COMPUTER PROCESSING OF CHINESE & ORIENTAL LANGUAGES
Vol. 4, No. 4, March 1990

A STATISTICAL METHOD FOR FINDING WORD BOUNDARIES IN CHINESE TEXT

Richard Sproat and Chilin Shih

ABSTRACT

A significant problem in Chinese text analysis is detecting where word boundaries lie. We have employed a statistical method to group Chinese characters into two-character words making use of a measure of character association based on mutual information. The statistics were derived from a corpus of approximately 2.6 million characters of Chinese newspaper text. The method has been tested on randomly selected texts from a corpus of Chinese newspaper text with quite favorable results.

Keywords: Chinese text processing, character-to-word assignment, natural language processing, parsing, statistical models of language, mutual information.

1. INTRODUCTION

One problem which arises in Chinese text which is not generally very problematic in English text is the identification of word boundary locations. In English and other languages which use Roman or Greek-based orthographies, for instance, word boundaries are often reliably indicated by spacing.

In Chinese the situation is different. Chinese characters (字, also referred to as Hanzi 漢 字) by and large represent morphemes. Chinese *words* (詞), like words in English, may be composed of either one or many morphemes. Some monomorphemic words are:

花　樹　筆　好　人

Some polymorphemic words are:

世界　需要　便宜　辦公室

On the other hand, since Chinese orthographic tradition does not typically sanction the use of spaces as separators of words, there is no immediate way of deciding which characters in a text should be grouped into words. So in a sentence such as:

Manuscript received on April 3, 1989; revised November 1, 1989. R. Sproat is with AT & T Bell Laboratories, 600 Mountain Ave., Murray Hill, N.J. 07974, and C. Shih is with the Department of Modern Languages and Linguistics, Cornell University, Ithaca, N.Y. 14853.

'my (younger) brother wants to ride the train home now'

我 弟 弟 現 在 要 坐 火 車 回 家

There is no indication in the orthography that 弟 弟 , 現 在 , 火 車 or 回 家 are words. However, according to [1: p. 9], about 69% of Chinese text consists of one character words; of the remainder all but 1% consists of two-character words. This suggests that we can handle about 99% of low-level Chinese text analysis if we have an algorithm which can correctly 'parse out' two-character words. We shall be solely concerned with two-character words in this paper.

There are a number of applications where one needs to know where the word boundaries lie in a Chinese text. One obvious application is as a preprocessor for a Chinese parser (see, e.g. [2]). Once one has chunked the input into words, the parser can go to work on grouping the words into larger phrases. However, knowing which characters to group together as words is useful even if one doesn't have a syntactic parser for Chinese. So, for Chinese text-to-speech synthesis, local groupings of characters into words is crucial for correctly assigning prosody (i.e., tone and intonation) to a sentence [3]. For example, the application of Mandarin Third Tone Sandhi [4] is sensitive to groupings of characters since it applies in a cyclic fashion first to the smallest groupings and then to larger groupings. Consider for example:

小 老 鼠

xiao3 [lao3 shu3] → xiao3 [lao2 shu3]

(We use the Hanyu Pinyin transcription, with numbers after the syllable indicating the tone).

In this example tone 3 (三 聲) in 老 changes to tone 2 (二 聲) by Third Tone Sandhi. Since 老 鼠 is a word of Mandarin, those two characters are grouped together and Third Tone Sandhi applies first to that sequence of morphemes. Once 老 's tone 3 is changed to tone 2, 小 is no longer preceding a tone 3, so there will be no further change. Compare this situation with the following:

老 鼠 小

[lao3 shu3] xiao3 → [lao2 shu2] xiao3

Again, Third Tone Sandhi applies to the smallest grouping first, so 老 's tone 3 is changed to tone 2. When we then look at the larger context, we find that tone 3 of 鼠 is subject to the same rule because it precedes the tone 3 of 小 . So Third Tone Sandhi applies twice, once on the smallest group and subsequently on the larger sequence of characters. Wrong grouping will yield unacceptable results as indicated by a '*' in front of the example.

