第 4 章

短単位・長単位データベース

山口昌也

本章では，「短単位・長単位データベース」について解説する (以後，特に断りがなければ，「単位データベース」と略記する)。

単位データベースは，転記テキストの基本形に対する短単位・長単位解析結果を格納するためのデータベースである。単位データの構築にあたっては，単位データを管理するという基本的な機能とともに，単位データの人手修正を支援するための基盤として機能してきた。そこで，本章では，人手を加えて，単位データを構築するという側面から，次の構成で単位データベースについて解説を行うことにする。

まず，4.1 節において，単位データベース自体についての設計，および，運用方法について解説する。4.2 節では，転記テキストから単位データベースを構築する手順について説明する。4.3 節では，単位データの人手修正を行うためのツールとその運用方法について述べる。このツールは，単位データベースのクライアントとして機能し，多数の作業者が同時に修正作業を行うことができる。最後に，4.4 節では，単位データの整合性を維持するための単位辞書，および，活用表について，作成方法も含めて説明する。これらは，情報通信研究機構が担当した自動短単位解析システムの辞書データとしても利用される。

なお，本章で示す単位データは，CSJ として公開されている「短単位・長単位データベース」に含まれる単位データとは，一部形式が異なる。これは，本章で示す形式には，管理用のデータなどが含まれているためである。公開版の単位データについては，CSJ 付属の「短単位・長単位データマニュアル」を参照されたい。

4.1　短単位・長単位データベースの設計と運用

4.1.1　概要

本節では，単位データベースの設計と運用について述べる。

すでに述べたように，単位データベースは，転記テキストの基本形に対する短単位・長単位解析結果を格納するためのデータベースである。転記テキストに付与されている情報のうち，コメントを除くすべての情報を保持している。したがって，単位データベースには，短単位・長単位という形態論情報の他に，転記基本単位や文節などの転記テキストに関する情報も格納することになる。

単位データベースには，解析手法，単位認定手法の点から，次の 4 種類の異なる単位データが格納される。

- 人手解析単位データ（短単位）
- 人手解析単位データ（長単位）
- 自動解析単位データ（短単位）
- 自動解析単位データ（長単位）

まず，形態論情報の解析手法の面から見ると，人手単位解析と自動単位解析に分けられる。人手単位解析では，基本的にすべての単位データに対して，人手でチェック・修正を行う。この単位データは，精密なアノテーションを必要とする言語研究や，自然言語処理システムにおける学習用データとしての利用を想定する。一方，後者は，基本的に自動的に単位解析を行い，解析結果に対する人手によるチェック・修正は部分的にしか行わない。自動単位解析では，人手単位解析によって得られた単位データから言語モデルを学習し，その言語モデルに基づいて解析する。なお，人手解析と自動解析単位データとでは，品詞情報に一部差異がある。詳細は，4.4.4 節を参照のこと。

次に，単位認定手法の面から見ると，短単位と長単位の 2 種類がある。ただし，2 種類の単位は同一の転記テキストに対して，並行して付与されるので，短単位，長単位ごとに個別の転記テキストがあるわけではないことに注意されたい。

これら 4 種類の単位データのうち，形態論情報付与に関して，国語研究所が担当したのが，人手解析単位データの短単位・長単位である。一方，自動解析単位データに関しては，情報通信研究機構が実現した自動短単位・長単位解析システムにより構築した（構築方法については，（内元 2000，2003）を参照していただきたい）。ただし，自動単位解析システム用の単位辞書の構築と解析精度向上のための人手修正については国語研究所が一部担当した。したがって，プロジェクトの中盤から後半にかけては，4 種類すべての単位データが単位データベースに格納，および，運用されていた。

次に単位データの形式だが，上記の 4 種類の単位データは，データベース形式上，同一の形式を持ち，すべて一つのテーブル（table）にまとめて格納・管理される。上記データの区別は，レコード中のデータベース管理用のフィールドで行う（詳細は，4.1.3.6 節を参照のこと）。

このテーブルの 1 レコードは，1 短単位に相当する。長単位は，構成要素の短単位の並びで表現する。したがって，単位データベース上で，長単位は一つ以上のレコードの並びで表現される。このように，単位データベースの構造は非常にシンプルである。単位データベースのレコード形式については，4.1.2 節で設計を行った上で，4.1.3 節で詳しく解説を行う。

次に，単位データの構築にあたって，単位データベースがどのような位置付けにあるかを概説しておく。単位データの構築環境の全体図を図 4.1 に示す。

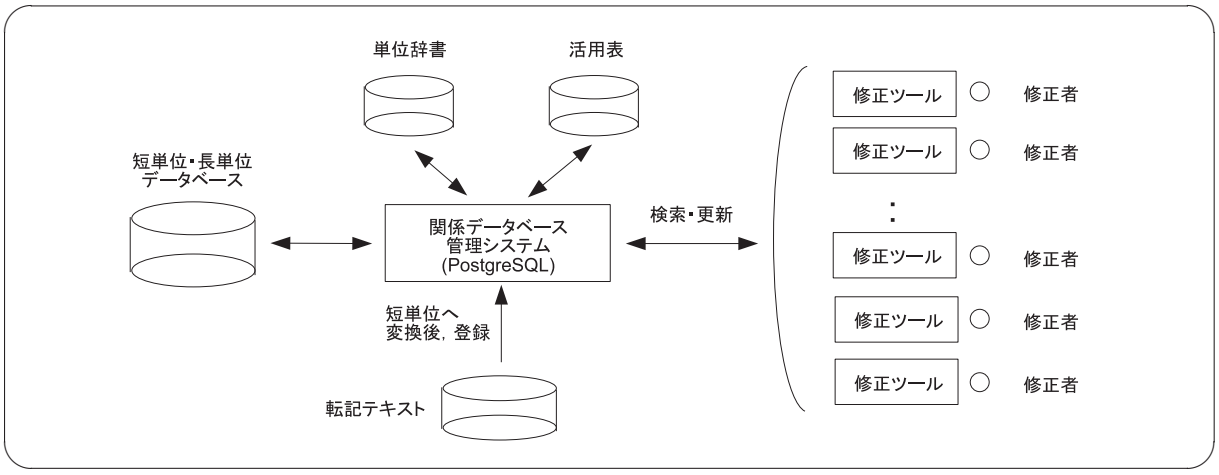


図 4.1 単位データの構築環境

国語研究所では単位データの人手解析を主として担当している事から，この単位データ構築環境は，人手単位解析の精度，効率，整合性を考慮して設計した。具体的に焦点を当てているのは，次の二つの事柄である。

- 単位データの人手解析・修正の支援による精度，効率の向上
- 単位データの言語学的・データ形式上の整合性の維持

単位データベースは，図 4.1 のとおり，「関係データベース管理システム」(Relational Database Management System)[*1]によって管理される。

単位データベースに対する各種の操作は，データベース管理システムから SQL を用いて行われるが，単位データの一括登録などの管理目的以外では，通常「単位データベース修正ツール」（以後，特に断りがない限り，「修正ツール」と略記）を用いる。修正ツールを用いれば，修正者が SQL を覚える必要はなく，容易に人手解析・修正ができるだけなく，単位データベースと転記テキストとの間の不整合を防ぐなど，単位データの整合性を維持するのに役立つ。修正ツールは，単位データ修正者の使用する PC から，関係データベース管理システムのクライアントとしてアクセスする。また，図 4.1 のとおり，複数の修正者が同時に人手単位解析を行うことができる。

次に，図中の「単位辞書」，「活用表」は，修正ツールによる修正内容の整合性（例：動詞「食べる」の未然形が「食べ」になっているかどうか）をチェックするために用いる。これらも，単位データベースと同様に，関係データベース管理システムによって管理されるデータベースである。詳しくは，4.4 節で説明する。

以上のように，単位データベースは，修正ツール，単位辞書，活用表など単位データを構築する環境の中で中核をなし，その構造は修正ツールで実現される機能や，単位辞書・活用表の構造に大きな影響を与える。この後の節では，まず，4.1.2 節で単位データベースの構築の背景についてふれた後，4.1.3 節で構築の背景に基づいて単位データベースの設計を行い，4.1.4 節で実際の運用方法について説明する。なお，修正ツールを使っ

[*1] 本プロジェクトでは，オープンソースの関係データベース管理システム PostgreSQL を用いた。使用バージョンは開発時期により異なるが，ver.6.5，ver.7.2 を順次利用した。PostgreSQL に関する詳細な情報は，http://www.postgresql.org を参照のこと。

た修正作業の詳細については，別途修正ツールの設計，実現方法とあわせて 4.3 節で述べる。また，この後述べるように，転記テキストから単位データへの変換については，作業効率を上げるために，計算機による単位解析をはじめとした自動化を行っている。これらの処理については，4.2 節で詳説する。

4.1.2　単位データベースの設計

4.1.2.1　構築の背景

単位データベースの設計に先だって，単位データベース構築の背景を示すことにより，設計方針を明確にする。そこで，ここでは，「単位データベースのサイズ」，「単位データの種類」，「単位データの人手解析」，「形態論情報の仕様」，「情報の共有」という観点から，いかなる単位データベースが必要となるのかを考察することにする。

■単位データベースのサイズ　CSJ の計画段階では，構築する単位データの量は，約 700 万語（短単位）であると見積もられていた（前川他 2000）。このうち，約 50 万語が人手解析による単位データ（「コア」と呼ばれる）であり，国語研究所が主体となって構築を担当する部分である。1 章で述べられているように，「コア」は，言語研究としての利用や，自動単位解析システムの学習用データとしての利用を考慮して，99.9% 程度の高精度を目指す。したがって，少なくとも 50 万語程度の単位データに対する人手解析を行うために，単位データの検索，更新，追加，削除などを高速に実行できるデータベースを用意する必要がある。

さらに，コア以外の残りの 650 万語については，情報通信研究機構の自動単位解析システムにより構築されることが計画されていたが，これらのデータを参照しつつ，コアを人手解析することも考えられる。そこで，700 万語程度の単位データを扱えるようなデータベースを設計する。

■単位データの種類　すでに述べたように，単位データベースには，4 種類の異なる単位データが登録されることになる。このような複数の種類の単位データを含んだデータベースを構築するために必要な事柄を次に示す。

- 使用目的により，単位データを適宜使い分けできるようにすること。例えば，人手解析単位データの人手修正作業では，人手解析単位データだけを抽出して，分析したり，人手修正することが考えられるだろう。その一方で，特定の語の分析を行う場合には，より多くの用例を利用できたほうが好ましく，人手解析単位データだけでなく自動解析単位データを参照する必要性も生じる。
- 短単位と長単位間の整合性を維持すること。例えば，同一の長単位は，構成要素の短単位が常に一定でなければならない。また，長単位の活用形は，構成要素の末尾の短単位の活用形と同期している必要がある。

■単位データの人手解析　人手解析に対する必要事項を次に挙げる。

- 複数の修正者が同時に人手解析できるような単位データベース。すでに述べたように，構築する単位データベースのサイズは，少なくとも 50 万語以上であり，複数の修正者による人手解析が必須である。したがって，高速な検索・更新処理を実現することや，複数の修正者が同時にデータベースに対して修正作業を行っても，データ形式上，矛盾が起きないようにすることが重要である。

- 効率的で，高精度な形態論情報付与を可能にするインターフェイス。単位分割位置の誤りや，品詞・活用の種類などの付与情報を，正確，かつ，容易に修正できるようにする。
- 修正者ごとの付与情報のゆれや修正者個人に起因する系統的な修正誤りの問題を解消する仕組み。この問題の解消には，誰がいつ，どの単位データを修正したか，ということを記録することが必要になるだろう。
- 人手単位解析時のリファレンスとしての短単位・長単位の辞書。複数の修正者が統一的な単位データを付与するには，辞書の構築は必要不可欠である。

■形態論情報の仕様 プロジェクト開始の段階で，形態論情報の仕様を完全に確定することは難しい。また，短単位・長単位の辞書も整備されていなかった。したがって，単位データの構築をしながら，形態論情報の仕様と短単位・長単位の辞書を整備していくことになる。このことは，単位データの構築の段階で，頻繁に辞書に変更が加わること，また，付与情報に揺れが生じる可能性があることを意味する。そのため，単位データベースには，次のことが求められると考えられる。

- 単位の仕様変更に伴う単位データベースの変更・拡張に柔軟に対応できること
- 単位の仕様変更時に付与情報の揺れを統一していく仕組み
- 段階的な単位辞書の整備

■情報の共有 すでに述べたように，単位データベースの作成には，複数の修正者が関わることが必要不可欠である。したがって，形態論情報の仕様をはじめとする各種の情報を修正者間で共有することが精度の高いデータを作成するために必要である。

さらに，CSJ コーパスの作成には，複数の異なるグループが関わっている。国語研究所の内部に限っても，転記テキスト，形態論情報，韻律・分節音の三つのグループが存在し，それぞれが担当するデータを個別に作成していく。したがって，形態論情報の修正者間における情報の共有のみならず，各グループ間のやりとりを緊密に行う必要がある。例えば，転記テキストが変更されれば，単位データベースに記述されている形態論情報も修正しなければならない。また，その逆もありえる。

4.1.2.2 単位データベースの特徴

以上の背景に基づき，単位データベースを設計した。本単位データベースの特徴は，次のとおりである。

■関係データベースによる管理 単位データベース，単位辞書，および，活用表（図 4.1 参照）は，関係データベースで管理する。関係データベースを用いる理由は，次の五つである。

- 短単位・長単位データの並びは表形式で表現できるので，関係データベースとの親和性が高い。また，単位辞書・活用表も同様に表形式で管理することが可能である。
- 関係データベースは 1970 年代から存在する技術であり，700 万語程度のデータに対して検索，更新，追加，削除などの処理を高速に，かつ，安定して実行できることが実証されている。
- SQL を用いて，多様な検索を行うことができる。
- サーバ・クライアント形式でのデータ処理の仕組みやデータ処理における排他処理機能が備えられているため，複数の修正者を想定したシステムを容易に構築することができる。

- また，オープンソフトウェアのデータベースソフトウェアが存在するので，システムを実際に稼働させる前に，人手単位解析処理にとって実用的な速度で動作するかなどのテストを，コストをかけることなく実施できる。

■シンプルな構造　単位データベースには，4 種類の異なる単位データが格納される。これらの形式は，すべて同一とし，単一の table 中に格納する。単位データの使い分けは単位データベースの table の設計（4.1.3 参照）によって解決する。このようなシンプルな構造を持った単位データベースには，次の利点がある。

- 単位データベースの構造の拡張や変更が容易である。実際，単位データベースは，プロジェクトの初期段階では，人手解析単位データを格納するだけの構造しか持っていなかった。後述する単位データベース（表 4.1 参照）における長単位データや予備的な情報に関連する構造は，プロジェクトの進行にしたがって，拡張していったものである。
- 単位データベースにアクセスするクライアント（例：修正ツール）の設計がシンプルになり，開発期間を短縮できる[*2]。

その一方で，単位データの整合性を確保するという観点から見ると，シンプルさは欠点にもなるが，修正ツールや単位データベースの運用方法を工夫することにより，その欠点を補う，という方針を取る。

例えば，本来，単位データベースと短単位・長単位の辞書は別の table とし，単位データベースには，短単位・長単位の辞書への参照のみを記述すべきである。なぜならば，同一の単位の付与情報を一括して変更できるなど，単位データを統一的に管理できるからである。しかし，単位辞書が未整備であり，人手解析中に単位辞書を保守する必要があるなど，修正者の手間が大きいと判断し，単位データベースから単位辞書への参照は行わなかった。そのかわり，修正ツールと単位データベース運用過程に単位データの整合性をチェックするための手段を用意した（4.1.4.1，4.3.7 節を参照）。

■言語分析用の情報の保持　人手単位解析では，情報付与のために，解析中の単位データに対して，簡単な分析ができる必要がある。例えば，活用形を付与するには，後文脈の情報が必須である。そこで，本単位データベースは，個々の単位データに対して，次の二つの情報を付与した。

- KWIC (Key Word In Context)
- 前後単位へのリンク

KWIC は，従来から言語分析用に用いられてきた手法であり，その有効性は実証されている。本データベースでは，KWIC の前後文脈を静的に付与しておくことにより，大量の単位データを高速に検索できるようにした。また，前後単位へのリンク情報を持つことにより，関係データベース管理システムの機能を用いて，前後単位に対する付与情報の参照や修正を高速に行うことができる。

■複数修正者の想定と単位ごとの変更管理　複数の修正者が人手解析を行うこと，また，高精度の単位データを構築することを考慮し，個々の単位データに管理用の情報を付与した。具体的には，各単位データごとに最終更新時間・更新者を保持している。これにより，以下の事柄を実現している。

[*2] 単位データベース，および，修正ツールの設計・開発期間は，2 ヵ月程度しかなかった。

- 複数の修正者による人手解析を実現するために必要な，単位データ更新時の排他処理（4.3.7 節を参照）
- 最終更新者による修正者ごとの系統的な誤り分析や，最終更新時間を利用して一定期間中に修正されたデータをチェックするなど，解析精度を向上させるための仕組み

■修正ツール，単位辞書，ニュースシステムとの連係 4.1.2.1 節において必要性を考察し，単位データベース自体に備わっていない機能は，修正ツール，単位辞書，ニュースシステムとの連係を図ることにより実現している（4.3 節参照）。

- 修正ツールの GUI により，SQL の知識がなくても単位データベースに対する修正ができるようにした。
- 修正ツールに単位データの誤修正を防止する機能や，効率的に付与情報を入力する機能を付加した。また，修正ツールで修正する際に，修正結果の整合性を単位辞書によりチェックするようにした。
- ニュースシステムを導入することにより，修正者間，グループ間（転記テキストグループと形態論情報グループ）で情報の共有が図れるようにした。

4.1.3 単位データベースの構造

4.1.3.1 単位データベースのレコード形式

単位データベースのレコード形式は，表 4.1 のとおりである。なお，表中，[長] とあるのは，長単位に関連するデータであることを示す。単位データベースの 1 レコードは 1 短単位に相当し，31 フィールドからなる。フィールドの内容は，レコード ID，転記テキストの情報，短単位の情報，長単位の情報，管理情報の五つ大きく分けられる。詳細については，この後の節で順に述べることにする。

4.1.3.2 レコード ID 関連

各レコードの第 1，2 フィールドがそれぞれ「ID」と「後続 ID」である。「ID」は，8 桁の通し番号（1 以上の値を取る）で，単位データベース中で一意に定まる値である。単位データベースの 1 レコードは 1 短単位に相当するので，短単位の ID でもある。

「後続 ID」は，転記テキストにおける出現順で，当該短単位（レコード）に後続する短単位の ID を保持する。なお，講演の末尾の短単位のように，後続する短単位が存在しない場合は，「後続 ID」を ‘00000000’ とする。「後続 ID」を使うと，後続する短単位，および，前接する短単位を高速，かつ，容易に検索することができる。例えば，ID ‘00000010’，後続 ID ‘00000011’ の短単位に後続，前接する短単位を SQL で検索すると次のようになる。なお，SQL 中の `unit` とは，単位データベースを表す table 名である。また，「ID」，「後続 ID」などのフィールド名は，説明のための便宜的なフィールド名である。

- 後続する短単位を検索する場合

```
select * from unit where ID='00000011'
```

- 前接する短単位を検索する場合

```
select * from unit where 後続ID='00000010'
```

表 4.1　単位データベースのレコード形式

番号	フィールド名	内容
1	ID	当該短単位の通し番号（8 桁）
2	後続 ID	後続する短単位の ID（後続する短単位が存在しない場合は，00000000）
3	講演 ID	当該短単位が収録されている転記テキストの講演 ID
4	転記情報	当該短単位を含む転記単位のタイムスタンプなど
5	前文脈	当該単位に先行する文脈（最長 15 短単位）
6	出現形	当該短単位の転記テキスト（基本形）における出現語形
7	後文脈	当該単位に後続する文脈（最長 15 短単位）
8	タグなし出現形	出現形から転記テキストのタグを取り除いたもの
9	代表形	出現形の標準的な語形（国語辞典の見出しに相当）
10	代表表記	代表形を漢字，仮名などで表記したもの
11	発音形	当該短単位の発音形（転記テキストの発音形に相当）
12	品詞	当該短単位の品詞
13	活用の種類	当該短単位の活用の種類（「カ行五段」等）
14	活用形	当該短単位の活用形（「連用形」等）
15	その他の情報 1	品詞の下位分類（「助詞」の下位分類として「格助詞」等）
16	その他の情報 2	語形の情報（「促音便」等）
17	その他の情報 3	「言いよどみ」「メタ」等の補足情報（複数情報がある場合は，全角スペースで区切る）
18	後続 ID [長]	後続する短単位の ID（後続する短単位が存在しない場合は，00000000）
19	品詞 [長]	長単位の品詞
20	活用の種類 [長]	長単位の活用の種類
21	活用形 [長]	長単位の活用形
22	その他の情報 1 [長]	長単位のその他の情報 1
23	その他の情報 2 [長]	長単位のその他の情報 2
24	その他の情報 3 [長]	長単位のその他の情報 3
25	代表形 [長]	長単位の代表形
26	代表表記 [長]	長単位の代表表記
27	最終更新者	当該レコードの最終更新者
28	最終更新時間	当該レコードの最終更新時間
29	予備 1	特定の目的を定めない，予備用のフィールド
30	予備 2	特定の目的を定めない，予備用のフィールド
31	予備 3	特定の目的を定めない，予備用のフィールド

4.1.3.3 転記テキストに関する情報

第 3，4 フィールドには，それぞれ「講演 ID」「転記情報」を格納している。単位データベースでは，この二つのフィールドの情報により，単位データが転記テキストのどの部分に対応するかを記述している。転記テキストに関する事柄については，2 章を参照のこと。

まず，「講演 ID」には，当該短単位が収録されている転記テキストの講演 ID が格納される。一方，「転記情報」は，当該短単位を包含する転記基本単位に関する情報を格納しており，転記テキストにおけるタイムスタンプに，短単位の位置情報を付加したものである。形式は，次のとおりである。

発話 ID タイムスタンプ 短単位位置情報

- **発話 ID：** 当該短・長単位を包含する転記基本単位の通し番号
- **タイムスタンプ：** その転記基本単位の開始時刻・終了時刻
- **短単位位置情報：** 転記基本単位の先頭からの行数（文節数），および，各行における先頭からのバイト数。バイト数は，出現形（転記テキストの基本形）を基準とする。文字のエンコーディングは Shift JIS である。[*3]。

