デスクトップ環境における利用者の検索意図に基づく情報獲得システムについての評価実験

金子 洋平† 鷹野 孝典‡ 陳 幸生‡

†‡神奈川工科大学情報学部情報工学科 〒243-0292 神奈川県厚木市下荻野 1030

E-mail: †s055044@cce.kanagawa-it.ac.jp, ‡{takano, chen}@ic.kanagawa-it.ac.jp

あらまし 本稿では，デスクトップ環境におけるファイル群を対象とした，利用者の検索意図や関心に基づく情報獲得システムについて述べる．本システムは，パーソナル・コンピュータ上にあるファイル群を効率的に獲得するために，個々の利用者が所有し，管理するファイル群を対象として，その意味や内容に応じた獲得を実現する．本システムは，比較的小さい計算コストで低次元の文書ベクトル空間を動的にかつ素早く構築することを可能とする．このため，利用者は，デスクトップ環境において検索要求が変化した場合においても，再構築された文書ベクトル空間を用いることにより，現在の意図や関心に応じて，適切な文書ファイルを獲得することができる．本研究では，検索実験により，本システムを用いて，デスクトップ環境における利用者の検索意図や関心に応じて，各文書の意味や内容に基づいた文書ファイル群の獲得が実現可能であることを確認した．

キーワード 文書検索，ベクトル空間モデル，意味的検索，特徴抽出，デスクトップ・サーチ

Experiments of the Document Search System Based on the User's intentions in the Desktop Environment

Yohei KANEKO† Kosuke TAKANO‡ and Xing CHEN‡

†‡Department of Computer and Information Sciences, Kanagawa Institute of Technology 1030 Shimo-ogino, Atsugi-shi, Kanagawa, 243-0292 Japan

E-mail: †s055044@cce.kanagawa-it.ac.jp, ‡{takano, chen}@ic.kanagawa-it.ac.jp

Abstract In this paper, we describe a document search system in the desktop environment based on intentions and interests of users. Our search system is implemented to search efficiently for personal files on the user's personal computer according to their contents and meaning. Our system allows users to re-build the retrieval space quickly according to change of their intentions. Therefore, when users' requirements change in the desktop environment, users can effectively retrieve personal files that match their current intentions and interests by using the re-built retrieval space. In this study, we confirmed the feasibility of our system by means of several experiments.

Keyword Document Retrieval, Vector Space Model, Semantic Search, Feature Extraction, Desktop Search

1. はじめに

パーソナル・コンピュータ(PC)や記憶装置の基本性能が急速に進歩するとともに，大量の電子データが PC 上で，蓄積，管理されている．PC の利用者が，自分の所有するファイル群を，効率よく蓄積し，獲得するために，デスクトップ検索エンジンを含めた個人用の情報管理(personal information management, PIM)ツール[1]が提案され，開発されている．The Stuff I've Seen (SIS) system[2]や，the Beagle system[3]はデスクトップ環境における利用者の活動や，その活動履歴に基づいて PC 上のファイル群を獲得する事を可能とする．商用的なデスクトップ検索エンジンとして，Google Desktop[5]，Spotlight[6]， Windows Search[7]，および Copernic Desktop Search[8]が，PC の利用者の間に広く普及している．しかしながら，これらの PIM ツールは，基本的に利用者が入力した検索キーワードや，それらの単語の出現頻度に基づいた検索を行うものであり，意味や内容に応じたファイル獲得の実現が，今もなお重要な課題となっている．

本稿では，デスクトップ環境における利用者の検索意図や関心に基づいて，PC 上に蓄積されているファイル群を，その意味や内容に応じて検索する事を可能とする，情報獲得システムについて述べる．本システムは，Feature Extraction Model (FEM) [4]に基づいて構築されており，以下，本システムを，FEM Desktop Search Tool (FDST) と呼ぶ．FDST の主な特徴を下記に示す．

(1) 意味的な情報獲得

FDST では，事前に集められたサンプル文書を，その意味や内容に基づいて，複数のクラスタに分類する．検索意図に関連のある単語群が，分類情報を利用することにより，サンプル文書から抽出される．抽出した単語を用いて文書ベクトル空間が生成され，PC 上のファイル群は，その文書ベクトル空間上でベクトル（以下，文書ベクトル）として表現される．生成された文書ベクトル空間は，意味や内容に基づいてそれぞれの文書ベクトルが分類されるという特徴を持っている．検索処理過程において，利用者が問い合わせとなる単語列を入力すると，その問い合わせに対応する部分空間が，文書ベクトル空間より選択され，ノルムの大きい文書ベクトルが獲得される．検索処理の結果に基づいて，利用者は PC 上から自分の必要なファイルを，その意味や内容に基づいて検索することができる．

