Markup for Chinese Text Processor

Shih Lin
Computing Centre, Academia Sinica

Abstract

Chinese Text Processor(CTP) is a text retrieval package for generic Chinese documents, especially for Chinese classic literature. CTP markup has been designed to extract document structure from various Chinese literature, so that text retrieval may work properly based on the structure. CTP markup separates document structure into two types: hierarchy and detailed. A book is composed of some chapters. A chapter is composed of sections and so on. That's the hierarchy structure. The basic component of hierarchy structure is a paragraph. The structure within it is called detailed structure. Detailed structure comprises headings, lists, tables, groups of sentences(i.e. paragraph in usual) or something else. The search range on CTP is defined according to the hierarchy structure. The markup of detailed structure is important for typesetting. It's also helpful to prevent the false drop caused by finding out a wrong word cross detailed structure.

CTP markup is simple and compact in order to reduce the labour of human work for large documents. Although its function is much simpler than HTML, it may not be replaced with HTML. Just like HTML, CTP markup is a descriptive markup. It can be mapped to HTML quickly.

中文全文檢索系統的標誌

林 晰
中央研究院計算中心

1.前言

中文全文檢索系統 CTP(Chinese Text Processor)是通用的全文檢索工具。它是臺灣最早的全文檢索軟體之一，1984 年七月由謝清俊教授主持開發，並持續改良、使用至今。經由它所製作完成的中文全文資料庫，累計已達一億一千萬字，其中絕大部分爲中國古籍，是目前最大的中文古籍全文資料庫[表一]。至於正在製作的資料庫，估計將近七千五百萬字。除了中央研究院之外，國內外有十九個學術機構安裝了部分的全文資料庫。

表一　已完成的中文古籍全文資料庫

資料庫	字　數	製作單位
漢籍全文資料庫		史語所　漢籍全文資料庫計畫
二十五史	39,969,533	
諸子	5,860,450	
十三經	8,600,316	
古籍十八種	8,049,602	
古籍三十四種	12,264,715	
大正新脩大藏經	5,054,793	
論孟老莊	152,933	
古漢語語料庫	9,141,018	史語所　文獻語料室
臺灣方志	7,537,840	臺史所　史籍自動化室
臺灣檔案	7,100,885	
文心雕龍	1,700,011	資訊所　文獻處理實驗室
佛經三論	104,257	
近現代中國史事日誌	2,085,815	近史所
新清史-本紀	878,629	國史館　清史組
合　計	108,774,958 字	

全文檢索子系統提供層級(hierarchy)目錄，以反應書本的層級結構，亦即類如篇、章、節、段落一般，由上而下、層層劃分的組織架構。使用者依據目錄或原書的頁碼調閱正文(text)，隨即逐段、或逐頁瀏覽正文。正文具備橫、直兩種顯示方向，編排方式仿照原書，且註文、補文、贅文與原文的區別十分顯著，便於閱讀。表格的編排比較繁複，大體上已獲解決。檢索條件由一個或多個字詞組成，任意字詞均能檢索。每個詞的前後可以附加排除字集，特別利於單字詞檢索的精確度(precision)。各詞之間的關係用布林(Boolean)邏輯描述，運算元包括「或」、「且」及「且非」。檢索的範圍依目錄設定，小至單獨的段落、大至整個資料庫，無所不宜。還能限定範圍於特定的文素類別。檢索完成時，列出符合檢索條件的段落，用書名及頁碼表明其出處，並能隨意調閱各段落，或者予以存檔、列印。也可以產生含檢索詞的句子。此外，具備引得(concordance)功能，亦即截取檢索詞前後定長的正文爲一行，各行按檢索詞及其前後文排序。此功能支援語言方面的研究。

全文檢索子系統有兩個版本。文字介面版功能比較繁複，主要爲資料庫建製者以及熟練的使用者所用。WWW 版程式的介面簡明，容易上手，如果有意試用，可由中央研究院的首頁(home page)切入，其 URL 位址爲

http://www.sinica.edu.tw

進入之後，點選(click)「資料庫」，再點選「中文全文檢索系統」即可。許多資料庫僅限於本院內使用，或者在本院內外的開放程度不一，Web 版檢索程式會進行必要的權限管理。