転記情報の例として，転記テキストとそれに対応する単位データをそれぞれ図 4.2，4.3 に示す。なお，図 4.3 の単位データには，転記情報と転記テキストの基本形（出現形）だけ示してあり，他のフィールドの値は省略している。

```
0017 00051.048-00056.945 L:
日本語の                          & ニホンゴノ
文法は                            & ブンポーワ
0018 00057.439-00061.747 L:
従来の                            & ジューライノ
```

図 4.2 転記情報の例（転記テキスト）

```
0017 00051.048-00056.945 L:-001-001    日本
0017 00051.048-00056.945 L:-001-005    語
0017 00051.048-00056.945 L:-001-007    の
0017 00051.048-00056.945 L:-002-001    文法
0017 00051.048-00056.945 L:-002-005    は
0018 00057.439-00061.747 L:-001-001    従来
0018 00057.439-00061.747 L:-001-005    の
```

図 4.3 転記情報の例（単位データ）

[*3] 文字数でなく，バイト数を使ったのは，4.3 節で示す単位データベース修正ツールの実装に利用した Microsoft Excel のマクロ言語が文字列をバイト列で表現していたためである。

これらの図に示したとおり，各短単位（レコード）には，発話 ID とタイムスタンプが付与され，コロン (:) の後に，当該短単位の転記テキストの位置を示す数値が付与される。例えば，短単位「語」の転記情報は，

```
0017 00051.048-00056.945 L:-001-005
```

である。これは，発話 ID が “0017”，タイムスタンプが “00051.048-00056.945 L:” である[*4]ことを表す。そのあとの “-001-005” は，短単位「語」が転記基本単位の 1 行目の 5 バイト目から始まることを意味する。なお，Shift JIS でエンコードしているので，ASCII コードに属する文字を除いて，1 文字 2 バイトになる。

以上のとおり，「講演 ID」「転記情報」フィールドは，当該の短単位の転記テキスト中の位置を表すものであるが，二つのフィールドを組み合わせることにより，短単位の ID としての機能も果たす。また，この二つのフィールドをキーとしてソートすれば，短単位を転記テキストの時系列順にソートすることも可能である。

4.1.3.4　短単位の情報

短単位の情報は，5〜17 フィールドに格納される。

第 6，11 フィールドに転記テキストの基本形（出現形）と発音形が格納される。出現形に対しては，第 5，6，7 フィールドで KWIC を構成するようになっている。また，第 8 フィールドのタグなし出現形は，人手解析時の検索の便を考えて，出現形から転記テキストのタグを取り除いたものを格納している。

KWIC の前文脈（第 5 フィールド）・後文脈（第 7 フィールド）は，それぞれ 15 短単位が静的に格納される。人手解析の便を図り，短単位と短単位の間は，空白で区切られている。動的に KWIC を生成する手法を用いていないので，高速に単位データを閲覧することができる。なお，短単位の区切り位置を修正した場合に限り，修正ツールが KWIC を生成しなおす処理を自動的に行っている。詳しくは，4.3.6.1 節を参照のこと。

第 9〜17 フィールドには，当該短単位に付与される形態論情報が格納される。短単位の形態論情報については，3 章を参照のこと。ただし，活用の種類，活用形については，人手解析単位データと自動解析単位データに違いがあり，自動解析単位データのほうが詳細な情報が付与されている。自動解析単位データに付与されている活用の種類，活用形については，4.4 節を参照していただきたい。

なお，転記テキスト中には，**<雑音>**，**<ベル>** などの非言語音タグが一つの転記単位全体に対して付与されている場合がある。この場合，非言語音タグを便宜上，一つの短単位として扱う。ただし，「出現形」，「発音形」フィールドに当該タグが入るだけで，「代表形」，「代表表記」，「品詞」など，短単位，長単位に関する情報は，付与しない。

非言語音タグの例として，転記テキストと単位データを対応づけて，図 4.4，4.5 に示す。図 4.4 の発話 ID 0203 の非言語音タグ **<雑音>** は，図 4.5 で，便宜上，1 短単位となっているのがわかる。その際，出現形と発音形のフィールドに **<雑音>** が格納され，それ以外の形態論情報に関するフィールドは空欄となる。

4.1.3.5　長単位の情報

18〜26 フィールドに長単位の形態論情報が格納される。付与される形態論情報の仕様は，短単位と同一である（したがって，人手解析単位データと自動解析単位データの活用形，および，活用の種類には短単位と同様の差異がある）。

長単位の形態論情報は，長単位を構成する先頭の短単位に相当するレコードのみに記載される。長単位を構

[*4] ‘L’ は，音声を録音する際のチャンネル（Left）を表す。

成する先頭以外の短単位のレコードは，長単位関連のフィールドが空欄である。図 4.6 に長単位の単位データの例を示す。

図 4.6 では，長単位「音響モデル」が複数の短単位（ID が 00000010 と 00000011）から構成されている。長単位「音響モデル」に対する形態論情報は，ID が 00000010 のレコードに付与され，00000011 のレコードの長単位関連のフィールドは空欄になる。ただし，後続する長単位の ID を示す「後続 ID [長]」フィールドは，00000010 と 00000011 のいずれのレコードでも同じ値 00000012 が格納される。これにより，「後続 ID [長]」の値が同じ短単位を取り出せば，一つの長単位を構成する，すべての短単位を抽出することができる。

```
0202 00498.324-00501.003 L:
コーパスの                    & コーパスノ
0203 00501.163-00502.587 L:<雑音>
0204 00503.031-00503.812 L:
内容は                        & ナイヨーワ
```

図 4.4 非言語音タグの例（転記テキスト）

転記情報	出現形	発音形	品詞	その他の情報 1
0202 00500.324-00501.003 L:-001-001	コーパス	コーパス	名詞	
0202 00500.324-00501.003 L:-001-009	の	ノ	助詞	格助詞
0203 00501.163-00502.587 L:-001-001	**<雑音>**	**<雑音>**		
0204 00503.031-00503.812 L:-001-001	内容	ナイヨー	名詞	
0204 00503.031-00503.812 L:-001-005	は	ワ	助詞	係助詞

図 4.5 非言語音タグの例（単位データ）

ID	後続 ID	出現形	品詞	代表表記 [長]	品詞 [長]	後続 ID [長]
00000010	00000011	音響	名詞	音響モデル	名詞	00000012
00000011	00000012	モデル	名詞			00000012
00000012	00000013	の	助詞	の	助詞	00000013
00000013	00000014	研究	名詞	研究	名詞	00000014
00000014	00000015	が	助詞	が	助詞	00000015

図 4.6 長単位データの例

4.1.3.6 管理情報

単位データベースの 27〜31 フィールドは，単位データの管理情報を格納する。このうち，27，28 フィールドは修正者管理用のフィールドである。残りの 29〜31 フィールドは，特定の目的を定めない，予備用のフィールドだが，人手単位解析時に利用することを想定して設けた。

まず，修正者管理用の 2 フィールドは，通常，4.3 節で述べる単位データベース修正ツールが自動的に記入する。このツールで，レコードの情報を変更すると，自動的に当該レコードの修正者名と更新時刻が記入されるようになっている。この二つのフィールドを使うことにより，修正者を管理し，人手単位解析の効率や精度を向上させることができる。例を次に示す。

- 特定の修正者の作業量を計測することができる。
- 修正者ごとの人手単位解析の精度や誤りの傾向をつかむことができる。
- 特定の日に人手単位解析されたレコードをチェックできる。

次に，29〜31 フィールドの予備用のフィールドだが，プロジェクト期間中は，次のように利用していた。

修正者のコメント： 特に，プロジェクト後半では精度を下げないために，付与情報のチェックを一般の修正者が行い，誤りを発見した場合は，コメントだけを残し，実際の修正は，修正管理者（4.3.3 節を参照）が単位データを確認してから行っていた。修正管理者は，前出の修正者管理用フィールドと併せて検索することにより，特定期間内にコメントが付け加えられた単位データだけをチェック・修正できる。

人手解析単位データと自動解析単位データの別： すでに述べたように，単位データベースには，人手解析単位データと自動解析単位データが混在している。そのため，予備用フィールドのフィールドを使って，両者を区別していた。

自動解析単位データの「確信度」： 自動解析単位データには，正しさの程度を示す「確信度」が付与されている。4.4.5.2 節で述べるように，確信度の大きさを目安にして，単位データを人手でチェックした。その際，自動解析結果の確信度を予備用のフィールドに格納し，確信度が低い単位データを優先してチェックした。SQL を用いれば，指定した範囲の確信度を持つレコードを検索したり，確信度をキーとしてソートすることは容易である。

4.1.4　単位データベースの運用

この節では，単位データベースを運用していく際に，定常的に行っていた事柄について述べることにする。なお，人手単位解析作業自体については，4.3 節で詳しく述べることにする。

4.1.4.1　単位データの整合性チェック

4.1.2.2 節で述べたように，人手単位解析時に付与される形態論情報に対しては，単位データベース側では制約を設けない。そのかわり，修正ツールによる整合性チェックと，スクリプトによる単位データ全体に対する整合性チェックを行う。

■修正ツールによる整合性チェック　整合性のチェックは，単位データベースに対する変更が行われる際に，次の項目に対して実施される（ただし，短単位のみ）。

- 品詞，活用の種類，活用形（3 章で規定されている値かどうか）
- 代表形，代表表記（単位辞書に登録されている値かどうか）
- 出現形，発音形（基本的に変更不可）
- 講演 ID，タイムスタンプ（変更不可）

上記の制約に違反があった場合は，単位データベースに変更が反映される直前に，警告だけを発し，変更自体は許可した（講演 ID，タイムスタンプは変更不可）。例えば，未出の単位を人手解析する場合，代表形や代表表記などの付与情報が「単位辞書に未登録である」という警告のみ行い，変更自体は許可される。

■スクリプトによるチェック 単位データベース全体に対して，Perl スクリプトを用いて，整合性のチェックを行う。チェックの実施時期は，1 ヵ月に一回程度であるが，単位データベースを他の作業グループ（転記テキストグループや情報通信研究機構）へ送付する際には必ず実施する。チェックの種類は，大きく分けて，次の 3 種類がある。

- 単位データ形式上の整合性
- 単位辞書との整合性
- 転記テキストとの差分

このうち，転記テキストとの整合性については，次節（4.1.4.2 節）で詳しく述べることとし，単位データ形式上の整合性，単位辞書との整合性について説明する。

まず，単位データ形式上の整合性のチェックとは，単位データの物理的仕様，形態論情報の仕様に関するチェックである。具体的には，次の 3 項目がある。

- 単位データに使用されている文字集合。規定された文字集合に含まれない文字がないかチェックする（例：機種依存文字や，転記テキストのタグ以外で出現する ASCII 文字）。
- 出現形フィールドの文字数と転記情報中の短単位位置情報。例えば，転記情報フィールドの値が，

  ```
  0202 00500.324-00501.003 L:-001-001
  ```

  で，出現形の長さが 2 文字（4 バイト）の場合，同一転記文節内で後続する短単位の転記情報フィールドは，次のようにならなければならない。

  ```
  0202 00500.324-00501.003 L:-001-005
  ```

- 形態論情報の形式： 3 章の付録 3.5 の「品詞情報一覧」の形式に一致しているかをチェックする。この際，品詞と「活用の種類」フィールドなど複数のフィールドを組合せも考慮する。例えば，形容詞のうち，活用の種類が文語形容詞型の場合は，活用形として「仮定形」が付与されていた場合は，誤りであると判定される。

もう一つは，単位辞書との整合性である。まず，短単位だが，4.4 節で示すように，短単位辞書は次の情報を保持している。これらの情報と単位データベース中の単位データとの整合性をチェックする。

- 代表形，代表表記
- 品詞，活用の種類
- 出現形（活用語の場合は，終止形）

単位データの出現形については，活用の種類と活用形の情報から，出現形を実際に活用させた上で，マッチングをとる。例えば，単位データの出現形が「食べ」で，活用の種類が「下一段活用」の場合は，単位辞書の出現形を下一段活用させてマッチングを試みる。

長単位の場合は，上記の情報のマッチングに加えて，構成要素の短単位のチェックを行う。長単位辞書には，長単位の構成要素となる短単位が活用形を含めて記述されている。例えば，長単位「に関して」は，短単位列「に/関し/て」から構成されることが記述される。この際，「関し」は「関する」の連用形であることもあわせて記述されている。ただし，長単位辞書は短単位辞書が完成したのちに作成されることから，実際に適用されたのは，プロジェクト後期になってからであり，一般公開もされなかった。したがって，本報告書でも簡単にふれるにとどめる。

以上のように人手単位解析時には緩い制約を用い，別途，スクリプトにより単位データ全体をチェックする方法を用いたのは，単位の仕様や単位辞書が単位データベースの構築とともに確定していったことによる。特に，単位辞書に関しては，プロジェクト初期には十分な量が存在せず，人手単位解析時に機械的なチェックを行うのが困難だった。そこで，形態論情報の仕様や単位辞書の整備がある程度確定した段階で，スクリプトにより一括して単位データ全体をチェックした。さらに，整合性の取れていない単位データを抽出し，人手修正した。この方法の利点は，次の 2 点である。

- 単位データの人手解析時に，単位辞書の管理者と修正者とのやり取りを軽減できる。例えば，人手単位解析時に付与情報に対する厳密なチェックを行うとすると，単位辞書に未登録の単位が出現した場合，単位辞書への登録をそとつど行わなければならない。しかし，上述の「修正ツールによる整合性チェック」のところで述べたように，修正時には警告を表示するのみにとどめることにより，単位辞書の管理者に問い合わせすることなしに人手単位解析を継続できる。
- 仕様上不整合な状態の単位データを一括してチェックすることになるので，単位の仕様に関して，全体的な整合性を取りつつ，単位データベースを修正したり，単位辞書を更新していくことが可能になる。人手単位解析時にそのつど形態論情報の仕様や単位辞書を修正すると，場あたり的な対処になり，全体的な整合性を取るのが難しくなる恐れがある。

4.1.4.2　転記テキストの修正と同期

すでに述べたように，単位データベースには転記テキストに含まれる情報をすべて含んでいる（ただし，コメントは除く）が，単位データベースと転記テキストの構築は，並行して行われていたので，両者の同期を取る必要があった。同期を取る必要があるのは，次の場合である。

- 転記テキストの確定版がリリースされた場合。単位データベースには，転記テキストグループがリリースする確定版の転記テキストを利用し，新しい確定版がリリースされた時に，それに同期する。より具体的に言えば，単位データベースの講演 ID，転記情報，出現形，発音形フィールドの値が同期対象となる。さらに，出現形，発音形が変更されることにより，前文脈・後文脈フィールドに変更が加わる。また，場合によっては，代表表記や品詞などの形態論情報の変更が必要になる場合もある。実際の変更は，次のように行われる。
  - 同期に際しては，形態論情報に影響を与えないものに関しては，機械的に単位データベースに変更を適用する。例えば，前文脈，後文脈フィールド，タグなし出現形フィールド，転記情報フィールドの文節情報などがこれに相当する。
  - 上記以外の変更については，形態論情報に変更がないか人手でチェックしつつ，単位データベースに変更を加える。

- 単位データベース構築の過程で転記テキストに誤りが発見された場合。例えば，誤字，脱字，表記の不統一などは，転記テキスト作成グループに報告され，修正の可否については，転記テキスト作成グループが判断し，その場では修正しない。修正が受理されれば，次回にリリースされる確定版との同期において，単位データベースに修正が加わる。報告の仕組みの詳細については，4.3.8 節で述べる。

なお，単位データの整合性チェックのため，定期的に単位データベースと確定版の転記テキストとの差分をとる[*5]。具体的には，単位データベースの講演 ID，転記情報，出現形，発音形から転記テキストを再生成し，両者の差がないかチェックする。

4.1.4.3 索引

単位データの参照を高速化するために，関係データベース管理システムの「索引」機能を用いる。索引づけは通常，フィールドごとに行われるが，過度な索引づけは，更新速度の低下[*6]と索引ファイルの増大につながる。そこで，検索キーとしての使用頻度が高い，次のフィールドに索引づけを行った。

- ID，後続 ID
- 講演 ID
- 出現形，代表形，代表表記，タグ無し出現形
- 発音形
- 品詞，
- 代表形 [長]，代表表記 [長]
- 予備 1 〜 3
- 最終更新時間

[*5] 通常，人手解析によって転記テキストと単位データベースの差分が発生する事はないが，修正ツールや保守用のスクリプトのバグなどにより差分が発生する可能性があるため，単位データベースと転記テキストとの差分のチェックを行う。

[*6] 更新する際に，索引づけを再度行うため，更新速度が低下する。

4.2　人手解析による短単位・長単位データの構築

本節では，人手解析により，短単位・長単位データを構築する方法について説明する。

4.2.1　構築する単位データの全体像

4.1.1 節で示した 4 種類の単位データのうち，本節で扱うのは，次の二つの人手解析単位データである。

- 人手解析単位データ（短単位）
- 人手解析単位データ（長単位）

人手解析単位データの構築に際して，焦点を当てるのは，構築の効率と精度である。精度向上のために，人手によるチェック・修正を前提にするとはいえ，すべてを人手で実施するのは，非効率である。そこで，既存の形態素解析システムなどの計算機システムを援用しつつ，効率的，かつ，高精度に人手単位解析する方法を考えることにする。より具体的に述べると，まず，既存の形態素解析システムで単位データの初期値を作成し，そのデータを人手修正する方法を用いる。本節では，単位データの初期値を効率的に高精度で作成する手順を示す。作成した単位データの人手修正については，4.1 節の図 4.1 に示した修正ツールを使うことになるが，このシステムについては，別途 4.3 節で解説することにする。

4.2.2　構築方法の概略

人手による，短単位と長単位の解析方法は，次のような流れで行う。

(1) 既存の形態素解析システムにより転記テキストを解析し，人手解析短単位データの初期値とする。
(2) 修正ツールを使用し，(1) のデータに対して，人手チェック・修正を行う。
(3) 短単位から長単位への変換規則に基づいた構文解析システムにより，(2) の短単位データ列を長単位解析し，長単位データの初期値とする。
(4) 修正ツールを使用し，(3) のデータに対して，人手チェック・修正を行う。

このように，まず，短単位の解析を行い，その後，長単位の解析を行う。この一つの理由は，既存の形態素解析システムが採用している「形態素」[*7]が短単位のほうに近く，短単位となるように人手修正したほうが，手間を少ないからである。もう一つの理由としては，長単位は一つ以上の短単位から構成されるので，(3) のように，短単位データから長単位データをある程度自動的に構成できることが挙げられる。

全体的には，以上の流れで構築が進んでいくが，(2) で短単位データが完成するわけではなく，(3) (4) と単位データの構築が進んだ後でも，さまざまな観点からの人手チェック・修正が継続して行われる。したがって，短単位のデータと長単位のデータの整合性（例えば，長単位の活用語の活用形は，それを構成する末尾の短単位の活用形と一致する必要がある）を取りつつ，人手チェック・修正を行うことになる。

[*7] 自然言語処理システムにおける「形態素」とは，多くの場合，解析用辞書に記述されている辞書項目のことを指す。

4.2.3　人手短単位解析の手順

4.2.3.1　概要

本節では，人手で短単位解析を行う手順を解説する。人手短単位データの構築においては，すべての短単位データに対して，人手でチェック・修正を行い，高精度なデータを目指す。図 4.7 に解析の流れを示す（図中の番号は処理の順番を表す）。

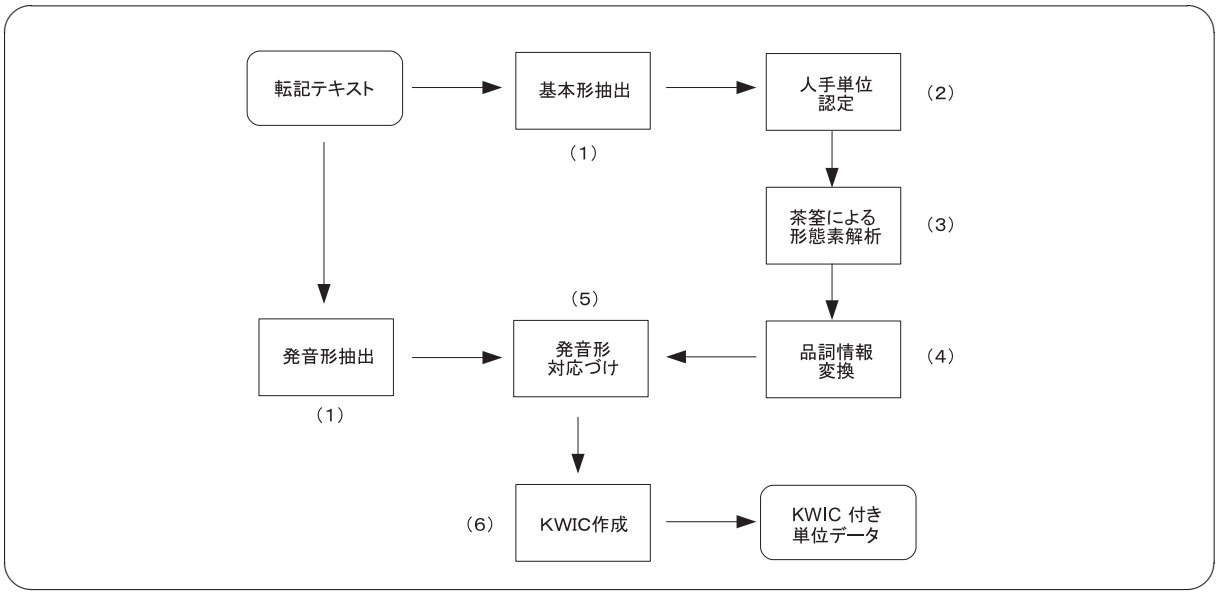


図 4.7　人手短単位解析の手順

図 4.7 のとおり，まず，転記テキストの基本形と発音形を分離し，基本形に対して人手で短単位認定を行う。そして，認定された短単位の境界を反映させつつ，形態素解析システム「茶筌」（奈良先端科学技術大学院大学が開発）[*8] により解析を行う。さらに，「茶筌」の品詞情報（品詞体系）から短単位の品詞情報への変換を行う。次に，基本形の単位境界位置から発音形側の単位境界位置を DP マッチングにより推定し，両者を対応づけする。最後に，個々の短単位に KWIC を付与し，単位データベースに登録する。個々の処理については，この後の節で順次説明する。

