A)
32

Full Text processing of Chinese Language

— an experimental system for studying Chinese History Literatures

Ching-Chun Hsieh

Research Fellow,

Institute of Information Science Academia Sinica, Taipei, ROC

Introduction

Full-text databases and associated processing technology are becoming increasingly important for library automation, on-line information service systems, and office information systems. They also play a key-note in developing new computer applications where automation of the filing and search procedures will free workers from these time-consuming tasks.

Some major research and development topics related to full-text database systems now underway include the studies of document representation, access methods for text, text analysis (in natual language), and the new hardware for text retrieval. An excellent review of text access methods and retrieval hardware was given by Faloutsos [ref. 1]. And, an excellent position paper about full-text databases was given by Tenopir [ref. 2]. Since full-text of document is usually in natual language form, the studies mentioned above are unavoidably language dependent, ie they may not directly applicable to those full-text processing systems which are not in English. This fact becomes more obvious while Chinese full-text database systems are confronted, although at present, only quite limited and scattered effort has been devoted to the development of full-text database systems in Chinese language.

In this paper, an experimental Chinese text processor(CTP) for stadying history literatures will be presented. The system provides both free-text and controlled vocabulary search for the text. The goal of this study is to find out the feasibility of using computers, especially small or microcomputers, as a tool for history studies. Our experiment shows that by applying existing technology, such a system is higyhly feasible.

The System

The project of CTP is a part of a long-term research plan, namely Computer Applications in Humanities(CAH) conducted by the Computing Center of Academia Sinica, Taipei, Taiwan, ROC. The project CTP was started in July, 1984. At present, the CTP is still an on-going project in its second phase. Therefore, as time goes, the system is unavoidably subjected to change. For the time being, we have two operational prototypes on a mini and a 16-bit micro, respectively.

A basic conceptual configuration of the CTP is shown in Figure-1. It can be considered basically as a workstation for humanity studies. As comparing the CTP to an information retrieval system, the major differences are : the abstract text-structure supporting mechanism, more supporting tools, and a reinforced application section which supports documentation generation functions, i.e. user's note-card system [ref. 7, 8], word processor etc. The most important supporting tool, at present stage, is considered to be the auto-indexing program. These major differences will be presented in the subsequent sections.

The CTP is designed to facilitate the access of full-text documents, creating user's own private index systems to both the original text and user's own note-cards, and the generations of new documents during/after studies.

