自 然 言 語 処 理 140－6
（2000．11．21）

コンセプト・プロジェクションにおける関連性フィードバックを用いた概念ベクトルの更新手法

佐々木 稔　獅々堀 正幹　北 研二

徳島大学 工学部

〒770-8506　徳島市南常三島町 2-1

{sasaki, bori, kita}@is.tokushima-u.ac.jp

概要

関連性フィードバックは，ユーザが検索結果の各文書が検索質問に関連があるか，関連がないかの判定を行い，この判定評価の情報を用いて初期検索要求に反映させる手法で，対話的な検索において非常に有効である．本稿では，テストコレクションに用意されている，検索質問に対してどの文書が関連しているかという情報から，我々の提案した次元圧縮手法であるコンセプト・プロジェクションに必要な概念ベクトルのパラメータをフィードバックさせる手法を提案し，検索精度の改善を試みる．

Updating Method for the Concept Vectors Based on Relevance Feedback Using Concept Projection

Minoru Sasaki　Masami Shishibori　Kenji Kita

Faculty of Engineering, Tokushima University

2-1, Minami-josanjima, Tokushima 770-8506, Japan

{sasaki, bori, kita}@is.tokushima-u.ac.jp

Abstract

Relevance feedback, which modifies queries using user's judgements of the relevance of a few highly-ranked documents, is very effective for increasing the performance of information retrieval systems historically. In this paper, we propose an updating method for concept vectors which is necessary for the concept projection by using the information on the judgements in the test collection. The experiment conducted on various feedback methods show that this methods are effective to the retrieval performance.

1 はじめに

近年，インターネットの普及とともに，個人でWWW (World Wide Web) を代表とするネットワーク上の大量の電子データやデータベースが取り扱えるようになり，膨大なテキストデータの中から必要な情報を取り出す機会が増加している．しかし，このようなデータの増加は必要な情報の抽出を困難とする原因となる．この状況を反映し，情報検索，情報フィルタリングや文書クラスタリング等の技術に関する研究開発が盛んに進められている．

このような情報検索システムの中でよく使われている検索モデルに，ベクトル空間モデル [6] がある．ベクトル空間モデルは，文書と検索要求を様々な特徴を要素とする多次元空間ベクトルとして表現する方法である．このベクトル空間モデルを用いた検索システムを新聞記事などの大量の文書データに対して適用した場合，文書データ全体に存在するタームの数が非常に多くなるため，文書ベクトルは高い次元を持つようになる．しかし，ひとつの文書データに存在するタームの数は文書データ全体のターム数に比べると非常に少なく，文書ベクトルは要素に0の多い，スパースなベクトルになる．このような文書ベクトルを用いて類似度を計算する際には，検索時間の増加や文書ベクトルを保存するために必要なメモリの量が大きな問題となる．

上記の問題を解決するベクトル空間モデルの次元圧縮手法に，我々が提案したコンセプト・プロジェクションが存在する [11]．コンセプト・プロジェクションは，クラスタリングなどにより得られる，文書の内容を表した概念ベクトルと文書ベクトルの内積を計算することで，次元圧縮を行う手法である．これにより，文書ベクトルは用意した概念ベクトルの数に次元圧縮され，内積計算のみを行うため少ない時間で圧縮ができる．また，検索性能に関しても，次元圧縮を行わないベクトル空間モデルよりも改善され，同様な次元圧縮手法であるLSI(Latent Semantic Indexing) に匹敵する検索性能が得られている．

本稿では，我々の提案したコンセプト・プロジェクションの応用として，関連性フィードバックによる検索モデルの更新手法について述べる．関連性フィードバックは検索結果の各文書が適合であるか，不適合であるかをユーザに判定させ，この判定評価の情報を用いて初期検索要求に反映させる手法である．これに対し，提案するフィードバック手法は，判定評価の情報を初期検索要求に反映させるのではなく，コンセプト・プロジェクションの概念ベクトルに反映させている．これにより，更新された概念ベクトルから検索要求や検索対象となる文書ベクトルの次元圧縮が行われるため，フィードバック学習の影響が検索要求だけでなく検索対象にも反映させることができる．関連性フィードバックによる様々な概念ベクトルの更新手法を提案し，テストコレクションによる検索実験結果を示し，更新手法の比較を行う．

