多様なトピックに対応する高精度の情報検索システム
Effective Information Retrieval System for Digital Documents of Various Topics

金沢輝一 [1)]
Teruhito KANAZAWA[1)]
有限会社 ケイ・ワイ・エイ・グループ
〒112-0002 東京都文京区小石川 5-29-7
tkana@kyagroup.com

ABSTRACT.

We developed an information retrieval (IR) system with high precisions for various archives. The RS model improves the precision of retrieval by the document feature vector modification with non-exclusive document clustering based on the topic information extracted from the documents. Automatic term extraction and cluster refining methods have broadened the applicability of the RS model for general databases of documents without given keywords. Our experimental results showed that my proposed RS (Relevance-based Superimposition) model for IR improved the retrieval effectiveness by 7–15%.

The developed system achieved:

- high performance both in index generation and in retrieval,
- high precisions for various archives (scientific papers and news articles).

1 背景

次世代の情報共有システムに求められている機能の一つが高機能の検索である。本プロジェクトはメイリングリストや Web 掲示板等で共有が行われているような多様な知識資源に着目し、それらに対する効率的な情報検索システムを、関連性の重ね合わせモデル (RS モデル) [1, 2, 3] に基づいて実装する。現在、Web 掲示板等の情報システムでは汎用の全文検索システムを流用して検索処理を行っているが、個々の文書の解析から得られる情報のみを用いた手法がほとんどである。Web 掲示板等ではトピックごとに情報がまとめられ、あるいは議論の流れが存在する。これらの情報を利用することで、検索の精度を高めることができると予想される。

2 目的

本プロジェクトは最新の情報検索技術を導入することで多様なトピックの情報に対応する実用的な高精度の情報検索システムを構築することを目指す。RS モデルは本プロジェクトの提案者らが開発を行ってきた情報検索の手法であり、文書間の関連性に基づいて索引情報を補正することにより、表層的な表現のマッチングによる従来手法との比較において体感できる優位な差を実現している。これまでは学術文献を主な検索対象として評価を行ってきたが、本プロジェクトではこれをより多様な知識資源に対応する形で実装し、その有効性を示す。

3 RS モデルの特長

検索システムの基本は、検索者が入力した問い合わせと同じ表現を含む文書をデータベースから探し出し、出現頻度に応じた得点順に列挙するという処理である。問い合わせとして「渋谷にある美味しいラーメン屋を教えて」といった自然文を入力できるものも、内部で単語に切り分けて処理をしている。言葉は曖昧性を持っているものであり、表現が異なっても同じ意味だと解釈してほしい場合もある。例えば「渋谷〜」の例では、「おいしい中華そばが食べられる道玄坂のお店」も検索されてほしい。従来の検索システムでは、検索者が入力したそのままの表現でなければ検索されず、検索者が頭を悩ませていろいろな表現を入力する必要があった。

RS モデルは、この問題を解消するために開発された技術であり、自動化された処理によって索引情報が補正され、例えば道玄坂に関する文書は「道玄坂」でも「渋谷」でも検索でき、ラーメンに関する文書は「ラーメン」でも「中華そば」でも検索できるようになる。これによって、検索者が問い合わせ表現を入力する際の苦労は低減され、同時に検索精度は向上する。

4 概要

4.1 本プロジェクトの目標

4.1.1 RS モデルの実用性実証

検索処理と索引作成の両方が高速に行なえることを実証する (6.1 章)。

4.1.2 検索単位の自動抽出

多様な検索要求に柔軟に対応するシステムの実現には、トピックによる適切な文書分類が不可欠である。従来、文書分類というと排他的すなわち一つの文書は高々一つの群にしか属さないとするやり方が主流であった。この考え方の下では例えば「マイナスイオン発生機能付きのエアコンの特価情報」は「マイナスイオン系家電製品」か「特価情報」のいずれか一方のトピックとの関連性しか利用されない。RS モデルでは、一つの文書が複数の文書分類に含まれることを許容している。この特長を検索精度の向上に結び付けるためには、できるだけ多くのトピックを、より的確に抽出して索引情報を作成することが要求される。過去における RS モデルの評価では明示的に付与されたキーワードなどを手がかりに分類を行なってきたが、本プロジェクトにおいてはトピックによる文書分類を自動で的確に行なう手法を開発し、明示的な付与キーワードの無い情報資源に対する有効性を示すことを目標とした (6.3 章)。