4.2.3.2　人手短単位認定

まず，転記テキストの基本形に対して，人手で短単位の境界位置を指定する。短単位の境界位置の認定を計算機で行わずに人手で行ったのは，既存の形態素解析システムでは，次の理由から単位認定の精度が十分でないことが予想されたためである。

- 今回使用した「茶筌」を含め，既存の形態素解析システムの単位の認定方法が短単位と異なること
- さらに，「茶筌」は新聞などの書き言葉を主な解析対象にしており，話し言葉特有の現象や文法に対応していないこと

[*8] http://chasen.naist.jp/hiki/ChaSen/

精度が十分でない場合，短単位境界の人手修正が多発することになる。短単位境界の人手修正は，短単位の再結合／再分割と品詞情報の再付与を行わなければならず，人手修正の手間が大きい。そこで，短単位境界は人手であらかじめ認定しておいた上で，品詞情報を計算機で付与することにした。

図 4.8 に短単位認定の例を示す。短単位の境界は，短単位の末尾で指定する。短単位境界を指定するための記号は，‘/’ とした。

```
0261 00671.207-00671.659 L:
(F えー)/(F い)/
0262 00672.450-00673.772 L:
ごめん/なさい/
これ/
間違い/です/
(D エルエルー)/
0263 00673.991-00676.025 L:
(A エル/エル/ワイ；ＬＬＹ)/です/ね/
上下/方向/です/
すい/ませ/ん/
```

図 4.8　単位境界認定の例

実際の短単位認定作業は，図 4.8 のように転記テキストから基本形だけ抽出したファイルをテキストエディタで編集することにより実施した。プロジェクト初期に行った作業量見積りのための試算結果であるが，人手解析速度は 1 人/日当たり約 8000 短単位，精度は約 99% であった。

4.2.3.3　「茶筌」による形態素解析

次に人手で短単位認定した結果に基づき，「茶筌」で解析する。ただし，転記テキストには，さまざまなタグが付与されているので，転記テキスト中のタグの情報を利用しつつ，「茶筌」で扱える形に整形して形態素解析する。形態素解析は，転記テキストの文節単位で行う。

前処理，後処理を含めた形態素解析の手順は，次のとおりである。

(1) 転記テキスト中のタグの処理

ここでは，例として，次の転記テキストを形態素解析することを考える。

```
(F えー)/
(A 千/二十/四；１０２４)/(D ん)/点/の/
```

「茶筌」では，転記テキストの中のタグ自体を通常の発話文と同様に扱ってしまうため，転記テキストから解析対象の部分を抽出して，「茶筌」に入力する必要がある。また，言いよどみのように，文中に挿入される断片的な要素は，発話文から取り除いたのちに解析したほうが解析精度の向上が望める場合もある。そこで，上記の例の場合は，次のように処理することになる。

- （F **えー**）からタグ（F）を除去し，「**えー**」を解析対象にすること
- （A **千/二十/四；１０２４**）から解析対象の文字列「**千二十四**」を取り出すとともに，言いよどみの（D **ん**）を解析対象から除外し，「**千/二十/四/点/の**」が解析対象になるようにする。

上記の処理を行うために，次のように，転記テキスト中のタグごとに処理方法を変えて形態素解析することにした。

タイプ１ ... **タグ**（F）： フィーラーは，発話文中に挿入される要素であり，フィラーを取り除いてから形態素解析を行うことが好ましい。ただし，フィラーの中にも複数の短単位からなるものが存在する。そこで，フィラーと残りの文字列をそれぞれ独立に形態素解析する。それぞれの形態素解析結果は，(3) で統合される。上の例のタグ（F）の場合，（F **えー**）の contents である「**えー**」と，（F **えー**）抽出後の残りの文字列（この例の場合は，空文字列となるが）をそれぞれ独立に形態素解析する。

タイプ２ ... **タグ**（A），（D2），（K），（M），（O），（R），（W），（?）： これらの転記テキスト中のタグの contents は，文の一部となる。そこで，転記テキスト中のタグの contents から形態素解析対象の文字列を抽出[*9]して，タグ前後の文字列と連結させた上で形態素解析を行う。上の例の（A **千/二十/四；１０２４**）の場合，「**千/二十/四**」を取り出し，前後の文字列と合わせて形態素解析する。タグ（D）の処理と合わせると，形態素解析対象の文字列は，「**千/二十/四/点/の/**」となる。

タイプ３ ... **タグ**（D），（?）： これらもフィラーと同様，発話文中に挿入されるタグであるが，フィラーと異なり，contents を形態素解析する必要がない。そこで，転記テキストのタグを抽出し，抽出後の残りの文字列だけを形態素解析する。上の例では，（D **ん**）を取り除いた上で，残りの文字列を形態素解析する。（D **ん**）は，「形態素」として認定するが，品詞（言いよどみ）以外の形態論情報は付与しない。また，ここで扱うタグ（?）は，「(?)」のように，推定されている文字列が存在しないものに限り，便宜的に何も形態論情報を持たない短単位として処理する。

以上の処理を行うと，次の (2) において，発話文を断片的に形態素解析することになるが，この後の (3) でそれらを統合する。

(2) 形態素解析の実行

4.2.3.2 節の単位認定の結果にしたがうように，(1) で決定された文字列を「茶筌」で解析する。なお，解析に利用した「茶筌」，および，ipadic[*10]のバージョンは，それぞれ ver.2.0，ver.2.2.9 である。

「茶筌」は，解析対象の文字列中の「形態素」境界の区切り記号[*11]を利用しつつ解析することができるので，この機能を使用する。ただし，(1) で指定した短単位境界以外でも分割してしまう可能性がある。この場合は，過分割が起こらなくなるまで次解を求める。過分割が起こらない解が見つからない場合は，当該の短単位の品詞を未知語として処理する。

「茶筌」による解析の結果は，1 形態素 1 行のタブ区切りテキストとなる。結果に含まれるのは，次の情報である。短単位に最終的に付与される形態論情報との対応関係とともに示す。

[*9] ここで扱うタグ（?）は，（? **タオングー**）のように，推定されている文字列が存在する場合である。（? **体積，堆積**）のように，複数の候補が列挙されている場合は，第一候補を形態素解析する。存在しない場合は，タイプ 3 のタグとして扱う。

[*10] 「茶筌」の解析用辞書

[*11] 「茶筌」の区切り記号はスペースやタブなどの空白文字なので，「茶筌」解析時には区切り記号を '/' からスペースに変換する。

- **出現形**　... 短単位の基本形である（ただし，(1) で述べたように，タグは除去されている）。
- **品詞**　... 短単位の品詞の初期値として利用する。
- **活用型**　... 短単位の「活用の種類」の初期値として利用する。
- **活用形**　... 短単位の活用形の初期値として利用する。
- **読み**　... 短単位の代表形の初期値として利用する。また，転記テキストの基本形と発音形とをマッチングさせるための基礎情報として利用する。
- **基本形**[*12] ... ipadic の見出し。短単位の代表表記の初期値として利用する。

これらの情報のうち，品詞，活用型（活用の種類），活用形については，「品詞情報の変換」（4.2.3.4 節参照）で自動的に短単位の品詞情報に変換する。「茶筌」の「読み」情報は，「茶筌」の解析用辞書に基づく当該「形態素」の読みだが，転記テキストの基本形と発音形との対応づけ処理（4.2.3.5 節）のための基礎情報として保持しておく。さらに，「茶筌」の基本形は短単位の代表表記の初期値として保持しておき，短単位の辞書が整備されてきた段階で変換表を作成し，統一を図った[*13]。

(3) タグ情報の再現

(2) の形態素解析のために (1) において削除した転記テキストのタグを復元するとともに，個別に形態素解析された，タグタイプ 1，3 の形態素をもとの位置に戻す。例えば，上の例の場合，次の処理が行われる。

- 「**えー**」と「**千二十四**」から除去されたタグ (F) とタグ (A) を復元する。「**千二十四**」については複数の短単位から構成されるが，先頭と末尾の短単位の基本形にタグが復元され，それぞれ「(A **千**」と「**四；１０２４**)」となる。
- (D **ん**) を「(A **千二十四；１０２４**)」と「**点**」の間に戻す。

4.2.3.4　品詞情報の変換

次に，「茶筌」の解析結果の品詞体系（つまり，ipadic の品詞体系）から短単位の品詞情報に変換する。変換には，品詞（71 項目），活用形（26 項目），活用の種類（78 項目）の三つの変換表を用いた。それぞれの変換表の一部を表 4.2，4.3，4.4 に示す。品詞，活用の種類，活用形の記述は，ほとんどの場合，「茶筌」のほうが詳細なので，一対一の対応関係を記述できる。ただし，次のように短単位の品詞情報のほうが詳細な場合は，確実に人手チェックの対象となるようにする。

- 「茶筌」の活用型が一段動詞の場合（短単位の上一段動詞，下一段動詞のいずれかに対応する）
- 「茶筌」の活用形が基本形[*14]の場合（短単位の終止形，連体形のいずれかに対応する[*15]）
- 「茶筌」の品詞が未知語の場合

[*13] 「茶筌」の基本形は，活用語の場合は，終止形となるが，非活用語の基本形は，当該「形態素」の表記どおりであり，表記が異なれば，別語となる。他方，短単位の場合，「おっきい」も「大きい」も代表表記は「オオキイ」となる。

[*14] 「茶筌」は終止形と連体形を識別せずに，「基本形」として解析する。

[*15] 初期値として，後節する短単位が名詞，もしくは，形状詞「よう」の場合は，連体形，それ以外は終止形とした。

表 4.2　品詞の変換表の例

「茶筌」	短単位（品詞）	短単位（その他の情報 1）
名詞-一般	名詞	
名詞-代名詞-一般	代名詞	
名詞-固有名詞-一般	名詞	固有名詞
名詞-数	名詞	数詞
名詞-接尾-一般	その他	接尾辞
動詞-自立	動詞	
助詞-格助詞-一般	助詞	格助詞
フィラー	感動詞	
未知語	その他	未知語

表 4.3　活用形の変換表の例

「茶筌」	短単位
未然形	未然形
未然ウ接続	未然形
連用形	連用形
連用タ接続	連用形
仮定形	仮定形
基本形	終止連体形
ガル接続	語幹
命令 e	命令形
命令 i	命令形

表 4.4　活用の種類（活用型）の変換表の例

「茶筌」（品詞）	「茶筌」（活用型）	短単位（品詞）	短単位（活用の種類）	その他の情報 2
動詞-自立	カ変・クル	動詞	カ行変格	
動詞-自立	五段・カ行イ音便	カ行五段		イ音便
動詞-自立	四段・カ行	文語カ行四段		文語
動詞-自立	一段	一段		

4.2.3.5　基本形と発音形との対応づけ

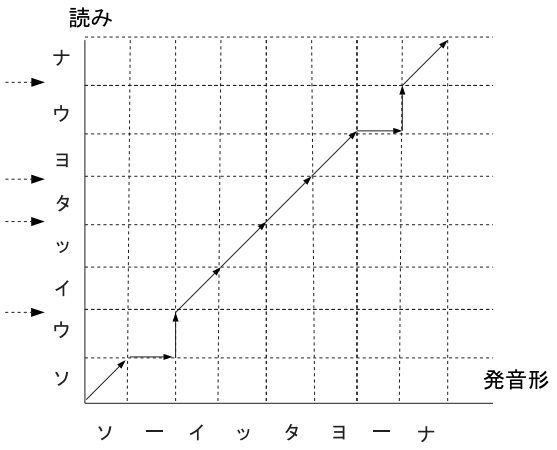
基本形と発音形との対応づけは，図 4.7 の「発音形抽出」と「発音形対応づけ」の処理に相当する。ここでは，短単位に分割した転記テキストの基本形と，まだ分割されていない発音形とを対応づける処理を行う。対応づけには，「茶筌」による解析結果として付与された各形態素の「読み」情報と発音形とを DP(Dynamic Programming) マッチングすることにより行う。DP マッチングを用いることにより，発音形と読みとの間の表記上の差異や解析誤りを考慮した上で，基本形と発音形の対応づけを行うことが可能になる。ここでは，DP マッチングの動作を例で説明することにする。図 4.9 は，短単位列「そう/いっ/た/よう/な」と発音形の文字列「ソーイッタヨーナ」を対応づける例である。短単位の区切り位置は，前節までの処理で認識され（縦軸に付与された矢印が区切り位置を表す），各短単位には，「茶筌」の読み情報が付与されているので，これを発音形と DP マッチングさせる。図 4.9 の横軸は発音形，縦軸は「茶筌」の読みである。それぞれの軸の個々の枠には，読み，発音形から 1 文字ずつ割り当てられる。

図 4.9 中の矢印（実線）は，マッチングを表す。一つの斜めの矢印は，発音形と読みの 1 文字が一致したことを表す。縦方向の一つの矢印は，読みの 1 文字が発音形の 0 文字とマッチしたことを表す。一方，横方向の一つの矢印は，発音形の 1 文字が読みの 0 文字と一致したことを意味する。

対応づけは，DP マッチングを用いて，図 4.9 の原点から終点（座標は（発音形の長さ，読みの長さ））までの最も「短い」経路を探索することにより行う。今回は，非常に単純だが，矢印の数を経路の長さとして定義した。図 4.9 で示した矢印の経路は，すべての経路の中で最も短いものである。対応づけの位置は，次のように決める。

- 「読み」軸上の読みの分割位置（点線の矢印）から，「発音形」軸に対して並行に直線を引き，経路の矢印（実線）と交わった点の「発音形」座標値が「発音形」の分割位置になる。
- ただし，経路の矢印と並行になって重なった状態になる場合は，並行ではない矢印と交わるところまで分割位置を伸ばす。これは，経路を選ぶ上で，→↑ も ↑→ も経路の長さが同じだからである。

したがって，読みの「ソウ」の分割位置と対応づけられるのは，発音形の「ソー」ということになる。同様に，「イッ/タ/ヨー/ニ」と対応づけられる。このように，表記の違いを吸収した上でマッチングを行うことができる。



基本形	読み（茶筌）	発音形
そう	ソウ	ソー
いっ	イッ	イッ
た	タ	タ
よう	ヨウ	ヨー
な	ナ	ナ

図 4.9　DP マッチングの例 1

図 4.10 の例は，解析誤りに対応している例である。この例では，「心的な」を「ココロ/テキ」と解析してしまっているが，適切に発音形が対応づけられている。図 4.11 は，転記テキストのタグの処理例である。この例では，発音形側にタグ (W) が含まれているが，適切に対応づけがなされている。

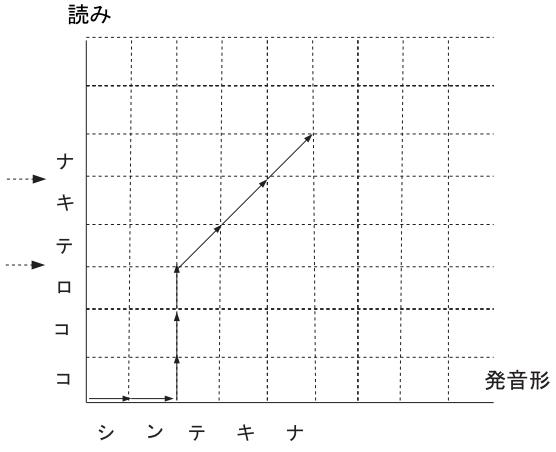


図 4.10　DP マッチングの例 2

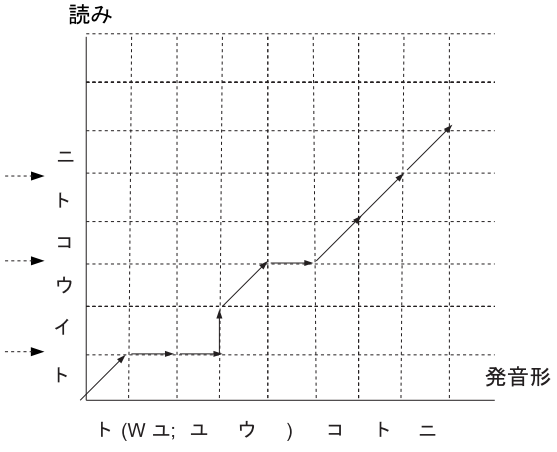


図 4.11　DP マッチングの例 3

なお，文字の対応づけには，次のように，読みと発音形の差異を吸収するためのルールと，転記テキストのタグのマッチングに関する特別なルールを設けている。

- 転記テキストのタグと単位解析外の属性（例：(W ユ; ユウ) の「(W ユ;」と閉じ括弧）はまとめて 1 文字として扱う。
- 読みが「ヲ」，発音形「オ」の場合は，一致させる
- 転記テキストのタグの開き括弧とタグ自体，タグの属性に相当する文字列については，後接する文字列に含める。例えば，図 4.11 では，タグ (W) と付随する属性である「(W ユ;」の部分は，後接する「ユ」に含める。
- 転記タグの閉じ括弧は，前接する文字列に含める。例えば，図 4.11 では，タグ (W) の閉じ括弧は，前接する文字「ウ」に含める。

4.2.3.6 単位データベースへの登録

以上の処理によって，短単位に付加されるすべての情報が揃い，転記テキストの出現順に並んだ，1 短単位 1 行のタブ区切りのテキストになる。単位データベースには，次の情報を付け加えた上で登録する。

- 当該短単位の講演 ID
- 当該短単位の転記情報（発話 ID，開始・終了時間，基本単位中の行，桁情報）
- ID，および，後続 ID
- 最終更新者（初期値は admin），および，最終更新時間（初期値は，単位データベースに登録する時間）

なお，プロジェクト初期段階の小規模な調査結果であるが，単位データベースに登録される短単位データの解析精度は，約 78% であった。調査時のサンプル数は 1995 短単位である。誤りの内訳は次のとおりである。括弧内の数値は，誤り全体に対する割合を示す。

- 終止・連体形の区別 ... 194 (44%)
- 一段動詞の活用の種類の区別 ... 49 (11%)
- 未知語 ... 43 (10%)
- 発音形の対応づけ誤り ... 41 (9%)

ただし，上記の結果は，品詞情報の変換において，必ずチェック対象となる情報（終止・連体形の区別，一段動詞の活用の種類の区別）については，誤りと判定している。したがって，実際の解析精度はこの結果よりも高くなる。終止・連体形の区別，一段動詞の活用の種類の区別による誤りを除けば，解析精度は約 90% となる。

4.2.4 人手長単位解析の手順

4.2.4.1 BUP による長単位解析

ここでは，解析済みの短単位データ列から，長単位データを生成する方法について説明する。

長単位の生成には，BUP システム[*16] を用いる。まず，文脈自由文法を拡張[*17]した DCG（Difinte Clause Grammer，確定節文法）で長単位の構造を解析する規則（以後，「長単位解析規則」と表記）を記述する。次に，BUP Translator により，DCG を左隅構文解析を行う Prolog プログラムに変換する。長単位の生成には，変換された構文解析プログラムを BUP Starter から呼び出し，転記テキストの文節ごとに短単位データ列を解析する。使用した Prolog の処理系は，SWI-Prolog (ver.3.30) である。

次に，DCG による長単位解析規則の記述方法を説明する。DCG は，次のように書き換え規則で記述する。非終端記号は，引数を持つこともできる。この例では，「**非終端記号 1**」は，「**非終端記号 2**」と「**非終端記号 3**」... に書き換えられることを意味する。なお，右辺には，非終端記号だけでなく，終端記号も記述することができる。

非終端記号 1 --> **非終端記号** 2，**非終端記号** 3 ...

