文字成分表を用いた効率的文書ランキング検索方式

小川泰嗣†

文書検索では，検索要求に対する文書の適切さ（文書スコア）で文書を順序付けするランキング検索が有効であることが知られている．単語の切れ目が明示されない日本語を対象とした場合，文書を形態素解析して単語単位にランキングする方式と，形態素解析を用いずにn-gram単位にランキングする方式がある．しかし，前者には形態素解析に必要な大規模単語辞書の作成拡充・登録速度，後者には検索精度・索引サイズの問題がある．本論文では，両者の利点を組み合わせたハイブリッド方式を提案する．ハイブリッド方式では，索引にはn-gram単位の文字成分表を採用することで登録の高速化と索引の小型化を実現し，文書スコア計算には簡易形態素解析を利用した単語単位の手法を採用することで，高い検索精度を実現するとともに辞書管理の手間を省いた．検索システム評価用のベンチマークBMIR-J1を用いた評価により，本方式の有効性が確認できた．

An Effecient Document Ranking Retrieval Method Using n-gram-based Signature Files

YASUSHI OGAWA†

Ranking retrieval methods that rank documents in order of their relevance to a retrieval request are known to be effective. To implement ranking retrieval for Japanese documents without obvious word separator between words, there are two methods; word-based and n-gram-based. However, the word-based method has problems such as troublesome maintenance of the dictionary, and slow registration speed. The n-gram-based method has problems such as low retrieval effectiveness and large index. This paper proposes an efficient ranking retrieval method that combines both methods. As an n-gram-based signature file is used for index, our method achieves fast registration and small index. Although word-based ranking is adopted for higher retrieval effectiveness, it is free from the dictionary maintenance problem because a retrieval request is morphologically analyzed in a statistical way. We evaluated the proposed hybrid method using BMIR-J1 benchmark, and found that it was quite effective.

1. はじめに

文書検索は文書データベースの中から所望の文書をできるだけ迅速に見つけ出すための技術であり，2つの観点から研究が進められている．

第一の観点は検索精度であり，所望の文書を精度良く見つけ出す検索手法が探求されている．文書は，意味のあいまいさを含む自然言語によって記述されている．したがって，ユーザの検索要求をキーワードを論理演算子で組み合わせたBoolean形式の検索条件で記述することは困難であり，Boolean形式の検索条件を完全に満たす文書を検索する完全照合（exact match）では不十分である．これに対し，自然言語で検索要求を記述したものを検索条件とし，検索要求文に対する文書の適切さ（以下，文書スコアと呼ぶ）を検索文書ごとに計算し，文書をランキングする最適照合（best match）がある[25]．英語を対象とした文書検索では，単語の頻度情報を用いて文書スコアを計算する統計的文書ランキング法が広く研究されており，大規模評価実験によってその有効性が示されている[8]．統計的ランキング法としてはベクトル空間法，確率法等の様々なモデルが提案されているが，その多くが3種類の頻度情報——文書データベースにおけるその単語を含む文書数である文書頻度（document frequency），検索要求におけるその単語の出現数である検索要求内頻度（in-query frequency），対象文書におけるその単語の出現数である文書内頻度（in-document frequency）——を用いて文書スコアを計算している[4),25)]．

文書検索研究のもう1つの観点は処理効率である．検索システムでは，検索処理の高速化のために索引が用いられる．英語の場合，単語の頻度情報を記録し

† 株式会社リコーソフトウェア研究所
Software Research Center, RICOH Co., Ltd.

た転置ファイル形式の索引を使用するのが一般的である[25]．最近では，検索システムが対象とするデータ量は爆発的に増大しており，インターネット上のサーチエンジンでは100ギガバイトを超えるものもある[17]．このような大量データを効率的に処理するため，登録・検索の高速化，圧縮による索引の小型化等の研究が進んでいる[4),28)]．

本論文は，日本語を対象とした統計的文書ランキング法を研究対象とする．日本語を対象とした場合，英語等と異なって，単語がスペース等で明示されないことが問題となる．日本語を対象とした文書ランキング法としては，以下の2つが考えられる．

(1) 単語単位のランキング
日本語文を形態素解析を用いて単語に分割し，単語を処理単位としてランキングを行う．この方式では，英語等のランキング法をそのまま用いることができる．しかし，形態素解析には一般に数万から数十万語の大規模単語辞書が必要であり，その作成に膨大な時間・コストがかかるとともに，新語追加などの継続的拡充も不可欠である．また，辞書更新時には検索用索引の作り直しが必要である[15),18)]．さらに，形態素解析による登録速度の低下，解析誤りによる検索精度の低下も問題である．

(2) n-gram単位のランキング
n-gramとはn文字から成る部分文字列のことであり，この方式ではn-gramを処理単位にランキングを行う☆．この方式では，単語分割は不要なので，前述した形態素解析の問題は発生しない．しかし，単語の持つ文法的・意味的な情報が失われることによる検索精度低下の問題がある．また，n-gram単位の索引はサイズが大きいことも問題である[1),26)]．

このように，両者とも高速登録，小型の索引，高い検索精度，低い運用コストといった要件を同時に満たすことはできなかった．

本論文では，これら要件を同時に満たすものとして，n-gram単位の索引と単語単位のランキングを組み合わせたハイブリッド方式を提案する[19),20)]．n-gram単位の索引としては，索引サイズが小さいことに特徴のある文字成分表[15),18)]を採用した．文字成分表を採用したことで，索引作成に関する形態素解析の問題を回避できる．また，単語単位のランキングを採用したことで，高い検索精度が期待できる．

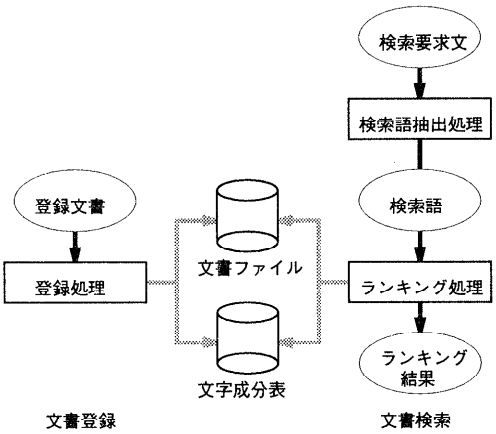

図1 処理フロー
Fig. 1 Processing flow.