4.2 有効性の評価

本プロジェクトでは WWW など実際的な知識資源に対する RS モデルの実効性を、以下の 2 つの観点から評価した。

4.2.1 実用性の評価

本プロジェクトの成果を組み込んだサーチエンジンを開発し、その優位性が利用者によって体感できるものであることを確認した (6.1 章)。

4.2.2 学術的評価

以下の 2 つを評価の指向性として掲げ、実践した。

- 大規模テストコレクションを用いたベンチマークによる総合的な評価 (6.3.2 章)。
- 小規模の文書集合を用いた、特長を示す具体例の提示 (6.2.3 章)。

ベンチマークによる統計的な評価は十分確立された技術であり、手法の特長を大局的に把握する良い手がかりとなる。本プロジェクトでは、NTCIR や TREC 等、国際的に広く用いられているテストコレクションで自動文書分類を適用した場合の評価を行った。

5 RS モデル

検索対象の文献と問い合わせ表現は共に自然言語の意味的曖昧性を持っている。すなわち同表記異義によって問い合わせとは無関係な文書が検索されたり、同義多表記によって検索されるべき文書が検索できない場合がある。意味的曖昧性による検索精度の低下を抑えることは情報検索の最も重要な課題の一つであり、これまで多くの研究がなされてきた。それらは大きく 3 つに分類できる。すなわち、query expansion(以下 QE) など問い合わせ表現を補正するもの、主成分分析などにより文書の特徴空間を変化させるもの、文書の特徴量を補正するもの、の 3 つである。

QE は検索者の入力した問い合わせ表現に関連する語句を加えることで問い合わせの特徴ベクトルを拡張するものである。問い合わせは情報量が比較的少ないため、これに基づいて検索者の意図を汲み取り適切な語句だけを自動的に加えることは困難である。実用上は検索対象のデータベースに合わせてパラメータの調整などを行う必要がある [4]。

特徴空間を変化させるアプローチは Latent Semantic Index [5] などの手法に代表されるように、主成分分析によって索引語を単位ベクトルとする特徴空間から概念を単位ベクトルとする低次元の特徴空間に射影することで意味のマッチングを行おうというものである。こられらの手法の課題は主成分分析の計算コストが他手法に比べて非常に大きいことであり、大規模のデータベースに対する適用に向けて研究が進められている。

文書の特徴量を補正するアプローチは、問い合わせ表現より多くの情報を用いて意味的曖昧性に対処する。これにより、問い合わせによっては検索精度が極端に低下するという、QE に発生しがちな現象を回避することができると考えられる。本プロジェクト開発者らの提案している**関連性の重ね合わせモデル** (Relevance-based Superimposition モデル、以下 RS モデル) は著者キーワードなどの情報に基づいて文書をクラスタリングし、これを解析することで文書の特徴ベクトルを補正するというものである。ここで言う文書クラスタは従来のクラスタリングに基づいた検索手法群における排他型のものではなく、一つの文書が複数のクラスタに属することを許している。排他型クラスタリングでは、例えば「ニューラルネットワークを用いた画像処理」に関する文書は「ニューラルネットワーク」か「画像処理」のいずれか一方の話題にのみ分類され、もう一方との関連性を表現することができないという問題があった。提案手法では一つの文書が複数の話題に属しているという、より自然なモデルを表現できる。以下に RS モデルの詳細

を述べる。

5.1 非排他型文書クラスタの生成

文書群 $\{\mathrm{d}_1, \mathrm{d}_1, \cdots, \mathrm{d}_n\}$ で構成されたデータベースを仮定する．また，各々の文書に対応する文書ベクトルを $\{\boldsymbol{d}_1, \boldsymbol{d}_1, \cdots, \boldsymbol{d}_n\}$ と定義する．RS モデルでは文書を非排他型クラスタ $\{C_1, C_2, \cdots, C_m\}$ に分類する．クラスタは文書から抽出したキーワード (重要語) によって形成されている．例えばデータベース中にキーワード A と B の 2 つのキーワードが存在した場合、キーワード A を含む文書はクラスタ C_A に、キーワード B を含む文書はクラスタ C_B に属する。また、キーワード A, B をともに含む文書は C_A と C_B の両方に属するものとする (図 1)。

