中世文学研究におけるコンピュータ利用

原野　昇
中川　正弘
太古　隆治
前田　弘隆

はじめに

文学作品をテクストデータとしてコンピュータで処理するということは，すでに何十年も前から行われてきた。例えば *Trésor de la langue française* は，国家事業として，文学作品を含んだ膨大な資料を，最初からコンピュータで処理して編纂されてきた。

また各作品ごとのコンコーダンスも，個人研究者によって，あるいは組織的に作成されてきた。例えば，中世文学の分野で言えば，M. STASSE, Jean Renart, *Le Lai de l'ombre,* Concordance et index établis d'après l'édition J. Orr, Publication de l'Institut de lexicologie française de l'Université de Liège, 1979 は個人によるものであり， G. de POERCK, R. van DEYCK et R. ZWENPOEL, *Le Charroi de Nîmes*, 1970, Librairie-Editions Mallier や，同じ著者たちによる *Les Oeuvres de François Villon* はグループで«Texte et Traitement Automatique» シリーズとして刊行しているものである。

I．パソコンによるコンコーダンス作成

これらはいずれも非常に大がかりな機械を利用したものであり，膨大なエネルギーを費やして実現されたものである。そういう意味では，出来上がったコンコーダンスが利用者にとって非常に便利で有効な研究道具ではあっても，誰でもが自分でコンコーダンスの作成を試みるというわけにはいかなかった。ところがこの分野における現今の急速な技術革新の結果，小型のパーソナルなコンピュータでかなりのことができるようになった。

とは言え，一人で多くの作品のコンコーダンスを作成するとなると，膨大なエネルギーを必要とするのみでなく，他の研究者と重複した場合には無駄になることもある。そこで複数の研究者がそれぞれ手分けして，それぞれ一つの作品のコンコーダンスを作成し，それらをお互いに共通のデータとして利用し合えば，格段に利用価値が増してくる。

例えば各作品ごとのコンコーダンスを重ねて，共通語彙を抜き出せば，基礎語

彙や共通性の高い語彙の一覧表が得られると同時に，残りの語彙から各作品独自の特徴的な語彙が浮かび上がってくることになる。

II．現在の状況

このような考え方が期せずして，広島大学フランス中世文学研究会のメンバーの間で自然発生的に沸き起こり，まず個別のコンコーダンス作成が先行し，現在そのグループ化および共通データベース化が遅ればせながら進行しつつあるところである。

まず中川が *Merlin* のテクストをデータベース化し，1995 年 3 月にそのコンコーダンスを完成した。続いて太古が *Enéas* のテクストをデータベース化し，1996 年 1 月にそのコンコーダンスを完成した。

原野はすでに1983年に *Le Roman de Renart* のテクストをキーボードによる手動入力によってデータベース化していたが，8 インチのフロッピーディスクに記録されたデータを現在のパソコンで読み取り可能なようにコンバートすることができず，やむなくこれを放棄し，このたび新たにデータベースの構築からやり直すことになり，現在，前田および重見晋也氏の手を借りて，テクストのデータベース化と，それと並行してコンコーダンス作成を行っている最中である。

前田は原野の *Le Roman de Renart* のデータベース化とコンコーダンス作成を手伝い，その蓄積を基に新たに *Prise d'Orange* のデータベース化とコンコーダンス作成を計画中である。

将来的にはこれら各作品ごとのコンコーダンスを結合して，より大きなデータベースを構築していく計画である。

なお原野担当の *Le Roman de Renart* のテクストデータベースとコンコーダンスの一部は，インターネット(WWW)を通して，世界中の研究者がどこからでもアクセスでき，誰でもが利用できるように解放してある。そのサイト（site）は広島大学文学部フランス語学フランス文学教室のホームページに接続してあり，そのアドレスは以下のとおりである。http//www.hiroshima-u.ac.jp.~france/RRenart.html#edition

III．テクストのデータベース化とコンコーダンス作成

具体的な作業は，各作品をコンピュータによる処理が可能なテクストデータベース化する作業と，それを利用してコンコーダンスを作成する作業とからなる。

前者はＯＣＲ（光学的文字読み取り）ソフトを利用する。以前はキーボードを利用して一文字ずつ入力しなければならなかったが，ＯＣＲの出現によってこの作業が格段に簡素化されたのである。ＯＣＲ自身の小型化も目覚ましく，以前に