2.文件結構與標誌

大部分中文全文檢索軟體在建製資料庫的過程中，不需要標誌文件，或者只要求最低限度的標誌。它們通常只用來檢索很簡單的文件，像是多篇新聞稿、多個小檔案等。這些文件被視作彼此獨立，而文件的內容簡短，不需要額外的處理。如此簡約的方法，顯然不適合書籍這般內容豐富的文件。

任何著作都有組織結構的脈絡可循，著作的規模越是宏大，往往就越需要良好的組織來支。著作的組織藉著章、節、段落的安排、標題的賦與、相關文字的參照，彰顯其輪廓；讀者則循著這個框架，融入有血有肉的文字中。文件的邏輯組織叫做文件結構(document structure)，大部分典籍的基本結構是層級式的，由類似篇、章、節、段落的層次構成。層級結構的組成單元通稱爲文素(text element)。最小、最基本的文素叫做段落，其內容可能是一段文字、一個表格、一組條列(list)等，或是這些「細部結構」的組合。

就資訊技術而言，去「理解」文字的內涵是極其困難的事，但是經由標誌將文件結構勾勒出來，卻相當容易。文件結構一旦浮現，不僅對讀者的閱讀、理解有益，對程式的查詢、檢索、版面編排同樣緊要。以檢索爲例，CTP 容許將檢索的範圍設爲某部「書」或某幾「卷」等，並找出合乎檢索條件的「段落」。前述的書、卷、段落等，通統是文件結構的文素。

自文字創作應有組織的概念產生之初，標誌可能已經應運而生。讀者閱讀此文時，可以輕易辨認出它的題目、每一節的標題、範圍以及各個段落。因爲透過空白、換行、空行的運用，它們各自呈現不同的版面。這樣的標誌是一種呈現式標誌(representational markup)。如果程式能判讀此一習用長久的標誌，自其抽繹出文件結構，該有多好！可惜各人呈現結構的習慣不同，程式難以兼顧。複雜的呈現形式如表格之類，其變異的程度更高，更難判讀。引用額外的標誌來凸顯文件結構，顯然是必要的。

到目前爲止，大部分的排版軟體或文字處理器(word processor)都採用程序式標誌(procedural markup)。這種標誌直接指揮編排的動作，像是跳行、內縮(indent)、放大等。程式不易藉著這些指令去掌握文件結構。直接描述文件結構的標誌稱爲描述式標誌(descriptive markup)，它標示文件中何處是題目、何處是段落、何處是條列等。它不規定這些題目、段落、條列如何處理，將其留給相關軟體決定。排版軟體拿它來編排，全文檢索軟體則拿它來促進檢索功能。HTML 是一種描述式標誌，由於 Web 的盛行而廣爲人知。一般人多著重於它編排文件、美化文件方面的用途，其實它值得 Web 上的全文檢索軟體多加利用。

3.CTP 標誌

爲了因應文獻典籍的多樣結構，CTP 在發展之初即有自己的標誌，日後並成爲 CTP 的主要特點。設計 CTP 標誌的時候，文件結構的觀念還很陌生，多虧計畫主持人的堅持，一套初步的描述式標誌總算誕生了。不過筆者利用呈現式標誌的念頭還未完全放

棄，因爲它能省略許多標誌工作。1991 年之後 CTP 處理的古籍日漸豐富，版面變化愈來愈，呈現式標誌已經無從掌握，只有擴充 CTP 標誌一途。

CTP 標誌分爲層級標誌及細部標誌兩部分。最初的 CTP 標誌只有層級標誌，它的主要作用正是標示文件的層級結構。所有的正文都包含在層級結構的段落裡,一無例外。過去，段落中的細部結構靠呈現式標誌辨認，所以繕打資料以前，先訂定繕打規範，規定怎樣的細部結構要怎樣留空白、換行等，以利辨認。細部結構愈趨多樣化以後，增設細部標誌來標示細部結構，不再判讀呈現式標誌。然而在某些狀況下，ASCII 的換行碼仍然有顯出細部結構的效果，因而減輕標誌工作。

3.1 層級標誌

層級標誌的符號由 ~ 開始，單獨標於一行，與正文隔離開來；

~b　　文素起始
~l　　段落起始
~e　　文素終止
~p<page>　第<page>頁開始。頁碼可以有雙階，如 3-2。
~f<file>　連至檔案<file>
~d<dir>　被連的檔案位於目錄<dir>
~B　　文章結構文素起始
~E　　文章結構文素終止
~#　　標誌者的說明、註解