5.2 代表ベクトルの生成

RS モデルによる文書ベクトルの拡張は，クラスタの代表ベクトル生成と，代表ベクトルを用いての文書ベクトルの実質的な修正の 2 つの段階を経て行われる．

まず最初の段階として，文書クラスタごとに代表となる特徴ベクトルを生成する．このベクトルは文書ベクトルと同じ特徴空間内のベクトルであり，同数の次元を持つ．クラスタ C の代表ベクトル $\boldsymbol{r}$ は C に属する全文書のベクトルを引数とする代表ベクトル生成関数によって生成される．α-関数族 [6] から派生する幾つかの関数の評価 [7] によると，最も良い性能を示す代表ベクトル生成関数は, Root-Mean-Square を用いたもので，代表ベクトル $\boldsymbol{r}$ の第 i 要素 r_i を次のように求める関数である．

$$r_i \equiv \sqrt{\frac{1}{|C|}\sum_{\mathrm{d}_j \in C} {d_{j,i}}^2} \tag{1}$$

ただし，$d_{j,i}$ は文書 d_j のベクトル $\boldsymbol{d}_j$ の第 i 要素である．

5.3 文書ベクトルの補正

次に，代表ベクトルを用いて各文書のベクトルを拡張する．文章が属する全ての文書群の代表ベクトルの Root-Mean-Square と，文書ベクトルとを要素毎に比較して，前者が大きければ文書ベクトルの新たな要素として置き換える．

$$d'_{j,i} \equiv \max(d_{j,i}, x_{j,i}), \tag{2}$$

$$x_{j,i} \equiv \sqrt{\frac{1}{m}\sum_{l=1}^{m} {r_{l,i}}^2} \tag{3}$$

ただし, $r_{1,i}, \cdots, r_{m,i}$ は文書 d_j が属す文書群 $\boldsymbol{r}_1, \cdots, \boldsymbol{r}_m$ の代表ベクトルの 第 i 要素 である．

以上の流れを図 2 にまとめた。

5.4 参考資料

NTCIR を用いての評価の詳細は [2, 8] を参照されたい。また、問い合わせ拡張 (query expansion) との相乗効果については [9]、言語横断検索への導入については [10, 11] で紹介している。

6 開発内容

6.1 大規模サーチエンジンへの適用

6.1.1 設計

- 国立情報学研究所 [12] 情報学資源研究センターで開発中の汎用型大規模サーチエンジンに組み込む形で実装を行なった。
- 百万〜一千万件規模のデータベースを対象とした検索サービスが実用的な速度で提供可能なことを実証することを主目的とする。ここで実用的な速度とは、検索処理自体は対話的な検索処理が快適に行なえること、索引ファイルを作成する段階では全件処理が数十時間のオーダーで完了することを目標とする。

6.1.2 実装

母体となるサーチエンジンは ANSI C で記述されており、これに RS モデルの機能 (自動重要語抽出、代表ベクトル生成関数、文書ベクトル補正関数) を ANSI C++ で記述して組み込んだ。

サーチエンジンは索引ファイルの作成、検索処理共に PC クラスタなど TCP/IP ネットワーク上に分散した複数の計算機による並列処理が可能なように設計されている。

6.1.3 評価

情報検索システムの性能評価に広く用いられている NTCIR のテストセットによって、索引ファイル作成と検索の処理時間を調べた。表 1 に検索の処理時間の例を示す。10 万件の文書を対象とした検索において、問い合わせの投入から上位 20 件を表示完了するまでに 0.5〜0.7 秒かかっている。これは一般的な使用には十分な反応時間である (参考として汎用のサーチエンジンである Namazu の例を併記した)。上位 1,000 件を表示する際に比較的長い時間を要しているのは、結果表示のために各文書のタイトルを索引ファイルから読み込む部分であり、同じ問い合わせで再度検索を行なうとこの情報が主記憶上にキャッシュされているため処理時間が短縮される。この部分は RS モデルの問題ではなく、母体としたサーチエンジンの改良が必要である。

表 2 は文書データベースの規模と索引ファイル作成各工程の処理時間の関係である。代表ベクトルの作成処理が処理件数に対する比例から大きく外れている要因は、文書数と共にクラスタ数も増加し、また代表ベクトルの非零要素が増えたためと考えられる。クラスタ数、非零要素数は文書数が 1,000 万件といった領域では収束する

