データベースシステム 116－47
（1998. 7. 9）

内容解析に基づく文書構造の自動抽出

品川 徳秀 † 北川 博之 ‡
† 筑波大学 工学研究科 ‡ 筑波大学 電子・情報工学系

概要

近年の計算機環境の普及に伴い、電子化文書の重要性は更に高まってきている。それらの潜在的な数、量は膨大なものであり、必要な情報を利用する事が容易ではなくなってきている。本稿では、文書中から話題の階層を抽出し、問合せに適合する部分を柔軟に検索するための手法を提案する。これにより、様々な抽象度の部分文書を、問合せ条件に応じて検索および呈示の単位として利用する事や、文書に内在する論理構造を用いた検索が可能になる。本手法について、実験を通じた評価を示すとともに、転置ファイルを用いて、問合せに応じて動的、局所的に文書構造を抽出する方法についても述べる。

Extraction of Document Structures Based on Contents Analysis

Norihide Shinagawa † Hiroyuki Kitagawa ‡
† Doctoral Program in Engineering, University of Tsukuba
‡ Institute of Information Sciences and Electronics, University of Tsukuba

Abstract

Importance of digital documents is growing with the recent advances of computing environments. The volume of document data is huge and it is not easy to access relevent information. In this paper, we propose a method to extract logical structures embeded in documents and retrieve relevent passages from documents flexibly. This method makes it possible to use passages of various abstraction levels as units in document retrieval. Experimental evaluation of this method is given. We also describe a technique to dynamically and locally extract logical strctures using inverted file strucures.

1 はじめに

近年、計算機環境の発展と普及に伴い、様々な情報資源が電子化され、活用されるようになってきている。中でも、電子化文書は重要な情報資源であり、それゆえにその潜在的な数、量は膨大なものである。このため、その全体を把握する事も、個々の内容を理解する事も必ずしも簡単ではなくなっている。この問題に対し、分類や検索などのプロセスを電子メディアの特性を生かして自動化するための研究がされている。

従来の文書検索においては、個々の論文や書籍などの文献をあらかじめ決められた検索対象単位として、問合せ処理や処理結果の呈示を行なうシステムが多かった。しかし、近年の全文テキストデータベースの増加や電子文書データの増加に伴い、具体的に文献データのどの部分が問合せに合致するかまでを特定する部分文書検索 (passage retrieval) の必要性が認識されている [1]-[9]。

部分文書検索において、検索対象を決定するための方法としては大きく分けて2種類の方法がある。一つは、あらかじめ著者によって明示的に与えられた章、節、段落のような構造を用いる方法である。もう一つは、内容の解析によって話題の推移点を推定し、それによって抽出される話題のまとまりを用いる手法である。

[4] は、前者の方法の一つを示しており、階層性に基づいてトップダウンに適合部分を特定する手法である。しかし、これらの方法においては、著者の視点で与えた文書構成要素が必ずしも話題の推移を反映していない場合が考えられる。例えば、一つの話題が複数の構成要素に渡って展開されていたり、逆に一つの構成要素が複数の話題を含んでいたりするなどである。このため、この構造はヒントと

しては有益であるが、決定的なものではない。また、トランスクリプト等のテキストデータの種類によっては、常に十分な構造をあらかじめ与えらる事が難しい場合もある。

一方、[2], [5] などは、文書内容の解析に基づき、話題のまとまりの抽出を行なう手法について述べている。これらにおいては、文書は自動抽出された話題の列として扱われ、個々の話題が問合せ処理および結果呈示の最小単位として用いられる。

ここでは、抽出された話題は最小粒度の部分文書として振舞うため、より小さな部分のみが問合せに適合する場合には、その詳細な構造を分析し直さない限り対応できないという問題がある。また、連続する複数の話題が適合する場合にはそれらを含む、より大きな部分文書を提示する事が、情報の分散と適合候補の増加を防ぐ上で有効である。

このような問題は、話題の境界は本来、その抽象化の度合に応じて定まるという相対的な面に起因している。それゆえ、話題の抽象化の度合を構造分析の段階で固定的に与え、その結果の話題のまとまりを最小粒度の部分文書として扱う事、そして、文書を単階層からなる単純な列として扱う事は得策ではない。この事から、必要に応じてその抽象化の度合を調整できる必要があると考える。

