汎用 EDR 電子化辞書検索システムの実現

片山研一　本田岳夫　太田千晶　藤原滋　徳田昌晃　奥村学*
北陸先端科学技術大学院大学 情報科学研究科
{k-kataya, honda, chiaki, shigeru, tokuda, oku}@jaist.ac.jp

1　はじめに

EDR 電子化辞書 (以後 EDR 辞書) は, 日本電子化辞書研究所 (EDR, Japan Electronic Dictionary Research Institute, Ltd.) [1]が作成した語彙に関する知識を集めた知識ベースである. この知識ベースは, 単語の文法的属性, 単語が表す概念, 概念を理解させるための基本的な知識などを大規模に体系的に蓄積したものである [EDR92]. これは, 単語辞書, 概念辞書, 対訳辞書, 日本 (英) 語共起辞書, 日本 (英) 語コーパス, 専門用語辞書から構成されている.

これらの辞書から, 知識を取り出すツールとして, EDR 自身が開発した EDBroW[EDRtool], 九大松尾研開発の Seep というデータベース管理システムでデータベース化したものがある [SEEP]. これらは, Windows 専用である, 専用のデータベース上でないと利用できないなど汎用性に欠けているといえる.

そこで我々は, 多くの OS で多目的に辞書が利用可能になるよう, クライアントサーバシステムを構築し, ソケット通信で簡単に辞書が索けるようにプロトコルの仕様を決めた.

本システムは、単語辞書、概念辞書、共起辞書、対訳辞書の各辞書に含まれる単語見出し、概念識別子、概念説明などをキーにして該当部分のレコード内容を引き出すことができる。

本稿では, システムの構成を簡単に説明し, クライアントの例として, Tcl/Tk, cgi, perl を利用したツールを紹介する.

[0]望月源, 近藤恵子, 鈴木勝仁, 川口恭伸との共同研究である.
[1]http://www.iij.or.jp/edr/J_index.html

2　システム概要

本システムは, クライアント-サーバ間の単純なプロトコルを規定することにより, 多くの OS 上かつ, 多種の言語で辞書検索ツールを実装することを可能にした[2]. サーバが行なっていることは, 検索プロトコルを受け取り, 該当する辞書のレコード内容を返しているだけである.

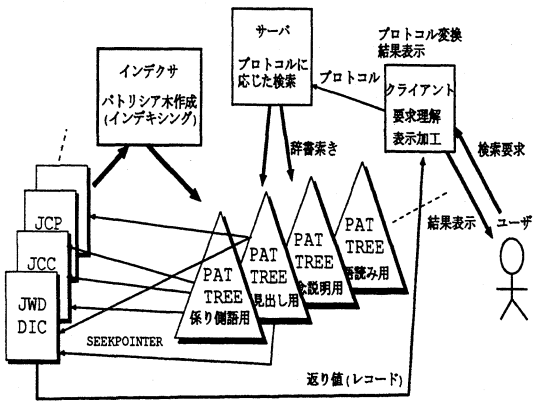

図 1: システム概要

本システムでは, 図 1に示すように, まず, インデクサで, EDR 辞書からインデックスを作成しておく. 実際に検索を行なう場合, 以下の流れになる.

1. ユーザが, クライアントツールに対し, 検索要求を出す.
2. クライアントツールは, これを解釈して, サーバとの決められたプロトコルに変換する.
3. サーバは, これを受け取り, 必要に応じて KEY 毎にインデックスを索きにいく.

[2]現在サーバは, UNIX 上で実行させることを前提にしているが, Perl5 と C(gcc) で実装されており, これらの言語とソケットさえ使えれば簡単に他の OS 上に移植可能である.

4. インデックスより, その KEY の KEYWORD を含む辞書のレコードに対するシークポインタが得られる.
5. このシークポインタを使って, 辞書からレコードを取り出す.
6. サーバは, 要求を満足するレコードのみクライアントシステムに返す.
7. クライアントツールは, そのレコードそのものもしくは, 加工してユーザに提示する.

本システムで用いるプロトコルは,

<COMMAND> <DIC> <KEY> <KEYWORD>[3]

の構造をしている. 次に, このプロトコルを使用した検索例を示す.

例1 ENTRY JWD WORD 明 READ めい AND

この例は, 日本語単語辞書で表記 "明", 読み "めい" であるレコードを索く命令である. このように, KEY と KEYWORD の組みを AND や, OR で, 逆ポーランド式に結びつけることによって複合検索ができる. 演算子としては, 他に差分リストを取り出す SUB が使える.

例2 EXTEND CON CPH 物語 絵本 会話

この例は, 概念見出し辞書中の概念説明で "物語", "絵本", "会話" を最も多く含むものを取り出しなさいという命令である. COMMAND "EXTEND" は, 拡張命令で "CON" 以外にも有用であると思われる拡張命令 "UCPH", "ID_PATH" などを用意している.[4] これらの拡張命令は, ENTRY COMMAND だけを使用してもクライアント側で制御してやれば取れて来れるが, 検索部分はなるべくサーバ側に持たせ, クライアント側には, ユーザインターフェースのみを実装するようにしたいと考えている. よって, これらのように要望の多い索き方は, 新たにプロトコルを追加していき, サーバ側でやらせる予定である.

