トライ構造を用いた共起情報の効率的検索アルゴリズム

5L－2

森田和宏　望月久稔　獅々堀正幹　青江順一
徳島大学工学部知能情報工学科

1．　はじめに

自然言語処理システムにおける単語の共起関係は有用であるが，実用システムとして有用な共起情報は膨大な数となり，それを格納する辞書の記憶倹約が必要不可欠となる．また，複合語に関しても，従来は複合語を基本単語に分割して記憶し，少ない語数で多くの複合語を扱う研究がなされてきた．しかし，この方法では複合語を一つの語として扱う事が困難であり，また，分割，復元のプロセスが複雑になりがちであった[3]．

そこで，本稿では二つの基本単語からなる共起単語（複合語も含める）を一つのトライ構造で効率的に記憶検索する手法を提案する．

2．　共起情報の記憶

2．1　トライ構造

以後，トライの葉とキーを１対１に対応させるために，キー自身には含まれない終端記号‘#’をキーの最後につけて説明する．

図１にキー集合K={“電気工学＃”，“電子工学＃”，“情報処理＃”，“共起情報＃”}に対するトライの例を示す．ここで，記号‘#’とはトライの葉とキーを１対１に対応させるために用いる終端記号である．トライによる検索は一文字ごとに進められるので，最悪の検索時間はキーの長さに比例し，キーの登録数とは無関係なので，登録キー集合が大きい場合でも，高速な検索が可能である．　また，各キー内の共通接頭部分が同じ遷移で共有されているため，記憶量は軽減している．　しかし，一度分岐した遷移は再び併合されず，共通接尾辞部分が共有されないため，登録キーが増加するとトライの状態数が大きくなり，記憶量は増加する．

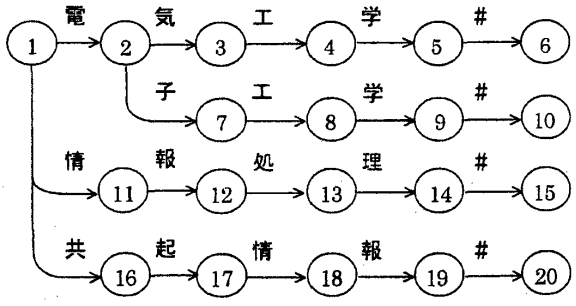

図１　Kに対するトライ

2．2　トライ構造による実現

本手法では図２のように複合語を構成する基本単語をすべて一つのトライに登録し，トライの葉と葉をリンクする．この際，複合語を構成する際の前後関係の情報をリンクに持たせる．たとえば，複合語“情報処理”は“情報”と“処理”をトライに登録し，“情報”の後ろに続くリンクを“処理”に，“処理”の前に続くリンクを“情報”に引くことによって複合語を登録する．これにより，“情報処理”，“共起情報”の二つの複合語を登録した場合でも“情報”という基本単語の登録は一回で済み，記憶量の節減になる．

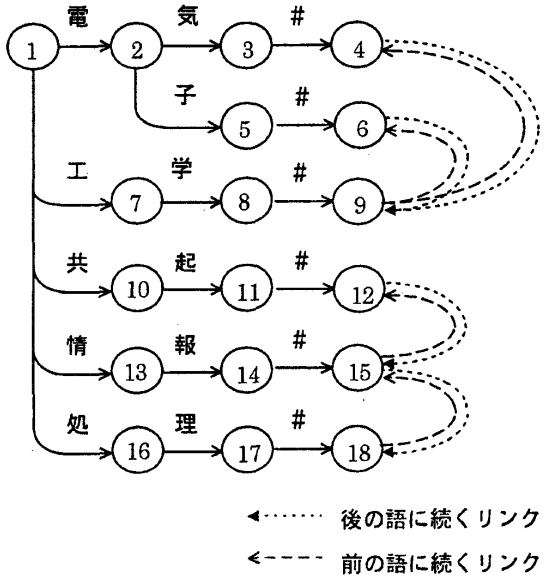

図２　Kに対する本手法のトライ

3．　データ構造

以後，ノードnからmへ枝ラベル‘a’が定義されていることをgoto関数gを使ってg(n,a)=m

An Efficient Retrieval Algorithm of Collocational Information Using Trie Structures
Kazuhiro MORITA, Hisatoshi MOCHIZUKI, Masami SHISHIBORI and Jun-ichi AOE
Faculty of Engineering, The University of Tokushima

