Unknown Word Detection and Segmentation of Chinese using Statistical and heuristic Knowledge

Jian-Yun Nie
Département d'informatique et Recherche oppérationnelle
University de Montreal
Quebec, H3C 3J7 Canada
nie@iro.umontreal.ca

Marie-Louise Hannan
Centre for Information Technologies Innovation
1575 Bd. Chomedey, Laval, Quebec, H7V 2X2 Canada
hannan@citi.doc.ca

Wanying Jin
Computing Research Laboratory, New Mexico State University
Las Cruces, NM 88003, U.S.A.
wanying@nmsu.edu

Received on 1 December 1993 and accepted on 2 October 1995

Abstract

A sentence in Chinese is written as a string of characters without separation between words. Before linguistic analyses on a text, it has to be first segmented into a sequence of words. This work is not trivial as a string of characters often may be segmented to several sequences of legitimate words. Two different approaches have been used in Chinese segmentation: rule and dictionary-based approach and statistical approach. Each approach has its inconvenience while offering its advantages. The hybrid approach described in this paper makes use of both approaches in a single process. A text is first segmented using the rule and dictionary-based method. Then a hybrid approach is applied to locate and propose candidates for the unknown words contained therein. This latter step is based on both heuristic knowledge about morphological properties of the language and statistical information about the relative frequencies of different candidate words in the text. The use of heuristic rules in the second component enables us to greatly reduce the complexity of the problem; while statistical methods used result in both an increase in the system's flexibility and a diluting of the undesirable effects of incompleteness in the lexical database. This approach has proven both reliable and efficient through our experiments.

Keywords: *segmentation of Chinese, unknown word detection, hybrid method, dictionary, statistics, rules*

1. Introduction

A Chinese sentence is written as an unseparated string of characters. In order to do further linguistic analysis, it should be transformed into a chain of words. This is the goal of segmentation.

The segmentation is not a trivial process of finding out legitimate words from the input string due to the ambiguities that may exist. A sentence may often correspond to several sequences of legitimate words. So a crucial problem in Chinese segmentation is to choose the correct solution among all the possible ones. Another problem concerns unknown words. Words in Chinese cannot be all defined in a dictionary as may be done in European languages. Authors are offered with more flexibility of creating new words by combining characters or words. It is neither possible for a dictionary to contain all the words in Chinese, nor to specify all the rules for word formation. So a segmentation process should be able to recognize possible unknown words.

There are two main methods for segmentation of Chinese: the statistical approach and the rule-based approach. Statistical approaches [1, 5, 12, 15, 16] use statistical information such as word and character (co-)occurrence frequencies in the text. Rule-based approaches use a dictionary and heuristic rules about word formation [3, 7-9, 11, 18-22].

Statistical approaches may be easily adapted to any application area. Thus they may be seen as application-independent. They also do not require a dictionary per se. However, these advantages are obtained at the price of a proper training of the model, i.e. the estimation of the probabilistic parameters so that to make the model correspond to the application domain. The training often requires a great amount of training data which should be prepared manually. The manual preparation often requires as much effort as preparing a dictionary. Moreover, the model is not completely stable in the sense that probabilities need to be revised constantly.

Aside from the practical problem noted above, a more serious problem concerns the model itself. Most statistical approaches are limited to first-order Markov models. It has been documented [9] that such first-order models can hardly handle words containing more than two characters. Chinese words, however, may contain several characters. If the first-order Markov model is to be extended to a higher order, however, two other problems may be introduced: 1). The prevalence of many "functional" characters with a particular grammatical function such as prepositions, interrogative/negative markers and conjunctions can cause the statistical data in terms of frequency of occurrence to be unfairly skewed when the model is extended to anything beyond a first-order scheme. 2). Collecting enough data to uniformly extend the model beyond the first-order level is difficult.

Several methods have been proposed to address these problems. To increase the accuracy of word identification, [4] proposes to incorporate an adaptive learning approach into the statistical algorithm; whereas [16] proposes a tagger-based first-order Markov model; still another [15] considers each adjacent pair of characters having a high frequency to be a word and incorporates this principle into a first-order Markov model. The fact remains that these approaches cause an increase in the model's complexity.

On the other hand, rule and dictionary-based approaches have the advantage of being simple and efficient: The heuristic knowledge built into the system corresponds closely to knowledge about linguistic phenomena occurring in Chinese words and this knowledge is represented in a straightforward way, allowing human experts to verify its correctness. It is shown that a simple rule-based approach may achieve a performance comparable to that of a sophisticated statistical approach. However, a prerequisite for high-quality results in rule-based segmentation is a dictionary which is *complete*. It is unrealistic to suppose that a truly complete Chinese dictionary will be available because of the enormous *size* such a potential dictionary would imply, its *domain dependency* (certain strings may be words in some domains while not in others), and the fact that new words are constantly being produced (the *creative* aspect of language). Thus unknown words will continue to be one of the main obstacles to performance improvement in rule-based approaches.

To address the unknown word problem, some previous work has suggested the use of statistical approaches [1, 13]. However, in these approaches, either the unknown word problem has been restricted to a particular type (e.g. proper names) or additional information (e.g. part-of-speech) about characters is required, even if the characters are part of an unknown word. In practice, such information is not always available, nor is it consistently reliable.

In the method presented here, we will enhance the role of heuristic knowledge about word formation patterns in order to detect unknown words, while at the same time taking statistical information into consideration.

