Lutèce 第30号抜刷
2002年12月31日発行

文学と電子テクスト

—電子テクストの可能性、技術的問題について—

小栗栖　等

大阪市立大学フランス文学会

文学と電子テキスト

——電子テキストの可能性、技術的問題について——

和歌山大学助教授　小栗栖　等

はじめに

最近では多くの文学作品が電子テキスト化され、インターネット上で入手できる。書店がCD付きのテキストを販売することも多い。文学研究において、電子テキストを利用することはもはや珍しいことではない。しかし、電子テキストをめぐる諸問題や電子テキストの適切な扱い方に関する議論は、少なくとも日本国内ではそれほど活発とは言えない。国外では電子テキストをテーマとする会議も開かれているようだが、文献学的観点から事を論じるといった向きが強いように思われる。どのような可能性を電子テキストそのものが孕んでいるのか、電子テキストのいかなる処理が技術的に可能なのか、といった問題を正面から扱った議論を、少なくとも筆者は耳にしたことがない。また、テキストの電子化にかかわる様々なノウハウ、電子化そのものにまつわる問題に関する情報もあまり流布していない。さらに、インターネット上で無料配付されている電子テキストに関する問題も、これまでのところ、ほとんど注目を浴びていない。

拙論では、上記の三つの問題を、論者が専門とする中世仏文学研究を巡る諸問題として取り上げる。とはいえ、多くの場合、電子テキストの技術的側面は中世仏文学研究のみで妥当性をもつというわけでない。細部では中世仏文学にしかあてはまらないような記述もあるだろうが、全体としては、電子テキストの一般的問題に触れることになるだろう。拙論の題名を「中世仏文学と電子テキスト」としなかったのは、そのためである。

インターネットと電子テキスト

電子テキストを無料で入手できるウェブ・サイトは相当数にのぼる。中世仏文学に関してのみであったとしても、そうしたウェブ・サイトを網羅的に紹介するのは不可能なほどである。主なものとしては、ABU

http://abu.cnam.fr/

やATHENA TEXTES FRANCAIS

http://hypo.ge-dip.etat-ge.ch/www/athena/html/francaut.html

をあげることができるだろう。電子化された作品はすでに百種類以上あるのではないかと思われる。中には『ロランの歌』のように複数の校定本が電子化されているものもある。

しかし、それらの電子テキストは総体として見た場合、怪し気なものが多い。Lancelotの独自の校定をアップロードしているプリンストン大学

http://www.princeton.edu/%7Elancelot/

やナント大学

http://palissy.humana.univ-nantes.fr/CETE/TXT/JAC/ANX/MA/MAframe.html

などの場合を別にすれば、多くの電子テキストは信頼度が著しく低い。ある電子テキストには、オリジナルの校定本が特定されていないし、他の場合には、電子テキストそのものの質が悪い。たとえば、BIBLIOTHECA AUGUSTANA

http://www.fh-augsburg.de/~harsch/gallica/Chronologie/11siecle/Roland/rol__intr.html

では、Raoul Mortier版のオックスフォード本『ロランの歌』がダウンロードできる。とはいえ、ダウンロードできる電子テキストの質はかなりお粗末だと言わねばならない。句読点の間違いはおそらく数百ケ所にのぼるし、Mortierが採用していないテキストへの修正が、そこここに見られる。もちろん、誰しも間違いは犯す。私自身、自分が電子化したテキストの間違いを指摘されたことはある。

しかし、BIBLIOTHECA AUGUSTANAの場合は、極端過ぎる。だが、それでも、まだましだと言わざるを得ないほど、ネット上の電子テキストはひどい。中には光学文字認識ソフト（OCR）を通しただけとしか思えないようなものもあるからである。

以上からも理解される通り、ネット上で入手した電子テキストをそのまま研究に利用するのは危険である。筆者自身、そうした電子テキストは一切利用しない。自分自身か他の専門家の手になる電子テキスト、ネット上の電子テキストに自分自身か他の専門家が徹底的に手を入れた電子テキスト、これら以外は信用に値しない。中世仏文学の領域に関して言えば、日本にはdoc-et-doilという優れたメーリングリストがある。このリスト上に、「中世電子テクストの会」という会があり、正式会員の間で電子テキストのやりとりが行われている。詳細は、管理者の岡田真知夫氏のサイト、ISLE D'AVALON

http://www.comp.metro-u.ac.jp/%7Eokadamac/

を参照されたい。

さて、一部の悲惨な電子テキストをとりあげて、ネット上のすべての電子テキストを否定するのは間違いではないかとの意見もあるかと思われる。しかし、どの電子テキストが優良であり、劣悪であるかが分からない場合、すべての電子テキストを見捨てざるを得ないのは明らかである。たとえば、落丁．乱丁だらけの書籍を置いている書店で、本を買おうと思う人はないだろう。信頼に足らない書物が過半数を越えるような出版社の本を買う人もいない。前者のような書店は現実には存在しないし、後者のような出版社は採算割れで姿を消す。しかし、ネット上ではどんなに劣悪な電子テキストでも、淘汰されて姿を消すことはない。しかも、一般の書籍のように批判を受けることもないし、質の善し悪しを伝える情報システムもない。つまり、インターネットは野放し状態なのである。

