

文節共起を利用した文章検索支援

中山 拓也, 松本 裕治

奈良先端科学技術大学院大学 情報科学研究科

{takuya-n,matsu}@is.aist-nara.ac.jp

文書検索システムを用いる際、データベース中の関連するすべての文書を得ること自体は目的ではなく、ある情報を得ることが目的であることが多い。そのような目的では、キーワード検索をした結果をさらに絞り込み易くするための手段が求められる。本稿では、文節係り受け関係による共起文節を用いて検索された文書を分類することにより、ユーザが目的の情報を容易に得るための支援システムを提案する。このシステムでは、文節インデックスと出現順データ(係り受けマトリックスを含む)というものをを用いて、文書の検索および二文節間の共起関係の強さの計算を行なう。計算された共起関係の強さ情報は、係り受けマトリックスや係り受け解析器にフィードバックできるので、使うことによって性能の向上が期待できる。

Text-search Support System Using Collocational Phrases

Takuya NAKAYAMA and Yuji MATSUMOTO

Graduate School of Information Science, Nara Institute of Science and Technology

{takuya-n,matsu}@is.aist-nara.ac.jp

We often use text-search system to get some information, not to get all relevant texts from the database. For such a purpose, some kind of methods are required to make the user easy to retrieve the information from a large number of keyword-matched texts. In this report, we propose a system which classifies keyword-matched texts using collocational phrases based on grammatical dependencies and supports the user to retrieve information. To actualize such a system, we use a phrase index and phrase appearance order data (including dependency matrixes) for text-search and calculation of the dependency weight of any two phrases. The calculated weight provides feedback to update the dependency matrixes or the dependency analyzer, so that the system is expected to be improved by using.

1 はじめに

近年、電子化された文書が増加し、多くの文書データの中から必要な情報を取り出す検索技術がますます重要になってきている。特に、人手による予備的な情報付与を必要としない全文検索技術

は、WWW(World Wide Web)ページの検索などで実用的に広く利用されている。基本的にこれらは、与えたキーワードを含む文書を探してユーザに提示するものであるが、検索漏れを防ぐために、シソーラス上で意味的に近い語も検索対象として加える方法や、ベクトル空間モデルを使って

文書をあらかじめ自動分類しておき、類似文書も同時に表示させる方法も提案されている。

ここでユーザの立場で考えると、実際に検索システムを利用する場合には、次のような要求があると思われる。

- キーワードを含む文書の取得。例えば文献データベースから「全文検索」に関する文献を探す場合、「全文検索」について知りたいというより、直接的には、関連する文献をすべて知ることに目的がある。
- ある事象に関する情報の取得。例えば WWW ページの検索では、ある事象 (例えば「全文検索」) に関する情報を得ることが目的で検索することが多い。この場合、目的の情報さえ得られれば、他の検索結果は不要になる。

前者の要求に応えるためには、先に述べた検索漏れを防ぐ技術が重要になるが、後者の場合はむしろ、検索結果を絞り込む方が重要になる。大規模な WWW ページ検索システムを使った場合など、検索結果が数百から数千になることも多い。

検索を絞り込むための方法としては、AND 検索、NEAR 検索といった論理検索、係り受け情報を用いた検索などがある。また、関連文書をスコアリングして、スコア順に並べて表示することで、ユーザが目的の情報を素早く得られるようにする手法もある。この場合、スコアリングには tf*idf を用いるのが一般的である。

検索の絞り込みを支援する方法として、我々は、文節間の共起 (係り受け) 情報を用いる手法を提案する。これは、検索結果を提示するときに、検索語を含む文節と係り受け関係のある文節を同時に示すことによって、提示された検索結果の中から目的の文書を見付け易くするものである。例えば「電子メール」で検索したとき、「電子メールの⇒使い方」「電子メールを⇒読む」などの文節係り関係によって¹、検索文書をクラスタリングして提示することで、ユーザの検索を支援するということである。本稿では、このような検索支援を実現するための一つの方法を提案する。

¹本稿では、一般的な意味での係り受け関係を $A \Rightarrow B$ (A が B に係る) として表す。

2 文節情報を用いた全文検索

2.1 インデックス型全文検索

全文検索エンジンで広く使われる手法として、転置インデックスを用いたものがある。転置インデックスを用いた方法は、あらかじめ対象文書集合に対してインデックスを作成する手間が必要であるが高速な検索が実現できる。