本研究では、これらの問題点に対応するため、文書データ中の話題の階層を抽出するボトムアップな手法を提案する。また、このように抽出された階層要素を問合せ処理および結果呈示の単位として用いる検索手法とその有効性評価を示す。更に、従来の転置ファイルを用いて、与えられた問合せに対して動的、局所的な構造抽出を行なう事で、本検索手法を実装するための方法を示す。

2 節では本研究で提案する文書の構造抽出法について説明し、3 節でこれに基づいた部分文書検索について説明する。4 節において、これらについての実験を通じた評価を行なう。5 節では、転置ファイルを用いて、問合せ時に動的、局所的に構造抽出を行なうための方法について説明する。最後にまとめと今後の課題を述べる。

2 文書構造の抽出

構造抽出のプロセスとしては、トップダウンに行うものとボトムアップに行うものが考えられる。本手法は、ボトムアップな手続きになっている。対象とする文書の構造の初期状態として十分に小さく機械的に境界を設定する。これによって作られる部分文書を**基底ブロック**と呼ぶ。また、基底ブロックからそれを結合したものとして段階的に構成される部分文書を **(複合) ブロック**と呼ぶ。基底ブロックは、最小粒度のブロックとして位置付ける事ができる。まず、構造の抽出手順について述べる前に、基本事項について説明する。

2.1 文書間の類似度

2.1.1 ベクトル表現とコサイン測度

二つの文書があった時、それらの類似性を測る基準として様々な測度が提唱されている。中でも、**tf*idf** と呼ばれる方法で語の重み付けを行ない、これによって得られたベクトル表現の**コサイン測度**が良く知られている。本手法でもこれに従う。

N 個の文書からなる文書集合 R が与えられた時、各文書 d_i は、それに出現する語 j の出現頻度 t_{ij} と、その語の出現する R 中の文書数 n_j を用いて次のようなベクトルとして表わされる。

$$v_i = (v_{ij})_{j=1,\ldots,k}$$

$$v_{ij} = \frac{t_{ij}}{\log(n_j/N+1)}$$

文書間の類似度は対応するベクトルによって与えられる。ここでは最も基本的な次のものを利用する。

$$sim(d_1,d_2) = cos(v_1,v_2) = \frac{v_1 \circ v_2}{\|v_1\| \cdot \|v_2\|}$$

2.1.2 文脈を用いたベクトルの補正

上記のようなベクトル表現は、文書集合における各文書の語彙の偏りによって記述される内容を表現するというものである。ある程度大きな文書においてはその記述が豊富になるため十分に表現されようが、本稿で対象とするブロックは部分文書であり、普通の文書に比べて非常に小さいものとなる場合がある。このようなブロックでは、その中に出現する語のみでその意味内容を判定する事が難しい事がある。そこで、各ブロックの前後の文脈の情報を用いて当該ブロックのベクトルを補正する事を考える。

一般に、あるブロックにおける記述量が不十分であるならば、その周辺と合わせて一つの話題である可能性が高く、即ちその周辺のブロックと補完しあう関係になる。また、記述位置が離れる程、その依存関係は薄くなる。このような特徴から、その前後のベクトルによって次式のように補正を行なう (図 1)。

$$v_i' = v_i + w_i^{(r)}$$

$$w_i^{(r)} = \sum_{p=1}^{r} \frac{w_{F-p} + w_{L+p}}{(1+p)^{(L-F+1)}}$$

ここで、F, L はそれぞれ、対象ブロック d_i の最初と最後の基底ブロック番号とし、v_i は d_i の、w_j は基底ブ

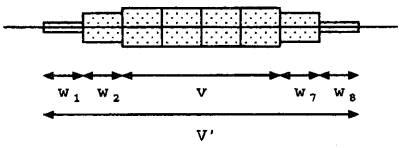


図 1: 隣接ブロックによる補正

ロック b_j の tf*idf によるベクトル表現とする。図 2 に示す重み付け関数の性質から、この補正ベクトルは対象部分文書が大きくなるに従って実効性を持たなくなる。また、同様に対象部分から離れた部分の影響力は急激に小さくなる。以下では、近傍 5 ブロックを用いる事として検討を行なう。

