

辞書ハイパーテキスト Hydra

根本治朗 内藤広志 山下真司 松山洋一 柵木孝一
キャノン株式会社 情報システム研究所

ハイパーテキスト技術を用いると、自由な文書の読み書きや強力な検索機能などが実現できるが、この技術を既存文書に対して適用するためには文書をノンリニア構造に変換しなくてはならない。本稿では、計算機可読辞書を題材に、既存文書をハイパーテキスト化する手法について述べる。この手法の特徴は、ハイパーテキストデータモデルを用いた原データの構造分析、抽象辞書を基にした汎用化への対応、データ記述の標準化への対応である。また、この手法に基づいて開発した辞書ハイパーテキスト Hydra(Hypertext Dictionary Reading Accessories) の概要についても述べる。

Hypertext Dictionary Hydra

Jiro Nemoto, Hiroshi Naito, Shinji Yamashita, Youichi Matsuyama, Koichi Masegi
Information Systems Research Center, Canon Inc.
890-12 Kashimada, Saiwai-Ku, Kawasaki, 211, Japan

Hypertext technology produces many advantages on reading and writing documents and retrieving information. To apply this technology to existing documents, it is indispensable to convert linear structured documents into non-linear structure.

This paper describes a method to convert linear machine readable dictionary to non-linear hyper-structured dictionary. Noticeable features of the method are structure analysis of original documents using Hypertext Data Model, generalization based on abstract dictionary, and standardized data representation. In addition, outline of Hypertext Dictionary Hydra(Hypertext Dictionary Reading Accessories) which is developed by the method is described.

1 はじめに

ノードとリンクからなる単純なモデルによって文書表現するハイパーテキストが注目を集めている。これは、ハイパーテキストのネットワーク構造を用いることにより、自由な文書の読み書きや強力な検索機能などが実現できるためである。

今後、このハイパーテキスト技術は文書作成のあらゆる場面に浸透していくことが期待される。このとき、新しい文書にノンリニア構造を持たせることも重要であるが、既存のリニアな構造を持つテキストデータをハイパーテキストのノンリニア構造へ変換することも、過去の資産の有効利用のために極めて重要である^{[5] [7] [13] [17] [15]}。

そこで我々は、このハイパーテキスト化技術の辞書への適用手法の検討を行い、同時にその手法に基づいた辞書ハイパーテキスト Hydra(Hypertext Dictionary Reading Accessories) の試作を行った。

本論文では、まず2章で計算機可読辞書のハイパーテキスト化に関する課題について述べる。そして、3章で本手法の全体像並びに詳細について述べる。さらに、4章において現在開発中の辞書ハイパーテキスト Hydra を紹介する。

2 辞書のハイパーテキスト化

CD-ROM などの普及により、パソコンやワークステーション上で手軽に電子化辞書が利用できるようになってきた。しかしそれらの電子化辞書の多くは、冊子体の辞書と同様に見出し語による検索しか出来ない。

辞書がハイパーテキスト化されれば、そのフィルタリング、ブラウジング、ナビゲーションなど様々な検索・表示方法を享受できるようになる。例えば、フィルタリング機能を用いて一般ユーザにとってあまり重要でない情報の表示を止めることができる。あるいは、ナビゲーション機能を用いて語義中の類義語や反対語をリンクのアンカーとし、ユーザのマウスクリックによって対応する見出し語へ移動することが可能となる。また、外来語を示す情報をリンクとすることで複数辞書の結合も可能となる。

一般にハイパーテキスト化とは、原文書をノードに分割し、リンクを明示化する作業である。辞書データの場合、サイズが極めて大きい(数MB~数百MB)、したがって辞書のハイパーテキスト化手法の第一の要件として、計算機を用いて処理を自動化することが挙げられる。

しかし、冊子体の辞書をそのまま計算機のデータとして入力した計算機可読辞書の多くは各辞書項目が明確に分離されていなかったり、計算機可読辞書の記述法には多くの曖昧性がある^{[10] [16]}などの問題点がある。

