文字列の配列順序についての問題*

西　村　恕　彦**

Abstract

A large list of sorted words and phrases is a basic material for research in linguistic information processing.

A computer arrangement of phrases differs partially from conventional arrangement and is found insufficient for human use when the phrases are spelled with hyphen, period, space, and other special characters as a computer locates a string in the list according to the collating sequence of characters.

On the contrary, a conventional arrangement of words in dictionaries is based on key strings composed of 26 letters excluding special characters. A large list for human examination will be sorted by a computer utilizing such key strings derived from the original spellings.

1. 問　題

計算機で処理されるデータの値の範囲は，数値データにとどまらず，文字データにもおよんでいる．数値データのとるふるまいについては，その問題の所在や対策などが，わりあいによく知られているようであるが，文字データに関しては，これまでにほとんど述べられたことがない．

文字データの取り扱いについて起こるトラブルの一つをここにあげ，その対策を示す．

それは，人間が見るための文書（索引や辞書）を作るのに，計算機の助けを借りて文字列（語や句）のソーティングを行なったときに，得られた出力の配列順序が，慣用の配列順序と食い違うことである．

このことはおそらく実務にたずさわっている人たちには気付かれているに相違ないが，問題として取り上げられたことはない．しかし，今後，計算機が情報検索，自然言語処理，自動編集などにどんどん使われ，大規模な言語情報が計算機と人間との交渉領域に発生するにつれ，トラブルとして認識されるようになるだろう．

ただしこの問題は，人間が見る，大きな辞書や索引を計算機でソーティングするときだけ意味があるのであって，辞書や索引の規模が小さいときには，問題にならない．また，それらを計算機の中だけで保持し，検索するときにも，問題とはならない．

2. 文字の大小順序

計算機の文字の組の中では，文字の大小順序が定義されていることが多い．これに関連して，各文字（や文字列）のあいだには，つぎのような性質が備わっていないといけない．

（1） 字形と符号とのあいだに，一対一の対応がある．この性質をかりに識別性と呼んでおくが，識別性の完全ではない符号系も実際には使われることがある．たとえばつぎの例のように，同じ字形を二つの符号のどちらででも表現できるとか，二つ以上の字形が同じ符号を共有するとかいうことがある．

印字	符号
+	12—6—8
+	12
%	0—4—8
(	0—4—8

（2） またプログラムの立場からも，二つの異なる文字（または文字列）を必ず異なると判定し，二つの同じ文字（または文字列）を必ず等しいと判定したいわけだが，処理系によってはこのことを保証できない

* A Trouble with Computer Arrangements of Natural Words, by Hirohiko Nisimura (Electrotechnical Laboratory, MITI, Tokyo)

** 通商産業省電気試験所

場合がある．

たとえば FORTRAN では，JIS 規格に厳密に合致していても，つぎの例のように，いやなことの起こる可能性がある．

例：違う文字列が等しいとされる．

```
X=3H00A          (=0.0, 正規化)
Y=3H00B          (=0.0, 正規化)
IF (X.EQ.Y)      (.TRUE.)
```

例：同じ文字が違う値とされる．

```
DATA U/1HC/      (=6HC00000)
READ ( ) V       (=6HC△△△△△)
FORMAT (80A1)
IF (U.EQ.V)      (.FALSE.)
```

（3） 各文字のあいだに大小順序が決まっていて，文字列の大小順序もまた，これにしたがって決まらなければならない．処理系によっては，つぎの例のように，文字列の大小順序が，個々の文字の大小順序からは決められていないことがある．

例：FORTRAN では文字列の大小順序は，文字の大小順序とは別になる．

```
Q=3HAAA          (=212121……)
R=3HABCとすると  (=212223……)
Q<R      となる．

S=3HJAA          (=−12121……)
T=3HJBC とすると (=−12223……)
S>T      となる．
```

（4） 各文字のあいだの大小順序が，慣用の配列順序に近いことも望ましい．たとえば，

```
空白<a<b<c<……<z
空白<ア<イ<ウ<……<ン
  0<1<2<3<……<9.
```