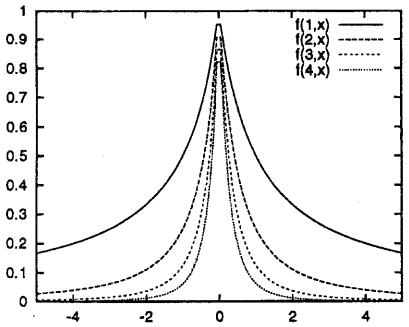


図 2: 重み付け関数 $f(n,x) = 1/(1+|x|)^n$

2.2 構造抽出手順

文書構造の抽出は、以下の二つのステップで行なう。

1. 基底ブロック毎に頻度情報を取得
2. 頻度情報を用いて構造を抽出

これらについて、以下で説明する。

2.2.1 頻度情報の取得

始めに、基本情報を取得する手順を示す (図 3)。

本節の始めに述べた通り、初期状態として基底ブロックの集合を考える。この基底ブロックは、与えられた条件に従って機械的に求められる。例えば、語数や文数を基準にして、それを上回るところを基底ブロックの境界とみなすなどである。

次に、このようにして定めた基底ブロック毎に、語の頻度情報を得る。その際、語幹抽出や不用語の除去などを行なう。また、必要ならば頻度情報を頻度情報ファイル中に格納しておく。

```
// 語の頻度情報を取得
get_doc_info( doc, cond )
    {
    // 条件に従って基底ブロックに分解
    blocks = split_to_block( doc, cond );

    // 基底ブロック毎に、語の出現頻度を数える
    for( block in blocks )
        for( term in block )
            termfreq[ block ][ term ] ++;

    return termfreq;
    }
```

図 3: 頻度情報の取得

2.2.2 構造抽出

構造抽出の手順は次のようなものである (図 4, 図 5)。まず、基底ブロック毎の頻度情報を頻度情報ファイルから取得し、ブロック集合の初期状態として、基底ブロック集合を割り当てる。

ブロック集合中の連続するペアのうち、最も類似度の高いものを選び、統合する。複数の候補が存在する場合、直前のステップで統合されたブロックの直後のものを用いる。次に、ブロック集合において、元のブロックを統合されたブロックで置き換える。以上を繰返し、一つの大きなブロックのみを含む集合となった時点で、構造の抽出は終了する。最終的に、構造は二分木として得られる。

ここで、ブロック間の類似度は、tf*idf によって構成されたベクトルのコサイン測度で測られるが、文書集合 R としては各繰返しステップ毎でのブロックの集合を用い、n_j は語 j の出現するブロック数とした。このため、ステップ毎に idf は異なる値を取る。即ち、ブロック間の類似度はその時点で抽出されている構造の状態に依存する事に注意を要する。

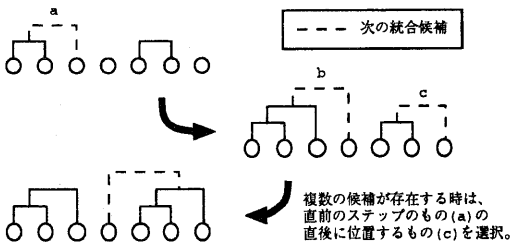


図 4: 構造抽出過程の一部

3 部分文書検索

本節では、前節で説明した手法によって抽出された構造を利用した部分文書検索について説明する。

文書中の検索候補は、前節の手順を適用して得られた二分木の任意の節である。この木構造は、文書中の話題の塊を反映しており、話題の抽象化が高いほど上位の節に対応する。この事から、話題の構造を反映した上で、問合せに対して最も適切な部分を特定できる事が期待される。

ここでも類似度の計算にはコサイン測度を用いているが、その際の idf は、全ての節に対応するブロックの集合に基づいて計算される。結果は、この類似度に基づいて順位付けされる。

また、このような階層化された話題の構造を利用する事によって、その構造を条件に含めた問合せの処理が可能であると考える。例えば次のような問合せが考えられる。

- 問合せとして与えられる話題全てを含む部分文書
- 話題 A に関する記述の中で展開されている話題 B に関する部分文書
- 話題 A と連続した話題 B に関する部分文書