実際に使用した DCG の一部を図 4.12 に示す[*18]。まず，規則 1 は，「名詞句」は，「接頭辞」と「名詞」に書き換えられることを意味する。「名詞句」「接頭辞」が非終端記号に相当し，それぞれ引数を持っている。引数は 9 個あり，左から解析木，および，単位に付与される形態論情報，つまり，品詞，活用の種類，活用形，その他の情報 1，2，3，代表形，代表表記である。引数中のアルファベットで始まる記号（例：B1）は変数を表す。同一規則内では，同名の変数は，同一の値を取らなければならない。例えば，規則 1 の右辺の変数 B1 は，左辺の非終端記号「名詞句」の引数 B1 と同じ値を取る。変数の値は，構文解析の過程で決定（unification）されていく。長単位解析結果として得られる形態論情報は，構文木の root の非終端記号「長単位」の引数ということになる。

規則 6 は複合辞「という」の規則である。この規則のように辞書的な規則を記述する場合は，引数の値に変数ではなく，特定の値を指定する。規則 6 の場合は，短単位「と」「言う」の品詞などを直接記述しておく。

上記の長単位解析規則を使って，「御客さんが」を解析した結果を図 4.13 に示す。この例では，「御客さん」と「が」の二つの長単位に解析される。「御客さん」の解析には，規則 1〜4 が適用され，「が」には規則 5 が適用される。図 4.13 の各ノードの横には，解析中に適用された規則番号を付記しておいた。

次に規則適用の過程を少し詳しく見てみる。例えば，図 4.13 のノード A では，規則 1 が適用される。規則 1 により，規則 1 の左辺の品詞に関連する引数 B1，B2，B3，B4a，B4b，B4c は右辺の非終端記号「名詞句」の引数と同一の値になる。これにより，ノード A の非終端記号「名詞句」は，「客」の品詞に関連する情報を引き継ぐことになる。一方，代表形と代表表記に相当する引数 A5+B5，A6+B6 は，右辺の非終端記号「接頭辞」「名詞句」の代表形，代表表記を連結した値になる。したがって，それぞれ「オ + キャク」「御 + 客」となる[*19]。

実際に長単位解析を行う際には，長単位解析結果の曖昧性が発生する場合もある。例えば，短単位列「観光/客」（'/' は短単位の分割位置）を上記の長単位解析規則で解析すると，「観光|客」と「観光客」（'|' は長単位の分割位置）の 2 通りの解析結果が得られる。なぜなら，規則 4 により名詞句「観光」も長単位になりうるからである。

この問題を解決するために，ここでは，「できるだけ長い長単位となるように解析する」というヒューリス

[*16] http://chasen.naist.jp/bup.html

[*17] 文法規則の中に補強項と呼ばれる Prolog プログラムを記述することができる。

[*18] 説明のため，一部を簡略化している。

[*19] 解析結果の代表形，代表表記に対しては，後処理を行う必要がある。例えば，複合動詞の前項の動詞は，代表形，代表表記を連用形に活用させたうえで連結する。

```
名詞句(名詞句(A+B),B1,B2,B3,B4a,B4b,B4c,A5+B5,A6+B6) -->
        接頭辞(A,A1,A2,A3,A4a,A4b,A4c,A5,A6),
        名詞句(B,B1,B2,B3,B4a,B4b,B4c,B5,B6).                    ..... 規則1
名詞句(名詞句(A+B),A1,A2,A3,A4a,A4b,A4c,A5+B5,A6+B6) -->
        名詞句(A,A1,A2,A3,A4a,A4b,A4c,A5,A6),                    ..... 規則2
        接尾辞(B,B1,B2,B3,B4a,B4b,B4c,B5,B6).
名詞句(名詞句(A+B),A1,A2,A3,A4a,A4b,A4c,A5+B5,A6+B6) -->
        名詞(A,A1,A2,A3,A4a,A4b,A4c,A5,A6),                      ..... 規則3
長単位(長単位(A),A1,A2,A3,A4a,A4b,A4c,A5,A6) -->
        名詞句(A,A1,A2,A3,A4a,A4b,A4c,A5,A6).                    ..... 規則4
長単位(長単位(A),A1,A2,A3,A4a,A4b,A4c,A5,A6) -->
        助詞(A,A1,A2,A3,A4a,A4b,A4c,A5,A6).                      ..... 規則5
助詞(助詞(Z0+Z1),助詞,_,_,格助詞,_,連語,トイウ,という) -->
        助詞(Z0,助詞,_,_,格助詞,_,_,ト,と),
        動詞(Z1,動詞,ワア行五段,連体形,_,_,_,イウ,言う).          ..... 規則6
```

図 4.12 DCG による長単位解析規則の例

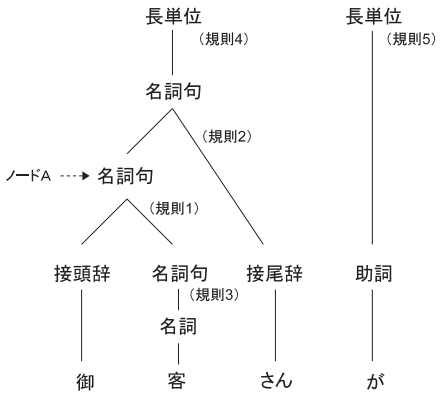


図 4.13 BUP による長単位解析例

ティクスを用いた。具体的な処理方法としては，解析対象の短単位列を末尾から徐々に短くしていき，最初に解析に成功した長単位を解析結果とした。「御客さんが」の解析は，次の手順で行われる。

(1) 「御客さんが」を対象に長単位解析
(2) 全体としては長単位として認定できず，解析に失敗
(3) 1 文字末尾の文字を削除し，「御客さん」で長単位解析
(4) 解析に成功。長単位として認定
(5) 残りの「が」を長単位解析
(6) 解析に成功。長単位として認定

なお，今回は，長単位解析結果の曖昧性を解消する手段として，上記のヒューリスティクス以外の方法を用

いていない。そのため，長単位内部の構造に曖昧性があるが，多くの場合，長単位に付与される情報に違いがないことが多い。例えば，「御客さん」の場合，上記の長単位構成規則で解析を行うと，構造上，**[[御 客] さん]**，**[御 [客 さん]]** の 2 通りの結果が考えられるが，全体としての長単位の付与情報には差がない。

4.2.4.2 単位データベースへの登録

前節の長単位解析が済んだデータは，単位データベースに登録する。単位データベースに登録するのは，次の情報である。なお，最終更新者と最終更新時間については，短単位の人手修正に関する情報がすでに格納されているので，更新しない。

- 長単位解析結果（品詞，活用の種類，活用形，その他の情報 1〜3，代表形，代表表記）
- 後続 ID [長]

4.3 単位データベース修正ツール

4.3.1 概要

本節では，単位データベース修正ツール（以後，「修正ツール」と表記）の設計，実現された機能を示すとともに，修正ツールの運用方法についても述べることにする。

修正ツールは，単位データベース管理システムに対するクライアントとして動作し，単位データの検索や修正を行うことができる。人手単位解析において，単位データベースへのアクセスは，通常，この修正ツールを介して行われる。修正ツールの主な特徴は，次のとおりである。

複数の修正者による単位データベースの修正： 単位データベースに不整合を起こすことなく，複数の修正者が同時に単位データを修正できる。

表計算ソフトウェアを用いた閲覧と修正： 単位データの閲覧と編集に表計算ソフトを用いることにより，修正者が容易に操作方法を覚えることが可能である。また，表計算ソフトウェアに付随するソートや絞込みなどの機能を活用することができる。

GUI を用いた検索： GUI を用いた検索条件入力インターフェイスを備えているため，SQL を知らない修正者でも容易に単位データベースを検索できる。

4.3.2 設計

本修正ツールを設計するにあたって，まず，既存の類似するシステムを概観することにした。自然言語処理の分野では，1990 年代始めごろからコーパスの利用が盛んに行われはじめた。そのため，形態素解析結果を人手修正するシステムとして，タグ付きコーパスの作成支援を行うシステムである VisualMorphs（松田 2000）や ViJUMAN （山下 1996）などが提案されている。また，近年提案されたものとして，タグ付きコーパスを格納・検索するツール「茶器」（松本 2004）がある[*20]。

しかし，構築対象のコーパスの規模が大きくなると，上記の提案システムが考慮していない，次の問題が発生する。

(1) 大量の単位データを修正するため，複数の修正者が同時に単位データベースにアクセスすることが前提となり，単位データベース全体の整合性を取る仕組みが必要になる。

(2) 修正者は多くの場合，言語学の素養を持っているが，必ずしもコンピュータの専門家であるとは限らない。したがって，SQL コマンドなど，コンピュータに関する専門的な知識を使わないで，単位データベースを検索したり，単位データベースの内容を修正できなければならない。

そこで，本修正ツールでは，次の 2 点に考慮して設計を行った。

(1) 単位データベースの論理的・言語情報的な整合性を維持しつつ，複数の修正者が同時に人手単位解析できるようにすること

(2) 誰でも容易にデータベースを検索・閲覧・修正できること

[*20] 設計当時は，「茶器」はまだ存在しなかった。

まず，1 点目に対しては，次の三つの仕組みを用意した。

- 関係データベース管理システムによる排他制御と，個々の単位データに付与された最終修正時刻を利用した排他制御により，単位データベース全体の整合性を保持する。
- 多数の用例を比較できる KWIC 表示機能を備えることにより，言語的な分析を効率化し，同一の単位に異なる形態論情報が付与されるなどの不整合が起きないようにする。
- 単位辞書を利用して，付与情報の言語的な整合性チェックを行う。
- ニュースシステムを導入し，人手単位解析作業の管理者から一般の修正者への情報伝達や，解析誤りの指摘などのやりとりを円滑に行えるようにする。ニュースシステムは修正ツール自体の機能ではないが，高精度の単位データを構築するための補助的なシステムとして位置づける。

次に，2 点目に関しては，次の機能を持たせる。

- 検索に関しては，GUI により，単位データベースへの問い合わせを実現することで，簡便性を確保する。なお，GUI では実現できない複雑な問い合わせを必要とする場合や大量の検索結果を扱わなければならない場合を考慮して，管理者が検索した結果を一般の修正者へ容易に配布できるようにする。
- 単位データの閲覧・修正の機能に関しては，人手修正の機能を表計算ソフトウェア上に実現する。表計算ソフトウェアは，（計算機の専門家ではない）一般の利用者が日常的に利用しているソフトウェアであり，操作方法などを容易に習得することが可能である。

4.3.3　システム構成

本修正ツールを含めた人手修正環境のシステム構成を図 4.14 に示す。なお，この図は，4.1.1 節の図 4.1 を修正ツールの観点から詳細化したものである。

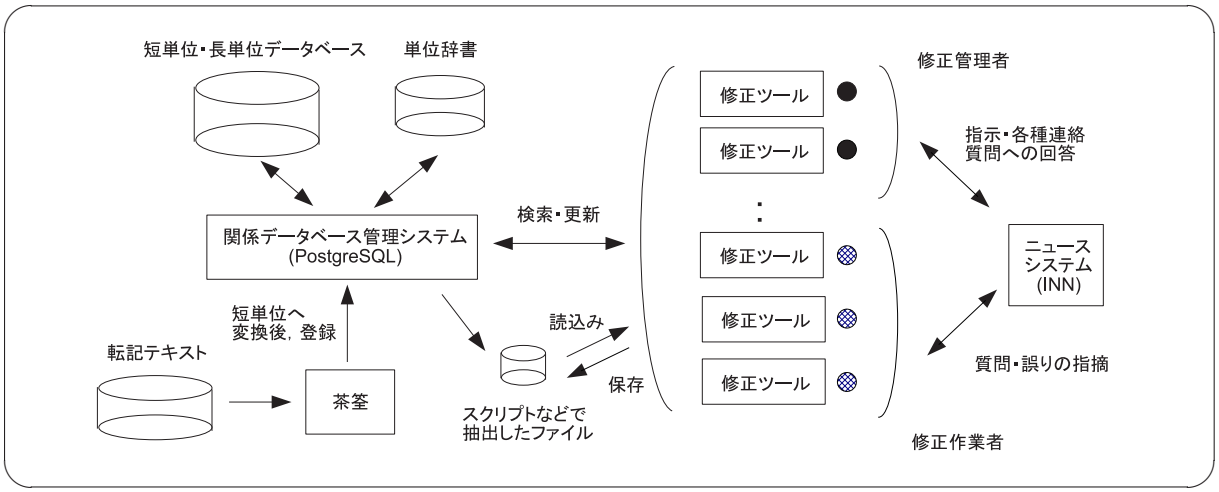


図 4.14　システム構成

図 4.14 のとおり，修正ツールは，関係データベース管理システムに対するクライアントとして，各修正者の PC 上で動作する。修正ツール自体は，表計算ソフトウェア Microsoft Excel のマクロとして実現している。

なお，プロジェクト期間中，修正者は，クロック 500〜700MHz 程度の PC を使用して修正作業を行っていた。OS は Windows98 である。

修正者は，GUI を用いて，単位データベースから単位データを検索する。検索された単位データは，修正ツールに読み込まれる。GUI を用いた検索の他に，スクリプトなどで単位データベースから抽出した単位データを通常の Excel ファイルと同様に読み込むこと（図 4.14 の中央下）も可能である。

読み込まれた単位データの編集は，基本的に表計算ソフトウェア上で普通のデータを編集するのと、ほぼ同じである。もちろん，単位データの修正を効率化したり，不注意による誤修正を防止する機能も用意している。詳細は，この後の節で説明する。

人手解析が終了した単位データは，修正ツールを介して，単位データベースに送信・更新される。データベースの更新は，修正者が明示的に行う（つまり，修正ツール上での編集結果がデータベースにすぐ反映されるわけではない）。

修正ツールは，ODBC（Open DataBase Connectivity）を介して，関係データベース管理システムにアクセスする。したがって，本プロジェクトでは関係データベース管理システムとして PostgreSQL を用いたが，Windows 用の ODBC ドライバを持つ関係データベース管理システムであれば，修正ツールに大きな変更を加えることなく利用できる。関係データベース管理システムには，単位データの他に，単位辞書，および，活用表が格納され，修正した単位データの整合性チェック，および，形態論情報の入力補助のために利用される。

単位データの修正者は，全員，言語学の素養があるものとし，同時修正者数は，最大 10 名程度とする。修正者のうち，4 名程度を修正管理者とする。修正管理者は，修正対象の指示や短単位・長単位の認定，および，単位辞書への登録など，コーパス全体の整合性を維持する役割を果たす。

図 4.14 右のニュースシステム（INN ver.2[*21]）は，4.1.2 節で述べたように，複数の修正者が存在することを考慮して，修正上の問題とその解決方法の共有を図るために用意した。具体的な利用方法としては，修正管理者から一般の修正者への連絡・指示が挙げられる。また，一般の修正者からの質問，単位データに対する誤りの指摘，それに対する修正管理者からの回答なども主要な利用例の一つである。なお，転記テキストに対する誤りもこのニュースシステムを介して，転記テキストグループへ伝達され，対応結果や対処方法についても転記テキストグループの担当者が返答する。このように，ニュースシステムは，単位データベース構築だけでなく，他のグループとのやりとりにも利用される。

4.3.4 修正ツールの概要

すでに述べたように，本修正ツールは，修正対象の単位データを表計算ソフトウェアに読み込み，その上で単位の区切り位置や付与されている形態論情報の修正を行う。単位データを読み込んだ状態を図 4.15 に示す。表計算ソフトウェア中の 1 行が単位データベースの 1 レコード，すなわち 1 短単位に相当する。単位データを読み込む際には，単位データベースの 1 レコードに含まれる 31 フィールドすべてを読み込む。各列の内容は，左から順に次のとおりである。詳細については，表 4.1 を参照していただきたい[*22]。全部で 31 フィールドと多くのフィールドがあるが，表計算ソフトウェアの機能を利用して，参照したいフィールドだけを表示するこ

[*21] http://www.isc.org/index.pl?/sw/inn/

[*22] なお，表 4.1 と異なり，管理情報のフィールドが二つの部分に分かれている。これは，管理情報のうち，最終更新者と最終更新時間は修正作業を行う上でよく参照される（例えば，修正作業をしているとき，修正管理者の編集した単位データが参考になる）ということもあるが，単位データベースを短単位，長単位の順で拡張していった経緯にもよる。

とができるので，編集上は大きな支障とはならない。

- 講演 ID，転記情報
- KWIC（前文脈，出現形，後文脈）
- 短単位関連情報（代表形，代表表記，発音形，品詞などの情報）
- 管理情報（最終更新者，最終更新時間）
- 長単位関連情報（代表形，代表表記，品詞などの情報）
- 管理情報（予備 1〜3）

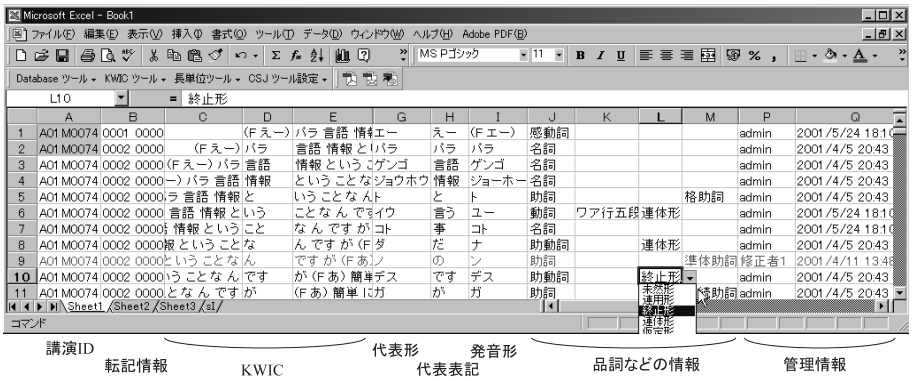


図 4.15　修正ツール

修正ツールを使った人手単位解析作業は，基本的に次の流れで行われる。この後の節では，それぞれの内容について，説明を行う。

- 単位データの検索と読み込み
- 単位データの修正
- 人手修正結果の単位データベースへの反映

4.3.5　単位データの検索と読み込み

単位データを検索し，修正ツールに読み込む方法としては，大きく分けて，次の四つの方法がある。詳細は，この後の節で順次説明する。読み込まれた単位データは，基本的に表計算ソフトウェアの一つのシートに読み込まれる。もちろん，複数のシートや別の表計算ソフトウェアに，複数の検索結果を読み込み，同時に編集することも可能である。

方法 1:　GUI を用いた方法
方法 2:　SQL を直接記述する方法
方法 3:　単位データファイルを読み込む方法
方法 4:　検索結果から再検索する方法

4.3.5.1 方法 1: GUI を用いた方法

まず，方法 1 は，図 4.16 に示した GUI を介して検索した結果を修正ツールに読み込む方法である。この GUI では，4.1.1 節で示したほとんどのフィールド（「ID」「後続 ID」など一般の修正者が通常利用しないフィールドを除く）に対して検索条件を指定できる。図 4.16 は，短単位の表記関連（図左），品詞関連（図右）のフィールドの指定画面である。

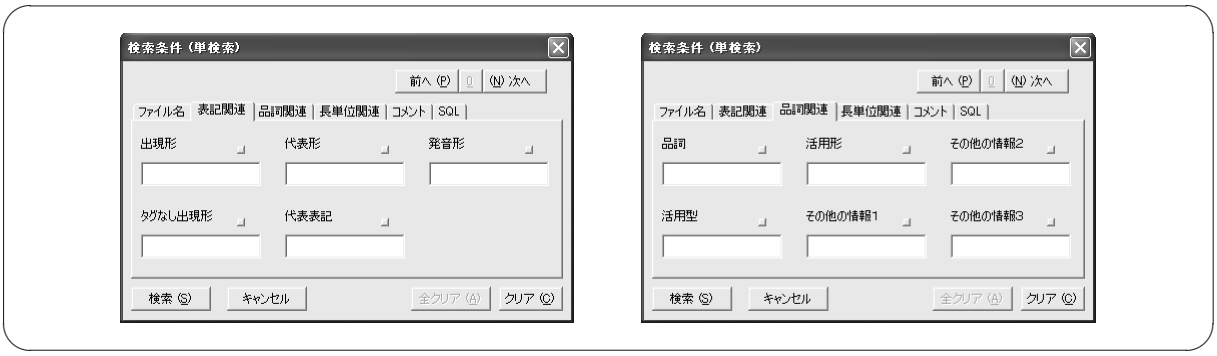


図 4.16 検索用 GUI

生成される検索式は，各フィールドに対する制約値の AND 条件となる。また，各フィールドの制約値を半角スペースで区切ることにより，当該のフィールド内で OR 条件を指定することができる。

各フィールドの制約値に対する一致条件は，完全一致の他に次の条件を指定することができる。なお，これらの一致条件を使用する場合は，下記脚注に示してあるとおり，効率的な検索を行うためには，索引機能のことを考慮して，検索式を指定する必要がある。

- **like 検索：** SQL-92 の like 検索で指定可能な文字列を指定[*23]
- **正規表現検索：** 正規表現を指定[*24]
- **任意条件指定検索：** 一致のための演算子を直接入力する。特に，否定条件を用いる時は，この方法を使用する。使用できる演算子や記述形式は，使用する関係データベース管理システムに依存する。

例 1: フィールド値が「国語」で始まらないもの

```
not like '国語%'
```

例 2: 指定したフィールドの値が pos フィールドと一致するもの

```
= pos
```

一致条件は，各フィールド欄の右上にある小さなボタンをクリックすると toggle するようになっている。一致条件は，次の四つの状態があるが，現在どの状態かは，各フィールド欄の色で区別とともに，フィールド欄

[*23] PostgreSQL では，フィールド値の先頭の値がリテラル値として特定されていないと，検索時に索引機能が働かず，検索速度が低下する。

[*24] PostgreSQL の拡張機能を利用している。like 検索と同様，フィールド値の先頭の値がリテラル値として特定されていないと，検索時に索引機能が働かない。

にカーソルを合わせると，現在の状態がポップアップで表示されるようになっている。

さらに，この GUI では，前後の隣接するレコードの情報を指定することも可能である。具体的には，図 4.16 の上部の「前へ」,「次へ」ボタンを押すと，指定レコードに対するフィールド欄ウィンドウが現れ，各フィールド値を指定できる。図 4.17 は，出現形に「外国」「語」を持つ連続する二つの短単位を指定した例である。検索結果はまとまった複数の短単位の組（この例の場合は,「外国」と「語」が組となる）になるので，先頭の短単位の前文脈と末尾の短単位の後文脈に下線を引いて，個々の組の範囲を明示している。

	A	B	C	D	E	F	G	H	I	J	K	L
1	A01F0090	0015 0004	に 対して	外国	語 話者 や	外国	ガイコク	外国	ガイコク	名詞		
2	A01F0090	0015 0004	して 外国	語	話者 や 特	語	ゴ	語	ゴ	名詞		
3	A01F0576	0109 0025	(F えと一)	外国	語 に 比べ	外国	ガイコク	外国	ガイコク	名詞		
4	A01F0576	0109 0025	と一) 外国	語	に 比べ ま	語	ゴ	語	ゴ	名詞		
5	A01F0861	0012 0002	対しまして	外国	語 話者 は	外国	ガイコク	外国	ガイコク	名詞		
6	A01F0861	0012 0002	して 外国	語	話者 は 長	語	ゴ	語	ゴ	名詞		
7	A01M0020	0027 0007	よ (F えー)	外国	語 教育 に	外国	ガイコク	外国	ガイコク	名詞		
8	A01M0020	0027 0007	えー) 外国	語	教育 に (F	語	ゴ	語	ゴ	名詞		
9	A01M0078	0004 0000	と (F えー)	外国	語 の 音声	外国	ガイコク	外国	ガイコク	名詞		
10	A01M0078	0004 0000	えー) 外国	語	の 音声 を	語	ゴ	語	ゴ	名詞		

図 4.17　前後のレコードを指定した検索 (「外国/語」の例)

4.3.5.2　方法 2: SQL 文を直接記述する方法

GUI から生成できない検索式については，図 4.18 のように，直接 SQL 文を記述して，問い合わせを実行することも可能である。例えば，検索結果の並び順の指定や SQL で定義されているさまざまな関数を利用することができる。図 4.18 の「現在のシートから」ボタンは，現在読み込まれているデータを検索した時に利用した SQL 文を SQL 文の指定欄に読み込むためのものである。なお，記述できる SQL 文は，使用している関係データベース管理システムに依存する。

図 4.18　検索用 GUI（SQL 文の記述）

SQL 文を直接指定して検索する際に注意すべきことは，検索結果は単純に修正ツールに読み込まれるだけだということである。つまり，検索結果の 1 レコードが表計算ソフトウェアの 1 行として，また，1 フィールドが 1 セルとして左から順に読み込まれるだけである。したがって，方法 1，2 と同様に検索結果を修正・更新する場合は，検索結果のレコードを修正ツールの仕様に合わせる必要がある。