(2) 利用者の意図や関心の反映

利用者の検索意図や関心に応じた，状況適応的なデスクトップ検索を実現するために，現在の利用者の作業コンテクストを反映することが重要である．例えば，PC 上でよく仕事をする利用者にとって，現在の作業に関係する情報を PC 上から獲得したいと要求が考えられる. 例えば, LSI という単語には，電子工学分野における集積回路を意味するもの（Large Scale Integration）と，情報検索分野における検索方式を意味するもの（Latent Semantic Indexing）がある．もし，PC 上の利用者がデータベースや情報検索関連のプレゼンテーション文書，PDF 文書等を閲覧していれば，利用者の現在の意図や関心は，データベースや情報検索にあると判断できる．FDST では，このような利用者の現在の意図や関心を反映することを目的として，デスクトップ環境において作業している利用者により選択，分類されたサンプル文書に基づいて，文書ベクトル空間を構築する．上記の例を用いると，FDST により，データベースや情報検索関連のプレゼンテーション文書，PDF 文書等をサンプル文書として用いて構築された文書ベクトル空間には，利用者がデータベースや情報検索に関する情報が欲しいという検索意図が反映される．この結果，利用者は LSI という単語を用いて検索を行った場合，情報検索方式である Latent Semantic Indexing，および，他の情報検索方式やデータベース・システム等に関連する文書を獲得することができる．また，FDST は，比較的小さい計算コストで低次元の文書ベクトル空間を生成する事を可能とし，利用者の要求が変化した場合においても，現在の要求に応じて選択されたサンプル文書に基づいて，文書ベクトル空間を動的にかつ素早く構築する．

本研究では，上記の(1)，(2)について，FDST の，意味的な情報獲得，およびデスクトップ環境における利用者の検索意図や関心の反映に関する基本的な性能を確認するための実験を行い，本システムの実現可能性を確認する．

2. 関連研究

2.1. Personal Information Management

利用者が PC 上にある情報を，効率よく蓄積し，獲得するために，デスクトップ検索エンジンを含めた様々な PIM ツール[1]が提案され開発されている．The Stuff I've Seen (SIS) system[2]では，利用者がその情報を既に見ているという情報に基づいて，時間，著者，サムネイル画像，プレビュー画像などの，コンテクスチュアルキューが情報の検索や表示に利用される．The Beagle system[3]は，電子メールや WEB 閲覧などのコンテクスト情報を利用することにより，デスクトップ上にあるファイルへの効率的な情報獲得機能を提供する．SIS システムや Beagle システムなどの，主な共通コンセプトは，デスクトップ環境における利用者の活動や，それらの履歴を利用することにより，利用者が要求するファイルを効率的に獲得する機会を増大させる点にある. また, Google Desktop[5], Spotlight[6], Windows Search[7], および，Copernic Desktop Search[8] などのデスクトップ検索エンジンが提案され, PC の利用者に広く利用されている．これらのデスクトップ検索エンジンは, PC 上で索引付けされた全てのファイルについて，高速に検索する機能を提供する．

多くの PIM ツールは, クローラーコンポーネントと，検索エンジンコンポーネントから構成される．クローラーコンポーネントは, PC 上のファイルの索引付けを行い，ファイル名や著者名や内容，作成日，ファイル形式などの，そのファイルに関する情報を，索引ファイル上にメタデータとして保存する．検索エンジンコンポーネントは，利用者に入力された検索キーワードに基づいてファイルを検索する．しかしながら，これらの PIM ツールは, 利用者に入力された検索キーワードや, それらの単語の出現頻度に基づいて, PC 上のファイル検索を行うため，基本的に，メタデータ中に検索キーワードを含んでいないファイルを獲得することができない．