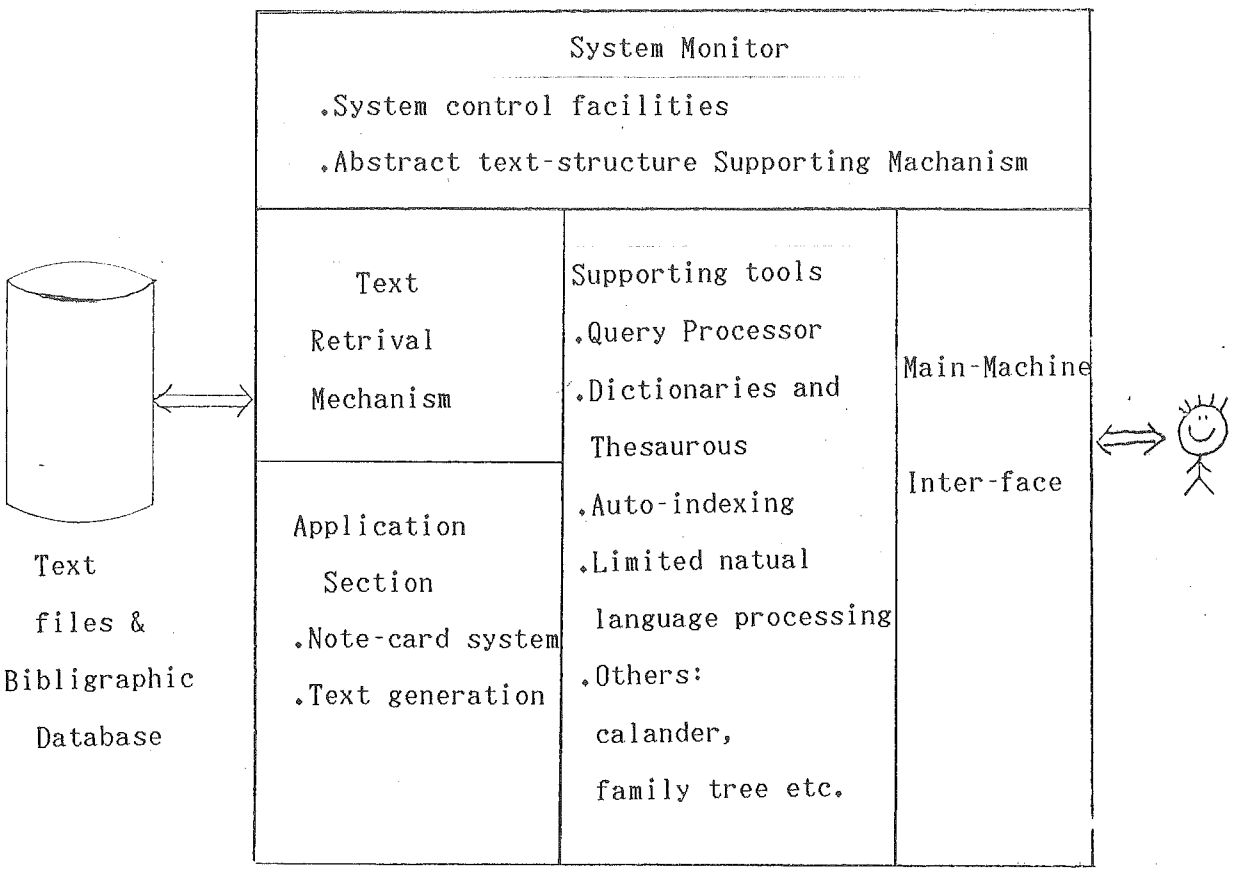


Figure-1 Base Configuration Concept of CTP

Text Representation

The object document in the full-text data-base under testing consists of all volumes of 食貨志(for food and goods in meaning) from the history books of 24 Dynasties(廿四史) of ancient China. An example of 食貨志 is shown in Figure-2 . It is of the form of an ordinary text book. In this paper, we will use full-text or text interchangably to indicate such a type of document. A complete list of the 食貨志 and associated number of characters in each book is shown in Table-1. [ref. 3]

One of the reasons to choose 食貨志 as object document is that there is no tables and figures in them. Therefore, its logical structure is simple and it is very nature to use a tree-structured representation scheme in computer to describe the text organization of 食貨志. The tree-structured representation of 食貨志 is shown in Figure-3. In Figure-3A, a logical structure of a generalized text-tree is presented. It is implemented as a part of the system monitor. In Figure-3B, an example of the physical structure of 食貨志 is shown. Each terminating node of this graph represents a paragraph of the text, respectively. Of course, for a more detailed representation, sentences may be considered to be the terminating nodes. But for the moment, we choose paragraph as the smallest unit used in the internal representation of text. If any address to the sentence level is necessary, it will be carried out by a text-scan and analysis program within a chosen paragraph.

The advantage of separating the logical structure and physical structure of the text is obvious: the system is generalized to have the capability of handling more tree-structured texts.

宋史卷一百八十一

志第一百三十四

食貨下三

會子　鹽上

會子、交子之法，蓋有取於唐之飛錢。眞宗時，張詠鎭蜀，患鬬人鐵錢重，不便貿易，設質劑之法，一交一緡，以三年爲一界而換之。六十五年爲二十二界，謂之交子，富民十六戶主之。後富民貲稍衰，不能償所負，爭訟不息。轉運使薛田、張若谷請廢益州交子務，以權其出入，私造者禁之。仁宗從其議。界以百二十五萬六千三百四十緡爲額。

神宗熙寧初，立僞造罪賞如官印文書法。河東運鐵錢勞費，公私苦之。二年，乃詔置交子務于潞州。轉運司以其法行則鹽、礬不售，有害入中糧草，遂奏罷之。四年，復行於陝西，

史記卷三十

平準書第八

漢興，接秦之弊，丈夫從軍旅，老弱轉糧饟，作業劇而財匱，自天子不能具鈞駟，(一)而將相或乘牛車，齊民無蓋藏。(二)於是爲秦錢重難用，(三)更令民鑄錢，(四)一黃金一斤，(五)約法省禁。而不軌逐利之民，蓄積餘業以稽市物，物踊騰貴，(六)米至石萬錢，馬一匹則百金。(七)

(一)索隱天子駕駟馬，其色宜齊同。今言國家貧，天子不能具鈞色之駟馬。漢書作「醇駟」，「醇」與「純」同，純一色也。或作「辭」，非也。

(二)集解如淳曰：「齊給無有貨賤，故謂之齊民。若今言『平民』矣。」晉灼曰：「中國被教之民也。」蘇林曰：「無物可蓋藏也。」

(三)集解漢書百官表曰大司農屬官有平準令。索隱大司農屬官有平準令丞者，以均天下郡國轉販，貴則賣之，賤則買之，貨賤相備偏，歸于京都，故命曰「平準」。

漢初貧弱

%通%

令民鑄錢

物價高漲

Figure-2 : An example of the text of 食貨志

Name of the books 書名	Number of characters 字数
漢書(上、下)	３６４１１
史記	１４３４３
舊唐書(上、下)	２４８０７
唐書(一至五)	３１３８０
隋書	１２０５３
魏書	１１６１５
晉書	１１０８５
舊五代史	８８１０
遼史(上、下)	４９５６
金史(一至五)	５２３２０
宋史(上、下)	２１４９９４
元史(一至五)	６８５３３
明史(一至六)	６８５１３
Total number of characters 共計	５５９８２０