ハイブリッド方式における登録・検索の処理手順を**図1**に示す．ここで，実線矢印は処理の流れ，点線矢印はデータの流れ（参照）を表している．文書登録では，文書を文書ファイルに格納すると同時に，文字成分表を更新する．文書検索では，まず検索要求文を単語分割して適切な複数の検索語を抽出し，次にこれら検索語を用いて文書スコアを計算してランキングを行う．

しかし，この方式を実現するには2つの問題点を解決しなければならない．1つは検索語抽出である．文字成分表を採用したことで文書登録時に形態素解析は必要なくなったが，検索要求文から検索語を抽出するため，文書検索時に形態素解析は必要である．この処理に従来型の形態素解析を用いたのでは前述した問題点を回避できないので，簡易形態素解析と統計情報に基づく分析を組み合わせた方式を考案した．この検索語抽出処理について2章で説明する．

もう1つの問題点は単語単位のランキングのための頻度情報獲得である．統計的ランキングに必要な検索語の頻度情報を文字成分表から獲得できないので，検索時に対象文書から頻度情報を得る必要がある．しかし，3章に示すように，このためには対象文書を二次記憶から読み出して検索語で文字列照合しなければならないので，検索時間が著しく増大する．そこで，文字成分表からランキングの補助になる情報を引き出し，その情報を用いて文書スコアの計算対象を削減することで，ランキング処理を高速化する．本論文ではこの手法を逐次確定法と呼び，4章で詳しく説明する．

さらに，情報処理学会が作成配付している情報検索用ベンチマークによる評価実験を5章で報告する．

☆ 文字単位のランキング法[5)]も研究されているが，これは$n=1$のn-gram方式ととらえることができるので，本論文では文字単位もn-gram単位に含めることとする．

6章では，ハイブリッド方式と，既存の単語あるいはn-gram単位による方式との比較を行う．

2. 簡易形態素解析に基づく検索語抽出処理

検索語抽出処理は，自然言語で記述された検索要求文を解析し，適切な検索語を選定することである．この章では，辞書の作成更新の問題がない，簡易形態素解析を利用した抽出方式を説明する．

2.1 処理手順

日本語の形態素解析法としては，漢字・ひらがな・カタカナ等の複数の文字種の使い分けに着目することで，機能語辞書のみを用いる簡易解析法がある[5),11),24)]．機能語辞書は語彙的に閉じており，小規模であるので，辞書に関連した問題は発生しない．しかし，従来の簡易解析法では，複合語をその構成単語に分割できない．複合語と同じ内容はその構成単語を含む句・文によっても記述できるので，検索処理では複合語を構成単語に分割する必要がある．

そこで，以下のような二段階処理を考案し，大規模辞書を用いることなく構成単語レベルの検索語抽出を実現した．

(1) 検索要求文を簡易形態素解析で複合語に分割し，助詞・助動詞など機能語を除いた残りを検索語候補とする．
簡易形態素解析にはQJP[11)]を使用した．QJPの解析辞書は機能語と平仮名語など少数の例外語から構成されており，約5000語と小規模である．

(2) 検索語候補に統計的分割処理を施し，分割結果を検索語とする．
ここで用いる分割法（詳細は次節で述べる）は文字の統計情報を用いるので，必要なデータはわずかである．

2.2 文字の統計情報に基づく複合語分割[19),20)]

提案方式では，二文字組の間が単語の切れ目である確率（分割点確率と呼ぶ）を利用する．パラメータとして与えられる閾値 P（複合語分割閾値と呼ぶ）以上の分割点確率を有する二文字組を単語の切れ目とすることとし，検索語候補を分割する．しかし，この方式を単純に実装するには，任意の二文字組に関する分割点確率をあらかじめ用意しておく必要がある．そのため，データの収集に大量のコーパスが必要，分割時に必要なデータ量が膨大等の問題が発生する．

そこで，文字の統計情報に基づいて分割点確率を計算する方式を考案した．文字の統計情報としては，文字ごとの単語の先頭・末尾に出現しやすさを数値化した単語頭確率・単語尾確率というものを利用し，二文字組の分割点確率を「前側文字の単語尾確率と後側文字の単語頭確率の積である」という仮定に基づいて算出する．単語頭確率・単語尾確率は，ある文字の出現頻度に対する，その文字が単語の先頭に出現した出現頻度およびその文字が単語の末尾に出現した出現頻度の比率である．

分割処理の例を示す．「政」の単語尾確率が0.20，「治」の単語頭確率が0.09であれば，「政治」に対する分割点確率は $0.20 \times 0.09 = 0.018$ となる．「政治改革」のすべての文字間の分割点確率が，“政0.018治0.163改0.039革”となった場合，$P = 0.1$ とすることで「政治」「改革」と正しく構成単語に分割できる．

単語頭確率・単語尾確率は，あらかじめ辞書化しておく必要がある．辞書化のためには，EDR（日本電子化辞書研究所）の日本語コーパスやRWCP（新情報処理開発機構）テキストデータベース[23)]等の形態素済みコーパスにおける上記頻度を調べ，確率値を計算すればよい．より簡便には，テキスト中の同一文字種の連続部分を単語と見立てて，収集することもできる☆．

二文字組ごとの分割確率を直接用いる方式と比較して，単語頭確率・単語尾確率は文字ごとに用意すればよいので，辞書化しておくべきデータ量は少ない☆☆．また，文字の異なり数は約7000と少ないので，異なり数がその二乗である二文字組の場合と比較して，少量のコーパスからも必要なデータを収集可能である．分割精度に関しては，提案手法は文字の単語頭確率・単語尾確率という単純な統計情報しか用いていないので，大規模辞書に基づく形態素解析を用いた場合や，より詳細な統計情報に基づく従来方式[14),27)]よりも劣るであろう．しかし，文字成分表では，単語でない任意の文字列について検索可能なので，単語単位の索引を用いた場合と比較して分割精度の低下が検索精度に及ぼす影響は小さいと考えられる．分割精度に影響については，6章の検索精度の項目を参照されたい．

3. 文字成分表を用いた文書ランキングの問題点

3.1 文字成分表

日本においては，単語分割の必要がないn-gram索引が広く研究開発されている[15)]．索引の代表的ファイ

☆ 5章の評価実験では，検索対象に用いた日本経済新聞CDROM 93年版に対し，この方法で収集した値を使用した．

☆☆ JIS漢字コード（X0208）の全6353文字について単語頭確率・単語尾確率を4バイトで持つ場合，必要なデータ量は約50Kバイトである．

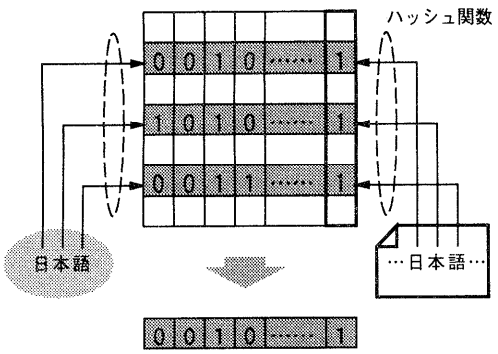

図2 文字成分表
Fig. 2 N-gram-based signature file.

