第 1 章

人文学のためのテキストデータの構築とは

永崎研宣

1．テキストデータベース構築に関する概況

テキストデータベースを作る、という取組みは、テキスト研究をしているとどうしても関心を持たざるを得ない。現在、テキストは、Unicode などの文字コードに準拠して文字を並べていけば高度な処理が比較的容易に可能となるが、作り方次第で後にできることがいろいろ変わってくる。したがって、貴重な時間や人的リソースを費やすことになるテキストデータベース構築をどのように実施するかということは、テキスト研究にあたっては重要な関心事となる。もちろん、Unicode などが出てくる以前から、いろいろなローカルな文字コードを駆使したテキストデータベース構築は行われてきており、日本でも 1980 年代にはすでにテキスト・データベース研究会等で活発な活動があったようである。

近年は、テキストデータと言えば、MS ワードや一太郎、LibreOffice、Google Docs 等の文書、エクセルやパワーポイント、Google や LibreOffice の同種のソフトで作られたデータのテキスト部分、日夜大量に書き込まれ続ける SNS やブログ記事など、いわゆるボーンデジタルのテキストデータが毎日大量に作成されており、それらをおおまかに処理するだけでも有用な分析がさまざまに実施できるだろう。ePub 等で販売されている電子書籍のデータもあり、テレビの字幕データや、他にもいろいろ有用そうなものがある。こういったデータを対象とした分析については、技術的には比較的容易であり、また、テキストの内容を人が読まずにコンピュータが分析するだけであれば著作権法において利用が認められるようになった（詳しくは、コラム「著作権法改正で Google Books のような検索サイトを作れるようになる？」を参照されたい）。しかしながら、分析に使用したデータを広く共有することは困難であり、その点に課題がある。

ボーンデジタルなテキストが技術的には使いやすく、権利関係において困難さを抱える一方で、そうでないテキストの場合には、著作権保護期間が終了しているものといないもの、著作権の状態が不明なもの、の 3 種類に分けて扱うことになる。

著作権保護期間が終了しているものは、パブリックドメインということになり、基本的に誰もが自由に利用できる。しかしながら、著作権保護期間が終了しているかどうかの確認は、権利者

の没年情報が必要になる。そのため、著名人の場合は比較的確認がしやすいものの、そうでない人は確認が難しいことが多い。また、著名人であっても没年の確認が難しいこともある。没年が不明な場合、著作権がどうなっているかわからないもの、という扱いになり、権利者不明作品、オーファンワークス、などと呼ばれる。オーファンワークスは、安全な利用を心がけるなら、著作権保護期間中のものと同様に扱うことになる。

基本的に、著作権保護期間が終了しているものは自由に使えるため、それをテキストデータを活用した教育や研究のベンチマークとして扱うのが一つのわかりやすく有用な道だろう。これを通じて明らかにできたさまざまな活用方法を、ボーンデジタルテキストの大規模な分析に活用することで、より深い社会文化研究にもつなげられる可能性がある。

とはいえ、著作権保護期間が終了しているものには、現代的な日本語で書かれているものはあまり多くない。しかも、少し時代をさかのぼると古文やくずし字が多いため、テキストデータベース作りにおける難易度は高くならざるを得ない上に、そのテキストの分析手法をそのまま現代に適用することもやや難しい。

比較的新しいテキストに焦点をあてた場合には、そういった問題はある程度回避できるかもしれない。明治中期〜昭和初期あたりであれば、2021 年度末、国立国会図書館の次世代デジタルライブラリーにおいて、OCR によるテキストデータ化に基づく全文検索システムが公開され、この時代のテキストの OCR 技術もかなり進んでいることが明らかになった。この成果を用いることで、これまでよりも一歩進んだ取組みが可能となるだろう。

なお、古い日本語の分析手法としては、すでに国立国語研究所により古い日本語を形態素解析するための辞書 UniDic が時代ごとに公開されており、それぞれの時代の日本語文の形態素解析がある程度は可能となっている。また、情報処理学会人文科学とコンピュータ研究会等の関連学会では江戸時代以前のテキストデータを対象とした固有表現抽出やトピックモデリングなどに関する発表が複数の研究グループにより着々と行われており、古い日本語のテキストデータの分析手法も、まったくできないわけではない。ただ、これはまだ研究段階であり、しかし研究段階だから面白いということでもある。

また、古い日本語テキストの場合、OCR の精度が低いという問題があり、テキストデータベース作りの難関の一つだったが、最近は、くずし字 OCR やクラウドソーシング翻刻など、自動文字読み取りという方向や人力作業の輪を広げていく方向など、現在の技術水準で可能なことが徐々にこの領域にも展開されるようになってきている。テキストデータベース構築を進めていくためには、こういった技術や枠組みを活かすことが今後は重要になっていくだろう。

2．元資料とテキストデータの整合性

ボーンデジタルなものには生じないが、デジタル以外の媒体に基づくテキストデータには大きな課題がある。それは、「元の媒体上でのテキストと完全に同じではない」ということだ。

まず、字形や文字の大きさが完全に同じになることはなかなか容易ではない。内容の相違ではないことが多いため、問題になることはあまりない。しかし、読み手側が受ける印象には少し違いが出てくる場合もあるだろう。近年の資料であれば、目が悪い人向けに少し大きな文字にしているかどうか、あるいは、ディスレクシア向けにユニバーサルデザイン（UD）フォントを使っているかどうか、ということも、やはり読み手を意識した時にはやや大きな違いとなるだろう。古い資料であっても、くずし字であればまったく同じ字形を再現することは困難であり、漢字であっても時代が異なれば字体も字形もさまざまである。テキストの内容を分析する場合でも、テキストに対して読み手がどう考えたかということが研究に含まれるのであれば、そのあたりの差異も関わってくることがあるかもしれない。さらに言えば、石に彫られたものと紙に印刷されたものとでは受ける印象は大きく異なると思われるが、石から文字起しされたプレーンなテキストデータにはそういった相違も特に反映されることはない。

また、字体の違いはかなり大きな問題になることもある。旧字体と新字体を丸めてしまうことは現在も比較的よく行われるようだが、その中には、著者は異なる文字として使い分けた二つの字を丸めてしまうということもあるかもしれない。できることなら、その使い分けにどういう意味があるかを判断するのは、それを読み分析する側でありたい。しかしながら、たとえば外注として企業に文字起こしを依頼する場合などは、あらかじめ JIS の第〇水準の文字で、という風に仕様書で限定をかけることになるので、どうしても丸めざるを得ないことになる。

ここまでは漢字の話だが､仮名も丸めたりいろいろなことをする場合がある。仮名については、たとえば源氏物語の研究でよく用いられてきたテキストの一つである『校異源氏物語』では字母の違う変体仮名を現代のひらがなに丸めてしまっている。一方で、近年は字母の違いを対象にした源氏物語研究も行われており、この点は今後大きな課題の一つになっていくかもしれない。

このようなルールベースでの問題とは別に、単なる誤転記という問題もある。これは、最終的には人力でチェックするしかないので、正確性を期すなら非常に難しい。むしろ、「少し間違っていても利用可能なもの」として流通させ利用することを考えるべきかもしれない。そもそも、紙媒体でも誤植が混入することはしばしば生じるのであり、デジタルだけの問題でもないと考えることもできるかもしれない。