今の所、筆者はネット上の電子テキストに関しては、悲観的にならざるを得ない。スキャナを利用して書籍を文字情報化し、誤認識を修正するという作業は単純で誰にでもできる。だが、出来上がった電子テキストの信頼度は、作成者の誠意に依存する。悪貨は良貨を駆逐する、ということわざの通り、優れた電子テキストが劣悪な電子テキストのなかに埋もれ、無視されるのは残念なことだが、手の施しようがないし、事態が改善する見込みもない。

上記のような現状を打開するには、一般書籍の校定本と同様、電子テキストも書評の対象となり、批判にさらされるほかない。出版社も新しい道を模索すべきである。校定本を常に電子テキストとセットにして販売することを検討すべき時だと思う。出版社が高いモラルをもって書籍を頒布していることは誰も疑わない。校定本に付された電子テキストはネット上のそれとは比較にならない信頼度を有するはずであるし、出版されたものであるからには、雑誌の書評欄などでも批判の対象となる。CD-ROM付き校定本を作るのは技術的にはさほど難しいことではない。多くの校定本はコンピューターで組版されており、いったん電子テキスト化の過程を経ているからである。

もちろん、出版社の恐れは理解できなくはない。電子テキストは簡単にコピーできるので違法コピーが増えるのではないかというわけである。しかし、実際の所、電子テキストは紙媒体の書籍を駆逐するものではない。コンピューターの画面上で読書などできないことは、今や常識である。プリントアウトする手間と費用を考えるなら、現在の書籍の価格は十分にリーズナブルだといえる。価格を10パーセントほど上乗せし、セットでのみ校定本を販売するなら、コピーの誘惑に駆られる者は少ないだろう。また、電子化はテキストだけで十分なので、批判註や序文は紙媒体のみでの提供とすればよい。そうすれば、電子テキストだけから、オリジナル書籍のコピーを作るのは不可能になる。むろん、書籍版とCDの両方をコピーする者がないとは言わない。だが、その手間を惜しまないものは、紙媒体の書籍でもコピーするだろうから、電子テキストでなくても違法コピーを防ぎようがない。

電子テキストの作り方

高機能で安価なスキャナが普及したことは、電子テキストの今後に好影響と悪影響を与えた。手入力で電子テキストを作らざるを得なかった時代には、手間のかかる作業を覚悟したうえで電子化に取りかかる者が多かったから、電子テキストの品質はおのずと高いものとなった。現在では、スキャナを使えば、とりあえず電子テキストが手に入ることから、たいした覚悟もなしに書籍をスキャンし、ろくすっぽ手間もかけないままに、テキストをネット上で公開するものが後を断たない。電子化にとりかかろうとする者は、スキャン後、相当に手間をかけなければならないことを十分わきまえなければならない。とはいえ、真面目に電子化

に取り組む者にとって、スキャナが心強い味方だというのは事実である。以下ではスキャナを利用した書籍の電子化の具体的ノウハウを論じた後、電子テキストの理想的な書式について説明する。

電子化に際して重要なのは、コンピュータにできる作業はコンピュータに任せ、人間にしかできない部分では手間を惜しまないということである。その意味で、いちいち書籍を開いてスキャナに押し当てるという作業に時間を費やすのは得策ではない。いったん書籍のコピーを取り、フィーダー（自動紙送り機）付きのスキャナを使うようにすれば、格段に能率があがる。フィーダーの使えるスキャナは機種が限られており、またフィーダーを別途購入しなければならないというデメリットはあるが、辛いスキャン作業がなくなることを考えれば、フィーダーの価格はすこぶる安い。

ところで、スキャンの読み取り原稿となる書籍のコピーについては、幾つかの点に注意するだけで、後の作業が相当楽になる。まず、見開きではなく、一頁ずつコピーをとるようにするとよい。コストが気になるのであれば、見開きでセンターがぶれないように丁寧にコピーし、後から半分に切る。さらにコピーの際に拡大しておくと認識精度が向上する。たとえば、CFMA や TLF のテキストであれば、一頁を B5 くらいに拡大するとよい（A4まで拡大しても、認識精度はあまり改善しない）。また、できる限りまっすぐにコピーをとるよう心掛けるべきである。たとえ OCR の自動補正機能を用いても、斜めになった原稿の認識精度ははっきりと悪化する。さらに、余白部分は白紙などで覆ってコピーするようにし、次頁の一部や、書籍のセンターのつなぎ目などがコピーされないようにしておけば、なお良い。ノイズが増えるほど、OCR は誤認識を起こしやすくなる。

スキャン作業とは異なり、文字認識をすべて OCR 任せにするのは考えものである。最近では、画像を連続認識し、一つのテキスト書類にまとめる機能が OCR ソフトには装備されている。だが、特に韻文作品で自動処理機能を使うと、後でひどく難渋することになる。OCR ソフトは、散文を前提として書籍の組版を認識するため、一行ごとに改行の入ったテキストの認識はあまり得意ではない。実際、スキャン画像の文字の部分だけを OCR に認識させると、韻文が改行なしで電子テキスト化され、後で改行を入れるのが大変な作業になる。一方、各行ごとの末尾に十分な余白を持たせて認識させれば、ほぼ確実に各行の末尾に改行が入る。逆から言えば、散文のテキストではできるかぎり、余白のない状態で文字を

