データベースシステム 90－6
（1992. 9. 11）

短単位キーワードに基づくテキストデータベースシステム

小川 泰嗣 別所 礼子 岩崎 雅二郎 西村 美苗 広瀬雅子

（株）リコー 研究開発本部 中央研究所

大量テキストを効率的に管理するテキストデータベース管理システムを研究開発中である。本システムはキーワードを用いた管理方式を採用しているが、特に部分一致検索の問題を複合語の構成単語を処理単位とする短単位キーワードによって解決している。本システムでは、テキスト登録時には自動的にキーワードを抽出、検索時には検索語に部分一致するテキストを一致度に応じてランキングする。本稿では、(1) キーワード素性を用いたキーワード抽出法、(2) 検索語重要度に基づくテキストランキング法、(3) 文字成分表を用いたプリサーチ法の 3 つの部分について詳細に説明する。さらに、これらアルゴリズムを実現するライブラリとその評価結果についても報告する。

A Text Database System based on Simple Words

Yasushi Ogawa, Ayako Bessho, Masajirou Iwasaki,
Mina Nishimura and Masako Hirose

Research & Development Center, RICOH Co., Ltd.

We have been developing a text database management system. The system adopts a keyword-based management method, which deal a keyword as a sequence of simple words to retrieve patially matching texts. The system, when text registration, automatically assigns keywords to a text, and, when text retrieve, ranks retrieved texts in order of coincidence (relevance) between query and keywords assigned to each text. We'll describe (1) a keyword assignment method using keyword features, (2) a text ranking method based on importance of each word in a query, (3) a pre-search method using the character-base signature method. A text database library and its evaluation results will also be presented.

1 はじめに

従来の DBMS は定型データの管理には広く用いられているが、テキスト・イメージなど非定型データのための機能は不十分である。一方、オフィスには大量のテキストが氾濫し、その管理が問題となっている。そこで、われわれは非定型データの一つであるテキストを対象とするテキストデータベース管理システムを研究開発している。

テキストの管理方法には、索引（キーワード）を用いる方法 [9, 12, 13] と索引は用いない全文検索法がある [1, 10, 11]。後者には（現在のハードウェア技術では）専用ハードウェアが必要である、文字列の比較のみでは十分な検索機能を実現しにくいなどの問題点がある。そこで、本システムは前者、そのなかでもキーワードを統制しない自由語方式を採用した。

キーワードに基づくシステムでは、検索要求はユーザが入力した検索語と登録テキストの索引語の一致から検索結果が生成される。そのため、検索語と索引語が部分一致しても検索できないシステムが多い。同義語辞書などによって部分一致検索を実現することも可能であるが、日本語には複合語が多いため全てを辞書に登録することは非現実的である。これに対し、複合語をまとまりとしてではなく構成単語ごとに扱うこと[1]で、部分一致検索を簡単に実現できる [8]。

本システムも短単位キーワード方式を採用した。キーワードに基づくテキスト管理では、登録時の索引作成（キーワード抽出）を自動化する必要がある。本システムでは、品詞より詳細な分類である「キーワード素性」を導入し、自由語キーワードを高精度に抽出する。検索時に検索語と索引語が部分一致した場合、一致度に応じて検索結果をランキングすることが求められる。本システムでは、検索語の構成単語にキーワード素性などを用いて重要度を付与し、それに基づいてランキングを行なう。その際、検索語の構成単語が異なる索引語と部分一致すること [8] のないように、抽出キーワードを単語列として保存する。そこで、通常のようにキーワードの転置ファイルを用いずに、ランキングの対象テキストを絞り込む文字成分表によるプリサーチを導入した。

以下、2 章でキーワード抽出法、3 章でテキストランキング法、4 章で文字成分表によるプリサーチ法について説明する。5 章では、これらアルゴリズムを採用して構成したテキスト管理ライブラリについて解説し、評価結果を 6 章に示す。

[1]複合語キーワードを構成単語の連鎖（単語列）として扱うものを短単位キーワードと呼ぶ。

2 キーワード抽出法

2.1 概要

従来のキーワード抽出法には、抽出するキーワードをあらかじめ限定する統制語方式と限定しない自由語方式がある [9]。統制語方式では、統制語辞書を作成しておき、登録テキストに統制語辞書に含まれる語が存在するか否かを照合すれば良い。この方式は抽出処理は簡単であるが、統制語辞書の作成・維持が問題である。自由語方式では、形態素解析などにより登録テキストを単語単位に分割した後、ルールにより必要な単語を選択しキーワードとする。この方式は処理は複雑になるが、ルールの管理は簡単にできる。

