「情報処理学会論文誌：データベース」

Vol.44 No.SIG 8（TOD18）別刷

平成15年6月発行

社団法人　情報処理学会

WWW検索における複数検索結果の統合処理とその評価

小作浩美[†,††] 内山将夫[†] 井佐原均[†]
河野恭之[††] 木戸出正継[††]

WWWの普及にともない，検索システムにおける検索処理の負荷は高くなってきている．そのため，処理負荷を分散させ，複数台のマシンで検索処理を行うのが一般的である．そして，複数の検索結果をどのように精度良く統合するか，様々な研究が行われている．同じ検索手法を用いた異なるマシンの検索結果を統合する場合，一般に最初に考えられる統合方法は，正規化である．これは，各マシンの管理するすべてのデータの文書頻度など，検索結果に関係する情報を統合し，統合した情報を利用して検索結果の正規化を行い，その正規化された検索結果（スコア）を順に並べることで最終検索結果を得るものである．しかし，正規化のために必要な情報をマシン間で通信し，統合することは処理負荷が高く，現在のWWWのように非常に大規模なデータに対して実行することは非現実的である．そのため，近似的な正規化方法などが提案されているが，そもそも無作為に収集し，分散した大規模なデータ（WWWドキュメント集合）から検索を行う状況では，上記のような正規化をしなくても同等のスコアを与える尺度を利用すれよい．我々は，正規化が不要なスコアを与える尺度（Okapi (BM25)，SMART）と正規化が必要なスコアを与える尺度（INQUERY）を用いた検索結果の統合実験を，NTCIR3のWebタスクのデータに対して行った．その結果，OkapiとSMARTのスコアをそのまま利用し，統合した結果はスコアを正規化して行った結果と同等であり，近似的な正規化を行った場合よりも精度が高いことが分かった．その一連の実験と結果について報告する．

A Comparative Study on Merging Results from WWW Retrieval System

Hiromi Itoh Ozaku,[†,††] Masao Utiyama,[†] Hitoshi Isahara,[†]
Yasuyuki Kono[††] and Masatsugu Kidode[††]

The World Wide Web (WWW) has become very popular and the number of WWW documents has increased dramatically. Along with the popularization of the WWW, the load on information retrieval systems is increasing rapidly. Usually, the information retrieval system is constituted of multiple machines for distributing the load of retrieval processes. Much research is ongoing on how to merge multiple results from information retrieval systems efficiently and with high quality. In general, the Score Normalization method is well known to get high quality. This method is ideal for merging the information, the same as making one index using all data. However, the Score Normalization method is time-comsuming and is unrealistic for huge data like WWW documents. There is some research on the Approximation Normalizing method (the Weighted Score method). Our interest is to find efficient methods that do not use the score normalization process, in order to retrieve WWW documents rapidly. If WWW documents are distributed uniformly, we think score normalization is not necessary. So, we compared the Score Normalization method and three famous IR methods (Okapi, SMART and INQUERY) with the Weighted Score method. We have found that the results of the Score Normalization method, the Okapi method and SMART method have almost the same precision rates. The Weighted Score method has no effect on this task.

1. はじめに

WWWの普及により，検索システムが処理すべきデータ量が増え，処理負荷は急激に高まっている．検索システムの処理効率をあげるため，複数のマシンで検索に関係する処理を分散させるのが一般的になってきている．そのため，WWW検索の研究には大きく分けて3つのポイントがあげられる[1)]．

(1) データの内容把握（resource description）．
(2) データベースの選択（resource selection）．
(3) 結果の統合（results merging）．

我々は検索精度の向上を目的として研究を行っており，検索結果の統合における精度の向上に特に興味があり，上記のポイントの(3)に注目している．

† 通信総合研究所
Communications Research Laboratory
†† 奈良先端科学技術大学院大学
Nara Institute of Science and Technology

分散されたそれぞれの検索システムからの結果をどのように統合するか，様々な研究が進んでいる．それらは，大きく分けると2つの場合について行われている．1つはMetasearchと一般にいわれている，検索方法が異なる検索システムを利用し結果を統合する研究と，もう1つは検索方法が同じ検索システムから得られる結果を統合する研究である．前者の場合，検索範囲を広くし確実に解を発見するタスクには向いている．これらは，検索精度そのものの向上のための研究というよりは，多数存在する検索エンジンを効率的に利用するための研究といえる．たとえば，通信コストを下げ，性能向上を目指す研究[2),3)]や各検索システムの特徴を利用し目的にあったシステムを選択しシステム全体としての効率をあげる研究[4)]がある．

これに対し，我々は膨大な数のWWWドキュメント集合に対して効率的に，なおかつ，精度良く検索するために，複数に分割したデータベースに対しそれぞれ検索し，その結果を統合する方法について研究を行っている．本稿では，ポイント(3)を達成することを目的に，後者の場合に相当する，検索方法が同じ検索システムを複数利用し複数のデータベースから結果を得，その結果を統合する状況における検索精度の評価を行う．

まず，一般に最初に考えられる統合方法は，各マシンの管理するすべての文書頻度などの情報を集計し，各データベースにおける文書頻度の差などから各検索結果に対して正規化を行い，最終的な検索結果を得る方法である．しかし，正規化のための情報を各マシン間で通信し集計することは処理負荷が高く，現在のWWW環境のような非常に大規模なデータに対して実行するのは，非現実的である．そのため，ポイント(2)に着目して，解を含む可能性のより高いデータベースを選択し，一部のデータベースから得た検索結果のみを統合するものや，部分的な情報から近似的な正規化を行う方法が提案されている．しかし，これらは，新聞記事を中心としたデータベースに対する評価が多く，実際のWWWドキュメントに対しての研究は少ない．

本研究では，NTCIR3☆のWebタスクデータを利用し，実際のWWWドキュメントに対しての実験を行う．そこで，WWWドキュメント集合の性質と検索尺度の性質を明らかにし，検索結果の統合方法について比較実験を行う．そして，その結果，単語の出現分布に偏りのないデータ集合に対して検索を行う場合，データベースサイズに影響されない検索尺度（Okapi（BM25）やSMART）を利用すれば，正規化を行わなくとも，正規化した場合とほぼ同等の精度を得られることが分かった．その一連の実験について述べ，考察する．

以下，2章で関連研究について述べる．3章で検索尺度と統合方式について述べ，4章で実験についての詳細と結果について報告する．5章でその結果に基づいた考察を行い，6章でまとめる．

2. 関連研究

複数のデータベースからの検索結果を統合する方法は，TREC☆☆において，1995年から検討されている．これは，複数の異種データベースを利用できるようになったことにより，どのようにこれらのデータベースから情報を抽出するのがよいか，また，データベースにより，検索精度がどのように変化するか，また，そのデータベースごとの特徴を利用しより良い結果を得る目的で行われている[5)]．

Callanらは TREC1 の新聞記事コレクションを利用し，INQUERY による検索を行い，次の4つの方式で最終結果を得，比較実験を行っている[6)]．

(1) 複数のデータベースの情報を1つにまとめて，検索を行う Score Normalization 法
(2) 各データベースから検索結果を得，そのランキング情報のみを利用して結果を得る Interleaved 法[7)]
(3) 各データベースからの検索結果のスコアをそのまま利用する Raw Score 法
(4) 各データベースから得た結果のスコアを，各データベースごとに各データベースの平均結果の値との差へ変換し統合する Weighted Merge 法

さらに，各データベースに解が含まれるかどうかの可能性を計算し，データベースの選択を行う CORI アルゴリズムを導入し，データベースの取捨選択も含めた精度の向上の研究もしている．それをさらに発展させた研究では，サンプルデータを利用し，そのデータに対する結果と各データベースから得た結果を比較し，スコアの正規化を行う Regression Model も提案して

☆ NTCIR (NACSIS Test Collection for Information Retrieval) プロジェクトは情報学研究所が中心に行っているプロジェクトであり，情報検索システムの大規模実験用データセットの構築を行い，共通の評価基盤の提供による情報検索関連研究の促進を目的とするものである．詳しくは http://research.nii.ac.jp/~ntcadm に掲載されている．

