自然言語処理 33－8
（1982. 10. 22）

拡張B－treeによる日本語単語辞書の作成

日高 達　吉田 将　稲永 紘之
（九大・工）　（九州芸工大）

1．はじめに

日本語は，単語単位にわかち書きされる欧米語等と異なり，べた書きされることから，日本語の機械処理では入力文の文字列を，単語機械辞書を用いて，単語列に変換するtry and errorの特殊な工程を必要とし，単語辞書の検索回数が非常に多くなる傾向がある．また，日本語では単語の正書法が確立していないため，漢字，漢字かな混り，かな書きによる単語表記が可能で，単語辞書の見出し語数は実質的な単語数よりもかなり多くなる．したがって，日本語の機械処理では単語機械辞書の検索能率が処理全体の能率に及ぼす影響が極めて大きく，日本語の特殊性を勘案した能率的な単語機械辞書を作成する必要がある．ここで，能率的とは次のような意味である．すなわち，日本語単語辞書は見出し語数が十万ないし二十万語にのぼるので，複数のブロックに分割してディスクパック等の二次記憶上に分散配置しておき，検索要求が生ずる度毎に検索に必要なブロックを一次記憶にデータ転送して検索を行なう（external search）ことになるが，二次記憶から一次記憶へのデータ転送は非常に時間がかかるため，

(a) 一回の検索で発生する，二次記憶から一次記憶へのデータ転送回数が少い．

ことは検索能率を高める上で最も重要である．また，

(b) 二次記憶上の利用効率が良い．

(c) 辞書項目の登録，削除の能率が良い．

ことも大切な点である．

(d) 一回の検索で，できるだけ多くの必要な情報を取り出すことにより，辞書の検索回数を低減することが望ましい．必要な情報は各々の処理毎に異なると考えられるが，べた書きを一大特色とする日本語文の機械処理では，与えられた文字列の左端に位置するすべての単語（“くるまだいそげ”の場合，“車”，“来る”，“車”，“車代”）が一時に検索できるようにすることはかなり一般的な要求と考えられる．例えば，与えられた文字列の左端に位置する最長の単語（“くるまだいそげ”の場合，“車代”）を検索することが，最長一致（longest match）の名称で多くの自然言語機械処理で採用されているが，多くの処理システムでは実際には，与えられた文字列の左部分列（“くるまだいそげ”の場合，“くるまだいそげ”，“くるまだいそ”，“くるまだい”，---，“くる”，“く”）を探索キーとして，複数回（最悪の場合には与えられた文字列の長さの回数）機械辞書の検索を行なうことで最長一致が実行されている．このようなことから，

(d') 与えられた文字列の左端に位置する単語が一回の検索ですべて取り出せる．

を日本語単語機械辞書の望ましい機能として，(d) の代りに採用する．

(d') の機能をもつデータ構造としてはmultiway treeのＴＲＩＥ[1]やbinary treeのＰＡＴＲＩＣＩＡ[1]があるが，いずれも (b) の点で不満足で，そのままの形態では大容量の機械単語辞書には向かない．[1] また，Ｂ－tree[1] は (a)，(b)，(c) の点で秀れたデータ構造であるが，(d') の機能がない．

本稿は，Ｂ－treeの拡張概念である拡張Ｂ－treeを提案し，この拡張Ｂ－tree

が (d') の機能をもち，かつ B-tree の秀れた機能 (a)，(b)，(c) を保存することを示す．すなわち，拡張 B-tree は次の i ～ iv の機能を有する

i　一回の検索で引き起こされる二次記憶から一次記憶へのデータ転送回数は $\log \frac{|D|}{2M} / \log \frac{C}{2M}$ 以下である．但し，|D| は辞書の見出し語総数，C は 1 ブロックの容量，M は辞書に収容される見出し語の集合によって決まる定数である．

ii　二次記憶の占有領域は 2 |D| 以下である．

iii　検索，登録，削除の各アルゴリズムは，B-tree の各アルゴリズムにほぼ準じた簡単なものである．

iv　(d') の機能を有する．

2．拡張 B-tree とその性質

キー集合の上の全順序 $\trianglelefteq$ と順序 $\fallingdotseq$ が定義され，次の二つの関係が成立するものとする．すなわち，任意のキー x，y，z に対して