老 鼠 小

lao3 [shu3 xiao3] → *lao3 [shu2 xiao3]

A solution to the problem of grouping is therefore clearly important. Furthermore, a solution which can handle unrestricted text is desirable, especially for the text-to-speech application described above. Ideally, one would like to be able to make use of a large online dictionary of Chinese words and fixed phrases for the purpose of grouping characters; such a dictionary would not just give a list of these words but also give grammatical information such as part of speech and meaning. Indeed, segmentation of input into words has been achieved using a dictionary in the experimental system described in [2,5]; however, this system works with a dictionary of about 800 words, which is far too small for handling unrestricted text. Unfortunately, it remains true that large online dictionaries of Chinese — even ones which merely consist of lists of words with no grammatical information — are hard to come by (though see [1] for discussion of a computer-readable list of over 6000 Chinese words). We have therefore explored a statistical approach to solving this problem, a description of which we now turn to.

2. DESCRIPTION AND ANALYSIS OF THE METHOD

2.1 The statistical method

One way to approach the problem of grouping Chinese text correctly into two-character words is to make use of some statistical measure of association between consecutive characters in a sentence; groupings of characters which are more highly associated should be preferred over groupings of characters which are less highly associated.

The statistical method of character grouping employed is based on the concept of mutual information, defined as follows (see, e.g., [6: pp. 28++]), for two events a and b with probabilities $P(a)$ and $P(b)$ and joint probability $P(a\ ,\ b)$:

$$(1)\quad I(a;b) \;=\; \log_2\left(\frac{P(a,b)}{P(a)P(b)}\right)$$

Roughly speaking, mutual information gives a measure of how strongly associated two events are; if $P(a\ ,\ b)$ is significantly higher than $P(a)P(b)$ — the probability of the occurrence of a together with b under the assumption of independence — then there is good reason to believe that a and b somehow 'belong together'. Now, we can get a measure of association of two characters by using an approximation to the mutual information between the characters. In particular, we can define the *association*, A, between two characters a followed by b as follows:

$$(2)\quad A(ab) \;=def\; \log_2 \frac{\frac{f(ab)}{N}}{\frac{f(a)}{N}\,\frac{f(b)}{N}} \;=\; \log_2(N) + \log_2\left(\frac{f(ab)}{f(a)f(b)}\right)$$

Here, $f(a)$ is the frequency of occurrence of character a in the corpus, $f(ab)$ is the

frequency of occurrence of the sequence of characters *a* followed by *b*, and *N* is the size of the corpus. This equation is related to the equation for mutual information given in (1) under the assumption that the probability of occurrence of a character is well approximated by the ratio of the frequency of occurrence in the corpus over the size of the corpus; this assumption becomes increasingly reasonable as the size of the corpus grows. There is, of course, another difference between association as defined in (2) and mutual information: *A(ab)* and *f(ab)*, as noted are respectively the association and the frequency in the corpus of character *a* followed by *b*. Mutual information, on the other hand, is a measure of the information of the two events cooccurring. So the definition of association in (2) has an additional constraint of ordering imposed upon it. This constraint, however, is important for natural language analysis since we usually want to know how highly associated two words are in a particular order. For example, if *United* and *States* occur together frequently in that order in a text database, we might want to deduce that the two words should be grouped together in a particular text if we find them adjacent to one another in that order. However, we probably would not want to deduce from this that the words should be grouped together if we happen to find them in the opposite order.

Using the metric of association defined above we implemented an algorithm for grouping characters into two-character words which performed quite acceptably. We obtained association measures from a database of 2.6 million characters of Chinese newspaper text. Association scores were collected for every sequence of two characters over the entire corpus. This information then served as the association database for the parsing algorithm.

The parser works by considering input sentences a phrase at a time; for our purposes a phrase is any string of characters flanked on each end by either a punctuation mark or the beginning or end of a story. Within each phrase, the association scores for each pair of adjacent characters is looked up. The pair having the highest score — or in cases where several pairs have the same highest score, the rightmost pair of this set — is grouped off. The algorithm is then applied recursively to the two pieces of the phrase not including the two characters just grouped. In practice, we found it useful to specify that no pair of characters with less than a prespecified association strength would be grouped.

As an example of this algorithm's application, consider the example sentence given above:

我 弟 弟 現 在 要 坐 火 車 回 家

'my (younger) brother wants to ride the train home now'

The association scores between successive pairs of characters are as given in the table below:

我 弟 0.00

弟 弟 10.44

弟 現 0.00

現 在 4.23

在 要 -2.79

要 坐 0.00

坐 火 0.00

火 車 7.31

車 回 2.06

回 家 4.69

The highest association strength is between the two instances of 弟 , so the algorithm groups them together first as 弟 弟 'younger brother'. This leaves the singleton 我 on the left, which has nothing to group with, and the remainder of the phrase on the right: 現 在 要 坐 火 車 回 家 . Within that chunk 火 車 , 'train', has the highest association. We are then left with two further chunks, 現 在 要 坐 and 回 家 . Supposing that we set a lower limit of 2.5, below which association strength we will not group characters, then of what is left, only 現 在 and 回 家 will be grouped. This will yield the following bracketing, which is correct:

我 (弟 弟) (現 在) 要 坐 (火 車) (回 家)

In the appendix we give a sample newspaper story showing the unedited results of the application of the algorithm, using a minimum association strength for grouping of 2.5.