4.3.5.3 方法 3: 単位データファイルを読み込む方法

これは，表計算ソフトウェアの形式で保存された単位データファイルを読み込む方法である。この機能は，本ツールにおいて，データベースから検索された単位データは，データベースとは完全に分離された状態にあり，本ツール上で修正を行っても，すぐにはデータベースに反映されない，という性質を利用している。

この方法は，大きく分けて三つの状況で利用する。

- 検索結果が大量になり，修正ツールに読み込めない場合[*25]
- SQL 文では目的の修正対象を検索することが困難で，スクリプトなどで抽出する必要がある場合
- 方法 1，2 で読み込んだデータをファイルに保存して，後で編集する場合

この方法は，複数の修正者を想定した修正作業において有用な機能である。次に実際の使用例を示す。

- 修正対象の単位データが膨大な場合（例：助詞の「の」のチェックを行う場合），単位データを複数のファイルに分割して複数の修正者に割り振れば，修正作業を分担して実施することができる。
- ファイルに作業途中の単位データを保存することができるため，修正者自身が進行状況を把握しやすくなる。また，修正管理者も進行状況を管理しやすくなる。例えば，修正管理者が複数の修正者に単位データのチェックを依頼する場合，作業中のファイルを見れば作業の進行状況をすぐ把握できる。方法 1，2 では，データベースの内容が常に変化しているため，進行状況を把握するのは難しい。

4.3.5.4 方法 4: 検索結果から再検索する方法

検索結果から再検索する機能として，次の三つの機能を実装した。

■転記基本単位の検索： 検索結果の単位データを分析しているとき，その前後の単位の情報を詳しく閲覧したい場合がよくある。そこで，指定した短単位（レコード）を含む転記基本単位，および，隣接する転記基本単位に含まれる短単位を検索する機能を実装した。

図 4.19 は，基本形が「国語」の短単位を検索した結果から，再度，周囲の転記基本単位を検索する例である。この例のように，前後の転記基本単位の範囲を指定する。検索結果は，新たに追加されるシートに読み込まれる。

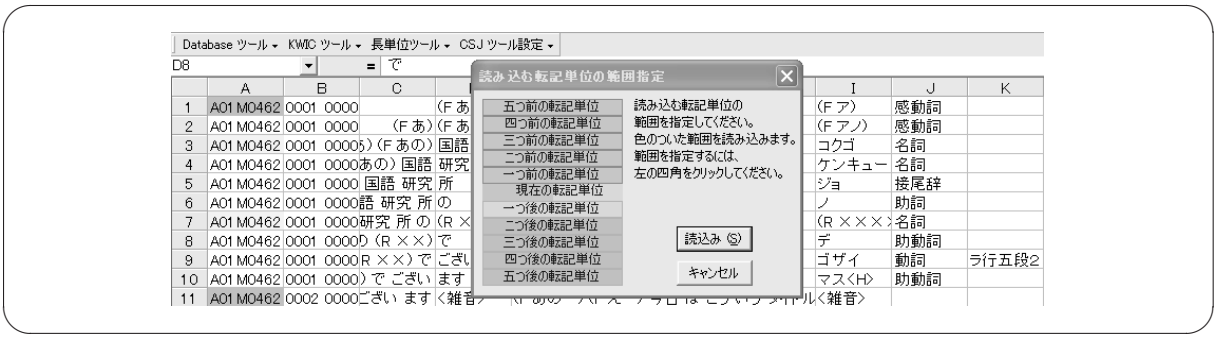


図 4.19 転記基本単位の検索

[*25] 修正ツールは Excel のマクロとして実現されているが，Excel のシートの最大読み込み数は 65536 行である。

■転記テキストの閲覧：　転記基本単位よりも広い範囲の文脈を参照したい場合は，転記テキスト全体を閲覧することが効果的である。そこで，指定された短単位（レコード）を含む転記テキストをテキストエディタで閲覧する機能を実装した[*26]。実行例を図 4.20 に示す。エディタは起動すると，自動的に当該の短単位を含む転記基本単位にカーソルを移動するようになっている。なお，エディタで閲覧するのは，単位データベースとは独立した転記テキストファイル（図 4.14 左下）である。

図 4.20　転記テキストの検索

■指定レコードの再読み込み：　4.3.7.2 節で述べるように，データ読み込み後に，別の修正者が変更したレコードを変更する場合は，当該のレコードを再度読み込み直して，レコードの内容を修正する必要がある。この手間を軽減するために，指定したレコードを再読み込みする機能を実装した。なお，再読み込みされた単位データは，文字色が緑色になり，通常の単位データと区別できるようになっている。

4.3.6　単位データの修正

修正ツールには，高精度，かつ，容易に単位データを修正するための機能として，大きくわけて，次の三つの機能がある。この後の節では，これらの機能について個々に説明する。

- 単位分割位置の修正機能
- 修正支援機能
- 単位辞書検索機能

4.3.6.1　単位分割位置の修正機能

まず，修正ツールの最も基本的な修正機能として，単位分割位置の修正について説明する。短単位の単位分割位置の修正機能には，分割，結合，移動の三つがある。

分割：　一つのレコードを二つに分割（例：外国語→外国/語）

結合：　二つのレコードを一つに結合（例：外/国→外国）

移動：　二つのレコード間での文字列の移動（例：外/国語→外国/語）

[*26] 本プロジェクトでは，閲覧用のテキストエディタとして，「秀丸」エディタを使用した。

■分割 一つのレコードを複数の単位に分割する修正である。例えば，本来「外国/語/大学」と単位認定されるべきところを「外国語大学」のように誤って 1 短単位に解析されている場合，三つに分割し直す。分割位置の指定は，図 4.21（左）のように，出現形フィールドに '/' を記入することにより指定する。

分割を実行すると，図 4.21（右）のように，三つのレコードに分割される。このとき，次の処理が自動的に実行される。

- 個々の出現形の長さに基づいて，転記情報フィールドの単位位置情報を修正する。
- 出現形の分割位置を考慮して，前文脈・後文脈フィールドの値を修正する。
- 分割する場合，単位データベース上に新たなレコードが追加されるので，新たな ID を発行し，それに合わせて，後続 ID を修正する。

ただし，発音形は自動的には分割されないので，分割後に別途修正する必要がある（修正には，この後述べるテキスト移動機能を使う）。また，品詞情報などの付与情報（表 4.1 の 9〜17 フィールドまで）については，分割前の付与情報が分割後の先頭の短単位に引き継がれ，先頭以外の短単位の付与情報は，空欄となる。付与情報の入力については，いくつかの入力支援機能がある。詳しくは，4.3.6.2 節で述べる。

なお，通常の編集操作（例：品詞の修正）では，明示的に更新処理を実行しないと，編集内容が単位データベースに反映されないが，分割の操作を実行すると，分割と同時に単位データベースの内容も変更される。これは，（分割により発生する）新規レコードの追加にともなう更新処理の複雑化を避けるためである。

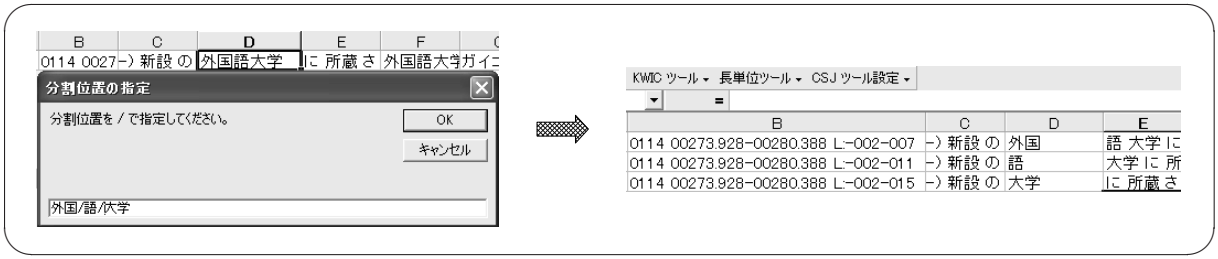


図 4.21 短単位の分割

■結合 二つのレコードを一つの短単位に結合する修正である。例えば，「外/国」のように誤って過分割されている場合は，二つの短単位を結合し「外国」に修正する。結合も分割と同様に，結合を実行した段階で，単位データベース上のデータが修正される。

結合する場合は，次の条件を満たす必要がある。満たさない条件があると，修正ツールはエラーを表示して，処理を中止する。

- 修正対象の 2 レコードが転記テキストの出現順に読み込まれていること（出現順にするには，後述するようにソート機能を使う）
- 二つのレコードが同一の文節に属していること。この条件が付加されているのは，この条件を満たさない二つのレコードの結合が転記テキストの修正を意味するからである。このような修正を行う場合は，4.3.8 節で述べるように，転記テキストグループに報告し，その結果を待って，修正を行う。

上記の条件を満たしていることが確認された段階で，次の処理を行う。

- 二つのレコードのタグなし出現形，代表形，代表表記，発音形フィールド値をそれぞれ合併する。
- 出現形の分割位置を考慮して，前文脈・後文脈フィールドの値を修正する。
- 結合前の後方のレコードを削除するとともに，結合結果のレコードの後続 ID を修正する。

■移動　「移動」は「外/国語」→「外国/語」のように，二つのレコード間での文字列の移動により修正を行う。この種類の短単位境界位置の修正は，「分割」「結合」を組み合わせれば実行できるが，修正の効率化のために個別に機能を実装した。修正の手順は，次のとおりである。

(1) 修正対象となる二つのレコードが転記テキストの出現順に並んで読み込まれていることを確認する。
(2) 「出現形，発音形の修正」機能（4.3.6.2 節参照）により，出現形の文字列を移動させる。誤修正を防ぐため，カーソルが出現形のセルにない場合は，文字列の移動は行われない。
(3) 更新を実行する。「移動」は，「分割」「結合」と異なり，単位データベース中のレコード数は変化しないので，通常の編集と同様，更新の実行は修正者が明示的に行う。

■結合（長単位）　次に，長単位の単位分割位置の修正機能について説明する。長単位の単位分割位置の修正機能は，「結合」のみ実装されている。長単位の結合例を図 4.22 に示す。この例は誤って分割されてる「外国語」と「話者」を結合している例である。結合する際には，後続長単位の代表形（長単位）と代表表記（長単位）を，先頭の長単位の代表形（長単位）と代表表記（長単位）にそれぞれ結合させる。また，品詞などの付与情報に関しては，結合前の長単位からコピーできるようになっている。コピー元の情報は，結合前の先頭，もしくは，末尾の長単位のいずれかから選択できる。

長単位の結合は，データベースのレコード数を変化させないので，修正ツール上の単位データに修正を加えるだけで，単位データベースへの変更結果の反映は修正者が明示的に行う。

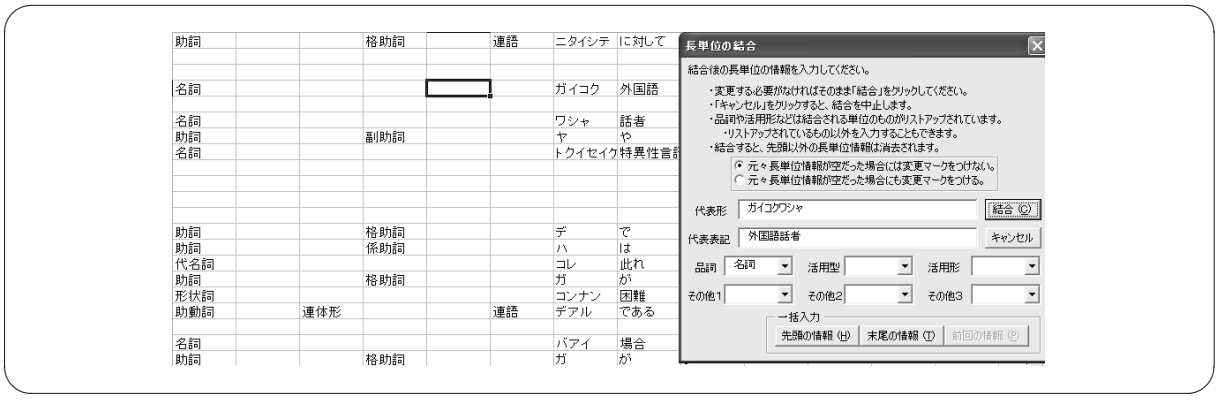


図 4.22　長単位の結合

4.3.6.2　修正支援機能

単位データの修正支援機能は，品詞などをはじめとする付与情報の修正を支援する機能である。修正支援機能には，修正ツール上に実装した機能と，表計算ソフトウェアにもともと付随している機能がある。ここでは，まず修正ツール上に実装した機能について見てみる。

■KWIC 機能 図 4.15 にも示したとおり，単位データベースには，前文脈，後文脈フィールドがあり，手軽に KWIC を利用することができる。この機能は，品詞，活用形などの人手修正には不可欠の機能である。また，単に KWIC を閲覧できるだけでなく，前文脈，後文脈でのソートが手軽にできるようになっている[*27]。修正ツールに実装されているソートの種類を次に示す。

- **短単位**
  - **出現順：** 転記テキストの出現順にソートする（転記 ID と転記情報をキーとしてソートする）。
  - **前文脈：** 前文脈をキーとしてソートする。ただし，ソートキーの優先順位は，短単位の代表形，代表表記，品詞，活用の種類，活用形（以後，これらのフィールドをまとめて，「辞書項目フィールド」と表記）[*28]，前文脈の順である。なお，前文脈をキーとして比較する際は，前文脈の末尾から先頭方向に向けて文字列の比較を行う。
  - **後文脈：** 後文脈をキーとしてソートする。ソートキーの優先順位は，短単位の辞書項目フィールド，後文脈の順である。
- **長単位**
  - **前文脈（構成要素ごと）：** 長単位の構成要素となる短単位（レコード）を組にして，前文脈をキーとしてソートする。ただし，長単位を構成する短単位は事前に修正ツールに読み込まれている必要がある。また，ソートキーの優先順位は，長単位の辞書項目フィールド，前文脈の順である。
  - **後文脈（構成要素ごと）：** 長単位の構成要素となる短単位（レコード）を組にして，後文脈をキーとしてソートする。ソートキーの優先順位は，長単位の辞書項目フィールド，後文脈の順である。
  - **前文脈（レコードごと）：** 長単位の構成要素となる短単位を考慮せずに，前文脈をキーとしてソートする。したがって，長単位を構成する短単位のうち，先頭以外の短単位のレコードについては，長単位の辞書項目フィールドが空欄であるものとして，ソートされる。ソートキーの優先順位は，長単位の辞書項目フィールド，前文脈の順である。
  - **後文脈（レコードごと）：** 長単位の構成要素となる短単位を考慮せずに，後文脈をキーとしてソートする。ソートキーの優先順位は，長単位の辞書項目フィールド，後文脈の順である。

■転記情報表示機能 単位データベースから読み込まれた単位データは，転記テキストの情報に基づいて色分けされて表示される。図 4.15 のように（白黒のため分かりづらいが），講演 ID（第 1 フィールド）は，当該短単位の転記テキスト上の位置により，次のような色分けがなされる。

- **灰色：**転記基本単位の先頭の短単位の場合
- **黄色：**（転記テキストの）文節の先頭の短単位の場合

これにより，文の先頭や文節の先頭であることが分かりやすくなり，形態論情報の付与に役立つ。例えば，接続詞の「デ」は転記基本単位の先頭に現れることが多く，助動詞「ダ」の連用形や格助詞の「デ」と区別するのに役立つ。また，仕様上，異なる転記基本単位，文節間では短単位は連結できないことになっているが，

[*27] 表計算ソフトウェアのソート機能を使わずに修正ツール側でソート機能を実現しているのは，表計算ソフトウェアが扱えるソートキーの数が 3 と少ないためである。例えば，代表形，代表表記，品詞，出現形というように，三つ以上のソートキーを使用することはよくある。

[*28] これにより，同一種類の短単位をまとまった形でソートすることが可能である。

上記のような色分けがなされていることにより，その仕様に反した転記テキストを発見しやすくなる。

■付与情報入力支援機能　単位の付与情報の入力支援機能には，次の機能を実装している。

入力値選択メニュー　形態論情報のうち，入力値の候補が決まっているフィールドは，メニュー選択で値を入力できるようになっている。これにより，不正な入力値を防ぐことができる。メニュー選択できるのは，品詞，活用形，その他の情報 1，3 である。

短単位情報のコピー　長単位に対する形態論情報付与に際しては，構成要素の短単位の形態論情報を利用できる場合も多い。顕著なのが，単一の短単位からなる長単位であり，構成要素の短単位と同一の形態論情報を持つ。このような長単位は，CSJ 全体の約 82% を占める。そこで，修正対象のレコード内の短単位の形態論情報を長単位のフィールドにコピーする機能を実装した。

長単位の情報の表示/非表示　表 4.1 に示したとおり，単位データベースのレコードは 31 フィールドもある。そのため，修正ツールに読み込んだ場合，管理情報関連や予備フィールドなど，表計算ソフトウェアの表の右方に表示されるフィールドは，閲覧しづらい。そこで，長単位の付与情報の修正を行わない場合には，長単位に関連するフィールドを非表示にする機能を実装した。

出現形，発音形の修正　出現形の編集（「分割」や「結合」など修正ツールの機能を使った編集を除く）は，短単位の「移動」（4.3.6.1 節参照）の際に行われる。また，発音形の編集は，単位分割位置の修正や DP マッチング（4.2.3.5 節参照）による誤りを修正する場合などに行われる。これらの修正は，テキストの脱落や不要な文字の追加など，転記テキストとの不整合を発生させる恐れがある。そこで，このような不整合を防ぐため，次の機能を実装した[*29]。

- **機能 1**：修正対象の短単位の出現形（または，発音形）の先頭の文字を，前接する短単位の出現形（または，発音形）の末尾に移動
- **機能 2**：修正対象の短単位の出現形（または，発音形）の末尾の文字を，後接する短単位の出現形（または，発音形）の先頭に移動

実行例を図 4.23 に示す。図中の I 欄が発音形である。左図の 1 行目にまとまっている「ガイコクゴダイガク」を 2，3 行目に分割していく（中央，右図）。この操作は，機能 2 を連続的に実行している。

図 4.23　発音形の修正

なお，この機能は，意図しない修正を防ぐために，次の条件をすべて満たす場合しか実行されない。

- 修正対象の短単位データの出現形，もしくは，発音形のセルにカーソルが存在すること
- 前接，もしくは，後接する短単位と修正対象の短単位が転記テキストの出現順に並んでいること

[*29] これらの機能は，ファンクションキーに割り当てている。

次に，表計算ソフトウェア自身の編集機能のうち，単位データの修正を効率的に行うのに有用な機能を次に示す。

- **フィルタ機能：** 各フィールドの値を制約するフィルタ機能は，検索結果の単位データの中から特定の単位データだけを抽出するのに有用である。例えば，名詞の単位データだけを表示するなど，さまざまな利用方法がある。
- **ソート機能：** 修正ツールには，短単位，長単位合わせて 4 種類のソートが用意されているが，よく使われるソートキーに限定して実装している。用意されていないフィールドをキーとしてソートする場合は，表計算ソフトウェア自身のソート機能を利用することができる。
- **文字列のコピー＆ペースト機能：** 語別に形態論情報をチェックしている場合など，同一の修正を行うことがよくある。このような場合，表計算ソフトウェアのコピー＆ペースト機能が有効である。また，修正対象の単位が修正ツール上で連続して表示されている場合，修正した単位の情報をドラッグして，複数の単位に連続的にコピーするなど，人手修正を効率化するのに役立つ。
- **セルの色やフォント：** セルの色やフォントの種類を，修正作業時の補助情報として，利用することができる。これらの情報は，単位データベースに反映することはできないが，修正作業の見直し時などに有効である。特に，4.3.5 節で述べたように，本修正ツールは，単位データベースから読み込んだ単位データを，表計算ソフトウェアのファイル形式で保存し，後日そのファイルを編集することができるので，後日の編集のためのメモとしての機能を果たすことができる。

4.3.6.3 単位辞書検索機能

単位辞書は，代表形，代表表記，活用型（活用語のみ）[*30]，品詞，その他の情報 1 を保持した短単位と長単位の辞書である。単位辞書の構成の詳細については，4.4 節で述べることとし，ここでは，修正ツールと連係した利用の方法について述べることにする。

修正ツールにおいて，単位辞書は，修正したデータのチェックと単位データの入力支援を行うために利用される。まず，修正したデータのチェックは，修正結果の単位データと単位辞書に格納されている辞書項目とを比較し，両者に矛盾がないかチェックする。この際，修正対象の単位データの活用の種類，活用形，および，活用表を参照し，単位辞書の基本形を活用させた上で比較する。このチェックは，修正した単位データの更新時に実行される。矛盾が発見された場合は，警告を発するとともに，矛盾が存在する単位データの色を変化させる。この際，警告は発するが，修正結果は単位データベースに反映される。なお，このチェックは，更新時だけでなく，チェック対象の単位データを指定（複数指定可）して明示的に実行することもできる。

もう一つの利用方法は，単位辞書を使った単位データの入力支援である。図 4.24 に示す GUI を介して，単位辞書にアクセスする。実装されている機能は，次のとおりである。

- 代表形，または，代表表記を指定して，単位辞書を検索すること
  - 単位データの検索と同様，完全一致検索，like 検索や正規表現などが可能である。
  - 検索結果のうち，選択した辞書項目の情報を編集中の短単位データに反映させることができる（図 4.24 の「挿入」ボタン）。

[*30] 単位辞書の仕様では，「活用の種類」を「活用型」と呼んでいる。詳しくは，4.4 を参照。

– なお，長単位の検索結果では，図 4.24（右）のように，構成要素の短単位の情報を表示することができる。この図では，長単位「ガイコクゴワシャ」の短単位構成情報を表示している。