本システムはテキストデータベースとしての柔軟性の高さとユーザが検索時に負担を感じることがない点から自由語方式を採用した。本システムにおける、キーワード抽出法は以下の通りである。

1. 形態素解析：
   入力テキストを形態素解析し、形態素（単語）に分割する [7]。
2. キーワード候補抽出：
   キーワードとなり得る単語を品詞により抽出する。
3. キーワード選択：
   キーワード候補の単語列から重要な部分のみを選択し、キーワードと判断する。

2.2 キーワード候補抽出

つぎの品詞の単語をキーワード候補として抽出する。

- 一般名詞・固有名詞
- その他名詞（サ変名詞、和語名詞、修飾名詞、相対名詞、複合名詞、転成名詞）
- 接頭辞・接尾辞
- 助詞（格助詞「の」）
- 数詞・助数詞
- 未登録語

2.3 キーワード選択

抽出ルールによってキーワード候補とされた単語列から最終的にキーワードとする部分を選択する。これは、同一品詞でも単語によってキーワードになりやすいか否かは異なるため、品詞だけではキーワードか否か正確に判断ができないからである。従来はこの問題を補うため不要語辞書を用意し、この辞書に含まれる単語を排除し、残った単語列をキーワードとするのが一般的であった[12]。しかし、不要語辞書の作成・維持が困難である、アプリケーションごとに不要語辞書を用意しなければならないなどの問題がある。

これに対し、本システムでは品詞より詳細な単語ごとの性質を記述するためにキーワード素性を導入した。素性により、アプリケーションに合わせたキーワード抽出を辞書（大量のデータ）を変更することなく、選択ルール（少量）の変更で対応可能となる。さらに、素性と品詞が独立しているため、辞書の維持が簡単になるという利点もある。なお、このようにキーワードとして選択された単語列は、つぎの章で説明するテキストランキング法に合わせて語順を保持したままキーワードファイルに保存される。

2.3.1 キーワード素性

前述したようにキーワード素性はキーワードの判定に用いる性質を表すもので、以下のものがある。

1. 複合語語基：複合語の末尾となりやすい名詞類
   cf. システム、装置、…
2. 固有名詞構成語：固有名詞の後続語になりやすい名詞類
   cf. 大学、駅、…
3. 修飾性接頭語：後続の語を修飾する働きの強い接頭辞
   cf. 高、大、…
4. 情報処理分野助数：情報処理分野に固有な助数
   cf. メガ、ドット、…
5. 地名識別語：地名の中でも識別性の低い固有名詞
   cf. 東京、横浜、…
6. 元号識別語：元号の一般名詞
   cf. 明治、大正、昭和

2.3.2 選択ルール

キーワード素性を導入しても、単語ごとの判断では正確な判定ができない。そこで、単語の連鎖関係も用いた選択ルールにより精度向上をはかる。主なルールはつぎの通り。

1. キーワード素性なしの一般名詞・固有名詞、未登録語は単独でもキーワードとする。
2. キーワード素性付きの一般名詞・固有名詞、その他名詞は後続語があればキーワードとする。
3. 数詞の連続＋素性なしの助数はキーワードとしない。
4. 数詞の連続＋素性付きの助数はキーワードとする。
5. 数詞の連続はキーワードとする。

2.4 例

対象テキストを「リコーの中央研究所は超音波センサーを使った形状識別装置を９月に開発した。」とする。形態素解析の結果は表１の通りである。また、結果欄で ○はキーワード候補、◎はキーワードとして選択された単語を示している。結局、抽出キーワードは「リコー／中央／研究／所」「超／音波／センサー」「形状／識別／装置」である。

3 テキストランキング法

3.1 概要

ユーザは検索対象を検索語として入力する。テキスト検索では、検索結果を検索要求と登録テキストの照合を２値で行なうことは困難であり、検索結果において各テキストの評価に基づいてランキングするのが一般的である[5]。ランキングの方法としては、単語の出現頻度に基づくベクトル空間法[5]、ファジィ理論に基づく方法[3, 4]などが知られている。

本システムでは、検索語の構成単語にキーワード素性などを用いて重要度を付与し、それに基づいてランキングを行なう。この方式では、テキストの一致度は検索語に対する重要度から計算されるため、抽出キーワードの頻度・重要度などの情報をキーワードファイルに保存する必要がない。テキストランキングは以下のように行なわれる。