例えば 図-1 に示すように、重要度によって「反対語」という辞書項目¹の出現する位置は異なる場合がある^[12]。

じょうき①②ジャウー【上記】-する 前に書きしるしてある・こと(文句)。↔下記

たんぱく①【淡泊】-な-に ①物事の感じ・味・色がしつこくない様子。あっさりしている様子。↔濃厚 ②物事にこだわらず、さっぱりしている様子。「-な人柄・-な態度」

たんよう②-エフ【単葉】①一枚の葉。②↔複葉 飛行機の主翼が一枚であること。「-機③」

じょうとう②ジャウー【上等】-な-に ①↔下等 ②等級が上である・様子(もの)。③すぐれていい・様子(もの)。

しょうがく②セウー 日【小額】↔高額 小さい単位の金額。「-紙幣⑤」
日【少額】↔多額 少ない金額。

きゅうき①キフー 日【吸気】胸に吸い込む息。↔呼気 日【給気】↔排気 内部に空気を送り込むこと。「-口③④」

図 1: 「反対語」の出現位置 ([12] より抜粋)

さらに、辞書項目の始まりや終わりを示すために種々の括弧、矢印などの特殊記号を用いることが多いが、図-2 に示すように、その用法が文脈によって変化する場合もある。これは、辞書の記述が文法の厳密さよりも読者にとって直感的に理解できることを重視しているためである。

*ぶんや【分野】(分けて)受け持つ範囲。

*ぶんぼうぐ【文房具】(「文房」は、書斎の意)ものを書くために必要な道具。

ぶんぼう【分封】(古)大名が自分の領地を割いて、臣下に分けること。

図 2: “()” の用法の揺れ([12] より抜粋)

このように辞書の構造に曖昧性や揺れがあるため、計算機が文脈を十分に理解できない現状ではハイパーテキスト化の完全自動化は困難である。したがって、辞書のハイパーテキスト化手法の第二の要件と

¹↔で示されている

²「分野」の()は語釈の補足的説明。「文房具」の()は熟語の構成。「分封」の()は説明的略号を表わす。

して、処理の最初の段階における人手による辞書の構造分析の支援が挙げられる。

また、他言語からの検索や、外来語の原語の検索などの機能を実現するためには、複数の種類の辞書をハイパーテキスト化する必要がある。しかし、辞書の構造は国語辞書、英和辞書、和英辞書などの辞書の種類によって違いがあり、かつ同種の辞書でも出版社によってその構造は異なる。したがって、辞書のハイパーテキスト化手法の第三の要件として、個々の辞書に依存しない汎用性が挙げられる。

以上をまとめると、辞書のハイパーテキスト化手法において重視すべき点として以下が挙げられる。

- ハイパーテキスト化処理の自動化
- 原文書の分析のサポート
- 手法の汎用化

3 辞書ハイパーテキスト構築手法

本章では Hydra 構築に際して採用したハイパーテキスト化手法について述べる。まず 3.1 節で我々の手法の概要について述べる。続いて、この手法中で用いられるデータモデルについて 3.2 節でまとめる。そして 3.3 節、3.4 節で我々の手法の詳細について述べる。

3.1 概要

図-3 はハイパーテキスト化手法の全体像である。本手法における作業の流れは、ハイパーテキストデータの構造(スキーマ)を設計するデータ定義部作成フローと、そこで設計されたスキーマに基づいて実際のデータを変換するデータ変換フローの二つに大別される。

このハイパーテキスト化手法の特徴は以下の三点である。

1. ハイパーテキストデータモデルを用いたハイパーテキスト構造の表現
2. 抽象辞書を基にした辞書の IS_A 階層を用いた汎用化への対応
3. データ記述の標準化への対応

1 は、データ定義部作成フローのモデル化ステップに対応する。ここでは、Coad らのダイアグラム [2]