In this approach, segmentation is performed in two stages: First, a rule-based method is applied to the input text in order to divide the text into as many recognized segments (words) as possible, resulting in a partially segmented text. The remaining unsegmented strings of the text are then submitted to a procedure to detect unknown words, the result of which is a list of potential word strings. Those strings whose frequency is very high are added to the dictionary, and the segmentation process is driven again.

In contrast to a pure rule-based approach, this hybrid approach is quite flexible. When compared to a statistical approach, our approach proves less complex in that it allows a crucial simplification to take place where strong heuristic knowledge is available.

2. Overview of the segmentation process

The complete segmentation process may be seen as shown in figure 1. The first of two steps is the segmentation of input texts using a maximum-matching approach based on dictionary and rules [10]. An unknown word detection phase is then applied to segmented texts to detect possible unknown words in them. The latter process proposes a list of character-groupings which are very likely to be words.

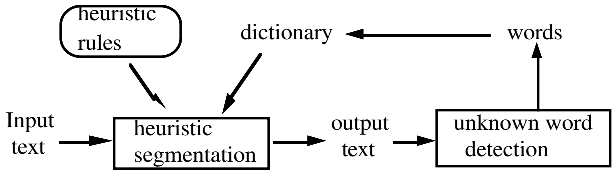


Figure 1. The segmentation process

Rule-based segmentation

As indicated in [9, 11], the maximum-matching (longest match) algorithm, which is also found in our system, allows the optimization of performance. The principle on which the algorithm works is the following:

The input character string is compared with the contents of the dictionary so that all sequences of characters constituting recognized lexical items can be highlighted. Words are linked from beginning to end of the input string, with several candidate word chains being proposed. Among all possible word chains, the one with the fewest and thus the longest words is considered to be the best segmentation.

Apart from the dictionary, a set of heuristic rules is also incorporated into our segmentation process in order to identify and segment words which follow some rules (for example, numbers and dates). In fact, if the dictionary look-up method corresponds to a declarative identification of words, then heuristic rules may be considered as a procedural word identification process. The advantage of incorporating the two identification methods into a single system is quite clear: some words have a fixed morphology and composition (e.g. common words), whereas others are subject to variation. While words of the first type may appropriately be stored in a dictionary, words of the second type may only be defined by rules.

The incorporation of two word identification strategies has been considered in some previous work on segmentation [11, 20]. In our study, we consider strings containing determiners, ordinal-number markers, cardinal numbers and classifiers. For example: 每一周 (*every week*), 这一回 (*this time*).

We first define the following categories of characters:

- determiners: 这 (*this*), 那 (*that*) 此 (*this*) 该 (*this*)
  其 (*its, his, her*) 每 (*each*) 各 (*every*)
  某 (*some*) 首 (*first*) 哪 (*which*) ...

- ordinal-number markers: 第 (*number*)

- cardinal numbers: 零 (*zero*) 一 (*one*) 壹 (*one*) 二 (*two*)
  贰(*two*) 十 (*ten*) 百(*hundred*) 半(*half*) ...

- classifiers: 班(*class*) 帮 (*band*) 包(*bag*) 杯(*cup*)
  辈(*generation*) 本 (*book*) 遍(*time*) 间(*room*)
  层 (*layer*) 年 (year) 月(month) 日(day)...

The rules concerning the formation of complex nominal pre-determiners (pre-det) from these characters are the following (where [...] indicates optional status and [...]* an optional arbitrary repetition):

ordinal cardinal [classifier] → pre-det
examples: 第一周 (*first week*) 第二 (*second*)

determiner [cardinal]* classifier → pre-det
examples: 这一回 (*this time*) 每层 (*every layer*)

cardinal [classifier] → pre-det
examples: 二十一(*twenty one*) 一百本 (*hundred books*)
一九九一年 (*in 1991*)

The combined use of dictionary look-up and the above heuristic rules enable the system to identify most words in the text.

Unknown word detection

In other rule-based segmentation approaches, once a text is segmented using the available dictionary or heuristic rules, the segmentation process is finished. We add an automatic unknown word detection component to the back-end of the whole segmentation process. Unknown word detection is based on both heuristic knowledge about word formation and statistical information on the occurrence rates of various character strings. A string will be attributed a high value showing its likelihood of being a word only if it both conforms to the heuristic word formation rules and it appears in texts frequently.

The advantage of including an unknown word detection phase in a rule-based approach is twofold: First, it allows a more flexible segmentation process that does not rely as heavily on the completeness of the dictionary as do previous rule-based approaches; Second, the architecture allows *local* words to be dealt with effectively. Local words are those words which are meaningful only in some special context. A typical example is that of proper names. The architecture proposed here allows only common words to be stored in the dictionary, while local words are to be detected later during the automatic unknown word detection process.

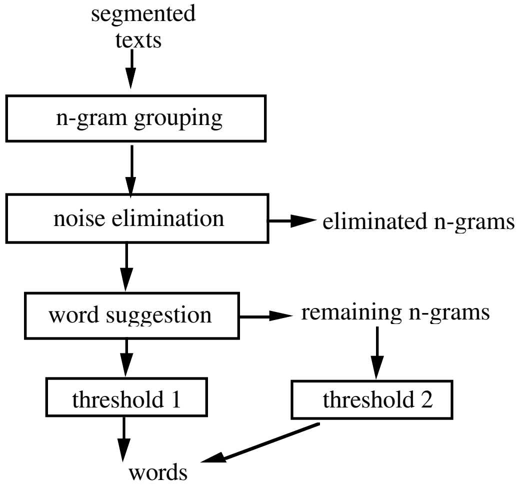


Figure 2. Detail of unknown word detection component

Unknown word detection proceeds as shown in Figure 2. Texts which have undergone the initial segmentation stage are submitted to a character grouping of n-grams (a string of n adjacent characters). These n-grams are then analyzed using heuristic morphological rules in order to filter out all n-grams that are not likely to be words, and to suggest those which are most likely to be words. Finally two different thresholds are used to filter the suggested and remaining n-grams respectively according to their frequency.