認識させれば、不要な改行が入りにくくなる。

OCRの自動処理を推賞しない理由は他にもある。OCRは画像上の文字を全て認識する。これが存外不便なのである。脚注やヘッダ、頁番号、行番号、詩節番号などが一ページに混在したテキストファイルは、電子テキストとしてはほとんど役に立たない。そうしたテキストが欲しいのであれば、単にコピーすればよいのであって、OCRを使うメリットはない。筆者は、OCRは必ず手動で用いる。認識して欲しい本文の文字だけを認識範囲として指定する。もちろん、改行が行末ごとに入るように、右側の余白はできる限り広くとり、左側の余白は可能な限り小さくする。多くの刊行本では行番号が行の右側に配置されているが、これは認識されないようにマスクする。

電子テキストの書式

前項の手順でOCRを使うと、出来上がる電子テキストは、本文のみのものとなる。詩節番号や行番号は残した方が良いと思う人もあるだろう。だが、後から述べる理由で、これらはコンピューター処理の障害となる。電子テキストは単なる書籍の複製ではなく、コンピュータ処理に適したデータでなければならない。したがって、電子テキストの質は、オリジナルテキストを忠実に再現しているかいなかだけではなく、電算処理に適した書式にテキストが再編成されているかどうかによっても、大きく左右される。以下では、理想的な電子テキストを作るためのノウハウを論じる。

OCR作業が終了した時点では、電子テキストは完成からはほど遠い。少なからぬ文字が誤認識されている。最近のOCRは内部に装備された辞書を使い単語単位で文字を認識するが、中世フランス語の単語はOCRの辞書にはない。そのため、中世のテキストは、現代仏語のテキストに比べて、OCRの認識率をかなり押し下げるのである。とはいえ、それらの誤認識の修正にとりかかる前に、改行の処理をおこなわねばならない。OCR作業後の電子テキストでは、必要なところに改行が入ってなかったり、不要なところに改行が入っていたりする。これを適切に処理することにより、オリジナルの書籍との比較が容易になるから、とりかかるべきは改行の修正である。

改行も常にオリジナルテキストのまま再現すれば事足りるというものではない。刊行本の中には、行番号にa,b,cといった文字を追加して、複数行に同一行番号

を与えるものが少なからずある。また、写本の行脱落に応じて、行番号がとんでいるものもある。こうしたイレギュラーを放置すると、電算処理の大きな障害になる。したがって、適切なやり方で電子テキストに改行を入れるようにしなければならない。詳細については別の機会に譲るとして、基本を指摘しておく。それは、オリジナルの書籍の行番号と電子テキストの段落番号が常に一致するようにする、ということである。書籍の行番号がとんでいる場合には、必要な数だけ電子テキストに空行を入れる。逆に、複数の行に同じ行番号がついている場合には、改行なしで、複数行を一段落内におさめる。ちなみに、段落番号は多くのワープロソフトやテキストエディタで自動表示される。

次に、詩節の処理も行わねばならない。基本は、詩節の第一行目に、本文で使われていない文字を一文字入れるだけである。たとえば、アステリスクが本文中で使われていなければ、詩節が変わるたびに、アステリスクを入力する。そんなことをするくらいなら、詩節番号を残しておけば良かったのにと思われるかも知れない。だが、詩節番号は多くの場合ローマ数字で表記されている。ローマ数字はOCRの認識率がすこぶる悪いうえに、ローマ数字の文字が本文中でも使われるために、電算処理を著しく困難にする。ところで、詩節の変わり目にアステリスクを入れるだけで、番号をつけなくて良いのかと思う人もあるだろう。結論から言えば、電算処理の観点からは必要ない。詩節の変わり目が確実にマーキングされていれば、詩節番号を自動的につけることができる。とはいえ、筆者自身は、マーキングを忘れがちなので、／で詩節番号を挟むという方式をとっている（もちろん、本文中で／が使われていれば別の方法をとる）。

誤認識の修正

書式が整ったら、初めて、誤認識によって生じた誤字・脱字の修正が始まる。最初に、文字認識が容易なフォントでテキストを表示させる。フォントによっては、大文字のIと小文字のlの区別がつけにくいものもある。Courierが最適だろう。文字サイズもできる限り大きくする。以上を行うだけでも、誤認識の見落としは格段に減少する。

修正の効率はワープロやエディタの一括置換（全文置換）の機能を使いこなせるかどうかで、かなりの差がでる。OCRソフトは、同じ間違いを何度も繰り返すので、最初の数百行の修正作業の際に出くわした誤認識は、できる限り一括置