☆☆ TREC (Text REtrieval Conference) とは，アメリカにおける大規模検索実験プロジェクトのことである．詳細は，http://trec.nist.gov/に掲載されている．

いる[8]．また，複数の検索手法を利用する場合と単一の検索手法を利用している場合についても考察を行っている．CORI アルゴリズムを利用した検索精度と比較すると，Regression モデルにおいて，複数の検索手法を利用した場合はかなり高い検索精度であるが，単一の検索手法の場合では検索精度は下がっている．

Aslam らは，Metasearch の技術として，検索スコアが必要な方法，不要な方法，学習が必要な方法，不要な方法．ランキング情報を利用する方法としない方法を組み合わせたシステムを取り上げ，それぞれ TREC のデータに対し比較実験を行っている[9]．この研究は，複数の検索手法を利用した検索結果を統合し，精度向上を目指したものであり，1 つの検索手法での検索結果を精度良く統合することは考察されていない．

Craswell らは，複数の検索エンジンにおいて，正規化のための情報を通信しあう Integrated 環境と，それぞれの検索エンジンが出したスコアやランキング情報を各検索エンジン内で近似的な正規化を行い利用する Isolated 環境に分けて実験を行っている[10]．この研究では，Isolated 環境において利用する Feature Distance Ranking 法を新たに提案し，評価している．Feature Distance Ranking 法は，文書中で直観的に不要と思われる単語の特徴（文書末に出現する単語，文書中に何度も現れている単語，データベース中の一般語など）から，近似的な正規化を行う計算式を提案し，それぞれの特徴におけるスコアを合算して，文書の重要度（不要度）を算出する．結果として，Okapi（BM25）とほぼ同等な精度が得られている．また，直観的に不要と思われる単語特徴が正当かどうかの調査実験も行っている．

上記の研究は，どれも新聞記事や政府の公開文書などの校閲を重ね，よく整えられたデータを対象に実験が行われており，それぞれのデータベースにある特有の特徴を前提にした検索方法を模索している．たとえば，Feature Distance Ranking 法は，新聞記事において考えられる不要語の特徴を最大限に活用している．一方，検索対象を WWW ドキュメント集合にした場合，文書の作成者が多様であり，文体や文書構造も様々で，統一的な基準により校閲なされたものではない．よって，新聞記事や政府の公開文書のデータベースから抽出された不要語の特徴などが，そのまま，WWW ドキュメント集合の検索に適用できるとは考えにくい．そこで，WWW ドキュメント集合を対象とした実験を行い，WWW ドキュメント集合の特徴などを明確にする必要がある．

```
<NW:DOC>
  <NW:META>
    <NW:DOCID>NW000054231</NW:DOCID>
    <NW:URL>http://www.crl.go.jp/</NW:URL>
    <NW:DATE>Wed, 18 Apr 2001 04:11:32 GMT</NW:DATE>
    ...
  </NW:META>
  <NW:DATA>
    <NW:DSIZE>873</NW:DSIZE>
    通信総合研究所（CRL）
    …
    CRL とは．
    …
    安心して暮らしやすい国民生活のために，…
    </NW:DATA>
</NW:DOC>
```

図 1 文書データ例
Fig. 1 Data sample.

3. 検索尺度と統合方法

本稿では，WWW ドキュメント集合を複数のデータベースに分け，それぞれのデータベースから単一の検索手法を利用し，複数の検索結果を得，それら複数の検索結果を 1 つに統合し，最終的な検索結果を得る実験を行う．その検索結果の検索精度を比較することで，新聞記事などの検索において有効な方法であるスコアの正規化による検索と同等の検索精度の検索尺度について報告し，WWW ドキュメント集合の特徴や統合方法について考察する．

なお，ここでは，すべてのデータベースから，3.1 節であげる検索尺度を利用して検索結果を抽出し，その結果を 3.3 節であげる 4 つの統合方法で，統合する．検索対象のデータには NTCIR3 の Web タスクにおける Small コレクション（10 GB）を利用した．このデータは図 1 のような構成をしている．我々は，NTCIR3 で公開されたデータから，タグ情報や無駄なスペース，コントロールコードなどを削除し，テキストと思われる範囲のみを検索対象文書として利用した☆．

本章では，我々の行った実験に用いた，検索尺度と統合方法について詳細に説明する．

3.1 検索尺度

我々は，検索手法として，Okapi，INQUERY，SMART の検索尺度を利用した．Okapi と INQUERY は TREC の Web タスク[11]において検索精度が高いものである．SMART は，Web タスクには参加していないが，TREC の複数の検索タスクにおいて検索

☆ この削除作業により，データ量は 2.1 GB になっている．総文書数は約 150 万文書である．

精度が高いと評価されているものである[12].

3.1.1 Okapi (BM25)

Robertson を中心に開発された Okapi と呼ばれる検索尺度は確率型の検索モデルである．質問 Q と文書 D_i が与えられたとき，その文書が質問に適合している確率 $P(R|Q,D_i)$ を推計する．ここでは，BM25 と呼ばれる以下の式で得られるスコアを利用する[13].

$$BM25(Q,D_i) = \sum_{T\in Q} w^{(1)} \times \frac{(k_1+1)tf}{K+tf} \times \frac{(k_3+1)qtf}{k_3+qtf} \quad (1)$$

ただし，T は Q に含まれる単語である．

tf は，D_i に含まれる T の数である．

qtf は，Q に含まれる T の数である．

$w^{(1)}$ は，以下の式で表される T の重みである．

$$w^{(1)} = \log \frac{N-n+0.5}{n+0.5} \quad (2)$$

N は，検索対象の文書集合における全文書数である．

n は，T を含む文書の数である．

K は以下の式で表される値である．

$$K = k_1\left((1-b)+b\frac{dl}{avdl}\right) \quad (3)$$

ただし，k_1，b，k_3 は経験的に定められる定数である．これらの値は $k_1 = b = 1$，$k_3 = 1000$ である☆.

また，dl は，D_i の長さであり，$avdl$ は，文書集合における文書の長さの平均値である．ただし，文書の長さは，その文書に含まれる単語数のことである．

3.1.2 SMART

SMART は，Salton を中心とするグループが開発した自動索引・検索システムで，ベクトル空間モデルとして確立した検索モデルである．各語の重みから構成されるベクトルとして文書と検索質問をそれぞれ表現し，その 2 つのベクトルの内積を利用して類似度を計算する点に特徴がある[15),16].

ある質問 Q と文書 D_i が与えられ，ある単語 $T(t_1 \le T \le t_m)$ が質問 Q と文書 D_i の両方に含まれているとき，

$$SMART(Q,D_i) = \sum_{k=1}^{m}(q_{iT} \times d_{iT}) \quad (4)$$

なお，$D_i = (d_{i1}, d_{i2}, \ldots d_{it})$ であり，

$$d_{iT} = \frac{\frac{1+\log(tf)}{1+\log(avtf)}}{(1-slope)\times pivot + slope \times utf} \quad (5)$$

で，計算される．

ただし，tf は，D_i に含まれる単語 T の出現数である．

$avtf$ は，1 文書に含まれる単語の出現数の平均である．

$pivot$ は 1 文書中の異なり単語数の平均である．

utf は D_i 中の異なり単語数である☆☆.