表 1: **大規模データの検索処理時間**

RS モデルの検索システムは 4 並列の検索サーバと 1 台の統合サーバで構成した。スペックは索引ファイル作成の時と同じ。

1,000 件表示で「2 度目」とあるのは、検索サーバの主記憶上に索引情報が存在する場合。

検索手法	RS	Namazu
検索対象	NTCIR 10 万件	NTCIR 5 万件
問い合わせ	ネットワーク・セキュリティ	
ヒット数	99,781 件	1,889 件
上位 20 件表示	0.544 秒	0.488 秒
1,000 件表示	8.352 秒	—
1,000 件表示 (2 度目)	1.743 秒	—
問い合わせ	ネットワーク　セキュリティ　インターネット　暗号	
ヒット数	99,839 件	2,534 件
上位 20 件表示	0.763 秒	0.255 秒
1,000 件表示 (2 度目)	1.860 秒	—

表 2: **大規模データの索引ファイル作成処理時間**

測定環境は 2.3GHz Intel Xeon × 2、主記憶 2GB、スワップ領域 2GB、Solaris 8。プログラムは C/C++ で記述し、gcc 2.95.2 で最適化を行なわずにコンパイルした。各欄は上から時間、秒単位での時間、5 万件の処理時間を 1 とした場合の比。5 万件、10 万件は NTCIR 2 日本語コーパスのサブセットによる実測値。1,000 万件は外挿による推定値。RS モデル固有の処理時間は、クラスタ抽出からベクトル修正までの合計と、索引ファイルの転置処理の 80%の和とした。転置処理の 80%としたのは、RS モデルの作用によってベクトルの非零要素が平均 5 倍に増加したという実験結果に基づく。

	5 万件	10 万件	1,000 万件 (外挿)	
並列処理	×	×	×	8 並列
形態素解析 (Juman サーバ)	1:37’41”	3:21’2”	17 日	51 時間
	5,861s	12,062s	1,457,945s	182,243s
	—	2.06	251	25.75
文書ベクトル作成	1’7”	2’49”	21:48’	2.7 時間
	67s	169s	78,475s	9,809s
	—	2.52	1,170	147
クラスタ抽出	32”	1’8”	6 時間	45’
	32s	68s	21,624s	2,703s
	—	2.13	318	39.75
代表ベクトル作成	1’5”	34’3”	5 日	14.2 時間
	65s	2,043s	408,600s	51,075s
	—	31.4	6,280	785
ベクトル修正	6’19”	17’39	11 日	34 時間
	379s	1,059s	977,772s	122,222s
	—	2.79	2,576	322
索引ファイルの転置	12’38”	24’24”	32 時間	4 時間
	758s	1,464s	115,543s	14,443s
	—	1.93	152	19.0
計	2 時間	4.7 時間	35.4 日	4.4 日
	7,162s	16,865s	3,059,959s	382,495s
RS モデル固有の処理時間	18 分	1.2 時間	17.4 日	2.2 日
	1,082s	4,341s	1,500,430	187,554

