高速全文検索FTSのシソーラス機能

5V－5

堀川 恵美　新谷 義弘　長坂 篤

沖電気工業（株）マルチメディア研究所

1　はじめに

大量に電子化された文書の中から容易に情報を検索できる全文検索システムへのニーズが高まっており、我々はメモリ効率が高く、検索速度の大きい全文検索システム (FTS) を開発した[4]。

本論文では、同義語などを格納するためのシソーラスデータベースの構成と、シソーラスを用いた検索時の検索方式について述べる。

2　シソーラスデータベース

2.1　概要

ある文字列がシソーラス定義語としてシソーラスデータベースに登録されているかどうかを参照する場合、その文字列とシソーラス定義語との文字列マッチングをとって、登録語を検索する方法が一般的である。ここで、いかに速く適当な文字列マッチングをとれるかが、シソーラス検索の速度に大きく影響してくる。我々はとくに日本語に適したマッチング方法を提案する。

日本語文書に使用される文字はＪＩＳコードのみでも、約１万語ほどもあり、日本語文字は、これらの多くの文字との組合せから構成される。そこで、あまり長くない文字列であっても、確率的に見ると、膨大な組合せのうちの一部の出現であるといえる。ゆえに、この文字の組合せのパターンを手がかりとすると、データの中から求める部分を特定、あるいは絞り込むことが容易であると考えられる。

このように文字列マッチングに文字列の組合せという考えを導入する時、その組合せの確率分布が偏らず、一様であることが望ましい。

以上により、我々は絞り込みを速くする方法として、２つの文字間の関係に着目した。全角文字“モ”“ジ”との組合せを例にとる。それぞれの文字コードは以下である。

モ: A5E2
ジ: A5B8

２バイトで考えた時、A5E2、A5B8 と見るより、それぞれ下位バイト、上位バイトをつなぎ合わせた E2B8,A5A5 の方が速く該当要素を絞り込むことができる。文字の使用頻度を調べると漢字よりもひらがな・カタカナが圧倒的に多い。“モ”に続く組合せは無数にあるが、下位バイト E2B8 を先に検査することで、１回の検査でこの２文字はほぼ確定する。こうして組合せの分布を分散させ、しかも、候補を最初の段階で絞り込むことが出来る。これを利用し、文字列を４文字ずつの下位バイト集合、上位バイト集合を作成し、検索を進めていく方法をとった。

A thesaurus facility of a fast Full-Text-Search system FTS
Emi Horikawa,Yoshihiro Shintani and Atsushi Nagasaka
Oki Electric Industry Co., Ltd. Media Laboratories

2.2　構造

このシソーラスデータベースは図１に示される木構造で実現されている。

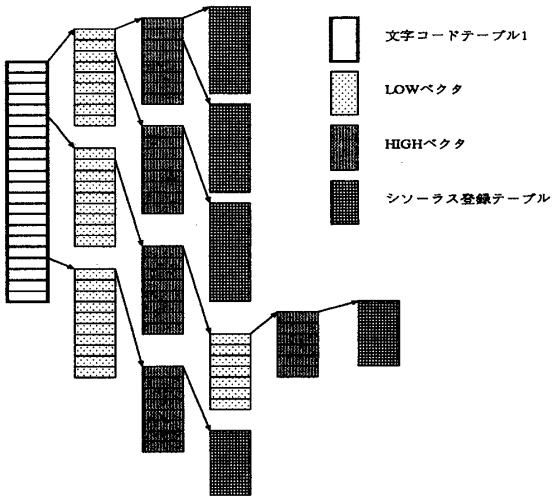

図 1: 木構造

- 文字コードテーブル１
  １次元配列で、各文字の文字コードをインデックスとして参照される。シソーラス定義語の１文字目についての情報を得るための配列。使用されている場合はLOW ベクタへの情報が入っており、未使用の場合は０が入っている。
- ＬＯＷベクタ
  シソーラス定義語の $4n+2, 4n+3, 4n+4, 4n+5$ 番目の文字の下位１ byte の順列からなる集合。
- ＨＩＧＨベクタ
  シソーラス定義語の $4n+2, 4n+3, 4n+4, 4n+5$ 番目の文字の上位１ byte の順列からなる集合。
- 真のシソーラスデータベース
  シソーラス定義語に対する登録語が入っている。

2.2.1　ＬＯＷベクタ

本構造では２～５文字目についての最初の LOW ベクタの要素の数は多くなると思われる。この中より該当要素を速く見つけるため、図２に示す構成をとっている。テーブルＡは登録された２文字目下位１バイトからなる。ベクタＢは登録された２文字目下位１バイトに対する３～５文字目の列である。