2.2. 意味的な情報獲得

意味的な情報獲得を実現するために，問い合わせと文書間の随時性に基づいて文書を獲得するには，ベク

トル空間モデルによる検索方式が有効であるとされている[9][10]．Latent Semantic Indexing (LSI)[11]は文書や単語間の等価性や，類似性を計算するために，文書ベクトル空間を作成する．しかしながら，LSI では文書ベクトル空間を作成するために，特異値分解計算を実行するために大きな計算コストを要する．このため，LSI を用いて PC 上にあるギガサイズやテラサイズのファイルを検索する場合，計算コストの問題が発生する．

オントロジーは，ある特定の領域における意味的な関係や is-a/part-of の関係のような概念的な関係を扱う．ファイルをある特定領域に適応されるオントロジーにマッピングすることにより，ファイルを複数のクラスタ上に分類することができる．しかし，基本的にある特定領域上で，効率的なオントロジーを構築するには，人間の知識や経験則が必要とされるので高いコストがかかる．テキスト分類の研究分野においては，テキスト分類の自動化方式が，現代のデータ分析分野において研究されている．しかし，テキスト分類の自動化方式においては，文書を表現するための適切なデータ構造を選択することや計算を最適化するための，適切な関数やアルゴリズムの選択が重要な課題となっている．

3. FEM 方式の概要

本章では，FEM 方式の概要について述べる．詳細は，文献[4]に述べられている．

3.1. 文書ベクトル空間の生成プロセス

3.1.1. サンプル文書

文書ベクトル空間の生成のために，サンプル文書群を用意する．サンプル文書群は，文書の意味や内容に基づいて，人間によりいくつかのクラスタに分類される．同一クラスタ中のサンプル文書群では，それぞれ意味や内容が類似している文書が選択されているので，同一クラスタに属する各サンプル文書中には，似たような単語（以下，特徴単語と呼ぶ）が出現すると仮定される．この仮定のもとで，各特徴単語は，あるクラスタに属するサンプル文書群における出現頻度が高いが，他のクラスタに属したサンプル文書群における出現頻度が低いという性質を持つ．

3.1.2. 検索空間の生成

サンプル文書群 $d_1, d_2, d_3 \cdots, d_m$ に対し，文書を q 個のクラスタに分ける．各クラスタを $C_1, C_2, \ldots, C_q$ で表す．クラスタ中の特徴単語を t とし，クラスタ C_i の特徴単語群を K_i とすると，

$$K_i = \{t \mid t \in C_i \wedge t \in C_j \wedge i \neq j\}(i, j = 1,2,\cdots,q) \quad (1)$$

クラスタ C_i を表現するベクトルを $\mathbf{C}_i$ とし，$\mathbf{C}_i$ の単位ベクトルを $\mathbf{c}_i$ とする．サンプル文書が q 個のクラスタに分類されるとすると，q 個のクラスタベクトル $\mathbf{c}_1$, $\mathbf{c}_2$, …, $\mathbf{c}_q$ が生成される．この q 個の単位クラスタベクトルで構成される空間を，検索空間と呼ぶ．

3.1.3. 文書ベクトルの射影

q 次元の検索空間上へ射影された各文書ベクトルは，クラスタ・ベクトル $\mathbf{c}_i$ を用いて，下記の式で表される．

$$\mathbf{d}_j = \sum_{i=1}^{q} e_{i,j}\mathbf{c}_i \quad (2)$$

文書ベクトル $\mathbf{d}_j$ の要素 $e_{i,j}$ の値は，クラスタ C_i における特徴単語が文書 d_j 中の出現頻度，または，重みつきの出現頻度である．クラスタ C_i の各特徴単語 t_1, t_2, …,t_a が文書 d_j の出現頻度を v_{t1}, v_{t2}, …, v_{ta}，および，それぞれ対応する重み係数を w_{t1}, w_{t2}, …,w_{ta} とすると，

$$e_{i,j} = v_{t_1} + v_{t_2} + \cdots + v_{t_a} \quad (3)$$

あるいは，

$$e_{i,j} = w_{t_1} \times v_{t_1} + w_{t_2} \times v_{t_2} + \cdots + w_{t_a} \times v_{t_a} \quad (4)$$