[3]COMMAND:コマンド名, DIC: 辞書名, KEY: 検索キー (表1に挙げてある構成要素), KEYWORD: キーワード

[4]この他に以下のような拡張コマンドを備えている
EXTEND UCPH concept_identifier…… concept_identifier の上位概念の CPH を索く
EXTEND ID_PATH HIGH|LOW concept_identifier…… concept_identifier の最 (上 | 下) 位までのパスを求める

このようにコマンドとキーを組み合わせることにより多種多様な検索を可能にする.

3 インデックスの作成

前節では, 検索要求を与える手法を示した. 本節では, サーバが実際に検索要求物を取り出すのに必要となるインデックスの作成手順を示す.

EDR 辞書には, 単語辞書, 概念辞書, 共起辞書, 対訳辞書など多くの辞書及びコーパスが含まれているが, 本システムでは, このうち全日本語 (単語, 概念, 共起) 辞書を利用できるようになっている[5]. これらの辞書はさらに, 表1のように細分類され, 各辞書の構成要素のうち, 必要と思われるものだけのインデックスを作成した.

インデックス形式だが, 開発当初本システムは, ハッシュ方式の DBM インデックスを作成したが, 前 (後) 一致に適していないという欠点があった. そこで, パトリシア木用のインデックス方式に変更することにより, 前 (後) 方一致[6]も高速に検索が可能 [PAT96] になった. このパトリシア木形成, 及び検索には, 奈良先端大松本研の開発した形態素解析システム 『茶筌』 [ChaSen] [7]のソースを改変したものを使用した.

本システムでは, 表1のように多くの構成要素から辞書索きを行なえるようになっている. さらに, "かな表記" は, ひらがな, カタカナのどちらでも索けるようにした. また, システムの最大の特徴として, 概念説明中に出現する自立語から辞書を索くことができるということが挙げられる. このために, まず概念説明を既存の日本語形態素解析ツール[8]により文を形態素に分ける. この形態素集合の中から自立語だけを取り出す. これを元にインデックス作成を行なった. 概念説明中の語から索けるようにすることで, 言い換え, 類似語検索にも使えるようになり活用の幅が広がったといえる. サーバは, 検索要求中の KEYWORD が, 現れる度にこのインデックスを索きにいっている.

[5]本システムは, クライアント-サーバ間のプロトコルを規定しているだけなので, インデックスさえ作れば, 他の EDR 辞書だけでなく, EDR 辞書以外の辞書も簡単に利用できる.

[6]後方一致は逆さ読みをキーとすることで可能にした

[7]近々, パトリシア木検索ツールが, 奈良先端大松本研から発表される予定である.

[8]ここでは, 形態素解析ツールの種類は問わない

表 1: 現在利用可能な辞書

辞書名	インデックスを作成した構成要素
日本語単語辞書 (JWD)	単語見出し, かな表記, 不変化部, 概念識別子, 概念見出し, 概念説明中に出現する見出し語
概念見出し辞書 (CPH)	概念見出し, 概念識別子, 概念説明中に出現する見出し語
概念体系辞書 (CPC)	上 (下) 位概念識別子
概念記述辞書 (CPT)	係り (受け) 側概念識別子, 概念関係子
日本語共起辞書 (JCC)	各共起要素の単語表記と読み表記, 係り (受け) 側各々の表記及び読み
動詞共起パターン副辞書 (JCP)	動詞表記, 動詞の概念識別子, 動詞表記に対して各表層 (深層) 格をとる概念識別子

パトリシア木による検索の高速性と柔軟性を利用することにより本システムは, より多様な検索が可能になった.

4 クライアント例

本研究の有用性を示すために以下の 3 つのクライアントを用意した.

1. コマンドライン入力による検索 (図 2)
2. Web を利用した検索 (図 3)
3. Tcl/Tk を利用した検索 (図 4)

1 は, Perl で書かれた簡単なスクリプトであるが, プロコルを直接引数として取るため, サーバの持つすべての検索機能を使用する事ができる. 2 は, Web 上で実装する事により Web クライアントを持つどのマシンからでも索けるようにしたものである. 3 は, Tcl/Tk によるグラフィカルユーザインターフェースの検索ツールで, ユーザが入力したもの (見出し語, 読み, 概念 ID) を自動的に判別して各辞書を索きに行くようにしている. さらに索けて来たものから, カット&ペーストで再検索したり, クリックによる概念階層を調べて行くなど, 優れたユーザインターフェース能力を備えていると言える. このように, 1 の多機能柔軟性, 2 の環境を選ばない検索, 3 の利便性を状況に応じて使い分けるとよいであろう.

```
tiffa% jdclient ENTRY JWD READ はし
JWD0513153 階 [ハシ] 階 (JLN1,JRN1)
    ハシ ハシ JN1 "" "" "" "" ""
     3cea4b "" 梯子 [ハシゴ]
     "a tool for climbing up"
    高いところに登るための道具
    "" 0/0 DATE="89/2/17"
JWD0003955 端 [ハシ] 端 (JLN1,JRN1)
    ハシ ハシ JN1 "" "" "" "" ""
    0e2f98 foothold 手がかり [テガカリ]
     "a cue to start something"
    物事を始めるときの糸口 ""
    0/74 DATE="94/6/25"
                .....
```