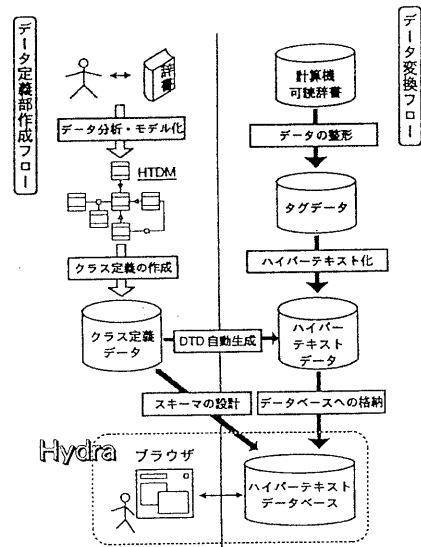


図 3: ハイパーテキスト化手法

を辞書の分析に必要なリンクの分類に基づいて拡張したダイアグラムを用いる。この詳細は 3.2 項で述べる。

2 は、図-4 に示すように、様々な辞書が共通に持っている構造を抽出した抽象辞書を定義し、この抽象辞書を詳細化することにより各辞書の構造を定義することである。これにより、新しい辞書をハイパーテキスト化する際も、その辞書固有の情報だけを抽象辞書に加えていくだけでモデル化が出来る。

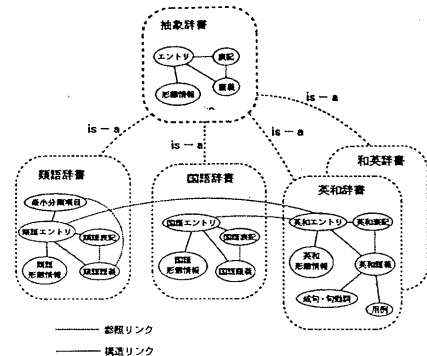


図 4: 抽象辞書

3 は、データ変換フローにおけるデータ記述に、ISO で標準に制定された SGML [1] [6] を用いたことである。

ここで、SGML について簡単に触れておく。SGML を用いて記述された文書は 図-5 に示すように、

- 文書内で用いられるシンタクスと文字セットを定義する SGML 宣言
- 文書の構造を定義する文書型宣言 (DTD: Document Type Definition)
- 文書インスタンス

から構成される。

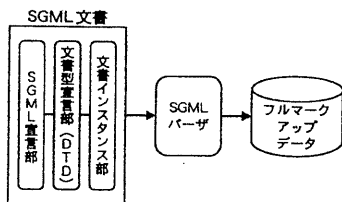


図 5: SGML概要

文書インスタンスは DTD に定義する構造でマークアップ (タグ付け) する。そして SGML パーザを用いてパーズングを行うことにより、DTD に定義する構造に合致しているかどうかを検証する。また、文書インスタンスのタグ付けは省略記法を用いることができ、省略されたタグは SGML パーザがパーズングの際に自動的に復元し、フルマークアップデータを作成する。

3.2 ハイパーテキストデータモデル

本節ではデータ定義部作成フローにおいて用いるデータモデルについて述べる。

3.2.1 リンクの種類

ハイパーテキストデータは、文書の断片であるノードをその間に存在している関係を表すリンクによって結びつけたネットワーク構造を用いて文書表現したものである。このとき、文章には各断片間に様々な

関係が存在しているため、文書の構造を表現するには複数の種類のリンクを必要とする。

これまでもいくつかの文献 [3] [4] [9] でリンクの分類が取り上げられているが、それらの多くはリンクの構造や意味だけでなく、その機能や実装法までを含む分類を行なっている。

例えば、[9] では九種類のリンクを挙げているが、その中にはリンク元のノードよりも詳細な情報を提示する Zoom Links, リンク元のノードよりも高い視点からの情報を提示する Pan links, プログラムを起動する Execute links などがある。