なお，Shinhal らの実験報告[15] から，$slope = 0.25$ としている．

$$q_{iT} = \frac{1+\log(qtf)}{1+\log(avqtf)} \times \log\frac{N}{n} \quad (6)$$

ただし，qtf は，Q に含まれる単語 T の出現頻度である．

$avqtf$ は，Q に含まれる単語出現頻度の平均である．

N は，検索対象の文書集合における全文書数である．

n は，T を含む文書の数である．

3.1.3 INQUERY

Croft を中心に開発された INQUERY は，ベイズ型推論ネットワークに基づく検索尺度である．この尺度では，文書 D_i が与えられたときの質問 Q の確信度 $B(Q|D_i)$ によって文書の順位が決められる．ここでは，以下のような式で確信度を計算し利用している[17].

$$INQ(Q,D_i) = \sum_{T\in Q\cap D_i}\left(0.4+0.6\times\frac{tf}{tf+0.5+1.5\times\frac{dl}{avdl}} \times \frac{\log\frac{N+0.5}{n}}{\log N+1}\right)\times\frac{qtf}{\sum_{T\in Q} qtf} \quad (7)$$

n は，T を含む文書の数である．

N は，検索対象の文書集合における全文書数である．

dl は，文書 D_i の文書の長さ（その文書に含

☆ この定数は，情報検索パッケージの設定に準ずる[14]．この実験においては，情報検索パッケージの 1.47 版を利用した．

☆☆ システムの都合上，$pivot$ を文書集合における文書に含まれる単語数の平均（BM25 の式 (3) における $advl$ と同じ），utf を文書に含まれる単語数（BM25 の式 (3) における dl）として実験を行った．

まれる単語数）である．

$avdl$ は，文書集合における文書の長さ（その文書に含まれる単語数）の平均である．

tf は文書 D_i に含まれる単語 t の数である．

qtf は質問 Q 中の単語 t の数である☆．

3.2 検索尺度の特徴

分割データベースに対して検索を行う場合，各データベースからの検索結果を何らかの方法で統合する必要がある．最初に考えられる方法として，正規化がある．それは，複数のデータベースに対する検索結果のスコアを，複数のデータベースの情報を統合し，単一のデータベースに対しての検索結果のスコアと同等になるようにスコアを変更し検索結果を得る方法である．正規化については 3.3 節で述べるが，WWW ドキュメントのような莫大な量のデータを扱う際，正規化のための処理を行うことは現実的な方法とはいえない．しかし，新聞記事のデータベースにおいては，正規化を行う検索方法が検索精度は高く，検索には必要不可欠であるといわれている．そこで，我々は正規化を行わずに，正規化を行ったのと同等あるいはそれ以上の検索精度を得る検索手法があるか模索している．本節では，正規化に関する観点から，我々が取り上げた検索尺度の特徴について述べる．

正規化した場合，すべての検索尺度の式において tf の値は正規化によって影響を受けない．なぜなら，tf は各文書中のある単語の出現頻度であり，各文書それぞれから決まる値であるからである．ここでの正規化は，複数のデータベースを単一のデータベースと同等になるように正規化することなので，全文書数 N とある単語の出現した文書数 n が大きく変動する☆☆．そのため，正規化した結果のスコアとそれぞれの検索結果のスコアに違いがあるとすれば，idf にあたる部分による影響である．

3.1 節に示した各検索尺度の式を比較すると idf に関わる部分に違いがある．

idf の部分とは，Okapi では式 (2) の

表 1 Okapi と INQUERY の idf の値
Table 1 Value of idf in the Okapi and the INQUERY.

N	n	Value of Ex. (8)	Value of Ex. (10)
1,000	1	6.501790046	6.908255154
10,000	10	6.858014663	2.09163582
100,000	100	6.901772242	1.23239082
1,000,000	1,000	6.906255404	0.8735419263
10,000,000	10,000	6.90670483	0.676545069
100,000,000	100,000	6.906749784	0.5520495829

$$\log \frac{N-n+0.5}{n+0.5} \tag{8}$$

であり，SMART では，式 (6) の

$$\log \frac{N}{n} \tag{9}$$

であり，INQUERY では，式 (7) の

$$\frac{\log \frac{N+0.5}{n}}{\log N+1} \tag{10}$$

の部分である．

単語の出現頻度の分布が不変であると仮定した場合，一般にデータベースのサイズが α 倍になったとすれば，全文書数 N とある単語を含む文書数 n も α 倍となる．

このとき，SMART の idf については全文書数 N に対し不変である．Okapi の式では，

$$\begin{aligned}\log \frac{\alpha(N-n)+0.5}{\alpha n+0.5} &\simeq \log \frac{\alpha(N-n)}{\alpha n} \\ &= \log \frac{N-n}{n} \\ &\simeq \log \frac{(N-n)+0.5}{n+0.5}\end{aligned} \tag{11}$$

となる．これは，全文書数 N にほぼ不変な式になっている．

一方，INQUERY の式では，

$$\begin{aligned}\frac{\log \frac{\alpha N+0.5}{\alpha n}}{\log \alpha N+1} &\simeq \frac{\log \frac{N+0.5}{n}}{\log \alpha N+1} \\ &= \frac{\log \frac{N+0.5}{n}}{\log \alpha+\log N+1}\end{aligned} \tag{12}$$

分母の $\log \alpha$ のために全文書数 N に対して不変ではない．このことを数値的に確かめるために，仮想的に全文書数 N を 1,000 から 100,000,000 まで，ある単語の含まれる文書数 n を 1 から 100,000 へ変動させてみると SMART の idf は 6.907755279 の固定値となる．Okapi と INQUERY の idf の値は表 1 のようになる．表 1 から分かるように，Okapi では一定の数値に近付くが，INQUERY では徐々に減少していく．

以上の議論から分かるように，検索対象文書を無作為にデータベースに分けたときに，Okapi (BM25) と

☆ 式 (7) では，qtf を質問長 $\sum_{T \in Q} qtf$ で割ることで正規化しているが，本実験中では，質問長はスコアの順位に影響を与えないことから，質問長で割ることを省略し実装した．

☆☆ なお，平均文書長 $avdl$ もデータベースの分割によって変動するが，本実験中では，ランダムに 5 データベースに分けた場合で，$avdl$ は 170 から 182 であり，正規化したデータと比較して，各ファイルにおける平均文書長の標準偏差が 0.8 から 1.8 程度しかなく，20 ファイルに分けた場合でも同程度の標準偏差であり，スコアに影響するほどの違いはなかった．そのため，本実験中では，idf の影響が一番大きいと判断した．

SMART ではどのようなデータサイズでも *idf* の値と正規化して得られる *idf* 値が一定であると考えられるが，INQUERY では違いを生じる可能性が高い．

仮に，検索対象中の単語の出現頻度が一様に分布している場合，正規化によってスコアに影響されない Okapi や SMART の検索尺度を利用すれば，検索のスコアは正規化したスコアと同等に扱うことができ，複数の結果を統合した結果は，正規化をしなくても，同等の検索精度になるはずである．我々は，検索対象文書を複数のデータベースに分け，それらの検索結果を統合し，その検索精度を比較することで，上記のことを検証する．

3.3 統合方式

先にも述べたように，分割されたデータベースから検索結果を得るには，それぞれのデータベースから得た検索結果を何らかの方法で統合する必要がある．まず，我々は，3.1 節であげた検索尺度を利用して，各データベースから検索結果を得る．その結果を統合する方法として以下の 4 つのアルゴリズムを利用する．

(1) Score Normalization（SN，正規化）

複数のデータベースに含まれるすべての単語の出現文書総数と平均文書長などスコア算出に必要な情報をデータベースすべてから抽出して，その数値を利用し，各検索結果の類似度を算出する．すべてのデータベースを 1 つにまとめてインデックス化し検索を行ったものと同等になるため，高精度の統合ができると考えられるが，すべてのデータベースに含まれる文書数などを集計するために，計算コストが高い欠点がある☆．

(2) Score

検索結果の統合の際に，検索尺度の算出するスコアをそのまま利用する．ここで，データベースの内容に偏りがある場合には，スコアをそのまま比較しても意味のある比較はできないと考えられる．しかし，データベースに含まれる出現単語（内容）が一様に分布していれば，そのままのスコアの比較は可能であると考えられる．この方法を利用した場合，SN と比較すると統合のコストは不要であるが，検索精度の面では保証されていないので，本稿の実験で確かめる．

(3) Weighted Score（WS）

このアルゴリズムは，各データベースの中で，出現単語の傾向を算出し，その傾向に合わせてスコアを変更することにより近似的に正規化を行う方法である．データベース全体で正規化するのではなく，各データベース内で正規化を行うため，Score と同様に統合のコストは不要である[6]．ある文書の類似度（スコア）w は以下の式により算出される．

$$w = 1 + |C| \times \frac{s - \overline{s}}{\overline{s}} \qquad (13)$$

ここで，$|C|$ は検索するデータベース数，s はそのデータベースでのその文書のスコア，$\overline{s}$ はそのデータベースでの全文書のスコアの平均である．