インデックスの作り方には以下のような方法が提案されている。

- 形態素区切りによるインデックス
- N-gram 区切りによるインデックス
- 最長一致形態素区切りによるインデックス [1]

形態素区切りによる方法では、インデックス作成に形態素解析が必要になるが、比較的インデックスサイズを小さくできる。形態素解析誤りが検索結果に悪影響を与える可能性もある。それに対して N-gram 区切りでは、インデックスサイズは比較的大きくなるものの、形態素解析は必要なく、インデックス作成に要する時間が短かくて済む。最長一致形態素区切りによる方法は、文字列の各位置から始まる最長一致形態素のうち、その文を覆う最小の集合を用いてインデックス作成をするものである。形態素解析で時間のかかる処理の一つ、最適形態素パスの選択が不要になるので、形態素区切りによる方法よりも高速にインデックスを作成できる。また、辞書未定義語を除けば、インデックス項目は辞書項目のサブセットになるので、インデックスサイズも比較的小さくなる。

一般に転置インデックスの項目の検索には、prefix 検索を使うことが多いが、例えば「大学院」というインデックス項目は、「大学」や「大学院」というキーで一致するが、「学院」というキーには一致しない。そのため、suffix 検索を併用する場合もある。この場合は、項目「大学院」は「学院」で一致するものの、それでも例えば項目「奈良先端科学技術大学院大学」の場合「大学院」では一致しないという不具合は残る²。なお、文献 [1] ではインデックス項目文字列を途中位置か

²このような不具合は、登録項目の長さを出来るだけ短かくすることで回避できる。しかし、例えば「京都 / 大学」と分けて登録した場合、実際に「京都大学」で検索すると「京都 AND 大学」の検索と等価となってしまう、適合率の低下を招く。この不具合をさらに改善するために、各項目の出現位置の情報も

ら検索するための手法を提案している。これにより、上記の不具合を解消すると共に、インデックスの項目数を減らし、インデックスサイズを小さくさせることができるとしている。

転置インデックス以外のインデックス型全文検索手法の一つとして、suffix array[2] を用いた検索がある。対象文書に含まれるすべての文字へのポインタを、そのポインタの指す場所から始まる文字列についてソートしたインデックス(suffix array)を作成しておき、インデックスの二分探索によって検索を行なうものである。文書内の任意の文字列を検索可能な点が長所であるが、インデックスのサイズは検索対象文書の2倍から4倍と大きく欠点がある³。しかも、インデックス自体はポインタの配列であり、実際の検索にはオリジナルの文書が必要となる。

さて我々の方法では、文節とその係り受け関係を利用した検索結果の提示を目的としているため、インデックス⁴作成の段階で文節区切りにしておくことが望ましい。しかし、文節をインデックス項目とする場合、その項目は例えば「株式相場を」のようになるため、これを「相場」というキーワードで検索可能にするためには、インデックス項目の任意位置からの検索が必要になる⁵。そこで、文節区切りによるインデックスの検索には、suffix array を用いて、あらゆる部分文字列が検索できるようにしている。オリジナルの文書に対して suffix array を作る方法に比べると、

- インデックス項目が文節に区切られているので、検索の際に文節解析する必要がない。
- 文節インデックスを作るので、検索時にはオリジナルの文書は不要になる。その反面、文節をまたぐ文字列の検索に制限が付くなど、検索の自由度が低下する。

インデックスに保持する方法、同一文字列に対して複数の項目を登録する方法がある。しかし、この場合はインデックスサイズが巨大になってしまうという欠点がある。

³形態素に区切られるところのポインタのみを使えば、インデックスのサイズを小さくできるが、任意文字列の検索に制限がかかる。この場合も、転置インデックスのように同一項目をまとめることをしないので、インデックスサイズは比較的大きくなる。