3．元資料との関係をどう位置づけるか

ここまでみてきたことに加えて、あらゆる面において元資料とまったく同じものをテキストデータで得ようとするのは不可能である、ということも改めて確認しておきたい。たとえば、テキストが書かれた紙や石といった元資料の任意の箇所の化学的組成を確認したいと思ったら、それを行うことが技術的には一定程度可能だったとしても、現在のストレージではデータ量が膨大になってしまうので実際にはほとんど不可能である。また、字形の微細な違いもやはり再現不可能なことがあり、さらに、Unicode で符号化されていない文字であれば自分の PC や Web ページ

では外字で対応するとしても、一般に広く簡単に利用することは難しい。そういった事柄をすべて解消できたとして、その次に来るのが、誤転記問題ということになる。

というわけで、まず、上記のような状況を斟酌しつつ、「テキストデータで何をどこまで再現したいか」ということを検討する必要がある。これは「どこまで手間暇（コスト）を費やせるか」ということでもある。このようなことになると、その時点での技術水準による制約も大きく関わってくるため、かつては 10 年単位で長く通用する具体的な対応策を立てることは難しかったが、近年は徐々に安定してきているように思われる。それについて以下にみてみよう。

3-1．文字が Unicode に入ってない場合

まず、使いたい文字がそもそも Unicode の文字コード表に入っていない、という問題がいまだに時折聞かれる。この場合、文字を表示せずに「〓」等で済ませるか、当該文字の文字画像かフォントを作って対応するか、後述する XML を使って記述するか、あるいは、Unicode で使えるようにすべく符号化提案してしまうか、という選択肢になる。Unicode に文字が登録されれば、その文字を用いるすべてのテキストが普通のテキストデータとして記述・処理できるようになるため、その文字を扱うすべての研究分野にとって非常に有益である。ただ、この場合、国際標準化機構等による ISO/IEC 10646 に文字の符号化提案をするという時間のかかる手続きを行うことになる。学術研究のために文字を符号化提案するためのルートはいくつか存在するが、いずれにしても、国際標準化機構の委員会での英文文書の交換に基づく議論に参加して、自らが必要とする文字を符号化する必要性を、そこでのルールに従った英文文書で提示しなければならない。これは、はやくても数年を要するプロセスであり、提案書や登録のための議論にもその都度時間をかけて対応しなければならない。

欧米の資料だとアルファベットだけで済むから楽だという話が聞かれることがあるが、中世の資料では字種が多様に存在し、Unicode では表現できない外字もまだ残されていることから、Medieval Unicode Font Initiative が Unicode への外字登録を目指した活動を続けている模様である。Unicode への文字の登録に関しては、近年、コンピュータの処理性能の大幅な向上に伴い、古典籍・古文書等に登場する学術用途でしか使われないような文字・文字体系も積極的に登録されるようになっている。手続きとしては、まず国際標準規格である ISO/IEC 10646 への追加が承認されてから Unicode 規格もそれに追従することになっており、新しい文字の追加は、ISO/IEC の規格への登録という形をとることになる。カリフォルニア大学バークレー校を拠点とする SEI（Script Encoding Initiative）という団体がこの動きを幅広くサポートしている。漢字の登録に関しては、IRG（Ideographic Research Group）という漢字検討の専門グループがいったん検討した上で ISO のワーキンググループ、ISO/IEC JTC1/SC2/WG2 に提案するという手順を踏むことになっている。したがって、漢字を登録する場合には、まずは IRG に提案しなければならないのが現状である。ただし、IRG も近年は学術用途の漢字登録に寛容になっており、文字同定や証拠資料に関する所定のルールを踏まえた上で要登録文字であると判断されれば基本的には登

録されるようになっている。

あるいは、現在は Unicode に入っていなくても、現在、符号化提案中となっている場合もある。符号化に関する議論の過程は、議論のための文書がすべて Web で公開されているため、符号化提案をする前に一度確認してみるとよいだろう。漢字に関しては IRG のサイト [1]、それ以外の文字に関しては ISO SC2/WG2 の文書リポジトリ [2] として公開されている。漢字は数年ごとに数千字が提案・議論されており、それ以外の文字に関しては、提案されたもののペンディングになっているものも多く、自分が使いたい文字がそこに含まれている場合は、提案者に連絡をとって符号化に向けて議論を進めるべく協力するという方法もあるだろう。

あるいは、コストを勘案した結果、Unicode での文字の符号化提案は諦めて、似た文字を使うとか、あるいは「外字」として扱うという方向性も考える必要はあるだろう。ただ、そのようにした場合、その文字が「本来はどういう文字であるか」を示すことがテキストデータだけではなかなか難しい。その状況を改善する手段の一つとして、人文学テキスト資料のための XML に準拠した記述手法を提示する TEI（Text Encoding Initiative）ガイドラインの gaiji モジュール [3] がある。ここでは、文字についてのさまざまな情報を記述した上で、本文中に記した文字に対してその文字情報を付与できる枠組みとなっている。

一方で、データとしての互換性は少し落としつつも字形をうまく表示したいという場合には、外字フォントを作成して表示させるという方法もある。フォントを作成すれば、文字を拡大・縮小した場合にもきれいに表示でき、Web ページでは Web フォントを用いることでいちいち専用のフォントをダウンロードしなくても用意したフォントを自動的に利用して表示できるため、ある程度の利便性は確保できる。フォントの作成は Glyphwiki を用いれば、やや時間はかかるものの比較的容易である。この場合、文字コードとしては、Unicode の Private Use Area（PUA、私用領域）と呼ばれる独自文字コード用の文字コード領域を用いてテキストデータを記述することになる。そのため、テキストのコピー＆ペーストをした際には同じフォントを用いなければ文字化けしてしまうことになり、また、同じ PUA の文字コード割り当てルールを用いたテキストデータとしか互換性を保てないため、作ったテキストデータを余所で作成されたテキストデータとあわせて幅広く利用しようとする場合にはかなり使いにくくなってしまう。つまり、独自フォントと独自文字割り当てルールを常にテキストデータと一緒に流通させなければならないということになる。そして、ここで設定した文字コード割り当てルールが失われてしまった場合、何が書いてあったのかわからなくなってしまうという難しさもある。この方法を選ぶ場合には、これらの点を踏まえた対応が必要になるだろう。

また、今昔文字鏡や GT 書体等のいわゆる多漢字フォントにおいては、複数のフォントファイルを切り替えることで同じ文字コードに複数の文字を割り当てて多数の文字の表示を実現しているため、文字に対するフォントの情報が失われるとどの文字だったわからなくなってしまうという難しさがある。この場合、テキストデータだけでは文字の情報を残すことができず、ワープロソフト等のフォントの情報を残せるソフトウェアが必須であり、他のソフトウェアにデータを移

管する際にもフォントの情報が失われないようにする必要がある。今昔文字鏡はさまざまな漢字を必要とする研究者の間で広く使われた時期があったものの、上記のような難しさに加えて利用条件の扱いの難しさもあり、利用の際には十分な注意が必要であり、また、すでに作成されたデータを維持したり利用したりする際にも上記のような事情によく留意する必要がある。

外字フォントよりもさらに簡便な方法として、文字の形を表現するために文字画像を用いるという方法もある。この場合、データの扱い方としては外字フォントと同じであり、やはり、外字に番号を割り当て、それに対応する字形を記した表を作成しておく必要がある。Web ページで表示する場合には外字フォントよりも仕組みが簡単だが、ワープロソフト等に貼り付ける場合には不便であり、Web 公開を前提としない場合にはあまりおすすめできない。

なお、「〓」の利用を除くいずれの場合にも、必要な文字に関しては、独自の文字コード割り当てルールとそれに対応する字形の対応表を作成し、維持していく必要がある。そして、それぞれの文字についての周辺情報を記録しておくことが望ましい。この対応表は、文字が少なければそれほどの手間はかからないが、数が増えると、作業者間での最新の対応表の共有や対応表に記載するにあたってのルールの設定など、相当の準備が必要となる。たとえば、SAT 大蔵経デー