で定義する．

本研究では，トライ構造部分の構築にコンパクト性と検索の高速性に優れたダブル配列[1],[2]を用いる．ダブル配列法では，二つの配列 BASE，CHECK でトライの遷移を表現する．ダブル配列上のインデックス番号はトライの状態番号と1対1に対応しており，以下の説明では簡単のため，両者の値を一致させて説明する．

ダブル配列は遷移 g(s, a)＝tに対して，次式を満足する．

$$t=\mathrm{BASE}[s]+a \quad (1)$$

$$\mathrm{CHECK}[t]=s \quad (2)$$

すなわち，g(s, a)の値は式(1)の tで決定され，CHECK[t]には，状態 t に入る遷移が状態 sから引かれていることを示す情報を格納する(式(2))．本手法では，リンク部分のアークを2次元配列 LA に蓄える．トライの葉のノード sからこの LA を参照するために，BASE[s]には LA の配列番号 qを格納する．ただし，トライの葉のノードを判定するためこの格納値は-qと符号を変えておく．

ダブル配列では，式(2)により CHECK に遷移前の状態番号を格納するので，最悪の時間計算量 O(1)で遷移を逆にたどることができる．

図2に対するダブル配列及び配列 LA を，入力記号の内部表現値とともに図3に示す． 図2の“工学”へのリンクのように複数のリンクがある場合は，LA[4]のようにリンクの番号を LA の2次元目に格納している．キーが追加された場合も LA の2次元目に追加していくことによりリンクを表現できる．

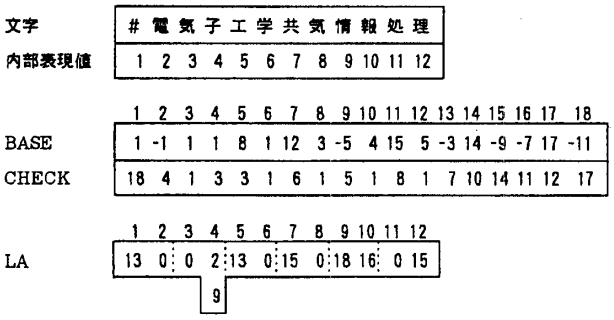

図3 図2に対するダブル配列

4． 実験結果

本手法の構成システムは約2,000行の C++言語で記述されており，DELL OptiPlex XMT5120 (CPU:pentium[120MHz])上で稼動している．

表1に示す実験結果は，5漢字からなる複合語をキーとした各々のキー数での結果である．表1の対象手法は通常のトライ構造のみのシステムとする．また，圧縮率は対象手法のノード数に対する本手法のノード数の割合を示し，記憶量の比率は，対象手法の記憶量に対する本手法の記憶量の比率を示している．検索時間は全キーを検索した場合の1語当たりに要する時間である．

本手法での総ノード数は，対象手法に比べて約24%～67%に減少している．これは，対象手法での接尾辞が効率的に圧縮されていることによる．また，記憶量についても，対象手法に比べて約56%～100%となり，本手法の有効性がわかる．

検索時間については対象手法の方が若干速くなっている．これは，本手法でのリンク部分の検索がリンクしている単語の数に比例するため，語数が増えると検索効率が悪くなるためである．

表1 実験結果

キー総数	100	1,000	10,000	50,000
総ノード数				
本手法	314	2,355	15,354	37,680
対象手法	471	3,454	33,755	154,959
圧縮率(%)	66.7	68.2	45.5	24.3
総リンク数	161	1,130	8,362	48,929
記憶量比率(%)	100.8	100.9	70.3	55.9
検索時間(ミリ秒)				
本手法	7.1	9.8	11.1	12.5
対象手法	7.7	8.4	10.2	11.8

5． まとめ

共起情報を効率的に格納するために，トライ構造を用いた辞書構造を提案し，それをダブル配列で実現する手法を提案した．また，二つの基本単語からなる複合語の共起情報に対して実験結果により空間的および時間的有効性を実証した．

今後の課題は，3つ以上の基本単語からなる共起情報のレコード情報の効率的格納法，検索時にキー数に依存しないようなリンク部分の格納構造の考案等である．

参考文献

[1] 青江順一：ダブル配列による高速ディジタル検索アルゴリズム，信学論（D），Vol.J71-D, No.9,pp.1592-1600 (1988).

[2] 青江順一：自然言語辞書の検索－ダブル配列による高速検索アルゴリズム，bit(共立出版) Vol.21, No.6, pp.776-784 (1990).

[3] 森本勝士，青江順一：トライ構造による形態素辞書と共起辞書の統合法，情処研資，91-NL-85,3(1991).