⁴ここで作成するインデックスは、インデックス項目(文節)を含む文書を記録したものではないので、いわゆる転置インデックスではない。これについては後述する。

⁵少なくとも形態素区切りの位置からの検索が必要。

- 文節インデックスはオリジナルの文書よりもサイズが小さくなるので、作成された suffix array も文節インデックスを用いる方が小さくできる。

という違いがある。

2.2 文節解析

一般的に文節の係り受け解析器では、文節の定義を次のようにしている。

(機能接頭形態素)* 自立語形態素 (機能接尾形態素)*

ただし、複合名詞は一つの自立語形態素として扱う。扱いに違いがあるとすれば、次のような点が挙げられる。

- 形式名詞「(～する)こと」や「(～する)の(は)」を自立的とするか否か。
- 形式的動詞「(～し)得る」、「(～し)なおす」などを自立的とするか否か。
- サ変名詞(「解析」や「実現」など)とサ変動詞(「する」など)の並びがあるとき、サ変動詞を機能語的に解釈するか否か。
- 句読点を自立的に一文節とするか否か。

他にも、扱いの難しいものとして、動詞連用形の名詞的用法がある。例えば「判断を / 誤り / 失敗 / した」という文の場合、動詞連用形と名詞を名詞連続と捉えると「誤り失敗」が一文節となってしまう。しかし逆に「/ 解析の / 誤り / 訂正を / する」という文の場合は、「/ 解析の / 誤り / 訂正を / する」のように分割してしまうと不具合が生じる。

我々の実装では、動詞連用形の問題は残るものの、現在のところは上記の点についてすべて分割する(文節としてまとめない)方針で解析している。

3 係り受け情報と文節共起情報

3.1 係り受け情報による絞り込み

検索において、構文的な情報を用いることで検索の精度(適合率)を向上させようとする試みとし

て、係り受け解析を使ったもの [3] や、格フレームを使ったものがある。文献 [3] の方法は

(名詞 + 格助詞) ⇒ 用言句

の形の係り受け関係のみを使ったものであるが、「A が B する」の形の検索要求に対して、「A」と「B(する)」を含む文書を取り出し、係り受け解析の結果 $A \Rightarrow B$ の係り受け関係を持つ文を含むものに絞り込むことで、適合率の向上を目指している。例えば、「メールを読む」という検索要求に対しては、まず「メール」と「読む(及びその活用形)」で AND 検索を行ない、一致した文について係り受け解析を行ない、「メール $\Rightarrow$ 読む」の係り受け関係が成立する文を含む文書を提示する。

本稿で提案する絞り込み手法も、基本的には係り受け関係を利用するものであるが、文献 [3] の方法とは以下の点で異なる。

- 一般的な係り受け解析器の出力のように、係り先を一意に決めた係り受けデータを用いるのではなく、係り先の候補を複数保持した緩い係り受けデータを用いる。
- 前節で述べたように、あらかじめ文節区切りでインデックスを作成する。また緩い係り受け関係情報(文内文節共起情報)も同時に付与する。そのため、検索時に係り受け解析をする手間が省ける。
- 係り受け関係の検索要求を入力とするのではなく、入力単なる単語とし、その語を含む文節の係り先(または係り元)を提示することで、ユーザの絞り込み作業を容易にすることを目的とする。

3.2 文内文節共起情報

係り受け情報を用いて検索結果を絞り込む方法は有効ではあるが、幾つか問題点が残る。

まず最初に、係り受け解析の精度に関する問題が挙げられる。形態素区切りや文節区切りの精度に比べると、係り受け解析の精度はそれほど高くない。そのため、一意に決まった係り受け解析の結果のみを絞り込みに使用すると、検索結果の再現率の低下を招く恐れがある。これについては、解析器の精度を向上させても、くだけた文章を多く含む Web ページや電子メールなどは非文法的な

文章を含むことがあるので、問題を完全に解消することは難しい。

また、正確な係り受け情報が必ずしも有用とは限らないことにも注目しなければならない。例えば、「株式の 40% を取得した」という文において、「株式の」は「40% を」に係る。しかし、実際には「40% を」ということはあまり重要ではなく、「株式の $\Rightarrow$ 40% を」という係り受け情報の提示はあまり意味がない。むしろ、この文においては「株式の(…%)取得(した)」ことの方が重要である。このように、ある文節の係り先の文節(この場合「40% を」)のさらに係り先(この場合「取得」)を提示する方が、ユーザにとって有益な情報となることがある。

以上を考慮し、ここでは一意的な文節間の係り受け関係ではなく、図 1 に示すような、係り受け関係の曖昧性を残した緩い係り受け情報を利用する。係り受け関係というよりは、一文内の文節共起関係と言う方が適切である。なお、マトリック