2 コンセプト・プロジェクションによるベクトルの次元圧縮

本節では，コンセプト・プロジェクションを用いたベクトル空間モデルの概観について述べる．まず初めに，コンセプト・プロジェクションに必要であるクラスタリングによって得られる概念ベクトルについて述べる．

2.1 概念ベクトル

ベクトルの集合をベクトル空間にプロットしたとき，同質のベクトルが多く存在する場合を除いて，いくつかのグループに分かれる．このようなグループはクラスタと呼ばれ，類似した内容をもつベクトルの集合が形成される．概念ベクトルはこのようなクラスタに属するベクトルの重心を求めることにより得られ，そのクラスタの内容を表す代表ベクトルである．

概念ベクトルを求める例として，正規化された N 個のベクトル $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$ を，異なる s $(s < N)$ 個のクラスタ $\pi_1, \pi_2, \cdots, \pi_s$ にクラスタリングすることを考える．このとき，ひとつのクラスタ π_j に含まれるベクトル x_i の平均である重心 $\mathbf{m}_j$ は以下のように表される．

$$\mathbf{m}_j = \frac{1}{n_j} \sum_{\mathbf{x}_i \in \pi_j} \mathbf{x}_i \qquad (1)$$

ここで n_j はクラスタ π_j に含まれるベクトルの数を表す．ベクトルの重心は単位長にはなっていないので，そのベクトルの長さで割ることにより概念ベクトル $\mathbf{c}_j$ を得る．

$$\mathbf{c}_j = \frac{\mathbf{m}_j}{\|\mathbf{m}_j\|} \tag{2}$$

2.2 コンセプト・プロジェクション

コンセプト・プロジェクションは，ひとつの文書データを n 次元空間上のベクトル $\mathbf{u}$ として表現するとき，このベクトルを k $(k < n)$ 次元空間に射影する手法である [11]．その際，クラスタリングなどにより求められた n 次元である k 個の概念ベクトル $\mathbf{r}_1, \cdots, \mathbf{r}_k$ を用意し，これらのベクトルと n 次元ベクトル $\mathbf{u}$ の内積，

$$\mathbf{u}'_1 = \mathbf{r}_1 \cdot \mathbf{u}, \cdots, \mathbf{u}'_k = \mathbf{r}_k \cdot \mathbf{u} \tag{3}$$

をそれぞれ計算する．その結果，k 次元に圧縮した $\mathbf{u}'_1, \cdots, \mathbf{u}'_k$ を要素とするベクトルが得られる．

次元圧縮に必要なベクトル $\mathbf{r}_1, \cdots, \mathbf{r}_k$ を列ベクトルとする $n \times k$ の行列 $\mathbf{R}$ を用いると，求める k 次元ベクトルは

$$\mathbf{u}' = \mathbf{R}^T \mathbf{u} \tag{4}$$

となり，コンセプト・プロジェクションは行列計算のみの簡単な形で表現することができる．この行列 $\mathbf{R}$ が任意の正規直交行列のとき，すなわち，行列 $\mathbf{R}$ の列ベクトルがすべて単位ベクトルで，かつ，相異なる列ベクトルが互いに直交していれば，コンセプト・プロジェクションは射影前後におけるベクトル間距離を近似的に保存する特性を持っている．

概念ベクトルからなる行列 $\mathbf{R}$ を求めるために，球面 k 平均アルゴリズム [3] と呼ばれるクラスタリング手法を用いる．球面 k 平均アルゴリズムは，各クラスタ $\pi_j (1 \leq j \leq s)$ の密度を

$$\sum_{\mathbf{x}_i \in \pi_j} \mathbf{x}_i^T \mathbf{c}_j \tag{5}$$

とし，クラスタの結合密度の和を目的関数とし，この目的関数が局所的に最大となるまで，高い次元でスパースな文書データ集合がクラスタリングされる．

$$D = \sum_{j=1}^{s} \sum_{\mathbf{x}_i \in \pi_j} \mathbf{x}_i^T \mathbf{c}_j \tag{6}$$