表 1-3-1　使いたい文字が Unicode のコード表に入っていない場合の対応の例

<table>
<tr><th></th><th>一律代替文字表記</th><th>文字画像表示</th><th>外字フォント利用</th><th>XML 注記</th><th>Unicode 符号化</th></tr>
<tr><td rowspan="3">メリット</td><td rowspan="2">入力時には時間も手間もかからない。</td><td rowspan="2">Web ブラウザ等では比較的うまく表示できる。</td><td rowspan="2">対応する環境が整えられればきれいに表示できる。</td><td>文字に関する情報を詳細に記述できる。</td><td>通常のテキストデータとしてのあらゆる恩恵を受けられる。</td></tr>
<tr><td colspan="2">テキストデータとしての持続可能性が高い。</td></tr>
<tr><td colspan="4">字形が具体的にわかる。</td></tr>
<tr><td colspan="5">そこに何らかの文字があったことはわかる。</td></tr>
<tr><td rowspan="5">デメリット</td><td rowspan="5">どんな文字かわからない。
検索ができない。</td><td>画像表示機能が必要。</td><td>他の外字フォントとの共存が困難。</td><td rowspan="2">XML を処理する必要がある。</td><td rowspan="4">手続きに数年間の大きな労力を要する。</td></tr>
<tr><td colspan="2">テキストデータ単体では文字情報を伝えられない。</td></tr>
<tr><td colspan="3">検索が難しい。</td></tr>
<tr><td colspan="3">独自の文字割り当てルールを策定し独自に配布し続ける必要がある。</td></tr>
<tr><td colspan="4">用意に手間がかかる。</td></tr>
<tr><td>具体的な手法の例</td><td>「〓」等で済ませる。</td><td>文字画像を貼り込む。</td><td>外字フォントを作成・利用（多漢字フォントの場合も同様）。</td><td>TEI ガイドライン等に準拠して記述。</td><td>Unicode に文字として登録。</td></tr>
</table>

タベース研究会では 1 万字を超える外字の対応表を作成・維持しており、これは共同作業可能な Web データベースとして運用されている。また、Unicode 登録のための符号化提案を念頭に置いている場合には、提案書提出時に文字の意味や登場箇所、複数の利用例を撮影したデジタル画像等が求められるため、対応表を作成する段階から提案書に必要とされる情報を踏まえて文字情報を集積しておくとよいだろう。

このような諸々の手間を省くための一つの方法として、Unicode に含まれる似た文字に置き換えてしまうという方法と、単に「〓」を置いてしまうという方法がある。前者はどれくらい意味内容に影響するかについての検討が必要だが、現実的な選択肢である。どのような置き換えを行ったか、という対応表を作成し用意しておければなおよいだろう。これについては次節も関わってくるので参照されたい。また、「〓」は最終手段ではあるが、後生に託すということで、これも一つの選択肢と考えるべきだろう。

このような、いわゆる外字の扱い方のメリット・デメリットを整理した表を【表 1-3-1】としておおまかにまとめたので参照されたい。

3-2．字形・字体の相違をどう扱うか

外字の扱い方をルール化できたなら、次は全体としてどのような方針で文字をテキスト化するかを決めることになる。あるいは、先にこの全体方針を決めた上で外字の扱い方を決めてもよいだろう。

字形の微細な違いに関しては、現在、技術的には、Unicode が提供する IVS（Ideographic Variation Sequenice）の仕組みを用いることでかなりの程度対応できる。IVS は、Unicode で登録されている文字との微妙な字形の差を枝番号で識別する仕組みであり、この仕組みに準拠して字形と枝番号を登録するデータベース、IVD（Ideographic Variation Database）のなかに目当てのものが用意されている場合もある。IVS の仕組みは MacOS や最近の MS-Windows でも標準装備されており、この場合、IVD のリストを確認するコストのみで済み、容易な検索ができるシステムも提供されているため、比較的現実的である。

一方、IVD に適切な字形を見つけられない場合に、それでも可用性を確保するために Unicode の枠組みで適切な字形を表示することにこだわるのであれば、IVD を Unicode に提案して自分のテキストに必要な字形を Unicode に登録してしまうという手もある。この場合、フォントも自分で作成しなければならないが、それに関しては Glyphwiki を使えば比較的簡単である。ただし、Unicode の IVD に自らのコード表を登録するために、提案書作成を始めとする手続きが必要であり、この IVD への字形の登録は、Unicode に文字を登録することに比べると比較的容易ではあるが、やはり一定の時間を要する。こうしたコストをかけるべきかどうか、ということは要検討事項となる。

このように、技術的にも手続き的にもいろいろなことが可能になっているが、そうすると、あとは、「どこまでコストを費やせるか」という問題になってしまう。字形の微細な違いをきちん

と反映しようと思った場合、「標準的な字形とは微細な違いのあるこの字形はこの資料の中に登場するすべての同字で共通なのか」を確認しなければ、意味が薄くなってしまう。それは人力で確認するのか、あるいは、画像処理プログラミングの得意な人であれば（あるいはそういう人に頼めるのであれば）、元資料をデジタル撮影／スキャンして文字を切り出して分類し、自動的／半自動的に確認するか。人力の場合には、文字入力、もしくはOCRの誤字チェックをしながら同時にチェックしていくことになるだろうか。いずれにしても、簡単にできることではなさそうである。そして、その微細な字形が既存のフォントでは表現できていない場合は、さらに、上記のように、IVDに登録する手続きを行うかどうかを検討することになる。

というようなことを踏まえて、字形の微細な違いに手間暇をかけられないと思ったなら、それは諦めることになる。ただ、諦めるにしても、「この字形はこの文字と同じとみなす」というようなルールの設定は必要になることが少なくない。それを簡素にするためには、既存の文字コードを活用することが有用である。「Unicodeの範囲で」「JIS第3水準までで」「新字体で」などというように、文字の範囲を決めておきつつ、そこから外れるものについては対応表を作ってデータ入力を行うことになる。

この対応表が大きくなると、いちいち探す羽目になってかえって入力作業が大変になってしまうこともあり、対応表など使わずにUnicodeで直接探した方がはやい、という入力者・企業もいるので、元資料の字体・字形の状況や入力担当社のスキル等の案配にも注意が必要である。

表1-3-2　テキスト化する文字の範囲と意義のおおまかな目安

文字の範囲	メリット	デメリット	便利な道具立ての例
IVS/IVDでの対応	既存の文字と同様に使える。 文字の意味を考えず字形で探せば済む。 対応可能な字形は非常に増える。	対応文字を探すのに少し手間が増える。 対応フォントが必要。 それでも字形がみつからない場合がある。	異体字セレクタセレクタ、CHISE、Web font、Glyphwiki
Unicodeでの対応	既存の文字と同様に使える。	対応フォントが必要な場合もある。	CHISE、Unihanデータベース、花園明朝フォント、Glyphwiki
JIS第n水準（nは、通常は2〜4が多い）	入力文字種を減らせる。 簡易な文字列検索には便利。	入力時に対応字を探しにくい場合がある。 入力できない文字が生じる可能性が少し高まる。	MJ縮退マップ、CHISE、異体字データベース
新字体を用いつつ何らかの文字の範囲を設定	入力文字種を減らせる。 簡易な文字列検索には便利。 入力できる文字は表示にも困らない。	入力時に対応字を探しにくい場合がある。 入力できない文字が生じる可能性がやや高まる。 元の資料の字とかなり変わってしまうことがある。	MJ縮退マップ、CHISE、異体字データベース