(4) Top

各検索結果の統合において，各検索結果のランキング情報だけを利用し，各データベースからの検索結果中で同じランキングになった結果を集め，ランキングごとに結果をランダムに並べ，上位のランキングになったものから順に最終結果としてリスト化する方法である[18]．この方法は，複数の結果を同ランクの結果ごとに分類する．上位ランクのグループから順にランダムに結果を 1 つずつ取り出し，それを並べて，最終結果とする．複数の検索エンジンを利用する際に，WWW 検索の結果のランキング情報だけしか利用できない場合を想定して提案されたものである．

4. 統合実験

4.1 実験方法

大規模データに対する検索を高速にかつ効率的に行うためには，複数のマシンでデータを分割し，検索を行い，効率良く結果を統合することが求められる．一般に WWW ドキュメントを収集する場合，ロボットによるクローリングが利用されている．一般的にクローリングにおいては URL ごとに収集が行われるため，各マシンの持つデータベースに蓄積されるデータは URL により決定される．そこで，WWW ドキュメントの収集状況を考慮して，検索対象のデータベースを URL ごとに分割した場合について調査した．URL

☆ 本実験システムは，検索速度を優先しているため，検索データの索引を前もって作成している．Xeon 2.0 GHz 2 CPU，メモリサイズ 6 GB，スワップ 2 GB の Linux マシンでも，500 MB 程度のデータの索引作成に 2 時間ほどかかる．今回の実験で利用したデータのすべての索引をこのマシン 1 台で作成する場合，10 時間かかる．そのすべての索引の中から，正規化の実験のために，単語の出現文書数，平均文書長，出現文書総数を抽出し，総計する作業には，このマシン 1 台で実行した場合，4 時間程度かかり，合計 14 時間を要する．索引作成の処理時間は，索引作成プログラムを Ruby や Perl のスクリプトにより実装していることに起因するとも考えられ，C 言語で実装すればより高速にすることが可能である．また，個々のデータの索引作成についてはマシンの台数が増えれば，作業時間を短縮することが可能である．しかし，正規化データの作成には，すべてのマシンからのすべてのデータを読み込み集計するため，効率的な並列計算を実装しない限り，取り扱う文書数が増えるに従い，計算時間は増大する．WWW ドキュメントは，日々更新されるため，正規化データの更新も頻繁に行う必要がある．そのような日々変化するようなデータを対象とした場合，正規化データの作成は非現実的な作業であるといえる．

表 2 Okapi+SN, SMART+SN, INQUERY+SN の精度比較（URL ごと 5 等分）
Table 2 Average precision and precision of 10, 20 docs in the Okapi+SN, the SMART+SN and the INQUERY+SN (5 DB same size per URL).

	Okapi+SN			SMART+SN			INQUERY+SN		
	Ave P	P@10	P@20	Ave P	P@10	P@20	Ave P	P@10	P@20
qp-cont	0.1834	0.1739	0.1554	0.1386	0.1891	0.1413	0.1476	0.1739	0.1413
qp-wlink	0.1572	0.2383	0.2106	0.1179	0.2383	0.1872	0.1067	0.1936	0.1553

ごとにデータベースを分割すると，データベースよって出現単語の分布に偏りが生じる可能性がある．その偏りにより，検索結果に影響がある可能性が考えられる．その可能性を確認するため，WWW ドキュメントをランダムに分割した場合との比較を行う☆1．また，WWW ドキュメントの収集をする際には，その収集を行うマシンのスペックやマシンのネットワーク環境により，各マシンごとに収集されるデータベースのサイズは異なると考えられる．3.2 節で述べたように，データベースのサイズに影響を受ける検索尺度もあるため，各データベースのサイズが同じになるように分割した場合と，上記のように実環境を想定し異なるサイズに分割した場合についても比較した．

手続きとしては，NTCIR3 の Web タスクにおける Small コレクションのデータを URL ごとまたは，ランダムに，複数のデータベースに分割する．各データベースから，Okapi, SMART, INQUERY を利用して，NTCIR3 で利用された各質問について検索し，各データベースから検索結果を 2,000 件ずつ抽出する．それを，Score Normalization (SN), Score, Weight Score (WS), Top の各統合方法により統合し，各質問に対し 1,000 件の結果を得る．そして，平均精度☆2と上位 10, 20 文書での精度を比較する．なお，ここでは，NTCIR の Web タスクにおける，サーベイ検索の検索課題を利用した．検索課題例を図 2 に示す．本実験では，質問として基本的な検索要求の記述で，検索要求を 1 文で表した DESC (DESCRIPTION) のみを検索キーとして利用した．

```
<TOPIC>
<NUM>0008</NUM>
<TITLE CASE=''b''>サルサ，学ぶ，方法</TITLE>
<DESC>サルサを踊れるようになる方法が知りたい</DESC>
<NARR>
 <BACK>最近はやっているサルサという踊りを
 学ぶためにどうすればよいか具体的な方法が
 知りたい．例えば教室に通うという場合には，
 その場所や授業形態など，具体的な内容を必
 要とする． </BACK>
 <RELE>具体的な方法の表記のない，流行であること
 のみを扱った文書は不適合とする．</RELE>
</NARR>
<CONC>サルサ，習う，方法，場所，カリキュラム</CONC>
<RDOC>NW011992774，NW011992731，NW011992734</RDOC>
<USER>大学院修士 1 年，女性，検索歴 2.5 年</USER>
</TOPIC>
```

図 2 NTCIR3 Web タスク検索課題例
Fig. 2 NTCIR3 Webtask sample query.

表 3 各検索尺度の平均精度の t 検定結果（内容のみの比較）
Table 3 Value of paired t test in the Okapi, the SMART and the INQUERY.

	SMART+SN	INQUERY+SN
Okapi+SN	0.0048 **	0.0301 *

平均精度（Ave P）と上位 10, 20 文書での精度（P@10, P@20）については，検索結果の文書の内容のみを対象に正解判断をした場合（qp-cont）とリンク先の文書まで考慮して正解判断をした場合（qp-wlink）の NTCIR3 で公開されている正解に対し，trec_eval☆3を利用して算出した．まず，各検索尺度の比較のために，URL ごとに 5 つの同サイズのデータベースに分割した際の，Okapi+SN, SMART+SN, INQUERY+SN の精度の比較を表 2 に示す☆4．

内容のみで正解判断した場合の Okapi+SN と SMART+SN と INQUERY+SN の平均精度と上位 10 文書での精度を質問ごとに算出し，対にして t 検

☆1 URL ごとの分割とは，まず，NTCIR3 の公開データを URL ごとのグループに分け，そのグループのデータを同サイズになるように統合し，最終的にあるサイズのデータベースへ分割することをいう．つまり，図 1 のように，⟨NW:DOC⟩ と ⟨/NW:DOC⟩ の DOC タグではさまれた範囲を 1 文書とした際に，その文書の URL タグをチェックし，URL の中のドメイン名が同じであれば，同じ URL のグループの記事としてまとめる．すべてのデータを URL ごとに分けた後，URL ごとのデータをデータサイズにバラツキがでないように，5, 10, 20 のデータベースに分割することをいう．ランダムに分割する場合は，DOC タグではさまれた範囲を 1 文書として，すべての文書をデータベース数に合わせて，ランダムに分割することをいう．

☆2 平均精度とは，最上位の文書から順に調べ，適合文書が出現した時点でそれぞれ精度を計算し，最後にそれらを平均したものである[19]．

☆3 trec_eval のプログラムは，ftp://ftp.cs.cornell.edu/pub/smart/trec_eval.v3beta.shar から入手できる．

☆4 本来であれば，1 つのデータベースにした場合と比較すべきであるが，システムの都合上，2 GB を超えるファイルを扱えないため，それと同等である SN の結果で比較を行った．

表 4 平均精度：URL ごと，同サイズの 5 DB，10 DB，20 DB に分けた場合（内容のみ比較）
Table 4 Results for average precision without considering links (5 DB, 10 DB, 20 DB per URL).