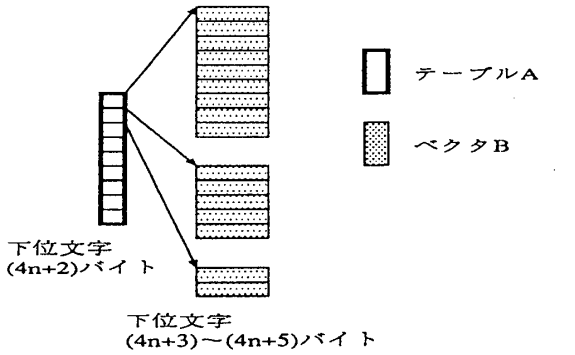

図 2: LOW ベクタ

テーブルAで絞り込むことによって、該当要素を速く見つけ出すことができる。

3 検索方式

シソーラスによる辞書展開、カタカナ・ひらがな、大文字・小文字、全角・半角など1つの検索文字列に対して複数の検索を行なう必要がある。また、その検索回数は、検索文字列が長くなるほど指数的に増大する傾向にある。例えば、「お母さん」という検索文字列がある場合、カタカナ・ひらがなの展開だけで、「オ母さん」、「お母サん」など8通りの文字列ができる。それら展開した全ての文字列を順に検索するのでは、文字列生成に複雑なプログラムを要し、速度的にも遅くなる。

本システムにおいては、以下のように検索を行ない上述のような速度的な問題や文字列生成の問題を最小限に抑えている。

本システムは、任意の文字列で検索できる。従って、意味をなさない文字列であっても、指定されたパターンの文字列があれば検索できることになる。

次に、検索は、文字列を前から順に区切って検索を行なう方式をとっている。前から順に文字列を調べ展開できる文字あるいは文字列がくれば、その時点での文字列で検索し、さらに、展開して検索する。2番目以降の検索は、前の検索結果と該当文字位置分だけずれている結果のものだけを候補に残す。この方式だと、文字列生成が非常に簡単で、検索回数は、最大でも検索文字列中の各文字の候補の総和となる。また、辞書展開において必ずしも形態素解析が必要ではない。

また、検索文字列は、

(1) 前文字列 — 展開できない固定文字列
(2) 展開文字列 — 展開する文字列
(3) 後ろ文字列 — 展開できない固定文字列

の3部構成とし、検索文字列をできるだけ長い文字列にする。これは、検索文字列が検索木の途中で終ると、それ以降の候補を全て取り出す必要があるが、検索文字列が短いと、その分候補が多くなり、時間がかかるためである。通常、1文字の場合が最長の時間がかかることになる。

上述の例の場合の検索は、

(1) 「お母」および「オ母」で検索し、結果を全て候補とする
(2) 「母さ」および「母サ」で検索し、(1) の検索結果と文字位置が1ずれているものを候補とする
(3) 「ん」および「ン」で検索し、(2) の検索結果と文字位置が3ずれているものを候補とする

となる。

4 評価

本検索方式を、単純展開方式（全て展開した後それぞれの文字列について検索する）と比較したのが図3である。尚、計測マシンは HP9000/735(メモリ 64M バイト) である。

この評価が示す通り、本方式によるシソーラス機能は、展開対象が多くなると効果を発揮し約2倍の速さで検索できる。

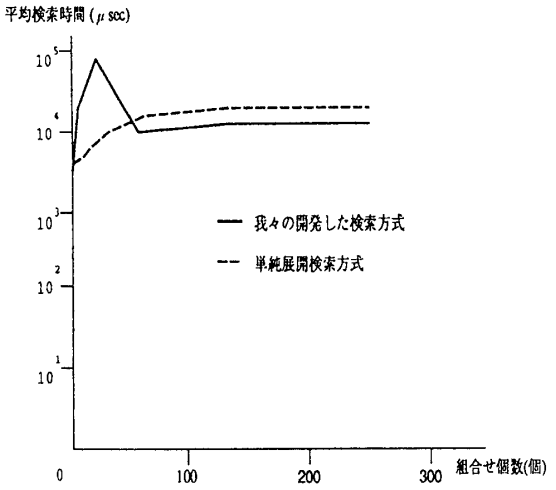

図 3: 単純展開検索との速度比較

5 まとめ

全文検索システム FTS のシソーラス辞書の構造とシソーラスの検索方式について報告した。

今後の課題として、実用化に向けさらに以下のような性能向上を行なっていく。

- メモリサイズの圧縮
- 検索速度の高速化

参考文献

[1] Williams B.Frakes, Richard Baexa-Yates ed.: "Information Retrieval - Data Structures & Algorithmhs -", Prentice Hall,1992
[2] Robert Sedgewick: "ALGORITHMS, 2nd ed.", Addison-Wesley Publishing Company,Inc.,1988
[3] 伊藤哲郎: "情報検索", ソフトウェア講座 19, 昭晃堂,1985
[4] 堀川他: "高速全文検索の一手法", 情報処理学会 第 48 回全国大会論文集 4E-2, 1994.3