なお、検索に際しては、異体字を同時に検索する仕組みを作成することも可能であり、また、後からどちらかの字体に変換することも容易にできるため、旧字体・新字体くらいの違いであれば、元の資料に即した字体で入力しておくと、後々、手戻りの可能性を少なくすることはできる。

以上のようなことを踏まえつつ、テキストデータを作成する際の文字の範囲についてのおおまかな目安を【表 1-3-2】としてまとめておいたので参照されたい。

3-3．文字の扱い方を記録しておく

文字の扱い方のルールを決めてそれに従ってテキストデータを作成したなら、それを可能な限り文書として残しておくことが重要である。「なぜこの文字が出てくるのか？」「この文字とこの文字はどういう関係なのか？」等々、ただテキストデータ化しただけで、元資料からどのようなルールで文字起しされたかが明示的に記述されていないと、他の人がそのテキストデータを使おうとした時に、よくわからないまま使うことになってしまう。誤字なのか、意図的にそうしたのか、読者・利用者が判断に迷うような場合には、そのルールが提供されていることで、適切な利用が可能となる。すなわち、そのような情報が提供されていないと、データの信頼性が低いとみなされて使われなくなってしまうことも十分にあり得る。データ作成者でない人が見たときに理解し利用できるようにデータを作っておくというのは、デジタルデータの長期保存に関する枠組みとして有名な OAIS 参照モデルでも提示されているような、教科書的な事柄である。これは、テキストファイルの冒頭に書き込んだり、テキストデータを zip 等で配布する際に説明文書として同梱しておく、といった方法がある。あるいは、TEI（Text Encoding Initiative）ガイドラインがそういった情報を記述しておくためのエレメントを提供しており、これを利用すると、そのあたりの処理の利便性を高められる。興味がある人は、特に The TEI Header の章[4]を参照されたい。この章は日本語訳も公開されている[5]ため、比較的読みやすいだろう。

3-4．誤転記を含むテキストの扱い

文字をデジタルに転記する方法に関しては、このようにして進めていくことができるが、次に出てくるのは、誤転記である。人が手で入力する場合には、コンピュータによる自動処理よりは正確なことが多いが、それでも入力ミスを完全に排除することはできない。コンピュータに OCR や HTR（Handwritten Text Recognition）で読み込ませても文字認識が完璧にできるとは限らない。それを人の目で修正しても、やはり間違いが残ってしまうこともあるだろう。そのようなテキストデータを用いて分析することで何か意味のある処理が可能なのか、という問題が生じてくる。この場合、テキストデータの量や処理の内容、つまり、統計的に分析するのか、単にテキスト検索ができればいいのか、テキスト検索の結果をコピー＆ペーストしてそのまま使えるようにしたいのか、といったことによって話が変わってくる。そして、もちろん、手間暇をかけるほど、正確性の高いデータを得られることは間違いなく、しかし、手間暇をかけられない場合には、何をどこまで妥協するのか、ということになる。

この場合には、やや難しい判断が必要になるが、基本的には二つの方向性を考えておけばよいだろう。一つは、なるべく正確なものを作成する方向であり、もう一つは、正確でなくてもいいから分析できるように分析の仕方を考えるという方向である。そして、どちらの方にどれくらい重きを置くか、ということを決めるのが、この局面で必要となる。

正確なものを作成する方向は、単に、全体として完全に正しいものを目指すというだけでなく、たとえば、特定の情報がほしいだけであれば、その部分だけは正確性が高まるように、人力と機械をうまく組み合わせてチェックをしたり、あるいは人力だけで根性で頑張るという手もあるだろう。

一方、正確でなくてもいいから分析できるように分析の仕方を考えるという方向については、特にデータ量が大きい場合には比較的有効だろう。単なるテキスト検索にしても、少々誤転記が多くても、大量にテキストデータがあれば、それなりに欲しい情報もヒットしてくれることがあるだろう。また、統計的に分析するにしても、統計的に有意であることを示すことが目的であれば、大量テキストデータでは多少データに誤転記が含まれていてもなんとかなることもあるだろう。この場合、対象となる大量テキストデータの信頼度がどれくらいかということもサンプル調査などをして明らかにしておけば説得力は増すかもしれない。

3-5．テキストデータ構築の深さ

前節を踏まえると、テキストデータの元資料への忠実さや付与される解釈の深さは、以下の2点に依拠するということになる。

・どのような人のどのようなニーズを主要な対象とするか
・どれくらいの手間暇をかけられるか

この2点を明確に定めたうえでテキストデータを作成すれば、目指すことはおおむね実現できるだろう。ただし、大規模なテキストになると、テキストデータを作成し始める段階では、手間暇の見通しを立てるのは難しいことも多い。そのような場合には、本格的な作業に入る前に、対象となる元資料の典型的な箇所をいくつかサンプル的に取りだしてテキストデータ化し、それを通じて全体にかかる手間暇を算定するのが穏当なやり方である。

どのような人のどのようなニーズを主要な対象とするか、というのは、テキストデータ作成者の立場によって大きく異なる。対象テキストを自分（たち）で研究したい人（たち）が作成するのであれば、それらの点は明確にしやすい。自分たちの研究のニーズに沿ったデータを作成すべく、自らの方法論を深めていけばよいということになるからである。それもまた、突き詰めるとなかなか難しいことにはなるものの、目的地点を定めやすく、さらに、それを追究すること自体がデジタル・ヒューマニティーズ分野においては研究発表にもつながり得るため、このような仕事に従事している場合には、研究活動の一環として位置づけることも可能である。

一方、図書館等のサービス提供者としてテキストデータを作成・提供しようという場合、その決定の仕方はやや難しいことになる。むしろ、手間暇（もしくは費用）をどれくらいかけられるか、

ということが前面に出てきて、それに応じて対象者やニーズを考える、という順番になることもあるだろう。大規模なところで見てみるなら、主に米国の大学図書館が共同運用する HathiTrust にしても、国立国会図書館の次世代デジタルライブラリーにしても、大規模な OCR テキストの全文検索システムを提供しているが、いずれも、基本的には、かけられるコストを踏まえて現在可能なものを作成し提供している。このような場合には、むしろ、利用者側がその有効活用の方法を考えることが発展的であり重要であると言えるだろう。

このような観点から有用なアプローチがあるので紹介しておきたい。前出の TEI ガイドラインを定めている TEI 協会の図書館分科会[6]が提供している Best Practices for TEI in Libraries[7]というルールがある。ここでは、テキストデータへのタグ付け（符号化、encoding）のレベルを以下のように 5 段階に分けて整理している。

> Level 1: OCR によって自動生成されたテキストにそのまま自動化可能な範囲でタグ付け
> Level 2: 最小限のテキストの構造をタグ付け
> Level 3: 内容に関するごく簡単な整理も含むタグ付け
> Level 4: 内容に関する基本的な整理・分析を含むタグ付け
> Level 5: 学術編集のためのタグ付け

このように整理した上で、それぞれのレベルで推奨されるタグ・オプション的なタグも提示している。このようなルールが公開されている場合、これらのいずれのレベルに準拠したか、ということさえ明示しておけば、利用者側がどう使えばいいかということを判断しやすくなるだろう。Level 1 のテキストであれば、利用者は、テキストの文字読み取りからして間違っているかもしれないという前提でテキストデータを扱うことができる。あるいは、Level 3 のテキストであれば、段落や章タイトルなどの基本的な構造がテキストデータに埋め込まれており、それを前提とした処理ができることになる。