表 1: キーワード抽出の例

単語	品詞	キーワード素性	結果
リコー	固有名詞		◎
の	格助詞		○
中央	名詞相対		◎
研究	名詞サ変	複合語語基	◎
所	接尾辞		◎
は	副助詞		
超	接頭辞		◎
音波	一般名詞		◎
センサー	一般名詞		◎
を	格助詞		
使っ	動詞 5		
た	助動詞		
形状	一般名詞		◎
識別	サ変名詞		◎
装置	一般名詞	複合語語基	◎
を	副助詞		
9	数詞		○
月	接尾辞		○
に	格助詞		
開発	名詞サ変	複合語語基	○
し	助動詞		
た	助動詞		

1. キーワード抽出：
   前節の方法にしたがい検索語から、実際に検索に用いられる単語列が選択される。

2. 重要度付与：
   重要度とは、検索語の形態素解析した結果得られる各単語に付与される各単語の重要性を表す値。後述するルールに従って各単語ごとに計算する。

3. キーワード一致度計算：
   キーワード一致度とは、索引語と検索語の一致の程度を表す値。各単語の重要度から、各テキストに付与されている索引語ごとに計算される。

4. テキスト一致度計算：
   テキスト一致度とは、登録テキストと検索語の一致の程度を表す値。登録テキストの各索引語と検索語の一致度から、テキストごとに計算される。

以下、2 ～ 4 の処理を詳細に説明する。

3.2 重要度付与

検索語は複数の単語から構成される複合語であることが多い。本システムでは、複合語は形態素解析により単語に分割されるため、部分一致検索は可能である。しかし、構成単語の重要性は複合語の品詞・位置などによって異なる。そこで、検索語の構成単語ごとにつぎのルールにしたがって重要度を付与する。

1. 検索語において最も語尾に近い品詞群 1 の単語の重要度は基本点。
2. それ以外の品詞群 1 の単語の重要度は、その位置より最も近い後方にある品詞群 1 の重要度に増加分を加えた値。
3. 「修飾性接頭語」付きの接頭辞の重要度は基本点。
4. 「修飾性接頭語」なしの接頭辞の重要度は 0。
5. 品詞群 2 の単語の重要度は、(1) 品詞群 1 の重要度の合計 (2) 接頭修飾付の接頭語の重要度 (3) その位置より後方にある品詞群 2 の重要度の合計の 3 つの合計に増加分を加えた値。
6. 上述以外の単語の重要度は 0。

ここで、修飾性接頭語はキーワード素性の 1 つである。また、品詞群 1 とはつぎの品詞である。

- キーワード素性付きの名詞類
- 数詞・助数詞
- 接尾辞

品詞群 2 とはつぎの品詞である。

- キーワード素性なしの名詞類
- 未登録語

3.3 キーワード一致度計算

検索語の各単語の重要度をもとに検索語と検索語の一致度であるキーワード一致度を計算する。キーワード一致度は単語一致点と隣接一致点の積を正規化したものである。

3.3.1 単語一致点

単語一致点は索引語に含まれる単語と一致する検索語の単語の重要度の積である。m 個の単語から構成されている検索語 Q を $q_1 \cdots q_m$、n 個の単語から構成されている索引語 K を $k_1 \cdots k_n$ で表す。このとき、単語一致点はつぎの式で計算できる。

$$r_{word} = \prod_{i=1}^{n} f(k_i)$$

ここで、$w(q)$ を検索語 Q の構成単語 q に付与された重要度とすると $f(k)$ は

$$f(k_i) = \begin{cases} w(q_j) & : \exists q_j, q_j = k_i \\ 1 & : otherwise \end{cases}$$

3.3.2 隣接一致点

隣接一致点は索引語に含まれる単語並びと検索語に含まれる単語並びの一致数だけ隣接点をかけた値である。単語一致点だけでは検索語と索引語の構成単語の語順の相違がキーワード一致度に反映されないため、隣接一致点を導入した。

隣接一致点はつぎの式で計算できる。

$$r_{adj} = \prod_{i=1}^{n-1} g(k_i)$$

ここで、隣接点を α とすると $g(k)$ は

$$g(k_i) = \begin{cases} \alpha & : \exists q_j, q_j = k_i \& q_{j+1} = k_{i+1} \\ 1 & : otherwise \end{cases}$$

3.3.3 正規化

索引語と検索語が完全一致する場合でも、単語一致点・隣接一致点の積は、重要度付与および検索語の構成単語数によって異なる。こうしたことが起こらないように、単語一致点・隣接一致点の積を正規化したものをキーワード一致度とする。