この条件が満たされていないときには，いわゆるコード変換の手続きによって，簡単に変更してやることができる．

以上に述べたことは基本的な要求であって，比較的容易に解決できるし，処理系によってはすでに固定的に解決ずみであろう．しかし以下の事項は，これらの要求が満足されていても，なお問題として残り，コード変換とは別の技巧が必要になる．

以下では上記の要求は満足されているものとみなすことにする．

3. 英字列の配列

（1） 語間の空白

見出し（キー）として採用される文字列が1つの語だけであって，その途中に空白を含まない場合には，計算機のソーティングは，慣用の配列順序と一致する．しかし見出しの文字列が2つ以上の語（句，熟語，複合語）であって，途中に空白が含まれる場合には，単純にそれをキーとしてソーティングを行なっては，慣用の順序とは合わなくなる．

それは，計算機では空白もまた固有の文字とみなして，対応する文字位置にある他の英字と比較するからである．ところが人間の場合には，語のあいだの空白は，あたかも空（null）であるかのように，非常に弱く評価され，それに後続するつぎの語頭の文字列のほうが，むしろ配列順序の決定に利用される．

たとえばつぎのようと対比できる．

計算機	慣　用
storm	storm
storm △ belt	storm belt
storm △ zone	stormbird
stormbird	stormwind
stormwind	stormy
stormy	storm zone

慣用におけるこのような配列順序は，複合語の区切り方に“ゆれ”があるのを救済する効果がある．たとえば，つぎのような3通りのつづり方がある．

look up	to night
look-up	to-night
lookup	tonight

これらは，計算機式の配列では（見出し語の語彙が大きいときには），まったく遠いところに分散されてしまうのに反し，慣用の配列では隣接し，検索が容易になる．

しかし慣用の配列そのものにも，たとえば辞書によって方式の違いが見られ，ときには一つの辞書の中でも統一されていないことがある．こういう現象はおそらく人間にとっての使いやすさの観点で説明されるのであろう．

たとえば，三省堂の新英文法辞典ではつぎのよう

に配列されていて，これは同じ中心語にたいするいろいろな熟語を一か所に集めるために，あえて普通の慣用の配列からはなれたのであろう．

新英文法辞典	慣　用
as	as
as for	as for
as long as	aside
as than	as long as
aside	aspect
aspect	as than

この例そのものは，計算機を用いた配列に有利であるが，さきに示したように，lookup や tonight が中心語から遠くはなれてしまう欠点は是正されていない．

（2） 特殊記号

語間の空白とまったく同じことが，見出し文字列中のハイフン，ピリオド（略語），アポストロフなどの特殊記号について起こる．慣用の配列では，それらはあたかも空であるかのようにみなされ，英字の部分だけで配列順序が定められる．

たとえばつぎのように対比できる．

計算機	慣　用
a	a
a. c.	able
a. m.	a. c
able	air
air	air-cooler
air-cooler	aircooler
air-line	airedale
aircooler	air-line
airedale	all
all	am
am	a. m.
and	and

（3） 大文字と小文字

すこし性質の違った問題がある．慣用の配列においては，大文字（A, B, C,…）と小文字（a, b, c,…）には対応があって，相互には弱く区別されるにすぎない．上位（左のほう）の文字列のつづりが同じで，ただ大文字・小文字の別だけが違っているときに，それらの別は配列順序の決定には利用されず，むしろ下位（右のほう）の文字列のつづりのほうが，配列順序を決める．

しかし計算機では，各文字はまったく別な存在として，上位から順に評価される．

計算機 1	計算機 2	慣　用
Babel	Babel	babble
Buddha	Buddha	Babel
babble	Esperanto	Buddha
but	Europe	but
Esperanto	Zamenhof	Esperanto
Europe	Zr	etc
etc	babble	Europe
Zamenhof	but	Zamenhof
Zr	etc	zoom
zoom	zoom	Zr

