AN IBM-PC ENVIRONMENT FOR CHINESE CORPUS ANALYSIS

*Robert Wing Pong **Luk***

Department of Chinese, Translation and Linguistics, City Polytechnic of Hong Kong
Email: CTRWPL92@CPHKVX.BITNET

ABSTRACT

This paper describes a set of computer programs for Chinese corpus analysis. These programs include (1) extraction of different characters, bigrams and words; (2) word segmentation based on bigram, maximal-matching and the combined technique; (3) identification of special terms; (4) Chinese concordancing; (5) compiling collocation statistics and (6) evaluation utilities. These programs run on the IBM-PC and batch programs co-ordinate the use of these programs.

I. INTRODUCTION

Corpus analysis utilities are developed and widely available for English. For example, the Oxford Concordance program is available for over 10 kinds of mainframe computer (Hockey and Martin, 1987) and the Longman mini-concordancer (Trible and Jones, 1990) is available for the sales. Further enhancement of these utilities include compiling collocation statistics (Smadja, 1993) and semi-automatic gloassary construction (Tong, 1993). Current research has focused on bilingual corpora (Gale and Church, 1993) with the alignment of parallel-text becomeing an important technical problem. However, there has been little development of corpus analysis tools for Chinese. Since using Chinese for computers has only become more generally availablein the last ten years, analysis utilities for Chinese are not widely. Although no integrated environment is available for Chinese corpus analysis, many specific analysis programs have been reported in the literature (Kit *et al.*, 1989; Tong *et al.*, 1993; Chang and Chen, 1993; Zhou *et al.*, 1993). A Chinese concordance program and a character-list extraction program are freely available from a Singapore (FTP) network site (Guo and Liu, 1992). However, the programs run in the SUN workstations while many users, particularly non-computing experts, interact with an IBM-PC in Chinese, rather than a SUN workstation.

The rapid advance of microcomputers has mitigated many storage and processing speed problems. As for storage, the hard disk capacity can reach as high as 340M bytes which is adequate in comparison with the demand for a corpus (8M bytes from the PH corpus) and a dictionary (10M bytes). Using a 486 processor, the processing speed is acceptable if the user expect data to be analyzed over-night, similar to submitting a batch job to a mainframe computer. For example, the utilities we are developing ranked around 42,000 words in a few minutes and produced about one-hundred lines of keyword-in-context in a few seconds for a 4 million character Chinese corpus.

This paper describes our effort to develop corpus analysis programs for Chinese. The programs are written in Turbo C++, implemented on an IBM-PC (486) with a 120M byte hard disk. The programs are divided into several types:

a. format conversion program
(norm.exe, phseg.exe, wform.exe)
b. extraction of characters, bigrams and words
(exsega.exe, exsegmi.exe,bigram.exe, miana.exe, worddh.exe, wranka.exe, wlranka.exe)
c. word segmentation programs
(biseg1.exe, whash.exe, bimaxn.exe)
d. concordancing programs
(kwic.exe, kwicw.exe)
e. collocation statistics programs
(cxtract.exe, cxtractw.exe, cxstat.exe, cxstatw.exe)
f. general (evaluation) programs
(wcomp.exe, segperf.exe)

To run these analysis utilities, a Chinese computing environment called Eten must be set up; otherwise Chinese characters cannot be displayed or entered. Since there are many different Chinese characters (i.e. 13,000) compared with Western languages, each Chinese character is specified by two bytes instead of one. However, many document includes both single-byte characters and two-byte Chinese characters. Thus, the conversion program, norm.exe, is used to convert all the single-byte characters (i.e. A..Z,a..z,,,,.;:;,!,@,#,$,%,^,&,*,(,),-,+,=,|,\,/,<,>,',{,},[,], 0..9, ~, <space>, _ and `) into their corresponding two-byte equivalent, for simplicity. For example, the-single-byte character "a" is converted to "ａ" (2-byte). This program also changes the document into a clause or phrase format, using the *-e option*, where a new line is inserted after a punctuation mark (e.g. comma or full stop):

香港自古以來就是中國的領土，
一八四〇年鴉片戰爭以後被英國佔領。
一九八四年十二月十九日，
中英兩國政府簽署了關於香港問題的聯合聲明，

Figure 1: Extract of the Hong Kong Basic Law in clause format.

