2008－NL－186（6）
2008／7／17

分散連想計算サーバー群を統合する連想検索システム「想・IMAGINE」

小池 勇治† 西岡 真吾† 森本 武資† 丸川 雄三† 高野 明彦†

†国立情報学研究所 〒101-8430 東京都千代田区一ツ橋 2-1-2

E-mail: †{yuji, nis, tak, maru, aki}@nii.ac.jp

あらまし 「想・IMAGINE」は，ネットワーク上に分散する複数の異種コーパスへの連想計算を統合する検索システムである．個々のコーパスに対する連想計算の結果を動的に組み合わせることにより，それら全体をあたかも一つの巨大なコーパスへの連想計算であるかのように扱うことができる．「想・IMAGINE」では，関連性フィードバックとして，複数の異種コーパスに由来する文書群をまとめて指定できる一方で，連想計算の結果をコーパスごとに個別に表示するインタフェースを備えている．ゆえに，「想・IMAGINE」の利用者は，個々のコーパスの特徴を活かしつつ，多角的な視点に立って，複数のコーパスにまたがる連想検索を対話的に繰り返すことができる．

キーワード 連想計算，連想検索，関連性フィードバック，Web サービス，検索システム

IMAGINE: A Scalable Distributed Associative Search System

Yuji KOIKE† Shingo NISHIOKA† Takeshi MORIMOTO† Yuzo MARUKAWA†

and Akihiko TAKANO†

†National Institute of Informatics 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430 Japan

E-mail: †{yuji, nis, tak, maru, aki}@nii.ac.jp

Abstract IMAGINE is a novel associative search interface that can access distributed heterogeneous corpora on the Internet. IMAGINE combines corpora into a single corpus virtually. Search results from different corpora are merged logically, while they are divided into groups and displayed according to the originating corpus of the each result. Users can choose multiple documents of different corpora as seeds for a relevance feedback. Therefore users can interactively and repeatedly consult the corpora and/or individual corpus, from various perspectives.

Keyword Association computation, Associative search, Relevance feedback, Web service, Retrieval system

1. はじめに

本稿では，分散する複数のコーパスに対する連想計算[1]を動的に組み合わせて，まるで一つのコーパスに対する連想計算であるかのように取り扱える連想検索システム「想・IMAGINE」について述べる．

連想計算とは，対象コーパスから，与えられた単語群に類似する文書群を見つけ出す計算であり，また，その双対な計算，すなわち，与えられた文書群に類似する単語群を見つけ出す計算である．連想計算を用いれば，情報検索におけるキーワード検索や，関連性フィードバック，文書の要約など，様々な処理を統一的かつ高い精度で行うことができる．さらに，汎用連想計算エンジン GETA[1]を用いれば，1000 万件規模のコーパスを対象とする場合においても，連想計算を高速に実現することができる．実際，我々はこれまでに GETA を用いて Webcat Plus[4]や新書マップ[5]，文化遺産オンライン[6]，BOOK TOWN じんぼう[7]など，実用的な様々な情報サービスを構築してきた．

我々の従来の情報サービスにおいては，興味の対象が漠然としている利用者であっても，連想計算に基づく関連性フィードバックによって，興味の対象を徐々に明確にしていくことができるという特徴がある．しかし，一方で，関連性フィードバックとして指定できる文書が限られており，それぞれの情報サービスの対象コーパス内の文書しか扱えないという問題点があった．そのため，ある情報サービスで得られた検索結果を，他の情報サービスに対する関連性フィードバックとして指定するというような，複数の情報サービスにまたがる検索を行うことができなかった．

この問題点を解決するため，我々は次の 3 点を実現する Web アプリケーション「想・IMAGINE」を構築した．

(1) ネットワーク上に分散する複数の異種コーパスに対して，同時に連想計算を行うことができる．

(2) 異なるコーパス由来の文書をも，関連性フィードバックの対象とすることができる．
(3) 連想計算の結果を，コーパスごとに個別に表示することができる．

ゆえに，「想・IMAGINE」を用いれば，複数のコーパス上の連想計算と関連性フィードバックを，まるで一つの巨大なコーパスへの連想計算であるかのように扱うことができる．また一方で，個々のコーパスの特徴や傾向を理解しつつ，多角的な視点に立って，検索結果を俯瞰することができる．