Table-1：All the volumes of 食貨志 and their number of characters

Access Method

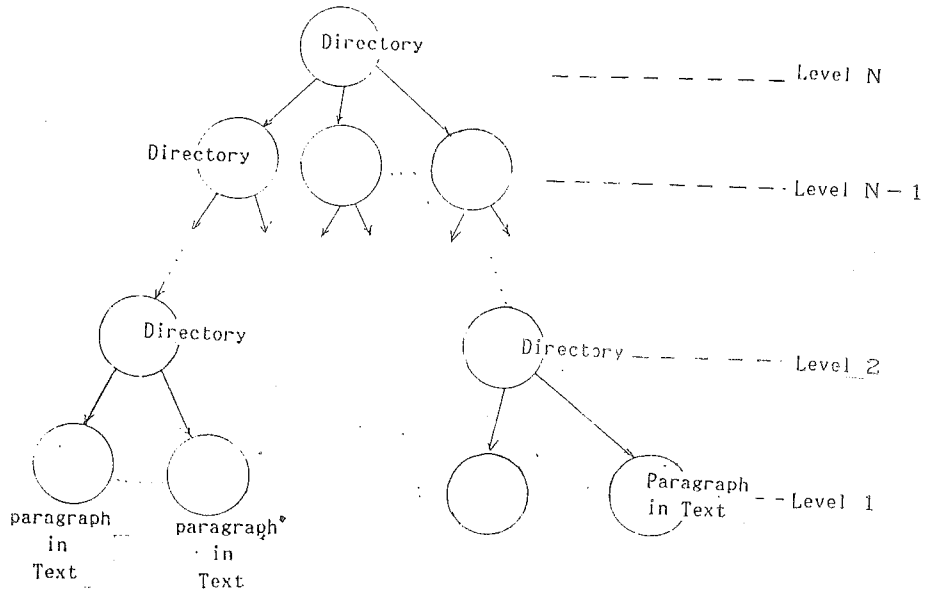
The system provides both the controlled vocabulary search and free-text search of the text database. At present, for controlled vocabulary search, only single key-word search is available. The keyword sets that the system provides include : the name of person(人名、字號等), the name of place, the name of official titiles(職官名), the name of books appeared in the text, the expressions of time in terms of an era, a dynasty or an emperor(朝代, 時代, 帝號, 年號, 名號名等) and 10 classes of special terminologies such as 賦稅類(the class of tax terms), 鹽鐵類(the class of salt and metals terms) etc. A more detailed example can be found in Table-2. [ref. 3]

For each set of keyword, an inverted index file which maps each keyword onto the related paragraphs of the text is generated in the system generation phase. A program that automatically generates these inverted index files will be described in the next section. At present, only a manual-driven man-machine interface is available for users to retrieve text by these system keywords. No logical operators are available at this stage.

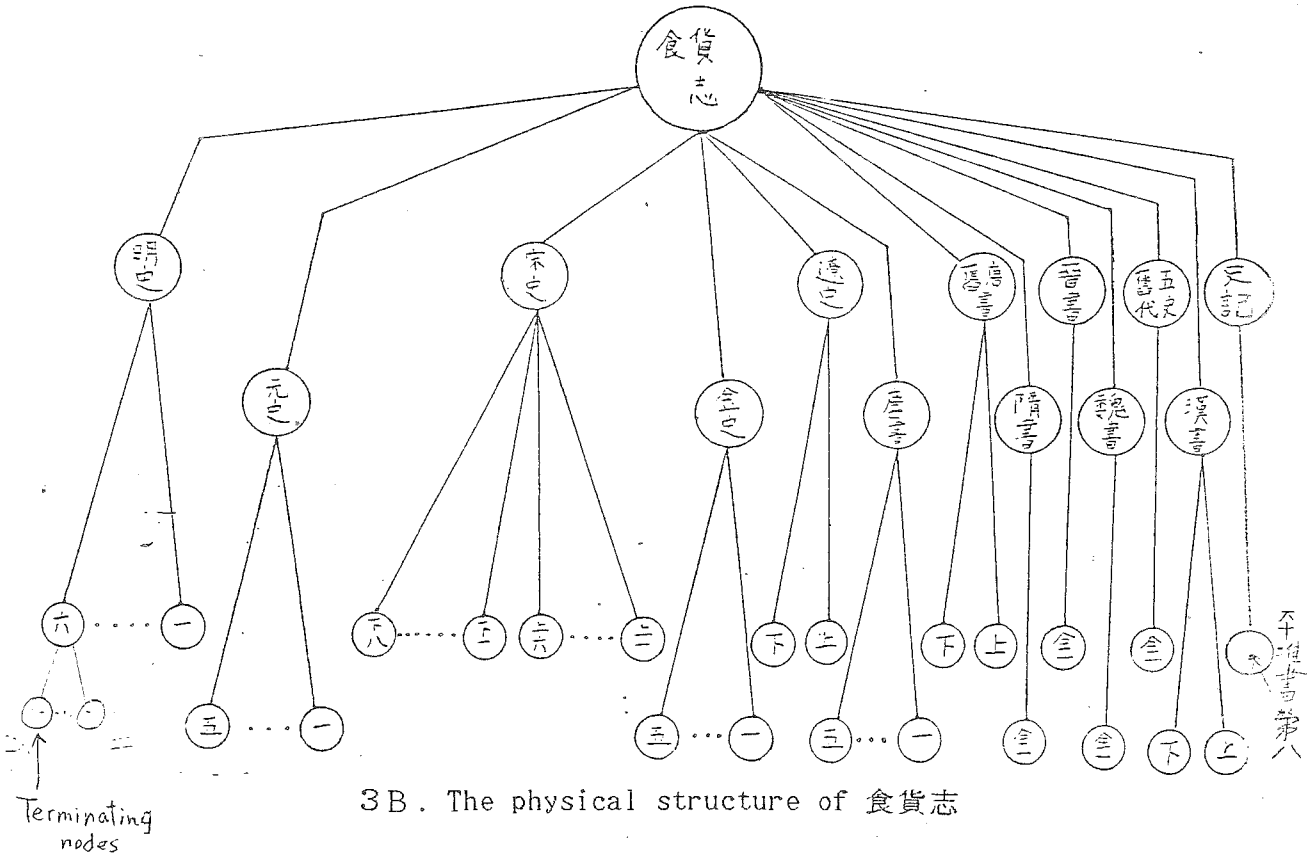
Besides these keywords provided by the system, one important theme of designing this system is to support user defined keyword searching facilities through the application section. In other words, this system supports two categories of keyword ; one is in public domain to be used by all users and provided by the system, and another is merely created and used by a user in private manner.