If the text are segmented into words by space or "/" markers, it is possible to change or delete these markers using the *-s option*. Once, the document is converted into two-byte format using norm.exe, the other utilities can be used. Batch programs can be written to use these utilities. For example, the following batch program extracts different characters, performs bigram segmentation, extracts different words and obtain only the top 10% of the extracted words for compiling key-word in contexts and collocation statistics.

```
norm -t %1 -o corp.tmp -e 2 -s 5
/* 1-byte to 2-byte; phrase format; delete space */
exsega -t corp.tmp -w 2 -b 0
/* extract different characters and bigrams */
bigram -m 10
/* sort bigram and extract top 10% */
biseg1
/* segment using the top 10% bigrams */
worddh
/* extract different words */
wranka -m 10
/* sort and extract top 10% words */
kwic -t corp.tmp -k words.cut > kwic.lst
/* concordancing on the top 10% words */
cxtract
/* extract different characters from contexts */
cxstat
/* compile collocation statistics */
```

II. EXTRACTION PROGRAMS

The extraction programs assume that the text is not segmented. Thus, norm.exe should be used to remove markers from the segmented text.

The programs, exsega.exe and exsegmi.exe, extract different characters and their co-occurring characters, stored in cfreq.tmp (Fig 2) and bifile/mifile.tmp, respectively. The first program obtains the co-occurrence frequencies while the second obtains the mutual information. By default, the programs do not count punctuation but this can be override using the *-a option*. The different characters can be supplemented with information about their frequencies, percentages and cumulative percentages if the *-w* option is set to *2*.

```
政/   909 3.529 3.5    1
的/   905 3.513 7.0    2
行/   789 3.063 10.1   3
法/   647 2.512 12.6   4
港/   645 2.504 15.1   5
香/   630 2.446 17.6   6
```

Figure 2: Part of the extracted single characters from the Hong Kong Basic Law. The characters are ranked by their frequencies. The first number is the frequency, followed by the percentage, cumulative percentage and rank number.

By default, all the different characters are stored. However, sometimes only the most frequently or infrequently occurring characters are interesting candidates for further investigation (e.g. concordancing). The user can select characters by their frequencies (i.e. *-f* and *-g options*), the top or bottom N% (i.e. *-m* and *-n options*), their ranks (i.e. *-r* and *-s options*) and by their frequencies above two standard deviations plus the mean (Smadja, 1993) (i.e. *-z option*).

By default, the extracted bigrams have frequencies above unity but this can be override using the *-b option*. The bigrams stored can be sorted according to their frequencies or their mutual information in descending order using bigram.exe and miana.exe, respectively. The sorted bigrams are stored in bifile.rnk or mifile.rnk. The user can select different bigrams using options available for exseg programs (i.e. *-f*, *-g*, *-m*, *-n*, *-r*, *-s* and *-z* options). Both programs give the frequency distribution of the bigram frequencies and the log of their frequencies. The selected bigrams will become useful for detecting compound nouns or word segmentation (Zhang *et al.*, 1992).

Given the text is segmented by "/" markers (space markers can be converted using norm.exe), worddh.exe can extract all the different words from the text and compute word frequencies. The program extracted 42,613 words from the PH corpus. There is no limit to the number of different words that it can extract but it needs some disk space to hold temporary files. The extracted words are stored in words.lst and they are sorted in descending frequencies using wranka.exe. In addition, wlranka.exe sorts the extracted words firstly by word length and secondly by their frequencies. This is particularly useful to examine compound nouns, technical terms and translated words as they tend to be long. Furthermore, the segmentation program, whash.exe, needs the words to be order by their length.

III. WORD SEGMENTATION PROGRAMS

Unlike English, Chinese words are not delimited by any tentative markers like spaces although Chinese clauses are easily identified (Fig 1). Many segmentation programs were proposed (Chiang *et al*, 1993; Fan and Tsai, 1988). We have re-implemented the maximal-matching technique (Kit *et al*, 1989) using a word list, *L*, because it is simple to program and achieved one of the best segmentation performance (1-2% error rate). However, the segmentation accuracy is degraded significantly (to 15% error rate in (Luk, 1993)) when the text has many compound nouns and technical terms since the accuracy depends on the coverage of *L*. A word segmentation program using bigrams as well as combining bigrams and maximal-matching was subsequently developed.

The basic idea of maximal-matching is to match the input clause from left-to-right with entries in the given word list, *L*. If there is more than one matches, the longest entry is selected. The process iterates with the remaining clause at the end with the clause matched with the longest entry. Apart from maximal-matching,

whash.exe divides and output the text in the clause format (Fig 2). The file that holds the word list can be specified using the *-b option* and the text using the *-t option*. The word list should rank the words, firstly, by their length in descending order (use wlranka) and, secondly, by their frequencies. Usually, the segmented clauses are displayed on the screen for visual inspection after which the output can be redirected using the *> option* (MS DOS 5.0 option). The current whash.exe program can hold around 20,000 Chinese words in the main memory for segmentation but this is not large enough for a general Chinese dictionary (Fu, 1987) which has about 54,000 entries.

