

倒置索引파일의 索引語切斷檢索

—TRRS를 中心으로—

한국에너지연구소 대덕공학센터

선임연구원 崔 錫 斗

I. 머리말

현재 利用되고 있는 대부분의 大單位 온·라인문헌데이터베이스에서는 데이터의 量에 거의 관계없이 高速檢索이 가능하다는 장점에서 예의없이 倒置索引파일 檢索方法을 채택하고 있다. 즉, 데이터베이스에서 索引語를 抽出하고 이 索引語가 데이터베이스內의 어느 레코드에 포함되어 있는가를 알 수 있는 索引파일(倒置索引파일 : inverted index file)을 만들고, 아울러 書誌事項索引파일(리니어파일 : linear file)을 만들어 전자 단독 혹은 양자 병용검색에 사용하는 것이다. 이 倒置파일의 檢索이 바로 情報檢索시스템의 主檢索作業이 되며 檢索의 방법중 일반적으로 알려진 것에는 索引의 照合條件에 관한 것으로 索引의 뒤쪽을 절단하는 語幹一致(前方一致), 앞쪽을 절단하는 語尾一致(後方一致), 및 어간·어미를 모두 절단하는 中間一致(scanning mode)가 있으며 이들 部分一致(substring match)와 반대로 절단하지 않는 完全一致가 있다.

한국에너지연구소 대덕공학센터는 28만건의 技術報告書를 데이터베이스化하고 키워드로서 檢索한다는 계획을 착수했을 때 부딪친 심각한 問題가 倒置索引파일檢索時 索引語의 切斷檢索(truncation)이었다. 索引語의 切斷檢索에 대한 方法과 論理를 찾기위하여 노력했으나 전혀 情報을 얻을 수 없었다. 결국 우리 나름대로의 用語切斷檢索方法을 개발하여 당센터의 技術報告書 檢索시스템(TRRS: Technical Reports Retrieval System)에서 시험했으며 현재 사용상 별 문제가 없음이 판명되었다. 本稿에서는 TRRS를 중심으로 語幹一致 및 語尾一致에 대하여 설명하고자 하며 中間一致檢索方法은 개발하지 못하여 省略하고자 한다.

II. 倒置索引파일

1. 倒置索引파일의 필요성

잘 알고 있는 CAS의 CA Condensates는 온라인으로 할 경우 索引을 포함하여 1년분이 約 250~300MB(2억5천만~3억자), 현재의 CA Search데이터베이스는 1년분이 約 1GB(10억자)의 디스크메모리(disc memory)를 필요로 한다. 후자인 경우 아무리 大容量·高速의 컴퓨터라 하더라도 디스크에서 주기억장치로 轉送하는에만 15분은 걸린다. 이것은 轉送時間이며 작업으로서 어떤 데이터를 search하면 그時間은 轉送時間과는 별도로 대단히 늘어나게 된다. 게다가 몇년분이 모이게 되면 데이터량은 간단히 취급할 수 없는 정도가 된다. 이 데이터에서 어떤 정보를 찾기 위하여 컴퓨터부터 끝까지 찾는다는 것은 비경제적이며 불가능에 가깝다.

倒置索引파일이 만들어지면 여러개의 터미널에서 들어오는 다양한 質問들을 동시에 2~3초, 길어도 수십초내에 處理하여 結果를 얻을 수 있으며 리니어파일을 찾아야만 하는 경우 索引語로서 조정한다거나 倒置索引파일을 일단 찾아서 對象을 축소하고 여기에서 리니어서치(linear search)를 하는 등의 방법으로 해결할 수 있다.

倒置索引파일은 1950년대 자기디스크장치의 출현과 함께 탄생한 온·라인文獻檢索시스템에서부터 사용했으며 大量的의 데이터에 관계없이 高速으로 檢索할 수 있으므로 현재에도 여전히 이 古典的인 方法을 쓰고 있는 것이다.

2. 倒置索引파일의 構成

倒置索引파일은 리니어파일에서 登錄番號와 키워드를 뽑아 <그림 1>과 같은 키워드도치색인파일을 만든다.