In the next section, we describe each operation in detail.

3. Unknown word detection

3.1. N-gram grouping

An n-gram is a string of n adjacent characters. It is extracted from the partially segmented texts, and is formed from isolated characters in unsegmented strings and shorter words. In this study, we only consider bi-, tri- and quadri-grams (also referred to as 2-, 3- and 4-grams).

348	的人	(-)
275	的是	(-)
241	上的	(-)
225	导报	(herald)
214	中的	(-, reach the target)
199	到了	(have arrived)
171	人的	(-)
136	成了	(have become)
128	来的	(-
126	钦本	(part of a name)
125	本立	(part of a name)
124	军涛	(part of a name)
122	请寄	(please send)
110	王军	(part of a name)
109	注明	(specify)
107	乔夫	(part of a name)
106	巴乔	(part of a name)
105	我在	(-)
105	出了	(have published)
103	新的	(new)
101	他在	(-)
99	戴晴	(name)
97	我是	(-)
96	戈巴	(part of a name)
94	的事	(-)
93	不到	(less than)
88	大的	(big)
86	他说	(he says)
85	江青	(name)
82	有了	(have had)
82	也不	(neither)
81	于观	(name)
81	来了	(have arrived)
79	叶立	(part of a name)
79	也有	(also have)
79	立钦	(part of a name)
79	李鹏	(name)
78	这一	(this one)
77	我说	(I say)
77	里的	(-)
75	是在	(-)
74	下的	(-)
74	上了	(have gone up)
71	之春	(spring of)
71	小平	(part of a name)
71	王蒙	(name)
71	民运	(democratic movement)
71	陈子	(part of a name)

Sample 1. Freely obtained bi-grams

To create n-grams from isolated characters, the only condition is their contiguous appearance in the text. To form n-grams involving shorter words, a further condition applies: there must be at least one isolated character in the n-gram. This is to avoid including words having an independent status in longer words, in particular, for 4-grams. For example, 信息科学 (*information science*) should not be considered as a new quadri-gram because 信息(*information*) and 科学(*science*) are words.

The algorithm is quite straightforward. The 2-gram samples extracted from the 50 first issues of Hua Xia Wen Zhai (华夏文摘), a weekly published electronic journal, are shown in sample 1, where the number denotes frequency, and the translation is between parentheses ('-' means no translation is possible).

Our first observation is that the number of freely grouped bi-grams is very high, and that the majority are not words. As we may also observe from this sample, the most frequent bi-grams are those containing a character which belongs to some special category such as preposition or adverb (的(of), 了(already)). These bi-grams are not words and represent "noise". So the focus of the next step in our approach is to eliminate these non-word n-grams using heuristic information about word formation.

3.2. Noise Elimination

A statistical study of Chinese words and characters is reported in [17]. The data used in this study were taken from Chinese-language textbooks used in the People's Republic of China. One result presented is that Chinese characters vary in their ability to combine with other character(s) to form a word according to the number of words they actually appear in. The more a character combines with other characters to form words, the higher its word formation power is. Our hypothesis is that a character with low word formation power is relatively unlikely to form a word in a text. These characters are called function characters. Thus a simple way to eliminate noise among n-grams is to eliminate those containing function characters.

However, the status of function characters is not solely determined by the number of words formed, but also the frequency of individual characters in the texts should be taken into account. This is a practical consideration for the sake of efficiency. The more frequently a character appears, the more its elimination will effectively reduce noise (if it is a function character). Thus a operation may be defined to determine the word formation power (*WFP*) of a character based on both the number of words in which it appears and its frequency as follows:

$$WFP(character) = \frac{\textit{number of words including the character}}{\textit{frequency of the character in texts}}$$

The following table gives the *WFP* calculated for some most frequent characters.

Character	NB. of words	Occurrence frequency	*WFP*	*WFP* Rank
的	3	20576	0.00014580	1
我	2	6482	0.00030855	2
了	3	9421	0.00031844	3
他	2	4483	0.00044613	4
和	1	2096	0.00047710	5
又	1	1507	0.00066357	6
说	2	2763	0.00072385	7
着	3	3890	0.00077121	8
也	2	2191	0.00091283	9
在	6	5829	0.00102934	10
就	3	2776	0.00108069	11
看	2	1850	0.00108108	12
你	2	1757	0.00113830	13
不	6	5145	0.00116618	14
都	2	1715	0.00116618	15
把	2	1685	0.00118694	16
还	2	1594	0.00125471	17
个	5	3900	0.00128205	18
是	9	6675	0.00134831	19
这	7	4383	0.00159708	20
一	16	9863	0.00162222	21
有	8	4505	0.00177580	22
小	3	1649	0.00181928	23
们	9	4754	0.00189314	24
到	6	3128	0.00191816	25

Characters with *WFP* lower than some threshold are considered to be function characters. The lower the threshold, the higher the efficiency will be, but the risk of eliminating words will also increase.

In analyzing the results of [17], we observe that characters with low *WFP* often belong to the following categories:

- prepositions: 的(of) 之(of)
在(at)
把 使 将 让 (make, see [Chen 91])

- pronouns: 我(I, me) 你(you) 他(he, him) 它(it) ...