$x \fallingdotseq y$ ならば，$x \trianglelefteq y$．　--- (1)

$x \trianglelefteq y \trianglelefteq z$，かつ $x \fallingdotseq z$ ならば，$x \fallingdotseq y$．　--- (2)

単語辞書の場合，キーは有限長の文字列だから，$\trianglelefteq$ は辞書式順序であり，$x \fallingdotseq y$ は文字列 x が文字列 y の最左部分語であることを示す順序と考えると，(1) および (2) が成立する．キーの列 x_1，x_2，---，x_M が

$x_1 \fallingdotseq x_2 \fallingdotseq \cdots \fallingdotseq x_M$

のとき，そのときに限り，キーの列 x_1，x_2，---，x_M は $\fallingdotseq$ に関する M 次の昇鎖（ascending chain）であるという．$\fallingdotseq$ に関する M 次の昇鎖が存在し，かつ，M より大きい次数の昇鎖が存在しないとき，そのときに限り $\fallingdotseq$ は M 次であるという．本稿では，$\fallingdotseq$ の次数を M と記す．

単語辞書の見出し語の全体集合を D と記す．D はキー集合の部分集合である．

本稿で提案する拡張 B-tree は大容量の辞書向きのデータ構造として広く用いられている B-tree の拡張であり，各節（node）は 2 つのキー集合 K_1，K_2 と $|K_1|+1$ 個の互いに異なる子（son）を指すポインター p_0，p_1，---，$p_{|K_1|}$ で次のように表わされる．但し，$|K_1|$ は集合 K_1 の要素の数（cardinal number）である．

$[(p_0, x_1, p_1, x_2, p_2, \cdots, x_{|K_1|}, p_{|K_1|}), K_2]$　--- (3)

但し，$K_1 = \{x_1, x_2, \cdots, x_{|K_1|}\}$，$x_1 \lneqq x_2 \lneqq \cdots \lneqq x_{|K_1|}$，$K_2 \subseteq D$ で，K_1 と K_2 の間には次の関係が成立する．葉以外の節 (3) において，

任意の $x \in K_2$ に対して $x_i \in K_1$ が存在し，$x \fallingdotseq x_i$．　--- (4)

K_1，K_2 に属するキーを，それぞれ，節 (3) の第一種のキー，第二種のキーという．すべての節の第二種のキーの総和は D に一致するものとする．

拡張 B-tree は次の i) ～ iv) の性質をもつ multiway tree である．

i)　すべての葉は同一の深さ（depth）に現われる．

ii)　根以外の節は $(\frac{C}{2} + 1 - M)$ 個以上で，かつ C 個以下のキーをもつ．

但し，$C \geq 2M$ とする．

iii)　節 (3) のポインター p_i が指す節を根とする部分木を T_i とする．T_i に含まれる任意のキー x に対して次のことが成立する．

$i = 0$ ならば $x \lneqq x_1$，$0 < i < |K_1|$ ならば $x_i \trianglelefteq x \lneqq x_{i+1}$，$i = |K_1|$ ならば $x_{|K_1|} \trianglelefteq x$．

iv) x を第一種のキーとする節のレベルを $l_1(x)$，y を第二種のキーとする節のレベルを $l_2(y)$ と記すと，$y \fallingdotseq x$ ならば $l_1(x) \leqq l_2(y)$ である．

順序 $\fallingdotseq$ が一次（$M=1$）の場合には，拡張B-treeの各節は第二種のキーを持たず，iv) は必ず成立するので，拡張B-treeはB-treeに一致する．

補題1．根と葉以外の節は $\lceil \frac{C+2}{2M} \rceil$ 個以上のまをもつ．

証明．節 (3) において，(4) および ii) より

$$|K_2| \leqq (M-1) \cdot |K_1|$$
$$\frac{C}{2} + 1 - M \leqq |K_1| + |K_2|$$

よって，$\frac{C}{2} + 1 - M \leqq M \cdot |K_1|$ となり，$\lceil \frac{C+2}{2M} \rceil \leqq |K_1| + 1$ となる．

（証明終り）

補題2．任意の第二種のキー x，x'（$x \fallingdotseq x'$）に対し，D の部分集合 D_x，${}_xD_{x'}$，${}_{x'}D$ を次のように定義する．

$$D_x \triangleq \{ y \mid y \in D,\ y \fallingdotseq x \}$$
$${}_xD_x \triangleq \{ y \mid y \in D,\ x \leqq y \fallingdotseq x' \}$$
$${}_{x'}D \triangleq \{ y \mid y \in D,\ x' \leqq y \}$$