「想・IMAGINE」の構築においては，個々のコーパスに対する連想計算を Web サービスとして提供する連想計算サーバーGETAssoc[2]を用いた．統合の対象とするコーパスにアクセスする際に必ず GETAssoc を経由することにより，ネットワーク上に分散する複数の異種コーパスへのアクセス方法を統一し，コーパスの柔軟な管理を実現した．

本稿の構成は次の通り．第 2 章では，連想計算の概要を述べ，連想計算に基づく我々の従来の情報サービスを紹介する．第 3 章では「想・IMAGINE」の設計と実装を説明する．具体的には，「想・IMAGINE」のインタフェース，および，複数コーパス上の関連性フィードバックの実現方法，連想計算サーバーGETAssoc の概要，「想・IMAGINE」のシステム構成を説明する．第 4 章では「想・IMAGINE」の運用実績を示し，第 5 章で「想・IMAGINE」の構築方式について議論する．最後に，第 6 章で本稿をまとめ，今後の課題を述べる．

2. 連想計算

連想計算とは単語群と文書群間の類似性を求める計算であり，双対な二つの基本計算がある．一つは，与えられた単語群に類似する文書群（関連文書群）を求める計算であり，もう一つは，与えられた文書群に類似する単語群（特徴語群）を求める計算である．ここで特徴語群とは，文書群に特徴的に現れる単語群である．

連想計算を組み合わせれば，文書群から関連文書を求めることも，単語群から類似する単語群を求めることもできる．例えば，文書群 A に関連する文書群 B を求めるには，まず，文書群から単語群への連想計算によって，文書群 A の特徴語群 W を求める．次に，単語群から文書群への連想計算を行えば，特徴語群 W から求める関連文書 B を得られる．単語群間の連想計算についても，同様の組み合わせによって行うことができる．このように，連想計算を用いれば，単語群や文書群をクエリとするような情報検索を実現できる．本論文では，連想計算に基づく検索を「連想検索」と呼ぶ．

連想計算は，汎用連想計算エンジン GETA（Generic Engine for Transposable Association）[1]を用いることで実現できる．GETA は 1000 万件規模の文書コーパスにおいても，高精度かつ高速な連想計算を可能にするソフトウェアである．連想計算は，ベクトル空間モデルを採用しているので，大規模なコーパスを対象とする場合，そのデータ量は極めて大きなものとなるが，GETA はベクトル空間のインデックス情報を圧縮することで高速化を実現している．

我々はこれまでに，GETA を用いて，実用に耐える各種情報サービスを構築・公開してきた．代表的なサービスに，以下のようなものがある．なお，GETA は連想計算のみを行うソフトウェアであり，GETA 単体には Web サーバーの機能はない．

・Webcat Plus
全国の 1000 館を超える大学や研究機関が所蔵する，日英図書 1023 万冊の目次や概要を，高速に連想検索できる図書情報検索ポータルサイト．（2002 年 10 月公開）

・新書マップ
連想検索を応用した新しい情報サービス「新書マップ」を立ち上げた．10000 冊以上の新書・選書が約 1000 個のテーマに分類され，テーマごとに本のリストと読書案内を読むことができる．（2004 年 6 月公開）

・文化遺産オンライン
全国の 80 館の美術館や博物館などが収蔵する，約 7000 件の文化遺産情報について，連想検索と閲覧ができる．（2004 年 4 月試行版公開，2008 年 3 月正式公開）

・BOOK TOWN じんぼう
世界一の古書店街である神保町のポータルサイト．神保町にある古書店のうち 160 店の詳しい紹介と共に，在庫 35 万冊の連想計算を提供している．（2005 年 10 月公開）