は事務机を３つ連ねたくらいのスペースが必要であった（それも数百万円もした）が，現在では餅箱くらいのスキャナー（数万円）が一つあれば用が足りるのである。スキャナーで読み取った画像を文字および単語として識別し，テクストベース化するソフトウェアとして我々は “Omni Page”を利用している。

こうして出来上がったテクストデータベースを利用してコンコーダンスを作成するのに我々が利用したソフトは “Nisus Writer” である。このなかの「索引作成」というプログラムを利用して，「全ての単語を索引に掲載する」というコマンド（コンピュータに対する命令）で，結果的にコンコーダンスが出来上がるという仕組みである。

IV.　具体的な作業手順

以下において“Omni Page”によるテクストデータベース作成と，“Nisus Writer”を利用してのコンコーダンス作成の手順を簡単に説明しておく。なおパソコンはいずれも Macintosh を利用している。

・*Merlin*のテクスト･データベースとコンコーダンスの作成

ここではテクスト･データベース作成とコンコーダンス作成というごく単純な作業を，そこで出合うさまざまなトラブルとともに紹介する。これによりコンピュータ使用の基礎とコンピュータにおける言語処理の一面を理解してもらいたい。また，ここに紹介したコンピュータの利用法が有効であると思われれば，これを簡易マニュアルとしてぜひ試みていただきたい。

1. OmniPage 上の作業

1.1. アプリケーションの設定

スキャナーの接続が正しく， OmniPage がちゃんと起動していても，使用するスキャナーを指定しておかなければ，第一段階の画像の取り込みができない。**Settings→Verify Scanner** でスキャナーのメーカーと機種を選んでおく。

出荷時の標準設定に特に変更を加えなければ， **Auto** をクリックすることで，**ScanImage**（画像取り込み）**→AutoZones**（認識ゾーン自動設定**→PerformOCR**（文字認識）まで自動的に処理される。しかし， *Merlin* (Robert de BORON, *Merlin* – roman du XIIIe siècle, édition critique par Alexandro Micha, textes littéraires français,Droz, 1979)のように行番号や脚注などでページのレイアウトが複雑になっている場合，ページ上のすべての文字を認識対象としてしまう **AutoZones** から **ManualZones** に切り替え, 必要なテクスト部分だけを認識させるべきだろう。

認識作業には，直ちに実行する **PerformOCR**（実行）だけでなく，**OCR&Check**（認識と確認）／**DeferOCR**（遅延）／**TrainOCR**（学習）というオプションがあるが，一連の操作に馴れるまでは **PerformOCR**（実行）のままでもいいだろう。**OCR&Check**（認識と確認）にすれば，文字認識後，その結果を認識確認用辞書にある語彙と見比べ，確認と修正をさせてくれる。**DeferOCR**（遅延）は後でまとめて文字認識するために，まず画像データを溜めておくためのものである。スキャナーに AutoSheetFeeder（自動給紙装置）を取り付け，元テクストをこれにセットできる形（認識ゾーンが一定になるように調整したコピー／本の背を切り取り解体したテクスト）に作り変えれば，時間のかかる画像取り込みを１冊分まとめて行うこともできる。また **Train OCR**（学習）は，文字欠けやかすれ，滲みなど，文字の「癖」を登録し，認識エラーを少なくするためのものである。ただ，この学習自体時間のかかる作業だが，*Merlin* のように滲みや文字欠けの多いテクストでは，いくら学習させてもきりがなく，認識作業も重くなってしまうのであまり勧められない。

1.2. テクスト画像から電子テクストへ

手作業で設定したゾーンは保存して繰り返し使うこともできるのだが，*Merlin* はページごとに行数もまちまちなので，画像取り込み１回（見開き２ページ）ごとに認識ゾーンの設定を行うことになる。

画像の右に見えるのが認識後のテクストである。確定できなかった単語はカラー表示されるので修正が容易だ。しかし，それ以外の部分も確定されたからと言って，間違いがないわけではない。