すると，$|D_x|$，$|{}_xD_{x'}|$，$|{}_{x'}D|$ はいずれも $\frac{C}{2} + 1 - M$ 以上である．

（証明略）

補題3．$x \fallingdotseq y$ である任意の第二種のキー x，y に対して，$L_2(x) > L_2(y)$ であるか，x は x および y が同じ節の第二種のキーである．

証明．iv) より，$l_2(x) = l_2(y)$ の場合に x および y が同じ節の第二種のキーであることを証明すればよい．いま，x と y が互いに異なる節の第二種のキーであると仮定すると，これらの二つの節の共通の先祖（ancestor）とそれに含まれる第一種のキー z が存在し，$x \fallingdotseq z \leqq y$ となる．ここで $x \fallingdotseq y$ だから，定義より $x \fallingdotseq z$ となり，iv) より $l_1(z) \leqq l_2(x)$ となる．
一方，z を第一種のキーとして含む節は x を第二種のキーとして含む節の先祖だから，$l_1(z) > l_2(x)$ となり矛盾を生ずる．

（証明終り）

補題3．は次のことを意味する．すなわち，任意に与えられた文字列（argument）x の左端に位置するすべての単語を $y_1, y_2, \cdots, y_m$（$y_i \in D$，$y_i \fallingdotseq x$，$i = 1, 2, \cdots, m$，$y_1 \fallingdotseq y_2 \fallingdotseq \cdots \fallingdotseq y_m$）とすると，定義より

$$y_1 \fallingdotseq y_2 \fallingdotseq \cdots \fallingdotseq y_m$$

となるが，補題3より，集合 $\{y_1, y_2, \cdots, y_m\}$ の要素を第二種のキーとして含む拡張B-treeの節は各レベルにおいて精々一つである．したがって，根から葉まで，これ等の節を次々に辿って行けば x の左端に位置するすべての単語を一回の検索で取り出すことができる．この場合，辿って行く節の個数は葉のレベル＋1以下である．以後，拡張B-treeの葉のレベルを L（根のレベル＝0）と記す．

定理1．

拡張B-treeが占有する領域の大きさ（＝C×節の個数）および葉のレベル L と単語辞書の見出し語数 $|D|$ の関係は次のとおりである．

$$\text{領域の大きさ} < C \cdot \left(\frac{\frac{C}{2} + 2 - M}{(\frac{C}{2} + 1 - M)^2} \cdot |D| + 1 \right)$$

$$L < \log_{\lceil \frac{C+2}{2M} \rceil} \{ (\lceil \frac{C+2}{2M} \rceil - 1) \cdot \frac{\frac{C}{2} + 2 - M}{2(\frac{C}{2} + 1 - M)^2} \cdot |D| + 1 \} .$$

証明. $L = 0$（根が葉である）の場合は自明だから，$L > 0$ の場合に本定理が成立することを示す．補題1よりレベル1，2，3，---，L の節の個数は，それぞれ，2，$2 \cdot \lceil \frac{C+2}{2M} \rceil$，$2 \cdot \lceil \frac{C+2}{2M} \rceil^2$，---，$2 \cdot \lceil \frac{C+2}{2M} \rceil^{L-1}$ 以上である．

$$\therefore \text{レベル}1\text{以上の節の総数} \geqq 2 \cdot \frac{\lceil \frac{C+2}{2M} \rceil^L - 1}{\lceil \frac{C+2}{2M} \rceil - 1} .$$

根（レベル0の節）は1個以上のキーを有し，レベル1以上の節は $(\frac{C}{2} + 1 - M)$ 以上のキーを有するから，

$$\text{キーの総数} \geqq 2 \cdot \frac{\lceil \frac{C+2}{2M} \rceil^L - 1}{\lceil \frac{C+2}{2M} \rceil - 1} \cdot (\frac{C}{2} + 1 - M) + 1 \qquad ---(5)$$