Ave P	Okapi				SMART				INQUERY			
	SN	Score	WS	Top	SN	Score	WS	Top	SN	Score	WS	Top
5 DB	0.1834	0.1841	0.1833	0.1350	0.1386	0.1341	0.1379	0.1044	0.1476	0.1497	0.1404	0.1236
10 DB	0.1834	0.1788	0.1788	0.1207	0.1386	0.1320	0.1402	0.0977	0.1476	0.1498	0.1493	0.1115
20 DB	0.1834	0.1775	0.1675	0.0963	0.1386	0.1292	0.1411	0.0809	0.1476	0.1519	0.1496	0.0950

表 5 P@10：URL ごと，同サイズ 5 DB，10 DB，20 DB に分けた場合（内容のみ比較）
Table 5 Results for precision at 10 docs without considering links (5 DB, 10DB, 20 DB per URL).

P@10	Okapi				SMART				INQUERY			
	SN	Score	WS	Top	SN	Score	WS	Top	SN	Score	WS	Top
5 DB	0.1739	0.1717	0.1739	0.1565	0.1891	0.1978	0.1848	0.1478	0.1739	0.1739	0.1674	0.1696
10 DB	0.1739	0.1717	0.1826	0.1500	0.1891	0.1870	0.1848	0.1370	0.1739	0.1696	0.1630	0.1609
20 DB	0.1739	0.1739	0.1826	0.1326	0.1891	0.1761	0.1848	0.1065	0.1739	0.1717	0.1609	0.1304

定の両側検定☆を行った（以降，t 検定の両側検定を「t 検定」と記述する）．上位 10 文書での精度においては，どの検索尺度においても有意差はなかったが，平均精度においては，Okapi+SN を基準としたときの結果に有意差があった．結果を表 3 に示す☆☆．このことから，各検索尺度において正規化を行った場合には，上位 10 文書での精度には差がないが，内容のみを対象とした評価をしたときの平均精度は Okapi, INQUERY, SMART の順に良いことが分かる．

4.1.1 データベースのサイズが同じ場合

各データベースのサイズが均等になるように 5，10，20 のデータベースに分割した場合について述べる．各データベースのサイズは 5 データベースに分割した場合（5 DB）は約 500 MB，10 データベースに分割した場合（10 DB）は約 250 MB，20 データベースに分割した場合（20 DB）は約 125 MB である．

最初に，URL ごとにデータを分割し，データベースの数のみに違いがある場合，についての調査を行った．内容のみを比較した場合（qp-cont）の平均精度（Ave P）の結果を表 4 に，上位 10 文書での精度（P@10）を表 5 に示す．そして，各検索尺度の各データベース数においての各統合方法による平均精度と上位 10 文書での精度を質問ごとに算出し，質問ごとに対にして t 検定を行った．つまり，Okapi, SMART, INQUERY の Score, WS, Top の 5，10，20 のデータ

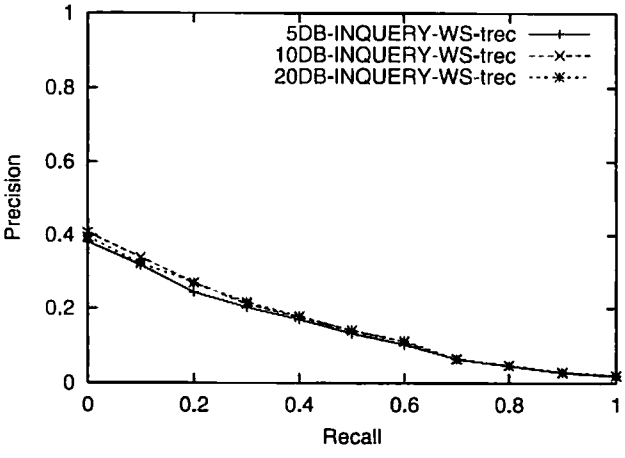

図 3 INQUERY+WS の 5 DB，10 DB，20 DB の再現率・精度グラフ（内容のみ比較）
Fig. 3 INQUERY+WS's recall-precision curves without considering links (5 DB, 10 DB, 20 DB).

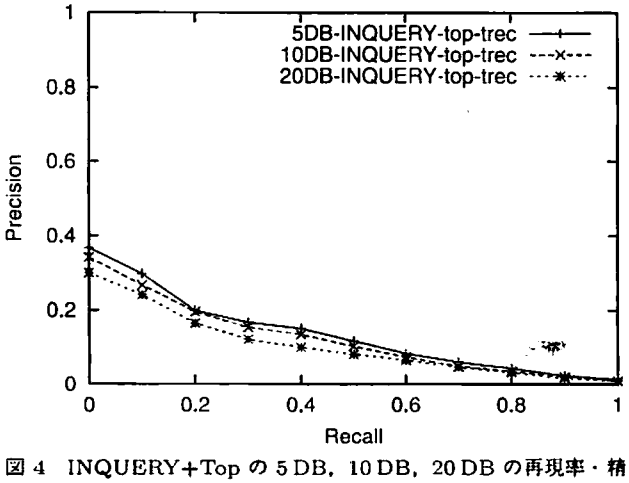

図 4 INQUERY+Top の 5 DB，10 DB，20 DB の再現率・精度グラフ（内容のみ比較）
Fig. 4 INQUERY+Top's recall-precision curves without considering links (5 DB, 10 DB, 20 DB).

☆ 本稿中の t 検定は，すべて質問ごとに平均精度などを算出し，対にして行った．質問数 47（自由度 $n = 46$）のときの t 検定の両側検定である．

☆☆ t 検定の結果に*が 1 つ付いている場合は，有意水準 5%で有意差があることを示している．*が 2 つ付いている場合は，有意水準 1%で有意差があることを示している．

表 6 平均精度：同サイズの 5 DB に URL ごととランダムに分けた場合（内容のみ比較）
Table 6 Results for average precision without considering links (5 DB same size per URL and at random).

Ave P	Okapi				SMART				INQUERY			
	SN	Score	WS	Top	SN	Score	WS	Top	SN	Score	WS	Top
URL	0.1834	0.1841	0.1833	0.1350	0.1386	0.1341	0.1379	0.1044	0.1476	0.1497	0.1404	0.1236
Random	0.1834	0.1842	0.1848	0.1519	0.1386	0.1351	0.1350	0.1305	0.1476	0.1490	0.1495	0.1438

表 7 P@10：同サイズの 5 DB に URL ごととランダムに分けた場合（内容のみ比較）
Table 7 Results for precision at 10 docs without considering links (5 DB same size per URL and at random).

P@10	Okapi				SMART				INQUERY			
	SN	Score	WS	Top	SN	Score	WS	Top	SN	Score	WS	Top
URL	0.1739	0.1717	0.1739	0.1565	0.1891	0.1978	0.1848	0.1478	0.1739	0.1739	0.1674	0.1696
Random	0.1739	0.1739	0.1717	0.1630	0.1891	0.1870	0.1891	0.1826	0.1739	0.1739	0.1717	0.1696

表 8 平均精度：ランダムに分割したサイズ違いの 5 DB の場合（内容のみ比較）
Table 8 Results for average precision without considering links (5 DB diff size at random).

Ave P	Okapi			SMART			INQUERY		
	SN	Score	WS	SN	Score	WS	SN	Score	WS
qp-cont	0.1840	0.1843	0.0929	0.1367	0.1371	0.0691	0.1488	0.1489	0.0752
qp-wlink	0.1579	0.1592	0.0785	0.1179	0.1163	0.0588	0.1071	0.1076	0.0539

ベースのときの平均精度と上位 10 文書での精度を質問ごとに算出し，質問ごとに対にして，質問数 47 のとき（自由度 $n = 46$）の t 検定の両側検定を行った．その結果，Score と WS の統合方式については，データベース数の違いによる有意差はどの検索尺度においてもなかったが，Top の統合方式のみデータベース数の違いにより有意差があった．データベース数による精度の変化を見るために，INQUERY+WS の場合と INQUERY+Top の場合の再現率・精度グラフを図 3 と図 4 に示す．t 検定の結果のとおり，Top の統合方式の場合は，データベース数が増えるに従い，精度が落ちているのが分かる．