- interrogatives: 谁(who) 哪(what) 怎(how) 啥(what)
咋(how) 岂(how) 呢(interrogative mark)...

- adverbs: 也(also) 已(already) 了(already) 亦(also)
又(again) 再(again) 曾(ever) 竟(even) ...

- conjunctions: 与(and) 和(and) 或(or) 因(because) 而(but)
但(but) 虽(although), 当(when) ...

- determiners: 这(this) 那(that) 此(this) 每(each)
各(every) 某(some) ...

- classifiers: 个 点 些 ...

Thus another way to determine which characters are function characters is by their syntactic behavior. Our list of function characters is determined by considering both the *WFP* and the character's behavior in different syntactic contexts.

Once function characters are identified, a filtering program eliminates all n-grams which involve them. After this elimination is carried out, we observe a sizable decrease in noise among the remaining n-grams. Below is the same bi-gram sample after function character elimination.

225 导报(herald)
126 钦本(part of name)
125 本立 (part of name)
124 军涛 (part of name)
122 请寄(please send)
110 王军(part of name)
109 注明 (specify)
107 乔夫(part o f name)
106 巴乔(part of name)
99 戴晴 (name)

96 戈巴 (part of name)
85 江青 (name)
81 于观 (name)
79 叶立 (part of name)
79 立钦 (part of name)
79 李鹏 (name)
71 小平 (part of name)
71 王蒙 (name)
71 民运 (democratic movement)
71 陈子 (part of name)

Sample 2. Remaining bi-grams after function character elimination

We should note that this elimination is not always "safe", in that correct words may sometimes be eliminated along with the function characters. However, as the word formation power of function characters is very low, the chance of incorrect elimination is also low. This has been confirmed by our experimentation (cf. section 4). Moreover, the words that may be formed using function characters are often commonly used words. They are either already present in the existing dictionary, or we may create an exhaustive list to include them. So the danger, although present, can be considered to be negligible in most cases.

3.4. Candidate word suggestion

When segmentation is applied to journalistic texts (which is the case for our experiments), one of the things that becomes evident in the use of n-grams, especially from tri-grams, is the high proportion of proper names. We also observe that often, a very common word is formed from a shorter word by adding a prefix or suffix. All these n-grams should be considered as words. Thus special attention is paid to these problems in order to suggest strong word candidates. Although these problems may also occur in bi- or quadri-grams, we will deal only with tri-grams.

Proper Names

Proper name suggestion is proceeded independently from the function character elimination. To deal with proper names, the following criteria are used:

1. For a 3-gram to be a name, the first character should be a family name. To test this condition, a set of the most usual family names is provided. For example, 张(Zhang) 李(Li) 王(Wang).

2. Some characters are almost never used in given names (for example, 哪(which) 狠(nasty) 恶(bad) 坏(bad)). 3-grams with these characters have little chance of being a name, and thus are eliminated.

A similar approach has been employed in other systems (e.g. [2]). Here are some of the proper names suggested by this process:

125	钦本立	39	刘晓波	33	朱若鹏
110	王军涛	39	焦志敏	32	周恩来
79	叶立钦	38	赵紫阳	32	张学良
69	陈子明	38	姚明辉	32	胡耀邦
58	邓小平	37	姚文元	30	江泽民
52	刘增哲	36	赵尧舜		
45	费先生 (Mr. Fei)	34	唐德明		

Sample 3. Suggested proper names

Except 费先生 (Mr. Fei) in this sample, all the strings are proper names. Nevertheless, 费先生 is also a word.

Affixes

For a word to be considered as having an internal affix structure, both of the following conditions should be respected:

1. The first (last) character should be a possible prefix (suffix). For example:

   prefix: 大(big) 小(small) 总(general) 副(vice), ...

   suffix: 人(person) 们(plural mark) 局(bureau)
   会(association) 化(-ize/-ization), ...

2. The remaining characters in the 3-gram should form a known word.

Below is a sample of words proposed by the system which have affix structure. The affixes are underlined. In this sample, all the strings, except 29 许多人 (which is a word), correspond to an affix structure.

41	领导人(leader)	29	基金会(fund association)
41	劳工们 (workers)	29	大陆人 (mainlander)
37	北京人 (Pekinese)	27	许多人 (many people)
36	负责人 (responsible person)	26	自由化 (liberalization)
35	政治局 (politburo)	22	新闻界 (press milieu)
32	朋友们 (friends)	21	劳改队 (prison camp)
29	小幽默 (small joke)	20	孩子们 (children)

Sample 4. Suggested prefix/suffix structure

3.3. N-gram overlapping

As the statistics for n-grams are calculated independently for each n, a single appearance of a given string in a text may be counted as part of several n-grams with different values for n. For example: the appearance of the string 王军涛(a proper name) is also counted for 王军 and 军涛 which are not words or proper names. We call this phenomenon the overlapping between n-grams.

To preclude the consideration of n-grams contained within longer n-grams as separate entities - the latter have a high probability of being words (with a high frequency) - a procedure for eliminating n-gram overlapping proceeds as follows:

Suppose the n-gram X is included in a longer n-gram Y, then

$$freq(X) = freq(X) - freq(Y).$$

That is, for each n-gram which is contained within another (longer) n-gram which itself has a high frequency, remove as many occurrences as that of the longer n-gram.

This method has proven to be an efficient way of reorganizing n-grams according to their real frequency. This process is also efficient for dealing with foreign names of cities or people such as 戈巴乔夫 (Gorbachev). As can be seen in the samples below, 戈巴, 巴乔 and 乔夫 are eliminated from the earlier bi-gram list (Sample 2) after consideration of overlapping with 3- and 4-grams. The remaining n-grams are of high likelihood to be words.