また，第二種のキーの総数は $|D|$ だから，補題2より，第一種のキーの総数は $|D| / (\frac{C}{2} + 1 - M) - 1$ 以下である．

$$\therefore \quad \text{キーの総数} < (1 + \frac{1}{\frac{C}{2} + 1 - M}) \cdot |D| \qquad ---(6)$$

よって，(5)，(6)より

$$2 \cdot \frac{\lceil \frac{C+2}{2M} \rceil^L - 1}{\lceil \frac{C+2}{2M} \rceil - 1} \cdot (\frac{C}{2} + 1 - M) + 1 < (1 + \frac{1}{\frac{C}{2} + 1 - M}) \cdot |D|$$

となり，本定理が成立する．（証明終り）

定理1．より，$C \gg M$ の場合には，

領域の大きさ $\leqq 2|D|$ ，

$$L \leqq \log_{\lceil \frac{C+2}{2M} \rceil} \frac{|D|}{2M} .$$

3．検索アルゴリズムと能率

任意に与えられた *argument* x に対する検索は，根から葉まで次の操作を繰返すことにより行なわれる．

いま，節(3)が辿られたとすると，$\{ y \mid y \in K_2, \text{かつ} y \fallingdotseq x \}$ を出力する．次に，節(3)が葉であるかまたは $x \fallingdotseq x_i$ である第一種のキー x_i を含むならば停止し，そうでない場合には，

$x \preccurlyeq x_1$ ならば p_0 が指す節を，$x_i \leqq x \preccurlyeq x_{i+1}$ ならば p_i が指す節を，

$x_{|K|} \leqq x$ ならば $p_{|K|}$ が指す節を次に辿る．

定理2．

検索アルゴリズムを実行し，最大 L 回ポインターを辿ることにより $\{ y \mid y \in D, \text{かつ} y \fallingdotseq x \}$ が出力される．

証明. 検索アルゴリズムによりレベル l の節(3)が辿られたとする．いま，節(3)以外のレベル l の節 N と N の第二種のキー y が存在し，$y \fallingdotseq x$ であると仮定すると，節(3)および N の共通の先祖 N' と N' に含まれる第一種のキー z が存在し，$y \preccurlyeq z \leqq x$ となる．このことから，補題3．の証明と同様にして，N のレベルは N' のレベルより小さくなり，矛盾が生ずる．したがって，$y \fallingdotseq x$ となる $y \in D$ を第二種のキーとして含む節はレベル l では節(3)以外にはない．また，$x \fallingdotseq x_i$ となる第一種のキー x_i を節(3)が含むならば，$y \fallingdotseq x$ である任意の $y \in D$ に対して $y \fallingdotseq x_i$ となり，補題3．より，y は l 以下のレベルの節の第二種のキーである．

以上のことから，本定理が成立する．　（証明終り）

argument x に最長一致する単語 y は，x の左端に位置する単語の中で順序 $\leqq$ に関して最大だから，補題3．より，y は一番後に出力されることになる．

4．登録アルゴリズム，削除アルゴリズム

単語 x の登録は，根から葉まで，次の操作を繰返すことにより行なわれる．

いま，節(3)が辿られたとする．節(3)が葉であるか又は $x \leqq x_i$ である第一種のキー x_i を含むならば，節(3)の第二種のキーに x を加える．この場合，節(3)のキー数が C を越えるならば，節(3)の分割操作に移る．また，節(3)が葉でなく，かつ $x \leqq x_i$ である第一種のキー x_i を含まないならば，検索アルゴリズムと同様にしてポインターを辿り，次のレベルの節に移行する．

節(3)の分割操作は，$C < |K_1| + |K_2| \leq 2C$ になると起動されるが，節(3)が葉でない場合と葉の場合では少し手順が異なる．

葉でない節の分割操作

キー x に対して，$K(x)$，$K'(x)$，$K''(x)$ は次のような，互いに素な K_2 の部分集合とする．

$$K(x) \triangleq \{ y \mid y \in K_2, \text{かつ} y \leqq x \},$$
$$K'(x) \triangleq \{ y \mid y \in K_2, \text{かつ} y \leqq x \} \cap K(x)^c,$$
$$K''(x) \triangleq \{ y \mid y \in K_2, \text{かつ} x \leq y \}.$$