term / classes	Example
田制類	井田之制　口分　上田　墾田 阡陌之法　世業之田　下田　大田 度田之制　莊田　荒田 世業　荒田　營田
水利類	縵田　占田　湯沐邑　歸田　受田　步 戍田　代田　籍田　公　上田　畮 屯田　井　民田　私　中田　夫 井田　王田　名田　公田　下田　屋
賦稅類	什一　賦斂　三十而稅一　算錢　什五 初稅畮　田租　什一　民租　賦斂 履畮　什五而稅一　田租　租稅　田租 十一之稅　租稅　口賦　田租　口賦
雜稅類	課　秋稅　調　兩稅　上供　折納　租、庸、調 租庸調　財賦　租賦　稅錢　留州　賦役　租 兩稅法　課役　庸調法　稅錢　送使　正賦　地租 田稅　庸　租賦　征役　貢賦
戶口類	六部民　賦役　奴婢　老　樂遷　隣長　力役　金戶 漏戶　戶口　良　小　男女　里長　里黨　沙門 逃戶　男夫　奴　癃　三長　黨長　三長　遷民 雜營戶　婦人　男夫　殘　蔭附　復　屯民
錢財類	比輪　五銖錢　五銖　鑄錢　白鹿皮幣　半兩錢　赤側之錢 四文　大錢　貨泉　錯刀　鑄錢　三銖錢　紫紺錢 沈郎錢　當千錢　鼓鑄　朋　赤側　盜鑄　赤側錢 比輪錢　小錢　內府　元龜　白金　復小　白金三品
鹽鐵類	鹽法　顆鹽　煎鹽　鹽務　煎鹹　糶鹽 鹽麴　私鹽　煎鍊　糴鹽　青白　食鹽 鹹鹺　末鹽　竈戶　食鹽　青鹽　大鹽 兩池　刮鹹　官鹽　鹽法　白鹽　甜次冷鹽
市糴類	六斡　均輸　入粟　入穀　耦犂　買爵 賒貸　常平　甘泉　酎金　輓犂　贖禁錮 輕重　贖禁錮　太倉　均輸　常平倉　輕重 平糴　五均　甘泉倉　平準　拜爵　顧租
漕運類	漕運　河陰倉　轉運　歐艎支江船　漕輓　太原倉　八遞　水陸運 漕米　柏崖倉　武牢倉　上門填闕船　水運　北運　宿場　飛輓 漕吏　集津倉　洛口倉　綯挽舟　車運　陸運　轉輸　水運 含嘉倉　鹽倉　河陽倉　歲運　運漕　運米丁　轉運　通運
非財經	五帝三皇　長城　百保鮮卑　八丁兵　偃武　夏人　華人 西戎　渾　華人　十二丁兵　隆基　洛陽宮　殿最 胡　鸞和　勇士　軍士　玳瑁樓　突厥　東宮 鴻都　六坊　寬鄉　侍官　六官　吐谷渾　修文

Table-2：The 10 classes of special terminologies used as keyword set for text retrieval.



3A : A Logical Structure of the text tree



3B. The physical structure of 食貨志

Figure-3 : Tree-Structured representation of text

The text-tree also provides a way of retrieval. It behaves like a content table of a book. Besides, it also provides proximity functions that will limite the search of text within a reasonably restricted scope.

For the free-text search, users are always been asked to define the scope of the search through the text-tree. The free-terms can be any length of consecutive characters. After a successful free-term search, the user may ask the system to update his/her private index file by adding the newly found entry.