3.2. 文書検索プロセス

文書検索プロセスは，(1) 指定された問い合わせに基づいた部分空間の選択処理，および，(2) 部分空間上で文書のランキング処理，という 2 つの処理により実現される．部分空間は，問い合わせに基づき，文書ベクトル空間上で選択される．q 次元の文書ベクトル空間より選択される部分空間 S は，v 次元のベクトル空間である．v 次元部分空間 S は，v 個のクラスタに相関を持つ．次に，各文書は，部分空間 S 上における各文書ベクトルのノルムを算出し，ランキングされる．

4. FEM Desktop Search Tool

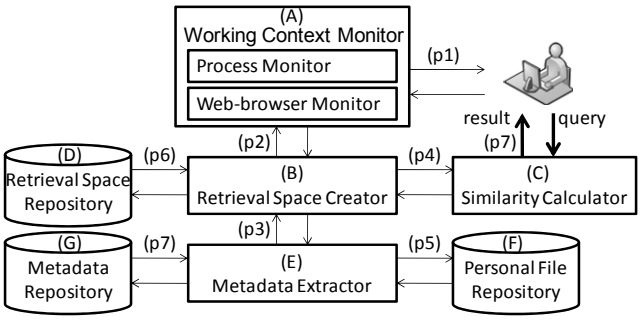

図 1 FDST の概要図

Mac OS X 上で JAVA 言語を用いて，FEM Desktop Search Tool (FDST)の実装を行った．FDST は，デスクトップ環境における利用者の意図や関心に基づいた，情報獲得を実現する．FDST の概要図を図 1 に示す．FDST では，デスクトップ環境における利用者の現在の作業コンテクストをモニタリングすることにより得

られるサンプル文書群を用いて，文書ベクトル空間を生成する．PC 上の個人ファイル群は，文書ベクトル空間上において，ベクトル形式で表現される．

Step-1.　FDST では，Working Context Monitor (A)が，プレゼンテーション，ビジネス文書，および Web ドキュメントなど，デスクトップ環境における，利用者の現在の作業ファイルを監視 (p1)し，リストアップする． FDST は，リストアップされた現在の作業ファイルをサンプル文書として利用する．利用者は，Working Context Monitor においてリストアップされたサンプル文書について，分類情報を手動で入力する．

Step-2.　Retrieval Space Creator (B)は, Step-1 において分類されたサンプル文書から特徴単語を抽出する．次に，抽出した特徴単語に基づいて文書ベクトル空間を生成する (p2).

Step-3.　PC 上のファイル・リポジトリ(F)をクロールすることにより (p5)，Metadata Extractor (E)はそれぞれのファイルから，文書内容やファイル名などのメタデータを抽出する．そして，それらのメタデータをメタデータ･リポジトリ(G)に格納する．それぞれのファイルの文書内容は，文書ファイル中に出現する単語群やそれらの出現頻度として抽出される．

Step-4.　PC 上の各ファイルは, Step-3 で抽出されたメタデータに基づいてベクトル化され，Step-2 で作成された文書ベクトル空間上に写像される (p3).

利用者が問い合わせ単語列を入力すると，Similarity Calculator (C)は，その問い合わせ単語列とファイル間の相関量を計算し，その計算結果に基づいて，その問い合わせ単語列に対するランキング結果を利用者に出力する(p7).

FDST により，利用者は検索意図の変化に応じて素早く文書ベクトル空間を再構築することができる．このため，デスクトップ環境において，利用者の要求が変化した場合においても，再構築された文書ベクトル空間を用いることにより、利用者の現在の意図や関心を満たすファイル群を効率的に獲得することができる．また，現在の FDST の実装では，Step-2 と Step-3 における単語抽出プロセスにおいて, テキスト文書, PDF, Micrsoft Word，Microsoft Excel，Microsoft PowerPoint など，各ファイル形式のファイルから特徴単語の抽出を行うことが可能となっている．