2.2 Performance of the method

We have tested the algorithm on 69 randomly selected stories from the newspaper corpus (i.e., on the corpus over which the data were collected); the 69 stories comprised about 19,500 characters or approximately 0.7% of our database. The algorithm was coded in C and tested on a Sun 3/280 with 32 Mbytes of memory. On this hardware the program ran at a rate of approximately 260 characters per second. In the current implementation, the association scores are computed on the fly — following equation (2) — from the independent frequencies of the characters (this latter information being stored in a histogram) and the joint frequency of the bigram. The bigram statistics are

stored in a table comprising bigrams which occur at least 5 times in the database of 2.6 million characters. The size of the bigram table is approximately 500 Kbytes. (8 bytes were used to store each bigram, so substantial compaction of the table could be achieved if desired.) The bigrams are sorted by a predefined character order and lookup is achieved by binary search.

The stories were parsed using the algorithm — with a minimum association strength for grouping of 2.5 — and the output then scored. We were naturally only concerned with scoring two-character combinations, considering whether they constituted a word, and whether the algorithm grouped them successfully. The following subsection outlines the criteria used to decide upon the correctness of a grouping.

2.2.1 Criteria for grouping. First of all, we give some uncontroversial examples of what we would consider a word; such uncontroversial examples might be reasonably expected to be found in a dictionary. The following are from the corpus and are successfully grouped by the algorithm:

消 息, 協 定, 娛 樂, 電 影

具 備, 提 供, 良 好, 觀 摩

Secondly, instances of productive morphological processes such as compounding or affixation were also considered to be words although one may or may not expect to find such cases in dictionaries in general. For example, nouns with preceding nominal or adjectival modifiers are treated as words in Chinese [7], so the following examples are considered correct groupings:

小 組, 酒 樓

When the algorithm didn't group such examples, it was marked as an error:

廢 土, 黑 鵝, 左 腹, 要 犯

A noun with a suffix was considered a word, for example:

事 兒

A verb with a resultative complement was considered a word [8]; not grouping such examples was considered an error:

射 穿, 擊 中, 割 斷

Idiomatic verb-object constructions were considered words, such as:

停 電, 減 速, 投 稿, 跳 繩

Non-idiomatic ones are optional, and the algorithm was not penalized if it failed to group such cases (see below for other cases of optional groupings):

洗 砂, 組 黨

Pronominal or other function word objects should not be grouped with the preceding verb, however, and such cases would be marked wrong:

愛 我，在 此

Examples of negation plus adjective were considered words, unless the adjective ought to be grouped with the following noun. Not grouping examples like the following was considered wrong:

不 多， 不 小

Month names and some references to years such as *this year* are considered words:

二 月， 五 月，今 年

Combinations of subject and verb are not considered words. So the following combination shouldn't be grouped (an '*' is used to indicate implausible groupings):

{* 他 看} {* 他 在}

A preposition and its object are not considered a word, e.g.

{* 被 李} 發 現

Given the fuzziness of the definition of word in Chinese, many cases fall in the borderline. Some classes of groupings were therefore considered optional. A single numeral and the following measure word are considered optional groupings and the algorithm is not penalized either for grouping or failing to group such examples. The following examples belong to this group:

兩 個， 一 家

A single noun and a following locative term (postposition) is another optional category:

心 中，店 裡

However, if the noun belongs to a preceding constituent, such grouping would be wrong:

{ 理 髮}{* 店 裡}

A single verb and a following grammatical marker constitutes an optional grouping:

變 了

Such grammatical markers function roughly as suffixes in Chinese and it is therefore acceptable to group them with a preceding verb as a word.