키 워 드	색 인 번 호	문 헌 수	문헌(3,000개까지)
X(20)	9(5)	9(4)	9(5)

<그림 1> 키워드도치색인파일의 구성

이 索引파일은 <그림 2>의 키워드도치파일작성프로그램중 INPUT-OUTPUT SECTION에서 선언했듯이 레코드 키는 키워드(그림 2의 L-KEY)이며 알터베이트레코드 키(alternate record key)는 索引番號(그림 2의 INX-KEY)가 된다.

```

INPUT-OUTPUT SECTION:
FILE-CONTROL,
  SELECT NAME-FILE ASSIGN TO "KLIBR",
  SELECT S-F ASSIGN TO COB#SORT,
  SELECT INV-FILE ASSIGN TO "INVER"
  ORGANIZATION IS INDEXED
  ACCESS MODE IS DYNAMIC
  RECORD KEY IS L-KEY
  ALTERNATE RECORD KEY IS INX-KEY
  BLOCK CONTAINS 15100 CHARACTERS.
    
```

<그림 2> 도치색인파일작성프로그램의 레코드키

3. 倒置索引파일의 檢索

TRRS는 모두 COBOL로 짜여져있으며 6개의 프로그램으로 構成되어있다. 利用者는 SSearch, FInd, COMbine, Review, Expand, Help, Print (P, P₁, P₂), Type(T, T₁, T₂) 및 END 등 9개의 명령어를 利用하여 필요한 文獻을 신속하게 檢索할 수 있다. 이들 명령어중 SSearch는 完全一致 檢索用이며 FInd는 語幹一致檢索에 사용한다. 語尾一致機能은 TRRS에 현재는 없으나 파일을 축소하여 시험을 완료하고 추가하려는 단계이므로 소개하고자 한다.

가. 語幹一致(前方一致) 檢索

1) 필요성

<그림 3>은 TRRS의 Expand명령으로 索引語 RADIOACTIVE-EFFLUENT를 중심으로 倒置索引파일의 일부를 본 것이다. <그림 3>에서 보는 바와 같이 대부분의 索引語가 複合語로 세분되어 있으며 일치되는 부분을 많이 갖고 있음을 알 수 있다.

FF		
EX	RADIOACTIVE-EFFLUENT	
	RADICALS	8
	RADIO-GALAXIES	1
	RADIOACTIVATION	4
	RADIOACTIVE-AEROSOLS	22
	RADIOACTIVE-CLOUDS	2
	RADIOACTIVE-DISPOSAL	1
	=RADIOACTIVE-EFFLUENT	55
	RADIOACTIVE-ISOTOPEs	1
	RADIOACTIVE-MANAGEMENT	1
	RADIOACTIVE-MATERIAL	77
	RADIOACTIVE-WASTE-DI	491
	RADIOACTIVE-WASTE-FA	241
	RADIOACTIVE-WASTE-IS	51
	RADIOACTIVE-WASTE-MA	1
	RADIOACTIVE-WASTE-FR	411
	RADIOACTIVE-WASTE-ST	280
	RADIOACTIVE-WASTES	355
	RADIOACTIVITY	57
	RADIOACTIVITY-TRANSF	20

<그림 3> Expand한 도치색인파일

예로 어떤 利用者가 RADIOACTIVE-WASTE에 관한 것이면 다 찾고싶다고 할 때 7종의 索引語를 다 쳐넣어야 하며 TRRS가 사용하고 있는 INIS thesaurus(rev. 20, 1981)에서 URANIUM으로 시작하는 索引語그룹은 75種이나 되어 語幹一致의 機能이 없으면 檢索이 불편하거나 不可能하기도 하다.

이때 語幹一致 檢索命令語인 FInd를 사용하면 한줄의 명령으로 전체를 다음과 같이 하나의 集合으로 만들어 준다.

```

??
FI RADIOACTIVE-WASTEX
  01          1838 RADIOACTIVE-WASTEX

```

2) 방법

프로그램상 語幹一致의 수행은 크게 4단계로 나누어진다. 그 1단계는 切斷된 語幹을 전체 倒置파일중에서 찾는 작업이다. 찾았을 경우 그 찾은 索引語를 중심으로 앞쪽을 계속찾아보는 작업이 2단계이며 다시 그 뒤쪽을 찾는 것이 3단계이다. 4단계는 찾은 文獻의 結果를 나타내주는 단계이다.