ここからは前節と接続する話になるが、テキストデータ作成の際に準拠したレベルについての情報が、この「Best Practices for TEI in Libraries」とともに示されていれば、利用者がデータを活用する際に大いに参考になるだろう。それをテキストデータのファイル内に書き込もうとするなら、TEI ガイドラインに準拠したテキストデータの場合には、`<teiHeader>` の中に配置可能な `<editorialDecl>` というエレメントに書き込むことができる。TEI ガイドラインに準拠してテキストデータを作成しておけば、TEI に準拠したさまざまなツールで活用でき、有用性を高め

```
 9    <LUW B="B" SL="v" l_lemma="原動力" l_lForm="ゲンドウリョク" l_wType="漢" l_pos="名
10        <SUW orderID="110" lemmaID="11737" lemma="原動" lForm="ゲンドウ" wType="漢" p
11        <SUW orderID="120" lemmaID="40327" lemma="力" lForm="リョク" wType="漢" pos="
12    </LUW>
```

図 1　BCCWJ における「原動力」へのマークアップの例

ることができる点も考慮するとよいだろう。

この件の解は、TEI に準拠するだけでなく、他にも、別のデータモデルを作ってみたり、それを RDF で書いてみたりすることも可能ではあるので、余裕があればいろいろな選択肢について検討してみるという手もあるかもしれない。

3-6．学術編集のためのタグ付けについて

テキストデータベースが「どういう深さのものか」を決めて、それを記述するというのが前節の到達点である。しかしながら、前回は、研究志向の強いものについては、「Level 5: 学術編集のためのタグ付け」で一括されてしまっていた。「学術編集のための」と言っても、分野や手法によって関心はさまざまであり、それに応じた深さの方向性がある。これをどうするか、というのが次の問題である。

たとえば、言語学のなかには、単語の品詞情報や発音・アクセントなどの情報が欲しい人がいるだろう。その場合、各単語にタグがつけられて、そのタグによって本文中の単語に対する付帯情報を取り出せるようになっているとよいだろう。有名なものの一つに国立国語研究所が作成した現代日本語書き言葉均衡コーパス（BCCWJ）がある。ここで「原動力」という単語をみてみると以下のようにタグ付けされている。【図 1】

<LUW> というタグで始まり、そのタグに対して l_lForm=" ゲンドウリョク " l_pos=" 名詞 - 普通名詞 - 一般 " といった形で属性を与えることで、</LUW> というタグで終わるところまでの文字列に対して、現代日本語の分析に必要な情報を与えている。さらに、<LUW> の次に <SUW> というタグもあり、これが「原動」と「力」にそれぞれついている。これは短い単位の単語区切りということで、この短いものに対しても、やはり属性を通じて日本語分析に必要な情報が与えられている。このようにして、本文中の「原動力」という単語に対してタグを用いて研究に有用な情報を付与しているのである。

あるいは、古典文学作品の代表格である『源氏物語』について少しみてみよう。『源氏物語』には実に多様な研究のアプローチの仕方があるが、ここでは校異情報に関するタグ付けをみてみよう。

『源氏物語』と言えば、あまりにも有名なので、紫式部が著わした文章そのものが残っているのではないかとつい期待してしまうところだが、実際には、紫式部自身が書いたものは現存せず、それを写した写本の形式で日本各地に伝承されている。そして、写本はいつも完璧に複製できるわけではなく、むしろ誤記や表現の仕方の変化などによって内容が少しずつ変わってしまうこと

```
詞-普通名詞-一般" l_formBase="ゲンドウリョク">
ps="名詞-普通名詞-一般" formBase="ゲンドウ" pron="ゲンドー" start="150" end="170">原動</SUW>
接尾辞-名詞的-一般" formBase="リョク" pron="リョク" start="170" end="180">力</SUW>
```

もある。われわれが読んでいる『源氏物語』とは、そのようにして伝わってきた写本を並べて差異を確認し、どれが紫式部が書いたものにより近いかを是々非々で検討した上で作成されたものである。聖書にしても仏典にしても、原著者が書いたものが残っていない場合には、そのような差異を考慮する必要が出てくる。そこで登場するのが、「各写本でこの箇所はどう書かれているか」を記述できるようなタグ付け方法である。この種のものは、前出の TEI ガイドラインが得意であり、たとえば『校異源氏物語』のある箇所をタグ付けすると以下のようになる。【図 2】

```
253        <app>
254          <lem>はまして</lem>
255          <rdg wit="#別陽">なとは</rdg>
256          <rdg wit="#別國">なとまて</rdg>
257        </app>
```

図 2　『校異源氏物語』における校異情報のタグ付けの例

ここでは、本文中で校異情報（伝本間を比較して相違のある箇所の情報）が存在する箇所をまず <lem>～</lem> というタグで囲み、これに対して、校異情報を <rdg>～</rdg> というタグで囲んだ上で <rdg> タグには wit="# 別陽 " あるいは wit="# 別國 " と記載している。これは、一つ目の <rdg> は「別本の陽明本」における記述であることを示し、二つ目の <rdg> は「別本の國冬本」であることを示している。その上で、<app>～</app> というタグで囲むことで、ここにはこの <lem>（Lemma、ここでは本文を意味する）と <rdg>（Reading、ここでは異文を意味する）を含む校異情報（Critical Apparatus）が存在していることを示している。

言語学・文献学と、ややマニアックな方向に行ってしまったが、少し戻って考えてみると、そもそも例えば、テキスト中に登場する人名や地名などをタグ付けしておけば、むしろ、人文学に限らず、さまざまな研究分野、さらには研究外での用途も期待できるかもしれない。たとえば以下のものはそのようなタグ付けの典型的な例である。【図 3】

```
<persName corresp="#メロス">メロス</persName>は激怒した。
必ず、かの<persName corresp="#ディオニス">邪智暴虐の王</persName>
を除かなければならぬと決意した。
```

図 3　『走れメロス』における人名のタグ付けの例

ここでは、人名や呼称を <persName>～</persName> で囲み、それが実際にはどういう人物であるかについて corresp= という属性を用いて同定している。つまり、邪智暴虐の王がディオニスであることを記述しているのである。

さらに、人称代名詞が誰を指しているか、ということもタグ付けすれば分析の幅は広がるだろう。この場合には例えば以下のようになる。【図 4】

```
「<persName corresp="#メロス">メロス</persName>、
<rs corresp="#セリヌンティウス">私</rs>を殴れ。
同じくらい音高く<rs corresp="#セリヌンティウス">私</rs>の頬を殴れ。
私はこの三日の間、 たった一度だけ、ちらと
<rs corresp="#メロス">君</rs>を疑った。生れて、はじめて
<rs corresp="#メロス">君</rs>を疑った。
<rs corresp="#メロス">君</rs>が
<rs corresp="#セリヌンティウス">私</rs>を殴ってくれなければ、
私は君と抱擁できない。」
```

図 4 『走れメロス』における人称代名詞のタグ付けの例

ここでは、人称代名詞を <rs> 〜というタグで囲み、属性として corresp="# メロス " などと書いておくことで、誰が話題になっているか、ということを自動的に検出できることになる。

