C.-N., AND LI, J.-F. 2003. Unsupervised training for overlapping ambiguity resolution in Chinese word segmentation. In *Proceedings of the second SIGHAN workshop on Chinese language processing 17 (SIGHAN '03).* Association for Computational Linguistics, Stroudsburg, PA, USA, 1-7.

LI, R., LIU, S.-H., YE, S.-W., AND SHI, Z.-Z. 2001. A Method of Crossing Ambiguities in Chinese Word Segmentation Based on SVM and k-NN (In Chinese). *Journal of Chinese Information Processing*. 15, 6, 13-18

LI, L., WANG, X., WANG, X.-L., AND YU, Y.-B. 2009. A Conditional Random Fields Approach to Chinese pinyin-to-character Conversion. *Journal of Communication and Computer*, 6, 4, 25-31.

LIANG, N.-Y. 1987. A written Chinese automatic segmentation system (In Chinese). *Journal of Chinese Information Processing*, 2, 44-52.

LIU, Y., WANG, Q.-Q. 2007. Chinese Pinyin Phrasal Input on Mobile Phone: Usability and Developing Trends. In *Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology* (Mobility '07). ACM, New York, NY, USA, 540-546. DOI=10.1145/1378063.1378151

LIU, B.-Q. AND WANG, X.-L. 2002. An approach to machine learning of Chinese Pinyin-to-character conversion for small-memory application. In *Proceedings of the First International Conference on Machine Learning and Cybernetics*. Beijing, China , 1287-1291.

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

MACKENZIE, S. I. AND SOUKOREFF, W. R. 2002. Text Entry for Mobile Computing: Models and Methods, Theory and Practice. *Human Computer Interaction*, 17, 2, 147-198.

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

QIAO, W., SUN, M.-S., AND MENZEL, W. 2008. Statistical Properties of Overlapping Ambiguities in Chinese Word Segmentation and a Strategy for Their Disambiguation. In *Proceedings of the 11th International Conference on Text, Speech and Dialogue (TSD '08)*. Springer-Verlag, Berlin, Heidelberg, 177-186.

SPROAT, R. AND EMERSON, T. 2003. The First International Chinese Word Segmentation Bakeoff. In *Proceedings of SIGHAN Workshop on Chinese Language Processing (SIGHAN '03).* Association for Computational Linguistics, Stroudsburg, PA, USA, 133-143.

SPROAT, R. AND SHIH, C.-L. 2001. Corpus-Based Methods in Chinese Morphology and Phonology. Technical report, Linguistic Society of America Summer Institute, Santa Barbara, CA, USA.

STOLCKE, A. 2000. Entropy-based pruning of backoff language models. Technical report, Speech Technology and Research Laboratory, SRI International, Menlo Park, CA, USA. arXiv:cs/0006025

STONEDAHL, F., RAND, W., AND WILENSKY, U. 2008. CrossNet: A Framework for Crossover with Network-based Chromosomal Representations. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation (GECCO '08).* ACM, New York, NY, USA, 1057-1064.

SUN, M.-S. AND ZUO, Z.-P. 1998. Overlapping Ambiguity in Chinese Text (In Chinese). *Quantitative and Computational Studies on the Chinese Language.* HK. 323-338.

SUN, W. 2011. A stacked sub-word model for joint Chinese word segmentation and part-of-speech tagging. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*. 1385-1394.

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

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

WANG, X., LI, L., YAO, L., ANWAR, W. 2006. A Maximum Entropy Approach to Chinese Pin Yin-To-Character Conversion. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics, 2006 (SMC '06)*. doi: 10.1109/ICSMC.2006.384567

WANG, Y.-H., SU, H.-J., AND MO, Y. 1990. Automatic Processing of Chinese Words. *Journal of Chinese Information Processing*, 4, 4, 1-11.

WARD, D. J., BLACKWELL, A. F., AND MACKAY, D. J. C. 2000. Dasher — a data entry interface using continuous gestures and language models. In *Proceedings of the 13th Annual ACM Symposium on User interface Software and Technology (UIST '00)*. ACM, New York, NY, USA, 129-137.

WEN, J., WANG, X.-J., XU, W.-Z., AND JIANG, H.-X. 2008. Ambiguity Solution of Pinyin Segmentation in Continuous Pinyin-to-Character Conversion. In *Proceedings of the IEEE 2008 International Conference on Natural Language Processing and Knowledge Engineering* (*NLP-KE '08*).

WHITTAKER, E. AND RAJ, B. 2001. Quantization-based language model compression. In *Proceedings of Eurospeech*, 33-36.

WU, G.-Q. AND ZHENG, F. 2003. A method to build a super small but practically accurate language model for handheld devices. *J. Comput. Sci. Technol.* 18, 6 (November 2003), 747-755. DOI=10.1007/BF02945463

XIAO, J.-H., LIU, B.-Q., AND WANG, X.-L. 2007. Exploiting Pinyin Constraints in Pinyin-to-Character Conversion Task: a Class-Based Maximum Entropy Markov Model Approach. *Computational Linguistics and Chinese Language Processing*, 12, 3, 325-348.

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

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

YANG, K.-C., HO, T.-H., CHIEN, L.-F., AND LEE, L.-S. 1998. Statistics-based segment pattern lexicon: A new direction for Chinese language modeling. In *Proceedings of the IEEE 1998 International Conference on Acoustic, Speech, Signal Processing*, 169-172.

ZHANG, K. AND SUN, M. 2011. A comparison study of candidate generation for Chinese word segmentation. In Proceedings of the 7th IEEE International Conference on Natural Language Processing and Knowledge Engineering. 60-67.

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

ZHAO, H., HUANG, C.-N., LI, M., AND LU, B.-L. 2010. A Unified Character-Based Tagging Framework for Chinese Word Segmentation. *ACM Trans. Asian Lang. Inform. Process.*, 9, 2, Article 5 (June 2010), 32 pages. DOI=10.1145/1781134.1781135

ZHAO, H. AND KIT, C.-Y. 2011. Integrating unsupervised and supervised word segmentation: The role of goodness measures. *Inf. Sci.* 181, 1 (January 2011), 163-183. DOI=10.1016/j.ins.2010.09.008 http://dx.doi.org/10.1016/j.ins.2010.09.008

ZHENG, F. 1999. A Syllable-Synchronous Network Search Algorithm for Word Decoding in Chinese Speech Recognition. In *Proceedings of the Acoustics, Speech, and Signal Processing (ICASSP '99), on 1999 IEEE International Conference - Volume 02*. IEEE Computer Society, Washington, DC, USA, 601-604.

Received August 2011; revised October 2011, January 2012; accepted February 2012