4 評価実験

以上に述べた、構造抽出とそれを用いた部分文書検索手法の有効性を評価するための実験を行なった。

4.1 対象データ

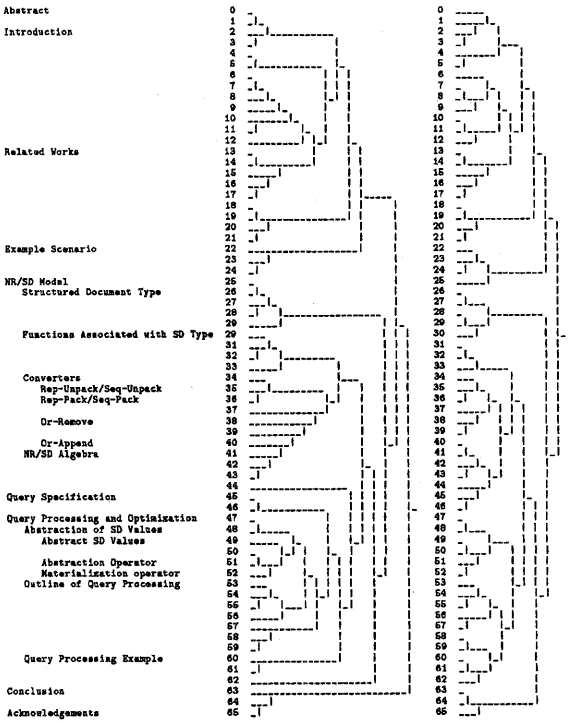
本研究では、本手法での構造抽出の有効性を評価するため、それに内在する文書構造が比較的明快な数本の英語論文を対象とした。これらの対象文書からは、あらかじめ、タイトルや節などの見出しを取り除き、本文のみを利用した。基底ブロックの境界として、25, 50, 100, 150 語以上を含むような文の区切りを用いた。また、評価の都合上、段落の境界は必ず基底ブロックの境界とした。これにより、元の文書での節の区切りの再現性を容易に見る事ができる。更に、段落の最後の基底ブロックが極端に小さい場合、語数が同程度になるように段落内で境界を調整した。

ここで、基底ブロックが段落に比べて十分に小さければ、上で与えた制約はそれほど重要な影響を与えない。逆に、基底ブロックを大きめにとった時には影響を与える可能性が考えられる。

以下では、100 語を基準にした場合について説明する。ここで実験結果を示す文献に対しては、基底ブロック数は 65 個であった。

```
// 構造抽出
extract_structure( doc, cond )
    {
    // 頻度情報を取得
    termfreq = get_doc_info( doc, cond );

    // 初期状態として、未統合ブロックの
    // 集合に基底ブロック集合を登録
    blocks = get_blocks( termfreq );

    // ブロック集合が一つだけになるまで
    while( blocks.size > 1 )
        {
        // 統合するペアを選んで
        pair = select_pair( blocks );

        // 統合したものでこのペアを置き換える
        blocks.insert( pair );
        blocks.remove( pair.left )
        blocks.remove( pair.right );
        }
    return blocks;
    }

// 統合するブロックのペアを求める
select_pair( bs )
    {
    top = 0;
    static next = bs.first;

    // 前のステップで選んだものから
    // 最後までを範囲として候補を選ぶ。
    p = select_pair( bs, next, bs.end );
    if( next != bs.first )
        {
        // 最初から前のステップで選んだ
        // ものまでを範囲として候補を選ぶ。
        q = select_pair( bs, bs.first, next );

        // 最終的な候補を選ぶ。
        // 同じ類似度のものが存在するなら
        // 前ステップでの候補の直後のものとする
        p = ( ( p.sim >= q.sim ) ? p : q );
        }
    next = p + 1;
    resurn p;
    }

    // 範囲を限定して候補を選ぶ
    select_pair( bs, first, end )
        {
        // このステップにおける idf
        docfreq = get_docfreq( bs );

        // 連続するブロック b1, b2 について
        for( b1 = first ; b1 != end - 1 ; ++ b1 )
            {
            b2 = b1 + 1;

            // その類似度を測る
            similarity = sim( b1, b2, df );

            // 最も高い類似度を持つもののうち
            // 最初に出現したものを候補とする
            if( top < similarity )
                {
                top = similarity;
                result.left  = b1;
                result.right = b2;
                result.sim   = top;
                }
            }
        return result;
        }
```