A 社は	1	1	3	3
株式の		3	3	2
40% を			3	3
取得				3
した				

図 1: 緩い係り受け情報(文節共起情報)

スの数値は文節の係り関係の強さ(文節共起の強さ)を表し、数値の大きいものほど関係が強い。ここでは、文法的に係り得る場合(曖昧性を残している)の得点を 3、係り先の文節に係り得る場合の得点を 2、それ以外を 1 とし、条件的に重なった場合は大きい方の得点を記入している。

3.3 文節共起情報の記録

文節インデックスに係り受け情報を付加するには、各文書において、その文節の係り先の文節項目へのポイント(文節番号)を記録する方法がまず考えられる(図 2)。これは、基本的には転置インデックスであり、係り受け情報を付加した形になっている。なお、一文書中に同一文節が複数含まれることもあるが、これは一つに統合されるので注意が必要である。

文節番号	文節	文書	右文節共起情報
0001	株式の	1	(0002,3), (0003,2)
		79	 (0003,2)
		⋮	⋮
0002	10%を	1	(0003,3)
0003	売却する	1	NULL
		⋮	⋮

図 2: 文節共起情報の記録方式 1

この方法では、もしも係り先が一意に決っていれば、係り先の文節番号 (32bits のアドレス値) を一つ付加すれば良いが、文節共起関係の場合は複数の候補があるため、付加すべき文節番号の数と係り易さを表す得点のデータがそれに応じて増加する。例えば、日経新聞'90 年記事コーパスを使って調べたところ、一文中の平均文節数は 8 文節程度であったので、一文節当り平均 4 個の係り先候補が存在することになる。得点データは、0 から 3 までの 4 段階で表すとすれば 2bits で表現できるので、一文節当りおよそ $4 \times 34(\text{bits}) = 136(\text{bits}) = 17(\text{bytes})$ が文節共起情報の記録に必要な⁶。

もう一つの方法として、文節の出現順序をそのまま記録する方法が考えられる (図 3)。この方法で

文節番号	文節 INDEX	出現順データ		
	文節	文書番号	文節番号	共起得点
0001	株式の	1	0001	3 2 2
0002	10%を		0002	3 3
0003	売却		0003	3
0004	する		0004	EOS
0005	所有	2	0001	3 2 2
0006	半分を		0006	3 3
⋮	⋮		0005	3
⋮	⋮		0004	EOS

図 3: 文節共起情報の記録方式 2

⁶一文書中に同一文節が複数含まれる場合、それらに共通する係り先文節はまとめられるので、実際には 17 bytes/phrase よりも小さくなる。

は、文節インデックスとは別に、どの文節がどの順序で出現したかを記録する (これを出現順データと呼ぶ)。出現順データには、さらに文書番号の情報が付与されている。そのため、文節インデックス側には、文節を含む文書番号情報を持たせない (つまり転置インデックスにはしない)。

出現順データを実現するには、出現した文節の種類⁷の記録 (文節インデックス項目へのポインタアドレス 4bytes/phrase) と、検索方法にも依るが、それを二分検索するとすれば sorted array (4bytes/phrase) が必要になる。文節共起関係については、文節の係り先は文末向きに限られるので、その文節から何番目の文節に係るかの情報を記録するだけでよい。先ほどの日経新聞'90 年記事コーパスの場合の平均 8 文節 / 文の文書では、一文節当り平均 $4 \times 2(\text{bits}) = 1(\text{byte})$ あれば、文節共起関係を (得点も含めて) 記録できる。つまり、合計で 9 bytes/phrase の増加で、文節共起情報を付加できる。

このデータを使った検索の方法は次のようにできる。

1. 文節インデックスの項目の中から、検索語を含む文節集合 $P = \{p_1, p_2, \dots, p_n\}$ を検索する。
2. それぞれの $p \in P$ について、文節番号 (文節インデックスにおける文節項目の先頭のポインタアドレス) $a(p)$ を求める。
3. 出現順データを文節番号 $a(p)$ で検索すると、文節 p を含む文書の文書番号が求まる。
4. さらに、文節共起情報を参照すると、文節 p の係り先文節 $P' = \{p'_1, p'_2, \dots, p'_m\}$ の文節番号 $A' = \{a(p') \mid p' \in P'\}$ が求まる。 A' は、文節インデックスの文節項目へのポインタであるので、そこから係り先文節の文字列は容易に再現できる。