正規化のために、まず検索語の得点を計算する。検索語の得点 v は索引語が検索語に等しい場合の一致度であり、つぎの式で計算できる。

$$v = \alpha^{m-1} \prod_{i=1}^{m} w(q_i)$$

このとき、完全一致時のキーワード一致度を β とすると、キーワード一致度 $r(Q,K)$ はつぎの式で計算できる。

$$r(Q,K) = \frac{\beta}{v} \times (r_{word} \times r_{adj})$$

3.4 テキスト一致度計算

テキストには複数の索引語が付与されている。各索引語に対するキーワード一致度から、テキストとしての一致度が計算される。テキスト一致度は索引語数に依存しないように、キーワード一致度の最大値をテキスト一致度とする。すなわち、テキスト T に対するキーワード一致度 $R(Q,T)$ はつぎの式で計算できる。

$$R(Q,T) = \max_{K \in T} r(Q,K)$$

表 2: 検索語への重要度付与

	品詞	キーワード素性	重要度
新	接頭辞	修飾性接頭語	2
素材	一般名詞		8
研究	サ変名詞	複合語語基	3
開発	サ変名詞	複合語語基	2

3.5 例

「新素材研究開発」を検索語とする。まず、検索語に重要度を付与する。重要度付与における基本点を 2 点、増加点を 1 点とすると、各単語の重要度は表 2 のようになる。正規化に用いる検索語の得点は、隣接点 $\alpha = 2$ として、$2^3 \times (2 \times 8 \times 3 \times 2) = 768$ である。

つぎに、テキスト一致度の計算する。索引語が「半導体レーザ開発」「新素材研究」であるテキストを想定する。まず、各索引語のキーワード一致度を計算する。「半導体レーザ開発」では、「開発」のみが検索語と一致するので単語一致点が 2、隣接一致点が 1 である。したがって、完全一致時のキーワード一致点 $\beta = 1000$ とすると、正規化により $(1000/768) \times 2 = 3$ 点となる。「新素材研究」では、単語として「新」「素材」「研究」が一致、単語並びとして「新／素材」「素材／研究」が一致している。したがって、単語一致点が $2 \times 8 \times 3 = 48$、隣接一致点が $(2 \times 2) = 4$ となり、正規化して $(1000/768) \times (48 \times 4) = 249$ 点となる。よって、テキスト一致度は 249 点となる。

4 文字成分表によるプリサーチ

4.1 概要

キーワードに基づくテキスト管理では転置ファイルを用いるのが一般的である。しかし、本システムは短単位キーワードに基づいているため、検索語ファイルでは複合語を単語列として保存しなければならない。そこで、転置ファイルは用いず、文字成分表によるプリサーチ法を採用した[2]。

[2]この方式ならば、テキストデータベースとしてキーワード検索だけでなく全文検索に対するプリサーチにも使用できる。

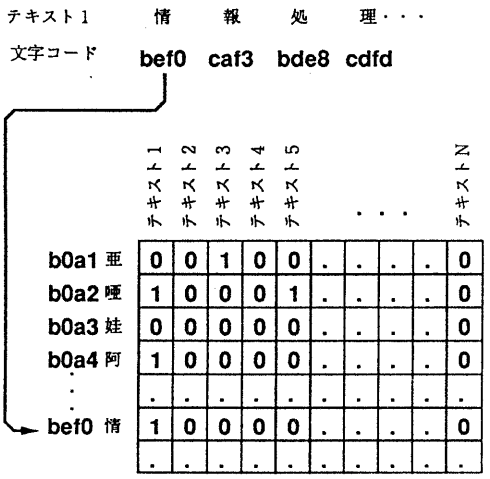

図 1: 一文字成分表

文字成分表とはテキストごとに出現した文字を記録する表であり、これを用いたブリサーチ法は 1 文字を 1 単語と捉えるスーパーインポーズドコード・シグニチャ (Superimposed Coding Signature) 法 [2] に等しい。テキストに出現する可能性のある文字セットに含まれる文字数を A とする。このとき、シグニチャーの大きさを A ビットとし、各文字に対し A ビットのうち 1 ビットのみがセットされるようにハッシュ関数を設定する。その結果、あるテキストのシグニチャはテキストに出現した文字に対応するビットが全てセットされたものとなる。この方式では、文字列が長くなるに従って検索精度が低下するという問題がある。これに対し、文字ごとではなく隣接する 2 文字ごとにシグニチャを作成する方式がある [6]。ただし、日本語の場合文字種が多い (約 7000) ためシグニチャの大きさは $7000^2 = 49 \times 10^6$ と非常に大きくなってしまう。