図 5: 構造抽出手順



(a) 基本的な測度を利用　(b) 文脈情報も利用

図 6: 抽出された構造

> The NR/SD model provides the apply operator (denoted by <>) to extract elements from sd values contained in relations. Region algebra expressions are used to give the extraction specification. Suppose that <> is a relation which has an SD type attribute <>, <> is a region algebra expression, and <> is a new attribute name. Then, <> adds the new attribute <> to the original relation <>. The attribute <> stores the SD values which are returned as the result of applying the region algebra expression <> to the SD value in the attribute <>.

図 7: 問合せとして用いた断片文書

4.2　基底ブロックの大きさと文脈情報

100 語を基準とした基底ブロックを用いて、実験対象文書の構造抽出した結果を図 6 に示す。図 6.(a) は標準的な tf*idf を用いた場合であり、図 6.(b) は文脈情報で補正を行なった tf*idf を用いた場合である。ここでは、基底ブロックの大きさの影響と 2.1.2 節で示した文脈を用いたベクトルの補正の効果について述べる。

図 6 に示した文書においては、ブロック間の類似度に文脈情報を加えた事で、構造抽出の精度がより向上していると考えられる。

まず目につくのは、基本的なコサイン測度を用いた場合、比較的遅くに統合される基底ブロックが多い事である。分割の都合上、基底ブロックには平均的な大きさの 1/3 程度のものが幾つか含まれているためかと考えられる。しかし、統合の遅いブロックとして、実際には平均以上の大きさを持ったものも幾つか見られた。また、小さな基底ブロックであっても早期に結合されるものもあり、一概にこのためだけと言えない。これに対し、文脈情報を用いた場合にはそのようなものはなく、どの深さでも比較的均質にブロックが結合されている。これは、文脈情報による補正によって、より内容の連続性が反映されたためと思われる。

また、内容的にはっきりと分かる境界に関しては、より忠実に再現されている。例えば、NR/SD Model に関する節での各小節の境界が同程度以上に抽出されているのが見て取れる。また、Query Spesification に関する節は、その次節の冒頭よりも NR/SD Model に関する節の後半に関連する記述がされており、この事も反映されている。

これらの傾向は、他の文書に対する実験結果についても同様に見られた。特に、文脈情報を用いた場合では、基底ブロックの取り方にはあまり依存せず、どれも類似した構造が得られている。但し、必ずしも改善される事が保証されるわけではなく、部分的に構造が崩れてしまう場合もあった。しかし、極端に悪くなる事はほとんど見受けられず、今回の実験の範囲では悪くても同程度の精度で構造を抽出できている。

以上の結果から、表現ベクトルを文脈情報で補正する事で、ある程度の大きさのばらつきに強く、より明確に境界を発見できるものと考えられる。但し、現時点では評価対象文書数が十分とは言えないため、ここで用いた対象に固有の傾向である可能性もあり、更なる調査が必要である。

4.3　部分文書検索

次に、問合せとして、この文書中のブロック 33 (図 7) を用いて検索を行なった結果の上位 20 件 (総件数 183) をリストしたのが図 8 である。(問合せ中の <> は数式が含まれていた事を示している)

(a) 基本的コサイン測度				(b) 文脈情報を考慮			
類似度	ブロック範囲		大きさ	類似度	ブロック範囲		大きさ
1.000	33	33	1	0.905	33	33	1
0.924	30	33	4	0.843	32	33	2
0.653	41	43	3	0.667	32	32	1
0.553	42	43	2	0.623	31	33	2
0.523	41	41	1	0.573	40	40	1
0.521	39	43	5	0.561	40	42	2
0.498	40	43	4	0.553	43	43	1
0.498	34	44	11	0.530	40	44	5
0.493	30	61	32	0.513	42	42	1
0.492	38	43	6	0.496	42	43	2
0.480	30	62	33	0.488	40	46	7
0.477	37	43	7	0.476	39	39	1
0.458	34	43	11	0.457	26	65	39
0.442	25	62	38	0.452	45	45	1
0.420	34	36	3	0.441	47	47	1
0.420	30	43	14	0.431	47	62	16
0.413	45	46	2	0.428	34	46	11
0.409	26	29	4	0.412	26	30	5
0.400	42	43	2	0.392	31	65	35
0.370	44	44	1	0.371	52	52	1