図 2: コマンドライン入力

5 おわりに

本システム[9] で, 検索部分をサーバにすべてまかせ, カプセル化することにより単純なプロトコルのやりとりだけで, 巨大な EDR 辞書が索けるようになった. これにより, ユーザインタフェースツールを作る際, 実際に検索する部分を無視できるので, 簡単にクライアントを作成することが出来た[10]. また, 概念説明中の自立語から辞書が索けるようになることで, より多方面

[9]本研究は, 現在 SunOS4.1.4, WindowsNT, FreeBSD 上で実験を進めている.

[10]Jdclient は, 30 行程度のプログラムであり, Web 版もプロトタイプは, 50 行程度であった

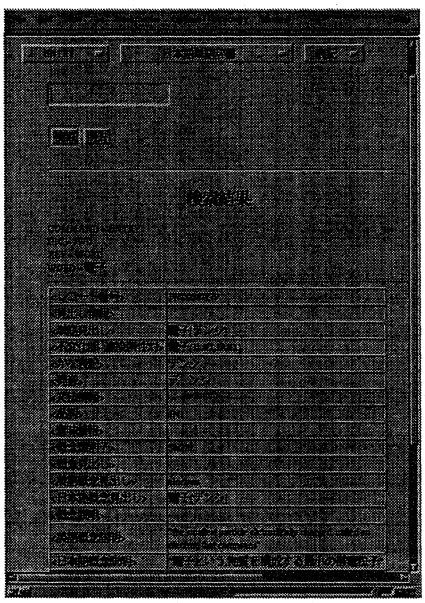

図 3: Web による辞書索き

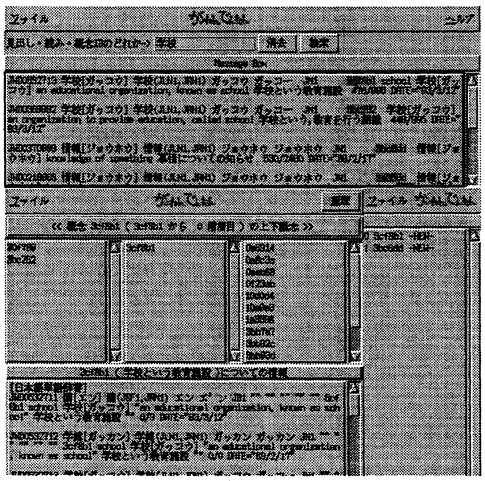

図 4: Tcl/Tk によるグラフィカルなユーザインターフェース

から EDR 辞書の利用が可能になった. さらに, and, or, 前 (後) 方一致など多くの検索法をサポートすることで柔軟で頑健な検索が可能になった. 本システムは, 近々公開を考えており, より使いやすい仕様を現在考えている. また, すべての EDR 辞書のインデックス作成, JAVA, Windows の Visual 言語, emacs lisp などでの実装なども行なっていきたいと考えている.

謝辞

本システムで使用したパトリシア・ツリーの作成・検索に関しては、茶筌パッケージに含まれる辞書インデクス作成・検索プログラムを使用させて頂きました。このパッケージの使用を許可して頂きました奈良先端大学院大学 松本研究室に深く感謝いたします。

参考文献

[EDR92] 横井俊夫, 「語彙知識の自己組織化に向けて」-EDR 電子化辞書の構造と開発手法-, 情報処理学会情報学基礎研究会 92-FI-26,P.1-8

[EDR95] (株) 日本電子化辞書研究所, EDR 電子化辞書仕様説明書 (第 2 版), 1995

[EDRtool] 辞書利用支援ツール http://www.iijnet.or.jp/edr/Tool.html

[SEEP] 小出東洋, 松山尚市, 竹田正幸, 松尾文碩, EDR 電子化辞書の検索システム, 情報処理学会第 52 回全国大会 (平成 8 年前期), 3-21(2B-2)

[PAT96] 山下達雄, ChaSen Technical Report CTR-1 パトリシア木を用いた形態素解析のための辞書検索, 第 1 版, ChaSen1.0b5 付属, 1996

[ChaSen] 松本裕治, 今一修, 山下達雄, 北内啓, 今村友明, 日本語形態素解析システム『茶筌』 version 1.0b5 使用説明書, 1996, http://cactus.aist-nara.ac.jp/lab/nlt/chasen.html