そこで、本システムでは以下に示す 2 つの文字成分表を用いる。

- 一文字成分表：
  各テキストごとに各文字が存在するか否かを示す表 (図 1)。l 個の文字から構成されているテキスト T を $t_1, \cdots, t_l$、T における文字 c に対応するビットを $S_{one}(c,T)$ で表す。このとき、

$$S_{one}(c,T) = \begin{cases} 1 & : \exists t_j, c = t_j \\ 0 & : otherwise \end{cases} \quad (1)$$

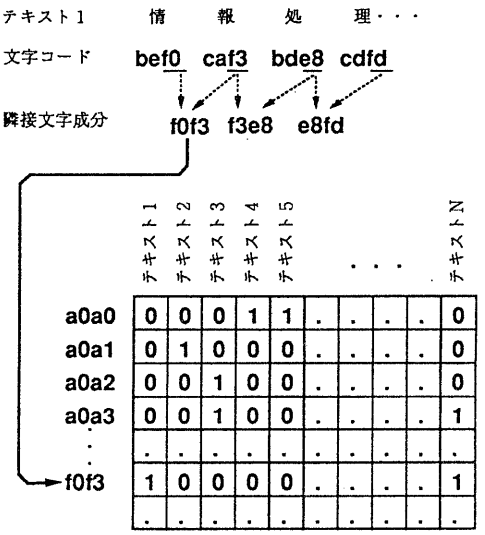

図 2: 隣接文字成分表

- 隣接文字成分表：
  各テキストごとに隣接文字が存在するか否かを示す表 (図 2)。ただし、隣接する文字の文字コードから下位 1 バイトずつ抽出し、連結した 2 バイトのビット列成分を 1 単語と捉える。隣接成分を抽出するフィルタを $F(t_i, t_{i+1})$、T における隣接成分 d に対応するビットを $S_{adj}(d,T)$ で表す。このとき、

$$S_{adj}(d,T) = \begin{cases} 1 & : \exists t_j, d = F(t_j, t_{j+1}) \\ 0 & : otherwise \end{cases} \quad (2)$$

隣接文字成分表では、文字コードの一部分しか用いていないため精度が低下する恐れがある。そこで、通常の一文字成分表も併用している。それでも、表の大きさは $7000 + 2^{14} = 23 \times 10^3$ であり[3]、2 文字ごとの場合よりはるかに小さい。

4.2 テキストの登録

テキスト登録時に、一文字成分表は式 (1) 、隣接文字成分表は式 (2) に従ってビットをセットする。

[3]本システムで日本語表現に用いたEUCでは、下位 1 バイトのうち実際には 7 ビットしか使用されていない。そこで 2^{16} ではなく 2^{14} となる。

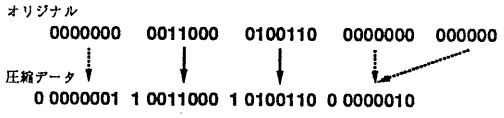

図 3: 文字成分表の圧縮

4.3 テキストの検索

m 個の文字から構成されている検索語 Q を $q_1 \cdots q_m$、一文字成分表においてある文字 c に対応するテキストの集合を $L_{one}(c) = \{T_i | S_{one}(c, T_i) = 1\}$、隣接文字成分表においてある隣接成分 d に対応するテキストの集合を $L_{adj}(d) = \{T_i | S_{adj}(d, T_i) = 1\}$ で表す。このとき、検索結果のテキスト集合は

$$\{\bigcap_i L_{one}(q_i)\} \bigcap \{\bigcap_j L_{adj}(F(q_j, q_{j+1}))\}$$

ただし、このなかにはテキスト中に検索語の連続する構成文字が分散して存在するテキスト、隣接文字成分として下位バイトしか抽出していないことに起因する誤検索のテキストも含まれる。

4.4 文字成分表の圧縮

本システムの構成でも、文字成分表は極めて大きい。そこで、文字成分表では 1 のビットの出現率が極めて小さいことに着目して、文字成分表の各文字成分エントリごとにデータを圧縮している。