続いて，ランダムに分割した場合と URL ごとに分割した場合において，すべての統合方法について比較を行う．内容のみを比較した場合の平均精度の結果を表 6 に，上位 10 文書の精度を表 7 に示す．この場合においても，URL ごとに分割した場合とランダムに分割した場合においての各検索尺度の平均精度と上位 10 文書での精度を利用し，t 検定を行った．その結果，Top の統合方式以外には，URL ごと，ランダム分割での有意差はどの検索尺度においてもなかった．

以上のことより，Top の統合方式はデータベースの数に影響されて，検索精度が悪化することが分かる．また，SN，Score，WS の統合方式ではデータベース数，また，URL ごととランダムの分割方法による検索精度への影響がないことが分かる．

4.1.2 データベースのサイズが違う場合

本項では，データベースのサイズが違う場合について実験を行う．まず，各データベースのサイズが，約 50 MB，約 100 MB，約 250 MB，約 600 MB，約 1 GB と異なるように 5 つのデータベースに分割した．ここでも分割方法としては URL ごとに分けた場合とランダムに分けた場合について実験を行った．しかし，前項で述べたのと同様に，Top での統合の場合を除いて，ランダム分割と URL ごとの分割には有意差がなく，精度への影響はなかった．このため，ここではサイズが異なるようにランダム分割した場合の SN，Score，WS の結果をあげる．平均精度と上位 10 文書での精度を表 8，表 9 に示す☆．

データベースのサイズが違うと，各検索尺度の算出するスコアには差が出てくると考えられる．それにともない，検索精度にも何らかの影響が出てくると考えられる．SN，Score，WS の統合方法について，データベースのサイズが同じ 5 つのデータベースの場合の各検索尺度の SN の結果を基準に両側検定で，すべて

☆ データベースサイズが異なる場合の SN の平均精度は，同サイズのデータベースから得られる SN の平均精度と若干の違いがある．これは，いくつかの質問の検索結果において，同スコアになる文書が多数存在し，その同スコアの文書が大きなサイズのデータベースに数多く存在するとき，上位 1,000 件の結果を得る場合に，下位付近で同スコアの別の文書が検索されることによると考えられる．しかし，同サイズの場合と比較し，t 検定をした結果，有意差はなく，同等の検索精度であるといえる．

表 9 P@10：ランダムに分割したサイズ違いの 5 DB の場合（内容のみ比較）
Table 9 Results for precision at 10 docs without considering links (5 DB diff size at random).

P@10	Okapi			SMART			INQUERY		
	SN	Score	WS	SN	Score	WS	SN	Score	WS
qp-cont	0.1739	0.1717	0.1239	0.1891	0.1870	0.0935	0.1696	0.1674	0.0913
qp-wlink	0.2404	0.2383	0.1596	0.2383	0.2362	0.1191	0.1894	0.1872	0.1128

表 10 サイズが同じ 5 DB の SN を基準としたサイズ違いの 5 DB の各統合方法の t 検定結果
Table 10 Results of paired t test on all merging methods (5 DB diff size at random).

	Okapi			SMART			INQUERY		
	SN	Score	WS	SN	Score	WS	SN	Score	WS
P@10	1	0.17803	0.0051 **	1	0.7040	1.52×10^{5} **	0.3979	0.1732	2.21×10^{5} **
Ave P	0.8104	0.7223	$1,55\times10^{6}$**	0.6108	0.6918	6.99×10^{7} **	0.6102	0.7246	2.59×10^{6} **

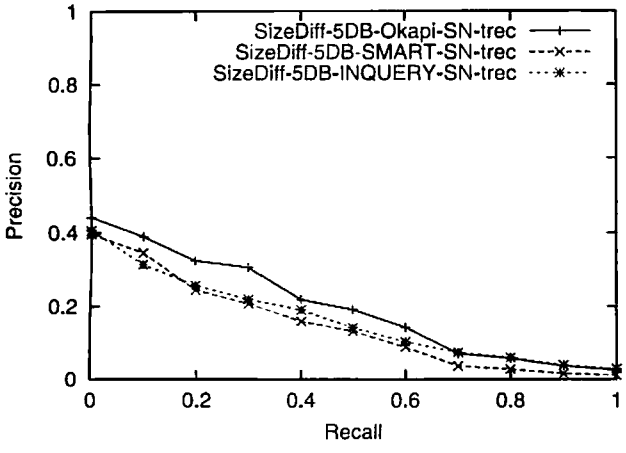

図 5 サイズ違いの 5 DB の SN 統合方式の再現率・精度グラフ（内容のみ比較）
Fig. 5 SN's recall-precision curves without considering links (5 DB diff size at random).

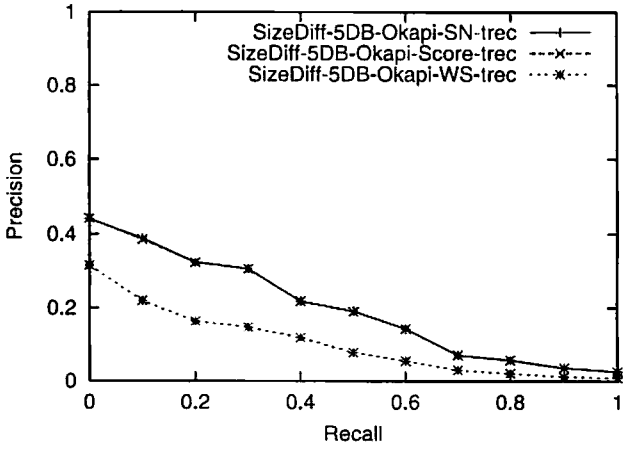

図 7 サイズ違いの 5 DB の Okapi の再現率・精度グラフ（内容のみ比較）
Fig. 7 Okapi's recall-precision curves without considering links (5 DB diff size at random).

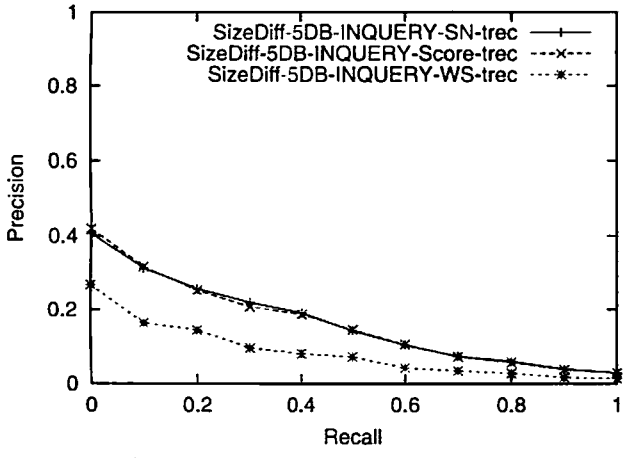

図 6 サイズ違いの 5 DB の INQUERY の再現率・精度グラフ（内容のみ比較）
Fig. 6 INQUERY's recall-precision curves without considering links (5 DB diff size at random).

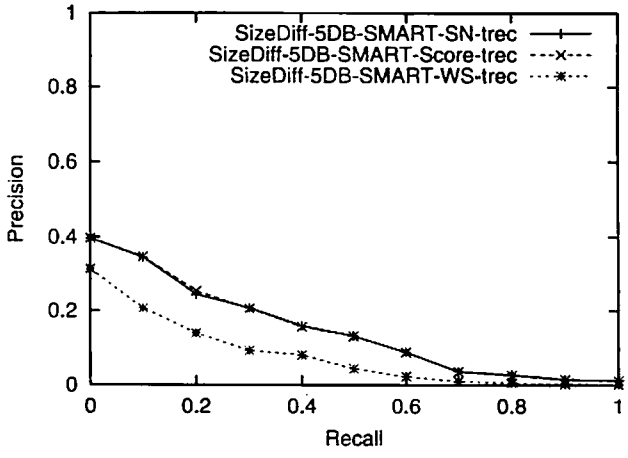

図 8 サイズ違いの 5 DB の SMART の再現率・精度グラフ（内容のみ比較）
Fig. 8 SMART's recall-precision curves without considering links (5 DB diff size at random).