図 8: 問合せ結果

図 8(a) が修正なしの測度を、図 8(b) が文脈情報を入れた測度を用いて構造の抽出をした場合の結果である。それぞれの問合せの類似度にも、構造抽出時に用いた測度と同じものを用いている。

4.3.1 ベクトル補正による影響

ベクトルの補正が検索においてどのような影響を与えたかについて述べる。

問合せ条件として与えたブロック 33 に着目する。当然、これは最上位に出現すべきものであり、実際、文脈情報を用いて修正したベクトルを用いても依然として高い類似度を示している。ここで、(b) においてのみブロック 32 が上位に出現しているが、これはやや小さいブロックであった。そのため、補正情報であるブロック 33 に強く影響を受け、その結果として上位に出現した。このようなブロックが幾つかあるためか、(a) の方が (b) に比べてブロックサイズが大きめのものが上位に選ばれている。

但し、これらをより大局的な視点で捉えると、多くの場合は同じ話題のものとして統合される。この事から、候補として選ばれたものを全て呈示するのではなく、細部に高い類似度を多く持つ場合には、ある程度の抽象化されたものを呈示した方がより利用しやすい結果であると考えられる。

	階級別			累計		
類似度	$f_S(A)$	$\#S$	$\#A$	$f_S(A)$	$\#S$	$\#A$
~ 0.95	3	3	5	3	3	5
~ 0.90	6	3	8	9	6	13
~ 0.85	4	2	7	13	8	20
~ 0.80	5	1	12	18	9	32
~ 0.75	5	3	49	23	12	81
~ 0.70	16	7	69	39	19	150
~ 0.65	14	0	148	53	19	298
~ 0.60	8	0	319	61	19	617
~ 0.55	13	0	800	74	19	1417
~ 0.50	6	0	1021	80	19	2438

図 9: 任意ブロックに対する検索との比較

4.3.2 任意のブロックでの検索との比較

部分文書検索に抽出した構造を利用する事によって、検索対象の候補が非常に少なくなっている。このため、本来なら適合すべきブロックが候補には入らないという可能性がある。ここでは、考えられる全てのブロックの集合に対して検索を行ない、それと本手法での結果とを比較する。全ブロック集合での検索においても、それぞれの構造抽出の際に用いた測度で類似度を与えた。ちなみに、抽出された構造中のブロックは $2n-1$ 個であるのに対し、全ブロック集合に含まれるブロックは $n(n+1)/2$ 個である。

検索対象の文書は 25 語程の基底ブロックに分割し、問合せには 100 語程の断片文書を用いた。ここで、構造を用いた場合の結果のうち、閾値 0.7 以上の類似度を持つものに限定して比較を行なった。以下、これを S とする。抽出構造中のブロックは 391 個で、0.7 以上の類似度を持つもの (S) は 19 件であった。一方、任意のブロックは 76245 個であった。これを同じく A とする。

比較の一部を 図 9 に示す。表中の各階級毎の値の意味は次の通りである。

$f_S(A)$ は、その階級の A の部分集合について、S 中に対応するものが存在するブロックの数である。対応するとは次の関係で与える。

$$f_S(A) = |\{a \in A | \exists s \in S(a \leftrightarrow s)\}|$$

$$a \leftrightarrow s \Longleftrightarrow \frac{|a \cap s|}{|a \cup s|} > \alpha$$

α は定数であり、ここでは 0.8 とした。この式より、一つのブロックが複数のブロックと対応する事があり得る。

また、$\#S, \#A$ は、その階級に含まれるブロックの数である。尚、S に対しては階級で要素の制限をしないため $f_S(A)$ には影響しないが、参考のために記した。