しかし、リンクの実装法は個々のアプリケーションの設計時に決定すべき事項であり、ハイパーテキストデータの概念設計時に決定すべき事項ではない。そこで我々はリンクの構造に基づいて、データの持つハイパーテキスト構造の表現を目的とした分類を行なった。

この分類ではリンクは大きく **Organizational** リンクと **Referential** リンクに分類される。

Organizational 文書の論理構造を表現するリンク
Referential ノード間の意味的關係を表現するリンク

さらに、Organizational リンクは以下のように分類される。

Aggregational 部分-全体関係を表現し、複合ノードを定義するリンク

Grouping 複数のノードの集まりであるノードを定義するリンク

辞書中にあるリンクが表現する関係の例を表 1 に示す。

3.2.2 データダイアグラム

当初我々は、ERダイアグラム [8] [14] を用いて辞書のモデル化を行なっていた。しかし、ERダイアグラムには以下の欠点があった。

1. 複合ノードの明示的な表現ができない
2. is_a 関係が表現しにくい
3. リンクの種類を明確に表現できない

表 1: リンクが表現する関係

リンク名	リンクが表現する関係の例
Aggregational	「辞典の各エントリは、見出しと語義から成る」という関係
Grouping	「類語辞典の上位カテゴリは複数の下位カテゴリに詳細化されている」という関係
Referential	「右手」という単語と「左手」という単語の間にある[反対語]という関係

これに対して、Coadらのダイアグラム^[2]は、Assembly Structure, Classification Structureによって上記の1と2を解決している。そこでこれをハイパーテキストの立場から整理し、3.2.1項で述べたリンクの分類に基づいて拡張した。以下、このダイアグラムに基づいたハイパーテキストデータモデルの表記法について述べる。

3.2.2.1 クラス

クラスは以下の三つの要素によって構成される。

class name そのクラス固有の名前

attributes そのクラスのインスタンスが持つデータ

methods そのクラスに対して指示できる操作

クラスの表記法は、Coadらのオブジェクト表記法^[2]とほぼ同一である(図-6 (a))。ただし、抽象クラスを図-6 (b)、各attributeの多値性を図-6 (c)のようにして表現する拡張を行なった。

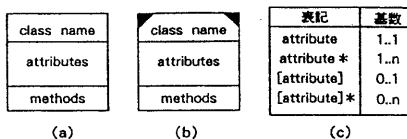


図 6: クラスの表記法

また、クラス間の継承関係を表すis_aリンクは図-7のようにして記述する。これは^[2]の Classification Structure に準じている。

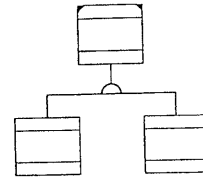


図 7: is_aリンクの表記法

3.2.2.2 リンク

標準的なリンクは以下の三つの要素によって表現される。

link name そのリンク固有の名前

type そのリンクのタイプ

foot 写像基数

type は3.2.1項で述べたリンクの分類に対応する。そして type がリンクの構造に基づいたものであったのに対して、link name にはその役割を反映したものをつける。

リンクの表記の例を図-8 (a)に、type及びfootの表記法を図-8 (b)に示す。attributesやmethodsが付け加えられたリンクは図-8 (c)のようにして記述する。また、embedded linkに関しては、どのattributeの中に埋め込まれているかを示すために、図-8 (d)のようにして記述する。

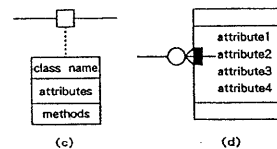
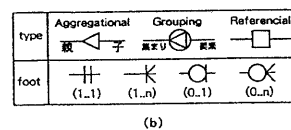
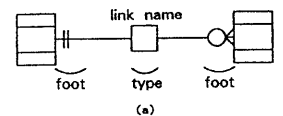


図 8: リンクの表記法

また、リンクの記述に以下の制約を設けている。

1. 一つの親クラスから複数の Grouping リンクが出ることはない
2. 一つの親クラスから Aggregational リンクと Grouping リンクが同時に出ることはない