圧縮アルゴリズムでは、7 ビット単位にオリジナルデータを調べ、バイト列である圧縮データを生成する。オリジナルの 7 ビットが全て 0 の場合、そのランレングス（7 ビットが全て 0 であるものの数）を圧縮データの下位 7 ビット、8 ビット目を 0 とする。7 ビットに 1 が含まれている場合、オリジナルの 7 ビットが圧縮データの下位 7 ビットとなり、8 ビット目を 1 とする（図 3）。圧縮データでは、各バイトの最上位ビットが下位 7 ビットの意味を決定する。圧縮データの最上位ビット 0 の場合、下位 7 ビットの値 X に対し、オリジナルは $X \times 7$ ビット連続の 0 である。最上位ビット 1 の場合、下位 7 ビットがオリジナルになる。

5 ライブラリ

これまで説明してきた機能を UNIX 上で利用可能なように C 言語のライブラリとした。

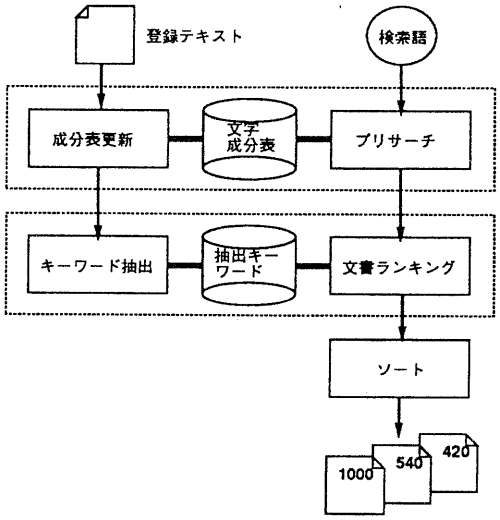

図 4: テキストの登録／検索処理概要

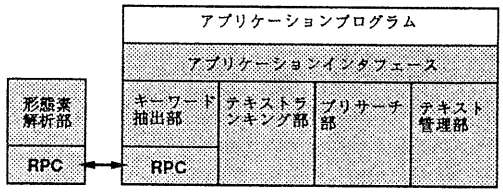

図 5: ライブラリの内部構成

テキストの登録・検索処理は図 4 のようになる。管理用データとして、プリサーチのための文字成分表とランキングのための抽出キーワードがある。テキスト登録時には、文字成分表の更新とキーワード抽出が行なわれる。テキスト検索時には、検索語に応じて、文字成分表を用いてプリサーチし、得られたテキストに得点付けし、得点順にソートして検索結果とする。

ライブラリの内部構成を図 5 に示す。キーワード抽出部、テキストランキング部、プリサーチ部は 2 ～ 4 章で説明した機能を実現するものである。テキスト管理部は登録テキストの id 管理を行なう。形態素解析部はキーワード抽出に必要な形態素解析を行なう。ただし、形態素解析には巨大な辞書が必要なため、アプリケーションごとに形態素解析部を持つことは非効率

的である。そこで、形態素解析部をサーバプログラムとし、アプリケーションは RPC (Remote Procedure Call) によりサーバと連携して処理する。

6 性能評価

6.1 精度

評価用データとして新聞記事 20000 件を用意した。精度の評価など詳細な検討が必要な場合には、このうち 200 件（情報処理分野 50 件とその他一般記事 150 件）を用いることとした。また、1 記事の平均長さは 650 文字（1300 バイト）である。

評価基準には、再現率と適合率を用いた [5]。再現率は、キーワード抽出においては抽出洩れの少なさ、テキストランキングにおいては検索洩れの少なさを示す。一方、適合率はノイズの少なさを示す。

6.1.1 キーワード抽出精度

本システムでは、キーワード抽出を検索のために行なうものと限定している。したがって、キーワード抽出結果のみの精度を評価することに大きな意味はないが、参考のために評価結果を示す。

200 件のデータについて再現率・適合率を求めた。その際、部分一致（人が判断したキーワードとシステムが抽出したキーワードが一部分でも一致すれば正解と判断する）と完全一致の両方について値を計算した。また、情報処理分野とそれ以外を独立に評価した。これはキーワード素性の付与において情報処理分野のエントリに重点をおいたためである。

評価結果を表 3 に示す。部分一致と完全一致を比較すると再現率・適合率ともに部分一致の方が高くなっている。短単位キーワードによる検索方式では、部分一致するものも検索結果となるので、本キーワード抽出法は検索方法から見て高精度といえる。また、情報処理分野と一般を比較すると、前者の方が再現率・適合率ともに高く、キーワード素性の重点的付与が有効であったことが確認できた。