ル形式には，索引単位である単語あるいはn-gramごとに出現した文書を記録する転置ファイルと，索引単位の出現をハッシュ関数を用いて重ね合わせて記録するシグニチャーファイルがあり[4),28)]，文字成分表はシグニチャーファイル形式のn-gram索引である．

$n=1$の場合の文字成分表を**図2**に示す．中央のマトリックスが文字成分表であり，各行がn-gram（この場合は文字）にハッシュ関数を適用した値，各列が文書に相当する．文書登録時には，文書中に存在する文字にハッシュ関数を適用してビット位置を計算し，そのビットを立てる．文書検索時には，検索語中に存在する文字から同様にビット位置を計算し，そのビットすべてが立っている文書を検索結果とする．この方式では，検索語を構成する文字がバラバラに出現している文書も検索され，誤検索が発生する．誤検索を除去するには，各検索文書を二次記憶から読み出し，検索語で文字列照合することで，検索語が実際に出現しているか確認しなければならない．この処理を誤検索除去と呼ぶ．

n-gram索引では，nの指数オーダで索引サイズが増大するので，通常$n=1$，2等の値が使用される[1),21),26)]．$n=2$の場合，転置ファイル形式では索引サイズがもと文書の2倍以上ときわめて大きいのに対し[1),26)]，文字成分表では1/3〜1/2と小型である．したがって，誤検索という問題点があるにもかかわらず後者も広く使用されている[6),7),9),21)]．

3.2 文字成分表のランキングへの適用の問題点

文字成分表をランキングに採用した場合，ランキングに必要な検索語の文書内頻度・文書頻度が記録されていないことが問題となる[19),20)]．

文書内頻度は文書における検索語の出現頻度であるので，対象文書を文字列照合して，検索語の出現回数を計数する必要がある．ランキング検索においては，検索語を1つでも含む文書がランキングの対象となるので，それら文書すべてについて文書内頻度を計数しなければならない．一方，文書頻度はその単語を含む文書数であるので，文字成分表を用いて行った検索結果の文書数を用いればよい．正確な文書頻度を得るには文字成分表検索にともなう誤検索を除去しなければならないが，文書内頻度計数結果が0であった文書は誤検索と判断できるので，文書内頻度計数を行えば誤検索除去のために新たな処理は不要である．

実際には，複数個の検索語がある場合にも同一文書を複数回処理することがないように，以下の手順でランキングを行う．

(1) 文字成分表検索により少なくとも1つの検索語を含む文書を特定し，ランキング候補とする．
(2) すべてのランキング候補に逐次的にアクセスし，各候補における各検索語の出現回数（文書内頻度）を計数・記録する．全候補を処理し終えた時点で，各検索語の文書頻度も明らかになる．
(3) 獲得された文書頻度・文書内頻度を用いて，ランキング候補ごとに文書スコアを計算する．
(4) ランキング候補を文書スコア順にソートし，検索結果とする．

以下，この処理手順を「一括確定法」と呼ぶ．

ステップ(2)では，ファイルアクセス・文字列照合というコストの高い処理をすべてのランキング候補について実施する．ランキング候補数はデータベースの登録文書数の50%以上にもなるという実験結果[2),16)]を考慮すると，ステップ(2)はかなりの処理量となり，検索時間を大幅に増大させる原因となる．したがって，検索速度の点から一括確定法は実用的ではない．

4. 逐次確定法による効率的ランキング処理

前章の議論から，ランキング検索の時間短縮には，ファイルアクセス・文字列照合の対象文書を削減することが不可欠であると分かった．この章では，逐次確定法という手法を採用することで，これらの処理対象文書を削減し，検索を高速化できることを示す．

4.1 逐次確定法

ランキング検索では，ユーザが実際に参照するのは比較的少数の上位にランキングされた文書のみであるので，比較的少数の上位ランキングの文書を高速に決定できればユーザを満足させることができる[3),29)]．逐次確定法では，文書スコアの上限（upper bound: 文書スコアより大きいという制約を満たすスコアの推定値）を利用することで上位ランキングの文書を高速に決定する[13),29)]．

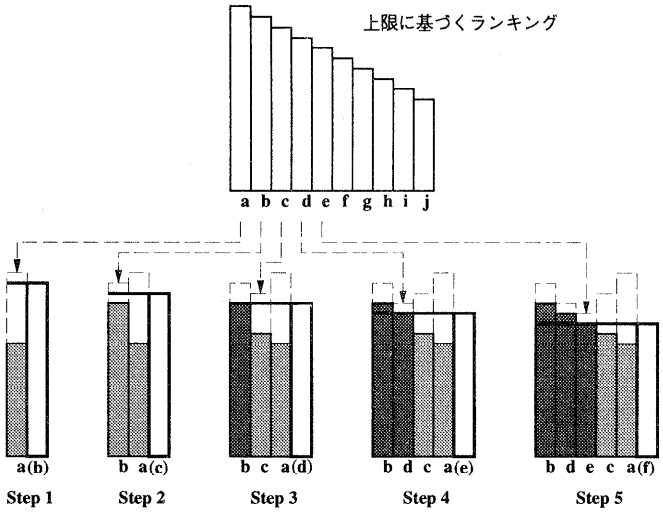

図3 逐次確定法による上位ランキング文書の決定
Fig. 3 Incremental ranking reevaluation process.

検索要求 Q に対する文書 D_i のスコア $r(D_i,Q)$ のとりうる値の上限を $s(D_i,Q)$ と書く．このとき，"$s(D_j,Q) \geq r(D_j,Q)$" なので，2つの異なる文書 D_i, D_j に対して以下の関係式が成立する．

$$r(D_i,Q) \geq s(D_j,Q) \Rightarrow r(D_i,Q) \geq r(D_j,Q) \quad (1)$$

ランキングの上位文書を決定するにあたっては，すべてのランキング候補について上限を計算し，上限の大きい順にランキング候補のスコア計算を行う[13),19),20)]．上限の大きい順に l 個の文書のスコアを計算した時点で，残りの文書の中では $l+1$ 位の文書が最も大きな上限を持っている．したがって，上限によるランキングの第 l 位の文書の ID を $o(l)$ と書くと，式 (1) から文書集合 $R_l = \{D_i | r(D_i,Q) \geq s(D_{o(l+1)},Q)\}$ に属する文書の最終ランキングを決定できることが分かる．ランキングの上位 k 文書を決定するには，"$|R_l| \geq k$" (これを終了条件と呼ぶ) を満たした時点で処理を終了すればよい．ここで，$|X|$ は集合 X の要素数とする．