Except with the optional cases as defined above, the scoring was fairly rigorous in that whenever there was some doubt as to the correctness of a grouping produced by the algorithm, that grouping was counted as wrong; included in those counted as wrong were

any attempt to group subparts of four-character idioms (成 語) — for example, phrases like

不 {* 以 為 } 然, {* 如 火} 如 荼

which as a class are not uncommon in Chinese text. Furthermore, any attempts to group parts of long strings of digits, names in transliteration, or other non-analyzable strings of characters are considered wrong, for example:

{* 一 九}{* 七 九}, {* 二 零}{* 六 號},

達{* 爾 文},{* 布 魯}{* 克 林}, 三{* 明 治}

We also marked as wrong cases where the algorithm missed a two-character proper name. In the case of three character names, it was considered an error to group the family name and part of the given name.

Finally, note that a wrong grouping very often simultaneously prevents the correct grouping and in such cases two errors were counted — one incorrect grouping and one missed grouping — as in the following case, which means *for a foreign country*:

{* 對 外} 國

2.2.2 Results. The results for the test were quite promising. The algorithm grouped 5828 pairs of characters. Of these, 567 were incorrect. In addition, there were 347 two-character words which should have been grouped but which were not found by the algorithm. So, of the groupings the algorithm produced, about 90% were correct ((5828−567)/5828). In the terminology of document retrieval, there was a *precision* of 90%. Of the total number of two-character words in the stories, the algorithm found 94% of them ((5828−567)/(5828−567+347)); again, in the terminology of document retrieval, there was a *recall* of 94%. On the other hand, of the 567 which the algorithm incorrectly grouped, 246 were trivially detectable: these were either groupings of Roman letters or other special (non-Hanzi) characters interspersed in the Chinese text (Chinese newspaper text often contains foreign names which are usually written in Roman letters where each letter takes up the space of one character) or groupings of strings of numbers. These kinds of errors could easily be detected and compensated for. Discounting these errors then, the errors which were not trivially detectable were 321 (567−246). Subtracting 246 from the total number of groupings produced yields 5582, and a total precision of about 94% ((5582−321)/5582).

What were the reasons for the less than perfect recall or the less than perfect precision? Clearly, the major reason for recall failure was a low association score for those pairs of characters which should have been grouped but weren't. A low association score may result either from a relatively low frequency rate of the character pair in question or from a relatively high frequency rate of the individual characters. These problems can in principle be corrected to some extent by lowering the minimum association score for grouping (which will unavoidably lower the precision), enlarging the

training corpus, or by using a dictionary. Among the recall failures we can discern a few different types: (i) Common words that simply didn't have enough representation in the corpus:

岳 母, 合 金, 扁 鑽, 花 瓶, 西 元

(ii) Technical terms that haven't become everyday vocabulary:

渦 輪

(iii) Styles not well represented in the corpus, such as fortune-telling:

手 相, 大 運

A final type of recall failure came from choosing the wrong pair out of two possible groupings (see above, end of section 2.2.1). This type of mistake is harder to solve in the general case since both groupings are in principle possible, and which one to choose in any given case may depend upon more than local information. Below are some further examples of this type of error:

中{* 山 區}, 六{* 月 份}

We turn now to imprecision: first of all note that the examples just discussed necessarily involve imprecision as well as recall failure. Other examples of non-trivially detectable imprecision commonly involve incorrect groupings of parts of names or long compounds. So, groupings of parts of longer names (including transliterated foreign names) and of idioms pose problems for the method. Again, examples such as

不 {* 以 為 } 然

{* 布 魯}{* 克 林}

serve to illustrate such cases. The point with at least some of these examples, as noted above, is that they have no correct internal analysis and it is therefore wrong to group parts of them. The particular groupings chosen in such cases depend, of course upon the association scores of the characters involved; the groupings chosen will have the (rightmost) highest association score for the string.

Similarly, parts of long compounds may be incorrectly grouped when adjacent characters within the compound happen to be of relative low independent frequency, but sufficiently high joint frequency. For example:

中{* 澳 漁} 業

Finally, some very frequently cooccurring words are also often wrongly grouped:

他 說

Even though each of the characters in this example is itself very common, the collocation

is also sufficiently common that the association score is high enough for the algorithm to group the characters together.

To summarize: the algorithm, with a minimum association strength for grouping of 2.5, had a recall of 94% and a precision also of 94%. Since there are on average, in our sample, about 83 two-character words per story, this amounts to an error rate of about 5 incorrectly grouped words per story and about 5 words missed per story.

It is important to bear in mind that the association data are derived from the entire corpus of 2.6 million characters. Therefore it is perfectly possible for a two-character word to occur quite frequently in a particular story, and yet not be grouped by the algorithm even when two-character words which are less frequent *in that story* are grouped.