上位の方では、S 中に比較的多くの適合するブロックが含まれている事が分かるが、全てを再現する事はできて

いない。一方で、類似度の閾値を下げる程、$\#A$ は急激に増えているのに対し、$\#S$ はそれほど増えない。これは組合せのオーダーによるものである。この事から、任意のブロックから検索するのに比べれば遥かに少ない計算量で、ある程度の適合ブロックを検索できるという事が考えられる。しかし、再現率は決して高くはないため、改良の余地がある。

5 転置ファイルを用いた動的、局所的な構造抽出

5.1 構造抽出範囲の局所化

本稿では、idf を繰返しステップ毎に、その時点で存在するブロック数を基準に求めていた。これは、ブロック数がその時点で識別されている話題の数を表わすものとし、idf を話題集合における語の重みとして扱っているためである。しかし、これは幾つかの問題を引き起こす。

ブロックの表現ベクトルがステップ毎の全体の状態に強く依存し、同じブロック間の類似度が繰返しステップ毎に変わってしまう。また、統合候補のペアが複数ある場合に、いずれを選択するかによってそれ以降構成される構造が変わってしまう。このため、あるステップでは 2 番目に類似度の高いペアが、次のステップではもっと低い順位になってしまう事がある。4.2 節で触れた文脈を考慮した場合に発生した構造の崩れも、これに起因しているものと思われる。

この問題は、idf としてステップ毎に固有の値を用いるのではなく、基底ブロックの集合のみの情報で与える事で改善する事ができる。これにより、初期状態から終了まで常に同じ値であるため、ステップ毎の状態依存性をなくす事ができる。ステップ毎での話題集合における語の重要度ではなく、その文書全体での語の分散状況に基づく重み付けであると捉えられるから、意味的にも問題はないと考えられる。

5.2 動的な構造抽出

idf を基底ブロック集合からのみ求める事によって、状態依存性をなくす事ができる。このため、全体の構造が分かっていなくても表現ベクトルを求める事ができる。即ち、同じ部分構造に含まれる範囲が分かっていれば、その範囲だけの局所的な構造抽出が可能になる。この事から、以下のように部分的な構造情報とそれに対する転置ファイルを保存しておくだけで、構造抽出を行なう範囲を問合せに応じて動的に局所化する事ができるようになる。

この転置ファイルは従来のものと基本的に同じものであり、それを付加する対象データとして部分構造が占める範囲の複合ブロックを用いるだけで良い。また、文書の使用頻度に応じて、保存しておく部分構造情報の粗さを調節し、記憶領域と実行測度のトレードオフを行なえるようになる。

動的な構造抽出のための手順を以下に示す (図 10)。

1. 準備段階

   (a) 対象文書に対して一度だけ構造の抽出を行なう

   (b) 抽出された構造を幾つかの部分木に分割し、その境界の位置だけを保存しておく

   (c) この部分木に対応する複合ブロックについて転置ファイルを構成する。

2. 検索手順

   (a) 境界情報とそれに付随する転置ファイルを取得する。

   (b) 各範囲 (複合ブロック) について、問合せに含まれる語を持つか調べ、その範囲に含まれる基底ブロックの頻度情報を取得する。

   (c) 各範囲において局所的な構造抽出を行なう。これによってその範囲は一つのブロックに統合される。

   (d) 以降は、手順に従って構造抽出を行なう。この時、ステップ 2c において構造抽出を行なわなかった各範囲は、途中の構造抽出を省き、その範囲全体を一つのブロックとして統合する。

   (e) 検索件数などの、問合せの付加的な条件が満たされるか、文書全体が一つのブロックに統合されるまで、ステップ 2d を繰返す。ここで、問合せの条件によっては、必ずしも全体を再構成する必要はない事を注意しておく。

この部分木への分割の戦略として、あるステップ以降に行われた結合やある閾値以下の結合を境界としたり、基底ブロック数がなるべく均一になるように分割したりといった事も考えられる。また、範囲が重ならない範囲では独立に扱える事を用いて並列化したり、マーク頻度の高いものから優先的に構造抽出を行なうというようなスケジューリングによる高速化を行なったりもできよう。これらは、基底ブロックにおける頻度情報の他に適量の範囲情報を保存するだけで可能になる。分割の荒さに関しても、その文書の利用頻度などに応じて最適化する事もできる。