5. 予備実験

図 2 は，FEM 方式と LSI 方式について，文書ベクトル空間の生成時間を測定した結果を示している．実験プログラムは，Mac OS X 10.5 (Memory: 2GB, CPU: 2.4GHz)上で，Octave 3.0.1 を用いて作成を行った．図 2 のグラフから、空間次元数が 3～30 次元の低次元な場合において, FEM 方式では短時間で空間を生成できることが確認できまる．この結果から，文書ベクトル空間の速さを重視する場合，3 次元～20 次元の文書ベクトル空間を構築することが妥当と考えられる. また，この結果では，60 次元あたりから，LSI 方式よりも、空間生成時間が長くなっているが, これは FEM 方式において，特徴単語から，共通単語を排除する処理（文書ベクトル空間の直交化処理）に時間を要しているためである．しかしながら，FEM 方式では，空間次元数が多くなるにつれて，文書ベクトル空間生成過程で，排除される共通単語の増加により，空間軸を形成する単語群が必要以上に減ってしまうことが予想される．この結果，次元数の大きな文書ベクトル空間を生成した場合，その意味的検索性能が低下してしまう可能性がある．逆に，FEM 方式では，次元数が小さな文書ベクトル空間を用いた場合でも，利用者の意図や関心に基づいた，意味的な情報獲得を実現できる．

以上，文書ベクトル空間の生成時間，および，意味的検索性能の観点から，デスクトップ環境における，利用者の意図や関心に適応的な情報獲得を実現するために，FEM 方式では，3～30 次元の低次元の文書ベクトル空間を構築することが適していると考えられる．

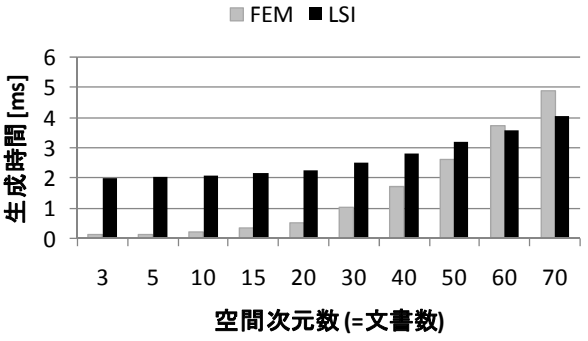

図 2 文書ベクトル空間の生成時間の比較

6. 実験

本実験では，利用者の意図や関心に応じた，意味的な情報検索システムの実現可能性を確認する．本実験では，利用者の PC 上に格納されている 200 件の文書ファイルを検索対象として, 3 章, 4 章で述べた方式により，実験検索システムを実装した．表 1 は，検索対象である各文書ファイルのトピック，文書 ID を示している．表 1 に示すように，各文書は、10 種類のトピックに分類されている．

表 1 各文書のトピック

No.	トピック	文書 ID
1	Coffee	$d_1 - d_{20}$
2	Baseball	$d_{21} - d_{40}$,, d_{90}
3	Basketball	$d_{41} - d_{60}$
4	Caribbean	$d_{61} - d_{80}$
5	Florida	$d_{81} - d_{101}$
6	Football	$d_{102} - d_{121}$
7	Mobile Phone	$d_{122} - d_{141}$
8	Moon	$d_{142} - d_{161}$
9	Tennis	$d_{162} - d_{181}$
10	Swimming	$d_{182} - d_{201}$

実験では，本システムを評価するために，表 2 に示すように2つの文書ベクトル空間 S_1 と S_2 を作成する．各文書ベクトル空間 S_1 と S_2 を作成するためのサンプル文書ファイルは，デスクトップ環境の利用者に，ある時点において閲覧，または編集されていると想定されている．例えば，S_1 を検索に利用する場合，利用者は，coffee bean，mars，および tennis に関する文書ファイルを開いているので，そのような話題に興味があると仮定される．文書ベクトル空間 S_1 は，以下の問い合わせ q_1 と q_2 を用いて検索実験を行い，評価する．また、S_2 は，q_3 と q_4 を用いて検索実験を行う．

q_1：Sharapova　(有名な女性テニス選手)
q_2：Mocha　　(コーヒー豆の一種)
q_3：Mike Piazza (有名な野球選手)
q_4：Hot Spot (インターネットのアクセスポイント)

表 2 文書ベクトル空間 S_1，S_2

空間 Id	軸 Id	サンプル文書 ID (ファイル名)	トピック
S_1	1	d_8 (Coffee Bean.doc)	Coffee Bean
	2	d_{152} (Mars.html)	Mars
	3	d_{194} (Tennis Court.html)	Tennis Court
S_2	1	d_{29} (Baseball.ppt)	Baseball
	2	d_{87} (Bluetooth.html)	Bluetooth
	3	d_{126} (FlagOfFlorida.html)	Flag of Florida