2.2.3 Varying parameters of the model. Clearly there are a number of parameters in the model which can be varied. One is the minimum association strength for grouping, which we set at 2.5 in the above evaluation. Another variable parameter is the size of the training corpus. In Figure 1 we plot the effects of varying the minimum association strength for grouping. As expected the percentage recall decreases with a increase in the minimum association strength, since fewer groupings are permitted. For the same reason, the precision increases: those groups which are permitted are more likely to be correct. The value of 2.5 chosen for the evaluations above is apparently a reasonable tradeoff between recall and precision since lowering the minimum association strength yields a fairly steep decline in precision and raising the minimum association strength yields a somewhat steeper decline in recall.

In Figure 2 we plot the effects of varying the corpus size. The three corpora used for this plot were: (1) half of the original 2.6 million characters (approx 1.3 million); (2) the original corpus of 2.6 million characters; (3) the original corpus plus another corpus of approximately equal size yielding a combined count of about 5.4 million characters. All three corpora subsumed the 69 stories referred to in the discussion above. In these tests the minimum association strength for grouping was kept constant at 2.5. Not surprisingly, the smallest corpus size yields worse results than the corpus of 2.6 million characters. In particular, since any given two-character word is generally less frequent in a smaller corpus than in a larger corpus, one would in particular expect to find a lower recall simply because some words are underrepresented in the smaller corpus. Unfortunately we do not, at this time, have so ready an explanation for why the largest corpus also shows a slightly poorer precision. We suspect however that the poorer precision may be due in part to the non-uniformity of style of the largest corpus. The original 2.6 million characters consisted virtually exclusively of newspaper text, and this generalization is necessarily true of any subset of that corpus including the smaller 1.3 million character corpus described above under (1); training corpora (1) and (2), therefore were fairly uniform in style. The added material in the third corpus was more varied in style, much of it coming from sources besides newspapers. Vocabulary in many languages — and especially in Chinese — is highly sensitive to the style of text, and we