ただしここで，文字の配列順序をつぎのように想定している．

計算機 1：A<a<B<b<C<c<……<Z<z
計算機 2：A<B<C<…Z<a<b<c<…<z

計算機では，大文字Bを全部配列し終わってから，小文字bが配列される．ところが慣用のほうは，いわば“つづり字”の順に配列されている．

4. 和字列の配列

日本語の文字配列についても，英語におけるとまったく同じ問題があることは，以上の議論からただちに類推できよう．詳細な議論や配列例は省略して簡単に箇条書きにしておこう．もちろん以下の問題はその計算機での文字の配列順序がどうであっても成立する．

（1） 語間の空白

日本語の辞書の多くは，分かち書きはしてないし，複合語なども見出しに立てることが少ないので，確実な証拠をあげることはむずかしいが，おそらく，英語におけると同様に，語間の空白は配列順序に影響を与えないと思われる．しかし計算機では語間の空白は文字として評価される．

（2） 中黒（中点）

外来語を表記するときに，もとの語の切れ目を日本語では中黒（中点）で表示することがある．たとえば「データ・プロセシング」という具合である．こういう中黒も，上記の空白や英語での特殊記号と同じように考えないといけない．

（3） かたかなとひらかな

（4） 清音と濁音

（5） 小さい字（や，ゆ，よ，っなど）

これらについては，英語での大文字と小文字の対応と同じ現象が見られる．つまり計算機の立場からは，それぞれの文字はまったく独立な別な字とされるのにたいして，慣用の配列では，いわば“基本のつづり”

とでもいうべき文字列を，四十数種類の文字（つまり上記のような区別のない文字）の組で作り出し，この基本のつづりにしたがって配列しているのである（なお，計算機の符号系で，濁点記号が独立している場合にも，もちろん配列は奇妙なものになる）．

日本語にしろ英語にしろ，見出しに立ててある文字列が，そのまま配列順序を決めると考えることは，計算機技術のうえからは困難がある．計算機で分類するキーは，見出し語の文字列から派生して作られた別の文字列でなければならない．このキーは，基本の文字の組から選ばれた文字だけで構成された“基本のつづり（primary key string）”が大分類キー（major sorting key）となる．

索引などにおいて，この基本のつづりそのものが見出しに立てられることもあるし，また，この基本のつづりは表面には出なくて，慣用の表記の原つづりのほうが見出しに立つこともある．

（6） 長　音

長音については，表記法の問題と配列法の問題とがあって，両者を分けて考えなければならない．前者の問題の例として，「お」の長音は基準の表記法によるとつぎのように書き分けられる．

おーい（呼び声）
おうい（王位）
おおい（多い）

このようないろいろな表記法を，項目の配列順にどう反映させるかについては，一定の慣用が確立されてはいない．すなわち，表記法におけるこのような書き分けの影響が配列順におよばないようにする，三省堂の明解国語辞典などの方法；長音記号はかなにおきかえて，かなの配列順にする，講談社の国語辞典などの方法；長音記号は空とみなす，平凡社の世界大百科事典などの方法が，あい対立して，並び行なわれている．百科事典配列は，「パラメーター」対「パラメタ」などの長音記号の用法の影響を受けにくいことが利点である．

それぞれの配列順と，基本のつづりとを例示すると，つぎのとおりである．

国語辞典	明解国語辞典	百科事典
オウサマ	オオカミ	オウサマ
オオカミ	オオケエ	オオカミ
オオケエ	オオサマ	オク
オオバア	オオバア	オケ
オク	オク	オバ

（7） 漢　字

一般の辞典や事典では，漢字で書く語についても，発音のかな表記を基本のつづりとして，このかな表記の五十音順に配列してある．ところが，電電公社の電話帳などでは，純粋の発音順配列にはなっていない．すなわち，まず漢字を五十音別に類別しておいて，それからこの漢字の順に配列してある．

国語辞典		電話帳
カワ	川	川
カワ	皮	川施餓鬼
カワカミ	川上	川上
カワキリ	皮切り	川原
カワセ	為替	皮
カワセガキ	川施餓鬼	皮一重
カワチ	河内	皮切り
カワヒトエ	皮一重	河内
カワラ	瓦	河原
カワラ	川原	為替
カワラ	河原	瓦