逐次確定法によるランキング決定の様子を図 **3** に示す．文書はアルファベット (a, b, ···) で識別するものとし，上段が上限によるランキング結果を示す．白抜き矩形の高さで上限，網掛け矩形の高さで文書スコアを表している．処理は下段の左から右と進み，ステップ l では l 番目の文書のスコアを計算し，$l+1$ 番目の上限との比較によってランキングが決定できるかの判定を行う．下段において，濃く網掛けになっているのがランキングの確定した文書であり，たとえばステップ5では b, d, e の3文書のランキングが確定している．$k=3$ であれば，ここで処理を終了できる．

4.2 逐次確定法の文字成分表への適用

逐次確定法を文字成分表を適用した場合，ランキングの処理手順は以下のようになる．

(1) プレランキングフェーズ
文字成分表を用いて，検索語を少なくとも1つ含む文書を特定し，ランキング候補とする．さらに，すべてのランキング候補について文書スコアの上限を計算し，ランキング候補を上限順にソートする．

(2) 逐次確定フェーズ
プレランキング順に候補文書にアクセスし，その文書における検索語の文書内頻度 (出現回数) を計数し，それを用いて文書スコアを計算する．終了条件を満たしたら，処理を終了する．

上記手順ではランキング候補すべてにアクセスするわけではないので，文字成分表検索で発生する誤検索が完全には除去されない．したがって，ランキング結果に対する誤検索の影響について検討する必要がある．誤検索の影響には，以下に示す2つがある[19),20)]．

1つは文書スコア計算への影響である．逐次確定法では，文字成分表による検索結果の文書数を文書頻度として使用する．ところが，誤検索が残っているため，この値は本来の本来の文書頻度より大きく，文書スコアに影響する．しかし，文書スコア計算では後述のように文書頻度の log をとった値を使用しているので，文字成分表のパラメータ[21)]を調整して誤検索率を低くおさえることで，誤検索による文書スコアの相違はわずかとなる．文書スコアが多少変化してもランキングに影響しない場合もあるので，誤検索を含んだ文書

数を使用してもランキング結果への影響は小さいと考えられる☆.

もう1つの問題は上限計算への影響である.上限の計算はプレランキングフェーズで行うので,文字成分表のみを用いて上限は計算しなければならない.そこで,文字成分表から知ることのできる文書に検索語が出現しているか否かに基づいて上限を計算する.しかし,誤検索のために,実際には検索語が出現していないにもかかわらず出現していると判断され,誤検索がない場合よりも上限が大きく計算されることがある.いま,誤検索を含む結果から計算された上限を $s'(D_i,Q)$, $R'_l=\{D_i|r(D_i,Q)\geq s'(D_{o(l+1)},Q)\}$ と書く.このとき,"$s'(D_i,Q)\geq s(D_i,Q)$" であるので,"$|R'_l|\leq|R_l|$" となる.その結果,処理ステップが増加し,検索が遅くなる可能性がある.しかし,誤検索率が低いことから $|R'_l|$ と $|R_l|$ の差は小さく,誤検索による速度低下はわずかと考えられる.

以上の議論から,誤検索がランキング結果に若干影響する懸念はあるが,全候補を誤検索除去する一括確定法と比べて検索時間の大幅な短縮が見込まれる.したがって,逐次確定法を文字成分表に適用して得られるメリットは大きい.

4.3 ランキングモデル

文書スコアの計算法を規定するランキングモデルの選択では,高い検索精度を達成できることとともに,逐次確定法を適用できることが条件となる.逐次確定法の観点からは,文書内頻度の正規化方法が選択のポイントとなる.検索精度向上には文書内頻度を文書の長さで正規化することが有効であり,対象文書に含まれるすべての単語の頻度情報によって正規化する方法が一般的である[4),25)].しかし,この方式では各文書における全単語が明らかにされなければならないので,文字成分表とは相容れない.これに対し,確率モデルの1つである Robertson モデル[22)]は,他の単語の頻度とは無関係に検索語の文書内頻度を正規化しており,逐次確定法向きである.さらに,英語文書を対象とした検索システムの評価コンテスト[8)]で高い成績を残していることから,Robertson モデルを採用した.

ただし,Robertson モデルそのものではなく,文書内頻度の正規化方式を改良して検索精度向上をはかった[20)].オリジナルの Robertson モデルでは,文書スコアを下式で計算する.

$$r(D_j;Q)=\sum_i\left\{\log(\frac{N}{df_i})\cdot\frac{qf_i}{Kq+qf_i}\cdot\frac{tf_{ij}}{Kd\frac{L_j}{L_{ave}}+tf_{ij}}\right\}\quad(2)$$

ここで,df_i は単語 T_i の文書頻度,qf_i は検索要求 Q における要求内頻度,tf_{ij} は文書 D_j における文書内頻度,N はデータベース中の文書数である.また,Kq, Kd は要求内頻度,文書内出現頻度の正規化パラメータ,L_j, L_{ave} は対象文書の文書長,および文書データベース全体における平均文書長である.

式(2)から分かるように,文書内頻度は文書長に比例した係数で正規化している.この正規化方式は文書の内容を記述するのに重要な単語は文書の長さに応じて使用されることを前提としているが,長い文書では複数の主題が述べられることもあるので[10)],検索要求に一致する内容が記述されている文書であっても,長さに比例した回数だけ検索語が使用されるとは期待できない.そこで,正規化係数に対する文書長の影響をパラメータで調整することとした☆☆.

改良モデルでは,文書スコアを以式で計算する.

$$r(D_j;Q)=\sum_i\left\{\log(\frac{N}{df'_i})\cdot\frac{qf_i}{Kq+qf_i}\cdot\frac{tf_{ij}}{Kd(\lambda\frac{L_j}{L_{ave}}+(1-\lambda))+tf_{ij}}\right\}\quad(3)$$

ここで,λ が今回導入した文書長の影響を制御するためパラメータである.また,第1項に現れる df'_i は文字成分表による検索文書数で,前節で述べた理由から本来の文書頻度 df_i の代わりに使用するものである.

文書スコアの上限は以式で計算する.

$$s(D_j;Q)=\sum_i\log(\frac{N}{df'_i})\cdot\frac{qf_i}{Kq+qf_i}\cdot\delta_{ij}\quad(4)$$

ここで,δ_{ij} は検索語 T_i による文字成分表検索結果を表しており,文書 D_j が結果に含まれているか否かを1/0で示す.