ここに挙げた情報サービスは，いずれも連想検索を応用しているので，検索結果の中から興味のある文書群を選んで，それらを新たなクエリとして再び文書群から文書群への連想検索を行うことができる．ゆえに，ひとたび連想検索を始めれば，利用者自身は検索キーワードを捻出する必要がない．したがって，興味の対象が漠然としている利用者であっても，連想計算を繰り返す過程で，興味の対象を明確にしていくことができる．逆に，対象を絞り込みすぎてしまった場合でも関連する周辺情報が表示されているため，容易に思考の幅を広げることができる．このように，連想計算に基づく情報サービスには，対話的な連想検索を行えるという利点がある．

しかし，これらの情報サービスでは，個々の情報サ

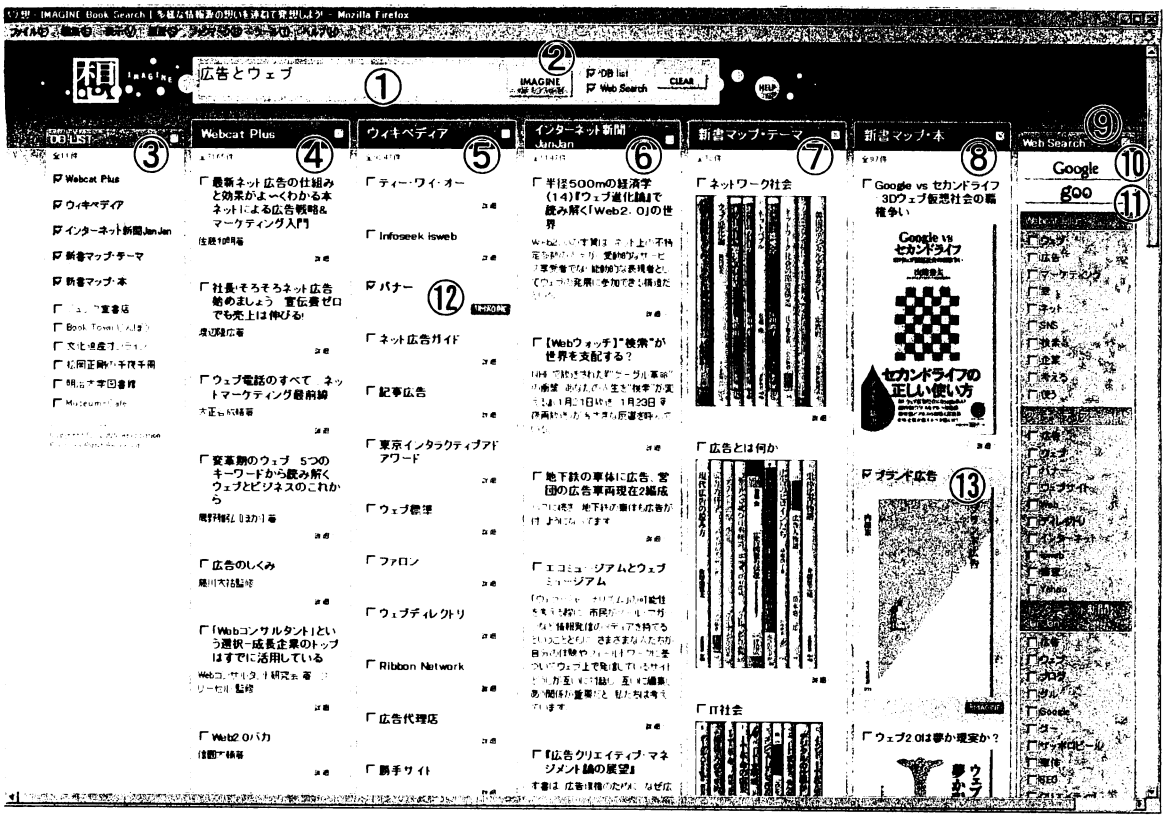

図 1.「想・IMAGINE」の検索画面

ービスの内部でしか連想検索を繰り返せないという問題点がある．そのため，例えば，ある情報サービスで得られた検索結果を新たなクエリとして，他の情報サービスで検索をするような，複数の情報サービスをまたがる検索を行うことができない．