出現順データを用いる方法の長所は、手順 4 のところで、ある文節 p の係り先文節 $p'(p \rightarrow p')$ の他に、 p に係ってくる文節 $p''(p'' \rightarrow p)$ を調べることが出来る点にある (以降、前者を右文節共起関係、後者を左文節共起関係と呼ぶ)。また、インデックスサイズ的にも転置インデックスに付与する方法よりも有利であることは先に述べた通りで

ある。このような理由から、我々は出現順データの方法を用いている。

4 検索結果の提示と再構成

4.1 文節共起情報を用いた分類

文節係り関係(文節共起関係)によって、検索文書をクラスタリングしてユーザに提示するため、ここでは次のような分類方法をとった。

1. 前節で述べた方法で、検索語に対して、検索語を含む文節 $K = \{k_1, k_2, \dots, k_n\}$ を得る。表示は、文節の出現回数が多い順に行なう。
2. それぞれの文節 $k \in K$ について、右文節共起関係 $k \rightarrow_s d$ となる文節 d の集合を得る(ここで $k \rightarrow_s b$ は、文節 b が得点 s で文節 k の右文節共起になっていることを意味する)。また、共起文節 d の種類毎に得点(係り受けマトリックス上の数値)を合計する。

$$S_{all}(k, d) = \sum_{k \rightarrow_s d} s$$

3. 得点 $S_{all}(k, d)$ の高い順に、共起文節 d について分類 $F(\{k, d\})$ を作成する。ただし、集合 $F(\{x_1, x_2, \dots, x_m\})$ は文節 $x_1, x_2, \dots, x_m$ を含む文書集合を表す。分類集合の和が、文節 k を含む文書集合と等しくなる、つまり、分類に使用した共起文節の集合を $D'(k) \subset \{d | k \rightarrow d\}$ としたとき、

$$F(\{k\}) = \cup_{d \in D'(k)} F(\{k, d\})$$

となれば分類の作成を止める。この分類では、 $F(\{k, d_a\}) \cap F(\{k, d_b\}) \neq \emptyset$ ($d_a \neq d_b$) の可能性がある、ある文書が複数の場所に分類される場合がある。提示は、得点 $S_{all}(k, d)$ の高い順に行なう。(図4)

右共起の文節をそのまま用いて分類しようとすると、data sparseness の問題が大きいので、少なくとも文節の主辞(活用語の場合はその基本形)を使って分類する方がよい。さらにシソーラスの分類を用いることも考えられるが、これに関して

は、シソーラスの検索コストやインデックスサイズと分類精度のトレードオフになる⁷。

ここでは、ある文節 k と共起関係の強い(つまり得点 $S_{all}(k, d)$ の高い)文節 d には、 $k \Rightarrow d$ あるいは $k \Rightarrow x \Rightarrow d$ の係り受け関係がある可能性が高いと仮定している。このような仮定は事例を用いた曖昧性解消手法においては一般的であり、日本語係り受け解析について言えば、文献[4, 5, 6, 7]などの研究においても使われている。例えば文献[4]では、関係が明らかで曖昧性のない係り受けデータのみを記録し、それを用いて他の係り受けの曖昧性を解消することを試みている。我々の方法は、係り関係の強弱を表す得点を記録した曖昧性を残した係り受けデータを用いている点が違うが、基本的な考え方は同じである。

ただ、このような分類方法の最大の欠点は、分類対象データが少ない場合、分類(係り受け関係の強い文節の選定)の精度が悪くなることにある。しかし、これは実際の利用に関しては問題とならない。なぜなら、検索の結果が少ない場合は、すべての文書をチェックしたところでユーザの負担は少ないので、分類することなくユーザに提示するだけで十分なのである。つまり、検索一致文書数が多い場合は分類提示し、少ない場合は無理に分類せずに、そのまま全て提示すれば良い。

4.2 文節共起情報の修正

右文節共起関係の強さを用いて二文節 k, d 間の係り受け関係 $k \Rightarrow d$ あるいは $k \Rightarrow x \Rightarrow d$ を推定する方法を先に述べた。ここでは、推定された係り受け関係から、文節共起情報(係り受けマトリックス)を更新する方法について述べる。WWW ページのように頻繁に更新される文書を対象とする場合は、このような手続きを検索システム利用時に行なう方が実際に検索した語についてのみ情報を付与することができるので、インデックス作成時に一括して処理するよりも効率的である。