この「属性として corresp="# メロス " などと書いておく」ことも重要である。『走れメロス』ではフィクションの登場人物なのであまり問題にならないが、これが歴史文書等の実在の人物と結びつくものであったり、聖書や仏典などのように多数の書物で参照される人が登場する場合には、「あちらのテキストに登場する A さんとこちらのテキストに登場する AA さんは同じ人だがそちらのテキストに登場する A さんは名前が同じであるだけで別の人」という情報があると、いろいろな処理がしやすくなる上に人が読んだり参照したりする際にも便利である。そのような場合に、一人目の A さん、二人目の A さんの ID を何らかの形で決めておいて（たとえば personA-1、personA-2 など）、corresp="#personA-1" という風に属性をタグに記述しておくと、これは二人目の A さんではなく一人目の A さんであることが明示され、機械的な処理ができるようになる。ただし、これだけだと人がみてもわかりにくい場合もある。人が読んでもわかりやすいようにするためには、たとえば、「corresp="#personA-1" という属性のついたタグが付与された文字列（人物名や代名詞など）にマウスのカーソルをあわせれば「一人目の A さん」と記述されたポップアップが表示される」という風にしておくことや、あるいは、行間にそのような注釈を表示させる、といった対応もあり得る。ここで重要なのは、「A さん」という名前を発見することはコンピュータの文字列検索で簡単にできるが、一人目の A さんと二人目の A さんの区別をつけるのはコンピュータでは非常に難しく、最近の AI 技術ではそういうことがかなりの確率で可能になってきている場合もあるものの、決して正確なものではない。人の判断も絶対に正しいとは限らないにせよ、多様な文脈から明白な根拠を以て判断するということについてはまだ専門家による人力に一日の長があり、専門家の判断力を少しでも多く活かし後世に残していくためにもこのような形で同名人物の区別を記述しておくことは有用である。

なお、人名や地名などの固有名詞は、表記が異なっているが同じものを指していることがあるため、上の図のように、同じものかどうかを属性として記述しておくと分析等をする際の精度はより高まるだろう。

事例がやや少ないものの、このようにして用途によって深さの方向性が異なっていて、それに用いるタグの種類も異なってくる、という点はご覧いただけたのではないかと思う。

3-7．そもそもタグ付けとは

このようにしてさまざまなタグの付け方があり、分野ごとに異なるタグが用意されることになるのであれば、タグの構造を設定したり使い方をレクチャーしたりする、かなり詳しい人が分野ごとに必要となりそうである。しかし、人文学分野は多岐にわたるものであり、個々の分野で見てみると人数もそれほど多くなく、それぞれの分野で技術レベルの高い人を養成することはなかなか難しい。そこで、分野横断で、共通化できるタグはなるべく共通化して、しかし共通化できないものは分野にあわせてタグを設定する、というやり方が一つの選択肢として出てくる。まさにそこを目指して作成されてきたタグ付けのルールが TEI ガイドラインであり、それゆえに、特定の分野に偏ることなく、コミュニティに参加する人文学研究者たちが取り組む分野全体に対応しつつ、個別分野にも丁寧に配慮しようとしてきたのである。

TEI ガイドラインはともかくとして、ここでは、「タグ」をつけることの可能性についてもう少し検討してみよう。

前節でみたように、タグの名前はタグが囲まれる文字列に対してなんらかの意味を付与することになる。人名であったり、手紙の宛先であったり、校異情報であったり、さまざまである。源氏物語の校異情報マークアップの例では <app> の中に <lem> と <rdg> が入っていたが、そのようにして入れ子構造を作っていくことで上位タグの意味を下位のタグに継承していくことも検討する必要がある。

これは、たとえば人名の例で考えてみると、

```
< 人名 > 森鴎外 </ 人名 >
```

というタグの付け方があったとして、これを姓名にわけると

```
< 人名 >< 姓 > 森 </ 姓 >< 名 > 鴎外 </ 名 ></ 人名 >
```

という風になる。ここで、「森」という < 姓 > の人の名は、一つ上の階層の < 人名 > にあがると、その下位に < 名 > である「鴎外」が確認できる。階層構造の活かし方はたとえばこのような感じになる。

ちなみに、森鴎外は、本名は森林太郎であり、他にも、観潮楼主人、千朶山房など、さまざまなペンネームを使用していたとのことである。そうすると、前節で示したような、複数の名称が同じ人であると示すことの有用性はより高くなる。この場合には、ペンネームと本名を示すこと、そして、それが一人の人物であること、を示したいということになる。これをタグ付けすると以下のようになるだろう。【図 5】

```
<人物>
    <本名><姓>森</姓><名>林太郎</名></本名>
    <ペンネーム><姓>森</姓><名>鴎外</名></ペンネーム>
    <ペンネーム>観潮楼主人</ペンネーム>
    <ペンネーム>千朶山房</ペンネーム>
</人物>
```

図 5

観潮楼主人と千朶山房については、姓名として区別できるかどうか筆者にはわからなかったため、姓名を区別するタグはつけていない。タグの階層をそろえるために姓か名かのどちらかのタグをつけるということも処理を効率化する上では考えられるのだが、姓か名かのタグをつけてしまうと、姓か名ではないものにいずれかであるという誤った情報を与えてしまうことになるため、むしろ階層をそろえることを犠牲にしてこのような記述にしている。この場合、処理する際には「ペンネームのタグの下位には姓・名のタグがあってそのなかに名前の文字列が入っている場合と、姓・名のタグがなくて名前の文字列がペンネームタグのなかに直接書かれている場合がある」という前提で処理をすることになる。

あるいは、処理上の例外を減らすために、「名前全体」というタグを作って、姓名を区別できないものも同じ階層にしておくという方法もある。この場合、以下のようになる。【図 6】

```
<人物>
    <本名><姓>森</姓><名>林太郎</名></本名>
    <ペンネーム><姓>森</姓><名>鴎外</名></ペンネーム>
    <ペンネーム><名前全体>観潮楼主人</名前全体></ペンネーム>
    <ペンネーム><名前全体>千朶山房</名前全体></ペンネーム>
</人物>
```

図 6

この場合、階層は同じなので、「ペンネームタグの下位には姓・名・名前全体のいずれかのタグがあり、そのなかに名前の文字列が入っている。」というルールで処理をすることになり、処理側の例外は少し減らすことができる。しかし一方で、「姓が来たときは名があるのでもう一つ処理をすることを前提にしなければならないが、名前全体がきたときは一つだけ」という処理が必要になる。どちらの処理の方が効率的・効果的か、というのは状況に応じて異なる。ただ、いずれにしても、＜名前全体＞があるタグ付け方法とないタグ付け方法は、この段階では機械的に置き換え可能であり、自動置き換え処理を差し挟むことができる状況であれば、いずれの方法を採っても問題ないだろう。

さて、前回記事の人名の書き方では、文章のなかに登場する人名や呼称を「属性で参照」することによって一意に同定できるようにしていた。では、この書き方の場合、それをどのように実現するのか、少し検討してみよう。たとえば、以下のような文章があったとしよう。

> 東京都文京区千駄木町には、登録有形文化財の和風住宅がある。ここは、明治の文豪である森鷗外と夏目漱石が、相次いで住んだことがあり、鷗外がここで執筆した小説に『文づかひ』がある。その後に住んだ漱石は『吾輩は猫である』をここで発表した。

この文章に含まれる人名にタグ付けすべく、対象文字列の始まりを＜人名＞、その終了箇所を＜/人名＞として囲んでみると以下のようになる。【図 7】

東京都文京区千駄木町には、夏目漱石旧居跡（猫の家）がある。
これは、明治20年頃に建てられた和風住宅であり、
明治の文豪である<人名>森鷗外</人名>と
<人名>夏目漱石</人名>が、相次いで住んだことがある。
<人名>鷗外</人名>がここで執筆した小説に『文づかひ』がある。
その後に住んだ<人名>漱石</人名>は『吾輩は猫である』を発表した。