換で修正した方がよい。ただし、一括置換を行う際には、過剰修正を行わないよう、十分に慎重を期さねばならない。たとえば、Rollant が頻繁に lSollant となっている場合、lS を R に全文置換するのはかなり危ない。大文字小文字の区別なしに置換してしまえば、nuls が nuR になったり、fils が fiR になってしまい、かえって手間がかかる。たとえ、大文字と小文字を区別しても、lS が常に R と等価になるとは限らない。筆者の利用している OCR ソフトは頻繁に小文字の s を大文字の S として認識する。だから、nulS や filS が nuR や fiR になってしまう。

過剰修正の不都合を避ける方法は二つある。まず、いきなり一括置換を行わずに、最初の五回ほどの置換を手動で繰り返すやり方がある。多くのワープロソフトやエディタには「置換後再検索」という機能がある。これを使えば、置換前に置換されるテキストを確認できる。次に置換対象となる文字列をなるべく長くするという方法がある。たとえば、lSollant を Rollant に修正する際、lS を R に一括置換するのではなく、lSollant を Rollant に一括置換するのである。これだけでも、思いもかけない単語が書き換えられてしまう危険はかなり減る。しかし、さらに慎重を期すなら、置換前の文字列 lSollant と置換後の文字列 Rollant の、それぞれの後ろにスペースを一つおく。これらなら、さらに過剰修正の危険は減じる。とはいえ、Rollant の後にはスペースではなく、句読点や改行が入っている場合もあるので、修正が不十分になる。こうした不都合はごく初歩的な正規表現を身につけることで解決できるが、あまり一括置換による修正に血眼になるのも良くない。OCR は同じ間違いを繰り返すが、必ずしも一貫して同じ間違いを繰り返すわけではない。lSollant ではなく、lSoHant ととか、lS6llant などといったふうに、別の間違いが組み合わさる場合もある。だから、lSollant のすべてを厳密に Rollant に置換できたところで、万事がうまく行くわけではない。

人により用いる OCR ソフトは違うので、一般論を言うことは難しいが、頻発する誤認識を筆者の経験から指摘しておく。

小文字の l は頻繁に数字の 1 になる。また、大文字の O や時には小文字の o も、数字の 0 になる。中世の作品の場合、本文中ではアラビア数字が使われないことが多い。そうであれば、まず最初に、これらの数字を一括置換することをお勧めする。OCR と同様、我々の目にも、l と 1、O と 0 の区別は困難である。他にもアルファベが数字と置き換わる例は多い。小文字の o が数字の 6、大文字の S や小文字の s が数字の 5、J の大文字が数字の7になるなどである。細かい修正を始

める前に、とりあえず、数字の2-8も検索してみた方が良いだろう。頻発する語でこうした数字との置き換えが起こっている場合、一括置換でかなりの手間が省けることもある。

!は頻繁に大文字のIと置き換わるし、定冠詞や人称代名詞のliは頻繁にhになる。前者では大文字のIを!に一括置換する方法は当然とれない。だが、回り道をして、大多数の不適切なIを!に修正することは可能である。まず、改行の前のスペースを完全に取り除き、「I＋改行」を「!＋改行」に置き換えるのである。liが誤認識されたhの方は、「スペース＋h＋スペース」で大部分を修正できる。

トレマのついた文字は極端に認識率が下がる。とはいえ、これを一括置換すると、しばしば、次の事実を知ることになる。すなわち、多くの校定者は多かれ少なかれトレマの用法に一貫性を欠いている。一方で、固有名詞のMarsillionのように、音節数が一定しない語もあるので、トレマに一括置換を使う際には十分に注意しなければならない。

上記のように一括置換を利用しておおまかに間違いを修正した後は、原本と見比べながら、修正を進める。これに関しては、効率をあげる工夫はない。ひたすら単純な作業である。とはいえ、スキャンする際に作ったコピー原稿は、この修正作業の際にも大いに役立つ。オリジナルの書籍よりは文字が大きくなっているし、本を押さえておく必要もない。何よりも、コンピューター用の原稿置きが使える。手作業による修正は辛いものなので、できる限り、作業のやりやすい環境を整えるべきである。

一通りの修正を終えた後は、筆者は多くの場合、プリントアウトする。これはコンピューターに向かう時間を減らすためと、こまごまとした空き時間を利用して修正を進めるためである。プリントアウトする場合には、できる限り、文字の大きさを、見比べる書籍やコピーと揃えた方がよい。大きさの異なる文字列を見比べるのは、案外疲れるものである。とはいえ、そもそも二つの文書を見比べる作業自体がかなり疲れる。筆者自身は、プリントアウト原稿と書籍のコピーの両方に定規を当て、一行ずつずらして見比べるという方法をとっている。行をとばしてしまわないためということもあるが、それ以上に疲れを防ぐためである。二つの原稿を眼で行き来すると、どうしてもすぐには目標の行が見つからず、目線がうろうろと動いてしまう。これが非常に疲れるのである。

再度の見直しの際には、特に注意すべき点がある。それは、第一回目の修正の

際に施した詩節番号のマーキングや句読点を個別に見直すことである。つまり、ひたすら電子テキストとオリジナルの行末だけを見比べ続けたり、詩節番号の変わり目だけを次々に確認したりするのである。少なくとも筆者は、誤字・脱字の修正に集中していると、こうした細部がどうしても蔑ろになる。