152	导报 (herald)	79	叶立钦 (Yeltsin)
125	钦本立 (name)	79	李鹏 (name)
122	请寄 (please send)	71	王蒙 (name)
110	王军涛 (name)	69	陈子明 (name)
99	戴晴 (name)	64	找到 (find out)
95	戈巴乔夫 (Gorbachev)	62	想到 (think about)
94	注明 (specify)	61	菊豆 (surname)
81	于观 (name)	58	邓小平 (name)
80	江青 (name)	57	自由导报(Liberalization herald)

Sample 5 Bi-grams after elimination of overlapping

After the n-gram elimination and suggestion, new words should be selected from both the lists of n-grams remained after elimination and the lists of suggested n-grams. The selection may be done according to the frequency of n-grams. An n-grams having a frequency higher than a certain threshold is considered as a new word. The determination of the thresholds, however is strongly dependent on the corpus. The experiments that we will describe in the next section will give some indications on this operation.

4. Experimentation of unknown word detection

The approach described here has been tested on a corpus established from the weekly Chinese electronic news journal - Hua Xia Wen Zhai (华夏文摘). The first 50 issues of the journal are included in the test corpus, which total 1.58 megabytes, or about 790 000 Chinese characters. We used a (limited) dictionary containing about 50 000 entries. The first segmentation found 178 564 compound words and left 74 279 isolated characters. Figure 3 shows the steps of our experiments: separated characters are first grouped up into bi-grams, tri-grams and quadri-grams separately. Noise elimination is applied to all these n-grams. Candidate-word suggestion is only applied to tri-grams. Finally, n-gram overlapping is applied to tri-grams using frequent quadri-grams, to bi-grams using frequent quadri-grams and tri-grams.

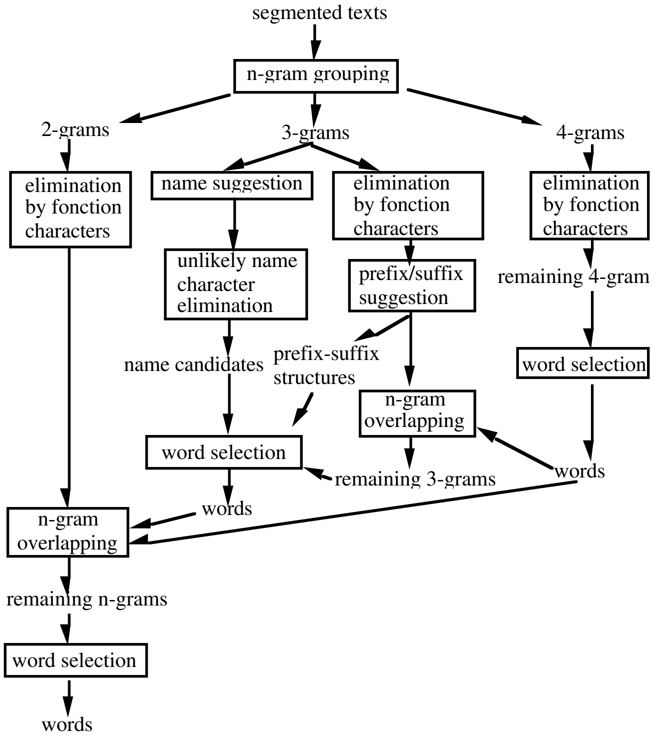


Figure 3. Flow of operations

Table 1 gives the number of distinct n-grams grouped up from the segmented texts. In table 2, we give the number of n-grams that remain after each step of elimination or suggestion.

	2-grams	3-grams	4-grams
NB of n-grams	44 620	118 146	101 833
max. frequency	348	269	95

Table 1. Freely grouped n-grams

	2-grams	3-grams	4-grams
initial nb.	44620	118146	101833
eliminated by funct. ch.	18247 (40.9%)	69346 (58.70%)	54237 (53.26%)
suggested as proper name		3059 (2.59%)	
suggested as affix struct.		2475 (2.09%)	
remaining n-grams	26373 (59.1%)	43266 (36.62%)	47596 (46.74%)

Table 2. Distribution of n-grams after each operation

We may observe that the function character elimination step is very efficient in that it eliminates roughly half of the n-grams. The question raised now concerns the quality of these operations. The remainder of the section gives an evaluation of the operation's quality.

4.1. On function character elimination

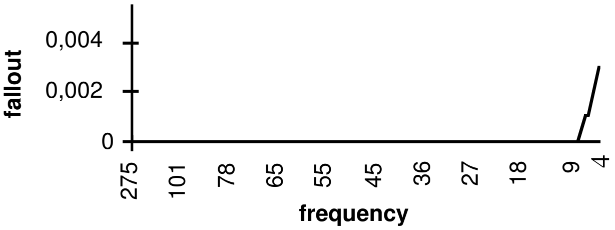
We examined the eliminated n-grams with frequency ≥4 in the texts, and this threshold seemed to cause few actual words to be eliminated. Among all the 2489 bi-grams with a frequency ≥4 eliminated, only 8 are words (the number is their frequency):

7 吉它(guitar)	4 以防(prevent)
5 让路(allow to pass)	4 雪地(snow field)
5 就坐(sit down)	4 鼓点(drum roll)
4 自此(from then on)	4 被窝(blanket)

To evaluate the risk involved in the elimination process, we make use of the function *fallout* defined as follows [14]:

$$Fallout = \frac{\textit{number of eliminated words}}{\textit{number of eliminated n-grams}}$$

The lower the fallout, the safer the process is. The following graph illustrates the fallout ratio for bi-gram elimination for different minimum frequency values. For example, if n-grams whose frequency is ≥ 4 (the rightmost point of the curve) are eliminated, we obtain fallout = 0,32%.