- 単位データベースから選択した辞書項目の用例を検索する（図 4.24 の「用例検索」ボタン）。検索結果の用例は，新たに追加されるシートに読み込まれる。

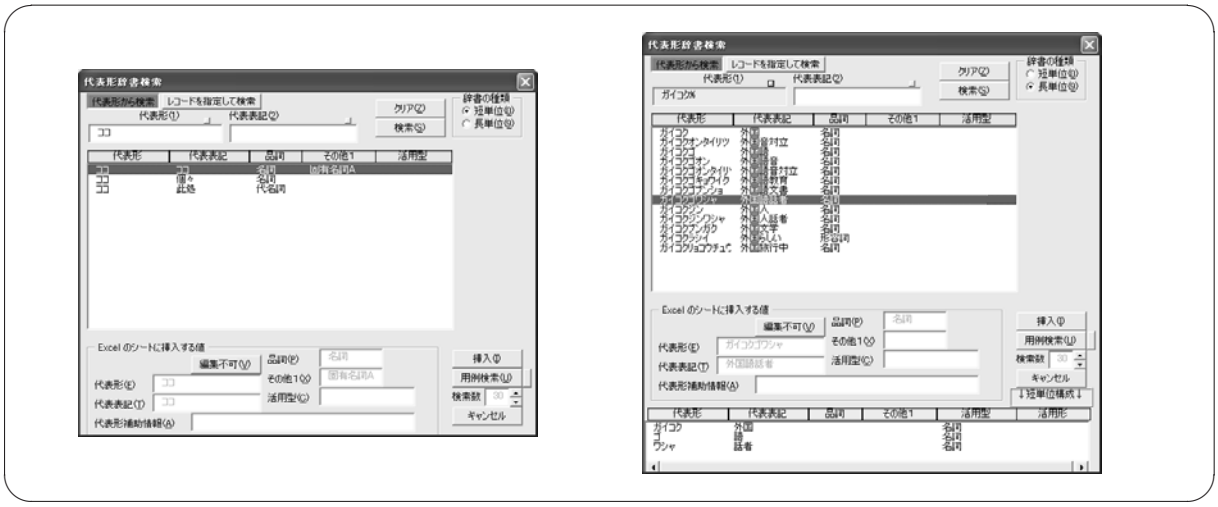


図 4.24　単位辞書検索用 GUI

4.3.7　単位データの更新と排他制御

4.3.7.1　更新処理

すでに述べたように，修正ツールに読み込まれた単位データに修正を加えても，その修正は，すぐには単位データベースに反映されない。修正内容の反映は，短単位の「分割」「結合」処理以外は，随時，修正者が明示的に実行する。更新処理は，次の手順で行われる。

(1) 修正された単位データの文字色を赤色に変更し（修正ツールに読み込まれたデータは，修正者が何らかの修正を加えると自動的に変更されるようになっている），更新対象行とする。

(2) 修正者により更新が指示され，更新処理が始まると，単位データベース上の当該レコードがロックされる。

(3) 単位辞書を用いて，当該単位データの整合性をチェックする（整合性チェックの処理内容は，4.3.6.3 節を参照）。

(4) 排他制御（次節を参照）を行いつつ，修正内容に基いて単位データベースを更新（update）する。更新に成功した単位データは，文字色を青色に変更する。

(5) 更新対象行が複数存在する場合は，2〜4 の処理を繰り返し実行する。

(6) 2〜4 の処理の過程でエラーが発生した場合は，エラーが発生したことを修正者に警告するとともに，当該単位データの文字色を次のように変更する。

- **紫色：**当該レコードが単位データベース上ですでに修正されている場合（単位データの「結合」などにより，当該レコードが存在しない場合も含む）
- **ピンク色：**当該単位データが整合性に矛盾が発見された場合

4.3.7.2 排他制御

4.3.1 節で述べたように，本修正ツールの特徴は，複数の修正者が同時に単位データベースを修正できることである。本修正ツールは，この機能を実現するために，排他制御を用いる。本修正ツールは「コピー/変更/更新」型の排他制御方式を採用した。この方式は，CVS[*31]などのバージョン管理システムなどで利用されている。単位データの読み込み（コピー）から更新までの一連の排他制御の手順は，次のとおりである。

(1) 修正ツールへ単位データの読み込みを行う。この際，読み込み時だけロックをかける（読み込み終了時にロックを解除する）。修正ツールに読み込まれる単位データは，単位データベース上のデータのコピーとなる。このため，同じ単位データを複数の修正ツールで同時に読み込み，編集することが可能になる。4.3.5.3 節「方法 3：単位データファイルを読み込む方法」のような読み込み方法を実現できるのは，この機能によるものである。

(2) 読み込んだ単位データを修正する。

(3) 修正が終了したら，更新処理を行う。更新する際は，（修正ツールに読み込まれている）更新対象の単位データの最終更新時刻と単位データベース側の最終更新時刻とが一致するか，検査する。

- **検査に適合した場合:** 更新対象のレコードの最終更新時間，および，最終更新者フィールドを現在の修正者[*32]，現在時刻に変更し，当該のレコードにロックをかけつつ，更新する。
- **適合しなかった場合:** 更新を中止する。なお，この場合，修正者は「レコード再読み込み」機能により当該レコードを再読み込みし，他の修正者がどのように変更したのかを確認する必要がある。

4.3.8 ニュースシステム

すでに述べたように，単位データベースの構築には，複数の修正者が修正を行う。また，単位データベースが情報的に包含する転記テキスト自体は，転記テキストグループにより構築される。そのため，修正者間，転記テキストグループとの間でさまざまなやりとりをする必要がある。実際に行われていたやりとりの例として，次の三つを挙げる。

- 修正管理者が単位データの仕様変更や修正ツールのバージョンアップを一般の修正者に連絡する。
- 同一の短単位データにもかかわらず，代表表記が異なる場合，修正作業者が修正管理者にどちらが正しいか確認を行う。
- 転記テキストに誤りを発見した場合，転記テキストグループの担当者に，誤りの報告と問い合わせを行う。

以上のようなやりとりは，構築する単位データベースの整合性や精度を向上させる上で非常に重要である。そこで，インターネットニュースシステムを導入した[*33]。このシステムのニュースグループとして扱うのは，主として，単位データ，および，転記テキストに関する事柄である。ニュースグループの一覧を投稿数とともに，表 4.5 に示す。

[*31] https://www.cvshome.org/

[*32] Microsoft Excel の「作成者」プロパティの値が使用される。修正者は事前に一意に定まる名前を記入しておくことになっている。

[*33] 国語研究所内のローカルネットワーク内でのみの運用である。

表 4.5　ニュースグループと投稿数

グループ名	概要	投稿数
announce	関係者への連絡用	129
tantan	短単位一般に関する事柄	966
chotan	長単位一般に関する事柄	280
tenki	転記テキスト一般に関する事柄	966
chotenki	転記テキストに関連する事柄のうち，長単位に関連するもの	651
daihyoukei	代表形に関する事柄	108
non-core	コア以外のデータに関連する事柄	237
system	単位データベース，修正ツールに関する事柄（リリースやバグ情報など）	132
misc	どのニュースグループにも属さない事柄	0
test	テスト用	83

形態論グループのメンバーは，その日の作業開始時にニュースグループを閲覧するように決められていた。したがって，連絡事項や他の修正者が直面した問題を全員で共有できる。また，tenki に投稿されたメッセージは転記テキストグループの担当者が適宜参照し，転記テキストの修正を行った。ただし，表 4.5 のとおり，転記テキストの修正量が多いので，必要がない限り，転記テキストグループの担当者からの返答はない。

使用したニュースシステムは，INN ver.2 である。ニュース閲覧用のクライアント[*34]から閲覧，投稿する。実際に投稿されたメッセージを，次に 2 例示す。

例 1 は，ニュースグループ tantan に投稿されたメッセージである。このメッセージでは，修正者が活用形の誤りを指摘し，修正管理者がそれに答えている。例 2 は，ニュースグループ tenki に投稿されたメッセージで，転記テキストの表記の誤りを指摘した例である。

- 例 1（tantan に投稿されたメッセージの例）

■修正者の質問

講演 ID: JL99OCT002
発話 ID: 0297
コメント: **「抽出した」の「た」の活用「終止形」となっていますが「連体形」では？**
0297 00798.143-00802.369 L:

（F え）菊沢が	**& （F エ）キクザワガ**
抽出した	**& チューシュツシタ**
女房詞の	**& ニョーボコトバノ**
四つの	**& ヨッツノ**
特徴を	**& トクチョーオ**
挙げます	**& アゲマス**

■修正管理者の回答

ご指摘のとおり「連体形」が正解です。

[*34] Microsoft Outlook や Netscape Communicator がよく用いられていた。

- 例 2（tenki に投稿されたメッセージの例）

> **講演 ID：ISO1NOV009**
> **発話 ID: 0512，基本形**
> **訂正前：（食べ）もの**
> **訂正後：（食べ）物**

4.3.9 修正ツールの運用

ここでは，修正ツールの実際の運用内容について述べる。

■転記テキスト単位でのチェック 4.2 節で述べたように，既存の形態素解析システムを使用するなどして，単位データの初期値を半自動的に作成する。したがって，作成された単位データを単位データベースに登録した直後は，すでに人手で修正されている単位データと精度が異なる。そこで，まず，新たに登録した単位データを転記テキスト単位でチェック・修正する。基本的に，一つの転記テキストに対して，一人の修正者が担当する。担当者は，修正対象の転記テキストの転記 ID を検索キーとして修正ツールで当該の転記テキストに含まれるすべての単位データを読み込み，修正作業を行う。

■語単位でのチェック 転記テキスト単位でチェックを行った単位データがある程度集積した段階[*35]で，語ごとのチェックを行った。これにより，主として，付与されている形態論情報（例えば，代表形や代表表記）にばらつきがないかをチェックすることができる。チェックの手順は次のとおりである。

- 単位データベース中に含まれるすべての単位データを抽出し，代表形，代表表記，品詞，「その他の情報 1」，「活用の種類」フィールドをキーとしてソートする。抽出される単位データは膨大なので，単位データベース管理システム上で抽出・ソートを行う。
- 1〜2 万レコードずつに分割し，Excel 形式のファイルに変換する。
- 各ファイルを修正者に割り当て，修正作業を行う。基本的に一つのファイルを一人の修正者が担当する。

■特定の言語表現に対するチェック 特定の言語表現に関して，系統的な誤りが発見された場合など，言語表現を限定して検索を行い，単位データをチェックする。例えば，終止・連体形の区別，助詞「で」と助動詞「だ」の区別などが挙げられる。なお，検索数が少ない場合は，修正ツールから直接検索を行うが，終止・連体形の区別のように検索数が膨大になる場合は，単位データベース管理システム上で検索を行い，検索結果を複数の Excel ファイルに分割した上で修正者に分配する。

■修正内容のチェック ある程度の解析精度に達した場合，例えば，プロジェクト終盤では，不用意な修正が解析精度の低下につながる可能性がある。そこで，次のように修正内容をチェックすることにより，精度の低下を防ぐ。

- 一般の修正者は誤りを発見しても修正はせず，当該単位の予備 3 フィールドに修正内容を記入する。次に，修正管理者が予備 3 フィールドが記入されている単位データを検索し，誤りであることを確認した

[*35] 人手短解析単位データは約 100 万短単位だが，おおむね 2 回に分けて実施した。

後に修正する。

- 最終更新時間フィールドの情報を用いて，一定期間（例えば，その日 1 日）に修正されたレコードを検索して，修正管理者が修正内容をチェックする。

■解析精度の測定　プロジェクトの過程では，構築した単位データの解析精度を測定していた。解析精度の測定は，単位データベースからスクリプトにより 10000 レコードランダムサンプリングし，Excel 形式に変換した後に，人手で再度チェックするというものである。短単位のサンプリングチェック結果を図 4.25 に示す。

この方法は，ファイルとして渡された単位データだけを確実にチェックすることができるので，次の利点がある。

- ある時点の精度を測定することができる。つまり，単位データをチェックしている間も他の修正者による修正が行われているので，単位データは常に変動していることになるが，その影響を除くことができる。
- 再度確認することが可能になり，チェックの精度を向上させることができる。

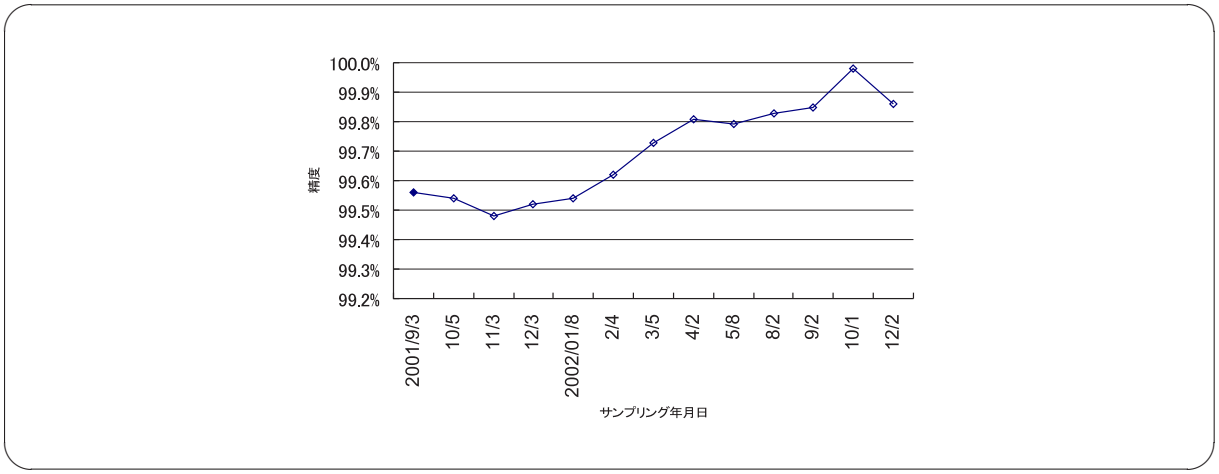


図 4.25　サンプリングチェックの結果

4.3.10　修正ツールによる単位データの構築結果

本修正ツールを CSJ に対する形態論情報の人手修正に適用した結果は，次のとおりである。まず，人手解析短単位データについては，延べ 1015589 短単位（異なり 19451）に対してチェック・修正を行い，解析精度は 99.9%（ランダムサンプリング，サンプル数 20000）となった。一方，情報通信研究機構の自動短単位解析システムが出力した短単位データに対する人手修正では，約 650 万短単位の中から，新たに 31456 の未知の短単位を抽出し，単位辞書への登録を行った。

4.4 短単位辞書の構築

4.4.1 概要

本節では，短単位辞書の構築方法について説明する。短単位辞書は，『日本語話し言葉コーパス』から，短単位をその用例とともに収集したものである。短単位辞書を作成した目的は，次のとおりである。

- 人手解析時に修正した短単位の整合性チェックを行うこと
- 自動短単位解析システム用の辞書として利用すること
- 自動短単位解析対象の転記テキストに含まれる短単位を実際の用例とともに列挙すること

短単位辞書に収録される短単位は，大きく分けて，次の二種類の転記テキストから収集した。

(1) 人手短単位解析対象の転記テキスト（人手解析単位データ）
(2) 自動短単位解析対象の転記テキスト（自動解析単位データ）

(1) から収集した辞書項目は，人手短単位解析結果を元に作成したものであり，(1) に含まれる短単位はすべて収録している。また，自動短単位解析システムで利用できるよう，活用形，活用型（活用の種類[*36]）の情報を詳細化している（4.4.4 節参照）。

一方，(2) から収集した辞書項目については，自動短単位解析結果を人手修正する過程で収集した短単位である。したがって，自動短単位解析対象の転記テキストに含まれる短単位を時間の許す範囲で収集したものであり，すべての短単位を網羅しているわけではない。

また，短単位辞書の活用型，活用形は，短単位の仕様とは一部異なる。これは，上で述べたように，自動短単位解析システム用の辞書としての利用も考慮しているため，より詳細な活用型，活用形が必要となったことによるものである。

この後の節の構成は，次のようになっている。まず，4.4.2 節で短単位辞書の設計ついて述べる。次に，4.4.3 節で短単位辞書の構造について述べる。4.4.4 節では，短単位辞書の活用型，活用形を規定する活用表について説明する。4.4.5 節では，人手・自動単位解析における短単位辞書の構築方法について説明する。

なお，単位データ構築環境（図 4.1）における単位辞書には，短単位辞書だけでなく，長単位辞書も含まれている。本来，長単位辞書についても解説すべきだが，4.1.4.1 節で述べたように，長単位辞書は，スクリプトによる人手解析長単位データの整合性チェックと，修正ツールから参考用の長単位辞書として利用されただけで，一般には公開されなかった。したがって，本節では短単位辞書について説明するにとどめる。

4.4.2 設計

短単位辞書は，上述した三つの目的のもとに作成された。これらの目的に基づいて，次の特徴を持つ短単位辞書を設計した。

[*36] 本節（4.4 節）では，「活用型」という用語を「活用の情報」のかわりに用いる。これは，短単位辞書を自然言語処理システム用の辞書として利用することを開発目的の一つとしており，自然言語処理の分野では一般的に「活用型」という用語が用いられるからである。

転記テキストに則した基本形： 各辞書項目には，CSJ の転記テキストの表記規則に則した基本形が格納される。また，異表記の存在を考慮して，一つの辞書項目が複数の基本形を保持する。例えば，代名詞「オマエ」（代表形）の基本形には「おまえ」と「おめえ」が格納される。これにより，異表記に対応した自動単位解析システムを構築できるものと思われる。

詳細な活用型と活用形： 短単位の人手解析データの品詞情報と比較して，活用型，活用形を詳細化させた。これは，自動短単位解析システムでは，実際の活用語形に一対一対応した活用型，活用形が必要となるからである。例えば，五段活用の動詞の未然形には，二つの語形が存在する（例：「書く」の場合，「書か」「書こ」の二つがある）が，短単位辞書では，それぞれの語形に対して，活用形を設けた。

特殊な語形への対応： CSJ は話し言葉を収録しているため，活用表に適合しない崩れた語形も現われる。例えば，次に示す例のように，未然形，終止形，連体形語尾が撥音になる場合である。

> **例：** 分からない → 分かんない，集めるのは → 集めんのは，歩いて → 歩って

このような活用表で対応できない，特殊な語形を持つ短単位については，その語形を基本形とする，独立した辞書項目として収録する（以後，このような辞書見出しを「個別辞書項目」と表記する）。なぜなら，活用語を自動解析する場合，活用表を用いて，各語形を認識することが多く，活用表で対応できない語形については，正しい認識ができないからである。また，人手修正作業における単位の整合性チェックの際にも同様の問題が起こる。

用例の付与： それぞれの辞書項目には，CSJ に実際に出現した用例を含める。用例を付与した目的は，個々の辞書項目に対応する短単位が CSJ 中のいなかる文脈で使われているかを明示するためである。これは，自動短単位解析データだけに含まれる短単位の用例を参照したいときなどに，特に有用であると考える。なぜならば，自動解析短単位データは，人手解析短単位データよりも解析精度が低く，必ずしも正しい用例を参照できるとは限らないからである。

以上のことを考慮して，短単位辞書の辞書項目は，次の情報を格納することにする。なお，このうち，＊の情報については，個別辞書項目だけに付与される。

- 基本形，代表形，代表表記
- 品詞，活用型，活用形 *，その他の情報 1，その他の情報 2 *
- 当該辞書項目に対応する用例

4.4.3 短辞書辞書の構造

4.4.3.1 辞書項目の例

短単位辞書は，XML で記述されている。ここでは，短単位辞書の構造を記述する前に，辞書に登録されている辞書項目の例を示しておく。

例 1 は，名詞「アイシャ」の例である。このように，一つの辞書項目は，「辞書項目」要素で記述され，「基本形」「代表形」「代表表記」「品詞」「その他の情報 1 」「用例集」要素を持つ。このうち，「基本形」要素については，複数の基本形があることを想定し，複数の li 要素を包含できるようになっている。また，活用語については，以上の要素の他に，「活用型」要素が加わる。動詞「アエル」の例を例 2 に示す。

例 1：名詞「アイシャ」

```
<辞書項目>
  <基本形><li>愛車</li></基本形>
  <代表形>アイシャ</代表形>
  <代表表記>愛車</代表表記>
  <品詞>名詞</品詞>
  <その他の情報 1></その他の情報 1>
  <用例集>
    <用例>
      <講演 ID>S02M1715</講演 ID>
      <転記情報>0301 00692.003-00695.787 L:-009-001</転記情報>
      <前文脈>いう こと は （F あ） もう 特に 考え て 今後 は もう 本当 に 自分 の</前文脈>
      <転記基本形>愛車</転記基本形>
      <後文脈>を 傷付け ない よう に 破壊 し ない よう に ええ 気 を 付け たい</後文脈>
      <転記発音形>アイシャ</転記発音形>
      <その他の情報 2></その他の情報 2>
    </用例>
  </用例集>
</辞書項目>
```

例 2：動詞「アエル」

```
<辞書項目>
  <基本形><li>会える</li></基本形>
  <代表形>アエル</代表形>
  <代表表記>会える</代表表記>
  <品詞>動詞</品詞>
  <活用型>ア行下一段 2</活用型>
  <その他の情報 1></その他の情報 1>
  <用例集>
    <用例>
      <講演 ID>S00F0173</講演 ID>
      <転記情報>0025 00066.183-00067.400 L:-003-001</転記情報>
      <前文脈>貨幣 も 拒み 昔 ながら の 伝統 を 守り ながら 暮らし て いる 人々 に</前文脈>
      <転記基本形>会える</転記基本形>
      <後文脈>（F え） そんな 話 を 本 で 読ん だ 私 は たまら なく この 土地 へ</後文脈>
      <転記発音形>アエル</転記発音形>
      <活用形>終止形</活用形>
      <その他の情報 2></その他の情報 2>
    </用例>
  </用例集>
</辞書項目>
```

「用例集」要素には，CSJ に含まれる実際の用例を一つ以上含む。自動解析単位データの場合，誤りを含む可能性があるが，収録されている用例は，正しく解析されているかどうかを人手で確認してある。また，実際の出現位置を特定できるように，講演 ID と発話 ID を保持している。

次に示す例 3 は，動詞「いりゃ」の辞書項目である。これは，個別辞書項目の例として示した。「いりゃ」は，ア行上一段動詞「イル」＋接続助詞の「バ」の融合した語形である。この例ように，活用型と活用形だけでは活用表から出現語形を導きだせない場合，独立した辞書項目を登録している。個別辞書項目では，この例のように，一般の辞書項目には含まれない「活用形」「その他の情報 2」要素を「辞書項目」要素中に含んでいる。