The bigram technique does not need any dictionary for segmentation. This technique needs a set of bigrams extracted from the text or from a general corpus. Typically, the top 10% of the bigrams are captured and ranked according to their co-occurrence frequencies (CF) or mutual information (MI). This is due to the fac that if the distributions of CF and MI are normal, then the top 10% corresponds to the 10% significance level. The distribution of MI typically does appear normal but not for CF. The top N% bigrams are stored in either bifile.cut or mifile.cut. The bigram segmentation program, biseg1.exe, loads the bigrams using the *-b option*. A segmentation marker is placed between two characters in the text if the bigram of these two adjacent characters does not appear in bifile.cut or mifile.cut. This segmentation is the same as performing nearest-neighbour clustering of substrings (Luk, 1993). The program detected many non-words depending on N. However, the number of non-words are significantly reduced if we restrict to examining only the top N% (say 10) of the frequently occurring words.

Both maximal-matching and bigram techniques were combined, in order to detect words not in the word list and reduce the amount of non-words detected (Luk, 1993). Maximal-matching is carried out first and the bigram technique is used to combine consecutive single-character words in the segmented text since words not in *L* are usually segmented into smaller ones by maximal-matching. The test data shows that the combined technique reduced the error rate by 33% and detected 33% of the desired words not in *L*. The combined techinque is written as a batch program as follow:

```
whash -b wordlst.txt -t text > text.tmp
/* maximal-match with existing word list */
bimaxn -t text.tmp
/* combine single-character words from segmented text */
worddh
/* extract words from segmented text */
wlranka -t words.lst
/* rank words by their lengths */
whash -b wordl.rnk -t text > text.res
/* maximal-match with identified words */
```

IV. CONCORDANCE PROGRAMS

We modified the concordance program by Guo and Lin (1992) since the program assumed that the main memory can hold the entire corpus or text. Instead, the modified program loads a portion called a page into the main memory and performs matching to find the appropriate contexts. The page size can be changed using the *-p option* but we found that the program operates well at *-p 10000* (which is the default size). The modified programs, kwic.exe and kwicw.exe, can process files of size just over 2G bytes which is much bigger than the hard disk.

```
key=社會/
38 民共和國成以後，我國<社會>逐步實現了由新民主主
38 實現了由新民主主義到<社會>主義的過渡． 生產資料
39 渡 生產資料私有制的<社會>主義改造已經完成，人剝
39 削人的制度已經消滅，<社會>主義 制度已經確立．工人
```

Figure 3: The keyword-in-context (kwic) format produced by kwic.exe. Note that the line numbers are on the left-most positions and the keyword is delimited by "<" and ">".

A keyword file must be specified using the *-k option* and each keyword should be terminated by "/". The number of characters in the left and right contexts can be specified in bytes, using the *-l* and *-r* options respectively. If *-n 0* is specified then line numbers will appear on the left. There are additional options for indexing in the original concordance programs but these options are not important in the current implementation. The kwicw.exe deals with segmented text. Here, the -l and -r options specify the number of words in the left and right contexts. The length of each context (approx. 1000 characters allocated) can hold 20 words assuming that each word has 24 characters.

V. COMPILING COLLOCATION STATISTICS

Collocation statistics (Fig 4) refers to the frequencies of each different words or characters at different positions in the contexts of a keyword. These frequencies are useful for detecting significant collocation in English but these frequencies are tedious and error prone to compile by hand. We have also written programs to compile these statistics for Chinese but factorial analysis (Biber, 1993) still remains to be implemented.

Chinese concordancing is carried out first to extract the relevant contexts. The output of concordancing should be stored in kwic.lst. Then, extract1.exe will extract all the different words in the context, using an FSM to decode the kwic format. The program sorts these words according to their frequency of occurrence in the context. The different words are stored in extract.crk and the user can select candidates using options as in exsega.exe Next, exstat.exe compile the frequencies of these different words at different positions in the contexts. The statistics are stored in extract.sta. For segmented text, kwicw.exe, extractw.exe and exstatw.exe are used instead.

```
key=的/
<的>/   [ 71]  0  0  0  0  0< 71>  0  0  0  0  0
和/     [ 50]  5  6  9  2  0< 0>   1  4 10  8  5
有/     [ 24]  3  7  1  2  1< 0>   0  2  1  1  2
國家/   [ 18]  0  0  2  1  1< 0>   1  0  1  1  2
```

Figure 4: Collocation statistics. The different words in the contexts are displayed on the left and the square brackets show the frequency of occurrence in the context of the keyword. The angle brackets indicate the position of the keyword.