5. 評 価

5.1 評価方法

本論文で提案したハイブリッド方式を検索精度と検索時間の両面から評価した.検索精度の評価には,(株)

☆ ここでは,正しい文書頻度から計算される文書スコアを中心に議論しているが,実際には文書スコア計算式自体が様々な仮説に基づいて導き出されたものであり,絶対的なものではない[4),25)].

☆☆ 要求内頻度もこれと同様に検索要求の長さを正規化することも考えられるが,1回の検索においては検索要求の長さは変化しないので,正規化を導入してもランキング結果は変わらない.そこで,検索要求長による正規化は行わなかった.

表1 検索対象文書と検索要求に関する統計
Table 1 Statistics of target documents and requests.

	BMIR-J1 検索要求	BMIR-J1 対象文書	日経 CD-ROM 対象文書
件数	47☆	600	163110
平均文字数	11	703	494
最小文字数	2	102	11
最大文字数	28	3802	16545
合計サイズ	—	872 KB	159 MB

日本経済新聞の協力によって,(社) 情報処理学会·データベースシステム研究会·情報検索システム評価用データベース構築ワーキンググループが, 1993 年 9 月 1 日から 12 月 31 日の日本経済新聞記事を基に構築した情報検索評価用データベース (テスト版) BMIR-J1 を利用した[12]. BMIR-J1 の規模は**表 1** に示すとおりである. なお, 文書にはあらかじめキーワード等の補足情報が付与されているが, 今回の実験ではタイトルと本文のみを対象文書として使用した.

検索精度の評価には, 再現率·適合率を用いた. 再現率は正解文書を洩れなく検索できる能力, 適合率は正解でない文書を検索しない能力を表すもので, 以下の式で計算される.

$$\text{再現率} = \frac{\text{検索された正解文書数}}{\text{正解文書数}} \tag{5}$$

$$\text{適合率} = \frac{\text{検索された正解文書数}}{\text{検索文書数}} \tag{6}$$

検索要求ごとに, ランキングごとの再現率·適合率から再現率が $0.0, 0.1, \cdots, 1.0$ における適合率を求め, さらに全検索要求での平均値を求めて評価指標とした[25]. この方法では, 再現率が 0.0, 0.1 等の低い場合の適合率がランキング上位, 再現率が 0.9, 1.0 等の高い場合の適合率がランキング下位の検索精度を表す. さらに, 必要に応じてすべての再現率での適合率の平均値 (これを平均適合率[25]と呼ぶ) も求めた.

検索時間の評価には, 日本経済新聞 CD-ROM 93 年版を利用した. これは, BMIR-J1 の対象文書数が 600 と処理時間の測定には少ないからである. この CD-ROM の規模も表 1 に示す. 検索要求は BMIR-J1 のものをそのまま使用した. 評価指標には, ディスクキャッシュが空の状態から, ランキングの上位 $k(=0, 10, 20, 50, 100)$ 文書を得るのに要した時間を測定し, 全検索要求での平均値を求めた. ここで, $k=0$ はプレランキングの処理時間を意味する. なお, 評価には Sun SPARCStation 20 model 70 を用いた.

☆ BMIR-J1 に含まれる 60 件の検索要求のうち, BMIR-J1 が設定している正解記事数の制約条件を満たすものの数.

表2 評価実験に用いたパラメータ一覧
Table 2 Parameters used in the experiments.

パラメータ		使用した値
複合語分割閾値	P	0.00, 0.05, 0.10, 0.15, 0.20, 1.00
文書内頻度正規化係数	Kd	0.00, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0
文書長係数	λ	0.0, 0.2, 0.4, 0.6, 0.8, 1.0

表3 評価結果
Table 3 Average precisions and response times.

優先条件	平均適合率		検索時間 (sec)	
	逐次確定	一括確定	逐次確定	一括確定
検索速度	0.2752	0.2756	6.98	66.4
検索精度	0.3618	0.3621	31.6	206.0

5.2 最適なパラメータ設定の決定

表 2 は, 評価で用いたパラメータの一覧表である☆☆. 各パラメータの値を表 2 に示したように変化させ, 検索精度·検索時間を測定した. すべての測定結果をここに示すことはできないので, 最短の検索時間および最高の検索精度を実現したパラメータの組合せにおける測定結果 (検索精度は平均適合率, 検索時間は $k=20$ での値) を**表 3** に示す.

検索精度優先の場合に一括確定法と逐次確定法を比較すると, 検索時間は約 1/6 に短縮されている. 検索精度が異なるのは, 一括確定法では正しい文書頻度を用いて文書スコアを計算しているのに対し, 逐次確定法では誤検索を含んだ検索文書数を用いて計算しているからである. しかし, 両者の差は 0.083%ときわめて小さく, 誤検索の影響はほとんど無視できる. 検索速度優先の場合でも検索精度は 0.15%しか低下していないのに対し, 検索時間は約 1/9 と大幅に短縮されている. これらの結果から, 逐次確定法の有効性が確認できた.

次に, 検索速度優先と検索精度優先を比較する. パラメータの組合せを (P, Kd, λ) と書くと, 検索速度優先では $(1.00, 0.0, 0.0)$, 検索精度優先では $(0.05, 0.5, 0.2)$ だった. 逐次確定法の場合, パラメータの調整により, 検索精度は 31.5%向上したが, 検索時間は約 4.5 倍になった. これは, 以下の理由による.

(1) $P=0.05$ とすることで複合語が適切に分割され, 検索洩れ·誤検索がなくなったので, 検索精度は向上した. しかし, 検索語が増大し, それにともないランキング候補数も増大したのでプレランキング時間が増大し, 検索時間が増大した.

☆☆ 式 (3) には検索要求内頻度の正規化係数 Kq がある. しかし, BMIR-J1 の検索要求が比較的短く, Kq は検索精度に影響しないので, 今回の評価実験では $Kq=0$ に固定した.