欧文テクストの認識がかなり正確（公称 98%）になるのは，文字自体の図像認識精度もさることながら，認識の際辞書に登録されたスペルを参照するからである。OmniPage には英語，フランス語以外にも多くの言語の辞書が標準で添付されているが，さすがに中世フランス語の辞書までは用意されていない。しかし，OmniPage は与えられたテクストからこのような辞書を極めて簡単に作成できるようになっている。そこで，すでにデータベース化されている中世フランス語のテクストがあれば，それから辞書を作り，認識時に参照させればいいだろう。ただし，中世フランス語のように綴りにヴァリエーションが多く，作品ごとに綴りの癖が違っていると，辞書を参照することで逆に誤認が出る

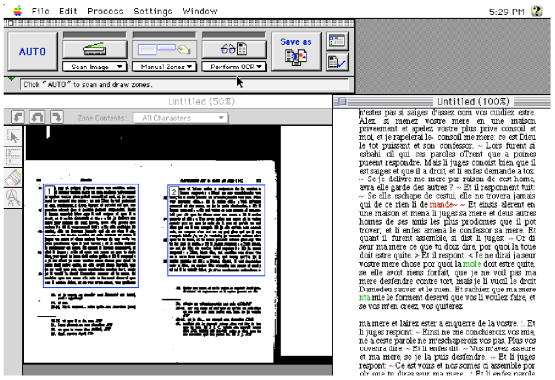


図 1

可能性もあるはずなので，元テクストの選定には慎重でなければならない。

また，そのようなテクストデータベースが入手できなければ，作業対象のテクスト，ここでは *Merlin* の始めの十分の一ほどを注意深く原文と照合確認し，その確認済みのテクストからとりあえず暫定辞書を作り，これをテクストの残りの認識作業に用いることも考えてよかろう。

1.3. 認識結果の修正

上でも触れたように， OmniPage は確定できなかった単語をカラー表示してくれる。また，チェックウィンドウでは辞書にある類似の語彙と見比べながら修正させてもくれる。だが, 誤りは目立たないところにもいろいろある。修正は OmniPage のテクスト上で行うとなると目で一つずつ確認しながら行うしかない。しかし， NisusWriter に写してからなら, 検索／置換を使って一気に修正ができる同一の誤りも多いので，ここで完全に修正しようと思わず，単発的なものだけに限ってもいい。

2. NisusWriter 上の作業

2.1. OmniPage テクストからの読み込み

OmniPage の文字認識結果はこれ固有のテクスト形式だけでなく， Macintosh で用いられるいろいろなワープロソフト（MicrosoftWord， WordPerfect， MacWrite など）の書類形式でも保存できる。しかし，ここで使いたい NisusWriter の書類として保存することはできない。また， NisusWriter の方も， OmniPage のテクストを直接読み込むことはできない。そこで， NisusWriter が読み込める形式，例えば MicrosoftWord （ vers.4-5 まで） の形式でひとまず保存し，これを NisusWriter で読み込むという迂回が必要だ。

MicrosoftWord の書類を Nisus で開くには, ファイルメニュー→ファイルアクセス→インポートでファイル選択ウィンドウを開き，目的の書類を選ぶだけだ。これが煩雑と思うなら， OmniPage のテクストの横に NisusWriter の新規書類を開き, １ページずつ手作業でコピー・アンド・ペースト[編集→全てを選択(command+A)→コピー(command+C) →ペースト(command+ V)]を繰り返す方法もある。

どちらの方法をとるにせよ， NisusWriter にテクストが写せたからといって, OmniPage の文字認識テクストをすぐ廃棄にはしないでしばらく保存しておくほうがいいだろう。操作ミスやコンピュータのエラーで作業中のテクストが消えてしまうことはよくある。そういう場合, 作業前段階のテクストが保存してあれば, そこから作業を再開できる。このようなテクストの転写に限らず，作業が次のステップに移るような時には必ずファイル→保存(command+S)することだ [1]。

2.2. 文字化け修正

OmniPage では正しく表示されていたフランス語が， NisusWriter に写した時，ところどころ妙な漢字に化けることがある。これは，コンピュータの標準表示フォントである Osaka フォントが仮名や漢字だけではなくアルファベットも持ってはいるのだが，フランス語のアクサンや特殊文字を持っていないためだ。