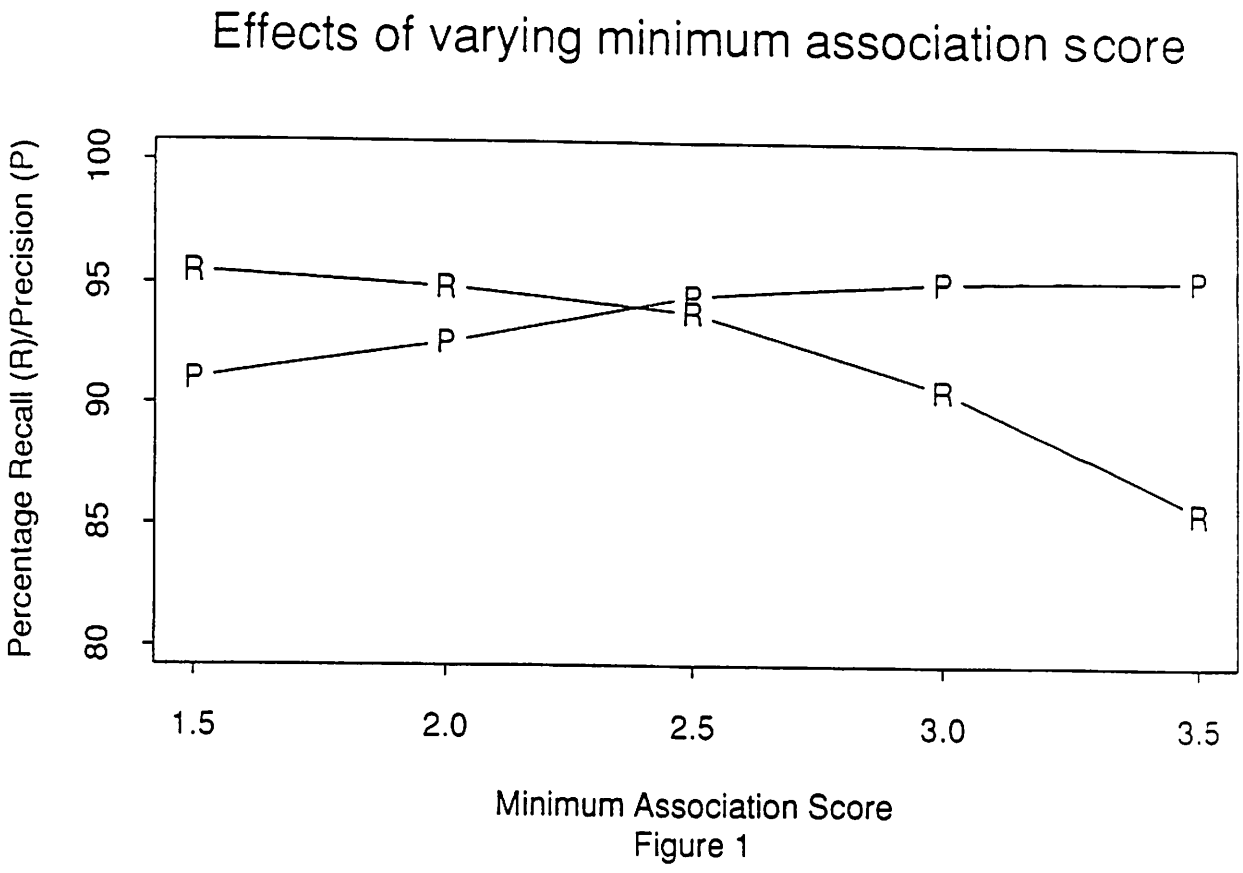

Figure 1

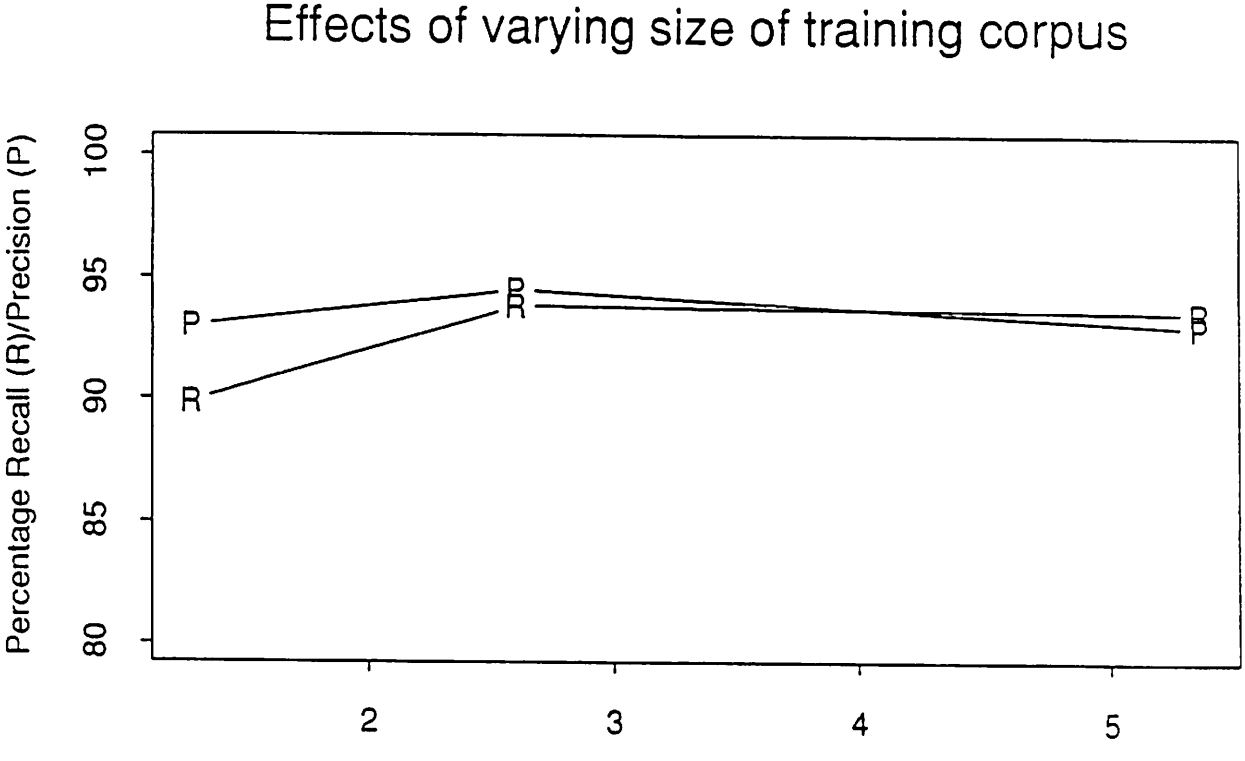

Figure 2

suspect that the non-uniformity of the larger corpus is responsible for the lower performance. To see this point consider the example