誤認識の修正にどれほど手間を掛けるかに関しては、明確な基準を設けがたい。二度見直しただけで、ほとんど間違いのない電子テキストを作ることができる人もあれば、そうでない人もいる。慣れの問題もある。しかし、一般論としては、二度丁寧に見直せば、電子テキストの品質はかなり高いものとなる。個人の手で電子テキストを作成するのであれば、二度の見直しあたりが限界だろう。他人に提供しても、ひどく恥ずかしい思いをすることはないと思う。とはいえ、筆者自身は粗忽者との自覚があるので、多くの場合、三度見直すことにしている。 最後に、出来上がった電子テキストに対し、次の点を確認しておくとよい。まず、最終行にも改行をつけておくこと。第二に改行の直前のスペースは取り除いておくこと。第三に句読点の前後のスペースを一貫させておくこと（行の途中でのポワン、ヴィルギュルの後にはスペースが必要。ドゥ・ポワン、ポワン・ヴィルギュル、ポワン・ダンテロガション、ポワン・デクスクラマションは、行の途中では前後に、行末では前にスペースが必要。第四にスペースが二個以上連続する場合には、一つにまとめること。

オリジナルテキストの修正

前項までは、オリジナル書籍のテキストをそのまま再現するという前提で話を進めた。しかし、幾つかの点では修正を行った方が良いと思われる。

まず、オリジナルのテキストにErrataが付いている場合、必ず、Errataの指示に従って、テキストを修正しなければならない。校定者が間違いだと認めているテキストを残す理由は何一つ無い。

明らかな誤植も修正すべきだろう。たとえば、詩節が台詞で終わっているのに、右ギュメで台詞が閉じられていないというのはありがちな誤植である。もっと微妙な判断を要する場合には、電子テキストを修正するのではなく、添付書類を作り、そこに修正案を記載する。

以上は、修正がほぼ義務になる場合である。一方、電算処理により適した電子

テキストを作りたいと考える人には、次のような修正を提案することができる。

まず、記号類を一貫した方法で一義的に用いるようにすれば、電算処理は大幅に容易になる。

たとえば、多くの校定者は、登場人物の台詞を時にはギュメで、時にはティレで挟んでいる。しかし、電算処理の観点からは、一貫してギュメを使用するべきである。ティレは左右の区別がないので電算処理を難しくするし、同意義で複数の記号を使うのは効率が悪い（ティレとギュメの微妙な違いを無視するわけではないが、コンピューターで用いることのできる記号は限られている）。

アポストロフや"を引用符として使用するのも避けた方がよい。エリジョンとの区別や、単語の強調との区別が困難になる。また、シングル・クオーテーションやダブル・クォーテーションの使用も勧められない。アポストロフなどと見た目で区別しにくいため、混乱を来しやすいからである。作成者以外の者が電算処理しようとした場合のみならず、作成者本人でさえも、作成後時間がたてば、これらの記号を見間違える可能性はある。さらに、校定本によっては、[]を加筆、()を削除の意味で使っている。そうした場合には、この二種のカッコを別の用途に使わないようにする。特に、()は句読点を補うものとして使われているが、＜＞などの別の記号で代用する。電子テキスト内で記号使用を一義化、一貫化する試みについては、筆者が校訂した電子テキスト版『ロランの歌』の序文を参照されたい。現在、ベータ版（未完成版）の扱いであるが、希望者には配付を行っている（筆者のホームページ、Hruodlandus et Alda : http://www.eonet.ne.jp/~ogurisu/を参照）。

コンピューターは固有名詞を他の語と見分けることはできない。それゆえ、固有名詞を語頭だけでなく、語全体にわたり大文字で表記するのも好ましい。詩行冒頭のGuiが人名なのか、動詞なのかの判断は、時に人間であっても難しいので、この表記法は電算処理を越えたメリットもある。だが、いかんせん慣習には反する。それゆえ、こうした書き換えを躊躇する人もあるかも知れない。実際の所、大文字のみで書かれた単語を冒頭部のみ大文字で始まる単語に書き換えるのは、電算処理で簡単にできる。

GUILLAUME_d'ORANGEのように固有名詞の複数の語をアンダースコアでつなげるのも一つの工夫である。固有名詞に限らず、この方法で連語をつないでおけば、連語を一つの単語として扱うことができる。Unix, Windows, Macintosh で利用できる、プログラミングソフトの Perl では、アンダースコア

はアルファベと同等の扱いをうける唯一の記号である。

データベース・データとしての電子テキスト

データベース・データはFileMakerProなどのデーターベース・ソフトでしか作れないと思っている人は意外に多い。Excelなどの表計算ソフトで作ったデータをデーターベース・ソフトで開くことができ、二種類のソフトのデータに互換性があることを知っている人でも、エディタやワープロでもデータベース・データを作ることができる、ということを知らなかったりする。

実のところ、データーベース・ソフト、表計算ソフト、エディタ（ワープロ）は、データーベース・データを異なった形式で開いたり、異なった方法で書き込んだりするに過ぎず、あらゆる文字データはある種のデーターベース・データなのである。したがって、電子テキストもデータベース・データである。