球面 k 平均アルゴリズムでは，ユークリッド空間内でベクトル間のなす角の余弦を類似度とし，多次元空間の単位円を分割することによりクラスタリングを行う．これにより，文書ベクトルの集合は指定した数の部分集合に分割され，各クラスタの中心を計算することで，容易に概念ベクトルを作ることができる．さらに，このアルゴリズムは文書ベクトルのスパースさを逆に利用して高速に収束する利点を持ち，得られる概念ベクトルは特異値分解を用いたものに非常に近いことが示されている [3]．しかし，球面 k 平均アルゴリズムにより得られる概念ベクトルは一般的に直交性を満たしているとは限らないため，概念ベクトルをランダム・プロジェクションに適用するには疑問が生じる．先に述べたように，距離を保存するには正規直交性を満たすベクトルを利用する必要があるが，この概念ベクトルをランダム・プロジェクションに適用する場合，直交性を満たしていないとしても独立であれば，任意の行列においても十分に距離を保存する可能性のあることが示されている [1]．球面 k 平均アルゴリズムでは，内容的に似通ったベクトルをクラスタとしてまとめるため，原理的には独立した概念ベクトルを生成すると考えられる．このため，直交性に関して，概念ベクトルをランダム・プロジェクションに適用するのは問題ないと考えられる．

2.3 球面 k 平均アルゴリズム

2.2 節で示した目的関数 D を最大にするように，ベクトルの集合を反復法によりクラスタリングする．文書ベクトル $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$ を s 個のクラスタ $\pi_1^\star, \pi_2^\star, \cdots, \pi_s^\star$ に分割するためのアルゴリズムを以下に示す．

1. すべての文書ベクトルを s 個のクラスタに任意に分割する．これらの部分集合を $\{\pi_j^{(0)}\}_{j=1}^s$ とし，これより求められた概念ベクトルの初期集合を $\{\mathbf{c}_j^{(0)}\}_{j=1}^s$ とする．また，t を繰り返しの回数とし，初期値は $t = 0$ である．

2. 各文書ベクトル $\mathbf{x}_i (1 \leq i \leq N)$ に対し，余弦が最も大きい，最も文書ベクトルに近い概念ベ

クトルを見つける．このとき，すべての概念ベクトルは正規化されているので，余弦は文書ベクトル $\mathbf{x}_i$ と 概念ベクトル $\mathbf{c}_j^{(t)}$ の内積を求めることと同値である．これにより，前回の繰り返しで求めた概念ベクトル $\{\mathbf{c}_j^{(t)}\}_{j=1}^{s}$ から，文書ベクトルが新たな部分集合 $\{\pi_j^{(t+1)}\}_{j=1}^{s}$ に分割される．

$$\pi_j^{(t+1)} = \{\mathbf{x}_i : \mathbf{x}_i^T \mathbf{c}_j^{(t)} \geq \mathbf{x}_i^T \mathbf{c}_l^{(t)}\} \quad (1 \leq l \leq N,\ 1 \leq j \leq s) \qquad (7)$$

ここで，$\pi_j^{(t+1)}$ は概念ベクトル $\mathbf{c}_j^{(t)}$ に近いすべての文書ベクトルの集合とする．

3. 新たに導かれた概念ベクトルの長さを正規化する．

$$\mathbf{c}_j^{(t+1)} = \frac{\mathbf{m}_j^{(t+1)}}{||\mathbf{m}_j^{(t+1)}||}, \quad (1 \leq j \leq s) \qquad (8)$$

ここで，$\mathbf{m}_j^{(t+1)}$ はクラスタ $\pi_j^{(t+1)}$ の文書ベクトルの重心を表す．

4. 目的関数 $D^{(t+1)}$ の値を求め．前回の繰り返しにおける目的関数の値 $D^{(t)}$ との差を計算する．このとき，

$$||D^{(t)} - D^{(t+1)}|| \leq 1 \qquad (9)$$

を満たす場合，$\pi_j^{\star} = \pi_j^{(t+1)}$，$\mathbf{c}_j^{\star} = \mathbf{c}_j^{(t+1)}$ $(1 \leq j \leq s)$ とし，アルゴリズムを終了する．停止基準を超えていない場合は，t に 1 を加え，ステップ 2 に戻る．ここで，停止基準における目的関数の差は，文書数が約 4000 で，クラスタの数が 8 よりも大きい場合，収束した時の目的関数は 1000 を超えることがこれまでの研究で報告されている [3]．このため，繰り返しでの 1 以下の差は無視できるとし，便宜的に 1 という値を設定した．

