

,, Budden brooks “ のデータ化とその索引作成について

樋口, 忠治
九州大学教養部

<https://doi.org/10.15017/6796130>

出版情報 : 言語科学. 16, pp.29-43, 1981-03. 九州大学教養部言語研究会
バージョン :
権利関係 :



„Buddenbrooks“ のデータ化とその索引 作成について

樋 口 忠 治

このたび Thomas Mann 全集 (S. Fischer 版 1974) の第 1 巻 „Buddenbrooks“ (750 頁) のデータ化を終え、これを用いて索引作成プログラムによる索引を作成した。ここにその概要を紹介することにしたい。これは文学研究、言語研究のための二次資料の作成が容易になったという点で大きな意味をもつものと考えられる。

1. データ

ドイツ語の場合、名詞の頭文字を大文字にする習慣があるため、この点を考慮した。また B は SS で表わすこともできるが、区別したい場合を考慮して、この文字は ¥ 記号によって表わした。またウムラウトは一般的に印字が使用できないので = (等記号) によって表わした。しかし、これらの記号は必要に応じて変更することができる。

データは 80 欄カード形式をとった。第 1 欄から第 75 欄までにテキストを入れ、第 76 欄から第 78 欄までに頁数、第 79 欄および第 80 欄に行数を入れた。

この全集本では 1 頁に最大 36 行が印刷されている。部および章の表示部分は省いて本文のみとした。こうした場合、行数は飛んでいる。

データの取り扱い上、各部ごとに分けてデータセットとしている。すなわち全体は 11 部からなっているので、11 のデータセットに分けて格納されている。これらのデータは必要に応じて、他の利用者のために複製することができる。

データはカードの形で読み込むこともできるし、すでに記憶装置に格納されているデータを利用することもできる。カードや磁気テープのような形で保存、運搬することができることはいうまでもない。

データを作成する場合、語と語、あるいは語と記号の間には空白を必ず置かなければならない。例えば an と an, とは別々の語形と見なされる。空白は 1 ケでも 2 ケでも同じである。語の長さは 25 文字までとしているが、実際にデータを作ってみて、これ以上のものはごく僅かであることがわかった。また索引としての性質上、26 文字目以上を省略しても目的を達することができる。

テキストの 1 行は例外的な場合もあるが、だいたい 75 欄までに収まる。分綴されている語はその

まゝでは2語として扱われるから、何れかの行に分綴しない形で収める必要がある。このような場合、まれに75欄までに収まらないことがあるが、ページ数だけの索引をつくるのであれば(行数を無視するのであれば)、行をずらせることによって解決できる。

データ作成に用いたコード一覧

コード

a ~ z	→ A ~ Z		? → ?
β	→ \$ (¥)	[groß → GROS\$]	! → !
ä	→ =A	[Wälder → < W = ALDER]	: → :
ö	→ =O	[öl → < = OL]	; → ;
ü	→ =U	[über → = UBER]	» « → ” ”
大文字記号 (<)	< A	[Arbeit → < ARBEIT]	> < → > >
.	→ .		- → -
,	→ ,		- → +

(25文字以上の単語は除外した。また分綴はせず、何れかの行に入れた。下の例では印刷の都合上、空白がつかまっている。)