每個文素的前後，不論大小，都要有文素起始和終止符號。段落起始處用~b 或~l 均可。資料量大的時候，可以分散到多個檔案，再用~f 將它們串接起來。~f 的作用相當於插入(insert)資料：把~f 所連檔案的內容插至出現~f 之處。假使檔案散布於多個目錄，用~d 來改變目前的目錄。~p 是唯一與層級結構無關的符號，用來記錄原書的頁碼，放在每頁第一行之前。

表二是史記標記主檔的部分內容。史記有一百三十萬字，包含原文、注釋及一些序文。每卷安置於一個檔案，以方便人工的標誌作業，總計有 137 個檔案。標誌主檔串接起全部的檔案，並呈露出整齊的上層結構，自書本至卷深達四層。~b 後面跟隨文素的屬性(attribute)，如 c=book 表示此文素爲書本，它的標題用 n="新校本史記三家注"說明。文素加以分類，稱做文素類別，分類表記載於檔案中，4.2 節會提及。表二第四行的 t=v，指定文素類別爲卷。

表二　史記標誌主檔（部分）

```
~b c=book, n="新校本史記三家注"
   ~b c=pgunit,     n="新校本史記"＆"史記"
    ~b t=part, n="史記"
     ~b  t=v, n="卷一        五帝本紀第一"
        ~f 2_1_1_1
     ~e
     ~# 卷二至卷十一省略
        .
        .
        .
     ~b  t=v, n="卷十二    孝武本紀第十二"
        ~f 2_1_1_12
     ~e
    ~e
    ~b t=part, n="表"
     ~# 除卷十六外, 卷十三至卷二十二省略
        .
        .
        .
     ~b  t=v, n="卷十六    秦楚之際月表第四"
        ~f 2_1_2_4
     ~e
        .
        .
        .
    ~e
    ~b t=part, n="書"
     ~# 卷二十三至卷三十省略
        .
        .
        .
    ~e
    ~b t=part, n="世家"
     ~# 卷三十一至卷六十省略
        .
        .
        .
    ~e
    ~b t=part, n="列傳"
     ~# 卷六十一至卷一百三十省略
        .
```

```
          .
          .
      ~e
    ~e
    ~b c=pgunit,     n="三家注序"＆"史注"
      ~# 內層標誌省略
        .
        .
        .
    ~e
    ~b c=pgunit,     n="點校後記"＆"史校"
       ~f  2_3
    ~e
~e
```

3.2 細部標誌

細部標誌分爲基本單元、表格、圈引、其他四類，見表三。細部標誌由 \ 起頭，直接插入正文所在的行中。既然正文多半是中文，英文的標誌符號雜處其間，不虞有不便判讀的困擾。基本單元標誌通常落在該單元第一行開始處。屬於條列的四種標誌，只要在條列連續各行的第一行標一次即可，各行的換行碼此刻爲有用的標誌。\h 其實也是一種條列標誌。表格標誌比 HTML 簡單好用，足以應付大部分的表格。表格的每一欄類似一個段落，有其細部結構，也適用各種細部標誌。

中文書籍的版面變化相當豐富，自設基本單元標誌既是描述式的，又能存錄版面訊息。此類標誌由 \U 起頭，後面加一個英文字母或數字，其排版方法連同 4.1 節提及的文素分類表載錄於同一檔案中，內容如表四。

表五的範例上承表二。表二中有標誌爲 ~f 2_1_2_4，檔案 2_1_2_4 裡則存有史記卷十六的資料。表五取自該檔案，作爲層級標誌與細部標誌的範例。

表三　細部標誌

(1) Elementary Unit

\s	common elementary unit
\U<x>	user-defined elementary unit in order to record layout
\h	heading
\ul	common list
\ui	list composed of items
\um	list in the middle
\ur	list on the right hand side

(2) Table

\th beginning of table heading
\tb beginning of table without heading
\tr beginning of row
\td beginning of column
\te end of table

(3) Quotation

\qa,\Qa beginning and end of additional text(補文)
\qd,\Qd beginning and end of dummy text(贅文)
\qn,\Qn beginning and end of note(注文、夾注)
\qf,\Qf beginning and end of special segament(特殊部位)

(4) Other

\l including leading spaces as part of text
\\ back slash in text
\~ tilde in the beginning of text line