3.2.3 ダイアグラムの例

国語辞典 [12] のモデル化の結果得られたダイアグラムの例を図-9 に示す。

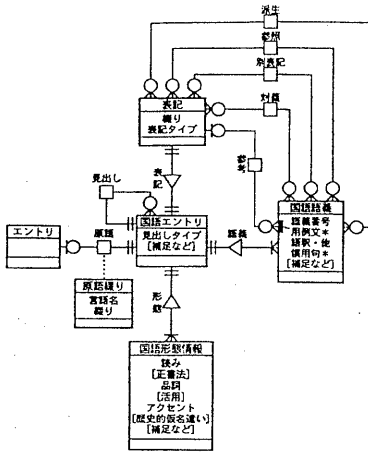


図 9: 国語辞典 [12] のデータモデル

3.3 データ定義部作成フロー

本章では、データ定義部作成フローについて述べる。

3.3.1 データ分析・モデル化

ハイパーテキスト化手法では原データに含まれているハイパーテキスト構造を抽出するために原データの構造分析が必要である。構造分析は BNF を用いて行い、辞書の構文規則を明確にし、そこに含まれる項目を抽出する。辞書項目は構文的に判断できるものを更に意味的な観点から詳細に分類する。この過程で辞書の構文的な曖昧さや意味的な曖昧さも明らかになる。

次に BNF で記述された辞書の構造を元に 3.2 節で述べたダイアグラムを用いてモデル化作業を行ない、ハイパーテキスト構造を決定する。その際、4 節で述べた抽象辞書をノード分割のテンプレートとして使用する。辞書項目の内、他の辞書項目との関係を表すものは参照的なリンクと考えられる。例えば、“類義語”や“対義語”などをリンクとしてダイアグラムで表現する。

3.3.2 クラス定義の作成

3.3.1 項の結果はダイアグラムであり、計算機処理可能なデータに変換する必要がある。そのデータがクラス定義データである。

クラス定義データのシンタックスを図-10 に示す。

```

<defclass> クラス名
  <superclass> スーパークラス名
  <attribute> アトリビュート名, データタイプ,
    デフォルト値, 最大基数, リンク条件
  .....
  <link> リンク名, リンクタイプ, ターゲットクラス,
    最小基数, 最大基数, リンク条件
  .....
</defclass>
  
```

図 10: クラス定義データのシンタックス

データ定義データの作成の際にはデータタイプやデフォルト値を決定することになるが、これは設計者が処理効率や実現性、使用する DBMS の制約条件などを総合的に判断して決定する。

クラス定義データの例を図-11 に示す。

```

<defclass> 国語語義
  <superclass> 語義
  <attribute> 語義番号, int, 1, 1, 1
  <attribute> 語義・他, text, 0, 1
  <attribute> 用例文, etext(類語ref), 1, *
  <attribute> 注記, text, 0, 1
  <link> 参照, R, 表記(綴り), 0, *
  <link> 対義, R, 表記(綴り), 0, *
  <link> 類語ref, R, 類語概念, 0, *
</defclass>
  
```

図 11: クラス定義データの例

3.3.3 ハイパーテキストフォーマットデータの DTD 自動生成

クラス定義データからハイパーテキストフォーマットデータの構造を定義する DTD を作成するためのステップが DTD 作成ステップである。このステップは SGML パーザである MARK-IT²が持つ Application Interface Language³を用いて自動化している。これにより、複雑な DTD を手作業で記述する必要がなくなり、ハイパーテキスト化作業全体が効率化できた。

このステップにおける処理の流れを図-12 に示す。例えば、国語辞書の DTD を作成するためには、まずそのスーパークラスを含む抽象辞書クラス定義データと国語辞書クラス定義データをそれぞれ DTD の記述形式に変換する。ただしこれら二つの DTD はまだ完全な DTD ではなく、これらを結合することによって完全な国語辞書 DTD が完成する。