データーベース・データは、フィールドに分割された一つ以上の情報からなるレコードが一つ以上集まったもの、と表現できる。とはいえ、複数のフィールドからなるレコードが複数あったとしても、それだけではデータベース・データには成り得ない。複数のレコードが同一のデータベース・データを形成するためには、各レコードのフィールドが一定の方法で整理されていなければならない。

たとえば、表計算ソフトで単語のデータを整理する場合を考えてみよう。この場合、一つの単語に対して、発音、品詞、語義、用例、語源といった情報が附随する。大事なのは、単語自体をはじめとして、発音、品詞といった情報は、それぞれ、表計算の枠組みの中で、どの列に入れるかを決めておかねばならないということである。すなわち、A列は単語そのもの、B列には発音、C列には品詞、といった具合である。こうすることにより、表計算の一行は一レコードを形成することになる。もし、どの列にどういった情報を入れるかを決めずに、やみくもに入力すれば、フィールドは同じやり方で整理されないことになり、表をうめた情報はデータベース・データとして機能しない。たとえば、発音は、不規則な単語に関してのみ記入するとして、発音が規則的な単語の場合は、A列は単語、B列は品詞としてしまたら、あるレコードではB列に品詞が、別のレコードでは同じB列に発音が入ってしまう。発音の記載が必要無い場合でも、B列は空欄で残しておかねばならないのである。意外に思う人もあるかも知れないが、「情報なし」も立派な情報なのである。

さて、もう一度、この項の冒頭で述べたことを考えてみよう。すなわち、「電子テキストもデーターベース・データ」だということである。電子テキストの一段落が一レコードで、本文という一フィールドが各レコードに入っていると考えた人は、データーベースをかなり理解していることになる。ただし、実際には電子テキストは二つ以上のフィールドを持っている。たとえば、プログラミングでテキスト・ファイルを処理する場合、段落の概念は重要な意味をもつ。電子テキストを作る際に行番号を入力する必要がないことはすでに述べたが、それは、コンピュータ処理では、テキストファイルの段落番号は常に自動的に検出されるからである。したがって、この観点から見れば、電子テキストは、本文と段落番号の二つのフィールドをもったデータベース・データだと言える。実際、通常のワープロのように欄外にではなく、本文に段落番号を書き込むソフトは非常に簡単に作成できる。潜在的に存在した第二フィールドの段落番号を取り出して、文字として書き出せば良いだけだからである。逆から言えば、オリジナルの書籍の行番号と、電子テキストの段落番号がずれるのは好ましくない。番号を自動的につけられなくなるだけではない。索引作成ソフトなどを使って、索引を作った場合、行参照がずれることになる。

電子テキストの書式変換

適切な書式を与えられた電子テキストが、それ自体、本文と行番号をフィールドとした行数分のレコードからなる、データベース・データとして機能するということは、前項で述べた通りである。もちろん、他の形式のデーターベース・データへの変換も可能である。たとえば、索引ソフトは電子テストの全単語のレコードを作り、第二のフィールドにその単語が見つかる行の番号を入れることで機能する。つまり、「本文の一行－行番号」というデーターベース・データ（「」は一レコードを表し、－はフィールド区切りを表す）が「単語－参照行番号」という形式で再編成されることになる。それが明らかになれば、詩節の句切れをマーキングするだけで、詩節番号は不要だと述べた理由も容易に理解されよう。電子テキストを行単位で区別する場合には改行がレコードの区切りを表す。それに対し、詩節の区切りとなる記号を基準にレコードをわければ、一詩節を一レコードとするデータベース・データとなる。コンピューターは数を数えるのが得意なので、詩節番号は行番号と同様に潜在的に準備されている。すなわち、プログラミング・

レベルでは、改行以外の区切り記号でテキストデータをフィールド分けし、番号つきでメモリに格納するのは単純作業に過ぎない。それゆえ、適切な書式に従った電子テキストには、行番号や詩節番号を自在に付すことができる。用途に応じ、あらゆる行に行番号を付し、詩節の変わり目に詩節番号を付すことができるばかりか、各行に詩節番号と行番号の両方を付すことも可能である。詩節番号のみをローマ数字に変換したり、行の左側に詩節番号、右側に行番号を付す、あるいはその逆といった具合に番号を配置したり、行番号を四行おき、五行おきにするといったことも、さして難しくない。筆者が作成したマッキントッシュ用のソフト、NUMEROTATION は、これらの番号付けを全てサポートしている。

ところで、番号は様々な文字や文字列を付加して書き込むことができる。行番号と本文の間にタブを入れたり、詩節番号の後に改行を入れたりできるのである。したがって、電子テキストを、ほぼ書籍と同じ書式で表示することができる。とはいえ、話はそれに留まらない。これまでのところ、「番号を書き込む」という表現を使って来たが、実際のプログラムの動作は、「番号と本文を同時に出力する」に近い。したがって、番号だけではなく、本文にも、様々な情報を追加できる。この際には、HTML や LaTeX といったマーキング言語が威力を発揮する。マーキング言語を使ってテキストファイルに様々な書式情報を書き込めば、専用のブラウザで書式付きのテキストを表示できる。一つの文書の中で本文と書式情報が混在するため、マーキング言語は敬遠されがちであるが、使い方によっては、マーキングを意識することなしに使うことができる。たとえば、行番号には本文より数ポイント小さい活字を指定するマーキングを付し、詩節番号には本文より数ポイント大きい活字、センター配置を指定するマーキングを施すようなプログラムを一度作ってしまえば、同じ書式に従ったあらゆる電子テキストを自動的に処理できる。マーキング言語に HTML を使えば、ホーム・ページで表示できるし、LaTeX を使えば、通常のプリンターのみならず、写植機でも出力できる。