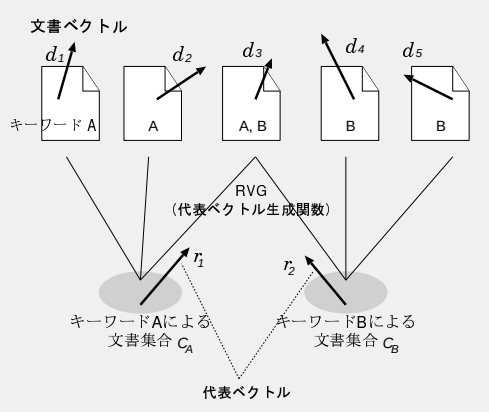

図 1: **キーワードによる関連文書クラスタの生成**

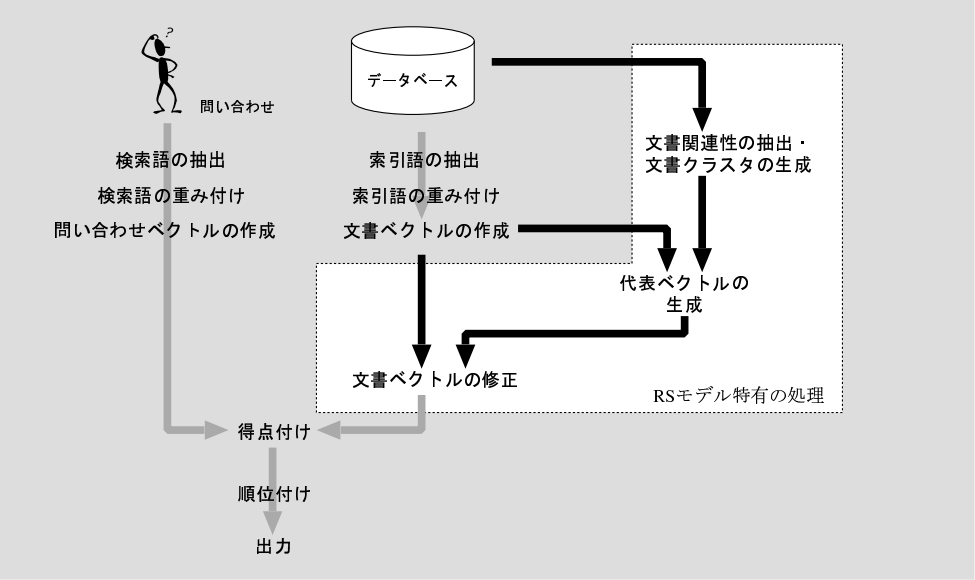

図 2: RS **モデルを導入した検索システムの処理の流れ**

と予想されるが推定が困難であったため、推定処理時間は単純に比例させたものに 2 を乗じた。

1,000 万件を対象とする場合、8 並列処理を行なうことにより約 4.4 日で索引ファイルを作成することができると見込まれる。これは当初の目標である実用的な処理速度を達成したと言ってよいだろう。ただし、より一層の高速化の余地は十分残されている。例えば、形態素解析処理をより高速なパッケージで置き換えるとか、代表ベクトルの作成時に微小な要素は零に丸めるなどの処理を行なうことでこれを更に半分程度に短縮できるのではないかと考えている。

6.2 RS **モデル実装パッケージ** for Java **の開発**

6.2.1 **設計**

- 前章で述べたサーチエンジンのように大規模なデータベースではなく、例えば個人のメイル管理や中小規模の掲示板のメッセージ検索に利用することを想定し、単一計算機上で軽快に動作することを目標とした。
- 索引ファイルの作成時に比較的大きな中間ファイルを作成する必要があり、この処理に memory mapped file 機構を応用することにより、メモリの効率的利用と高速な処理の両立を図った。
- Java の代表的な文書検索フレームワークである Jakarta Lucene [13] を利用することで、索引作成

の前処理である語句解析を分離。言語等に応じてLucene API に対応した各種の語句解析パッケージと組み合わせることを可能とした。

6.2.2 実装

■動作に必要な条件、環境

- Java 2 SE (V1.4.0 以上) の JRE(実行環境) [14]
- Jakarta Lucene 1.2 以上

■インストール 環境変数 CLASSPATH に本パッケージの jar ファイルと lucene の jar ファイルを追加する。

■パッケージ概要

- com.kyagroup.ir.DocumentID2Title … 検索結果を表示する際のラベルとなる文字列を管理する。
- com.kyagroup.ir.FreqToWeightConverter … 単語の頻度ファイルを元に単語の重みを tf·idf によって計算し、ファイルに出力する。
- com.kyagroup.ir.IndexInverter … 索引ファイルを転置する。
- com.kyagroup.ir.MaxTFExtractor … tf·idf 値の正規化に用いる、各文書での単語の出現頻度の最大値をファイルに出力する。
- com.kyagroup.ir.Retriever … 問い合わせベクトルと全文書ベクトルとの間で関連度 (内積) を計算し、値の大きい順に全文書の ID を並べる。
- com.kyagroup.ir.parser.QueryParser … 自然文から検索語を抽出し、問い合わせベクトルを構築する。
- com.kyagroup.ir.rs.FrequentTermClusterGenerator … 各文書から重要語を自動抽出し、その重要語に基づいて文書をクラスタリングする。
- com.kyagroup.ir.rs.RepresentativeVectorGenerator … クラスタの代表ベクトルを生成する。
- com.kyagroup.ir.rs.VectorModifier.java … 文書ベクトルを修正する。
- com.kyagroup.ir.util.DocumentTermViewer … 文書に含まれている索引語の一覧を表示する。
- com.kyagroup.ir.util.IndexCompressor … 索引ファイルを圧縮する。