(2) $Kd = 0.5$ とすることで検索語の文書内頻度がランキングに適切に反映されるようになったので，検索精度は向上した．しかし，上限と文書スコアの差が増大するようになったので，上位文書の決定に要するステップが増大し，検索時間が増大した．

(3) $\lambda = 0.2$ とすることで文書の長さがランキングに適切に反映されるようになったので，検索精度は向上した．(2) と同様に，上限と文書スコアの差が増大するようになったので，検索時間が増大した．

なお，検索時に検索精度か検索速度のどちらを優先するかは，ユーザの検索意図や対象データベースの規模に応じて異なることが多い．したがって，上述のように，検索時に各パラメータを調整することでランキング検索の特性を調整できるハイブリッド方式は，ユーザにとって使いやすい検索方式といえる．

5.3 各パラメータの影響

前節では最適なパラメータ組合せについて調べたが，この節では，検索精度優先のパラメータ組合せから特定のパラメータだけを変更することで，パラメータの影響を個別に調べる．

5.3.1 複合語分割閾値の影響

$Kd = 0.5$, $\lambda = 0.2$ に固定した場合の複合語分割閾値 P の検索精度への影響を図 **4** に示す．$P = 1.00$（複合語分割を用いない場合）と比較し，P が小さくなるに従って適合率は大きく向上しており，複合語分割の有効性が確認できる．P が 0.05 から 0.00 になる際に例外的に適合率が若干低下しているのは，複合語がほとんど単一文字に分割されてしまうため，不適切な文書が上位にランキングされるからである．

図 **5** は検索時間の測定結果である．P が小さくなるに従って検索時間は増大しているが，これは，(1) P が小さくなるほど検索語数が増大し，文字成分表検索自体の処理時間がかかること，(2) 検索語数の増大とともにランキング候補数が増大するため，上限の計算時間が増加すること，の 2 つの理由がある．実際，$P = 0.00$, 0.05, 0.10, 1.00 における，検索要求から抽出される平均検索語数は 7.72, 4.89, 3.76, 2.75 であり，平均ランキング候補数は 120712, 80744, 48343, 23205 であった．

5.3.2 文書内頻度正規化係数の影響

$P = 0.05$, $\lambda = 0.2$ とした場合の文書内頻度正規化係数 Kd の検索精度への影響を図 **6** に示す．$Kd = 0.0$（文書内頻度は無視して検索語の有無のみから文書スコアを計算する場合）において検索精度は最低であり，ランキングにおける文書内頻度の重要性が確認できる．しかし，Kd を大きくすると検索精度は向上するが，5.0 では低下しており，文書内頻度のランキング決定への寄与を大きくしすぎては良くないことが分かる．

検索時間を図 **7** に示す．Kd を大きくするに従って検索時間が増大しているが，Kd の影響は P の場合とは異なっている．すなわち，図 7 から分かるように Kd はプレランキングフェーズの処理時間である $k = 0$ の検索時間にはまったく影響せず，$k > 0$ における検索時間のみを増大させている．これは，P が検索語数を調整するパラメータなのでプレランキングフェーズに影響するのに対し，Kd は文書スコア計算

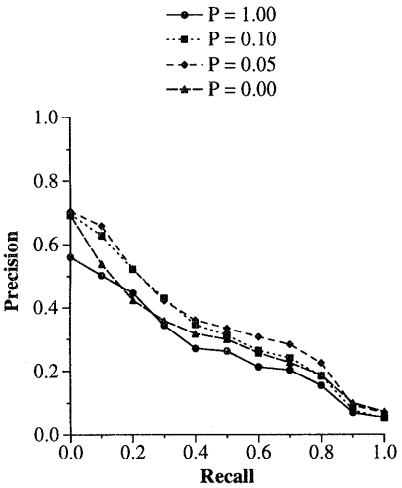

図 **4** 複合語分割閾値 P の検索精度への影響
Fig. 4 Effects of P to recall vs. precision.

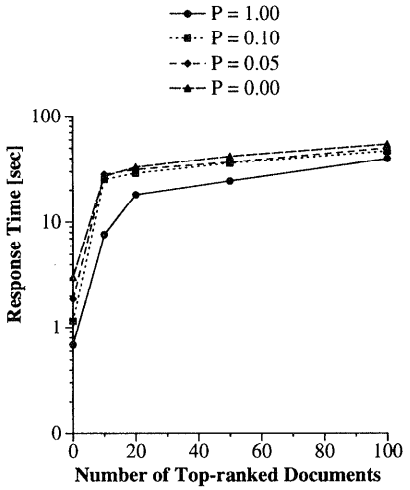

図 **5** 複合語分割閾値 P の検索時間への影響
Fig. 5 Effects of P to response time.

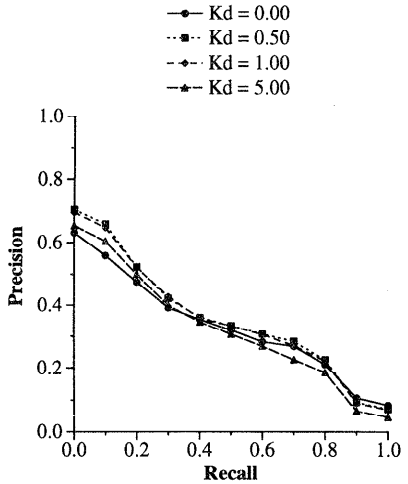

図6 文書内頻度正規化係数 Kd の検索精度への影響
Fig. 6 Effects of Kd to recall vs. precision.

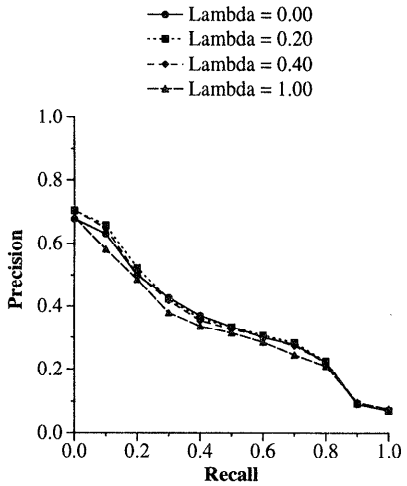

図8 文書長係数 λ の検索精度への影響
Fig. 8 Effects of λ to recall vs. precision.

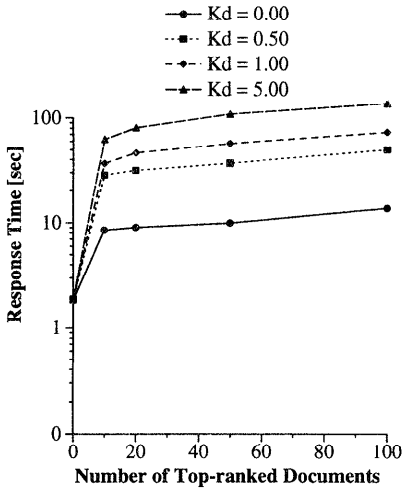

図7 文書内頻度正規化係数 Kd の検索時間への影響
Fig. 7 Effects of Kd to response time.

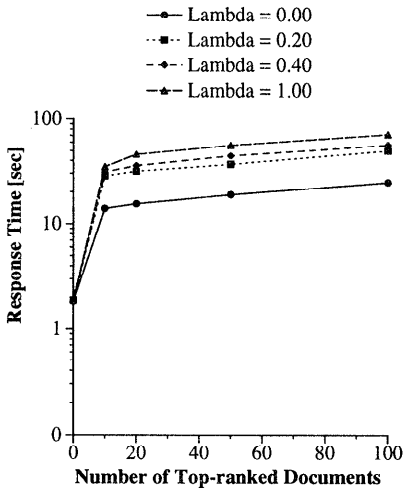

図9 文書長係数 λ の検索時間への影響
Fig. 9 Effects of λ to response time.