表 3，4 に，それぞれの問い合わせに対する上位 20 件の検索結果を示す．表 3，4 において，右に*のある文書は，サンプル文書ファイルである．また，網掛けで示される文書は，問い合わせに関連がある文書であることを示している．表 5 は，Mac OS X 上の標準検索ツールである Spotlight[6]により，同じ問い合わせ q_1 ～q_4 を用いた場合の検索結果を示している．Spotlight は，基本的に利用者が入力した検索キーワードや，それらの単語の出現頻度に基づいて，デスクトップ上にあるファイルの検索を行う．本システムと Spotlight の検索結果を比較することにより，本システムを用いた検索では，意味と内容に基づいて文書ファイルを検索することが可能であるため，Spotlight を用いた場合より，多くの関連文書ファイルを検索できることが確認できる．q_1 の検索結果において，文書 d_{203} は優れた男性テニス選手について述べており，q_1 の問い合わせ単語である Sharapowa は，女性のトップ・テニス選手の 1 人であるので，d_{203} は q_1 に関連があることが確認できる．しかし，Spotlight では，d_{203} がメタデータ（ファイル名，作成者名，内容等）中にシャラポワ（女性選手）を含んでいないため，d_{203} を検索することができない．一方，本システムにおいては，S_1 を用いて，意味や内容に応じた相関量計算することにより，利用者の興味に応じて，d_{203} を検索することができる．

表 3 検索結果 (q_1，q_2)

q_1			q_2		
順位	文書 ID	相関量	順位	文書 ID	相関量
1	d_{192}	1.000	1	d_{192}	1.021
2	d_{57}	0.873	2	d_9	1.008
3	d_{24}	0.856	3	d_{15}	0.930
4	d_{194}*	0.807	4	d_{57}	0.899
5	d_{201}	0.744	5	d_{24}	0.898
6	d_{43}	0.661	6	d_{18}	0.896
7	d_{106}	0.588	7	d_{19}	0.826
8	d_{195}	0.570	8	d_{194}	0.807
9	d_{203}	0.541	9	d_{31}	0.786
10	d_{63}	0.534	10	d_{201}	0.769
11	d_{46}	0.489	11	d_8*	0.706
12	d_{107}	0.477	12	d_{43}	0.666
13	d_{112}	0.474	13	d_{140}	0.648
14	d_{110}	0.447	14	d_{106}	0.600
15	d_{109}	0.432	15	d_7	0.579
16	d_4	0.431	16	d_{195}	0.577
17	d_{60}	0.430	17	d_{203}	0.559
18	d_{47}	0.430	18	d_4	0.555
19	d_{133}	0.429	19	d_6	0.548
20	d_{27}	0.427	20	d_{133}	0.546

また，S_2 を検索に利用する場合，利用者は野球，移動通信，その他に興味があると仮定される．q_4 の検索結果において，文書 d_{131} は全地球位置測定システム（GPS）について述べており，q_4 の問い合わせ単語である Hot Spot はインターネットのアクセスポイントであるので， d_{131} は移動通信に関して q_4 に関連があることがわかる．しかし，Spotlight では，d_{131} がメタデータ中に Hot Spot を含んでいないため，d_{133} を検索することができない．一方，本システムにおいては，S_2 を用いて，意味や内容に応じた相関量計算することにより，利用者の目的に応じて，d_{131} を検索することができる．

これらの実験結果から，本システムにより，意味や内容に応じて, PC 上の文書ファイル獲得可能なことが確認できる．また，本実験結果は，デスクトップ環境

において,利用者の要求が変化した場合(例えば,S_1 から S_2）においても，再構築された文書ベクトル空間を用いることにより適応し，利用者の現在の意図や関心に応じて，適切な文書ファイルを獲得できることを示している．