こうなった時は，編集→全てを選択でテクスト全体を黒く反転させ，バーメニューのフォント選択で Chicago， Geneva， Times のような欧文フォントに切り替えると直ることが多い。また，コピー・アンド・ペーストで OmniPage から Nisus Writer に手作業で写すのなら， NisusWriter の新規書類に何もテクストがない段階でフォントを欧文フォントに替えておけば化けにくい。これでだめな場合は，文字化け修正専門の FontPatcin’というシェアウェア（ NiftyServe やコンピュータ雑誌の付録 CD で配布されている有料ソフト）を使うか，英語版 MicrosoftWord で一度開き，再保存してから NisusWriter に取り込ませるような回り道をするしかない。

2.3. OmniPage の認識ミス修正

OmniPage の認識後テクストにカラー表示される未確定文字を直し，さらに，「**i-j**」，「**b-p**」，「**b-h**」，「**n-m-in**」，「**c-e-o** 」，「**i-r-n**」など， OmniPage が確定はしていても文字欠けや滲みのために誤認されている文字を元テクストと照合して正すことはできるのだが，まだ文字のデザイン自体が原因の誤りが隠れている。

「**I-l**-1 （**i** の大文字-**L** の小文字-数字の１）」，「**o-O**-0 （**o** の小文字-**o** の大文字-数字の０）」などはデザインが酷似（全く同一のデザインを使っているフォントもある）しており，文字認識の際， OmniPage はスペルを参照する辞書がないと，これらの文字をデタラメと見えるほどばらまいてしまう。

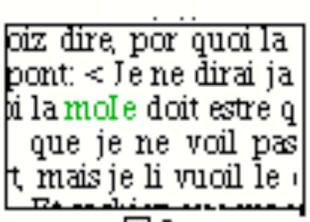


図２

つまり，左に見るように **moie** となるべきところが **moIe** （大文字の **i** 含み）や **mo1e** （数字の１含み）と認識されてしまう。これらは人間なら「正しく」読めることも多く，また見ただけでは区別できないものもあるので，テクストをプリントして使うぶんにはさほど問題がない。ところが，コンピュータ上では大問題になる。コンピュータの有用な機能の多くはすばやく正確にできる「検索」能力の上に築かれているのだが，いくら似ているからと言っても，登録コードの異なる文字は別のものとして処理するので， Index を作ってみるとどうみても同じスペルと見える単語が別項目で出てきたり，単語検索をすると目当ての単語で拾えなかったりする。

このような「大文字」，「数字」の誤認は NisusWriter ツールメニューの検索／

置換で一つずつ確認しながら入れ替えるしかないが，単純な作業なのでさほど時間はとらない。

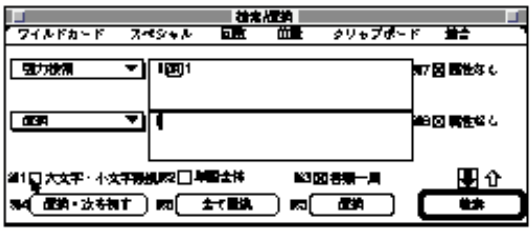


図３

作業目標は正確なコンコーダンスの作成だが，作業途中のスペルチェックのための暫定索引を作り，これでチェックした誤りをテクスト上で検索／置換するという手もある。完全なテクストをまず作成して，それから索引作りができれば理想的だが，誤りをチェックする最大の道具が総索引なのだから，作業用の総索引を暫定的に作り，それを基にテクストをチェックする，この繰り返しで精度を高めていくべきだろう。

2.4. テクストの書式設定

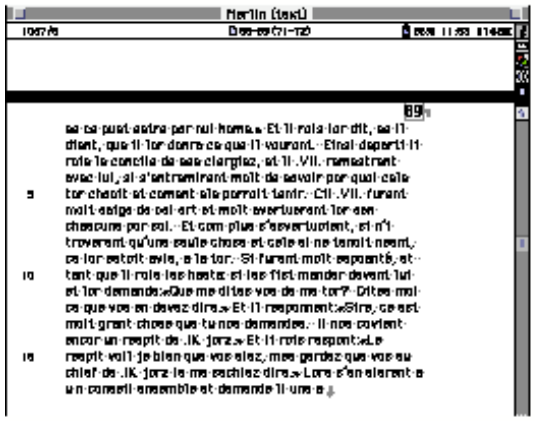


図４

索引作成を目標としているのだから， NisusWriter に写したテクストは原本通りのページ番号が打たれていなければならない。原本で見開き２ページだったものが OmniPage では１ページとなっているので，どうしても NisusWriter に写してから，ページ分かれのチェック，行末の単語分かれの回復が必要だ。