3 フィードバックによる概念ベクトルの更新手法

情報検索システムはユーザに検索結果を提示し，ユーザはその結果に対して関連のある文書であると判定する．適合性フィードバックはその判定結果を元に，システムの挙動を変化させるようにパラメータを調節し，システムに反映させるものである．この関連性フィードバックがパラメータを調節する対象としては，検索質問，検索対象となる文書，または検索モデルが考えられる．よく知られているフィードバックに検索質問拡張があるが，検索質問のみを修正することは，システムに対して長期的な効果が得られるとは限らない [7][9]．本節では，テストコレクションに用意されている，検索質問に対してどの文書が適合しているかという情報を用いて，コンセプト・プロジェクションにおける概念ベクトルのパラメータ更新手法を提案し，検索精度の改善を試みる．

概念ベクトルのパラメータ更新の基本は，上位に検索された関連のある，または関連のない文書ベクトルをそれぞれ k 個の概念ベクトル $\mathbf{r}_1, \cdots, \mathbf{r}_k$ に加えて更新をする．これにより，概念ベクトルの持つ文書の内容がより検索質問の内容に近づき，検索精度が向上することが期待できる．また，検索結果からコンセプト・プロジェクションの概念ベクトルを，より検索したい内容の概念ベクトルに変更し，次元圧縮後は検索要求だけでなく，検索対象にもフィードバックを行うことができる．

具体的に，文書ベクトルを概念ベクトルに加える手法として以下の 5 種類を考慮し，関連のある場合，ない場合に対してこれらの手法を組み合わせて実験を行う．

1. 文書ベクトルとの内積が最も大きい概念ベクトル $\mathbf{r}_l$ を見つけ，その概念ベクトルに文書ベクトルを加えて，正規化を行う．
2. 文書ベクトルとの内積がある閾値 σ 以上の概念ベクトル集合を見つけ，それらの概念ベクトルにそれぞれ文書ベクトルを加えて，正規化を行う．
3. システムが検索した関連のある，または関連のない文書集合の重心との内積が，最も大きい概念ベクトル $\mathbf{r}_l$ を見つけ，その概念ベクトルに文書ベクトルを加えて，正規化を行う．
4. システムが検索した関連のある，または関連の

ない文書集合の重心との内積が，ある閾値 γ 以上である概念ベクトル集合を見つけ，それらの概念ベクトルにそれぞれ文書ベクトルを加えて，正規化を行う．

5. 適合性フィードバックの基本である手法で，i 回目の検索質問ベクトル Q_i から $i+1$ 回目の検索に向けて索引語の重みを修正した検索質問ベクトル Q_{i+1} を求める式を以下のように表した手法である [5]．

$$Q_{i+1} = Q_i + \alpha \sum_{\mathbf{x} \in R_r} \mathbf{x} - \beta \sum_{\mathbf{y} \in R_n} \mathbf{y} \qquad (10)$$

ここで，R_r は i 回目において検索された関連文書集合，R_n は i 回目において検索された関連のない文書集合である．また，α，β は定数であり，それぞれ関連文書，関連のない文書をどの程度重要視するかを調整する．

6. フィードバックは先の 5 に示した手法で行い，検索質問ベクトルを修正した後で，コンセプト・プロジェクションを行い次元圧縮を行う．

4 実験

4.1 実験の概要と結果

コンセプト・プロジェクションを用いたフィードバック検索モデル構築し，その検索性能を示す実験を行った．実験には，情報検索システムの評価用テストコレクションである MEDLINE を利用した．MEDLINE は医学・生物学分野における英文の文献情報データベースで，検索の対象となる文書の件数は 1033 件で，約 1Mbyte の容量を持つテキストデータである．また，MEDLINE には 30 個の評価用検索質問文とそれらの関連記事が用意されている．

まず，前処理として MEDLINE の記事全体から抽出した 1033 件の記事から一般的な 439 個の英単語をストップワードに指定して，文書の内容と関係のほとんどない単語は削除した．この前処理の結果，4329 個のタームが索引語として抽出された．