As a final remark of this section, we like to point out that the nodes of text-tree are excellent entry points for inverted index files. Linking the text-tree to the indexing system can provide all information about the organization and associated attributes of the text to the retrieval system. And thus, greatly enhanced the performance of the system. For instance, a field in a node descriptor may be used to describe the length of the text covered by that node. And from this information, we may calculate precisely the time needed for a free-term search in advance. As another example, a broader-term in an thesaurous may be assigned to a node of the text-tree automatically if its auccessor nodes are indexed by most of the terms under that broader-term.

An auto-indexing method for Chinese text

The importance of auto-indexing ability to an information retrieval system is obvious and has been discussed in many papers [ref. 1, 2]. For Chinese text database systems, the auto-indexing ability is even more important and can be considered as a necessity for overcoming the feasibility requirements of implementing such a system. In this research, a fairly simple auto-indexing method has been proposed and, to our suprise, the results are especially encouraging. The method is described as follows.

In Figure-4, a block diagram of the auto-indexing system is presented. The inputs to the auto-indexing system are the text file and the keyword file. The first part of the auto-indexing system is a matching program that scans each characters in the text file and matches it against a tree of words [ref. 4, 5] in the keyword file. An example of a tree of keywords are shown in Figure-5. As you see that all of the keywords that share the same leading characters will be organized in a tree. Therefore in the keyword file, there is a forest of keyword trees.

By this arrangement, the time needed for matching can be reduced. Let n denotes the number of character in text and T denotes the average time required to match a character against a tree of keywords, then, the total time required to build the index file is nT. Of course, the result is application depentent. It depends upon the content of the text and the complexity of the trees of keywords. For the text of ancient document, we found that each tree is rather simple, most of them are trieval, i.e. only one or two words in a tree. Since hashing technigue can be easily applied to located the related tree for a specific character, the value of nT is

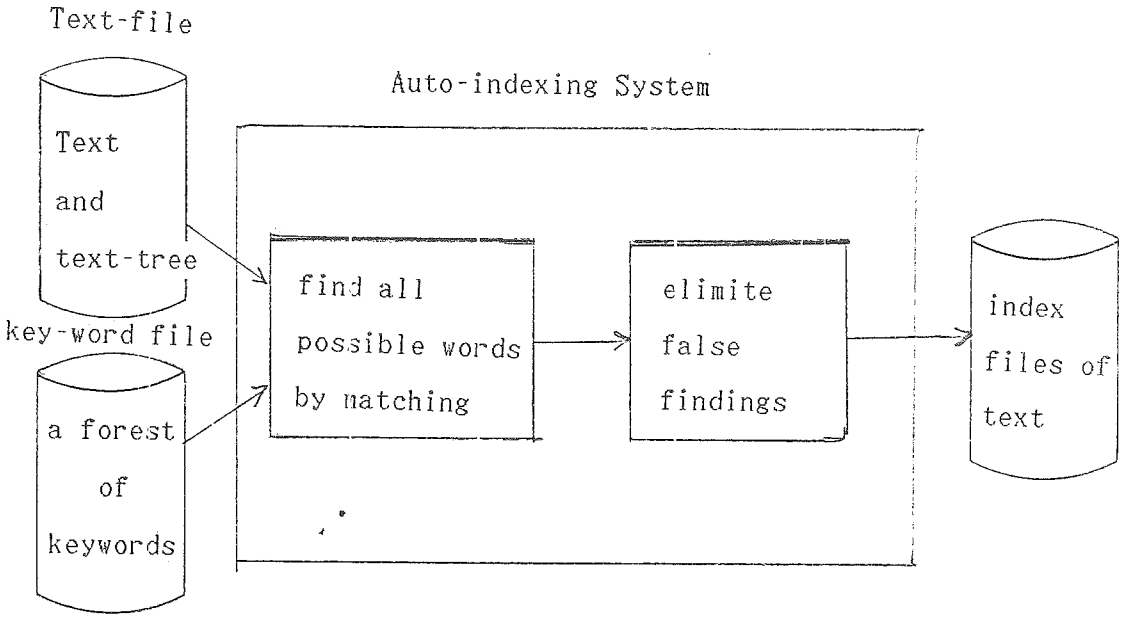


Figure 4 : A block diagram of Auto-indexing method

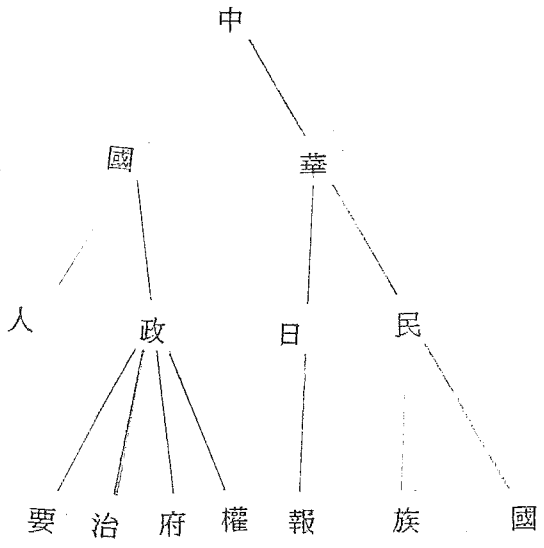


Figure 5 : An example of a tree of words, that share the same leading characters. The node with a circle indicate a termination of a word.

much less than the time required by a straight-forward matching method whose matching time required is n.m.l.t, where l represents the average length of keywords, m the total number of keywords in the forest and t the time required to match two characters.