Unlike Smadja (1993), the keyword may be part of a Chinese word. Thus, the program can compile statistics about different prefixes, suffixes or stems of a Chinese word. This is particularly interesting for investigating translated terms and compound nouns.

VI. EVALUATION PROGRAMS

Two programs were written to measure the performance of word segmentation and word identification. For segmentation, segperf.exe examines two identical texts that were segmented by different methods. The program shows the amount of segmentation error, the number of clauses, the number of clause that are segmented correctly and the amount of over- or under-segmentation. Files of the segmented texts are specified by the *-a* and *-m* options. The user can inspect parallel clauses to examine individual differences in segmentation by setting the *-d* (diagnostic) option to *1*.

For word identification, wcomp.exe compares two sets of different word lists and determines the amount of word overlap. The program shows the distribution of word overlap for different length of words. This is important since long words tend to be compound nouns that are not in a general dictionary. Using the *-i* and *-j options*, the program saves words that overlap and words that do not overlap, respectively.

REFERENCES

BIBER, D. (1993) "Co-occurrence patterns among collocations: a tool for corpus-based lexical knowledge acquisition", *Computational Linguistics*, **19**, n3, pp.531-538.

CHANG, C-H. AND C-D. CHEN (1993) "Chinese part-of-speech tagging using an HMM", *Proceedings of Computational Linguistics: Research and Applications*, Xiamen, PRC, pp. 114-119.

CHIANG, T.H., J.S. CHANG, M.Y. LIM AND K.Y. SU (1993) "Statistical models for word segmentation and unkown word resolution", *Proceedings of ROCLING V '93*, pp. 123-146.

FAN, C.K. AND W.H. TSAI (1988) "Automatic word identification in Chinese sentences by the relaxation technique", *Computer Processing of Chinese and Oriental Languages*, **4**, n1, pp. 33-56.

FU, X-L. (1987) *Xiandiao Hanyu Tunrun Cidian*, Waiyu Jiaoxue Yu Yanjiu Publishing House: Beijing, PRC.

GALE, W.A. AND K.W. CHURCH (1993) "A program for aligning sentences in bilingual corpora", *Computational Linguistics*, **19**, n1, pp. 75-102.

GUO, J. AND H.C. LIU (1992) "PH - a Chinese corpus for pinyin-hanzi transcription", *ISS Technical report TR93-112-0*, Institute of Systems Science, National University of Singapore.

HOCKEY, S. AND J. MARTIN (1987) "The Oxford concordance program version 2", *Literary and Linguistic Computing*, **19**, n1, pp. 75--102.

KIT, C., Y. LIU AND N. LIANG (1989) "On methods of Chinese automatic word segmentation", *Journal of Chinese Information Processing*, **3**, n1, pp. 13-20.

LUK, R.W.P. (1994) "Chinese word segmentation using maximal-matching and bigram techniques", *submitted to ROCLING '94*, Taiwan.

TONG, K. S-T. (1993) "From single parent to bound-pairs: the secret life of computerese", in PEMBERTON R. AND TSANG, E. S-C. (1993) *Studies in Lexis*, Language Centre, Hong Kong University of Science and Technology, pp. 196-214.

TONG, X., C. HUANG AND C. GUO (1993) "Example-based sense tagging of running Chinese text", *Proceedings of the Workshop on Very Large Corpora, ACL-93*, Ohio State University , Columbus, 22 June.

SMADJA, F. (1993) "Retrieving collocations from text: Xtract", *Computational Linguistics*, **19**, n1, pp.141-177.

TRIBBLE, C. AND G. JONES (1990) *Concordance in the classroom*, Suffolk: Longman.

ZHANG, J-S., S. CHEN, Y. ZHENG, X-Z. LIU AND S-J. KE (1992) "Automatic recognition of Chinese full name depending on multiple corpus", *Journal of Chinese Information Processing*, **6**, n3, pp. 7-15.

ZHOU, M., C. HUANG AND J. YANG (1993) "CSTT: Chinese syntactic tagging tool with self-learning ability", *Proceedings of Computational Linguistics: Research and Applications*, Xiamen, PRC, pp. 155-160.

ACKNOWDLGEMENT

Thanks to Dr. Webster for correcting the grammatical mistakes in this paper.