예로 <그림 3>에 있는 索引語중 RADIOACTIVE-WASTE로 語幹一致檢索을 한다면 FI RADIOACTIVE-WASTE%(%는 절단의 위치표시)라고 명령하면 된다. 진행의 過程을 보면 우선 첫번째 索引番號와 끝번째 索引番號를 합하여 2로 나눈다. 그 숫자를 3000번이라 가정하면 索引番號 3000번에 해당하는 레코드 키인 索引語(키워드)를 옮겨와서 檢索語와 %가 있는 곳까지 한자한자 比較해 간다. 比較하는 중 比較對象索引語가 %가 나오기 이전에 RADIOACTIVE-WASTE보다 큰 것(뒤에 배열되는 것)이면 RADIOACTIVE-WASTE로 시작하는 索引語는 3000번보다 앞쪽에 있으므로 첫번째 索引番號와 3000번을 합하여 다시 2로 나눈다. 마찬가지로 %가 나오기 이전에 RADIOACTIVE-WASTE보다 작은 것(앞쪽에 배열되는 것)이면 RADIOACTIVE-WASTE로 시작되는 索引語는 3000번보다 뒤쪽에 있으므로 3000번과 끝번째 索引番號를 합하여 2로 나눈다. 결과에 따라 같은 방법으로 계속된다. 이 2分探索(binary search)을 반복하면 TRRS에서는 최대 14회의 照合으로 어떤 索引語라도 찾을 수 있다. (INIS thesaurus의 索引語數는 최대 16, 093개이므로 $\bar{n} = \log_2 N$ 즉, 14回).

2分探索의 결과 처음으로 RADIOACTIVE-WASTE까지 일치하고 %가 나오면 RADIOACTIVE-WASTE로 시작되는 여러 索引語중 어느 하나를 찾게된 것이다. 이것의 索引番號를 5800번이라 하면 우선 索引番號 5800번이 갖고 있는 文獻番號를 기억시킨다. 다음에는 5800번의 앞쪽인 5799번의 레코드키인 索引語를 옮겨와서 한자한자 比較하여 %전까지 일치하면 그 文獻番號도 기억시킨다. 이것은 2分探索을 했을 때 여러 同一語幹의 索引語중 처음에 어느 索引語를 만날지 모르기 때문이다. 索引番號를 1씩 계속 감소시키면서 일치되는 索引語의 文獻番號를 기억시키다가 <그림 3>에서 처럼 RADIOACTIVE-M이 나오면 比較를 중단하고 다시 索引番號 5800번에서 1씩 증가시키면서 索引語를 比較하여 일치되는 索引語의 文獻番號를 기억시켜 나간다. 증가시켜 比較하는중 <그림 3>에서 처럼 RADIOACTIVI가 나오면 檢索은 끝나게 되며, 기억되어 있는 전체 文獻番號를 比較하여 동일한 文獻番號가 있으

면 하나만 남기고 除去하여 檢索의 結果를 알려주게 된다. 이 結果가 FI RADIOACTIVE-WASTE의 結果이며 RADIOACTIVE-WASTE-DI에서 RADIOACTIVE-WASTES까지가 된다.

이 語幹一致의 과정은 比較가 많은 部分이기 때문에 COBOL보다는 FOTRAN이 훨씬 효율적이며 프로그램도 쉽다. 그래서 이 부분을 FORTRAN subroutine으로 고쳐 본프로그램에서 Call해서 사용하는 것을 시험중이다. 檢索이 끝난 후 동일한 文獻番號가 있을 때 하나만 남기는 과정도 임시파일을 만들고 그 파일을 다시 읽어 정리(sort)하는 방법을 시험하여 結果에 따라 採擇할 방침이다.

3) 프로그램 및 플로우차트

語幹一致에 해당하는 프로그램의 例와 플로우차트를 보면 <그림 4> 및 <그림 5>와 같다. 語幹一致에 직접적인 관계가 없는 부분은 중간중간 省略했다. 프로그램의 이해를 돕기위하여 몇가지를 설명하면 다음과 같다.

- L-KEY : 키워드(색인어). 레코드 키
- INX-KEY : 색인번호. 알터네이티브키
- W-FIR : 색인번호의 시작
- W-END : 색인번호의 끝
- K3(혹은 W3) : 명령어와 검색어를 함께 넣을 데이터베이스
- A(혹은A3) : 명령어를 넣을 데이터베이스
- SEA-RTN6 : Help
- M : 검색어가 명령어 다음 몇번째 글자부터인가를 알기위함
- COM-ERR : "Syntax error" 메시지

나. 語尾一致(後方一致)

1) 필요성

索引時 다음과 같은 索引語들이 抽出되었다하자.

- LIQUID-METALS
- URANIUM-FLUORIDES
- CARBON-DIOXIDE-LASERS
- RAIN-WATER

이들 索引語가 포함되어 있는 文獻들을 語尾로서 檢索할 수 있게 하기 위해서는 거의다 上位概念語(broader term)인 METALS, FLUORIDES, LASERS, WATER 등을 별도의 索引語로서 추가해야 한다. 語幹의 上位概念語인 LIQUIDS, URANIUM, RAIN 및 ELEMENTS, GAS-LASERS까지도 포함시키게 되면 索引파일이 훨씬 커지게 된다.