⁷条件にも依ると思われるが、シソーラスを用いた方が係り受け ($A \Rightarrow B$) の解析精度が悪くなることが文献[4]で報告されている。

⁸日経新聞'90年記事コーパスを用いて、検索語「株式」で検索した場合の例。矢印「 $-(S)->$ 」は共起関係 $\rightarrow_s$ を表す。「FILE=」の行がそれぞれ文書を表わしており、括弧内の先頭の数字が文書番号、それ以降が「株式」を含む文節以降に続く文字列を表す。

★[株式の]★

[株式の] -(3.00)-> % (...%までしか, ...%に, ...%, ...%の, ...%と, ...%から, ...%)

|FILE=(205 五二%を保有する京成グループは大きな...)

|FILE=(889 一七%を取得し第三位の株主になった...)

|FILE=(2057 四〇%を取得した。)

[株式の] -(2.23)-> 取得(取得を, 取得)

|FILE=(889 一七%を取得し第三位の株主になった...)

|FILE=(2057 四〇%を取得した。)

|FILE=(2553 過半数を取得。)

★[株式を]★

[株式を] -(1.69)-> 取得(取得)

|FILE=(586 取得する際にいったんは多額の資金を...)

|FILE=(868 取得することで八九年十二月中旬合意...)

|FILE=(3576 限度いっぱい取得しており、大手生保二社が...)

図 4: 検索結果の分類表示例⁸

4.2.1 得点 S_{all} による共起情報の更新

文節インデックスや出現順データの作成段階では、一文のみを見て文節共起情報を作成する。これを検索対象文書全体を見て計算される得点 S_{all} に応じて適切に変更すれば、 S_{all} の計算に関わった文節の文節共起情報に関して、多少の改善が期待できる。

マトリックス上の数値の変更方法には、単純には次のようなものが考えられる。

- $S_{all}(a, b)$ の大きい文節 a, b の係り受けについて、マトリックス上での得点を大きくする。
- $S_{all}(a, b)$ の小さい文節 a, b の係り受けについて、マトリックス上での得点を小さくする。

どのような方法が良いかについては、今後詳しく調査して考察する必要がある。

4.2.2 係り受け関係の非交差性

日本語文節係り受け解析をする時、一般的に良く用いられるヒューリスティックスとして、係り受け関係非交差性が挙げられる。これは、ある文節 A, B, C, D が一文内にこの順序で現れた場合、 $A \Rightarrow C$ の係り受けと $B \Rightarrow D$ の係り受け関係は共存しないというもので、例えば「土地を / 半分 / 売って / 手に入れた / 」という文では、「土地を」

と「半分」の係り先は「売って」「手に入れた」のいずれかで曖昧性があるが、「土地を $\Rightarrow$ 売って」が成立する場合には「半分 $\Rightarrow$ 手に入れた」な成立しえないことになる。この性質は、ほとんどの文で成り立つので、解析時にはかなり強い制約として用いられる。

この非交差性を、間接的係り受け関係 ($A \rightarrow C) = (A \Rightarrow B \Rightarrow C)$ について拡張して考えると、ある条件の下で成立することが分かる。まず、 A よりも左にある文節 $X_{(-A)}$ の場合、 A と C の間にある文節 $X_{(A-C)}$ との共起関係は、 $X_{(A-C)} = B$ でない限り排除される。つまり、

$$X_{(-A)} \nrightarrow X_{(A-C)} \quad (\text{if } X_{(A-C)} \neq B)$$

また、 C よりも右にある文節 $X_{(C-)}$ については、 $X_{(A-C)} = B$ でない限り制限される。つまり、

$$X_{(A-C)} \nrightarrow X_{(C-)} \quad (\text{if } X_{(A-C)} \neq B)$$

もし、4.2.1節で述べた方法などによって、ある文節に関する係り受け関係を制限できれば、この非交差の性質を利用することで、その周囲の文節に関する共起情報についても制約を加えられる可能性がある。