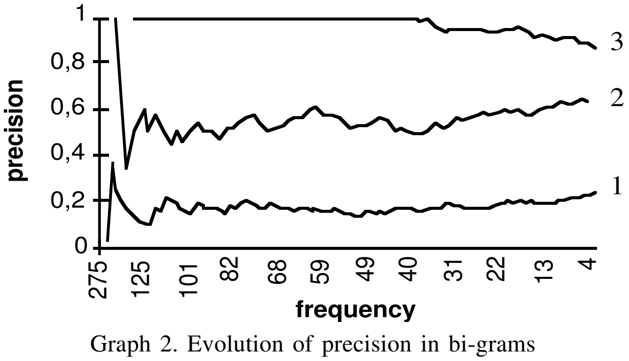
Graph 1. Fallout ratio for function character elimination on bi-grams

It may be observed that few real words are eliminated during the function character elimination process, which indicates that this process is "safe" enough to be reliable even in the absence of manual verification.

We also performed a manual examination of all n-grams proposed or remaining with a frequency ≥ 4. We obtained an exhaustive list of 2431 unknown words appearing at least 4 times in the corpus. This list is then used to give a partial evaluation of *precision* of proposed or remaining n-grams. Precision is defined as follows [14]:

$$precision = \frac{\textit{number of real words found}}{\textit{number of n-grams}}$$

Graph 2 gives the precision ratio for remaining bi-grams at different frequency levels. The precision value at a particular frequency point (f) is the precision obtained if all n-grams with frequency ≥ f are considered. We may see the evolution of precision as the operations proceed.



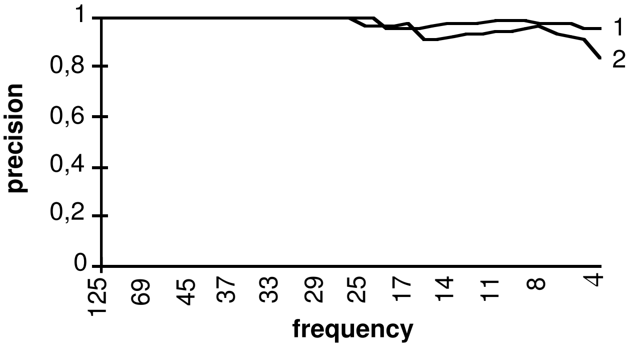
Graph 2. Evolution of precision in bi-grams

(1: freely grouped bi-grams; 2: after function character elimination; 3: after overlapping elimination)

A sizable increase in precision is obtained with function character elimination and overlapping n-gram elimination. After the elimination of overlapping segments, bi-grams of frequency ≥ 4 have a precision of more than 80%; whereas those of frequency ≥ 40 have a precision near 100%. This proves that our n-gram elimination process is very efficient. If a remaining n-gram has a high frequency, it is almost sure to be a word. In this sense, high-ranking n-grams may be added to the dictionary without human verification. For our corpus, according to the accuracy expected, a frequency threshold for word selection set somewhere above 4 seems quite reasonable.

4.2. On the suggestions

On the n-gram suggestions of proper names and affix structures, the precision evaluations obtained are illustrated in graph 3. They show that the n-gram suggestion mechanism implemented is highly accurate. In most cases, the n-grams suggested are real words.



Graph 3. Precision of suggested 3-grams

(1: suggested as affix structure; 2: suggested as proper name)

5. Using detected unknown words in segmentation

Once the n-grams that are possible words are detected, they have to be integrated to the segmentation process. There are several ways to do this. Here we examine the following two ways:

1. add the detected frequent n-grams to the dictionary;

2. determine a probability for detected n-grams according to their frequency of occurrences, while at the same time, a default probability is attributed to all the entries of the original dictionary.

Two texts randomly selected from the corpus are used as test issues. The following table gives their sizes and the success ratios using the longest-matching algorithm with the original dictionary:

text	nb. of words	success ratio (errors)
test issue 1	7741	82.75% (1335)
test issue 2	6730	93.91% (410)

Table 3. Performance of the initial segmentation

We can notice here that the success ratio for the first issue is very low. This is mainly due to the fact that there is a huge amount of proper names in this issue. There are also a lot of proper names in the second, but less than in the first issue. Another reason for the poor performance is the incompleteness of the original dictionary (slightly more than 50 000 items). Many usual words are not included in it.

5.1. Adding the unknown words in the dictionary

We added the detected n-grams successively according to their frequency of occurrences in the corpus. In this method, when an n-gram is added, it is considered in the same way as an item in the original dictionary. Table 4 gives the results of the segmentation for this method.

We can observe that the choice of the frequency threshold for the added n-grams is important. With a too high threshold, few unknown words are added; but with a too low threshold, a lot of noise strings are considered as words. In our case, the best performance for both test issues is obtained when the threshold is set at 4.