しかもここでは例示しないが，まず同姓の名前をそろえておいて，その中で名前を配列するようになっている．

このやり方を計算機で再現することは，国語辞典配列よりはむしろ容易であると思われる．電話帳のこのような配列は，おそらく同姓とか，同姓同名とかいう現象がかなり多くみられる点で，国語辞典の場合とは異なっていることを考慮したものであろう．

しかし一般の語彙表に適用することが妥当であるとは思えない．よほど慎重な評価が必要であろう．以下の議論では，電話帳配列については除外する．

5. 解決案

以上のように問題点を洗い出してしまえば，そのトラブルの解決は簡単に思いつくだろう．それはさきにもちょっとふれたとおり，“基本のつづり”を大分類キーとして分類してやることである．基本のつづりというのは，大文字小文字とか，特殊記号とかいった文字の変種に関する情報を除き去った残りの文字列である．この文字列は途中の空白などを除いて，左端につめ合わせ，右端に空白をそえておく．

普通の索引では，このようにして導出された基本のつづりを分類キーとするだけでも，十分に実用的な配列が得られる．しかしこのやり方では，基本のつづりを同じくする語同士つまり家族語（family strings）とでも称すべきものの内部での類別が十分ではない．ここで家族語というのは，つぎのようなものである（電

話帳などは，それぞれの家族語のサイズが極端に大きい例である）．

a. m. | am
かき | かぎ | がき

こういう家族語の内部での類別（たとえば KWIC 索引では，同じ語が何回も現われる）をきちんと行なわせるには，基本のつづりを大分類キー，字種の指示を中分類キー，原つづりを小分類キーとして分類すれば，ほぼその目的を達成できる．字種の指示というのは，原つづりのそれぞれの文字位置にくる文字の種類を，適当な大小順序を有する文字で指定したものである．

字種の指示をつぎのようにすることを提案する．

英　字	
空白	0
特殊記号	3
大文字	6
小文字	8

か　な	
空白	0
特殊記号	1
小さいひらかな	2
小さいかたかな	3
清音ひらかな	4
清音かたかな	5
濁音ひらかな	6
濁音かたかな	7
半濁音ひらかな	8
半濁音かたかな	9

この適用はつぎのような例による．

大分類キー (基本のつづり)	中分類キー (字種の指示)	小分類キー (原つづり)
aircooler△	88838888880…	air-cooler
aircooler△	8888888880…	aircooler
allfoolsday△	68806888830688	All Fools' Day
am△	830830…	a. m.
am△	880…	am

大分類キー：ソフトウエアキシユツノトウコウ△△△
中分類キー：５５５５３５０６６２４０４０６４４４
小分類キー：ソフトウェア△ぎじゅつ△の△どうこう

ただし，家族語のサイズが小さい場合には，こういう厳密なやり方は必要なくて，大分類キーだけでもよい．家族語のサイズがかなり大きくても，一家族の中で同じ原つづりの語がまとまって並ぶだけでよくて，家族内での異なった原つづりの語のあいだの配列順序が，それほど重要でないことも考えられる．

実際問題としても，基本つづりを共有する家族語が一か所にまとめられ（もちろん，家族間の配列順序は重要である），また一家族内でも同じ語はひとまとめにされるが，家族内での異なった語のあいだの配列順序の慣用が確立されているとは思えない．実用上はそれで十分なのであろう．

だからこのような場合には，大分類キー（基本つづり）と，小分類キー（原つづり）とだけで分類してもよいだろう．計算機を用いた配列順序が，慣用の配列順序（細部までは慣用は確立していない）を近似するためには，（1）大分類キー（基本のつづり）だけ，（2）大分類キー（基本のつづり）と小分類キー（原つづり），（3）大中小のすべてのキーを用いるなどのいろいろの段階が考えられる．

6. むすび

見出し語などの言語情報を計算機で分類したときに，その配列順序が慣用の順序と異なることを示し，その対策を論じた．ここでふたたび強調しておかなければならないが，この問題は計算機から人間への交渉場面で，しかも情報の量が相当に大きいときにだけトラブルになるということである．

（昭和 43 年 9 月 25 日受付）

× × ×