6.2.3 適用例

オープンソースプロジェクト jxta のメイリング・リスト (約 1 万 5 千文書) を対象に検索する Servlet を開発し、ケイ・ワイ・エイ・グループ社の社内で利用している (図 3)。

「firewall」という問い合わせを用いて従来型検索と RS モデルの比較を行なった。従来型検索で 3 位になったメイル・スレッド「HTTP Proxies with Password authentication」は複数の質問に対する質疑応答が含まれており、firewall はその一部にしか関係しない。RS モデルではこのようにスレッド内で偏って出現する語句の得点は低く扱われ、より望ましい (firewall を話題の中心とした) スレッドが順位を上げるという結果が得られた。

6.3 重要語抽出による文書クラスタリング手法の開発

6.3.1 設計と特徴

本プロジェクト開始以前、RS モデルは人間が各文書に付与したキーワードを元に文書クラスタリングを行なうという方式であった。しかし、全ての文書にキーワードを付与する作業は容易ではなく、これに代わるアプローチが求められていた。

本プロジェクトでは、人間の代わりに計算機が自動で各文書のキーワード (重要語) を選択する手法を考案した。そのような技術は既に**重要語抽出**として存在していたが、RS モデルで利用する場合の、

- 一つの文書から抽出される重要語は複数である
- 各文書から幾つの重要語を抽出するのが適当か予め分からない

等の条件下で、人間が付与したキーワードを用いた場合と同程度の性能を達成することは困難であった。

今回開発した手法は、統計的情報 (tf·idf) に基づいて高速に重要語を抽出しながらも、新聞記事における「記事」、学術文献における「実験」等のありふれた語 (ストップワード) が誤って重要語と認識される率を大幅に低減した。

6.3.2 評価

NTCIR の学術文献 (約 32 万件) で最大約 7.6%、TREC の新聞記事 (約 9 万 3 千件) で最大約 15.7%の検索精度 (平均適合率) の向上が確認された。これら大規模なテストセットを用いた評価において 5%の精度向上は有意な差として認識できるとされており [15]、有効性が十分に示された結果と言える。また人間が付与したキーワードを用いた場合の 5.9%(NTCIR) と比較しても高い性能を示している (表 3)。

7 今後の課題、展望

7.1 データベースの高速な更新

一般的にデータベースは随時更新 (追加、削除) されるものであり、データベースの内容変更が検索サービスを停止せずに行なえることが求められる。RS モデルを用いる場合、文書の追加削除は変更のあった文書以外のベクトルも変化する可能性があるが、大規模なデータ