表 4 検索結果（q_3，q_4）

q_3			q_4		
順位	文書 ID	相関量	順位	文書 ID	相関量
1	d_{24}	1.0281	1	d_{87}*	1.000
2	d_{126}	1.000	2	d_{126}*	1.000
3	d_{57}	0.949	3	d_{140}	0.585
4	d_{43}	0.815	4	d_{133}	0.560
5	d_{27}	0.787	5	d_{54}	0.556
6	d_{41}	0.725	6	d_{91}	0.548
7	d_{33}	0.677	7	d_{138}	0.539
8	d_{106}	0.667	8	d_{101}	0.520
9	d_{140}	0.640	9	d_{99}	0.471
10	d_{133}	0.608	10	d_{131}	0.457
11	d_{138}	0.557	11	d_{88}	0.440
12	d_{107}	0.542	12	d_{66}	0.440
13	d_{29}*	0.541	13	d_{142}	0.438
14	d_{90}	0.515	14	d_{102}	0.413
15	d_{113}	0.504	15	d_{109}	0.399
16	d_{131}	0.468	16	d_{185}	0.376
17	d_{110}	0.464	17	d_{53}	0.374
18	d_{31}	0.457	18	d_{132}	0.370
19	d_{192}	0.454	19	d_{68}	0.357
20	d_{46}	0.452	20	d_{110}	0.352

表 5 検索結果 (Spotlight)

問い合わせ	文書 ID	トピック
q_1	d_{185}	Tennis US Open 2006
	d_{192}	Tennis
q_2	d_5	Café Menu Board
	d_7	Coffea Arabica
	d_9	Coffee
	d_{13}	Drip Brew
	d_{15}	Espresso
	d_{18}	History of Coffee
	d_{19}	Instant Coffee
q_3	d_{90}	Florida Marines
q_4	d_{134}	Telephone
	d_{142}	Hotspot (Wi-Fi)

7. おわりに

本稿では，利用者のデスクトップ環境におけるファイル群を対象とした情報獲得システム FDST（FEM Desktop Search Tool）について述べ，検索実験により，本システムが，デスクトップ環境における利用者の検索意図や関心に沿った文書群の獲得が可能であることを確認した．

FEM 方式では，利用者が手動でサンプル文書の選択・分類を行う．この場合，利用者が適切にサンプル文書の分類できる場合でも,利用者の意図によっては、類似する文書を異なるカテゴリに，無関係な文書を同じカテゴリに分類する可能性がある．この結果，利用者の意図をより反映できていると考えることができるものの，検索精度が低下してしまう可能性も考えられる．これに対処するために，文書ベクトル空間の検索精度を改善するための学習機能を，本システムに実装していく予定である．

さらに，利用者の嗜好に応じた情報獲得の精度向上を目的とした本システムの拡張において，デスクトップ環境におけるスケジュール管理ソフトに記録されている予定表等の，利用者の興味や関心を判断可能な他の情報源からメタデータを抽出する機能や，利用者の所有する PDA や携帯電話から得られる個人情報と連携する機能等，よりリッチな利用者コンテクストの取得を可能とする機能群の実現が今後の課題として挙げられる．

文　献

[1] J. Teevan, W. Jones and B. B. Bederson, ``Personal infromation management," *Communications of the ACM, vol. 49, No. 1*, 40-43, 2006, 40-43.

[2] S. Dumais, E. Cutrell, JJ Cadiz, G. Jancke, R. Sarin and D. C. Robbins, ``Stuff I've seen: A system for personal information retrieval and re-use," *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2003, 72–79.

[3] W. Nejdl and R. Paiu, ``Desktop search. How Contextual information influences search results and rankings," Proceedings of the IRiX Workshop at the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 2005.

[4] Chen, X. and Kiyoki, Y., ``A Dynamic Retrieval Space Creation Method for Semantic Information Retrieval," Information Modelling and Knowledge Bases, Vol.XVI, IOS Press, pp.46-63, 2005.

[5] Google, Inc., Google Desktop, http://desktop.google.com/, 2008.

[6] Apple Computer, Inc., Spotlight, http:// www.apple.com/macosx/features/300.html#spotlight, 2008.

[7] Microsft Corporation, Windows Search 4.0, http://www.microsoft.com/windows/products/winfamily/desktopsearch/default.mspx, 2008.

[8] Copernic, Copernic desktop search, http://www.copernic.com/, 2007.

[9] Baeza-Yates R., Ribeiro-Neto, B., ``Modern Information Retrieval," Addison Wesley, 1999.

[10] Wong, S. K. M., Ziarko, W., Wong, P. C. N., ``Generalized Vector Space Model in Information Retrieval, SIGIR," pp.18-25, 1985.

[11] Deerwester, S. C., Dumais, S. T., Furnas, G.W., Landauer, T. K. and Harshman, R. A., ``Indexing by latent semantic analysis," Journal of the American Society for Information Science, Vol. 41, No. 6, pp. 391-407, 1991.