こうしてみると、電子テキストを作成する際に書籍の書式に似せようと努力するのが、いかに無駄なことかがわかる。実際、書籍をまねればまねるほど、電子テキストは汎用性を失う。適切な形式の電子テキストは、書籍の書式で出力できるだけなく、様々なデータベース・データに変換可能なのは確認した通りである。一方、書籍をまねた電子テキストから、なんらかのデータベース・データを作ろうとすれば、非常に手間がかかることになる。

とはいえ、ここで誤解を解いておこう。筆者は電子テキストをホームページにしたり、書籍として写植出力したりすることを無条件で勧めているのではない。それらは著作権を侵害する可能性がある。だが、すでに著作権の期限が切れ、書物そのものも容易に手に入らない刊行本も少なくない。そうしたものを汎用性のある書式で電子化することは、倫理的にも法的にもとがめられる故はない。また、こちらの方が重要だが、自分でテキスト校定を行う場合、ワープロで書式を付けることがいかに無意味かを理解して欲しいのである。電子テキストと同様に汎用性のある形式でデータを作っておけば、有効な使い方ができる。そして、願わくば、出版の際には、そのデータがCD-ROMに収録されますように。

これまでのところ、行番号や詩節番号のみしぼって話を進めてきた。だが、LaTeXの場合、さらに様々なことが可能である。すべてを説明することはできないが、特に、自前の校定本を作成する場合にぜひとも必要と思われる脚注に触れておこう。近年発行される刊行本の多くは、ヴァリアント等を脚注に収録している。これは読者にとっては親切なはからいである。著作権期限切れの刊行本や自前の校定本を電子化した後、印刷の際に同様の書式を採用したいと考えるのは自然なことだろう。

脚注は本文とは異なった書類として作成する。本文と脚注のウィンドウを別々に開くのは煩わしいと思われるかも知れないが、実行してみれば、ワープロなどの脚注よりもずっと扱いやすいことがわかるはずである。脚注用の書類には、あらかじめ本文の行数分だけの行を確保し、行番号とタブを付しておく。後は、たとえば、本文の第10行に脚注を付けたい場合には、脚注書類の行番号10の行に書き込むのである。この際、タブの後から註を書きはじめなければならない。脚注本文内ではタブと改行文字は使えない。改行したければ、￥￥を入力しておく。タブは ￥hspace{10mm} などと長さで指定する。これが煩わしいと思うのであれば、脚注書類の行番号の左右を異なった記号で挟んでおけばよい（それらの記号は脚注内で使用してはいけない）。脚注書類が完成した時点で、一括置換を使えば、簡単に上に述べた書式に変換できる。

上記の通り脚注書類が準備されていれば、LaTeXの場合、簡単に本文内に脚注を流し込むことができる。LaTeXは ￥footnote{ というマーキングの{と}の間に入力された文字列を脚注に仕立てるからである。本文行の末尾に、脚注書類の対応する文をマーキングとともに書き込むというテキスト操作もプログラミン

グの観点からは単純作業に属する。

ここまでの話の進め方に不満を感じる人もあるだろう。様々な書式やデーター形式の変換ができることはわかっても、それをどのようにすれば良いかが分からないからである。しかしながら、本稿の目的は、論者が主張する「汎用性のある書式」に従うことで、何が可能になるのかを、プログラミングに関する知識のない人に理解してもらうことにある。筆者は上に述べたような作業を自動化するPerlスクリプトを全て準備しているし、実際に使用もしている。これらのスクリプトはいずれ筆者のホームページで公開する予定である。個人的な問い合わせにも応じる容易がある。また、さらに突っ込んだ技術的な問題に関しては稿を改めて論じる予定である。

電子テキストのデータ形式変換

すでにのべた通り、索引はある種のデータベース・データを整形したものである。すなわち、「単語－参照行番号」というデータをソートし、同一つづりの単語の参照行番号を、一つの見出し語の後にまとめて作られる。

ところで、コンピューターで全語彙索引を作成するのは、専用ソフトを使えば簡単にできるが、出来上がった索引は学者が作った網羅的語彙集とは比べ物にならないほど貧弱である。人間はたとえ同一のつづりであっても、文脈から品詞を特定したり、変化形の原形を見極めたりするが、コンピュータにはそれができないからである。もちろん、だからこそ、文献学者の存在意義があるわけなので、この部分で手を抜きたいと思う者はないだろう。とはいえ、不要な手間をかけるのは時間の無駄に過ぎない。コンピューター上での作業では、工夫次第で、様々な作業を簡便にし、間違いを最小限に押さえることができる。