6.1.2 テキストランキング精度

本システムでは検索結果はランキングされているため、全検索結果および上位 10 位のみを対象とした場合の評価値を計算した。

表 3: キーワード抽出精度

	部分一致		完全一致	
	再現率	適合率	再現率	適合率
情報処理分野	95.3	58.2	64.3	26.5
一般	94.2	54.3	60.8	26.0
平均	94.5	55.1	61.4	26.1

評価結果は表 4 に示す。全検索結果と上位 10 位を比較すると、再現率は若干低下しているが適合率は向上しており、検索結果の上位に正解が含まれていることがわかる。この点では、テキストランキング法の有効性が確認できた。しかし、再現率と比べ適合率がやや低く、適合率が向上するように重要度付与ルールなどを改良する必要がある。

表 4: テキスト検索精度

検索語	全検索結果		上位 10 位	
	再現率	適合率	再現率	適合率
BS	100	100	100	100
景気	100	100	100	100
NTT	100	100	100	100
パソコン通信	100	22	67	40
品質管理	100	45	100	30
バブル経済	100	32	86	60
NKK 半導体開発	100	28	45	90
抗ガン剤	100	82	100	90
29 型くらいの大画面テレビ	100	50	100	100
新素材研開	100	29	57	40
平均	100	60	86	75

6.1.3 プリサーチ精度

文字長ごとにリサーチを行なった場合の適合率を求めた[4]。その際、一文字成分表のみを用いた適合率とつぎのように定義される排除率も求めた。

$$\text{排除率} = \frac{\text{誤検索数 1} - \text{誤検索数 2}}{\text{誤検索数 1}}$$

ここで、誤検索数 1 は一文字成分表のみを用いた場合の誤検索数、誤検索数 2 は一文字成分表と隣接文字成分表をともに用いた場合の誤検索数である。排除率は隣接文字成分表を導入することで、誤検索をどの程度減らせたかを示している。

[4]再現率は必ず 100 % となるので測定しなかった。

表 5: プリサーチの適合率及び排除率

文字長	適合率		排除率
	一文字+隣接	一文字	
2	95.3	61.9	90.3
3	81.1	37.4	93.3
4	82.5	30.0	95.7
5	72.6	28.5	91.4
平均	82.9	39.5	92.7

結果を表5に示す。若干の誤検索があるが、プリサーチの誤検索はテキストランキングによって排除できるので、80 % 程度の適合率は絞り込みには十分な値と言えよう。また、隣接文字成分表の導入により誤検索を 90 % 以上減らすことができたことがわかり、隣接文字成分表の有効性が確認できた。

6.2 速度

使用マシンは Sun SPARCstation 2 (SPARC CPU 40MHz、メインメモリ 16 Mbyte)、内蔵 SCSI ディスク（アクセスタイム 16 msec、転送速度 4 Mbyte/sec）である。OS は Sun OS 4.1.1 で、処理時間はシングルユーザモードで測定した。

6.2.1 テキスト登録

テキスト登録処理には、文字成分表更新・キーワード抽出の 2 つの処理がある。200 件のデータで処理時間を測定したが、いづれの処理時間も登録テキストの長さと比例関係にあった。この関係を使って登録速度を計算したものが表 6 である。

表 6: テキスト登録速度

	速度 (chr/sec)
文字成分表更新	135.1
キーワード抽出	140.9
全体	69.0

6.2.2 テキスト検索

テキスト検索処理には、文字成分表によるプリサーチ・テキストランキングの 2 つの処理がある。20000 件のデータを使用し、ディスク上のデータがキャッシュされていない状態（cold start）とキャッシュされている状態（hot start）で測定した。

表 7: プリサーチ速度

文字数	検索速度 (text/sec)	
	cold start	hot start
1	69100	10200000
2	15800	92900
3	8970	36900
4	6610	28300
5	5310	24700

表 8: テキスト検索速度

絞り込み数	検索速度 (text/sec)	
	cold start	hot start
1000	1180	5000
2000	1000	2500
4000	830	1250
6000	690	833
8000	588	625

まず、プリサーチの測定結果を表 7 に示す。実際の利用環境では、ある程度キャッシュが効いた状態で利用されるので十分に実用的な速度である。

つぎに、テキストランキングも含めた検索速度を測定した。その結果、検索時間はプリサーチによって絞り込まれたテキスト数に依存していることがわかった。その関係から、絞り込み数に対する検索速度を計算したものが表 8 である。絞り込み数が増えるにともない検索速度が遅くなっている。また、予想通り host start の方が cold start より検索速度は速いが、両者の差は絞り込み数が少ない時ほど大きくなっている。これは絞り込み数が大きくなるにつれてヒット率が下がり、キャッシュの効果が小さくなるためである。