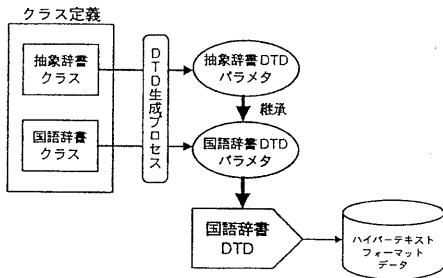


図 12: DTD 生成法

3.3.4 データベーススキーマの作成

クラス定義データに基づいてデータベースのスキーマを設計する。現在用いている DBMS は RDB なのでテーブルを設計することになる。各テーブルを設計する際、設計ルールがあり、これに従うことでクラス定義より一意にテーブルを設計することができる。

- 各クラスに対応して

²SEMA GROUP BELGIUM S.A. が商品化した SGML パーザ

³SGML で記述された文書を任意の形式の文書に変換するコンバータ

1. インスタンス格納用テーブル
2. 多値属性格納用テーブル(インスタンスの属性が多値の場合のみ)

- 各リンクに対応して

3. リンク格納用テーブル

インスタンス格納用テーブルは ID フィールドを持ち、これによりインスタンスの識別が行なわれる。ID は各データベース毎に一意性を保つ。各リンク格納用テーブルはこの ID を用いてインスタンス間の参照を行なう。

リンク格納用テーブルはリンクタイプに対応して次の規則を用いてテーブルを設計する。

1. Organizational リンク用のテーブル作成規則 (複数の親を取り得る場合のみ)
2. Referential リンク用のテーブル作成規則
3. 外部のデータベースに対する Referential リンク用のテーブル作成規則

また、RDB に格納されたデータを問い合わせるコードのプログラミングを容易にするため、DB クラスライブラリが用意されている。DB クラスライブラリは、RDB との SQL インターフェースを隠蔽し、オブジェクトを単位とする DB 問い合わせやリンクに基づいた検索を実現する。

3.4 データ変換フロー

本章ではデータ変換フローにおける各ステップについて述べる。それらのうち、トランスレータとジェネレータについては Application Interface Language を用いて自動化した。

3.4.1 コンバータ

書籍形式の記述をそのまま計算機に入力した計算機可読辞書は多くの曖昧性を含んでおり、そのままではハイパーテキスト化できない。そこで、SGML を用いた曖昧性のない表現 (タグデータ) へと変換する。

このステップでは、まずコンバータによって原データに対してタグ付けを行う。しかし、原データには入力ミスなどによってコンバータが正常にタグ付けできない箇所があるため、人手によるチェックも行う。

#ことば#3 [言葉] (「は」は端の意)
 1) 社会ごとに決まっている、音声による表現(行為)。〔広義では、文字による表現(行為)をも指す〕「意味の分からない」〔=単語〕・推薦の〔=文章〕・一を返す〔=A返答する。B相手の言葉を反駁(ハンバク)する〕・一〔=言い方〕が悪い・話し・書き・いなかー
 2) 々〔詞〕〔地の文に対して〕〔小説・戯曲の〕会話(文)。
 ……

図 13: 国語辞書の原データ([12]より抜粋)

```
<一般の単語>
  <マーク>2
  <見出し>ことば
  <補足>
<ACC>3
<正書法>言葉
<補足的説明>「は」は、端の意
  <語義><小語義>
<小語義番号>1
  <小語義本体>
  <語義文>社会ごとに決まっている、音声による表現(行為)。〔広義では、文字による表現……
  <用例等>意味の分からない～〔=単語〕……
  ……
</一般の単語>
```