すると，次の(7)が成立するような，節(3)の第一種のキー x_i が存在する．

$$\left.\begin{array}{l} \frac{C}{2} + 1 - M \leqq (i-1) + |K'(x_i)| \leq C \\ \frac{C}{2} + 1 - M \leq (|K_1| - i) + |K''(x_i)| \leq C \end{array}\right\} \quad \text{---(7)}$$

この x_i で節(3)を次のような二つの節に分割する．

$$[(p_0, x_1, p_1, \cdots, x_{i-1}, p_{i-1}), K'(x_i)] \quad \text{---(8)}$$
$$[(p_i, x_{i+1}, p_{i+1}, \cdots, x_{|K_1|}, p_{|K_1|}), K''(x_i)] \quad \text{---(9)}$$

また，x_i と $K(x_i)$ をそれぞれ第一種のキー，第二種のキーとして節(3)の親に挿入する．すなわち，節(3)の親を

$$[(\cdots, x'_j, p'_j, x'_{j+1}, \cdots), K'_2] \quad \text{---(10)}$$

とする．ここで，ポインター p'_j は節(3)を指すものとする．

すると，x_i と $K(x_i)$ の挿入により親(10)は次の(11)に変る．但し，p'，p'' はそれぞれ節(8)，節(9)を指すものとする．

$$[(\cdots, x'_j, p', x_i, p'', x'_{j+1}, \cdots), K'_2 \cup K(x_i)] \quad \text{---(11)}$$

節(11)のキーの総数が C を越えれば，節(11)の分割操作が続いて起る．

葉の分割操作

次の(12)が成立するような，葉(3)の第二種のキー x が存在する．

$$\left.\begin{array}{l} \frac{C}{2} + 1 - M \leqq |K'(x)| \leq C \\ \frac{C}{2} + 1 - M \leq |K''(x)| \leq C \end{array}\right\} \quad \text{---(12)}$$

この x で葉(3)を次のような二つの節に分割する．

$$[K'(x)] \quad \text{---(13)}$$
$$[K''(x)] \quad \text{---(14)}$$

さらに，x と $K(x)$ をそれぞれ第一種のキー，第二種のキーとして親(10)に挿入する．すなわち，親(10)は次の(15)に変る．但し，p'，p'' はそれぞれ葉(13)，葉(14)を指すものとする．

[(---, x_j, p', x, p'', x_{j+1}, ---), $K_2 \cup K(x)$] --- (15)

節(15)のキーの総数がCを越えれば，節(15)の分割操作が続いて起る．

以上の分割操作において，節(3)が親をもつと仮定したが，節(3)が根の場合は自明であるから省略する．また，登録アルゴリズムの実行によって，拡張B-treeの性質i)～iv)が保存されることの証明は紙面の都合上省略する．

単語xの削除は，単語xの登録の逆過程である．したがって，削除アルゴリズムは登録アルゴリズムより容易に推量できるので述べない．

4. あと書き

日本語文はべた書きされるので，機械処理では入力文の文字列を，単語辞書を用いて，単語列に変換する特殊な工程を要し，単語機械辞書の検索回数が非常に多くなる傾向がある。したがって，単語機械辞書の検索能率が処理全体の能率に及ぼす影響が極めて大きい。

本稿では，単語辞書のデータ構造としてB-treeの拡張概念である拡張B-treeを提案し，拡張B-treeが次のi～ivの機能を有することを示した。

i　一回の検索で引き起こされる二次記憶から一次記憶へのデータ転送回数(B-treeの高さ)は，$\log \frac{|D|}{2M} / \log \frac{C}{2M}$ 以下である。

ii　拡張B-treeの占有領域は，$2|D|$ 以下である。

iii　検索，登録，削除アルゴリズムは，B-treeの各アルゴリズムにほぼ準じた，比較的簡単なものである。

iv　任意に与えられた文字列の左端に位置するすべての単語が一回の検索で取り出せる。

我々は，拡張B-treeを用いて基本単語数83,000の日本語単語辞書を作成し，日本語の機械処理に用いている。本単語辞書の木の高さは2(L=2)，したがって，一回の検索で引き起こされる二次記憶から一次記憶へのデータ転送回数は2以下である。

文献

[1] D. E. Knuth, "The Art of Computer Programming" Addison-Wesley Publishing Company, vol. 3.

[2] 日高，吉田，"効率的日本語単語辞書"，情報処理学会第24回全国大会予稿集，p. 1003(昭和57年3月)．