例 3：動詞「いりゃ」

```
<辞書項目>
  <基本形><li>いりゃ</li></基本形>
  <代表形>イル</代表形>
  <代表表記>居る</代表表記>
  <品詞>動詞</品詞>
  <活用型>ア行上一段１</活用型>
  <活用形>仮定形</活用形>
  <その他の情報 1></その他の情報 1>
  <その他の情報 2>融合</その他の情報 2>
  <用例集>
    <用例>
      <講演 ID>S05M0016</講演 ID>
      <転記情報>0162 00354.293-00358.181 L:-002-001</転記情報>
      <前文脈>今 で も （D つ）付き合っ て ます です から 友達 って の は 何 人</前文脈>
      <転記基本形>いりゃ</転記基本形>
      <後文脈>いい って もん じゃ なく 本当 に 気持ち が 分かる 人間 一人 で も いれ</後文脈>
      <転記発音形>イリャ</転記発音形>
      <活用形>仮定形</活用形>
      <その他の情報 2>融合</その他の情報 2>
    </用例>
  </用例集>
</辞書項目>
```

4.4.3.2　DTD

短単位辞書は，XML で記述され，その形式は DTD で規定されている。ここでは，DTD に記述されている要素と対応させて，短単位辞書の構造を説明していくことにする。

■「短単位辞書」要素

- 「短単位辞書」要素は短単位辞書自体を表す要素であり，一つ以上の「辞書項目」要素を含む。
- 属性 version は，短単位辞書の版を表す。
- 属性 release_date は，短単位辞書のリリース年月日を表す。
- 属性 name は，短単位辞書の名前を表す。

```
<!ELEMENT      短単位辞書      (辞書項目 +)>
<!ATTLIST      短単位辞書
               version          CDATA   #REQUIRED
               release_date     CDATA   #REQUIRED
               name             CDATA   #REQUIRED
>
```

■「辞書項目」要素

- 「辞書項目」要素は，短単位辞書の個々の辞書項目を表す。
- 「基本形」「代表形」「代表表記」「品詞」「活用型」「活用形」「その他の情報 1」「その他の情報 2」「用例集」要素を含む。
- 「活用型」は，活用語だけに付与される。
- 「活用形」要素,「その他の情報 2」要素は，個別辞書項目のみに付与される。

```
<!ELEMENT      辞書項目        (基本形，代表形，代表表記，
                                 品詞，活用型?，活用形?，
                                 その他の情報 1，その他の情報 2?，
                                 用例集)>
```

■「基本形」要素

- 「基本形」要素は，短単位辞書の見出しに相当する。表記は，転記テキストに準ずるが，転記テキストに付与されている各種のタグは，取り除いてある。
- 活用語の場合は，終止形で表記する。
- 複数の表記が存在する場合を考慮し，li 要素で列挙する。

```
<!ELEMENT      基本形          (li+)>
<!ELEMENT      li              (#PCDATA)>
```

■「代表形」要素

- 形態論情報の「代表形」を格納する。短単位辞書の見出しに相当する。代表形についての詳細は，3.3 節を参照のこと。

```
<!ELEMENT      代表形          (#PCDATA)>
```

■「代表表記」要素

- 形態論情報の「代表表記」を格納する。代表表記についての詳細は，3.3 節を参照のこと。

- ただし，自動解析単位データにしか現れない短単位の代表表記については，人手解析単位データの代表表記と異なり，厳密な表記基準を定めておらず，同一表記の代表形を持つ辞書項目を分別できるよう定めているだけである。

```
<!ELEMENT      代表表記          (#PCDATA)>
```

■「品詞」要素

- 形態論情報の「品詞」を格納する。品詞についての詳細は，3.3 節を参照のこと。

```
<!ELEMENT      品詞              (#PCDATA)>
```

■「活用型」要素

- 形態論情報の「活用型」を格納する。なお，すでに述べたように，「活用型」は 3.3 節の「活用の種類」に相当するが，自然言語処理システムでの利用を想定して，一部詳細化してある。詳しくは，4.4.4 節の「活用表」を参照のこと。
- この要素は，「辞書項目」要素と「用例」要素に現れるが，「辞書項目」要素では，個別辞書項目の場合のみ記述される。

```
<!ELEMENT      活用型            (#PCDATA)>
```

「活用形」要素

- 形態論情報の「活用形」を格納する。自然言語処理システムでの利用を想定して，3.3 節の「活用形」よりも詳細化してある。詳しくは，4.4.4 節の「活用表」を参照のこと。
- この要素は，「辞書項目」要素と「用例」要素に現れるが，「辞書項目」要素では，個別辞書項目の場合のみ記述される。

```
<!ELEMENT      活用形            (#PCDATA)>
```

■「その他の情報 1」要素

- 形態論情報の「その他の情報 1」，すなわち品詞の細分類に関する情報を格納する。詳細は，3.3 節を参照のこと。

```
<!ELEMENT      その他の情報 1    (#PCDATA)>
```

■「その他の情報 2」要素

- 形態論情報の「その他の情報 2」，すなわち語形に関する情報を格納する。詳細は，3.3 節を参照のこと。

- この要素は，「辞書項目」要素と「用例」要素に現れるが，「辞書項目」要素では，個別辞書項目の場合のみ記述される。付与されるのが音便情報の場合は，「撥音便A」「促音便A」などと末尾に「A」が付加される。この仕様の詳細についても 3.3 節を参照のこと。

```
<!ELEMENT      その他の情報 2    (#PCDATA)>
```

■「用例集」要素

- 辞書項目に対する用例集を表す。
- 0 個以上の用例を含む（CSJ として配布したデータでは，最大一つ）。
- 用例には，前文脈，後文脈，活用形，その他の情報 2 が付与される。なお，活用型，代表形，代表表記などは親要素に記述されているので，「用例集」要素では陽に記述しない。
- 前文脈，後文脈の長さは，15 短単位である。短単位間は，空白で区切ってある。表記は，転記テキストの基本形に基づく。
- 「講演 ID」要素：当該短単位が含まれる転記テキストの講演 ID （2 章参照）
- 「転記情報」要素：当該短単位が含まれる転記基本単位に関する情報（2 章参照）
- 「転記基本形」要素：当該短単位の転記テキストにおける基本形
- 「転記発音形」要素：当該短単位の転記テキストにおける発音形

```
<!ELEMENT      用例集            (用例*)>
<!ELEMENT      用例              (講演 ID, 転記情報,
                                  前文脈，転記基本形，後文脈,
                                  転記発音形,
                                  活用形?，その他の情報 2)>
<!ELEMENT      講演 ID           (#PCDATA)>
<!ELEMENT      転記情報          (#PCDATA)>
<!ELEMENT      転記基本形        (#PCDATA)>
<!ELEMENT      前文脈            (#PCDATA)>
<!ELEMENT      後文脈            (#PCDATA)>
<!ELEMENT      転記発音形        (#PCDATA)>
```

4.4.4 活用表

4.4.4.1 概要

本節では，活用表に関する解説とデータ形式を示す。

ここで示す活用表は，短単位辞書中の活用語に対する活用語形を規定するものである。主として，短単位解析システムで利用することを想定して作成した。そのため，人手単位解析用に設計された人手解析単位データの活用型（活用の種類），活用形よりも詳細化されている。自動解析単位データに付与されている活用型，活用形はこの本活用表の仕様に準拠する。人手解析単位データの活用型，活用形と異なるのは，次の点である。

- 後続する単位により，未然形，連用形を細分化した。「未然形 1 」のように 1〜4 の数字で細分化を表示

する。この数字を除去したものが人手解析単位データの活用形に対応する。

- 活用型は，「カ行五段１」，「カ行五段２」といった形式で，細分化している。活用形と同様，末尾の数字を除去したものが人手解析単位データの活用の種類に対応する（なお，人手解析単位データでも細分化されている「文語形容詞型１～３」は除く。また，CSJ の自動解析単位データでは，上一段活用の詳細化がなされていない）。

4.4.4.2　活用表の記述例

活用表は，XML で記述している。次に示したのは，「カ行五段１」動詞の活用表の例である。個々の活用は，conj 要素で記述され，属性として品詞（pos），口語・文語の別（style），活用名（name），行（column）を持つ。個々の活用語形は form 要素として列挙される。

```
<conj pos ="動詞" style="口語" name="五段" column="カ行１">
  <form name="未然形１"><f_item>か</f_item></form>
  <form name="未然形２"><f_item>こ</f_item></form>
  <form name="未然形３"><f_item>か</f_item></form>
  <form name="未然形４"><f_item>か</f_item></form>
  <form name="連用形１"><f_item>き</f_item></form>
  <form name="連用形２"><f_item sound_change="イ音便">い</f_item></form>
  <form name="終止形"><f_item>く</f_item></form>
  <form name="連体形"><f_item>く</f_item></form>
  <form name="仮定形"><f_item>け</f_item></form>
  <form name="命令形"><f_item>け</f_item></form>
  <example><e_item>書く</e_item><e_item>動く</e_item></example>
</conj>
```

4.4.4.3　DTD

活用表の XML 文書形式は，次の DTD で定義される。

■conjtable 要素（活用表）

- conjtable 要素は，活用表自体を表し，一つ以上の conj 要素（活用型）から構成される。
- 属性 version は，活用表の版を表す。
- 属性 name は，活用表の名称を表す。
- 属性 release_date は，リリース日の情報を表す。

```
<!ELEMENT conjtable (conj)+>
<!ATTLIST conjtable
         version         CDATA   #REQUIRED
         name            CDATA   #REQUIRED
         release_date    CDATA   #REQUIRED
>
```

■conj 要素（活用型）

- conj 要素は，活用型を表す。活用型要素は，複数の form 要素からなり，個々の活用形の語形は form 要素で記述される。また，当該活用型の語例を格納する example 要素を一つ以上含む。
- pos 属性は，当該活用型の品詞を表す。
- style 属性は，当該活用型の口語，文語の別を表す。
- name 属性は，当該活用型の名称を表す。
- column 属性は，当該活用型の活用行を表す。

```
<!ELEMENT conj ((form+),example?)>
<!ATTLIST conj
        pos             CDATA   #REQUIRED
        style           CDATA   #REQUIRED
        name            CDATA   #REQUIRED
        column          CDATA   #IMPLIED
>
```

■form 要素（活用形）

- form 要素は，活用型における個々の活用語形を表す。複数の活用語形を持つことを考慮し（例：上一段動詞の命令形），form 要素は，0 個以上の f_item 要素を含む。
  - 活用語形は，f_item 要素に格納される。
  - 活用語形中に語幹に相当する部分が存在する場合は，その部分を base タグでマークアップする。例えば，表 4.6 中のナ行上一段動詞「にる」の「に」は base 要素となる。
  - f_item の sound_change 属性は，音便の種類を記述する。
- name 属性は，活用形の名称を表す。

```
<!ELEMENT form (f_item)*>
<!ATTLIST form
        name            CDATA   #REQUIRED
>
<!ELEMENT f_item (#PCDATA|base)*>
<!ATTLIST f_item
        sound_change    CDATA   #IMPLIED
>
```

■example 要素（語例）

- 個々の活用型の語例を格納する。語例は終止形で表記する。CSJ として公開した活用表には，一つか二つの語例を示した。複数の語例を示すことを考慮し，実際の語例は e_item 要素として列挙する。

```
<!ELEMENT example (e_item)+>
<!ELEMENT e_item (#PCDATA)*>
```

4.4.4.4　解説

この後の節では，次に示す五つの活用表について解説を行う。なお，活用形に関する説明は，後続の要素に注記が必要なものしか示していない。また，実際の単位データには，活用形が「語幹」となる場合もあることに注意されたい。

- 口語動詞，文語動詞
- 形容詞
- 接尾辞
- 助動詞

■口語動詞（表 4.6，4.7）

- **活用形**
  - **未然形 1 ：** 助動詞ナイに連なる形
  - **未然形 2 ：** 助動詞ウ・ヨウに連なる形
  - **未然形 3 ：** 助動詞レル・ラレルに連なる形
  - **未然形 4 ：** 助動詞ズに連なる形
  - **連用形 1 ：** 助動詞マスに連なる形
  - **連用形 2 ：** 助動詞タ，接続助詞テに連なる形
  - **終止形 ：** 文末，または，引用の助詞ト，助動詞ダ・デスに連なる形
  - **連体形 ：** 名詞，または，準体助詞ノに連なる形
  - **仮定形 ：** 接続助詞バに連なる形
  - **命令形 ：** 文末，または，引用の助詞トに連なって命令を表わす形
- **注記**
  - 「生ずる」「生じる」のように，「ずる」，および，「じる」語尾の付きうる語については，通常，別語として扱い，それぞれサ行変格，ザ行上一段と称しているが，本活用表では，この種の動詞を一つにまとめて，一類を立て，ザ行変格と呼ぶ。代表形は「〜ずる」とするが，活用語尾としては，サ変と上一段の両方の語形を認める。
  - 未然形が四つに分かれるのは，サ変とザ変のためであり，他の動詞においては，一ないし二で十分である。
  - 語尾の一部が括弧で括ってあるのは，語幹と重なる部分であり，配布した XML 形式の活用表では，base 要素として記述されている。なお，語幹は機械的に語尾として認定できるように仮名表記している。
  - 一つの欄に複数の語形が入っている場合があるが，これは機能的には同じものである。

表 4.6 口語動詞 [1/2]

活用型	活用行	未然形 1	未然形 2	未然形 3	未然形 4	連用形 1	連用形 2	終止形	連体形	仮定形	命令形	語例
五段	カ行 1	か	こ	か	か	き	い [イ音便]	く	く	け	け	書く，動く
	カ行 2	か	こ	か	か	き	っ [促音便]	く	く	け	け	行く
	ガ行	が	ご	が	が	ぎ	い [イ音便]	ぐ	ぐ	げ	げ	漕ぐ，騒ぐ
	サ行	さ	そ	さ	さ	し	し	す	す	せ	せ	探す，渡す
	タ行	た	と	た	た	ち	っ [促音便]	つ	つ	て	て	立つ，勝つ
	ナ行	な	の	な	な	に	ん [撥音便]	ぬ	ぬ	ね	ね	死ぬ
	バ行	ば	ぼ	ば	ば	び	ん [撥音便]	ぶ	ぶ	べ	べ	飛ぶ，及ぶ
	マ行	ま	も	ま	ま	み	ん [撥音便]	む	む	め	め	沈む，頼む
	ラ行 1	ら	ろ	ら	ら	り	っ [促音便]	る	る	れ	れ	取る，凍る
	ラ行 2	ら	ろ	ら	ら	い	っ [促音便]	る	る	れ	い	いらっしゃる なさる
	ワア行	わ	お	わ	わ	い	っ [促音便]	う	う	え	え	言う，思う
上一段	ア行 1	(い)	(い)	(い)	(い)	(い)	(い)	(い)る	(い)る	(い)れ	(い) ろ (い) よ	居る，射る
	ア行 2	い	い	い	い	い	い	いる	いる	いれ	いろ いよ	強いる 報いる
	カ行 1	(き)	(き)	(き)	(き)	(き)	(き)	(き)る	(き)る	(き)れ	(き) ろ (き) よ	着る
	カ行 2	き	き	き	き	き	き	きる	きる	きれ	きろ きよ	起きる 尽きる
	ガ行	ぎ	ぎ	ぎ	ぎ	ぎ	ぎ	ぎる	ぎる	ぎれ	ぎろ ぎよ	過ぎる
	ザ行	じ	じ	じ	じ	じ	じ	じる	じる	じれ	じろ じよ	閉じる 信じる
	タ行	ち	ち	ち	ち	ち	ち	ちる	ちる	ちれ	ちろ ちよ	落ちる 朽ちる
	ナ行	(に)	(に)	(に)	(に)	(に)	(に)	(に)る	(に)る	(に)れ	(に) ろ (に) よ	似る，煮る
	バ行	び	び	び	び	び	び	びる	びる	びれ	びろ びよ	伸びる 詫びる
	マ行 1	(み)	(み)	(み)	(み)	(み)	(み)	(み)る	(み)る	(み)れ	(み) ろ (み) よ	見る
	マ行 2	み	み	み	み	み	み	みる	みる	みれ	みろ みよ	沁みる 試みる
	ラ行	り	り	り	り	り	り	りる	りる	りれ	りろ りよ	降りる 借りる

表 4.7　口語動詞 [2/2]

活用型	活用行	未然形 1	未然形 2	未然形 3	未然形 4	連用形 1	連用形 2	終止形	連体形	仮定形	命令形	語例
下一段	ア行 1	(え)	(え)	(え)	(え)	(え)	(え)	(え) る (う) る	(え) る (う) る	(え) れ	(え) ろ (え) よ	得る 心得る
	ア行 2	え	え	え	え	え	え	える	える	えれ	えろ えよ	考える 見える
	カ行	け	け	け	け	け	け	ける	ける	けれ	けろ けよ	受ける 付ける
	ガ行	げ	げ	げ	げ	げ	げ	げる	げる	げれ,	げろ げよ	挙げる 逃げる
	サ行	せ	せ	せ	せ	せ	せ	せる	せる	せれ	せろ せよ	見せる 任せる
	ザ行	ぜ	ぜ	ぜ	ぜ	ぜ	ぜ	ぜる	ぜる	ぜれ,	ぜろ ぜよ	混ぜる 爆ぜる
	タ行	て	て	て	て	て	て	てる	てる	てれ	てろ てよ	捨てる 立てる
	ダ行 1	(で)	(で)	(で)	(で)	(で)	(で)	(で) る	(で) る	(で) れ	(で) ろ (で) よ	出る
	ダ行 2	で	で	で	で	で	で	でる	でる	でれ	でろ でよ	撫でる 奏でる
	ナ行 1	(ね)	(ね)	(ね)	(ね)	(ね)	(ね)	(ね) る	(ね) る	(ね) れ	(ね) ろ (ね) よ	寝る 真似る
	ナ行 2	ね	ね	ね	ね	ね	ね	ねる	ねる	ねれ	ねろ ねよ	重ねる 跳ねる
	ハ行	(へ)	(へ)	(へ)	(へ)	(へ)	(へ)	(へ) る	(へ) る	(へ) れ	(へ) ろ (へ) よ	経る
	バ行	べ	べ	べ	べ	べ	べ	べる	べる	べれ	べろ べよ	述べる 食べる
	マ行	め	め	め	め	め	め	める	める	めれ	めろ めよ	決める 止める
	ラ行 1	れ	れ	れ	れ	れ	れ	れる	れる	れれ	れろ れよ	知れる 崩れる
	ラ行 2	れ	れ	れ	れ	れ	れ	れる	れる	れれ	れ, れろ れよ	呉れる
変格	カ行	(こ)	(こ)	(こ)	(こ)	(き)	(き)	(く) る	(く) る	(く) れ	(こ) い	来る
	サ行	(し)	(し)	(さ)	(せ)	(し)	(し)	(す) る	(す) る	(す) れ	(し) ろ (せ) よ	為る
	ザ行	じ	じ	じ ぜ	じ ぜ	じ	じ	じる ずる	じる ずる	じれ ずれ	じろ ぜよ	生ずる 禁ずる

■文語動詞（表 4.8）

- **活用形**
  - **未然形：** 口語動詞の未然形 1〜4 の機能を併せ持つほか，助詞バが付いて，仮定を表わす。
  - **連用形：** 助動詞キ・ケリ・タリに連なる形
  - **終止形：** 文末，または，引用の助詞トに連なる形
  - **連体形：** 名詞に連なる形
  - **已然形：** 接続助詞バ・ドが付いて，既定（順接・逆接）を表わす。
  - **命令形：** 文末，または，引用の助詞トに連なって，命令を表わす形
- **注記**
  - 下一段活用としては，「蹴る」が挙げられるが，文語といっても近世においては，四段に変わったものと思われるので，実際にはほとんど出現しない。
  - 基本形，並びに，代表表記が現代仮名づかいなので，行名をどうするかが問題になる。「思う」などをワア行四段とするか，ハ行四段とするかについては，後者とした。それに合わせて，「強う（強いる）」はハ行上二段，「与う，仕う，捕らう」などはハ行下二段とした。また「植う，餓う，据う」など，元がワ行下二段活用のものをワ行としたが，これも表記上はハ行２と同じである。

表 4.8　文語動詞

活用型	活用行	未然形	連用形	終止形	連体形	已然形	命令形	語例
四段	カ行	か	き	く	く	け	け	書く，行く
	ガ行	が	ぎ	ぐ	ぐ	げ	げ	漕ぐ，騒ぐ
	サ行	さ	し	す	す	せ	せ	探す，渡す
	タ行	た	ち	つ	つ	て	て	立つ，勝つ
	ハ行	わ	い	う	う	え	え	言う，給う
	バ行	ば	び	ぶ	ぶ	べ	べ	呼ぶ，忍ぶ
	マ行	ま	み	む	む	め	め	沈む，頼む
	ラ行	ら	り	る	る	れ	れ	取る，遣る
上一段	ア行	(い)	(い)	(い) る	(い) る	(い) れ	(い) よ	居る，射る
	カ行	(き)	(き)	(き) る	(き) る	(き) れ	(き) よ	着る
	ナ行	(に)	(に)	(に) る	(に) る	(に) れ	(に) よ	似る，煮る
	ハ行	(ひ)	(ひ)	(ひ) る	(ひ) る	(ひ) れ	(ひ) よ	干る
	マ行	(み)	(み)	(み) る	(み) る	(み) れ	(み) よ	見る
上二段	カ行	き	き	く	くる	くれ	きよ	起く，尽く
	ガ行	ぎ	ぎ	ぐ	ぐる	ぐれ	ぎよ	過ぐ
	タ行	ち	ち	つ	つる	つれ	ちよ	落つ
	ダ行	じ	じ	ず	ずる	ずれ	じよ	閉ず，恥ず
	ハ行	い	い	う	うる	うれ	いよ	強う
	バ行	び	び	ぶ	ぶる	ぶれ	びよ	伸ぶ，詫ぶ
	マ行	み	み	む	むる	むれ	みよ	沁む，恨む
	ヤ行	い	い	ゆ	ゆる	ゆれ	いよ	悔ゆ，報ゆ
	ラ行	り	り	る	るる	るれ	りよ	降る，借る
下一段	カ行	(け)	(け)	(け) る	(け) る	(け) れ	(け) よ	蹴る
下二段	ア行	(え)	(え)	(う)	(う) る	(う) れ	(え) よ	得
	カ行	け	け	く	くる	くれ	けよ	受く，付く
	ガ行	げ	げ	ぐ	ぐる	ぐれ	げよ	挙ぐ，遂ぐ
	サ行	せ	せ	す	する	すれ	せよ	見す，任す
	ザ行	ぜ	ぜ	ず	ずる	ずれ	ぜよ	混ず，爆ず
	タ行	て	て	つ	つる	つれ	てよ	捨つ，立つ
	ダ行	で	で	ず	ずる	ずれ	でよ	出ず，撫ず
	ナ行 1	(ね)	(ね)	(ぬ)	(ぬ) る	(ぬ) れ	(ね) よ	寝
	ナ行 2	ね	ね	ぬ	ぬる	ぬれ	ねよ	重ぬ，跳ぬ
	ハ行 1	(へ)	(へ)	(ふ)	(ふ) る	(ふ) れ	(へ) よ	経
	ハ行 2	え	え	う	うる	うれ	えよ	与う，仕う
	バ行	べ	べ	ぶ	ぶる	ぶれ	べよ	食ぶ，述ぶ
	マ行	め	め	む	むる	むれ	めよ	極む，止む
	ヤ行	え	え	ゆ	ゆる	ゆれ	えよ	消ゆ，見ゆ
	ラ行	れ	れ	る	るる	るれ	れよ	恐る，崩る
	ワ行	え	え	う	うる	うれ	えよ	植う，据う
変格	カ行	(こ)	(き)	(く)	(く) る	(く) れ	(こ)	来
	サ行	(せ)	(し)	(す)	(す) る	(す) れ	(せ)，(せ) よ	為
	ザ行	ぜ	じ	ず	ずる	ずれ	ぜよ	禁ず，存ず
	ナ行	な	に	ぬ	ぬる	ぬれ	ね	死ぬ，去ぬ
	ラ行	ら	り	り	る	れ	れ	有り，居り