As for the recall and precision functions of the retrieval method are concerned, an example is given in Table-3. It can be easily proved that according to our indexing method, the index file made by human professionals is a sub-set of the index file found by the machine. According to Table-3, the percentage of the false indexes are very low. It is usually less than 5%.

The second part of the auto-indexing system is a text analysis program that will eliminate some possible false indexes. The text analysis program consists of a rule-based analyzer [ref. 6] which will eliminate some false findings by examing the syntax information obtained during matching. An example is given in Figure-6. Since the number of false indexes are considerably small, this correcting process may be omitted in some applications.

```
#----- 以下為 auto-index 所建立的資料 ----- #
#                                            #
#       621 Mar 11 09:04 his2azzbk.ind       #
#       156 Mar 11 09:04 his2azzbk.iov       #
#      2622 Mar 11 09:04 his2azznm.ind       #
#       416 Mar 11 09:04 his2azznm.iov       #
#      5589 Mar 11 09:04 his2azznu.ind       #
#       104 Mar 11 09:04 his2azznu.iov       #
#      2070 Mar 11 09:04 his2azzof.ind       #
#         0 Mar 11 09:04 his2azzof.iov       #
#      6279 Mar 11 09:04 his2azzpl.ind       #
#       728 Mar 11 09:04 his2azzpl.iov       #
#      1587 Mar 11 09:04 his2azzyr.ind       #
#       572 Mar 11 09:04 his2azzyr.iov       #
--------------------------------------------

Totle memory space : 20744 bytes

--------------------------------------------
#--- 以下為 partial auto-index 所建資料 --#
#                                            #
#       621 Mar 11 09:54 his2mzzbk.ind       #
#       156 Mar 11 09:54 his2mzzbk.iov       #
#      2622 Mar 11 09:54 his2mzznm.ind       #
#       416 Mar 11 09:54 his2mzznm.iov       #
#      5589 Mar 11 09:54 his2mzznu.ind       #
#        52 Mar 11 09:54 his2mzznu.iov       #
#      2070 Mar 11 09:54 his2mzzof.ind       #
#         0 Mar 11 09:54 his2mzzof.iov       #
#      6279 Mar 11 09:54 his2mzzpl.ind       #
#       208 Mar 11 09:54 his2mzzpl.iov       #
#      1587 Mar 11 09:54 his2mzzyr.ind       #
#       624 Mar 11 09:54 his2mzzyr.iov       #
--------------------------------------------

Totle memory space : 20197 bytes

The difference : 547 bytes (about 2.6% )
```

Table-3 : A comparision of Auto-indexing and human corrected partial auto-indexing of all index files in 遼史食貨志.