DIE <KONSULIN <BUDDENBROOK, NEBEN IHRER <SCHWIEGERMUTTER	910
AUF DEM GERADLINIGEN, WEI¥LACKIERTEN UND MIT EINEM GOLDENEN	911
<L=OWENKOPF VERZIERTEN <SOFA, DESSEN <POLSTER HELLGELB=UBERZOGEN	912
WAREN, WARF EINEN <BLICK AUF IHREN <GATTEN, DER IN EINEM <ARMSSEL	913
BEI IHR SA¥, UND KAM IHRER KLEINEN <TOCHTER ZU <HILFE, DIE	914
DER <GRO¥VATER AM <FENSTER AUF DEN <KNIE HIELT.	915
" <TONY " SAGTE SIE, " ICH GLAUBE, DA¥ MICH <GOTT + "	916
UND DIE KLEINE <ANTONIE, ACHTJ=AHRIG UND ZARTGEBAUT, IN EINEM	917
<KLEIDCHEN AUS GANZ LEICHTER CHANGIERENDER <SEIDE, DEN H=UBSCHEN	918
<BLONDKOPF EIN WENIG VOM <GESICHTE DES <GRO¥VATERS ABGEWANDT,	919
BLICKTE AUS IHREN GRAUBLAUEN <AUGEN ANGESTRENGT NACHDENKEND	920
UND OHNE ETWAS ZU SEHEN INS <ZIMMER HINEIN, WIEDERHOLTE NOCH	921
EINMAL : " WAS IST DAS ", SPRACH DARAUF LANGSAM : " ICH GLAUBE,	922
DA¥ MICH <GOTT ", F=UGTE, W=AHREND IHR <GESICHT SICH AUFKL=ARTE, RASCH	923
HINZU : " + GESCHAFFEN HAT SAMT ALLEN <KREATUREN ", WAR PL=OTZLICH	924
AUF GLATTE <BAHN GERATEN UND SCHNURRTE NUN, GL=UCKSTRAHLEND UND	925
UNAUFHALTSAM, DEN GANZEN <ARTIKEL DAHER, GETREU NACH DEM <KATECHISMUS	926
, WIE ER SOEBEN, <ANNO 1835, UNTER <GENEHMIGUNG EINES	927
HOHEN UND WOHLWEISEN <SENATES, NEU REVIDIERT HERAUSGEGEBEN	928
WAR. WENN MAN IM <GANGE WAR, DACHTE SIE, WAR ES EIN <GEF=UHL,	929
WIE WENN MAN IM <WINTER AUF DEM KLEINEN <HANDSCHLITTEN MIT	930
DEN <BR=UDERN DEN <JERUSALEMSBERG HINUNTERFUHR : ES VERGINGEN	931
EINEM GERADEZU DIE <GEDANKEN DABEI, UND MAN KONNTE NICHT EINHALTEN,	932
WENN MAN AUCH WOLLTE .	933

2. データ作成上の問題点

- 1) 連続する2つの複合語の後半が共通な場合などに行われる一方の後半部分の省略は、テキストの通りにしているが、これは語形のリストとしては不十分であり、やはり、補っておくほうがよいと考えられる。すなわち、前後の関連性がなくなった場合には、省略形は意味がわからなくなるからである。
 - 2) いわゆる複合動詞のzu不定詞のように1語として扱われるものは、他のzu不定詞の場合のように2語として扱うべきものであるから、事後処理を必要とする場合がある。
 - 3) 同形異語の場合の区別はできない。例えば、定冠詞、関係代名詞、指示代名詞のder等、語形が同じものの区別はできない。従ってまた、前置詞、分離前綴りのan等の区別もできない。
- 以上の、データ作成上の諸問題は必要に応じて、端末操作により修正・変更等を加えることができるから、決定的な問題ではないと考えられる。

3. データの作成

テキストを入力データとして作成するためには主として次の方法がある。

1) カード

標準的な80欄カードにパンチする方法である。直接、自分でテキストを見ながら穿孔していくこともできるし、コーディングシートに書き写したものをパンチャーに穿孔させることもできる。

2) TSS

オンラインの端末から入力して、直接データをつくる方法である。大きなテキストを入力データにする場合は、長時間にわたって回線を専有するこの方法は不都合である。

3) OCR

光学読取装置を用いる場合には一定の形式で特別なコーディングシートに書き写したものを読取るわけであるが、文字の書き方に注意しなければならない。

4. データの修正

データの修正はTSS端末の操作によって簡単に行うことができる。修正のみならず、文字や記号の削除、変更等も容易であるから、この方法によって完全なデータを作ることができる。

データだけでなく、処理の結果としての出力リストも端末操作によって修正・変更ができるから、このときに修正することもできる。データの誤りは発見しにくい、出力リストの形で見ると誤りは判り易い。

5. 処理上の制約

計算機の利用に当っては、それぞれの計算機センターにおける利用上の規定に従うことはいうまでもないが、そのほかデータの大きさ、計算時間などに制限があることを認識しておかなければならない。プログラムの性質やデータの量に応じて、どれだけの時間と記憶領域を必要とするのかなどを、予め知っておく必要がある。参考のために若干の資料を次に示しておく。

表 1

九州大学大型計算機センターのジョブクラスと制限値			
ジョブ クラス	領域サイズ	CPU 時間	出力 LP 枚数
A	1024KB 以下	1 分以内	60 枚以内
B	〃	5 分 〃	200 〃
C	〃	20 分 〃	300 〃
F	6144KB 以下	5 分 〃	200 〃
D	(要審査：上記の規模を超えるもの)		