(앞부분 생략)

```

FD INV-FILE
  LABEL RECORD IS STANDARD
  RECORD VARYING FROM 25 TO 15029 CHARACTERS
  DATA RECORD IS INV-R.
01 INV-R.
  02 IN-R1.
    03 L-KEY PIC X(20).
    03 INX-KEY PIC 9(5).
  02 IN-R2.
    03 LBO PIC 9999.
    03 LBO-NO PIC 9(5) OCCURS 1 TO 3000
      DEPENDING ON LBO.

```

```

WORKING-STORAGE SECTION.
77 I PIC 9(5) VALUE ZERO.
77 M PIC 9(5) VALUE ZERO.
77 J PIC 9(5) VALUE ZERO.
77 K PIC 9(5) VALUE ZERO.
77 W-COM PIC 9(4) VALUE ZERO.
77 SW1 PIC 9 VALUE ZERO.
77 SW2 PIC 9(5) VALUE ZERO.
77 SW3 PIC 9(5) VALUE ZERO.
77 W-FIR PIC 9(5) VALUE 1.
77 W-END PIC 9(5) VALUE 5977.
77 W-INXB PIC 9(5) VALUE ZERO.
77 W-INXE PIC 9(5) VALUE ZERO.
77 W-INX PIC 9(5) VALUE ZERO.
77 W-INX1 PIC 9(5) VALUE ZERO.
77 W-INX2 PIC 9(5) VALUE ZERO.
77 W-INX3 PIC 9(5) VALUE ZERO.

```

(일부 생략)

```

01 K1 PIC X(20) VALUE SPACE.
01 WBANG1 REDEFINES K1.
  03 WBANG OCCURS 20 PIC X.
01 K2 PIC X(20) VALUE SPACE.
01 W-KEY3 REDEFINES K2.
  03 W-KEY OCCURS 20 PIC X.
01 A PIC X(7) VALUE SPACE.
01 A2 REDEFINES A.
  03 A3 OCCURS 7 PIC X.
01 K3 PIC X(30) VALUE SPACE.
01 W-KEY3 REDEFINES K3.
  03 W3 OCCURS 30 PIC X.
01 WKEY-SAVE.
  02 WKEY-SA OCCURS 10.
    03 WK-NUM PIC 99.
    03 WK-ITM PIC 9999.
    03 WK-DAT PIC X(20).
    03 WK-SAZ OCCURS 9999.
    05 WK-SA PIC 9(5).
01 IM-SAV.
  02 IM-SAV1 OCCURS 4.
    03 IM-NUM PIC 99.
    03 IM-LBO PIC 9999.
    03 IM-LBO2 OCCURS 9999.
    05 IM-LBEND PIC 9(5).

```

(일부 생략)

PROCEDURE DIVISION,
 MAIN SECTION,
 START1.

OPEN INPUT INV-FILE NAME-FILE
 YL-FILE OUTPUT P-F.

START2.

MOVE ZERO TO SEQ WKEY-SA(1)
 WKEY-SA(3),
 MOVE ZERO TO WKEY-SA(4) WKEY-SA(5)
 WKEY-SA(6) WKEY-SA(7),
 MOVE ZERO TO WKEY-SA(8) WKEY-SA(9)
 WKEY-SA(10),

RE-RTN.

DISPLAY "??" UPON DWTY.
 ACCEPT K3.
 IF W3(1) = "H" GO TO SEA-RTN6.
 MOVE SPACE TO A.
 PERFORM W-SEA THRU W-SEA1 VARYING
 I FROM 1 BY 1 UNTIL I > 7.

W-SEA4.

IF A = "FI" MOVE 4 TO M GO TO SEA-RTN2.
 IF A = "FIND" MOVE 6 TO M GO TO SEA-RTN2.

(일부 생략)

DISPLAY " **** COMMAND ERROR **** " K3
 UPON DWTY.

GO TO RE-RTN.

W-SEA.

IF W3(I) = " " GO TO W-SEA4.
 MOVE W3(I) TO A3(I),

W-SEA1.

EXIT.

SEA-RTN.

(생략)

ANS-RTN.

(생략)

W-SEA2.

ADD 1 TO K.
 MOVE W3(I) TO W-KEY(K).

W-SEA3.

EXIT.

IMS-RTN.

(생략)

IMS-RTN2.

(생략)

IMS-RTN4.