6.3 データ量

6.3.1 抽出キーワード

平均すると 1 テキストに 48.6 検索語が付与されており、また 1 検索語の構成単語数は 2.47 であった。この他に検索語の出現位置などのデータを保持しているため 1 テキストあたりのキーワードデータの量は 583 バイトとなった。また、テキストデータと比較すると 583/1300 = 44.8% である。

まだ、1 テキスト当たりのデータ量が多い。現在、

キーワード抽出の適合率が部分一致でも 55 % と低く、これを高めることでデータ量を減らすことが可能である。さらに、キーワードファイルのデータ構造を改良することなどでさらに改善可能である。

6.3.2 文字成分表

本システムの文字成分表は 2 文字ごとのシグニチャと比較して、$(23 \times 10^3)/(49 \times 10^6) = 0.05\%$ と非常に小さくなっている。さらに、圧縮によりデータ量を約 6.5% に圧縮している。結局、1 テキストあたりの文字成分表のデータ量は 603 バイトであった。また、テキストデータと比較すると $603/1300 = 46.4\%$ である。

これでも、文字成分表のサイズは小さいとは言えない。しかし、圧縮法の改良、文字成分表ファイルのデータ構造の改良によりサイズを小さくできる。

7 おわりに

われわれは短単位キーワードに基づくテキストデータベース管理システムを研究開発中である。本稿では、(1) キーワード素性を用いたキーワード抽出法、(2) 検索語重要度に基づくテキストランキング法、(3) 文字成分表によるプリサーチ法について説明した。キーワード素性を用いたキーワード抽出法では、辞書データ管理の簡素化、さまざまなアプリケーションに適用可能な柔軟性を達成できた。また、検索語重要度に基づくテキストランキングでは、複合語の一致度に基づく検索結果のランキング、高精度な検索を実現できた。さらに、文字成分表によるプリサーチでは、隣接文字成分表の導入による高精度化、ビット抽出とデータ圧縮による省データ化を実現できた。

今後は、検索精度とくに適合率の向上、より一層の省データ化などを行ないたい。

参考文献

[1] 加藤寛次, 藤沢浩道, 大山光男. 全文検索用テキストサーチマシンの開発. 電子情報通信学会技術研究報告, Vol. DE89, No. 38, 1989.

[2] C. Faloutsos and Christodoulakis S. Signature files: An access method for documents and its analytical performance evaluation. *ACM Trans. on Office Information Systems*, Vol. 2, No. 4, pp. 267–288, 1984.

[3] D. H. Kraft and D. A. Buell. Fuzzy sets and generalized boolean retrieval systems. *International Journal of Man-Machine Studies*, Vol. 19, pp. 45–56, 1983.

[4] Y. Ogawa, T. Morita, and K. Kobayashi. A fuzzy document retrieval system using the keyword connection matrix and its learning method. *Fuzzy Sets and Systems*, Vol. 38, pp. 17–41, 1991.

[5] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, 1983.

[6] N. Tavakoli and A. Ray. A new signature approach for retrieval of documents from free-text databases. *Information Procesing & Management*, Vol. 28, No. 2, pp. 153–163, 1992.

[7] 伊藤篤, 望主雅子, 小島裕一. 日本語形態素解析における素性を用いた解析方式. 第 *43* 回情報処理学会全国大会予稿集, 1991.

[8] 今郷韶, 望主雅子. 短単位キーワードを用いた文書ファイリングシステム. 第 *43* 回情報処理学会全国大会予稿集, 1991.

[9] 諸橋正幸. 自動索引付け研究の動向. 情報処理, Vol. 25, No. 9, pp. 918–925, 1984.

[10] 小川隆一, 菊池芳秀, 高橋恒介. フルテキスト・データベースの技術動向. 情報処理, Vol. 33, No. 4, pp. 404–412, 1992.

[11] 菅野祐司, 安藤敦史, 伊藤正雄. ワークステーション内蔵型フルテキストデータベースプロ セッサ sdp. 情報処理学会研究報告, Vol. 91ARC, No. 90–8, 1989.

[12] 堀池博巳, 久富丈志, 小澤義明. 日本語解析処理システム happiness について. 京都大学大型計算機センター広報, Vol. 24, No. 1, pp. 16–25, 1990.

[13] 木本晴夫. 日本語新聞記事からのキーワード自動抽出と重要度評価. 電子情報通信学会論文誌 *D-1*, Vol. J74-D-1, pp. 556–566, 1991.