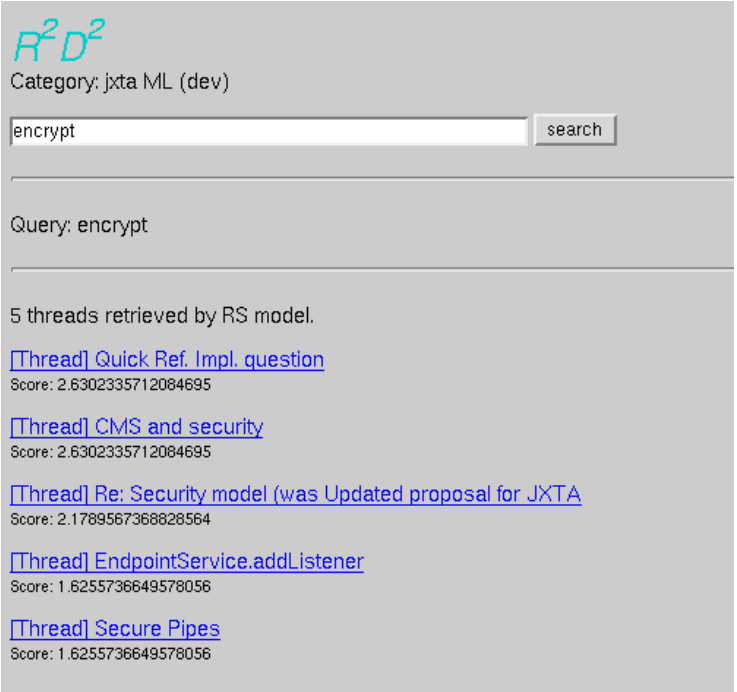

図 3: **メイリング・リスト内でのスレッド検索**

表 3: **自動重要語抽出を適用した場合の検索精度**

method	平均適合率 (向上率)	
	NTCIR	TREC SJM
RS 無し	.2984	.1773
RS / 著者が付与したキーワード	.3160 (+5.90%)	—
RS / 重要語抽出、ストップワード除去無し	.3054 (+2.35%)	—
RS / 重要語抽出、ストップワード除去	.3211 (+7.61%)	.2051 (+15.68%)
RS / 著者キーワード、ストップワード除去	.3168 (+6.17%)	—

ベースに対して小量の変更があった場合は変更文書のベクトル補正のみで近似することで高速に対応し、正確な状態は別プロセスで処理したものを定期的に反映させる (図 4)。

7.2 リンク的情報に基づく文書クラスタリング

本プロジェクトでは RS モデルが必要とする文書間の関連性情報を、重要語抽出の応用で得るアプローチをとった。文書の関連性は、他にも文献参照やハイパーリンク、書誌情報などから抽出でき、これらを組み合わせていくことによって更に高精度の検索が行なえると考えている。

8 参加企業及び機関

RS モデル実装パッケージ for Java の実装などにおいて、ケイ・ワイ・エイ・グループ社の協力をいただいた。

9 まとめ

本プロジェクトは RS モデルを導入することで、多様な知識資源に対応する効率的な情報検索システムを実装し、その有効性を様々なアプローチで実証した。

従来、RS モデルは人間が検索対象の全文書にキーワードを付与する作業が必要だったが、RS モデルに適した重要語抽出技術を開発したことによって自動化され、人間がキーワードを付与した場合に比べて同等以上の検索精度を達成した。

また、RS モデルは索引ファイルの作成処理段階で工程が増えるため処理時間は増加するが、大規模データベースの処理時間を実測し、1,000 万件程度ならば 8 並列の比較的小規模な PC クラスタを用いても十分実用的な処理時間を達成することを確認した。

本プロジェクトによって、RS モデルが電子図書館の

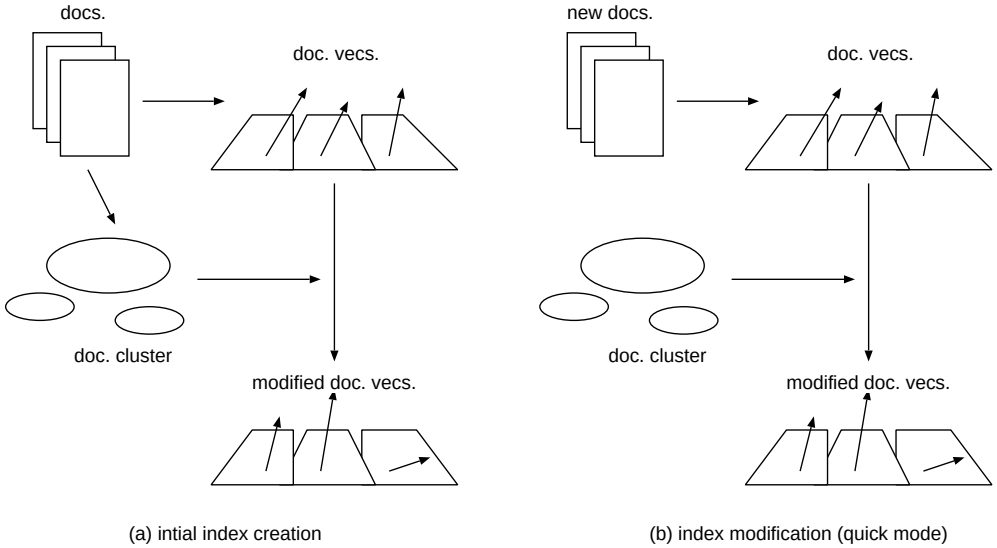

図 4: **高速なデータベース変更**

ような大規模な知識資源からメイリングリストのような小規模のものまで幅広く対応していることが示された。今後は様々な場面で RS モデルが応用されていくことを期待し、またそのようなシステムの開発を続けていくつもりである。