の質問における平均精度を対にし，t 検定を行った．その結果を表 10 に示す．表 10 から分かるように，データベースのサイズが違う場合，各検索尺度において，SN と Score には有意差はない．WS の統合方式の場合において，有意水準 1%で有意差がある．それぞれのデータベース内で近似的に正規化する WS は，データベースのサイズがほぼ同じである場合に有効であるが，サイズが違う場合には，機能しないことが分かる．

サイズが違う5つのデータベースでのOkapi+SN, INQUERY+SN, SMART+SNの精度を比較する．再現率・精度グラフを図5に示す．この図5からOkapiの検索精度が良いことが分かる．また，サイズの違う5データベースでの各検索尺度の統合方法による検索精度の違いも比較する．INQUERYの各統合方法の再現率・精度グラフを図6に，Okapiの各統合方法のグラフを図7に，SMARTの各統合方法のグラフを図8に示す．各検索尺度とも，SNとScoreのグラフは，ほぼ重なっており，検索精度に差がないことが分かる．

4.2 順位の比較

3.2節で述べたように正規化したスコアと各検索尺度のスコアに違いがあるとすれば，*idf* 値の違いによるものである．*idf* 値の違いは，各検索尺度のスコアを比較することで調査できる．各検索尺度におけるすべての質問に対する，すべての結果のスコアを比較することは繁雑なので，我々は，Scoreを利用した統合方法の平均精度により，比較をした．4.1.2項の結果より，データベースサイズが違う場合のどの検索尺度においてもSNとScoreには有意差はなかった．INQUERY+SNとINQUERY+Scoreの結果においてデータベースのサイズが違う場合の有意差がないのは，我々の予想に反している．これは，スコアが違うとしても，検索結果の順位が変わらないのであれば，精度には影響しないことが関係していると考えられる．そこで，それぞれのSNの精度を基準にKendallの順位相関による比較を行った．つまり，Okapi+SNとOkapi+Score，INQUERY+SNとINQUERY+Score，SMART+SNとSMART+Scoreのすべての質問に対する解の順位を比較し，その順位がどれだけ似通っているかを順位相関係数で表す．その順位相関係数の平均が高いほど，正の相関が高く，分散が小さいほど安定して正の相関が高い．それらの結果を表11に示す．どの検索尺度もSNとScoreの相関が高いが，OkapiとSMARTはINQUERYに比べると，より相関が高い．さらに，INQUERY+SNとINQUERY+Scoreは分散も大きい．その理由として，データベースのサイズの違いにより，精度が極端に下がる質問が含まれているため，平均としてはそれほど影響を受けないが，分散が大きくなっていると考えられる．一方，OkapiとSMARTのSNとScoreの順位の相関は高く，分散も小さく，データベースのサイズに影響を受けていないことが分かる．さらに，OkapiとSMART，SMARTとINQUERY，INQUERYとOkapiのそれぞれの順位相関係数を対にして行ったt検定の結果，OkapiとSMARTは，0.0635と有意差は見られないが，OkapiとINQUERYは，3.31×10^{22}，SMARTとINQUERYは，6.767×10^{23}と有意水準1%で有意差がある．このことからも，OkapiとSMARTはデータベースのサイズに影響を受けない検索尺度であり，INQUERYはデータベースのサイズに影響される検索尺度であることが分かる．

表11 サイズ違いの5DBの各統合方法の順位相関係数の平均と分散

Table 11 Mean and variance of Kendall's rank correlation coefficients (5 DB diff size at random).

	Okapi	SMART	INQUERY
平均	0.97565	0.97799	0.8250
分散	0.00016	4.59×10^{23}	0.00484

5. 考　　察

INQUERY+SN, INQUERY+WSと，Okapi+SN, Okapi+Score，およびSMART+SN, SMART+Scoreの方式を中心に評価する．これは，新聞記事データの関連研究より，SNの方式が精度が高いといわれている．さらに，関連研究[8]より，近似的な正規化を利用する場合，INQUERY+WSの検索精度が高いことが分かっているためである．さらに，我々としては，3.2節で述べたように，仮に検索対象のデータにおいて出現単語（内容）が一様に分布しているのであれば，*idf* の値は変化しないため，OkapiとSMARTのScoreの検索精度がSNと同等になるのではないかと考えているからである．

データベースのサイズが同じ場合は，Okapi, SMART, INQUERYにおけるSN，Score および，WSの統合方式による検索精度の比較では，データベースの数にも関係なく，検索精度に有意な差は存在しない．これより，データベースのサイズが同じ場合には，各検索尺度においてSNとScore，WSの統合方式は同等な検索精度であると考えられる．一方，データベースサイズが違う場合には，各検索尺度におけるSNとScoreには有意差はないが，WSの統合方式を利用した場合には有意な差が存在し，WSのときの検索精度が下がっていた．これから，近似的な正規化を行った場合（WS）の統合方式が，データベースのサイズがほぼ均等であるときにのみ有効である方式であるといえる．

一般的なWWW環境を考えると，データ分割をどのようにするかによるが，各データ収集ロボットが独立にWWWデータを集める場合，URLごとに発見されたデータから順に収集するため，同サイズのデータベースになる可能性は低い．また，各データ収集ロ

ボットの処理能力により，収集されるデータベースのサイズが異なると考えられる．

Okapi と SMART に対して INQUERY の式は *idf* の部分に大きな違いが存在する．INQUERY の式 (12) を見ても分かるようにデータベースのサイズに依存する検索結果を出す．一方，Okapi の式 (11) ではサイズに依存しない．SMART においては，一定値となり，サイズによる影響はない．データベースのサイズが異なる場合，4.1.2 項で述べたとおり，Okapi と SMART における Score での精度は正規化を行った場合 (SN) とほぼ等価であり，順位相関も高いことが分かる．このことより，データベースのサイズが同じ場合でもデータベースのサイズが異なる場合でも Okapi と SMART の Score による統合結果は SN のときと検索精度に違いがなく，WWW 環境からロボットが無作為に URL ごとにデータを収集する実環境への適用が期待できる．つまり，ロボットが無作為に WWW データを収集し，そのデータを一度どこかに収集し分割するような作業をしなくても，データベースごとに検索処理を行うことが可能であることを示している．

検索対象が新聞記事のデータベースの場合，データベースの内容に経済やスポーツといった話題による出現単語に偏りがあり，スコアの正規化が有効に働くが，WWW ドキュメント集合は，URL ごとに分割した場合とランダムに分割した場合で検索精度に差が出なかったこと，Okapi と SMART における SN と Score の検索精度に差が出なかったことからも分かるように，話題や内容に偏りがない文書集合であり，データベースのサイズに関係なく出現単語が一様に分布しているためと考えられる．つまり，WWW ドキュメント集合は，文書数が多く，URL ごとにまとめられたとしても，内容によるクラスタリングを行ったものや新聞記事のデータベースとは違い，出現単語が一様に分布していると考えられる．そのため，データベースのサイズに影響されない Okapi や SMART のような検索尺度を利用すれば，正規化をしなくても各検索尺度が算出したスコアをそのまま利用することで複数の検索結果を統合することが可能であると考えられる．

最後に Okapi と SMART の検索精度について述べる．どちらの検索尺度もデータベースのサイズに影響を受けず，検索結果を出すことができる．平均精度の比較では Okapi が良いが，上位 10 文書での精度では，SMART の結果の方が良い．WWW ドキュメントの検索の際には，ユーザが上位数文書の結果しか見ないということから考えると，SMART の方が WWW 検索には向いている可能性がある．これは，SMART の式 (5) における，*slope* の影響が大きいと思われる．そのため，Okapi の上位 10 文書での精度は，Okapi の式 (3) の b を変更することで，改善できると考える．また，SMART においては，システムの実装上，*utf* と *pivot* を文書に含まれる単語数 (Okapi における *dl*) とその平均 (Okapi における *avdl*) に置き換えてある．この数値の違いが平均精度に影響している可能性もある．しかしながら，どちらの検索尺度においても，大きな検索精度の違いはなく，データベースのサイズにも影響されないため，WWW ドキュメントのような大規模な文書の検索には向いており，正規化を行わなくても，複数の検索結果を統合することが可能である．