これらの索引語を要素とする文書ベクトルを作成するとき，索引語の頻度に重みを加えた数値をベクトルの要素とする．数多く提案されている重みづけ手法で，今回の実験では以下の式で定義された対数エントロピー重み [2] を用いた．L_{ij} は j 番目の文書に対する i 番目のタームへの重み，G_i は文書全体に対する i 番目のタームへの重みを表す．

$$L_{ij} = \begin{cases} 1 + \log f_{ij} & (f_{ij} > 0) \\ 0 & (f_{ij} = 0) \end{cases} \qquad (11)$$

$$G_i = 1 + \sum_{j=1}^{n} \frac{\frac{f_{ij}}{F_i} \log \frac{f_{ij}}{F_i}}{\log n} \qquad (12)$$

ここで，n は全文書数，f_{ij} は j 番目の文書に出現する i 番目のタームの頻度，F_i は文書集合全体における i 番目のタームの頻度を表す．

得られた文書ベクトルから，球面 k 平均アルゴリズムを用い，これらの文書ベクトルより指定した 500 の概念ベクトル作成する．作成した概念ベクトルを結合した行列に対し，ランダム・プロジェクションを行い，文書ベクトル，検索質問ベクトルの次元を 500 に削減する．次元の削減されたベクトルに対し，内積の計算を行い，その値を各文書ベクトルへの検索スコアとする．これらのスコアのうち，上位 50 文書を検索結果として，出力する．

検索システムの精度の評価には，一般的に用いられている適合率 (Precision) と再現率 (Recall) を用いた [4][8]．

$$\text{Recall} = \frac{\text{システムが出力した適合文書数}}{\text{全適合文書数}} \qquad (13)$$

$$\text{Precision} = \frac{\text{システムが出力した適合文書数}}{\text{システムが出力した文書数}} \qquad (14)$$

再現率と適合率は，それぞれ個別に用いて，システム評価を行うことができるが，本実験では，一般にランクづけ検索システムの評価に用いられる再現率-適合率曲線を用い，システムの評価を行った．本稿の検索システム評価には，繰り返しの回数に従って，平均適合率がどのように変化しているのか示し，その中でもっともフィードバックの効果の高かった手法についての、全質問に対する各再現率での平均を計算した再現率-適合率曲線を示すことにより行った．この概要の元で，先に示した手法を用いて様々な実験を行った結果，フィードバックの有効な効果が得られた手法の，繰り返しの回数による平均適合率の変化を表 1 から 8 に表す．

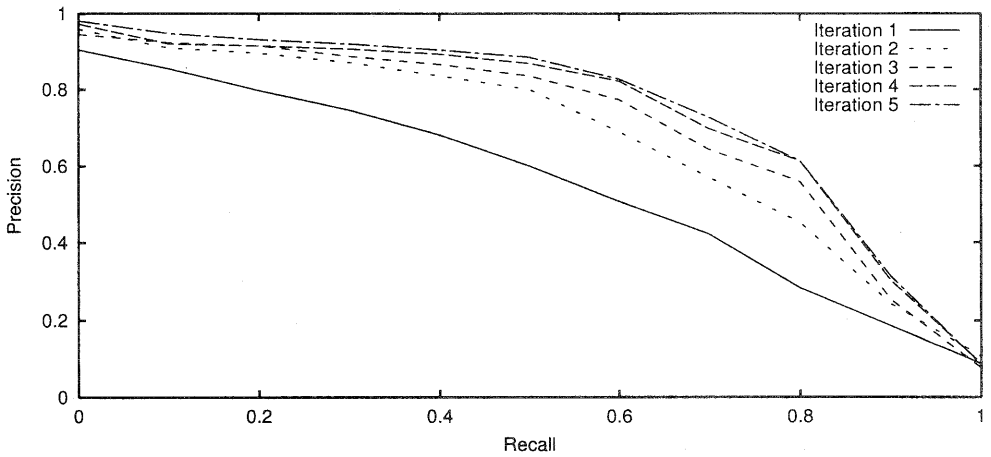

図 1: 表 7 における再現率-適合率曲線

表 1: 各繰り返し回数での平均適合率 1

	適合:手法 3, 不適合:手法 4($\gamma = 0.3$)
繰り返し回数	平均適合率
1	0.5552
2	0.6715
3	0.6723
4	0.6782
5	0.6861