3.「想・IMAGINE」

3.1. 実現する機能

「想・IMAGINE」は，ネットワーク上に分散する複数の異種コーパスへの連想計算を統合するシステムである．個々のコーパスに対する連想計算の結果を動的に組み合わせることにより，それら全体をあたかも一つの巨大なコーパスへの連想計算であるかのように扱うことができる（図 1）．ゆえに，「想・IMAGINE」においては，複数のコーパスをまたがった連想検索を，対話的に繰り返すことができる．

「想・IMAGINE」における連想検索の様子を，「広告とウェブ」という文章をクエリとした場合（図 1）を例に取り，説明する．まず，連想検索を行うには，①テキストボックスにクエリ「広告とウェブ」を入力し，②「IMAGINE」ボタンをクリックする．すると，④「Webcat Plus」や⑤「ウィキペディア」，⑥「インターネット新聞 JanJan」など，各種のコーパスにおける関連文書群が，検索結果として（類似度の高い順に）表示される．利用者は，検索結果をコーパスごとに参照しながら，個々のコーパスの特徴や傾向を理解した上で，多角的な視点に立って関心事の整理をすることができる．

「想・IMAGINE」では，さらに，複数のコーパス上の検索結果から選んだ文書群を新たなクエリとして，再び連想計算を行うことができる．検索結果の文書を新たなクエリとして選択するには，文書内のチェックボックスにチェックを入れる．例えば，図 1 では，二つの検索結果⑫「バナー」と⑬「ブランド広告」を，次の連想検索のためのクエリとして選択している．この状態で②「IMAGINE」ボタンをクリックすれば，この新たなクエリによる連想検索を行うことができる．

「想・IMAGINE」では検索結果として，関連文書群の他にもう一つ，関連文書群に関する特徴語群が⑨「Web Search」に表示される．例えば，図 1 では，「マーケティング」や「SEO」といった関連キーワード群

（特徴語群）が提示されている．関連文書と同様，個々の特徴語についても，チェックを入れることによって，次の連想検索のためのクエリに含めることができる．また，チェックを入れた関連キーワードは，連想検索のためのクエリ以外にも，Google や goo といった外部の検索サービスに対するクエリとしても，利用することもできる．その場合には，⑩「Google」ボタンや⑪「goo」ボタンをクリックする．

以上のように，「想・IMAGINE」においては，テキストボックスに入力した自由文書と，検索結果の関連文書群，関連文書群の特徴語群の三種類の情報を組み合わせて，新たなクエリを表現し，連想検索を繰り返していくことができる．

なお，利用可能なコーパスの一覧は③「DB LIST」に表示されており，個々のコーパスのチェックボックスにチェックを入れることで，検索対象を自由に選択できる．「DB LIST」でのコーパスの並び順（上から下）は，検索結果の並び順（左から右）に対応している．コーパスは，「DB LIST」内のコーパス名をドラッグ＆ドロップすることによって並び替えることができる．

3.2. 複数のコーパスにおける連想計算

第 2 章で紹介した我々の個々の情報サービス（Webcat Plus や新書マップなど）は，いずれも，単一のコーパスに対して連想計算を行うものであった．すなわち，クエリとなる文書群と，検索結果として得られる文書群は，どちらも，同じコーパスに含まれる文書群であった．

一方，「想・IMAGINE」における，複数のコーパスに対する連想計算では，クエリとなる文書群と，検索結果として得られる文書群は，それぞれ異なるコーパスに含まれる．そのため，コーパスをまたがった連想検索を実現するためには，単一コーパスにおける検索手法を一般化して，複数のコーパスも扱えるようにする必要がある．

単一コーパス上の連想検索を実現するための手順は，第 2 章で述べたように，次のようなものであった．例えば，あるコーパス A に含まれる文書群をクエリとして，同じコーパス A の中から関連する文書群を得るには，まず，コーパス A 上の文書群（クエリ）の特徴語を抽出し，次に，その特徴語に関するコーパス A 上の関連文書群を求めればよかった．