式 (3) に表れるパラメータなので，確定処理だけに影響するからである．Kd が大きくなると式 (3) の第三項が小さくなり，上限と文書スコアの差が増大する．その結果，上位 k 文書を確定するために文書スコアを計算しなければならない文書数も増加し，確定処理時間が増大する．実際，たとえば $k = 20$ における文書スコアを計算した文書数は，$Kd = 0.0$, 0.5, 1.0, 5.0 において 1791, 2208, 3198, 16255 のように増加していた．

5.3.3 文書長係数の影響

$P = 0.05$, $Kd = 0.5$ とした場合の文書長係数 λ の検索精度への影響を図 **8** に示す．$\lambda = 0$ （文書内頻度の正規化において文書長を考慮しない場合）と比較して，ある程度文書長を文書スコアに反映させる $\lambda = 0.2$ では検索精度は改善され，最高値を示した．しかし，λ を大きくして文書長を強く反映させると $\lambda = 0$ よりも悪化している．この結果は 4.3 節に示した考察と一致しており，改良した文書スコア計算式の正規化法の有効性が確認された．

検索速度への影響を図 **9** に示す．λ を大きくすると検索時間も増大しているが，これは以下の理由による．文書内頻度は，前述のように文書長に比例することはないにせよ，文書長に応じて長くなる傾向がある．λ を大きくして文書長を文書スコアに反映させる場合，

文書長が大きいほど，文書内頻度のスコアへの貢献が小さくなり，同じ文書内頻度であっても文書スコアは小さくなる．すなわち，文書スコアと上限の差が拡大するので，Kd の場合と同様にして検索時間が増大する．実際，$k = 20$ における文書スコアを計算した文書数は，$\lambda = 0.0$，0.2，0.4，1.0 において 1932，2208，2490，3164 のように増加していた．

6. 考　　察

この章では，ハイブリッド方式と，既存の文書検索法である転置ファイルを用いた単語単位/n-gram 単位の方式（以下では単語方式/n-gram 方式と呼ぶ）を比較する．

(1) 登録速度

ハイブリッド方式では，形態素解析を必要としない文字成分表を採用している．文字成分表への文書の登録時間は，日経 CD-ROM の記事 159 MB を対象とした場合 4016.4 sec（= 1.12 hour）であった．単語方式については，転置ファイルモジュールが手元にないため，登録処理に必要な対象文書の形態素解析時間のみを測定した．パブリックドメインで提供されている大規模辞書を用いた高速形態素解析系である「茶筅」☆を用いて解析に要した時間は，14435 sec（= 4.01 hour）であった．これは，転置ファイルへの登録時間を含まない数字であるにもかかわらず，文字成分表登録時間の約 4 倍かかっている．単語方式では，形態素解析辞書を更新するたびに索引の再作成が必要であることを考慮すると，形態素解析のオーバヘッドは大きな問題であり，文字成分表を採用したことの有効性が確認できる．

n-gram 方式では，形態素解析をしないという点では高速である．しかし，転置ファイルに文書内頻度を保存するので，二次記憶への書き込むデータ量が増大し，登録時間も増大すると考えられる．

(2) 索引サイズ

日経 CD-ROM を対象とした場合の文字成分表は 70.0 MB，もと文書に対する大きさ（以下，もと文書比と呼ぶ）は 70.0/159 = 44.1%であった．一方，単語単位の転置ファイルについてはもと文書比が 1.23[5]，転置ファイル形式の n-gram 索引では約 2.0[1),26)] という報告がある．したがって，文字成分表を用いたハイブリッド方式が索引サイズの点で優れている．

☆ http://cactus.aist-nara.ac.jp/lab/nlt/chasen.html から入手可能である．

表 4 単語分割法の検索精度への影響
Table 4 Effects of word segmentation methods to average precision.

ランキング法	ハイブリッド	単語	n-gram
平均適合率	0.3618	0.3790	0.3487

(3) 検索精度

ハイブリッド方式では，簡易形態素解析と統計的複合語分割を組み合わせた検索語抽出法を用いて，単語単位のランキングを実現しており，その有効性を検索精度の点から比較する．単語方式については，単語単位の索引とランキングを組み合わせた従来方式の検索精度を実測することができなかったので，ハイブリッド方式の簡易形態素解析の代わりに茶筅を用いて測定を行った☆☆．n-gram 方式については，ランキングに一括確定法を用いることで転置ファイルモジュールがなくとも n-gram 方式の検索精度を測定可能であり，$n = 1$，2，3 について実測した．検索精度の測定結果を**表 4** に示す．単語方式の検索精度が最も良く，大規模辞書を用いて複合語を正確に構成単語に分割することの有効性が示されている．しかし，ハイブリッド方式との差は 4.5%と小さい．一方，n-gram 方式については，最も検索精度の良かった $n = 2$ の結果を表 4 に示したが，ハイブリッド方式より劣っており，簡易形態素解析の有効性が確認できる．

(4) 検索時間

検索時間を直接比較することはできないので，差異の検討を行う☆☆☆．逐次確定法を用いても処理コストの高い文書内頻度計数処理をなくすことはできないので，ハイブリッド方式が単語方式/n-gram 方式よりも検索時間がかかることは否めない．しかし，逐次確定法では，コストの高い文書スコア計算処理を行う対象文書数は登録文書数には依存せず，登録文書数に依存するのはプレランキング処理のみである．一方，転置ファイルを用いる単語方式/n-gram 方式では文書内頻度を転置ファイルから読み出して文書スコアを計算するので，登録文書数に依存して検索時間が増大する[16),29)]．したがって，大規模データベースに対しては，ハイブリッド方式と転置ファイルによる方式の検索時間の差が小さくなると期待できる．

☆☆ 索引も単語単位とする従来型と比較すると，登録時の誤解析による精度低下はなくなるが，非単語文字列との照合（たとえば「帯電（electrification）」と「携帯電話」が誤って照合すること）による精度低下の可能性がある．

☆☆☆ 日本語を対象としたランキング検索システムの検索速度に関する研究報告が見当たらないので，他のシステムとの比較もできなかった．

(5) 辞書管理

単語方式の実現には，大規模単語辞書に基づく形態素解析が不可欠である．これは，5.3.1 項の結果からも分かるように，複合語分割を行わない簡易形態素解析のみでは検索精度が低いからである．しかし，大規模辞書を用いる場合，辞書の作成・継続的拡充の問題がある．