ページは挿入メニューから改頁マークを打っていけばいいが，ミスが確認しやすいように改行マーク，改頁マークなどは画面に表示させておくほうがいい（メニューバー→挿入→改頁／メニューバー→ツール→表示→表示オプション→スペース・タブ・改行をチェック）。

単語を二つに切り離しているハイフンは削除しなければならないので，検索／置換を使い，「-」を「」（ブランク）に置き換える。

韻文テクストなどで，索引にページではなく行を表示させたい場合は１ページ１行のレイアウトにすればいい[2)]。

2.5. コンコーダンス作成

多機能ワープロならたいてい索引作成機能，検索／置換機能を持っている。ただし，テクスト上のすべての単語を索引に載せようとすると，すべての単語一つずつに**索引に載せる**ための処理（索引記載マーク付け）をしなければならない。NisusWriter が他のワープロより便利なのは，独特の強力検索機能を持っているか

らであり，これを使えばこの特殊なマーク付けが簡単に行える。

索引作成では対象テクストを特殊なマーク付きのものに作り替えることになるので, 元になる NisusWriter テクストはオリジナルとでも名前を付けて新規保存しておいたほうがいい。データ消失の可能性が常にあるので，コンピュータ上の作業はコピーを取ってから行うのが鉄則だ。

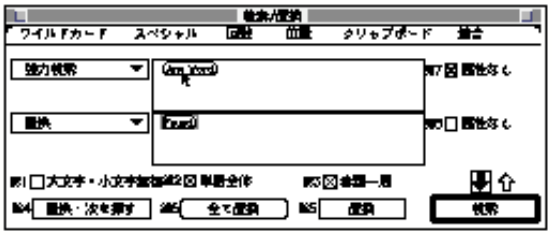


図5

作業は，まずツールメニューから検索／置換を選び，検索ウィンドウが現れたら，通常検索ではなく**強力検索**を選ぶ。そして，英字入力（メニューバーの右端から二つ目のアイコンで切り替える）になっていることを確認し，ワイルドカードから**任意の単語**を選ぶと検索覧に **AnyWord** と書き込まれる。次に，**tab** キーを打つか，ポインター（マウスの動きに応じる矢印）を置換ウィンドウまで持っていきクリックし，入力ポイントをそこで点滅させる。そして，「適合**→適合語**」を選ぶと，置換ウィンドウには **Found** と現れるはずだ。この **Found** を反転させておいて，「ツール→索引**→索引に載せる**」を選ぶと， **Found** にグレーの枠が付く。この枠が索引記載マークである。後は，置換覧の「属性のある／なし」，「大文字・小文字を無視する／しない」，「単語全体／全体でない」，「書類一周する／しない」をチェックする。これで, **全て置換**をクリックすれば, テクスト上のすべての単語に索引記載マークが付けられる。

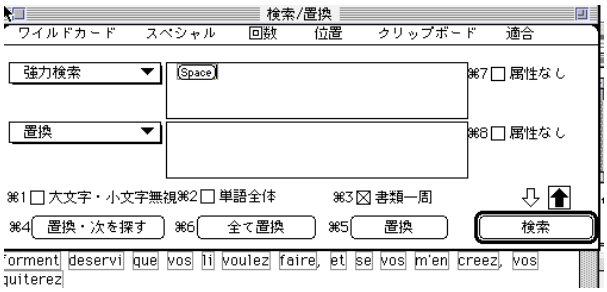

図6

この作業はコンピュータにとってかなり負担のかかるものなので，コンピュータの機種によっては１冊分のテクスト処理に数時間を要する。またメモリー不足からエラーが出たり，フリーズしてしまうこともある。このような時はいったんアプリケーションを終了し， Nisus Writer のアイコンをクリックで反転させ，「ファイル→**情報**」を選ぶ。すると，情報ウィンドウが現れるので, メモリーの使用サイズをテクストのページ数にもよるが, 搭載メモリーに余裕があれば 10**M**(10000**k**)以上割り当てておく（メモリーに余裕がない場合は，処理時間が相当長くなるのを覚悟で仮想メモリー[3] を使うか，テクスト１冊分まとめて処理しようとせず，５分の１ずつぐらいに分割し，索引記載マークを張り付けた後で一つにまとめればいい）。