表四　二十五史的自設基本單元及文素類別表

```
# 自設基本單元：
# 規定\s 及\U<x>的排版方式。每行三欄，首欄是\s 或\U<x>；次欄是此
# 排版單元第一行的起始空白數，末欄是其餘各行的起始空白數。\s 預設的編
# 排方式爲 4/0，不過可以調整。

\s      4       0
\Un     0       6       # 注釋、校勘記(史記)
\U1     0       14      # 校勘記(漢書、後漢--兩位頁數)
\U2     0       16      # 校勘記(漢書、後漢--三位頁數)
\U3     0       18      # 校勘記(漢書、後漢--四位頁數)
\Us     8       4       # 段內之段
\Ul     0       0       # 自段內之段復原
\Ui     6       6       # 段內之段又內縮

# 文素類別：
# 每行二或三欄，首欄是類別代號，次欄是的類別名稱。如果某類文素是段落，
# 可以加上第三欄，預設此類段落第一個基本單元的標誌。如遇例外，在該段首
# 加上正確的標誌即可。第三欄未設者，視同\s。第三欄還可於基本單元標誌(
# 或省略)之後緊接一個圈引起號，其效果和在段首直接設圈引起號相同。常用
# 的起號爲\qn 或\qf。

part    紀志表傳
v       卷
a       標題節
s       引言
```

t　　表格
m　　段落含註
p　　段
h　　卷標頭　　\h
d　　註釋群　　\Un\qn
o　　註釋項　　\Un\qn
g　　校勘記　　\h\qn

表五　史記卷十六秦楚之際月表標誌範例

~b t=h
~p 759
\h 史記卷十六
　　秦楚之際月表第四
\s　　\qn 索隱張晏曰：「時天下未定，參錯變易，不可以年記，故列其月．」
今案：秦楚之際，擾攘僭篡，運數又促，故以月紀事名表也．\Qn
~e
.
.
.
~b t=m
　~b t=p
　秦既稱帝，患兵革不休，以有諸侯也，於是無尺土之封，墮壞名城，銷鋒鏑，[一]鉏豪桀，維萬世[二]之安．然王跡之興，起於閭巷，合從討伐，軼於三代，鄉秦之禁，適足以資賢者[三]爲驅除難耳．故憤發其所爲天下雄，[四]安在無土不王．[五]此乃傳之所謂大聖乎？[六]豈非天哉，豈非天哉！非大聖孰能當此受命而帝者乎？
　~e
　~b t=d
\Un[一]　集解徐廣曰：「一作『鍉』．」　索隱鏑音的．注「鍉」字亦音的．
　　案：秦銷鋒鏑，作金人十二，以弱天下之兵也．
　.
　.
　.
\Un[六]　索隱言高祖起布衣，卒傳之天位，實所謂大聖．
　~e
~e
~b t=t
~p 761
\th 公元前\td 秦\td 楚\td 項\td 趙\td 齊\td 漢\td 燕\td 魏\td 韓
\tr209\td 二世元年\qn 集解徐廣曰：「壬辰。」正義七月，陳涉起陳。八月，武臣起趙。九月，項梁起吳，田儋起齊，沛公初起，韓廣起燕。十二月，魏咎起魏，陳王立之。二年六月，韓成起韓，項梁立之也。\Qn

~p 763
\tr\td 七月\td 楚隱王陳涉起兵入秦。\Ul\qn 索隱二月，葛嬰立襄彊，涉之二月也。至戲，葛嬰殺彊。五月，周文死。六月，陳涉死。然涉起凡六月，當二世元年十二月也。\Qn
\tr\td 八月\td 二
\Ul 葛嬰爲涉徇九江，立襄彊爲楚王。\td\td 武臣始至邯鄲，自立爲趙王，始。
\Ul\qn 索隱凡四月，爲李良所殺，當二世元年八月也。\Qn
.
.
.
\te
~e

4.CTP 標誌與 **HTML**

改良 CTP 標誌時，HTML 已經問世，功能也相當繁複。經過一番考慮，仍然決定加強 CTP 標誌的功用，而非棄之不顧，改採 HTML，原因何在?