(생략)

```

SEA-RTN2.
  MOVE ZERO TO K SW1 SW2 SW3.
  PERFORM W-SEA2 THRU W-SEA3 VARYING
    I FROM M BY 1 UNTIL I > 26.
  PERFORM STA-RTN1 VARYING
    I FROM 1 BY 1 UNTIL I > 20.
  GO TO CON-ERR.
STA-RTN1.
  IF W-KEY(I) = "%" GO TO INX-RTN.
INX-RTN.
  MOVE I TO W-CON.
  MOVE 1 TO W-INXB.
  MOVE 5978 TO W-INXE.
INX-RTN1.
  COMPUTE W-INX = (W-INXB + W-INXE) / 2.
INX-RTN2.
  MOVE W-INX TO INX-KEY.
  READ INV-FILE RECORD KEY IS INX-KEY
    INVALID KEY GO TO INVA-RTN.
  MOVE L-KEY TO K1.
  PERFORM STA-RTN2 THRU STA-RTN3
    VARYING I FROM 1 BY 1 UNTIL I > W-CON.
  GO TO CON-ERR.
STA-RTN2.
  IF W-KEY(I) = "%" MOVE INX-KEY TO W-INX2
    GO TO SAV-RTN.
  IF W-KEY(I) = WBANG(I) GO TO STA-RTN3.
  IF W-KEY(I) < WBANG(I) GO TO INX-RTN3.
  IF W-KEY(I) > WBANG(I) GO TO INX-RTN4.
STA-RTN3.
  EXIT.
INX-RTN3.
  MOVE INX-KEY TO W-INXE.
  PERFORM INX-RTN1.
  IF W-INX = W-FIR GO TO CON-RTN5.
  IF W-INX = W-INXB GO TO CON-RTN4.
  GO TO INX-RTN2.
CON-RTN5.
  (생략)
INX-RTN4.
  MOVE INX-KEY TO W-INXB.
  PERFORM INX-RTN1.
  IF W-INX = W-END GO TO CON-RTN5.
  IF W-INX = W-INXB GO TO CON-RTN4.
  GO TO INX-RTN2.
CON-RTN4.
  (생략)
SAV-RTN.
  PERFORM INSAV1 THRU INSAV2 VARYING
    J FROM 1 BY 1 UNTIL J > LBO.
SAV-RTN1.
  MOVE W-INX2 TO INX-KEY.
  GO TO SAV-RTN22.