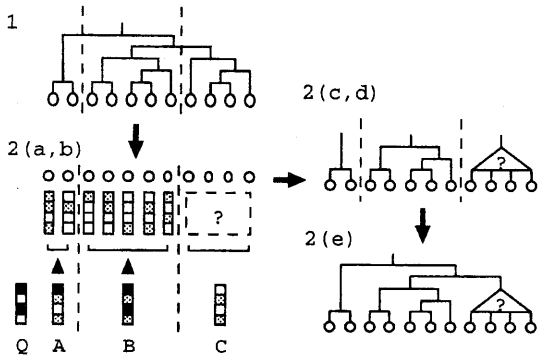


図 10: 転置ファイルを用いた動的な構造抽出手順

6 まとめ

文書の検索にあたって、文書の適合部分を特定するような検索を行なう事は、今後重要な問題となってくると考えられる。その際に、章や節などのあらかじめ明示的に与えられた構造を用いたり、文書を単純に話題の列として扱ったりするだけではなく、問合せ処理および結果出力の粒度をより柔軟に調整できる事が望ましい。

本研究ではこれを実現する一つの手法を提案し、実験を通じた評価を行なった。現状ではまだ評価対象のデータ件数も十分であるとはいえないが、おおむね良好な結果が得られた。特に、構造抽出は予想していたよりも良い結果を示した。また、転置ファイルを用いて、与えられた問合せに応じて動的かつ局所的に構造抽出を行ない、適切な粒度の問合せ結果を得るための手法を示した。

今後は、他の測度を用いた精度の測定と、最後に述べた構造抽出法の実装を施した場合の評価を行なう予定である。また、本稿では対象を詳細に調べるため、内容をよく把握している構造のはっきりとした小数の文書に対してのみ実験を行なったが、構造があらかじめ分かっていない対象や大量の文書に対して適用して、実験評価を進めていく必要がある。

また、問合せ結果をどのようにユーザに呈示するか、ブラウジングとの有機的統合の方法についても検討する必要がある。

謝辞

本研究の一部は、文部省科学研究費補助金特定領域研究「高度データベース」(08244101)、基盤研究(c)(09680321)、電気通信普及財団、ならびに筑波大学「東西言語文化の類型論」特別プロジェクトの助成による。

参考文献

[1] J.P.Callan, "Passage-Level Evidence in Document Retrieval", Proc. of SIGIR '94, ACM, pp.310-309, Dublin, Ireland, 1994.

[2] M.A.Hearst and C.Plaunt, "Subtopic Structuring for Full-Length Document Access", Proc. of SIGIR '93, ACM, pp.59-68, Pittsburg, 1993.

[3] E.Mittendorf and P.Schäuble, "Document and Passage Retrieval Based on Hidden Markov Models", Proc. of SIGIR '94, ACM, pp.318-327, Dublin, Ireland, 1994.

[4] G.Salton, J.Allan and C.Buckley, "Approaches to Passage Retrieval in Full Text Information Systems", Proc. of SIGIR '93, ACM, pp.49-58, Pittsburg, 1993.

[5] G.Salton, A.Singhal, M.Mitra and C.Buckley, "Automatic Text Decomposition Using Text Segments and Text Themes", Proc. of Hypertext '93, ACM, pp.53-65, Washington DC, 1993.

[6] G.Salton, A.Singhal, M.Mitra and C.Buckley, "Automatic Text Suructuring and Summarization", Information Processing and Management, vol.33, no.2, 1997.

[7] A.Singhal, C.Buckley and M.Mitra, "Pivoted Document Length Normalization", Proc. of SIGIR '96, ACM, pp.21-29, Zurich, Switzerland, 1996.

[8] R.Wilkinson, "Effective Retrieval of Suructured Documents", Proc. of SIGIR '94, ACM, pp.311-317, Dublin, Ireland, 1994.

[9] M.Kaszkiek and J.Zobel, "Passage Ritrirval Revisited", Proc. of SIGIR '97, ACM, pp.178-185, Philadelphia, US, 1997.