図 7

このようにしてタグをつけた場合、「人名」タグを対象とした取り出しの処理をすることで以下のようなデータを列挙できる。【図 8】

<人名>森鷗外</人名>
<人名>夏目漱石</人名>
<人名>鷗外</人名>
<人名>漱石</人名>

図 8

この場合、まったく同じ文字列ではないので、機械で処理した場合、「多分同じ人物」という情報しか得られない。この短い文章であればそれでも大丈夫だが、大量のテキストデータのなかでこのようなことが起きると、同定はやや難しい。そこで、タグに対して本名が何かという情報を与えてみたのが以下のものである。【図 9】

東京都文京区千駄木町には、夏目漱石旧居跡（猫の家）がある。
これは、明治20年頃に建てられた　　和風住宅であり、
明治の文豪である<人名 本名="森林太郎">森鷗外</人名>と
<人名 本名="夏目金之助">夏目漱石</人名>が、相次いで住んだことがある。
<人名 本名="森林太郎">鷗外</人名>がここで執筆した小説に『文づかひ』がある。
その後に住んだ<人名 本名="夏目金之助">漱石</人名>は『吾輩は猫である』を発表した。

図 9

上記では、＜人名＞というタグに対して、本名という属性を与え、その値として本名の文字列を指定している。このようにした場合、「人名タグの本名属性」を見ることで、同一人物かどう

かを確実に判定できることになる。

ただし、これでは、まだ不足な面がある。この本名というのがどういう情報なのか、本名がすごく長い場合はどうするのか、この人物の本名以外の情報はどうなっているのか、等々、もっとさまざまな情報を付与できた方がいい場合もある。そこで出てくるのが、以下のようにして、人物情報を別に作り、そこにリンクするという方法である。【図 10】

```
<人物 id="PS1">
    <本名><姓>森</姓><名>林太郎</名></本名>
    <ペンネーム><姓>森</姓><名>鷗外</名></ペンネーム>
    <ペンネーム>観潮楼主人</ペンネーム>
    <ペンネーム>千朶山房</ペンネーム>
</人物>
<人物 id="PS2">
    <本名><姓>夏目</姓><名>金之助</名></本名>
    <ペンネーム><姓>夏目</姓><名>漱石</名></ペンネーム>
</人物>

東京都文京区千駄木町には、夏目漱石旧居跡（猫の家）がある。
これは、明治20年頃に建てられた　和風住宅であり、
明治の文豪である<人名 人物="PS1">森鷗外</人名>と
<人名 人物="PS2">夏目漱石</人名>が、相次いで住んだことがある。
<人名 人物="PS1">鷗外</人名>がここで執筆した小説に『文づかひ』がある。
その後に住んだ<人名 人物="PS2">漱石</人名>は『吾輩は猫である』を発表した。
```

図 10

このように、人物タグを別に作成してそこに人物に関する情報をさまざまに記載しつつ、人物に対して id="PS1" のように、PS1 という値を持つ id という属性を与え、それに対して本文中の文字列に付与したタグからは属性「人物」を用いて参照する、という風にすれば、属性情報のリンクをたどることで本文と登場人物のさまざまな情報を確実に結びつけられる。

タグとして付与できる情報にはこれ以外にもさまざまなものがある。今回の文章では地名や建築物、年代、書名などがあり、それもタグ付けしてみると以下のようになる。【図 11】

```
<地名>東京都文京区千駄木町</地名>には、
<建築物>夏目漱石旧居跡（猫の家）</建築物>
がある。これは、<年代>明治20年</年代>頃に建てられた
和風住宅であり、明治の文豪である<人名 人物="PS1">森鷗外</人名>と
<人名 人物="PS2">夏目漱石</人名>が、相次いで住んだことがある。
<人名 人物="PS1">鷗外</人名>が
ここで執筆した小説に<書名>『文づかひ』</書名>がある。
その後に住んだ<人名 人物="PS2">漱石</人名>は
<書名>『吾輩は猫である』</書名>を発表した。
```

図 11

このテキストは四つの文から成っており、それは句点で区切られているため、区切り自体は自明である。しかし、現代文はともかく、古文や漢文の場合には句点がないこともある。そのような場合、タグで文を区切るという選択肢が出てくる。区切り方としては、いくつかの方法があるが、たとえば、＜文＞タグで囲むやり方が考えられる。【図 12】

```
<文><地名>東京都文京区千駄木町</地名>には、
    <建築物>夏目漱石旧居跡（猫の家）</建築物>がある。</文>
<文>これは、<年代 西暦="1887">明治20年</年代>頃に建てられた
和風住宅であり、明治の文豪である<人名 人物="PS1">森鷗外</人名>と
<人名 人物="PS2">夏目漱石</人名>が、相次いで住んだことがある。</文>
<文><人名 人物="PS1">鷗外</人名>がここで執筆した小説に
    <書名>『文づかひ』</書名>がある。</文>
<文>その後に住んだ<人名 人物="PS2">漱石</人名>は
    <書名>『吾輩は猫である』</書名>を発表した。</文>
```

図 12

扱うテキストに現代文だけでなく古文や漢文が入っていた場合には、このように＜文＞タグで区切られていると、同じ処理方法で処理できることになり、より有用性が高まることになる。

また、タグの種類が増えすぎると、タグを作った人はともかくとして、新たにタグ付け作業に参加する人やタグを用いて処理する人にとっては難易度が高くなってしまう。そこで、タグをまとめるということも考える必要が出てくる。上述の人物情報でみてみるなら、たとえば、＜本名＞や＜ペンネーム＞などは人名の一種であると理解しやすい。そこで、この二つは＜人名＞の変種として捉えることにして、＜人名＞タグに対して以下のように属性「タイプ」を用いて区別している。【図 13】

```
<人物 id="PS1">
    <人名 タイプ="本名"><姓>森</姓><名>林太郎</名></人名>
    <人名 タイプ="ペンネーム"><姓>森</姓><名>鷗外</名></人名>
    <人名 タイプ="ペンネーム">観潮楼主人</人名>
    <人名 タイプ="ペンネーム">千朶山房</人名>
</人物>
```

図 13

3-8．タグを介した外部情報との連結

このようにタグを付けたデータは、タグを介して外部の情報とリンクすることもできる。たとえば、著書を残したことがある人の多くは VIAF（http://viaf.org/）という国際的な典拠情報において ID が割り当てられている。VIAF は、主に世界中の国立図書館が提供した典拠ファイルを統合して提供しているものであり、図書館が提供する著者・著作の典拠情報に関しては網羅性が高い。VIAF の ID、もしくは URI を参照すれば、著者としての人物を同定できるとともに、関連する著作とも関連づけられるようになり、より広い知識ネットワークの中に手元のテキストデータを位置づけられるようになる。

今回の 2 名の場合、それぞれに VIAF での登録があり、これを参照することによって外部の豊かなデータと接続できることになる。今回は著作者の人物情報としての VIAF とリンクすることが目的であるため、この VIAF の ID は、＜人物＞タグの属性か、もしくはその下位に何らかの形でタグを付与しつつ記述することになる。たとえば以下のようになる。【図 14】

```
<人物 id="PS1">
    <VIAF>http://viaf.org/viaf/15096</VIAF>
    <本名><姓>森</姓><名>林太郎</名></本名>
    <ペンネーム><姓>森</姓><名>鷗外</名></ペンネーム>
    <ペンネーム>観潮楼主人</ペンネーム>
    <ペンネーム>千朶山房</ペンネーム>
</人物>
<人物 id="PS2">
    <VIAF>http://viaf.org/viaf/56614190</VIAF>
    <本名><姓>夏目</姓><名>金之助</名></本名>
    <ペンネーム><姓>夏目</姓><名>漱石</名></ペンネーム>
</人物>
```