5 実装について

これまでに述べた方法で、文節インデックスおよび出現順データを作成するツールをC++によって実装した。文節区切り解析をするにあたっては、形態素解析器MOZ[8]をベースとした文節解析器を作成し、インデксаに組み込んでいる。実験では、文節インデックスおよび出現順データの合計サイズは全文書の2~3倍になった(図5(1))。単純な転置インデックス(図5(2)(3))に比べると大きいものの⁹、単純に suffix array を使う方法(元テキストの2~4倍のインデックスと元テキストが必要になる)に比べると小さい。

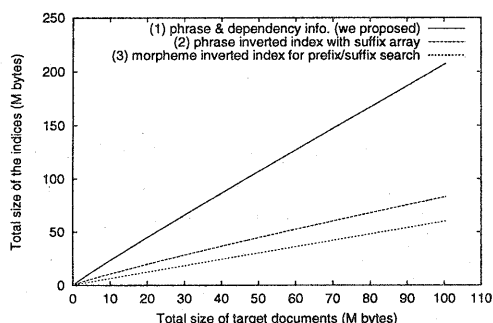


図 5: インデックスのサイズ

インデックス作成に要する時間は、現在の実装では約850kbytes/min¹⁰と遅いものの、文節区切り解析の高速化などにより、ある程度改善できる。また検索時間については、現在 Perl による試作ツールしか無いが、検索一致文節が数万件であっても数分で表示できている。これについても高速化する予定である。

6 まとめ

検索一致文書を係り受け関係をもとにした文節共起関係によって分類してユーザに提示する検索支援インタフェースを提案し、その実現方法について述べた。形態素解析/文節区切りの精度に比

⁹もちろん、図5(2)(3)の方法では係り受け情報や文節/形態素の順番は記録されていないことが、その理由である。

¹⁰SUN UltraSPARC-1 200MHz Memory 576MB 上での計測値

べると係り受け解析の精度はそれほど良くないので、それによる一意の解析結果をそのまま実用的な検索システムに利用することは難しい。本稿で述べた手法では、曖昧性を保持した係り受け情報(右文節共起情報)を元に、検索時に検索対象文書を事例ベース的に利用することで、共起関係の強さを推定する。そして、この係り受け関係の強さによって重み付けされた共起情報は、文書の分類に利用される。なお、本稿では右文節共起についてのみ言及したが、左文節共起についても同様に実現可能である。

また、推定された共起関係の強さの情報を、検索用データにフィードバックし、次の検索に役立てる方法についても述べた。さらに係り受け解析器にも情報をフィードバックさせることができれば、より良いシステムとなるが、これについては今後の課題とする。

参考文献

- [1] Noguchi, N., Kanno, Y. and Inaba, M.: New Indices for Japanese Text: A New Word-based Index of Non-segmented Text for Fast Full-text-search Systems, *Transactions of IPSJ*, Vol. 39, No. 4, pp. 1098-1107 (1998).
- [2] Manber, U. and Myers, G.: Suffix Arrays: A New Method for On-line String Searches, in *1st ACM-SIAM Symposium on Discrete Algorithms*, pp. 319-327 (1990).
- [3] 麻生和裕, 峯恒憲, 雨宮真人: 単語の係り受け構造を利用した WWW 上での日本語テキスト検索システム, 言語処理学会第3回年次大会発表論文集, pp. 253-256 (1997).
- [4] 湖武志, 米澤明憲, 田中康仁: 係り受けの曖昧性解消における共起データの効果の測定, EDR 電子化辞書利用シンポジウム論文集, pp. 11-18 日本電子化辞書研究所 (1995).
- [5] 安原宏: 共起データを用いた係り受け解析の学習効果, 言語処理学会第2回年次大会発表論文集, pp. 357-360 (1996).
- [6] 廣田啓一, 椎野努, 河合敦夫, 田添文博: 類似する用例を用いた日本語係り受け解析とその評価, 情報処理学会第53回全国大会, pp. 19-20 (1996).
- [7] 佐々木美樹, 坂本仁: 文章一括処理による係り受け関係の解析, 言語処理学会第1回年次大会発表論文集, pp. 101-104 (1995).
- [8] 山下達雄: MOZ と LimaTK の説明書 for version 0.1.4, 奈良先端科学技術大学院大学情報科学研究科自然言語処理学講座 (1999).
- [9] 長尾真: 自然言語処理, 岩波講座 ソフトウェア科学, 岩波書店 (1996).
- [10] 馬場肇: 日本語全文検索システムの構築と活用, ソフトバンク (1998).