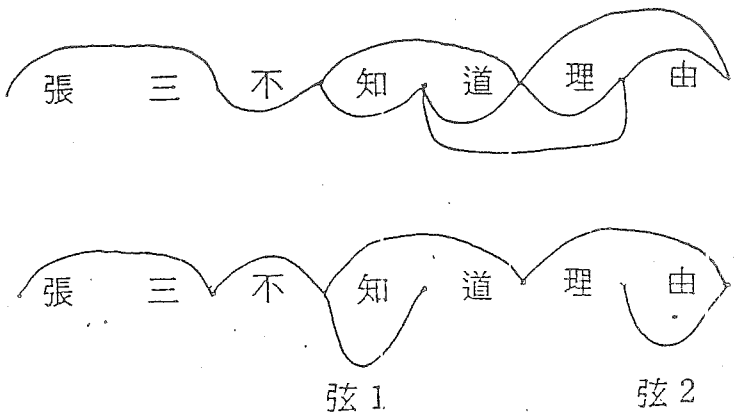


Figure 6 An example of eliminating false indexes by text analysis

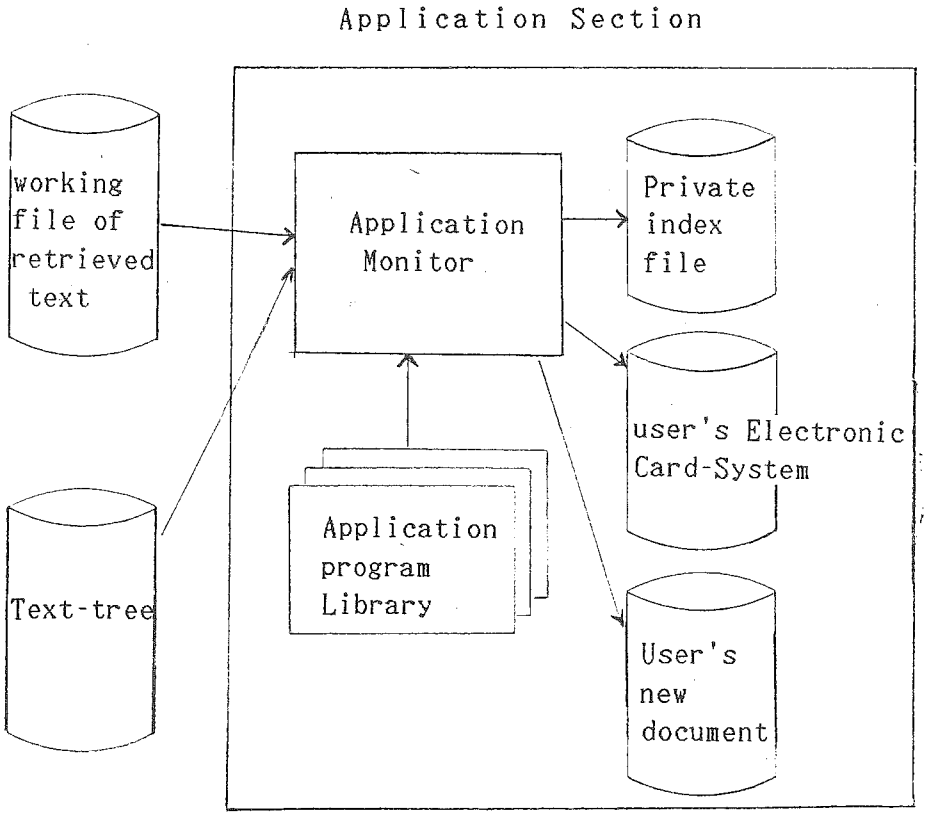


Figure 7 : A block diagram of Application Section and its operational environment.

Application section

In our system, an application section is designed for interfacing the retrieved text to user's applications. At present, the application section provides two major functions ; to establish user's own private indexing system and to generate a card-system after user studied the text. A block diagram that shows the application section and its related system parts is shown in Figure-7.

The application section will be activated by user's request during retrieval. It provide users a manual-driven interface. Through the manual, the user can establish his/her own private index file, and generates a deck of cards on which a memo or a digest of the original text can be written. After a deck of cards has been established, the user can use them to create new document, such as essays, study reports, etc.

Conclusion

For illustration, two examples of search are given in Figure-8 and Figure-9, respectively.

By the experience of project CTP, we like to state the following comments as the conclusion of this paper：

1. There is a high demand of automating tool books usually referenced by humanist. Most of these tools are knowledge-based in nature. And thus, the automation involves the application of knowledge-based systems and AI technology. This is a field that open to us for future development.
2. Natual language processing capability will directly enhence the system performance and improve the friendliness of man-machine interface. Even with very limited natual language processing technigues, such as morphological analysis, do help a lot.
3. The representation of text is still worth doing research.
4. Application section needs further enhencement. Word processor, text editor spread-sheet, note-card system, statistical package, etc, can greatly facilitate the power of text-processing.

Acknowledgement

The author would like to express his sincerely thanks to the Council for Culture Planning and Development, Executive yan (行政院文化建設委員會)for supporting this project.

人名查詢

請鍵入查詢書名:　　後按換行鍵

請鍵入始詢頁數:　　後按換行鍵

請鍵入終詢頁數:　　後按換行鍵

請鍵入查詢人名: 董仲舒　後按換行鍵

請鍵入列印元件 : 0(0:顯示幕,1:印表機,2:不印,9:放棄)後按換行鍵
找到了 !!! 2　筆　08:52:28　08:53:55

1史記 2漢書 3晉書 4魏書 5隋書 6舊唐書 7唐書 8舊五代 9 0
【中 文】【半 形】　字根:_

代號	內容	書名	頁數	段數
1	董仲舒.	漢書	1137	2
2	仲舒	漢書	1137	2

請鍵入始印代號 : 2 後按換行鍵
請鍵入終印代號 : 2 後按換行鍵
請鍵入列印格式 : 1(0:直印,1:橫印)後按換行鍵
請鍵入列印元件 : 0(0:顯示幕,1:印表機)後按換行鍵

1史記 2漢書 3晉書 4魏書 5隋書 6舊唐書 7唐書 8舊五代 9 0
【中 文】【半 形】　字根:_

仲舒　漢書　頁1137　段 2

是後,外事四夷,內興功利,役費並興,而民去本.董仲舒說上曰:「春秋它穀不書,至於麥禾不成則書之,以此見聖人於五穀最重麥與禾也.今關中俗不好種麥,是歲失春秋之所重,而損生民之具也.願陛下幸詔大司農,使關中民益種宿麥,令毋後時.」〔一〕又言:「古者稅民不過什一,其求易共;〔二〕使民不過三日,其力易足.民財內足以養老盡孝,外足以事上共稅,下足以畜妻子極愛,故民說從上.〔三〕至秦則不然,用商鞅之法,改帝王之制,除井田,民得賣買,富者田連仟伯,貧者亡立錐之地.又顓川澤之利,管山林之饒,〔四〕荒淫越制,踰侈以相高;邑有人君之尊,里有公侯之富,小民安得不困?又加月為更卒,已復為正,一歲屯戍,一歲力役,三十倍於古;〔五〕田租口賦,鹽鐵之利,二十倍於古.〔六〕或耕豪之田,見稅什五.〔七〕故貧民常衣牛馬之衣,而食犬彘之食.重以貪暴之吏,刑戮妄加.〔八〕民愁亡聊,亡逃山林,轉為盜賊,赭衣半道,斷獄歲以千萬數.漢興,循而未改.古井田法雖難卒行,宜少近古,〔九〕限民名田,以澹不足,〔一〇〕塞井兼之路.鹽鐵皆歸於民.去奴婢,專殺之威.〔一一〕薄賦斂,省繇役,以寬民力.然後可善治也.」仲舒死後,功費愈甚,天下虛耗,人復相食.〔一二〕