図 14: 国語辞書のタグデータ

図-13 は [12] のデータの例であり、図-14 は 図-13 にタグ付けを行った例である。

タグデータの構造は DTD で定義された構造にしたがっており、原データの持っていた曖昧性は排除されている。

3.4.2 トランスレータ

トランスレータの目的は、原データが持っていたノンリニア性をタグデータに埋め込むことである。トランスレータの役割は以下の三つである。

- タグ名の変換
 タグデータとハイパーテキストフォーマットデータの間で対応するデータのタグ名が違う場合、その変換を行う。
- データ構造の変換

タグデータとハイパーテキストフォーマットデータの間でデータの構造が変わる場合、その変換を行う。また、対応するタグがない場合は、新しいタグの作成を行う。

- リンク関係を明示するための ID の付与
 リンクを記述するためにはリンク先やリンク元が明確に指し示すことができなければならない。そのために、ノードの識別子としてIDを付与する。

図-15 は図-14 をトランスレータによって変換したハイパーテキストフォーマットデータである。

```
<国語概念 label=ことば-1>
  <形態><国語形態情報>
  <読み>ことば
  <正書法>言葉
  <アクセント>3
  <語義><国語語義>
  <語義番号>1
  <語釈・他>社会ごとに決まっている、音声による表現(行為)。〔広義では、文字による表現……
  <用例文>意味の分からない～〔=単語〕……
  ……
</国語概念>
```

図 15: ハイパーテキストフォーマットデータ

3.4.3 ジェネレータ

ジェネレータは、SGML で明示的に構造が記述されたハイパーテキストフォーマットデータを読み込み、DBMS に格納する。

現段階では、RDB を使用しているので、ジェネレータはSGMLデータであるハイパーテキストフォーマットデータから、RDB へ格納するための SQL 文を生成する。そしてその SQL 文を評価することによって、ハイパーテキストフォーマットデータをDBMS に格納する。

4 辞書ハイパーテキスト Hydra の概要

以上述べてきた手法に従って、辞書ハイパーテキスト Hydra の作成を行った。仕様は表 2 の通りである。

表 2: Hydra の仕様

マシン	NeXT
言語	Objective-C
ユーザインタフェース	NeXTStep
DBMS	Sybase (RDB)
原データ	新明解国語辞典

Hydra の機能は以下の通りである。

1. 見出し語検索
2. リンクによる関連見出し(対義語, 参照語, 派生語)への移動
3. 埋め込みリンクによる関連見出しへの移動
4. ヒストリ機能

2 は, 辞書の各エントリ中に書いてある対義語, 参照語, 派生語などに, リンクをたどって直接アクセスする機能である。ボタン, 及びメニューをマウスクリックするだけで, あらためて単語を入力することなく対象のエントリに飛ぶことが出来る。

3 は語釈文などの中にある単語からそのエントリに飛ぶためのリンクである。2 のリンクが原データの特定の項目にしか張れないのに対して, このリンクは任意のデータに対して張ることが出来る。

4 は, 見出し語検索やリンクによる検索によって一度検索した単語を自動的に記憶しておき, そこに直接戻ることができるようにする機能である。複数の単語を見比べたりするときに用いる。

図-16 に Hydra の画面例を示す。

5 まとめ

本論文ではハイパーテキスト化技術の大規模データへの適用手法の研究として, 計算機可読辞書のハイパーテキスト化手法について述べた。そしてその手法を適用して構築した辞書ハイパーテキスト Hydra の概要について述べた。

本論文で述べた手法の特徴は以下の通りである。

- ハイパーテキストデータモデルによるハイパーテキスト構造の表現
- 抽象辞書を基にした辞書の IS_A 階層による汎用化への対応
- SGML に基づくデータ記述の標準化への対応