{ 自 行 } 車

which is parsed correctly, as shown, with training corpus (2) but incorrectly, with the last bigram grouped, using training corpus (3). If the two corpora represent the same style we could legitimately think of corpus (3) as being a sample from the same population as that from which corpus (2) was taken. Under that assumption, since corpus (3) is roughly twice the size of corpus (2) we would expect that the unigram and bigram counts would roughly double for the larger corpus. In fact what happens is that the first bigram has a lower than expected count in the larger corpus (311 in corpus (3) versus 218 in corpus (2)), whereas the second bigram is about double in count, as expected (214 versus 121). Furthermore, the second and third characters have a lower than expected representation for the larger corpus (12100 versus 8087 for 行; 7789 versus 4432 for 車), whereas the first character has about the expected representation (13288 versus 6107). It is easy to verify that these counts will yield the results obtained, and they suggest that the corpora are samples from somewhat different populations or styles.

Unfortunately, we do not currently have a clearly uniform corpus larger than the 2.6 million characters used for the majority of the tests reported on in this paper. In any event, what is clear from the plot in Figure 2 is that the relationship between corpus size and performance is more complex than the relationship between minimum association strength for grouping and performance.

3. DISCUSSION

We have concentrated on grouping characters into two-character words. While this covers a large percentage of the problem of low-level text analysis for Chinese, there are cases where the specific approach discussed in this paper fails. For example, four character idioms (as discussed above) are obviously not correctly treated by this method, and neither are long proper names. One could in principle generalize the equation for association given in (2) above to handle the association of a sequence of characters of length n, where n is greater than two. We have begun to pursue similar statistical methods for grouping strings of characters longer than 2, using data from larger corpora, but we have not performed any analyses of the results to date. More problematic for statistical approaches in general are various productive word-building processes in Chinese which would not be readily handled by such methods; rather they would require some sort of morphological analysis in the general case. Among such processes are so-called A-not-A verbs used in 'yes'/'no' questions (看 不 看), and reduplication of certain forms with (often) intensifying meaning (高 高 興 興 , 冷 冷). [9] (Chapter 3) contains some discussion of productive morphology and the issues it raises for preprocessing Chinese text.

The work reported on here relates to a couple of previous pieces of research. [10] reports on an experimental system for automatic word identification for Chinese which

uses the relaxation technique combined with a dictionary of 3748 words. Their system is not limited by design to identifying two-character words, though it is clear from the examples they give that many of the identified words are two characters in length — again because of the overwhelming frequency of such words among the cases of multi-character words in Chinese. The dictionary in this system serves as a filter on possible analyses, only allowing analyses which would yield words known from the dictionary. [10] reports optimal performance with a 95.62 percent identification rate, which seems comparable to our rate though they are not entirely explicit about their criteria for correctness. Under this optimal condition their algorithm requires about 18 iterations through the sentence and runs at a rate of 0.8561 seconds per character. In contrast, our algorithm requires only one iteration (following the recursive splitting described in section 2.1) and runs at a much faster rate (260 *characters per second*), though it is clear that part of the speed difference is due to differences in hardware rather than algorithmic differences ([10]'s algorithm was implemented on an IBM PC/AT compatible machine, with 640K of memory).

Secondly, the current research is related to an early piece of research on stochastic grammars reported in [11]. This work describes a statistical discovery procedure for acquiring a wordlist for English from texts from which whitespace has deliberately been removed; the problem posed then was entirely similar to the problem posed by real Chinese text. The criterion used by [11] for measuring success was the ability of the algorithm to discover the positions of the original whitespace in the text, a metric which for obvious reasons cannot be used to evaluate our algorithm for Chinese. On that task his algorithm performed at about 75 percent at the end of training.

Obviously the identification of multi-character words would be improved by having an online Chinese dictionary. As noted, dictionaries suitable for coverage of unrestricted text are not conveniently available at the present time. However, the approach taken here shows that even without a dictionary, reasonable local groupings of characters can be achieved.

4. ACKNOWLEDGMENTS

We wish to thank United Informatics, Inc., for providing us with the databases of Chinese text and for giving us permission to use a small excerpt of the text in the appendix; B-H. Juang and C. H. Lee of AT&T Bell Laboratories were also instrumental in acquiring the database. We also wish to thank K. Church and N. Jablon for helpful discussion.