表 2

Buddenbrooks の索引作成に要した時間等				
	データの行数	語形数	CPU 時間	ジョブクラス
1. Teil	1508	3686	130 秒	B
2. Teil	1466	3842	153 秒	B
3. Teil	2750	5210	266 秒	F
	以下略			

言語の処理に際しては語のアルファベット順配列をつくることが多いが(頻度計算、索引作成等)、こうした場合、語形の数はそのまま計算機の制限(記憶領域)に関わることに留意しておかなければならない。もちろん要審査ジョブとして承認をうければこの限りではない。

6. データセット

作成したプログラムやデータはカードの形で保存したり、利用したりするよりも、データセットとして計算機の記憶装置(ディスク)や磁気テープ(MT)に格納しておくのが便利である。これらのデ

ータセットにはそれぞれ識別のための名前をつけておき、計算処理の場合に制御カード1枚でこれを読み出すことができる。

記憶されたデータセットはTSS端末によって容易にコピーをとることができるほか、印刷した形で内容を取り出すこともできる。一般に、基本的なデータは磁気テープに格納し、必要に応じて直接アクセス装置（ディスク）に呼び出す（コピーする）ようにするのがよい。そして、このディスク上のデータを目的に応じて加工し、使用すると便利である。

7. プログラム

1) 頻度計算のためのプログラム

任意のテキスト中に現われるすべての語形をアルファベット順に配列して、その出現頻度を印刷する。データは第1欄から第75欄までにテキストが入っているものとする。

2) 索引作成のためのプログラム

頁数が3ケタ以内の数字で表わされるものに限る。1000頁を越えるものは扱えない。データの数値は第76欄から第78欄に入れる。出力結果はLP用紙を2ツ折りにして製本できるように印刷される。

3) 小文字変換プログラム

ドイツ語特有の書き方、すなわち名詞の頭文字は大文字にするという習慣に合わせて、データ作成に当たって名詞の前に「<」記号を置いているので、この記号の直後の文字のみ大文字とし、他は小文字とするためのプログラム。データそのものをこのプログラムによって大小文字にすることもできるし、索引表や頻度リストに適用してもよい。

プログラムは印刷した形で紹介しても、単なる1文字、1記号の誤りで働かないし、新たに作るよりもむしろコピーした方が安全であるから、ここにはあげない。利用を希望される向きは申し込まれたい。

8. 制御カード

(1) プログラム INDEX・PLI を用いて索引を作成するために必要なカードの配列を次に示す。

1) / / A123401 JOB F1034, TANAKA

この1枚目のカードの内容をジョブ文という。第3欄目のAはジョブのクラスを示す。計算時間が1分以下であること、出力行数が4000まで、領域が標準 256K、最大1024K（この場合、REGION = 1024K と指定する）でなければならない。Bを指定すれば出力行数が12000行ま

でとなり、領域の標準も 512KB となる（最大値は A と同じ）。時間は 5 分まで。1234 は利用者固有の番号。01 はジョブに対して任意につける識別番号であり、16 でも 99 でも 2 ケタの数字であればかまわない。次に空白を 2 ケ置いて 12 欄目から JOB と書く。次に空白を 1 ケ置いて利用者番号 F1234。そのあとコンマを置いて登録したパスワード (TANAKA) を書く (8 文字以内)。