複数コーパスにおける連想検索も，基本的な考え方は，単一コーパスの場合と同じである．すなわち，例えば，あるコーパス A に含まれる文書群（クエリ）から，別のコーパス B に含まれる関連文書群を検索したい場合（図 2）には，クエリとなる文書群について，特徴語群を抽出し，コーパス B の中から，その特徴語群の関連文書群を求めればよい．

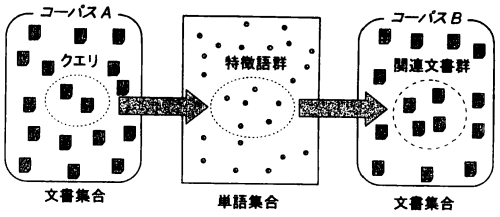

図 2. 複数コーパスにおける連想計算

3.3. 連想計算サーバー

「想・IMAGINE」が統合の対象とするコーパスはネットワーク上に分散しているので，これらのコーパスを統合するためには，「想・IMAGINE」と個々のコーパスとの間で通信を行うための，何らかの仕組みが必要である．我々は，個々のコーパスを管理するための「連想計算サーバー」GETAssoc[2]を開発することで，「想・IMAGINE」とコーパス間の通信を実現した．

GETAssoc は，汎用連想計算エンジン GETA とほぼ同様の連想計算機能を Web サービスとして提供するアプリケーションである．GETAssoc を用いれば，コーパスの追加と削除や，登録済みのコーパスに対する連想計算などを，ネットワーク経由で行うことができる．

コーパスへのアクセス方法を統一するため，「想・IMAGINE」では，統合の対象とするコーパスをまず GETAssoc に登録し，コーパスにアクセスする際には必ず GETAssoc を経由するようにした．

このように，GETAssoc を用いれば，複数のコーパスにおける連想計算（4.1 節）を，ネットワーク上で実現することができ，したがって，「想・IMAGINE」の基本的な機能を実装できる．我々は，さらに，「想・IMAGINE」を真に実用的な情報サービスにするために，上述の基本機能に加えて，以下に述べる 2 つの機能を GETAssoc に追加した．

一つは，典型的な一連の連想計算の組み合わせを，1 回のアクセスでまとめて行えるようにしたことである．この機能は，連想計算に伴う通信コストを削減するためのものである．例えば，文書群から関連文書群を求める連想検索には，少なくとも 2 回の連想計算（文書群から単語群，単語群から文書群）が必要であるが，GETAssoc においては，このような典型的な連想計算の組み合わせを，1 回のアクセスにまとめて行うことができる．

もう一つは，コーパスのカタログ機能であり，登録済みのコーパスのリストを GETAssoc 内部に保持し，そのリストをネットワーク経由で参照できるようにした．コーパスのカタログ機能を導入することの利点は，

「想・IMAGINE」が統合するコーパス全体の管理と，個々のコーパスの管理とを切り分けられることである．すなわち，「想・IMAGINE」は統合対象のコーパスが登録されている GETAssoc のリストを保持するだけで，統合するコーパス全体を管理できるようになる．一方，「想・IMAGINE」に登録済みのいずれかの GETAssoc にアクセスしさえすれば，「想・IMAGINE」が保持するリストを変更せずとも，個々のコーパスの追加や更新，削除を柔軟に行うことができる．

3.4. システム構成

「想・IMAGINE」のシステム構成を図 3 に示す．「想・IMAGINE」は Web アプリケーションであり，フロント部分は，Web ブラウザからクエリを受け取って検索結果を返す．バックグラウンド部分は，Web ブラウザから受け取ったクエリに基づいて，GETAssoc に問い合わせを行う．GETAssoc は，「想・IMAGINE」から渡された問い合わせに応じて，対象となるコーパスの上で連想計算を行い，その結果を「想・IMAGINE」に返す．