〔一〕師古曰:「宿麥,謂其苗經冬.」

(0)繼續(1)印上一頁(2)印下一頁(9)停止?

1史記 2漢書 3晉書 4魏書 5隋書 6舊唐書 7唐書 8舊五代 9 0
【中 文】【半 形】　字根:_

Figure-8 An example of search by "董仲舒"

官名查詢

請鍵入查詢書名:　後按換行鍵

請鍵入始詢頁數:　後按換行鍵

請鍵入終詢頁數:　後按換行鍵

請鍵入查詢官名: 尚書　後按換行鍵

請鍵入列印元件 : 0(0:顯示幕,1:印表機,2:不印,9:放棄)後按換行鍵
找到了 !!! 33 筆　11:55:59　11:57:31

1史記 2漢書 3晉書 4魏書 5隋書 6舊唐書 7唐書 8舊五代 9 0
【中 文】【半 形】　字根:_

代號	內容	書名	頁數	段數
17	尚書度支郎中	魏書	2859	2
18	尚書僕射	魏書	2860	2
19	尚書令	魏書	2863	4
20	尚書令	隋書	686	2
21	尚書左丞	隋書	676	4
22	尚書左丞	舊唐書	2122	4
23	尚書左丞	唐書	1344	2
24	尚書省	舊唐書	2086	1
25	尚書省	舊唐書	2089	2
26	尚書省	舊唐書	2118	1
27	尚書省	唐書	1369	3
28	尚書省	唐書	1372	1
29	尚書省	唐書	1400	5
30	尚書省	唐書	1402	1
31	尚書右僕射	舊唐書	2118	4
32	尚書比部	唐書	1347	3

繼續?(Y/N)

1史記 2漢書 3晉書 4魏書 5隋書 6舊唐書 7唐書 8舊五代 9 0
【中 文】【半 形】　字根:_

代號	內容	書名	頁數	段數
1	尚書	晉書	782	1
2	尚書	晉書	789	1
3	尚書	晉書	792	3
4	尚書	晉書	793	2
5	尚書	晉書	793	2
6	尚書	晉書	793	2
7	尚書	晉書	798	1
8	尚書	晉書	798	1
9	尚書	魏書	2852	2
10	尚書	魏書	2859	2
11	尚書	魏書	2862	2
12	尚書	魏書	2865	2
13	尚書	隋書	675	2
14	尚書	隋書	682	2
15	尚書	舊唐書	2093	1
16	尚書	唐書	1402	4

繼續?(Y/N)

1史記 2漢書 3晉書 4魏書 5隋書 6舊唐書 7唐書 8舊五代 9 0
【中 文】【半 形】　字根:_

代號	內容	書名	頁數	段數
33	尚書省都事	唐書	1404	1

請鍵入始印代號 : 20 後按換行鍵
請鍵入終印代號 : 20 後按換行鍵
請鍵入列印格式 : 1(0:直印,1:橫印)後按換行鍵
請鍵入列印元件 : 0(0:顯示幕,1:印表機)後按換行鍵

1史記 2漢書 3晉書 4魏書 5隋書 6舊唐書 7唐書 8舊五代 9 0
【中 文】【半 形】　字根:_

Figure-9 An example of search by "尚書"

References

1. Christor Faloutsos, "Access methods for text", Computing Surveys, ACM volum 17, NO.1, March 1985.

2. Carol Tenopir, "Full-text databases", Annual Review of Infornation Science and Techndogy, vol. 19, PP215-246, ASIS, 1984.

3. 毛漢光：史籍自動化第一年總報告，中央研究院，台北市，中華民國，1985.

4. C. C. Hsieh, "Experiments on the search and recognition of word from Chinese text", The first Asia-Pacific Conference on Library Science, Taipei, ROC, March. 1983.

5. 何文雄，中文斷詞的研究，國立台灣工業技術學院 工程技術研究所電子所 碩士論文，July, 1983.

6. 陳正佳，一套中文語法分析系統的研究與設計，國立台灣大學 資訊工程學研究所碩士論文. June, 1985.

7. 林瑞華，卡系與資料整理，中央卡系統推廣中心，鳳山，高雄，台灣，中華民國，1978.

8. 梅棹忠夫原著，余阿勳、劉焜輝澤，知識誕生的奧秘，晨鐘出版公司，台北市，中華民國，1976.