2) // EXEC PLIXCLG

2 枚目のカードは実行文と呼ばれ、PLI のプログラムを翻訳・実行する命令文である。

3) // PLI・SYSIN DD DSN = F1234・INDEX・PLI, DISP = SHR

3 枚目のカードはデータセットに入っているインデックス作成用プログラム INDEX・PLI を呼び出すためのものである。

4) // GO・SYSIN DD DSN = F1234・ABC・DATA, DISP = SHR

4 枚目のカードはデータセットに入っているデータ ABC・DATA を呼び出すためのもの。

5) //

5 枚目のカードは空文と呼ばれ、ジョブの終りを示す。

以上 5 枚のカードを読み込ませることによってデータセットに入っているデータ ABC の索引を LP に出力させることができる。

(2) 頻度リストをつくるためのプログラム HINDO・FORT を使用する場合の制御カード配列

```
// B123401    JOB    F1234, TANAKA
//          EXEC   FORT X CLG, PARM・FORT = 'OPT (0)'
// FORT・SYSIN DD    DSN = F1234・HINDO・FORT, DISP = SHR
// GO・SYSIN  DD    *
20000
//          DD    DSN = F1234・MANN01・DATA, DISP = SHR
//          DD    *
'END
// GO・FT08F001 DD    UNIT = WORK, SPACE = (TRK, (50, 50)),
//          DCB = (RECFM = FB, LRECL = 80, BLKSIZE = 1600)
// GO・FT09F001 DD    UNIT = WORK, SPACE = (TRK, (50, 50)),
//          DCB = (RECFM = FB, LRECL = 80, BLKSIZE = 1600)
//
```

上にはデータがデータセット (MANN01・DATA) に入っている場合の例を示した。データをカード形式で入力する場合には、⑥行と⑦行を省いて、この部分にデータ・カードを入れる。

⑤行目の 20000 は処理単語数の制限値。小さなデータならば 10000 あるいは 5000 でもよい。

⑧行目の END はデータの終りを示すダミー・カード。⑤と⑧を予めデータの前後に入れておけば、④と⑥は 1 枚のカードにしておいてもよい。

<W=ANDEN	3	W=AHREND	24
<W=ORTER	1	W=ARE	6
<W=UNSCHE	1	W=AREN	2
<WACHSEN	1	W=UNSCHE	1
<WACHSGERUCH	1	W=UNSCHT	1
<WAGEN	1	WAR	99
<WAHNSINN	1	WARD	6
<WAHRES	1	WAREN	27
<WAISE	1	WARF	4
<WAND	4	WARMER	1
<WANGE	1	WARTE	2
<WANGEN	1	WARTET	2
<WANGENKNOCHEN	1	WARTETE	1
<WASCHHAUS	1	WARUM	4
<WASSERFLUT	1	WAS	43
<WASSERS	1	WASCHBAREN	1
<WECHSEL	1	WASCHEN	1
<WEG	2	WAT	1
<WEGES	1	WEDER	1
<WEGLOSSUNG	1	WEI¥	13
<WEGER=AUMEN	1	WEI¥E	6
<WEI¥WEIN-BOUTEILLE	1	WEI¥EN	7
<WEIB	1	WEI¥LACKIERTEN	3
<WEINE	1	WEI¥T	1
<WEINH=ANDLER	3		
<WEISE	5		
<WEISHEIT	1		

(3) プログラム KOMOJI・FORT の場合

計算結果としての頻度リストや索引の各見出し語を普通の形式、すなわち名詞ならば頭文字だけを大文字にして、他はすべて小文字に変換して出力するためのプログラムである。もちろん、データそのものをこのプログラムによって処理すれば、名詞の頭文字のみを大文字にして、他はすべて小文字にしたリストを出力する。例えば、出力結果をデータセット XXX・OUTLIST に入れ、それをデータとしてこのプログラムによって処理する場合の制御カードは次のようになる。

```
// B003601 JOB F0036, TANAKA, REGION = 1024K
// EXEC FORTCGO, PARM = 'NOSOURCE'
// FORT. SYSIN DD DSN = F0036 · KOMOJI · FORT, DISP = SHR
// FT08F001 DD DSN = F0036 · xxx · OUTLIST, DISP = SHR
//
```

この5枚のカードをカード読取装置から入力すればよいし、あるいは TSS 端末から FIB ジョブとして処理依頼してもよい。

9. データの管理

作成したカード形式のデータは直接アクセス装置（ディスク）に記憶させておくと、そのまま処理に使用できるので便利であるが、直ちに使用しない場合はなるべく磁気テープに移しておいて、必要ときにディスクに移すのがよい。ディスクにあまり多くのデータを入れたままにしておくのは、ディスクの効率的な利用という点からも避けるべきである。

計算結果も一応ディスクに入れ、適当な編集を行った上で、これをテープに移しておけば、同じ計算を繰り返す必要がないことになる。次にデータの移し方について例を示す。

1) カード形式のデータを直接アクセス装置（ディスク）へ

```
// B123401 JOB F1234, TANAKA
// EXEC PSCOPY
// SYSUT2 DD DSN = F1234 · ABC · DATA,
// DISP = (NEW, CATLG), UNIT = PUB,
// SPACE = (TRK, (10, 10), RLSE),
// DCB = (LRECL = 80, RECFM = FB,
// BLKSIZE = 1200)
// SYSUT1 DD *
```