全語彙索引を作る方法として多くの人が考えるのはデータベース・ソフトを使うことだろう。たしかに、「単語－行参照」というデータを作り、データベースに読み込むのは難しくない。だが、追加データの入力作業は想像以上に辛い。いちいち元のテキストと見比べながら、特定の行の品詞などを入力しなければならないからである。ちょうどかつての文献学者がテキストを読み進めつつ、カードを作っていったようにである。独立した二つのデータ（テキストとデータ）を行き来するという作業は想像以上に手間がかかる。テキスト読解を主にして、デー

タベースで見出し語を検索しつつ、必要事項を記入していくか、逆にデータベースの見出し語を、テキスト内に見つけ出し、記すべき情報を特定していくか、いずれの方法をとるにしても、テキストかデータかを常に検索しつづけなければならない。

上記のような手間を避ける一番の方法は、テキストに直接文法情報を書き込んでしまうというやりかたである。これにより、二つのデータを行き来するという手間がなくなる。書き込みが満載された電子テキストは後で一気に索引作成ソフトで索引化できる。もちろん、何の取り決めもなしに書き込みを行えば、索引作成ソフトが正常に動作しなくなるので、ソフトの裏をかく必要はある。すなわち、索引作成ソフトは句読点やスペース、改行をたよりに個別の単語を認識する。逆から言えば、単語に接した文字列は単語の一部と見なされるのである。だから、単語の後に、単語の文法情報を直接書き込めば良い。ただし、直接くっつけたのでは後の処理が厄介になるので、たとえば、

li{li | art | suj}
cuens{conte | nm | suj}

などとする（{{や | が本文中で他の用途に使われていないことが前提である）。こうしておけば、索引作成ソフトは、単語の綴りが同一なだけでは、一つの見出し語の後に参照行番号をまとめない。文法情報までが同一の場合にのみ、同一単語と見なし、ひとまとめにするのである。

注意しなければならないことが幾つかある。まず、{{内は潜在的に追加された諸フィールドに相当にする。{{内では | がフィールドを区切る役割を果たしている。したがって、データベース・データに関する注意事項は、ここでも厳密に守らなければならない。すなわち、フィールドの数をふぞろいにしたり、フィールドの順番を変えたり、同一のフィールドにまちまちな情報が入っていてはいけない。そういうことをすれば、思った通りの索引ができなくなる可能性が生じるばかりか、今後、それらのデータを多様な用途に用いるための汎用性が失われる。上に述べたことを守っていれば、できあがった索引をさらに加工して、単語、原形、品詞、格、参照行番号といったフィールドに分割されたデータを作ることもできる。

筆者の提案する方式の欠点は、通常のエディタやワープロソフトでは、いちい

ち文法情報を手入力しなければならないことである。データベース・ソフトならば、一旦登録した単語をポップアップメニューの選択によって入力できる。これはきわめて都合が良い。入力が楽になるだけでなく、タイプミスを避けることができるからである。とはいえ、エディタでも自動入力は不可能ではない。マッキントッシュならば、AppleScript を使うことでメニュー選択による自動入力が可能である。また、日本語の OS を使う限り、かな漢字変換ソフトでユーザー辞書を作り、たとえば、「ひんし」と入力すれば、「art, nm, ns, verb....」といった選択肢が表示されるようにすることができる。筆者は現在 AppleScript を使用しているが、専用のエディタを作成できないか検討中である。

結論にかえて

本稿の目的は、現在、個人個人でまちまちに行われている電子テキストの作業に、一定の指針を提案するということにある。現在のところ、電子テキストの書式は人により様々である。むろん、それらの電子テキストの多くは、個別的には一貫した基準にそって作られている。しかし、今後も電子化されるテキストの数が増えていくことを思えば、いずれは、電子テキストを書式に悩まされる日が来るだろう。現在の状況では、電子テキストの書式を書き換えたり、データ形式を変換するためのソフトを作っても、筆者自身しか使うことができない。ソフトを使う前に、テキストの書式を整える必要があるためである。時間をかけて作ったソフトが自分にしか利用できないのは不本意であるし、多くの人が時間を費やした電子テキストが有効に利用されないままになるのは、もっと残念なことである。もちろん、私が提案した書式は万全とは言えないかも知れない。細かな例外の対処法などは紙幅の関係で割愛せざるをえなかった。とまれ、本稿は一種の公約でもある。本稿に提案した書式に従った電子テキストを前提にしたソフトを今後公表するし、すでに公表した多くのソフトも同じ書式を前提としている。筆者が公開しているソフトは全て、筆者のホームページ、H&ASoftware

http://hp.vector.co.jp/authors/VA026513/

でダウンロードすることができる。

なお、現在筆者は本稿と同様、テキスト電算処理を扱った「テキストの電算処理――電子テキストと解析ツール――」、「電子テキストの書式――より快適な電算処理のために――」という論文も準備している。これらの発表雑誌は現在未定であるが、発表雑誌刊行後に、筆者のホームページ

Hruodlandus et Alda : http://www.eonet.ne.jp/~ogurisu/

に本稿とともに全文掲載する予定である。