謝辞

本プロジェクトでは「国立情報学研究所共同研究員規程」に基づく共同研究として、国立情報学研究所 (NII) の構築した情報検索システム評価用テストコレクション NTCIR-1 および NTCIR-2（本格版研究目的使用）を使用した。このテストコレクションには、`http://research.nii.ac.jp/ntcir/acknowledge/thanks1-ja.html` のリストに示されている学協会によって開催された学会における発表論文の要旨、および、文部省科学研究費補助金研究成果の概要が含まれている。

本プロジェクトでは Linguistic Data Consortiums が配布している TIPSTER corpus [16]、TREC プロジェクト [17] が構築した情報検索システム評価用質問セットおよび正解判定リストを使用した、

本プロジェクトでは Apache Software License 1.1 [18] に基づいて配布されている Jakarta Lucene を使用した。

参考文献

[1] 金沢輝一, 高須淳宏, 安達淳, “文書関連性を考慮した検索方式,” 情報処理学会 データベースワークショップ ’98 (情報処理学会研究報告), Vol. 98, No.58, 98-DBS-116(2), pp. 165–172, 福井, July 1998.

[2] T. Kanazawa, “R^2D^2 at NTCIR: Using the Relevance-based Superimposition Model,” *NTCIR Workshop 1 Proc.*, pp. 83–88, Tokyo, Aug. 1999.

[3] T. Kanazawa, A. Takasu, and J. Adachi, “Effect of the Relevance-based Superimposition Model on Information Retrieval,” *IPSJ Database workshop 2000 (IPSJ SIG Notes)*, Vol. 2000, No.69, 2000-DBS-122, pp. 57–64, Iwate, July 2000.

[4] M. Mitra, A. Singhal, and C. Buckley, “Improving Automatic Query Expansion,” *SIGIR ’98*, pp. 206–214, 1998.

[5] S. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, “indexing by latent semantic analysis,” *J. American Society for Information Science*, Vol. 41, No.6, pp. 391–407, 1990.

[6] 林幸雄, “個人選考による情報アクセスに適したデータモデルについて,” 情報処理学会 データベースワークショップ ’98 (情報処理学会研究報告), Vol. 98, No.58, 98-DBS-116(2), pp. 381–388, July 1998.

[7] 金沢輝一, 高須淳宏, 安達淳, “関連性の重ね合わせモデルによる文書検索,” 電子情報通信学会 データ工学ワークショップ ’99 (電子情報通信学会研究報告), Vol. 99, 鹿児島, March 1999.

[8] T. Kanazawa, A. Takasu, and J. Adachi, “R^2D^2 at NTCIR 2 Ad-hoc Task: Relevance-based Superimposition Model for IR,” *NTCIR Workshop 2 Proc.*,

pp. 204–210, Tokyo, March 2001.

[9] 金沢輝一, 高須淳宏, 安達淳, “関連性の重ね合わせモデルによる問い合わせ表現の自動拡張手法,” 第59回 (平成11年後期) 情報処理学会全国大会, 分冊3, pp. 21–22, 岩手, Sept. 1999.

[10] T. Kanazawa, A. Aizawa, A. Takasu, and J. Adachi, “The Effects of the Relevance-based Superimposition Model in Cross-Language Information Retrieval,” *Proc. 5th European Conference on Research and Advanced Technology for Digital Libraries*, pp. 312–324, Darmstadt, Sept. 2001.

[11] 金沢輝一, 相澤彰子, 高須淳宏, 安達淳, “日英言語横断検索における関連性の重ね合わせモデルの効果,” 情報処理学会論文誌「データベース」, No. 13, Feb. 2002.

[12] `http://www.nii.ac.jp/`.

[13] `http://jakarta.apache.org/lucene/docs/index.html`.

[14] `http://java.sun.com/j2se/1.4.1/ja/index.html`.

[15] E. Voorhees, “Variations in Relevance Judgements and the Measure of Retrieval Effectivenss,” *SIGIR ’98*, pp. 315–323, 1998.

[16] TIPSTER at the Linguistic Data Consortiums: `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93T3A`.

[17] TREC Home Page: `http://trec.nist.gov/`.

[18] `http://www.apache.org/LICENSE`.