カード

/ *

2) ディスクのデータセットを磁気テープへ

```
// L123402 JOB F1234, TANAKA
// EXEC PSCOPY
// SYSUT1 DD DSN = F1234 · ABC · DATA, DISP = SHR
// SYSUT2 DD DSN = TEKITO, DISP = (NEW, KEEP),
```

```
//          LABEL = 1, SL), VOL = SER = 通し番号,
//
//          UNIT = CLSMT,
//
//          DCB =(LRECL = 80, RECFM = FB, BLKSIZE = 1200)
3) 磁気テープ (MT) のデータセットを → 直接アクセス装置 (ディスク) へ
// L 0000 01  JOB   F 0000 , パスワード
//
//      EXEC  PSCOPY
//
// SYSUT1  DD   DSN = データセット名 , DISP = OLD,
//
//          LABEL = ( 1 , SL,, IN), VOL = SER = 通し番号 ,
//
//          UNIT = CLSMT
//
// SYSUT2  DD   DSN = F 0000 . データセット名 . DATA,
//
//          DISP = (NEW, CATLG), UNIT = PUB,
//
//          SPACE = (TRK, (10, 10)),
//
//          DCB =(LRECL = 80 ,RECFM = FB, BLKSIZE = 1200 )
//
```

- 1) 新しいデータセット名 ABC をつけてディスクにデータを格納する。DCB (データ・コントロール・ブロック) の論理レコード長 LRECL = 80 は120にしておくくと TSS 端末で行番号がつけられて、取り扱いが便利になる。
- 2) 磁気テープの使用は利用者自身が取り扱うオープン使用と、センターで保管し管理するクローズド使用がある。後者の場合、センターにテープを持ち込み、通し番号を受け取った上で、これを使用することになる。ラベル番号1はデータセットの格納順番である。
- 3) テープからディスクへコピーする場合には新しい名前をつけてやる必要がある。もしすでに同一名のデータセットがディスクにあるとコピーされないの、予め確認しておかなければならない。コピーが正常に行われたかどうかは、完了コードが=0000になることで確認される。

10. TSS 端末

カード形式で計算機に読み込ませるほかに、TSS 端末を用いて直接プログラムやデータを取り扱うことができる。端末を利用するためには、その利用の仕方を予め十分理解しておく必要があるが、FIB ジョブのためだけであれば若干の知識で利用することができる。

計算結果を直ちに印刷された形、あるいはスクリーン上で必要とする場合を除けば、一般的には出力結果を再びデータセットとして記憶装置に入れ、必要に応じてこれを取り出すようにすれば、同じ

計算を反復する無駄が省かれる。この場合、FIB ジョブならば、OUTPUT コマンドで PRINT (データセット名) とすれば新しいデータセットが出来る。このデータセットから不必要なメッセージ等を削除して、必要な内容だけに編集しなおしておけばよい。

READY

OUTPUT ジョブ名 PRINT (XYZ) Ⓜ

TSS からジョブのキャンセルをする場合には次の方法がある。

1) 計算の途中で

READY

CANCEL ジョブ名 Ⓜ

2) 出力待ちの段階で

READY

OUTPUT ジョブ名 DEL Ⓜ

不要なデータセットの消法

READY

DEL データセット名 Ⓜ

11. データの編集

いったんデータセットに入れられたデータ(テキスト)は TSS 端末の操作によって自由に編集しなおすることができる。データの修正、変更、削除、追加、複製などはきわめて容易である。次に、その機能のうち最も基本的でかつ必要なものを若干示しておく。READY モードで用いるコマンドと、EDIT モードで用いるコマンドに分けて示す(Ⓜはリターン・キーあるいは復改キー)。

A) READY モードのコマンド

1) データセットを編集する場合の呼び出し。

READY

EDIT ABC DATA Ⓜ

2) データセットのコピーを作る。(ABC のコピー、CBA をつくる)。

READY

COPY ABC CBA Ⓜ

データセット・タイプ、論理長等を変更する場合もこのコマンドを用いる。

READY

COPY ABCD · OUTLIST DCBA · DATA B(1200) L(80) REC(FB) Ⓜ

B) EDIT モードのコマンド

- 1) 文字 (列) の検索 (例えば、AUFを見つける)。この場合、**VERIFY** コマンドを用いると、その文字 (列) を含む最初の行が表示される。

```

EDIT
FIND / AUF / (R)
EDIT
V (R)
EDIT
FIND / AUF / (R)