Freq.(nb) of added n-grams	succ. ratio (errors) issue 1	succ. ratio (errors) issue 2
≥7 (756)	93.32 (517)	94.73 (355)
≥6 (996)	93.32 (517)	94.84 (347)
≥5 (1333)	93.46 (506)	95.08 (331)
≥4 (1980)	93.81 (479)	95.66 (292)
≥3 (3284)	93.46 (506)	94.92 (342)
≥2 (6934)	92.33 (594)	94.00 (404)

Table 4. Segmentation performance with the simple addition of the detected n-grams to the dictionary

5.2. Unknown words combined with their statistics

One may think that, as the detected n-grams are only *possible* words, they should not be considered in the same way as the entries in the original dictionary which are words. The intuitive way may be to attribute a probability value to the detected unknown words according to their frequency of occurrence, while attribute a higher probability to the entries of the dictionary. However, as we only have the frequency of occurrence of the detected possible words, but not their frequency as words, we cannot calculate the probability of the detected strings as follows:

$$prob(n\text{-}gram) = \frac{occ.\ of\ the\ n\text{-}gram\ as\ word}{total\ occ.\ of\ the\ n\text{-}gram}$$

It seems reasonable to approximate this probability as follows:

$$prob1(\text{n-gram}) = \frac{nb.\ of\ occurrences\ of\ n\text{-}gram}{max.\ nb.\ of\ occ.\ of\ the\ detected\ n\text{-}grams}$$

$$prob2(n\text{-}gram) = \frac{nb.\ of\ occurrences\ of\ n\text{-}gram}{\Sigma\ nb.\ of\ occ.\ of\ the\ component\ characters}$$

default prob. for dict. items	succ. ratio with prob 1		succ. ratio with prob 2	
	issue 1	issue 2	issue 1	issue 2
0.01	91.55 (654)	93.31 (450)	91.46 (661)	93.37 (446)
0.05	93.99 (465)	95.65 (293)	93.97 (467)	95.73 (287)
0.075	95.12 (378)	96.39 (243)	95.19 (372)	96.58 (230)
0.1	95.56 (344)	96.69 (223)	95.49 (349)	96.82 (214)
0.2	94.87 (397)	95.60 (296)	94.87 (397)	95.61 (295)
0.3	93.88 (474)	93.98 (405)	93.76 (483)	94.14 (394)
0.4	92.16 (607)	92.97 (473)	92.08 (613)	93.13 (462)
0.5	91.98 (621)	92.78 (486)	91.87 (629)	92.87 (480)

Table 5. Segmentation with the probabilistic integration

The probability attributed to the dictionary items is called the default probability. By varying the default probability, we can vary the importance of the statistical information and the dictionary in the segmentation process. The higher the default probability for the items in the dictionary, the lower the importance of the detected n-grams. With the above assignment of probability, the segmentation process can operate in exactly the same way as statistical approaches. That is, if a string is segmented into a word chain, the probability of the chain is calculated as the multiplication of the probabilities of all the constituent words. Table 5 shows the performance of the segmentation with different default probability attributed to the items of the dictionary, and *prob*1 and *prob*2 as probability for the detected n-grams.

We can observe that with either a too high or too low default probability, we do not obtain the best performance. The reason is similar to that given for a too high or too low frequency threshold in the previous section. In our case, the best performance is obtained when it is set to 0.1.

As to *prob*1 and *prob*2, we do not observe a significant difference. The two measures give similar results. In comparison with the simple addition of the detected n-grams, we observe a slight improvement in this second method: the best performance is around 95.5% vs. 93.81% for issue 1 and around 96.7% vs. 95.66 for issue 2. It may be concluded that incorporating the detected n-grams together with their statistics is a better method than the simple addition to the original dictionary.

Finally, we notice that the best performance, around 96%, is much lower than those reported in some other studies (of as high as 99.9%). This may be explained by 1) our experiments are driven using raw texts without any filtering; 2) our texts are mainly news reports and contain a great number of proper names. The major part of proper names appear only a few times. Our unknown word detection is unable to detect them.

However, with our unknown word detection, a significant improvement is observed (comparing to the original performance shown in table 3). This confirms the value of this process for the segmentation of Chinese texts.

6. Further improvement

Local statistical information

In our experiments, statistics is made on the entire corpus. Such global statistical information is unable to reflect local words. For example, using global statistical information, the following sentence will fail to be segmented correctly:

当天青和菊豆回家时
(when Tian-Qing and Ju-Dou go home)
1.(incorrect)
当天 青 和 菊豆 回家 时
(the same day, when Qing and Ju-Dou go home)
2.(correct)
当 天青 和 菊豆 回家 时
(when Tian-Qing and Ju-Dou go home)

This occurs because, according to the global statistical information, 当天 (*the same day*) is much more usual than 天青 which is a proper name in this text. It is only when local statistical information (i.e. that of 天青) is enhanced that the correct segmentation can be selected.

Thus, it is desirable to incorporate local statistical information As well when segmenting a particular text.

Segmentation should not be limited to the lexical level

It should be understood that segmentation based on statistical or heuristic lexical information will still leave many ambiguities. Syntactic, semantic or even pragmatic and contextual information will be necessary is some cases. Here we give some examples found in our corpus in order to illustrate the need of broader information in segmentation of Chinese.

syntactic information:

这包括 (*this includes*) may be segmented as
这(*this*) 包括(*include*)
or 这包(*this bag of*) 括(*quote*)

In this case, syntactic information is necessary and sufficient to choose the correct (first) segmentation because in the second, the word 括 (*quote*) can only be verb, which is syntactically incompatible with 这包(*this bag of*).

semantic information:

每周五出版(*published every Friday*) may be segmented as
每周(*every week*) 五出(*five plays of*) 版(*edition*)
or 每(*every*) 周五(*Friday*) 出版(*publish*)
(= published every Friday)

The first (incorrect) segmentation may be rejected only when the semantic incompatibility between the classifier 出(*play*) and the noun 版(*edition*) is brought to light.

pragmatic information:

Several cases are strongly linked with the problem of text style or sentence rhythm. For example:

他是一名诗人 may be segmented as
他(*he*) 是(*is*) 一名(*a*) 诗人(*poet*)
or 他(*he*) 是(*is*) 一(*a*)名诗人(*famous poet*);

这一场景 (*this scene*)
as 这一(*this one*) 场景(*scene*)
or 这一场(*this scene of*) 景(*scene*).