(6) まとめ

5 つの観点からの比較をまとめると以下のようになる．

- 単語方式と比較した場合，検索精度・検索速度の点で劣るが，登録速度・索引サイズ・辞書管理の点で有利．
- n-gram 方式と比較した場合，検索速度の点で劣るが，索引サイズ・検索精度の点で有利．

7. おわりに

本論文では，日本語を対象とした統計的文書ランキング法を効率的に実現するハイブリッド方式を提案した．本方式では，索引にはシグニチャファイル形式の n-gram 索引である文字成分表，ランキングには簡易形態素解析による単語単位の手法を採用した．文字成分表を採用した結果，登録の高速化と索引の小型化を実現した．また，単語単位のランキング法を採用した結果，高い検索精度を実現した．その際，簡易形態素解析と統計的複合語分割を組み合わせた検索語抽出法により，従来型の形態素解析に不可欠であった辞書管理の手間も省くことができた．検索速度に関しては，文字成分表の誤検索を考慮して逐次確定法を適用することで，検索語の頻度情報を獲得による検索時間の増大を最小限におさえることができた．日本語検索システム評価用のベンチマーク BMIR-J1 を用いて評価した結果，ハイブリッド方式の有効性が確認できた．

謝辞 日本経済新聞 CD-ROM 93 年版の使用を了解いただいた日本経済新聞社に感謝いたします．また，本研究にご協力いただいたリコー情報通信研究所の岩崎雅二郎氏，亀田雅之氏，および本論文をまとめるに当たり貴重な意見をくださった江尻公一氏に感謝いたします．

参考文献

1) 赤峯 亨，福島俊一：高速全文検索のためのフレキシブル文字列インバージョン法 (2)，第 53 回情報処理学会全国大会論文集 (3)，pp.241–242 (1996).
2) Brown, E.: Fast Evaluation of Structured Queries for Information Retrieval, *Proc. 18th ACM SIGIR Conf.*, pp.30–38 (1995).
3) Buckley, C. and Lewit, A.: Optimization of inverted vector searches, *Proc. 8th ACM SIGIR Conf.*, pp.97–110 (1985).
4) Frakes, W. and Basza-Yates, R.: *Information Retrieval: Data Structures and Algorithms*, Prentice-Hall, New Jersey (1992).
5) Fujii, H. and Croft, W.B.: A Comparison of Indexing Techniques for Japanese Text Retrieval, *Proc. 16th ACM SIGIR Conf.*, pp.237–246 (1993).
6) 藤井洋一，望月泰行，鈴木克志，丸山冬樹：全文検索システムにおける文字成分表の作成手法，第 48 回情報処理学会全国大会論文集 (4)，pp.159–160 (1994).
7) 古瀬一隆，浅田一繁，飯沢篤志：DBMS へのシグネチャファイルの実装について，信学技報，Vol.DE94-58, pp.23–30 (1994).
8) Harman, D. (Ed.): *The 3rd Text REtrieval Conference (TREC-3)*, National Institute of Standards and Technology (1995).
9) 畠山 敦，浅川悟志，加藤寛次：ソフトウェアによるテキストサーチマシンの実現，情報処理学会研究会報告，FI25, pp.19–26 (1992).
10) Hearst, M. and Plaunt, C.: Subtopic structuring for full-length document access, *Proc. 16th ACM SIGIR Conf.*, pp.59–68 (1993).
11) 亀田雅之：軽量・高速な日本語解析ツール「簡易日本語解析系 QJP」，第 1 回言語処理学会年次大会，pp.349–352 (1995).
12) 芥子育雄ほか：情報検索システム評価用ベンチマーク Ver.1.0 (BMIR-J1) について，情報処理学会研究会報告，DBS-106, pp.139–145 (1996).
13) Knaus, D. and Schäuble, P.: Effective and Efficient Retrieval from Large and Dynamic Document Collections, *Proc. 2nd TREC*, pp.163–170 (1994).
14) 小林義行，山本修司，徳永健伸，田中穂積：語の共起を用いた複合名詞の解析，情報処理学会研究会報告，NL-101, pp.1–8 (1994).
15) 道本健二，真島 馨：高速全文検索の威力，日経バイト，No.156, pp.142–168 (1996).
16) Moffat, A. and Zobel, J.: Fast Ranking in Limited Space, *Proc. Int. Conf. on Data Engineering*, pp.428–437 (1994).
17) 丹波廣寅，硯 耕一：AltaVista における大規模検索，アドバンストデータベースシステムシンポジウム 96, pp.19–25, 情報処理学会 (1996).
18) 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).
19) 小川泰嗣：文字成分表を用いた効率的文書ランキング法の提案，アドバンストデータベースシ

ステムシンポジウム'95, pp.29–38, 情報処理学会 (1995).
20) Ogawa, Y.: Effective and efficient document ranking without using a large lexicon, *Proc. 22nd VLDB Conf.*, pp.192–202 (1996).
21) 小川泰嗣：日本語文書検索のための頻度情報を用いた効率的部分文字列索引の提案，情報処理学会論文誌，Vol.37, No.10, pp.114–120 (1996).
22) Robertson, S. and Walker, S.: Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval, *Proc. 17th ACM SIGIR Conf.*, pp.232–241 (1994).
23) RWCデータベース・ワークショップ（編）: RWCテキストデータベース（CD-ROM），メディアドライブ (1996).
24) 坂本義行：文節単位の自動分割法，計量国語学，Vol.11, No.6, pp.265–276 (1978).
25) Salton, G. and McGill, M.J.: *Introduction to Modern Information Retrieval*, McGraw-Hill, New York (1983).
26) 菅谷奈津子，川口久光，畠山 敦，多田勝己，加藤寛次：n-gram型大規模全文検索方式の開発～インクリメンタル型n-gramインデックス方式，第53回情報処理学会全国大会論文集（3），pp.235–236 (1996).
27) 武田浩一，藤崎哲之介：統計的手法による漢字複合語の自動分割，情報処理学会論文誌，Vol.28, No.9, pp.952–961 (1987).
28) Witten, I., Moffat, A. and Bell, T.: *Managing Gigabytes: Compressing and Indexing Documents and Images*, Van Nostrand Reinhold (1994).
29) Wong, W. and Lee, D.: Implementations of partial document ranking using inverted files, *Information Processing and Management*, Vol.29, No.5, pp.647–669 (1993).

(平成8年12月16日受付)
(平成9年 9 月10日採録)

小川 泰嗣（正会員）

昭和37年生．昭和63年東京大学大学院工学系研究科情報工学専攻修士課程修了．同年（株）リコー入社．情報検索・データベース等の研究開発に従事．言語処理学会，IEEE, ACM各会員．