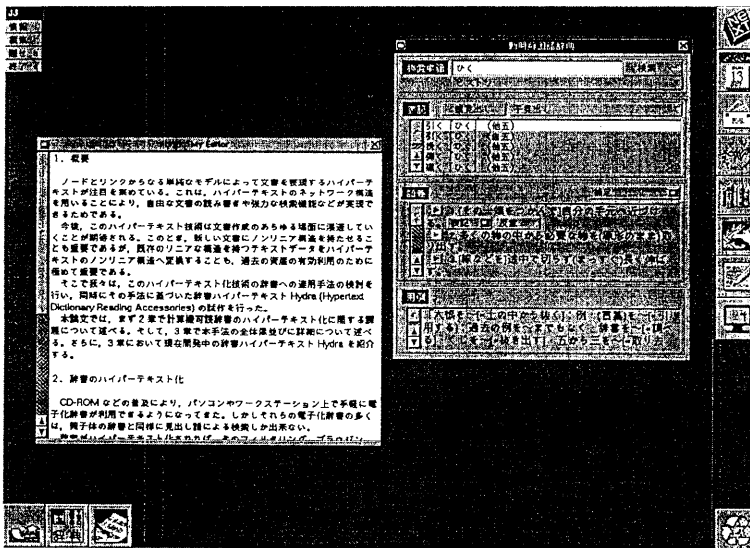


図 16: Hydra の画面例

また今後検討すべき点としては以下が挙げられる。

- 他の辞書への適用実験に基づく手法の有効性の検証
- 複数の種類の辞書を統合した辞書ハイパーテキストシステムの構築
- 辞書以外の一般文書への対応

謝辞

本研究を進めるに当たりご指導いただいた知能工学研究部の田村部長, ならびに辞書データの分析についてご協力いただいた知能工学13研究室の皆さんに感謝します。

参考文献

- [1] Bryan, M., SGML an Author's Guide to the Standard Generalized Markup Language Addison-Wesley publishing, 1988.
- [2] Coad, P., Yourdon, E., Object-Oriented Analysis, Prentice-Hall, 1990.
- [3] Conklin, J., Begeman, M. L., gIBIS: A Hypertext Tool for Exploratory Policy Discussion ACM Transactions on Office Information Systems, Vol. 6, No. 4, Oct 1988, pp. 303-331.
- [4] DeRose, S. J., Expanding the Notion of Links, Proceedings of Hypertext '89, ACM press, 1989, pp. 249-257.
- [5] Furuta, R., Plaisant, C., Shneiderman, B., Automatically transforming regularly structured linear documents into Hypertext, Electronic Publishing, Vol. 2(4), 1989, pp. 211-229.
- [6] ISO 8879, Information processing - Text and office systems - Standard Generalized Markup Language (SGML), 1986.
- [7] Martin, J., HYPERDOCUMENTS & How to Create Them, Prentice-Hall, 1990.
- [8] Martin, J., McClure, C., ソフトウェア構造化技法, 近代科学社, 1988.
- [9] Parsaye, K. et al., INTELLIGENT DATABASES, John Wiley & Sons, Inc., 1989.
- [10] Raymond, D. R., Tompa, F. W., Hypertext and the Oxford English Dictionary, Communications of the ACM, Vol. 31, No. 7, pp. 871-879, 1988.
- [11] 大野 晋, 浜西 正人, 類語国語辞典, 角川書店, 1985.
- [12] 金田一京助, 新明解国語辞典, 三省堂, 1989.
- [13] 黒橋 禎夫 他, 専門用語辞典のハイパーテキストシステム, 情報メディア, 1-4, 1991.
- [14] 酒井 博敬, 情報資源管理の技法, オーム社, 1987.
- [15] 島 健一, 通信文書体系化システム - COSMOS, 信学技法データ工学, DE89-35, 1989.
- [16] 鶴丸 弘昭, 内田 彰, 国語辞典からの情報抽出と構造化について, 長崎大学工学部研究報告, 15(24):41-48, 1985.
- [17] 土井 美和子 他, プレーンテキスト/ハイパーテキスト間の変換, 情処研報, Vol. 89, No. 47, 13-5, 1989.
- [18] 内藤広志 他, 辞書ハイパーテキスト **Hydra(1)** — データベース構築手法 —, 情報処理学会第42回全国大会, 1991.
- [19] 根本治朗 他, 辞書ハイパーテキスト **Hydra(2)** — データモデル —, 情報処理学会第42回全国大会, 1991.