```



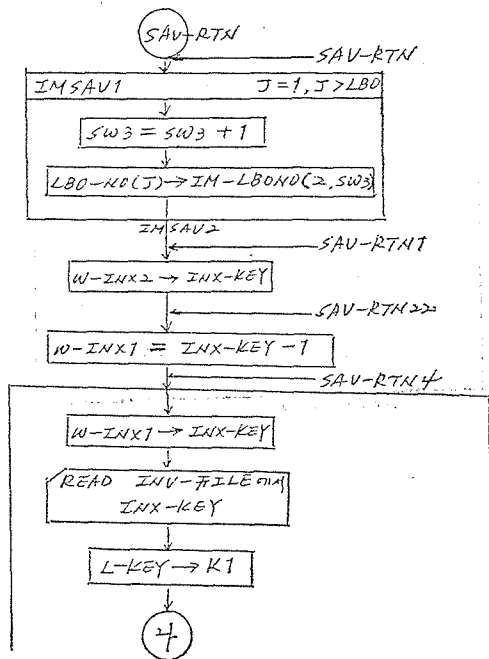
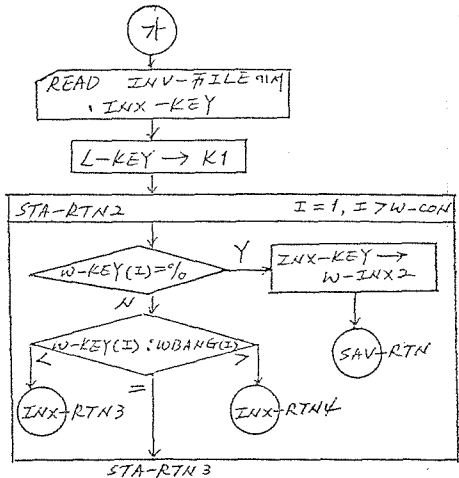
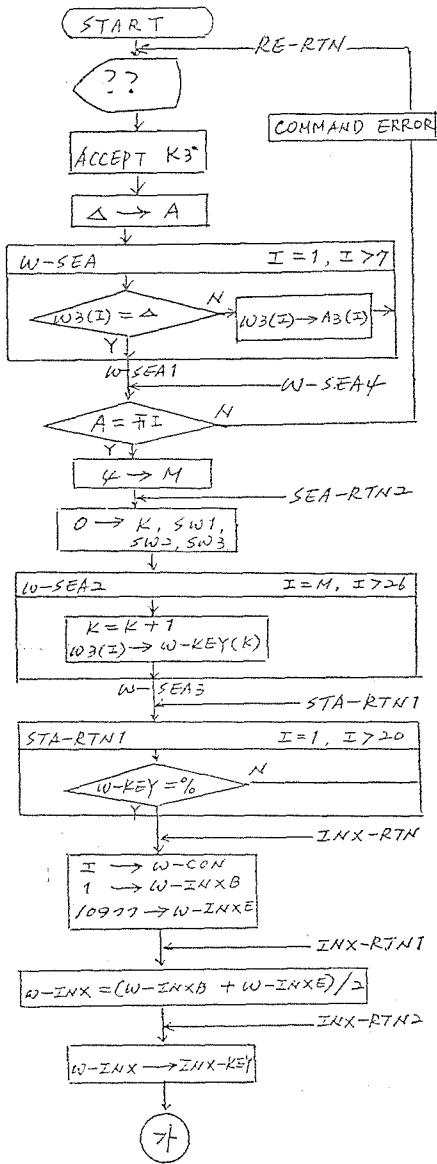
```

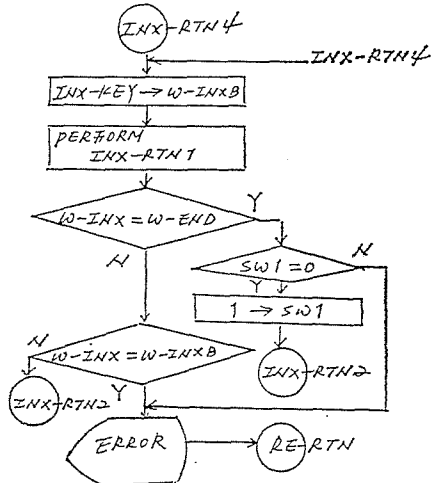
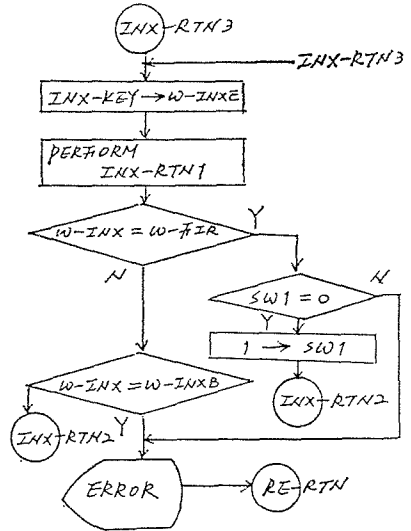
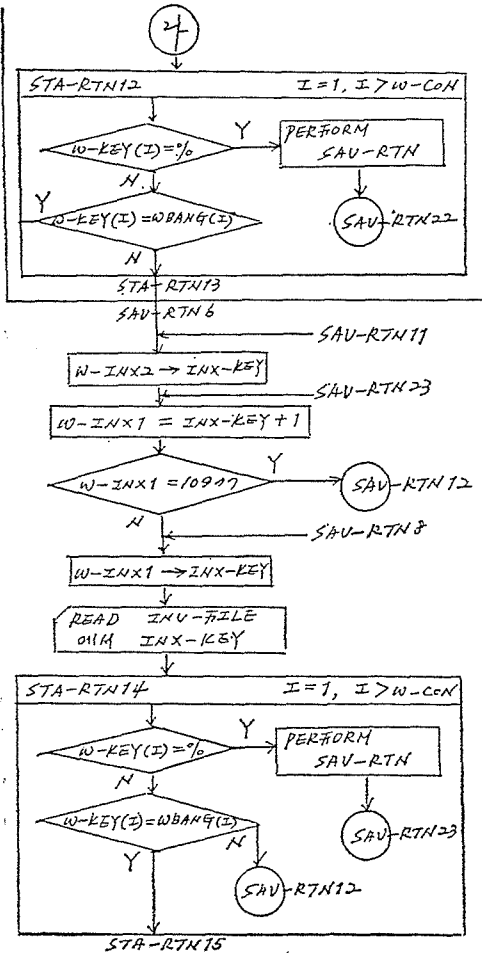
SAV-RTN11,
  MOVE W-INX2 TO INX-KEY,
  GO TO SAV-RTN23.
SAV-RTN12,
  (생략)
SAV-RTN22,
  SUBTRACT 1 FROM INX-KEY GIVING W-INX1.
  PERFORM SAV-RTN4 THRU SAV-RTN6.
  GO TO SAV-RTN11.
SAV-RTN23,
  ADD 1 INX-KEY GIVING W-INX1,
  IF W-INX1 = 5978 GO TO SAV-RTN12.
  PERFORM SAV-RTN8 THRU SAV-RTN9.
  GO TO SAV-RTN12.
SAV-RTN4,
  MOVE W-INX1 TO INX-KEY,
  READ INV-FILE RECORD KEY IS INX-KEY
  INVALID KEY GO TO INVA-RTN.
  MOVE L-KEY TO K1.
  PERFORM STA-RTN12 THRU STA-RTN13
  VARYING I FROM 1 BY 1 UNTIL I > W-COM.
SAV-RTN6,
  EXIT.
STA-RTN12,
  IF W-KEY(I) = "Z" PERFORM SAV-RTN GO TO SAV-RTN22.
  IF W-KEY(I) = WBANG(I) GO TO STA-RTN13.
  GO TO SAV-RTN6.
STA-RTN13,
  EXIT.
SAV-RTN8,
  MOVE W-INX1 TO INX-KEY,
  READ INV-FILE RECORD KEY IS INX-KEY
  INVALID KEY GO TO INVA-RTN.
  MOVE L-KEY TO K1.
  PERFORM STA-RTN14 THRU STA-RTN15
  VARYING I FROM 1 BY 1 UNTIL I > W-COM.
SAV-RTN9,
  EXIT.
STA-RTN14,
  IF W-KEY(I) = "Z" PERFORM SAV-RTN GO TO SAV-RTN23.
  IF W-KEY(I) = WBANG(I) GO TO STA-RTN15.
  GO TO SAV-RTN9.
STA-RTN15,
  EXIT.
INSAV1,
  ADD 1 TO SW3.
  MOVE LBO-NO(J) TO IM-LBONO(2,SW3).
INSAV2,
  EXIT.
CHA-RTN,