場合によっては，ここで単語の枠付けが不規則にスペース含みとなり，それが原因で単語どうしの癒着が起こることがある。そんな時は，索引記載マークを付ける前に，検索／置換でスペシャル→スペース**→Space** を **SpaceSpace** （ダブ

ル・スペース）と入れ替え，単語どうしを引き離しておくか，索引記載マークを付ける時に,置換欄の枠付きの **Found** の前にマークなしの **Space** を入れておく。

また，単語の癒着はなくても，語末になぜかスペースが一つか二つ，時には三つ付いてしまう時がある。これはスペル間違いではないのだが， **Index** を作ってみると，尻尾にスペースがないものとあるものとは項目としては別になるので，取り除いておかねばならない（索引記載マーク付きの **Space** をブランクに置き換える）。



また，アポストロフは単語の切れ目として登録されていないので，「' 」を「'**Space**」に置き換える（アポストロフはデザインの違うものが２種類あるのでこれも検索／置換を使って統一する）。

以上の手順で索引記載マークの付いたテクストができていれば，後は，ツール**→索引→索引作成**を選ぶだけである。できあがった索引にミスが見つかればその語彙をテクスト（索引記載マーク付き）上で検索／置換で修正し，その修正後テクストからより正確な索引を作ることが望まれる。

3. NisusWriter の活用

高機能ワープロが多く出てきた現在， NisusWriter でなければできないようなことはあまりなくなってきた。欧文処理では最も優れているとの評判が定着してはいるが, 他のワープロソフトではできない作業に NisusWriter を使っている例をそう聞かない。

NisusWriter にしか搭載されていない特有の機能は，強力検索とマクロ編集だが，研究の下作業に使える便利なこれらの機能の具体的な使用例がもっと報告され，多くの人に利用されることが望まれる。

3.1. マクロ編集

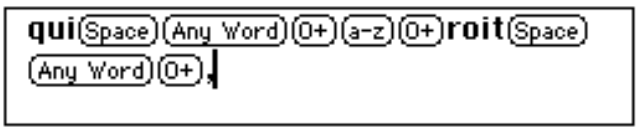
これは一連の作業手順を音声録音のような手軽さで記録し，自作のコンピュータ機能として登録するものである。例えば，欧文文字を固有の配列で使っている富士通オアシス上で作成したフランス語文書を MS-DOS ファイル経由で Macintosh 上に展開した場合，特殊文字が固有の文字化けを起こすのだが，検索･置換を使ってこれを一種ずつ正していく作業を登録しておけば，「オアシス／ Macintosh 変換」という処理メニューを NisusWriter につけ加えることができる。

3.2. 強力検索

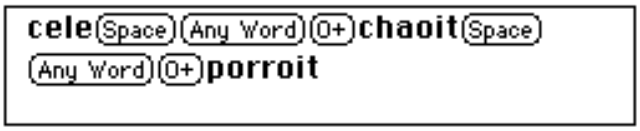
語彙，文字の単純な検索･置換ならたいていのワープロでできるが， NisusWriter では以下のようなメニューを組み合わせることでかなり複雑な条件での検索が可能だ。

（入力モードを欧文にすると，これらのメニューは英語で検索ウィンドウに書き込まれる。）

条件の組立方としては，例えば，「qui と条件法-roit 語尾の動詞を含む句」を拾いたければ，条件を次のようにする。



また，特定の単語を３つ含む文を拾いたい，例えば， cele/chaoit/porroit を含む文なら，次のようにする。



« avec lui, si s’entremirent molt de savoir por quoi cele tor chaoit et coment ele porroit tenir. Cil .VII. furent molt saige de cel art et molt avertuerent lor sen »の下線部が拾える。

まずはこの検索条件の組み立てを変形し，試みていただきたい。（了）

注

[1] 作業中何度も繰り返す作業はマウスを使うよりキーボード･ショートカット（キーの組み合わせによるコマンド）が便利なので、本文中に示したものぐらいは最小限覚えた方がいい。

[2] NisusWriter のマニュアルを参照。

[3] Macintosh のマニュアルを参照。