■形容詞（表 4.9，4.10）

- **活用形 （口語）**
  - **未然形：** 助動詞ウに連接
  - **連用形 1：** 中止
  - **連用形 2：** 助動詞タに連接
- **活用形 （文語）**
  - **未然形 1：** 助動詞ズ・ムに連接
  - **未然形 2：** 助詞バに連なって仮定を表わす
  - **連用形 1：** 中止
  - **連用形 2：** 助動詞キ・ケリに連接
- **注記**
  - 動詞の場合も同じであるが，文語では，未然形に接続助詞「ば」が付いて，仮定を表わし，已然形＋「ば」は別の意味を持つ。
  - 上記の他に，連用形 1 のウ音便があるが，これが語幹と融合して音韻変化を生ずるので，詳細に記述するためには，行によって分ける必要がある。したがって，これは活用表に記載せず，実際に出現したものだけを個別辞書項目として，短単位辞書に登録した。

    例： お/はよう（はやい），お/めでとう（めでたい），ありがとう（ありがたい），
    うれしゅう（うれしい），少のう（少ない）
  - 存在しない活用形には，「—」としてある。

表 4.9 形容詞（口語）

活用型	未然形	連用形 1	連用形 2	終止形	連体形	仮定形	命令形	語例
形容詞型	かろ	く	かっ [促音便]	い	い	けれ	—	無い，良い，正しい

表 4.10 形容詞（文語）

活用型	未然形 1	未然形 2	連用形 1	連用形 2	終止形	連体形	已然形	命令形	語例
文語形容詞型 1	から	く	く	かり	し	き	けれ	かれ	無し 良し
文語形容詞型 2	しから	しく	しく	しかり	し	しき	しけれ	しかれ	悪し 正し
文語形容詞型 3	から	く	く	かり	し かり	き	けれ	かれ	多し

■接尾辞

- 接尾辞のうち，動詞性の活用をもつものには，動詞と同じ活用型が記入されている。
  例: がる（ラ行五段），兼ねる（ナ行下一段）
- 形容詞性の活用を持つものには，形容詞型，文語形容詞型 1，文語形容詞型 2 などの情報が付く。
  例: がたい，がたし，がましい，くさい，たらしい，づらい，にくい，ぽい，やすい

■助動詞（表 4.11，4.12，4.13，4.14，4.15）

- **活用型**
  - 活用型を (1) 口語動詞型，(2) 文語動詞型，(3) 口語形容詞型，(4) 文語形容詞型に大別する。
  - 基本的に，(2) と (4) は，純粋に文語のみの場合とし，語形や用法が混じるようなものは，口語の方に入れる。しかし，助動詞には古くからあるものが多く，語形面でも，名付けの面でもきれいに分けるのが困難である。例えば，「ず」に「ば」の付く形は，「ずば」「ざれば」「ねば」の三つある。このうち，「ずば」は意味的には仮定だが，呼び名としては文語の未然形だと思われる（CSJ には存在しない）。「ざれば」は已然形といってよいと思うが，「ねば」には両方の用法がある。また，「まずれば」というのはやや古めかしい言い方であるが，やはり両様に使いそうである。「ざる」は本来文語であるが，「ざるをえない」の形で口語にもしばしば用いられる。そのため，文語，口語の両方に同一見出し，同一語形が含まれる場合がある。
  - 自立語の場合と異なるのは，助動詞間の接続の順序がほぼ決まっているため，それが活用にも影響することである。そのため，存在しない部分は「—」で埋めてある。
- **活用形**
  - **口語動詞型**
    - **未然形 1**： ナイ・ズ・レル・ラレルに連接
    - **未然形 2**： ウ・ヨウに連接
    - **連用形 1**： マスに連接
    - **連用形 2**： タに連接
  - **文語動詞型**
    - **未然形**： ズ・ル・ラル・ムに連接
    - **連用形**： キ・ケリ・タリに連接
  - **口語形容詞型**
    - **連用形 1**： 中止
    - **連用形 2**： タに連接
  - **文語形容詞型**
    - **未然形 1**： ズ・ムに連接
    - **未然形 2**： バに連接
    - **連用形 1**： 中止
    - **連用形 2**： キ・ケリに連接
- **注記**
  - 動詞における未然形 3，4 が未然形 1 に統合される他は，口語動詞と同じである。
  - タとダのように連濁によって出現形が変わったものは，同一見出しとしている。
  - コーパス中に出現しないものでも，データの拡張ということも考えて，日常見聞きする範囲のものは活用表に含めた（**例**： 教師にある/まじき/振舞い，得/べかり/し/収入）。
  - 同語源であっても，文語と口語とでは，代表形が異なるのが普通であるが，「ず」の場合だけは例外的に同一である。

表 4.11 助動詞（口語動詞型）[1/2]

見出し語	未然形 1	未然形 2	連用形 1	連用形 2	終止形	連体形	仮定形	命令形
せる	せ，さ	せ	せ，し	せ，し	せる	せる	せれ	せろ せよ
させる	させ	させ	させ，さし	させ，さし	させる	させる	させれ	させろ させよ
しめる	しめ	しめ	しめ	しめ	しめる	しめる	しめれ	しめよ
す	—	—	—	—	す	す	—	—
さす	—	—	—	—	さす	さす	—	—
れる	れ	れ	れ	れ	れる	れる	れれ	れろ れよ
られる	られ	られ	られ	られ	られる	られる	られれ	られろ られよ
たがる	たがら	たがろ	たがり	たがっ [促音便]	たがる	たがる	たがれ	—
ます	ませ	ましょ	—	まし	ます まする	ます まする	ますれ	まし ませ
んす	んせ	んしょ	—	んし	んす	んす	んすれ	んし んせ
じゃ	—	じゃろ	—	じゃっ [促音便]	じゃ	—	—	—
だ	—	だろ	で，に	だっ [促音便]	だ	な	なら	—
です	—	でしょ	—	でし	です	です	—	—
どす	—	—	—	どし	どす	—	—	—
やす	やせ	やしょ	—	やし	やす	—	—	やす
はる	はら	—	はり	はっ [促音便]	はる	はる	—	はれ
ざます	—	—	—	ざまし	ざます	—	—	—
や	や	やろ	—	やっ [促音便]	や	—	—	—
ねん	—	—	—	—	ねん	—	—	—
ず	—	—	ず	—	ぬ，ん，ず	ぬ，ん ざる	ね	—
た	—	たろ だろ	—	—	た，だ	た，だ	たら，だら	—
う	—	—	—	—	う	う	—	—
よう	—	—	—	—	よう	—	—	—
てる	て，で	て，で	て，で	て，で	てる，でる	てる，でる	てれ，でれ	てろ でろ
てらっしゃる	てらっしゃら でらっしゃら	—	てらっしゃい でらっしゃい	てらっしゃっ [促音便]，てらし，でらっしゃっ [促音便]，でらし	てらっしゃる でらっしゃる	てらっしゃる でらっしゃる	てらっしゃれ でらっしゃれ	てらっしゃい，でらっしゃい

表 4.12　助動詞（口語動詞型）[2/2]

見出し語	未然形 1	未然形 2	連用形 1	連用形 2	終止形	連体形	仮定形	命令形
てく	てか でか	てこ でこ	てき でき	てっ [促音便] でっ [促音便]	てく，でく	てく でく	てけ でけ	てけ でけ
てける	てけ でけ	—	てけ でけ	てけ でけ	てける でける	てける でける	てけれ でけれ	—
とく	とか どか	とこ どこ	とき どき	とい [イ音便] どい [イ音便]	とく どく	とく どく	とけ どけ	とけ どけ
とける	とけ どけ	—	とけ どけ	とけ どけ	とける どける	とける どける	とけれ どけれ	—
とる	とら どら	とろ どろ	とり どり	とっ [促音便] どっ [促音便]	とる どる	とる どる	とれ どれ	とれ どれ
ちまう	ちまわ じまわ	ちまお じまお	ちまい じまい	ちまっ [促音便] じまっ [促音便]	ちまう じまう	ちまう じまう	ちまえ じまえ	ちまえ じまえ
ちゃう	ちゃわ じゃわ	ちゃお じゃお	ちゃい じゃい	ちゃっ [促音便] じゃっ [促音便]	ちゃう じゃう	ちゃう じゃう	ちゃえ じゃえ	ちゃえ じゃえ
たげる	たげ	たげ	たげ	たげ	たげる	たげる	たげれ	たげろ
たる	たら	たろ	—	たっ [促音便]	たる	たる	たれ	たれ たり
ちゃる	—	ちゃろ	—	—	ちゃる	ちゃる	ちゃれ	ちゃれ
つう	—	—	—	つっ [促音便] っつっ [促音便] ちゅっ [促音便] っちゅっ [促音便]	つう，っつう ちゅう っちゅう てえ，ってえ	つう，っつう ちゅう っちゅう，て	ちゃ っちゃ	—
しゃる	しゃら っしゃら	—	しゃり っしゃり	しゃっ [促音便] っしゃっ [促音便]	しゃる っしゃる	しゃる っしゃる	しゃれ っしゃれ	しゃれ っしゃれ

表 4.13　助動詞（文語動詞型）

見出し語	未然形	連用形	終止形	連体形	已然形	命令形
む	—	—	む，ん	む，ん	め	—
らむ	—	—	らむ，らん	らむ，らん	らめ	—
うず	—	—	うず	うずる	うずれ	—
めり	—	—	めり	める	めれ	—
べらなり	—	—	べらなり	べらなる	べらなれ	—
き	せ	—	き	し	しか	—
けり	—	—	けり	ける	けれ	—
つ	て	て	つ	つる	つれ	てよ
ぬ	な	に	ぬ	ぬる	ぬれ	ね
り	—	—	り	る	れ	—
なり	なら	に，なり	なり	なる	なれ	なれ
たり	たら	と，たり	たり	たる	たれ	たれ
る	れ	れ	る	るる	るれ	れよ
らる	られ	られ	らる	らるる	らるれ	られよ
しむ	しめ	しめ	しむ	しむる	しむれ	しめよ

表 4.14　助動詞（口語形容詞型）

見出し語	未然形	連用形 1	連用形 2	終止形	連体形	仮定形	命令形
たい	たかろ	たく とう [ウ音便]	たかっ [促音便]	たい	たい	たけれ	—
ない	なかろ	なく	なかっ [促音便]	ない	ない	なけれ	—
らしい	らしかろ	らしく らしゅう [ウ音便]	らしかっ [促音便]	らしい	らしい	らしけれ	—
まい	—	—	—	まい	まい	—	—

表 4.15　助動詞（文語形容詞型）

見出し語	未然形 1	未然形 2	連用形 1	連用形 2	終止形	連体形	已然形	命令形
たし	たから	たく	たく	たかり	たし	たき	たけれ	—
べし	べから	—	べく	べかり	べし	べき	べけれ	—
ごとし	—	—	ごとく	—	ごとし	ごとき	—	—
らし	らしから	—	—	—	らし	らしき	—	—
まじ	—	—	まじく	—	まじ	まじき	まじけれ	—
じ	—	—	—	—	じ	じ	じ	—
ず	ざら	ず	ず	ざり	ず	ざる	ざれ，ね	—

4.4.5　短単位辞書の構築

短単位辞書の構築は，大きく分けて二つの方法で行った。一つは，人手短解析単位データを構築する過程で派生的に蓄積していくものである。もう一つは，人手解析対象とならない転記テキスト（自動解析単位データ用の転記テキスト）に対して，能動的に短単位辞書を構築していったものである。この後の節では，それぞれの構築方法について説明していくことにする。

4.4.5.1　人手解析単位データの構築に伴うもの

4.1.1 節で述べたように，人手短単位解析では，すべての単位データに対して，人手でチェック・修正を行う。したがって，基本的には，人手短単位解析の結果の異なりを求めれば，短単位辞書を作成することができる。

ただし，人手単位解析の誤りや個別辞書項目として登録すべき単位データに対応する必要がある。それに付け加え，短単位辞書の作成時期を考慮する必要もある。なぜならば，短単位辞書は修正ツールで単位データの整合性をチェックするために利用されているので，頻繁な変更は修正者に混乱を引き起こす恐れがあるからである。そのため，プロジェクト中は，転記テキストのリリーススケジュールに合わせて，短単位辞書の構築を行うことが多かった。これは，新規転記テキストの追加やバージョンアップは，短単位辞書の辞書項目の増減[*37]に大きな影響を与えるからである。

以上のことから，ここでは，転記テキストのリリーススケジュールを考慮した短単位辞書の作成手順を次に示す。なお，それぞれの辞書項目に付与される用例については，この後の 4.4.5.3 節で述べる。

(1) 新規転記テキストのリリース
(2) 新規転記テキストを 4.2.2 節の方法で人手短単位解析し，単位データベースに登録する。
(3) 登録された短単位データを人手修正する。
(4) 一定量の人手修正が終了した時点で，人手修正済みの短単位データから短単位辞書を生成する。この段階では，短単位に付与されている次の情報の組で異なり作成する。
- 代表形，代表表記
- 品詞，活用型，その他の情報 1

(5) 新規に作成された短単位辞書と既存の短単位辞書との差分を作成する。
(6) 差分として検出された辞書項目の用例を人手でチェックし，本当に新たな辞書項目かをチェックする。
(7) 新たな辞書項目であることが確認された時点で，該当する用例から基本形（活用語の場合は，終止形の語形）のリストを作成する。さらに，活用型，活用形を詳細化し，短単位辞書に追加する。
(8) 新たに作成された短単位辞書に基づいて，すべての単位データの出現形をチェックし，個別辞書項目を探す。検出された単位データは人手チェックを行った上で個別辞書項目として短単位辞書に追加する。
(9) 新たに生成された短単位辞書をデータベース管理システムに登録する（修正ツールの整合性チェック用の短単位辞書が更新されることを意味する）。

[*37] 新たな転記テキストが加わることによる新規辞書項目の追加だけでなく，転記テキストの表記の統一などによる辞書項目の減少も考えられる。

4.4.5.2 自動解析単位データの人手修正に伴うもの

次に，自動解析単位データに対する短単位辞書の構築方法について説明する。自動解析単位データは，人手解析単位データと異なり，部分的にしか人手のチェック・修正を行わない。したがって，解析結果の異なりをそのまま短単位辞書に登録することはできない。そこで，次の二つの方法で短単位辞書の増補を行った。

■「茶筌」の未知語をチェックする方法 この方法は，情報通信研究機構の自動短単位解析システムと学習用の短単位データ（つまり，人手解析短単位データ）の準備ができるまでの暫定的な方法である。プロジェクト中期頃には，自動単位解析対象の転記テキストの準備が整いつつ合ったが，自動短単位解析システムと学習用の短単位データの準備ができていなかった。そこで，「茶筌」を使って，転記テキストを解析し，品詞が未知語と認定された文字列を人手でチェックし，短単位辞書に登録することにした。

チェックの方法は，通常の人手短単位解析手順（4.2.3 節の図 4.7）のうち，「基本形抽出」「茶筌による形態素解析」および「KWIC 作成」だけを行ったのちに，関係データベース管理システムに通常の単位データベースとして登録する[*38]。そして，品詞が未知語の単位データを検索し，適宜，短単位辞書に追加する。

■情報通信研究機構の自動短単位解析システムによる方法 4.1 節で述べたように，CSJ の自動単位解析は，情報通信研究機構の自動短単位解析システムによって行われた。自動短単位解析システムは，最大エントロピー法を採用しており，さまざまな素性を使って解析を行う（内元 2004）。国語研究所では，短単位辞書と活用表，および，自動単位解析結果の一部に対する人手修正結果を，情報通信研究機構に提供することにより，自動解析単位データの精度向上を図った。両組織間のデータのやりとりの流れを次に示す。この流れを繰り返すことにより，解析精度を向上させる。

(1) 短単位辞書と活用表の提供（国語研究所）
(2) 自動短単位解析システムによる解析（情報通信研究機構）
(3) 解析された単位データに対する部分的人手修正（国語研究所）
(4) 短単位辞書と活用表の更新（国語研究所）

次に，実際の短単位辞書構築方法について説明する。辞書の構築は，上記（3）の人手修正作業の中で実施する。具体的な方法は，次のとおりである。

(1) 自動短単位解析された単位データを関係データベース管理システムに登録する。自動解析単位データに付与されている確信度を単位データベースの「予備 1 」フィールドに格納する。確信度は，単位データの正確さの度合を示し，0〜1 の値を取る。
(2) 次の条件に一致する転記基本単位を抽出し[*39]，抽出された転記基本単位中のすべての短単位を修正ツールでチェックする。
- 包含する短単位の確信度の積が 95% 以下の転記基本単位
- ただし，チェックする短単位の数が当該転記テキストに含まれる短単位数の 1% を越える場合は，上位の 1% を選択してチェックする。

[*38] この作業により，人手単位解析対象にならない転記テキストを単位データベースに登録・運用できることを，実際のデータを使って確認することができた。
[*39] 抽出には，情報通信研究機構の内元清貴氏作成のスクリプトを用いた。

(3) チェックしたデータは，情報通信研究機構に学習用短単位データとして提供する。

(4) 学習用短単位データを人手解析単位データと同様に処理（4.4.5.1 節参照）し，短単位辞書に追加する。

4.4.5.3 用例の付与

用例の付与は，短単位辞書に登録されている辞書項目と合致する短単位データを単位データベースから検索し，人手でその採否を決定した。手順は，次のとおりである。なお，用例の付与を行ったのは，自動解析単位データの解析精度が向上したプロジェクト後期である。

(1) 単位データベース（すべての人手・自動解析単位データ）から，各辞書項目の用例をそれぞれ 10 例ランダムに抽出する。ただし，抽出する際は，人手修正を行ったものを優先する。

(2) 下記の基準に基づき，好ましい用例を人手で選択する。選択作業には，修正ツールを用いる。選択する際は，まず，抽出された単位データを Excel ファイルの形式にする。そして，用例として採用するレコードの「予備 3」フィールドに採用情報を付与した。CSJ として配布した短単位辞書には 1 辞書項目 1 用例を収録しているので，一つの単位データだけに採用情報を付与する。

- 基本形部分が当該の見出しに対応する語であること
- 基本形部分を当該の見出しと判断するに十分な文脈を持っていること。この条件により，フィーラーや言いよどみなど文脈の特定に役立たない要素をなるべく排除する。
- 基本形部分にタグがついている場合は，タグがついていない用例を優先する。この条件は，「(? 言葉)」のように不確かな用例や「(M 言葉)」のようにメタ的に使われている用例を排除するためである。

4.5 まとめ

本章では，短単位・長単位データベース，および，その構築環境について，次のことを示した。

- 短単位・長単位データベースの設計・運用方法
- 単位データベース修正ツールの設計・実現方法
- 短単位辞書の設計・構築方法

4.1.1 節で述べたとおり，短単位・長単位データベース，および，その構築環境は，人手単位解析の精度，効率，整合性を考慮し，次の点に焦点を当てて設計・実現した。

- 単位データの人手解析・修正の支援による精度，効率の向上
- 単位データの言語学的・データ形式上の整合性の維持

実現した単位データ構築環境は，プロジェクト完了時まで大きな問題もなく動作し，単位データベースも目標の解析精度を達成している（もちろん，解析精度の高さは，修正者の努力によるところが大きい）ことから，上記の目標自体は達成したと考えられる。ただし，今後，より巨大なデータベースを構築し，それを言語研究に応用する場合は，単位データベースと単位辞書との統合や多様なアノテーションへの対応，より柔軟な条件での検索などが求められるようになると思われる。

謝辞

単位データベースの構築のみならず，短単位辞書，および，活用表の作成に多大な貢献をされた木村睦子氏（元国立国語研究所国語辞典編集室長，CSJ のプロジェクト期間中は非常勤研究員），単位データ修正ツールを共同で開発してくださった小島丈幸氏（元国立国語研究所非常勤研究員，現管理工学研究所），転記テキストと単位データベースとの同期や外来語辞書の提供をしてくださった西川賢哉氏（慶應義塾大学大学院・国語研究所非常勤研究員），自動単位解析デーや単位データの誤り候補の提供してくださった内元清貴氏，村田真樹氏をはじめとする情報通信研究機構（旧通信総合研究所）の方々，大量の単位データに対して人手修正をするとともに，単位データ修正ツールへの改良点を提案してくださった修正者各氏に心より感謝いたします。