(1) HTML 不便表現文件的層級結構。HTML 容納一些層級結構的概念，譬如可以想像標題的標誌 h1、h2、...、h6 有層級的意涵，可是不具強制性，且六層的深度亦嫌不足。
(2) HTML 不利處理大型文件如二十五史者。大型文件的資料散佈在許多檔案中，宜有簡易的連接、組合方式，並便於層級結構的描述。HTML 的超鍵結(hyperlink)係供瀏灠(browse)之需，無助於大型文件的組合。
(3) HTML 依據一般西方文件設計，中文古籍的某些特別結構，如注疏、夾註或補、贅文等不在考慮之列。使用 HTML 僅能排出版面，卻喪失背後的意涵。譬如當代排印古籍，習慣以小字前後加上（與）的型式，代表贅文。CTP 標誌將之取代爲\qd 及\Qd。檢索時依據這對標誌剔除贅文，排版時仍然還其原貌。
(4) HTML 的語法比較複雜，標誌作業較費時，佔用的空間也較多。
(5) HTML 不能保存原書的版面訊息。由於先有書本，再轉製成全文資料庫，如果檢索程式能模仿書本的版面， 許多使用者會感到親切，不過 HTML 的設計宗旨與此毫不相干。

CTP 標誌的設計則原是滿足常見的檢索需求，同時簡單易用。CTP 標誌的整體功能明顯遜於 HTML，因爲我們但求滿足一般中文書籍，特別是古籍的檢索需求，並無意追求其他方面的周延及完備。建製資料庫的成本極其可觀，筆者估算，在臺灣僅是人力成本即達每字 0.35 元以上，一般的情況可能遠過於此。標誌作業如果過於繁重，對建製大量資料庫的機構而言，是一項負擔。我們嘗試使 CTP 標誌簡單易用，希望能控制標誌的成本於最低限。

或許有人擔憂 CTP 標誌在 Web 環境中毫無作用，導至無法藉由 Web 檢索全文資料庫的困境。其實 CTP 標誌與 HTML 都是描述式標誌，且前者比後者單純，很容易由前者即時轉換爲後者，獲致良好的版面。主要的轉換規則列於表六。層級標誌和編排無關，

並且產生資料庫時已轉化成樹(tree)資料結構，不復夾纏於正文檔案中矣！

表六　**CTP** 標誌至 **HTML** 轉換表

標誌性質	CTP 標誌	HTML
段起始	\s,\U<x>	<p>
標題	\h	每行之末加
條列	\ul，\ui	每行之末加
居中	\um	...
偏右	\ur	...
無表頭表格起始	\tb	<table><tr>
有表頭表格起始	\th	<table><th>
表格一筆起始	\tr	<tr>
表格一欄起始	\td	<td>
表格結束	\te	</table>
注文	\qn...\Qn	...
補文	\qa...\Qa	[...]
贅文	\qd...\Qd	(...)
特殊部位	\qf...\Qf	...
空行	<newline>	
頁開始	\p	<hr>

5. 結語

CTP 標誌勾勒出中文典籍的基本結構，並兼攝主要的版面訊息，爲中文典籍檢索奠立可依憑的框架。CTP 標誌考量中文書籍的一般特性，又非常簡單，容易學習使用。它能立即轉換成 HTML，所以無礙於 Web 環境中的操作。

即令 CTP 標誌好用，終究需要一些付出，若要做大規模的標誌，仍應講究技巧，不宜純靠人力硬幹。在繕打資料以前，務必留意怎樣輸入，既不妨礙輸入效率，又能省減標誌作業。假如各種版面的輸入格式十分嚴整，其中保有許多有用的呈現式標誌。很容易撰寫簡單的程式，來辨認特殊的呈現式標誌，進而自動添加 CTP 標誌。就算像表格這麼複雜的東西，也有辦法兼顧直覺快速的輸入，以及程式輔助標誌。二十五史堂堂四千萬字的細部標誌，僅靠一人工作半年，要訣正在於此。大量的出版商引入排版軟體，經過編排的檔案含有各式標誌，不論是描述式的或程序式的，都有轉換成 CTP 標誌的可能，不宜輕忽。

中文古籍的原文與注疏間存在著繁複的關係，CTP 標誌應設法擴充，將彼此的關係勾連得更緊密。最起碼使原文與注疏，以及注與疏之間互連起來，類似於 HTML 的超鍊結，突破層級結構的侷限。表格標誌有所不足，宜向 HTML 借鏡。HTML 和 CTP 標誌混合使用，也值的一試。