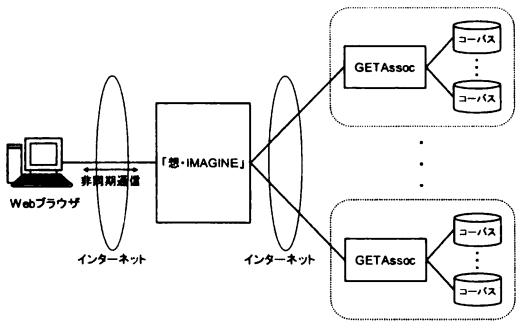

図 3．システム構成図

「想・IMAGINE」では，フロント部もバックグラウンド部もインターネット経由で通信を行うので，通信遅延が無視できない．そこで，Web ブラウザと「想・IMAGINE」との間の通信を，コーパスごとに（Ajax を用いて）非同期に行う．これにより，「想・IMAGINE」全体の性能が，一部の低速なコーパスに引き摺られ，低下してしまうことを回避できる．すなわち，Web ブラウザは，コーパスごとに非同期に検索結果を受け取るので，結果を受け取ったコーパスから次々に表示をすることができる．ゆえに，たとえ，ネットワーク障害や GETAssoc の稼働サーバーの故障などによって，一部のコーパスの検索結果が得られなくても，それ以外のコーパスについては検索結果を表示することができる．

4. 運用実績

「想・IMAGINE」システムを用いた実運用サービスとして，本に関するコーパスを中心として集めた「想・IMAGINE Book Search」を構築し，2006 年 6 月から一般公開している．統合対象のコーパスは，2008 年 6 月現在，10 個あり，第 2 章で紹介した Webcat Plus や新書マップなどに加え，ジュンク堂書店の在庫情報，ウィキペディア，インターネット新聞 JanJan など，本に関する多様なコーパスに対して，連想計算を行うことができる．公開中のコーパスのうちの 6 個の，文書数とデータサイズを表 1 に示す．

表 1．コーパスの文書数とサイズ

コーパス	文書数	サイズ[MB]
Webcat Plus	10,473,742	6,767
新書マップ・テーマ	1,046	15
新書マップ・本	10,882	15
文化遺産オンライン	7,120	9
Book Town じんぼう	354,468	130
ウィキペディア	491,850	2,475

「想・IMAGINE Book Search」の運用においては，10 個のコーパスを 5 台の GETAssoc 用サーバーに分散して登録している．これら 5 台と，「想・IMAGINE」用の 1 台のサーバー，計 6 台のサーバーにより，「想・IMAGINE Book Search」のサービスを行っている．

「想・IMAGINE Book Search」の応答性能を調べるために，連想検索にかかる時間を測定した（表 2）．クエリには，関連性フィードバックとして，直前の連想検索の結果から 3 つの文書を指定した．ここで，検索時間とは，GETAssoc の内部において，連想検索に要した時間であり，応答時間とは「想・IMAGINE」が Web ブラウザから要求を受け取ってから検索結果を返すまでに要した時間である．

表 2．検索時間と応答時間

コーパス	検索時間[ms]	応答時間[ms]
Webcat Plus	306	373
新書マップ・テーマ	11	76
新書マップ・本	10	72
文化遺産オンライン	7	75
Book Town じんぼう	20	79
ウィキペディア	129	185

測定結果によると，Webcat Plus のような 1000 万件規模のコーパスであっても，応答時間は高々380ms 程度であり，実用的な応答性能を実現している．

なお，応答時間と検索時間の差は，「想・IMAGINE」と GETAssoc 間での，通信と（クエリや検索結果の授受のための）XML の生成・解析に要した時間である．

この差は，いずれのコーパスについても 62ms 程度であった．

また，詳細は述べないが，130 個以上のコーパスを 10 台の連想計算サーバーに分散配置した場合についても，上述とほぼ同等の応答性能が得られることを確認している．

5. 議論

本システムでは，複数コーパスに対する連想計算を統合するために，複数の連想計算サーバーを動的に組み合わせる方式を採用した．しかし，この方式の他にも，以下に述べるように，コーパスを集約する方式（以下，集約コーパス方式と呼ぶ）を考えることができる．実際，我々は，本稿の方式を採用する前に，集約コーパス方式による試作を行ったが，その結果，検索精度，計算コスト，運用コストにおいて，以下に述べる問題が生じるという知見を得た．