REFERENCES

[1] C. Y. Suen. "Computational Studies of the Most Frequent Chinese Words and Sounds", World Scientific, Singapore, 1986.

[2] Y. Yang. "Studies on an Analysis System for Chinese Sentences", PhD dissertation, Kyoto University, 1985.

[3] C-L. Shih and M. Liberman, "A Chinese Tone Synthesizer", ATT Bell Labs Technical Memorandum, 1986.

[4] C-L. Shih, "The Prosodic Domain of Tone Sandhi in Chinese", PhD dissertation, University of California, San Diego, 1986.

[5] Y. Yang. "Combining Prediction, Syntactic Analysis and Semantic Analysis in Chinese Sentence Analysis", Proceedings of the International Joint Conference on Artificial Intelligence, pp. 679-681, 1987.

[6] R. Fano, "Transmission of Information", MIT Press, Cambridge MA, 1961.

[7] 朱德熙,<<現代漢語形容詞研究>>語言研究 1: 83-112. 1956.

[8] C. Li and S. Thompson, "Mandarin Chinese: a Functional Reference Grammar", University of California Press, Berkeley, 1981.

[9] L-J. Lin, "A Syntactic Analysis System for Chinese Sentences", M.S. dissertation, National Taiwan University, 1985.

[10] C.-K. Fan and W.-H. Tsai. "Automatic Word Identification in Chinese Sentences by the Relaxation Technique", Computer Processing of Chinese and Oriental Languages, 4: 33-56, 1988.

[11] D. C. Olivier. "Stochastic Grammars and Language Acquisition Mechanisms", Ph.D. dissertation, Harvard University, 1968.

APPENDIX
Output from the Parser

The following is a complete unedited parsed story. It is to be read like a Chinese newspaper, top to bottom and right to left. The groupings are indicated by pairs of curly braces.

堤	市	⌒	府	居		⌣	在	不	⌣	林	訊	
外	⌒	取	⌣	民	⌒	⌒	⌒	法	河	⌣	⌣	假
的	所	締	⌒	⌣	士	情	該	⌒	與	區		×
⌒	轄	⌣	加	⌒	林	形	處	砂	⌒	社	⌒	
基	⌣	。	強	要	⌣	⌣	⌣	石	淡	子	台	⌐
隆	的		⌣	求	⌒	⌒	盜	⌣	水	堤	北	
⌣	⌒	⌒	⌒	⌣	地	嚴	採	⌒	⌣	外	⌣	⌒
河	士	台	派	市	區	重	⌒	業	河	⌒	市	台
與	林	北	員	⌒	⌣	⌣	砂	者	,	基	⌒	北
⌒	⌣	⌣	⌣	政	⌒	,	石	⌣		隆	士	⌣

取	⌒	⌒	用	砂	最	⌣	,		川	府	⌣	淡
⌣	申	紛	⌣	⌣	近	洗		⌒	⌣	⌣	⌒	水
⌒	請	紛	⌒	⌒	⌣	砂	即	執	公	⌒	五	⌣
河	⌣	⌣	原	業	有	,	⌒	行	地	配	年	河
水	水	向	有	者	⌒		不	⌣	⌒	合	⌣	,
⌣	權	⌒	⌣	⌣	部	⌒	再	⌒	收	⌣	元	
⌒	,	經	⌒	,	份	但	⌣	拆	回	⌐	月	自
進		濟	設		⌣	是	⌒	除	⌣		市	⌒
行	⌒	⌣	備	⌒	⌒	⌣	許	⌣	案	⌒	⌒	七
⌣	抽	部	⌣	利	採	⌒	可	後	⌙	河	政	十

洗砂，（而且）藉洗砂之名，暗行盜採之實，（嚴重）（違反）（水利）法。市（政府）工（務局）（表示），市（政府）對不法（砂石）（業者）盜採（砂石）之（行為）（甚為）（重視），（市府）已（函請）（經濟）部（審慎）（管制）水權之（申請）。（市府）為（有效）（遏阻）不法（業者）盜

採（砂石），除已飭（士林）（分局）督飭社子（聯合）（檢查）站（加強）（巡邏）（查緝）外，（同時）（配合）省市共管（河川）（聯合）（取締）（小組），以（埋伏）、（突擊）及站崗（交替）（方式）（嚴格）（查緝）。凡（查獲）（違反）（水利）法者（一律）（依法）（處理）。