Although all the segmentations are syntactically and semantically acceptable, in each case, the second choice does not have a proper rhythm. Thus the first segmentations are preferred.

A related problem is text style. Different kinds of texts use different sentence styles. According to the style of the text, 着急得手汗直冒 may be segmented as

着急(*worry*) 得(*so that*) 手汗(*sweat in hands*) 直冒(*pour out*)
(= to be so worried that the hands sweat)

or 着急(*worry about*) 得手(*success*) 汗(*sweat*) 直冒(*pour out*)
(= to be so worried to succeed as to sweat)

In general cases, the first segmentation is more natural, but in some types of novels, sentences as segmented in the second way are often used.

However, it remains difficult at present to obtain and formulate all of the above information, in particular for text style or sentence rhythm, because there has been little formal study in these areas until now. Therefore, this observation may be considered as an appeal to broaden the parameters on the Chinese segmentation problem.

References

1. J.-S. Chang and e. al., Chinese word segmentation through constraint satisfaction and statistical optimization. *ROCLING-IV*, Taiwan, 147-165 (1991).
2. J.-S. Chang and e. al., A multiple-corpus approach to identification of Chinese Surname-names. *Natural Language Processing Pacific Rim Symposium*, Singapore, 87-91 (1991).
3. K.-J. Chen and S.-H. Kiu, Word identification for Mandarin Chinese sentences. *5th International Conference on Computational Linguistics*, 101-107 (1992).
4. T.-H. Chiang and e. al., Statistical models for segmentation and unknown word resolution. *5th R.O.C. Computational Linguistics Conference*, 123-146 (1992).
5. T. Dunning, Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics*, vol. 19, pp. 61-74 (1993).
6. J. Guo and H.-C. Liu, PH - A Chinese corpus for PINYIN--HANZI transcription. National University of Singapore, Institute of Systems Science, Singapore, ISS Technical Report (1992).
7. K.-K. He, H. Xu, and B. Sun, The Design Principle for a Written Chinese Automatic Segmentation Expert System. *Journal of Chinese Information Processing*, vol. 5, pp. 1-14 (1991).
8. W. Jin and J.-Y. Nie, Segmentation du Chinois - une Etape Cruciale vers la Traduction Automatique du Chinois. in *La Traductique*, P. Bouillon and A. Clas, Eds. Montreal: Les presses de l'Université de Montréal, pp. 349-363 (1993).
9. B.-I. Li and e. al., A maximal matching automatic Chinese word segmentation algorithm using corpus tagging for ambiguity resolution. *R.O.C. Computational Linguistics Conference*, Taiwan, 135-146 (1991).
10. N. Y. Liang, The Automatic Segmentation in Written Chinese and an Automatic Segmentation System - CDWS. *The Academic Journal of Beijing Institute of Aeronautics and Astronautics*, vol. 4 (1984).
11. N. Y. Liang and Y.-B. Zhen, A Chinese word segmentation model and a Chinese word segmentation system PC-CWSS. *COLIPS*, vol. 1, pp. 51-55 (1991).
12. M.-Y. Lin, T.-H. Chiang, and K.-Y. Su, A preliminary study on unknown word problem in Chinese word segmentation. *ROCLING V*, 147-176 (1992).
13. Z.-Y. Peng and J.-S. Chang, A study of various meaning in Chinese words -- segmentation and words feature marker. *R.O.C. Computational Linguistics Conference*, 173-193 (1991).
14. G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*: McGraw-Hill (1983).
15. R. Sproat and C. Shih, A statistical method for finding word boundaries in Chinese text. *Computer Processing of Chinese and Oriental Languages*, vol. 4, pp. 336-351 (1991).
16. M.-S. Sun and e. al., Some Issues on the statistical approach to Chinese Word Identification. *3rd International Conference on Chinese Information Processing*, 246-253 (1992).
17. H. Wang and e. al., *Statistics and Analysis of Chinese lexicon*. Beijing: Foreign languages' teaching and research pub. comp. (1984).
18. L.-J. Wang, T. Pei, W.-C. Li, and L.-C. Huang, A Parsing method for identifying words in Mandarin Chinese sentences. *12th International Joint Conference on Artificial Intelligence*, Sydney, Australia, 1018-1023 (1991).
19. H. Xu, K.-K. He, and B. Sun, The implementation of a written Chinese automatic segmentation expert system. *Journal of Chinese Information Processing*, vol. 5, pp. 38-47 (1991).
20. T.-S. Yao, G.-P. Zhang, and Y.-M. Wu, A rule-based Chinese automatic segmentation system. *Journal of Chinese Information Processing*, vol. 4, pp. 37-43 (1990).
21. C.-L. Yeh and e. al, Rule-based word identification for Mandarin Chinese sentences - A unification approach. *Computer processing of Chinese and Oriental Languages*, vol. 5 (1991).
22. Y.-X. Zhou and W.-T. Wu, A Practical Method of Segmentation of Chinese -- A Method Based upon Chain Table. *Journal of Chinese Information Processing*, vol. 4, pp. 34-41 (1989).