6. おわりに

我々は，膨大な数の WWW 文書を効率的に，なおかつ，精度良く検索するために，複数に分割したデータベースに対しそれぞれ検索し，その結果を統合する方法について，研究を行っている．新聞記事のデータベースにおいては，正規化を行う検索方法が精度は高く，検索には必要不可欠であるといわれている．しかし，正規化のための処理は処理効率が悪く，WWW ドキュメント集合のような大規模データベースを扱うのは困難である．そこで，我々は正規化を行わずに，正規化を行ったのと同等あるいはそれ以上の精度を得る検索手法について比較実験を行った．

本実験により，Okapi と SMART はデータベースサイズにほぼ不変な値をとる検索尺度であるため，それぞれの結果のスコアを利用し統合をすれば，正規化を行った場合と同等の精度が得られることが分かった．これは，WWW ドキュメント集合が，内容的に偏りのない文書集合であることも示しているといえる．しかし，この結果はあくまでも今回利用した WWW データにおいての結果であるとも考えられる．つまり，実際の WWW データは更新とともに出現傾向が変化する，あるいは収集方法によっても何らかの変化がある可能性も考えられる．この点においては，WWW データを多角的に収集し，より詳細に調査していく必要があると考える．

本稿で述べた Okapi+Score の精度は NTCIR3 に参加した他のシステムの精度と比較をすると上位の精度である．しかしながら，現在の検索精度は，新聞記事検索にはまだまだ及ばず，満足のできるものではない．検索精度をあげるためには WWW 特有の Link 情報を利用する方法[20] や Relevance Feedback を利用した方法[21] があると考えられる．Link 情報を利用

する方法においては，大きく精度に結び付く利用方法はあまり提案されていない．Link 情報の詳しい解析を行って精度に大きく貢献できる利用方法の研究を進める必要がある．また，単純に Feedback をしてしまうと，WWW 検索の場合は，Feedback するときに不必要なキーワードが多く含まれてしまい精度を向上させることができない．どのキーワードを Feedback すべきか，詳細に判断する方法[22] などの工夫をする必要がある．

WWW ドキュメントは，サイズや表現方法など，様々な形式をしており，ノイズが多く含まれている．データベースのサイズの異なり具合によっても検索精度は影響を受ける可能性がある．それについては，今後，詳細に調査する必要がある．さらに，ノイズを効率良く削除する方法やノイズに強い検索方法などの研究が WWW 検索の精度向上には必要であると考える．

参 考 文 献

1) Callan, J.: Distributed information retrieval, *Advance in Information Retrieval*, Croft, W.B. (Ed), Recent Research from the CIIR, chapter 5, pp.127–150 (2000).
2) 守村 篤，清木 康：WWW を対象としたサーチエンジン間の統合機能の実現，信学技報 DE2001-30，電子情報通信学会 (2001).
3) 佐藤 永，上原 稔，酒井義文，森 秀樹：最新情報の検索のための分散型サーチエンジン，情報処理学会誌，Vol.43, No.2, pp.321–331 (2002).
4) Sugiura, A. and Etzioni, O.: Query routing for web search engines: Architecture and experiments, *Proc. 9th World Wide Web Conference* (2000).
5) Harman, D.: Overview of the fourth text retrieval conference (trec-4), *Proc. Text REtrieval Conference* (1995).
6) Callan, J.P., Lu, Z. and Croft, W.B.: Searching Distributed Collections with Inference Networks, *Proc. 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle, Washington, pp.21–28, ACM Press (1995).
7) Voorhees, E.M., Gupta, N.K. and Johnson-Laird, B.: The collection fusion problem, *Proc. Text REtrieval Conference* (1994).
8) Si, L. and Callan, J.: Using Sampled Data and Regression to Merge Search Engine Results, *Proc. 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.19–26, ACM Press (2002).
9) Aslam, J.A. and Montague, M.: Models for Metasearch, *Proc. 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.276–284, ACM Press (2001).
10) Craswell, N., Hawking, D. and Thistlewaite, P.: Merging results from isolated search engines, *Proc. 10th Australasian Databese Conference*, pp.189–200 (1999).
11) Hawing, D., Voorhees, E., Craswell, N. and Bailey, P.: Overview of the trec-8 web track, *Proc. Text REtrieval Conference* (2000).
12) Buckley, C. and Walz, J.: The trec-8 Query Track, *Proc. Text REtrieval Conference* (2000).
13) Robertson, S.E. and Walker, S.: Okapi/keenbow at trec-8, *Proc. Text REtrieval Conference* (2000).
14) 内山将夫，井佐原均：情報検索パッケージの実装，情報学基礎 63-8，情報処理学会 (2001).
15) Singhal, A., Buckley, C. and Mitra, M.: Pivoted Document Length Normalization, *Proc. 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.21–29, ACM Press (1996).
16) 汎用連想計算エンジン（GETA）（第 3 版；複数 CPU 用）導入・操作マニュアル．http://geta.ex.nii.ac.jp/
17) Allan, J., Callan, J., Feng, F.F. and Malin, D.: INQUERY and TREC-8, *Proc. Text REtrieval Conference* (2000).
18) Ozaku, H., Utiyama, M., Isahara, H., Kono, Y. and Kidode, M.: Study on merging multiple results from information retrieval system, *NTCIR 2002* (2002).
19) 岸田和明，岩山 真，江口浩二：検索実験の方法と実際：NTCIR ワークショップの試み，NTCIR 3 Pre-meeting Lecture Reports.
20) Craswell, N., Hawking, D. and Robertson, S.: Effective site finding using link anchor information, *Proc. 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.250–257, ACM Press (2001).
21) Baseza-Yates, R. and Ribeiro-Neto, B.: *Modern Information Retrieval*, Addison Wesley (1999).
22) Murata, M., Ma, Q. and Isahara, H.: Applying multiple characteristics and techniques to obtain high levels of performance in information retrieval, *NTCIR 2002* (2002).

(平成 14 年 12 月 27 日受付)
(平成 15 年 4 月 8 日採録)

(担当編集委員 大山 敬三)

小作 浩美（正会員）

独立行政法人通信総合研究所研究員．奈良先端科学技術大学院大学博士後期課程在学中．自然言語処理，情報検索システムの研究開発に従事．システムのユーザビリティや知的 HI に興味を持つ．言語処理学会，人工知能学会，電子情報通信学会，ACM 各会員．

内山 将夫（正会員）

1992 年筑波大学第三学群情報学類卒業．1997 年同大学院博士課程修了，博士（工学）．同年信州大学工学部助手．1999 年郵政省通信総合研究所非常勤職員．現在，独立行政法人通信総合研究所任期付き研究員．言語処理学会，人工知能学会，日本音響学会，ACL 各会員．

井佐原 均（正会員）

1978 年京都大学工学部卒業．1980 年同大学院修士課程修了．博士（工学）．同年通商産業省電子技術総合研究所入所．1995 年郵政省通信総合研究所入所．現在，独立行政法人通信総合研究所けいはんな情報通信融合研究センター自然言語グループリーダー．自然言語処理，機械翻訳の研究に従事．人工知能学会，日本認知科学会，言語処理学会各会員．

河野 恭之（正会員）

1989 年大阪大学基礎工学部情報工学科卒業．1994 年同大学院基礎工学研究科博士後期課程修了．博士（工学）．同年（株）東芝入社．同社関西研究所主務等を経て，2000 年 4 月奈良先端科学技術大学院大学情報科学研究科助教授．知的 CAI，マルチモーダル理解，音声対話 HI，知的インタフェース，ウェアラブルインタフェースの研究に従事．電子情報通信学会，人工知能学会，ヒューマンインタフェース学会，IEEE，ACM 各会員．

木戸出正継（正会員）

1970 年京都大学大学院工学研究科修士課程修了．同年東京芝浦電気（現，東芝）総合研究所入社．1975 年～1977 年米国バーデュ大学客員研究員．1997 年～1999 年東芝アメリカ社副社長．2000 年奈良先端科学技術大学院大学情報科学研究科教授，現在に至る．知能情報処理，特に画像処理とヒューマンインタフェースの研究に従事．1985 年オーム技術賞，1994 年電子情報通信学会業績賞．著書「画像データベース」（オーム社),「コンピュータ画像処理入門」（総研出版，共著）等，京都大学工学博士．電子情報通信学会，人工知能学会，IEEE 等会員．国際パターン認識協会（IAPR）フェロー．