集約コーパス方式とは，連想計算を行う前に，あらかじめ全てのコーパスを結合して一つの巨大な「集約コーパス」を作成しておき，この集約コーパスに対して連想計算を行うというものである．集約コーパスに対する連想検索の結果には，全てのコーパスから集められた関連文書が混在するので，検索結果をコーパスごとに表示するために，コーパスを結合する際，各文書に対して，由来コーパスを識別するための特殊なキーワードを追加しておく．追加した特殊なキーワードを必須キーワードとするフィルタリングを行えば，特定のコーパスに由来する関連文書のみを得ることができる．

さて，集約コーパス方式の一つめの問題点は，検索精度の低下である．異なるコーパスを連結すると，単語と文書の分布が崩れてしまうので，類似度尺度（例えば idf : inverse document frequency など）を正しく測れなくなり，検索精度が低下する．例として，神保町の古書店の紹介文書 160 件からなるコーパスを，集約コーパス方式によって他のコーパスに連結した場合を考えてみよう．古書店の紹介文書にはいずれも，「店主」という単語が頻繁に出現するので，古書店コーパス単体の中では，「店主」はどの紹介文書においても特徴的な単語ではない．すなわち，「店主」という単語の idf は小さく，類似度のスコアにあまり影響を与えない．ところが，この古書店コーパスを，「店主」という単語のほとんど出現しない他のコーパスと連結すると，結果としてできあがった巨大な集約コーパスの中では，「店主」という単語の idf は大きくなり，類似度のスコアに大きな影響を与えるようになってしまう．一方，本稿で採用した分散連想計算サーバー方式では，コーパスごとに連想計算を行うので，このような問題は生じない．

二つめの問題点は，計算コストの増大である．集約コーパス方式においては，集約するコーパスの個数が多くなると集約コーパスが巨大化するが，巨大なコーパスを扱うためには，大容量の主記憶装置を備える高性能な（したがって高価な）計算機が必要である．また，一部のコーパスのみに対する連想計算を行いたい場合であっても，集約コーパス全体に対する連想計算を行うので，無駄な計算コストがかかるという問題点もある．一方，分散連想計算サーバー方式では，コーパスは複数の計算機に分散することができ，また，連想計算は必要なコーパスに対してのみ行われるので，このような問題は生じない．

三つめの問題点は，管理コストの増大である．集約コーパス方式では，コーパスを追加，削除，更新するたびに，集約コーパス全体を再構築しなければならない．一方，分散連想計算サーバー方式では，連想計算サーバーのカタログ機能によって，コーパスの管理を柔軟に行うことができる．

6. おわりに

我々は，連想計算サーバー GETAssoc を用いてネットワーク上の異種コーパスへの連想計算を統合することにより，異なるコーパス由来の文書をも関連性フィードバックに指定可能な統合検索システム「想・IMAGINE」を構築した．「想・IMAGINE」は，2006 年から一般公開（http://imagine.bookmap.info/）している実用的な情報システムであり，統合対象のコーパス数は年々増え続けている．今後は，多数のコーパスを検索対象とした場合に，関連性の特に高いコーパスを利用者に提示する機構，ならびに，コーパスのみならず連想計算サーバーをも自動的に登録・削除・検索できるようにする管理機構を考案・構築していく予定である．

文　献

[1] 高野明彦，西岡真吾，他，“汎用連想計算エンジンの開発と大規模文書分析への応用”，http://geta.ex.nii.ac.jp/，2002.

[2] 西岡真吾，“連想計算に基づく分散型知識共有プラットフォーム”，夏のプログラミング・シンポジウム，pp.93-100，2006.

[3] 小池勇治，矢島匡人，高野明彦，絹川博之，“Web サービスを利用した総合電子文書検索システム”，FIT2004，E-003，pp.117-118，2004.

[4] Webcat Plus，http://webcatplus.nii.ac.jp/.

[5] 新書マップ，http://shinshomap.info/.

[6] 文化遺産オンライン，http://bunka.nii.ac.jp/.

[7] BOOK TOWN じんぼう，http://jimbou.info/.