```

ひき続いて同じ文字 (列) を探す場合には **FIND** のみ入力すればよい。このようにして簡単な検索は行うことができる。

- 2) 文字列 **ABC** を含むすべての行を端末に出力する場合。

```

EDIT
LIST /ABC/ (R)

```

- 3) 文字 (列) の変更。A を B に変更する。

```

EDIT
V (R)
EDIT
C /A/B/ (R)

```

行番号 10 から 50000 (5000 行) の = A をすべて **AE** に変更する。

```

EDIT
C 10 50000 /=A/AE/ALL (R)

```

行番号 10 から 20000 までの **DER** をすべて削除する (空白にする)。

```

EDIT
C 10 20000 / DER / /ALL (R)

```

- 4) 行の削除。1 行目から 10 行目までをすべて削除する。

```

EDIT
TOP (R)
EDIT
DELETE * 10 (R)
TOP OF DATA SET

```

- 5) 文字 (列) の追加、行番号 100 の **LESE** を **LESEN** に変更する。

```

EDIT

```

C 100 / LESE / LESEN / (R)

5000行以内のデータ全部について同様の変更をする場合 (GEHT → GEHEN).

EDIT

C 10 50000 / GEHT / GEHEN / ALL (R)

その他、データセットの内容を見る場合。

EDIT

L (R)

行番号 100 から 200 までの内容を見る場合。

EDIT

L 100 200 (R)

12. FIB ジョブ

データおよびプログラムがすでにデータセットに入っている場合には、計算は数枚の制御カードのみによって行うことができる。しかし、センターに行くまでもなく、TSS 端末によってこれらの制御カードを読み込ませ、計算を行うことができる。このジョブを **FIB** ジョブと呼んでいる。次に **FIB** ジョブの一例を示す。

READY

EDIT ABCD NEW CNTL (R) (1)

INPUT

00010 // A003601 JOB F0036, TANAKA (R) (2)

00020 // EXEC PLI X CLG (R) (3)

00030 // PLI · SYSIN DD DSN = F0036 · INDEX · PLI, DISP = SHR (R) (4)

00040 // GO · SYSIN DD DSN = F0036 · ABC · DATA, DISP = SHR (R) (5)

00050 // (R) (6)

00060 (R)

EDIT

SUBMIT (R) (7)

A003601 SUBMITTED

A003601 JOB ACCEPTED (8)

READY

STATUS F0036 (R) (9)

JOB STATUS JOBNAME (10)

ACCEPTED (11)

SYSOUT A003601 (12)

READY

OUTPUT A003601 PRINT (XYZ) (R) (13)

13. 索引の利用

文法上のさまざまな問題に対して具体的な用例をあげる場合には、索引が大いに役立つ。例えば「werden + 不定詞」の形式^(注)について用例を調べたい場合には、まず werden の用例を調べ、当該の形式に用いられたものかどうかを見ればよい。具体的な例を示すと、Th. Mann の „Buddenbrooks“ 第 1 部には werden の用例が次の通り出てくる。

ward	6	worden	5
werde	7	wurde	8
werden	8	wurden	2
wird	13	würde	9
wirst	1	würden	2
		geworden	10

合計 71 回

このうち「werden + 不定詞」の形式に用いられる可能性があるのは 29 例であるから、これについてのみ調べればよいことになる。索引にはこれらの語形が現われるページ数が示されているから、テキストを見て当該用法であるかどうかを確認すればよい。

このような研究の補助手段として、テキストのすべての語形の索引リストが必要である。このようなリストを数多くのテキストについて作成することによって、言語の研究はきわめて能率的になるであろう。

単に能率的になるばかりではない。用例が組織化されているということは重要なことであり、ある問題の証明に用いられる用例が出所不明であるとか、好都合なものだけを集めたというのでは信頼性が損なわれる。こうした意味においても、言語の研究にはテキストの索引や頻度リストが是非とも必要であるといわなければならない。

今日の段階ではテキストのデータ化はまだ多くの人手を要するが、しかしこの点は将来改善される余地があるし、いったんデータ化されたテキストは多数の研究者によって共同利用ができ、複製することがきわめて簡単にできるから、その利用効果は相対的にみて、決して小さくないということができよう。