```

(이하 생략)

<그림 4> 프로그램의 예





*** SAV-RTN12 이하 생략.

<그림 5> 플로우 차트

실제로 어떤 시스템에서는 이 問題를 감안하여 시소러스파일을 만들고 up-posting處理(索引된 각 디스크립터에 대하여 그 디스크립터에 관련된 모든 上位概念語가 자동적으로 부여되며 또한 檢索된다) 過程을 두고 있으나 이때의 檢索파일은 방대하게 늘어난다. 또한 語尾로서의 METALS, FLUORIDES 등은 單獨索引語인 METALS, FLUORIDES등과 함께 함으로서 索引語로서의 역할이 稀釋되며 語尾로서 複合된 索引語로서는 찾기가 불편하거나 不可能하게 된다.

다음에 LASERS의 例로 보자. LASERS로 끝나는 索引語는 대략

- CHEMICAL-LASERS
- DYD-LASERS
- GAS-LASERS
- CARBON-DIOXIDE-LASERS
- CARBON-MONOXIDE-LASERS
- HELIUM-NEON-LASERS
- SOLID-STATE-LASERS
- NEODYMIUM-LASERS
- RUBY-LASERS
- SEMICONDUCTOR-LASERS

X-RAY-LASERS등이 있다. 語尾一致檢索이 되지않을 경우 이들 索引語를 모두 찾아야 하지만 다음과 같이 語尾一致檢索이 되는 경우에는

```
??
FI %LASERS
    01          49          %LASERS
```

처럼 한번의 檢索命令으로 檢索할 수 있다. 특히 GASERS나 MASERS가 포함되면

```
??
FI %ASERS
    02          52          %ASERS
```

로 檢索할 수 있는 장점이 있다.

2) 방법

다음의 語尾一致方法은 당센터에서 개발한 방법이 아니라 日本東京大學 大型計算機센터에서 개발한 TOOL-IR(University of Tokyo On-line Information Retrieval)의 檢索機能을 응용한 것이다.

우선 倒置索引파일에서 키워드부분을 다음 <그림 6>의 LASERS에 관한 부분과 같이 프로그램으로 글자의 순서를 뒤집어 놓고 다시 알파벳순으로 배열하여 동일한 內容의 倒置索引파일을 만든다.

```

SRESAG                0005900020000200020
SRESAL-ETATS-DILUS   000600007000030000400017
SRESAL-EYD            000610010000210002300025
SRESAL-LACIMEHC      00062000200049000101
SRESAL-MUIMYDOEN     000630011000050001400019
SRESAL-NOEN-MULEH    000640013000260002700033
SRESAL-ROTCUINOCIMES 000650003000030001500088
SRESAL-SAG           000660014000130002700030
SRESAL-YAR-X         000670012000010001400017
SRESAL-YBUR          00068000100016
SRESAM               000690009100075
    
```

<그림 6> 어미일치검색용 파일

실제의 檢索에 있어서 FI %LASERS라고 入力하면 시스템은 語尾檢索으로 받아들여 FI SRESAL%로 語幹一致檢索을 행하게 되며 檢索의 結果는 語尾一致檢索의 結果가 되는 것이다. 이 語尾一致檢索方法은 索引語의 順序를 바꾸어 새로운 키워드 도치색인파일을 만드는 일과 % LASERS를 SRESAL%로 받아들이는 部分프로그램밍過程의에는 語幹一致方法과 같고 비교적 간단하므로 프로그램 및 플로우차트는 省略한다.