図 14

このように記述することで、本文につけられた＜人名＞タグから人物の ID を介して＜人物＞タグに至り、そこに記述された **<VIAF>** の URI を取得できるようになる。また、逆に、VIAF の URI から､その人物を指す本文中の記述箇所をピックアップできるようになる。このようにして、本文が国際的な典拠情報とつながることになるのである。

地名や建築物、書名等も、人名における VIAF と同様に、それぞれ外部のデータに紐付けることができる。地名であれば、住所や地理座標、建築物であれば竣工日や施主、使用者、住所、書名であれば出版年や出版社等々、いろいろな関連情報があり、それを人物と同様に記述して、ID で紐付ける。それによって、手元のテキストデータは世界中で構築されつつある知識のネットワークのさまざまなポイントと連結され、より深く活用できることになるのである。

また、年代に関しては、西暦年の情報を与えるのであれば、人物情報などと異なり、西暦年は基本的には自明であり、それ自体がコンピュータでも処理できるため、以下のように、属性「西暦」のみで表現することも可能だろう。【図 15】

```
<地名>東京都文京区千駄木町</地名>には、<建築物>夏目漱石旧居跡（猫の家）</建築物>がある。
これは、<年代 西暦="1887">明治20年</年代>頃に建てられた
和風住宅であり、明治の文豪である<人名 人物="PS1">森鷗外</人名>と
<人名 人物="PS2">夏目漱石</人名>が、相次いで住んだことがある。
<人名 人物="PS1">鷗外</人名>が
ここで執筆した小説に<書名>『文づかひ』</書名>がある。
その後に住んだ<人名 人物="PS2">漱石</人名>は
<書名>『吾輩は猫である』</書名>を発表した。
```

図 15

3-9．参照情報ファイルを独立させる

以上のようにして ID で参照すべく作成した人物・地名・建築物・書名等の情報は、このテキストが長く大きなものになったとしても同様に参照可能であり、これを他のファイルにコピペして利用することも可能である。

あるいは、参照情報を一つのファイルにまとめておいて、本文とは別なファイルとしつつ、本文からは外部のファイルを参照するという形でその参照用ファイル内の情報を参照することもあり得る。それにより、複数のファイルから参照することも可能になり、新たな本文ファイルを作成していく際に、その都度参照情報を本文ファイルに書き込む必要がなくなる。さらに、参照情報ファイル内のデータを修正・更新するだけで本文ファイルにも反映されるため、データを管理するコストを低減できる。

あるいはまた、この参照用データを Web で公開して外部からも ID 等で参照できるようにすると、自らのデータだけでなく世界のさまざまなデータからも参照できるようになる。これは参照情報の有用性を高めることにもつながるため、構築したテキストデータを公開する際には、この種の参照情報も同時に公開することをぜひ検討されたい。

3-10．タグの共通化に向けて

このようにしてタグ付けを進めていく場合、多くの人が同じルールに従ってタグを付与しながらデータを作成していけば､多くのデータを横断的に検索･処理できるようになる｡しかしながら、自分で設定したタグやその付け方を他の人にも広めて同じように作業してもらうのはそれほど容易なことではない。他の人と共同作業をしていると、テキストに対する観点はそれぞれ異なるために、自分は付けたいとは思わないタグを使いたいという人が必ず出てくる。共通のタグを利用しようとすると、そのたびに、新しいタグはどのように使用されるべきであり、既存の他のタグ

```
<s><placeName>東京都文京区千駄木町</placeName>には、
    <object>夏目漱石旧居跡（猫の家）</object>がある。</s>
<s>これは、<date when="1887">明治20年</date>頃に建てられた
    和風住宅であり、明治の文豪である<persName corresp="PS1">森鷗外</persName>と
    <persName corresp="PS2">夏目漱石</persName>が、相次いで住んだことがある。</s>
<s><persName corresp="PS1">鷗外</persName>がここで執筆した小説に
    <bibl>『文づかひ』</bibl>がある。</s>
<s>その後に住んだ<人名 人物="PS2">漱石</人名>は
    <bibl>『吾輩は猫である』</bibl>を発表した。</s>

<person xml:id="PS1">
    <persName type="autonym"><surname>森</surname><forename>林太郎</forename></persName>
    <persName type="pseudonym"><surname>森</surname><forename>鷗外</forename></persName>
    <persName type="pseudonym">観潮楼主人</persName>
    <persName type="pseudonym">千朶山房</persName>
</person>
<person xml:id="PS2">
    <persName type="autonym"><surname>夏目</surname><forename>金之助</forename></persName>
    <persName type="pseudonym"><surname>夏目</surname><forename>漱石</forename></persName>
</person>
```

図 16

とどのような関係にすべきか、ということを検討しなければならなくなる。さらに、変更があれば、同じルールを採用している人たち全員に周知しなければならない。こういったことはコストとしては少なくないものであり、既存のルールがあればなるべく利用したいところである。そこで出てくるのが前出の TEI ガイドラインである。たとえば、ここまでタグ付けしてきたものを TEI ガイドラインに準拠させると【図 16】のようになる。

ここまでに用いてきたタグはいずれも TEI ガイドラインに対応するものが用意されているものであり、タグの組み合わせ方についてもすでに決められている。日本語のテキストなのにタグ名が英語というのは少しやりにくい面もあるかもしれないが、上記のように、TEI ガイドラインのタグを日本語に置き換えて作業して、作業が一段落したところで英語のタグに置換するという方法もあるだろう。いずれにしても、既存のルールをいかにしてうまく活用するか、ということが、限られた人手と費用でよりよいテキストデータを作成していくためには重要である。

TEI ガイドラインは人文学分野における汎用性と人文学個別分野の特性の双方に配慮しつつ全体として人文学においてデータを共有するための枠組みとして 30 年以上の時間をかけて人文学のコミュニティにおいて育まれてきた。しかしながら、コミュニティ駆動型のガイドラインでありながら欧米地域以外の研究者が少ないコミュニティであったため、欧米地域以外への配慮は必ずしも十分ではなかった。2016 年の東アジア／日本語分科会設立を機に、徐々に欧米地域外からのコミュニティへの参加が増えてきており、それに伴って、他地域のテキストへの配慮も手厚くなってきた。本書の第 2 部・実践編の TEI 入門では、そういった事情も踏まえつつ、TEI ガイドラインを通じて作ったテキストデータをどのように利便性の高いものにしていくか、ということについて解説する。

注

1 IRG (Ideographic Research Group) , https://appsrv.cse.cuhk.edu.hk/~irg/.

2 ISO/IEC JTC1/SC WG2 Document Registry, http://www.unicode.org/wg2/WG2-registry.html.

3 TEI ガイドライン 第五章 5 Characters, Glyphs, and Writing Modes（外字等）, https://tei-c.org/release/doc/tei-p5-doc/en/html/WD.html.

4 TEI ガイドライン第二章 The TEI Header, https://tei-c.org/release/doc/tei-p5-doc/en/html/HD.html.

5 TEI ガイドライン第二章 TEI ヘッダー日本語訳, https://www.dh.ku-orcas.kansai-u.ac.jp/?p=791.

6 TEI 協会「言語学者のための TEI」分科会, https://wiki.tei-c.org/index.php/SIG:TEI_for_Linguists.

7 Best Practices for TEI in Libraries, https://candra.dhii.jp/nagasaki/tei_lib/bptl-driver.html.