다만, 디스크메모리를 줄이기 위하여 索引語의 글자수를 제한하여 잘리는 索引語가 생기게 되면 語尾一致檢索을 위한 파일에서의 위치가 달라져서 그 索引語는 檢索이 되지않는 경우가 생기므로 시스템을 고려할 때 유의해야 할 것이다. TRRS에서는 글자수를 20字로 制限하고 있기 때문에 上記 例의 CARBON-DIOXIDE-LASERS와 CARBON-MONOXIDE-LASERS는 각각 RESAL-EDIXOID-NOBRAC와 ESAL-EDIXONOM-NOBRAC이 되며 R과 E部分에 배열되기 때문에 FI %LASERS로서는 檢索되지 않는다.

Ⅲ. 맺는 말

倒置索引파일에서의 索引語切斷檢索에 대하여 시험한 결과를 설명하여 보았다. 이상의 방법이 최선의 방법이라고 생각하며 진행하여 왔으나 더 좋은 방법이 얼마든지 있을 줄 안다. 더 効果的이고 能率的이며 多樣性이 있는 방법이 개발되기를 기대하며 비슷한 시스템을 고려하는데 도움이 되었으면 한다.

<參 考 文 獻>

1. 韓國科學技術情報센터. "國際原子力情報시스템의 國內活用을 위한 檢索시스템 開發事業(INIS-RS)." 서울: 科學技術處(1979).
2. —. "Dialog 利用方法" 서울: KORSTIC, (1980).
3. —. "Orbit: data base利用法" 第3卷. 서울: KORSTIC, (1981).
4. 東京大學大型計算機センター. "TOOL-IR と その 周邊(TOOL-IRドキュメント集 NO. 12)"東京: 同센터. (1979).

5. —. "TOOL-IR의 應用例(TOOL-IRドキュメント集NO. 10)." 東京: 同센터(1978).
6. 日本情報處理開發協會. "日本語 情報處理システムの研究開發." 東京: JIPDEC, (1977).
7. Bellcomm, Inc. "BCMRET: the Bellcomm information retrieval system" HQN-10426. Barrow-Hall: COSMIC, (n.d.)
8. Calkins, M.L. "On-line literature searching and data base at the USEPA." PB-300 293. (1978).
9. Cooper, M.D. Dewath, N.A. "The cost of On-line bibliographic searching." ED-118 087. ERIC, (1975).
10. Haeuslein, G.K. "On the design of an interactive information retrieval program." CONF-780990-1. (n.d.)
11. Informatics Information Systems Co. "NASA/RECON system; reference manual." HQN-10694. Barrow-Hall: COSMIC, (1974).
12. Kurtz, P. "RADC automatic document classification on-line(RADCOL). program documentation." AD-A011 922. Rome Air Development Center, (1975).
13. Lindberger, D.A.B. and Reichertz, P.L. ed. "The Stockholm county medical information system." Berlin: Springer-Verlag, (1978).
14. Martin, T.H. "A feature analysis of interactive retrieval systems. FR." ED-105 891. ERIC, (1974).
15. Library of Congress. "User's guide to SCORPIO (subject-content oriented retriever for processing information on-line)." ED-108 672. Information Systems Office, (1975).
16. Moore, E.A. "EPA library data processing systems: procedure and user's guide." PB-258 487. (1976).
17. Singletary, V.A. "An on-line conversational retrieval system for orchis text-oriented data base; user's manual." ORNL-4951 (rev. 1). (1975).
18. Teichroew, D. "Information processing systems analysis and design." COO-2544-1. Michigan Univ., (1975).

고객 여러 분의 꾸준한 지도와 편달로 성장한
 폐시는 항상 여러 분의 편익에 도움이
 되고자 봉사하는 자세로
 일하고 있습니다.

- 유네스코 쿠폰 대행기관
- 단행본 및 학술잡지
- 구미서적 수입판매

科學書籍센터

SCIENCE PUBLICATIONS CENTER

서울特別市 鍾路区 鍾路1街21番地 光化門私書函333号

電話 (723) 6719 (725) 0934