表 2: 各繰り返し回数での平均適合率 2

	適合:手法 4($\gamma = 0.3$), 不適合:なし
繰り返し回数	平均適合率
1	0.4893
2	0.6903
3	0.7017
4	0.6635
5	0.6892

表 3: 各繰り返し回数での平均適合率 3

	適合:手法 4($\gamma = 0.3$), 不適合:手法 2($\sigma = 0.5$)
繰り返し回数	平均適合率
1	0.5216
2	0.6885
3	0.7303
4	0.6975
5	0.6807

表 4: 各繰り返し回数での平均適合率 4

	適合:手法 4($\gamma = 0.3$), 不適合:手法 1
繰り返し回数	平均適合率
1	0.5118
2	0.6536
3	0.7314
4	0.7079
5	0.7289

表 5: 各繰り返し回数での平均適合率 5

	適合:手法 4($\gamma = 0.4$), 不適合:手法 2($\sigma = 0.5$)
繰り返し回数	平均適合率
1	0.5434
2	0.6714
3	0.7140
4	0.7419
5	0.7532

表 6: 各繰り返し回数での平均適合率 6

	手法 5($\alpha = 1.0$, $\beta = 0.5$)
繰り返し回数	平均適合率
1	0.4936
2	0.8662
3	0.9361
4	0.9593
5	0.9587

表 7: 各繰り返し回数での平均適合率 7

	手法 6($\alpha = 1.0, \beta = 0.5$)
繰り返し回数	平均適合率
1	0.5682
2	0.5687
3	0.6178
4	0.6411
5	0.6451

表 8: 各繰り返し回数での平均適合率 8

	手法 6($\alpha = 1.0, \beta = 0.0$)
繰り返し回数	平均適合率
1	0.5682
2	0.6599
3	0.6613
4	0.6623
5	0.6629

4.2 考察

これらの表からも分かる通り，コンセプト・プロジェクションを用いて 5 回のフィードバックを行った結果，最小で約 0.08，最大では約 0.21 の平均適合率の上昇が見られた．これにより，コンセプト・プロジェクションによるフィードバック手法の有効性を示すことができた．また，図 1 では，フィードバックによる繰り返し回数に従って再現率-適合率曲線の適合率の減少率が少なくっていることも分かる．これは，フィードバックの結果，関連のある文書がより上位に検索されるためで，このグラフからもこの手法の有効性が理解できる．これらのことは，フィードバックが行われることにより，これまで文書集合をクラスタリングして得られた概念ベクトルが，より検索質問が必要としている概念ベクトルに更新されていると考えられる．

より細かく各フィードバック手法を比較すると，関連のある文書を概念ベクトルに最も有効に更新したのは手法 4 で，個々の文書ベクトルを概念ベクトルに反映させるのではなく，検索された関連文書に対する概念ベクトルをフィードバックした方がより効果的であった．また，この手法はフィードバックの効果が早く，3 回程度の学習で平均適合率が最も大きくなっている．このことは，1 回のフィードバックの影響が大きく，さらに，概念ベクトルが検索質問に含まれる概念的な内容に大きく近づいているためであると考えることができる．

しかし，関連のある文書を手法 4 で更新し，さらに，上位に検索されたが検索質問と関連のない文書を概念ベクトルにフィードバックした場合，表 4 から表 6 に示したように，何もフィードバックしなかった場合とほとんど同じ結果となった．関連のない文書集合には，それぞれ内容的に関連性のない文書が存在する可能性があるため，手法 1 や手法 2 のようにひとつの文書を概念ベクトルに反映させた．しかし，ひとつの文書を概念ベクトルに反映させた場合，ひとつの文書ベクトルがスパースなベクトルであるためか，フィードバックの効果が少ない結果となった．

これらの手法に対する有効性を比較するために，最も一般的なフィードバック手法である手法 5 を用いて実験した結果，表 6 に示すように適合率は約 0.96 となった．この手法と比較した場合，我々の提案した手法は良い検索結果を得ることができなかった．しかし，本手法と同様にシステムに対してフィードバックを加える文献 [9] で提案された手法と比較すると，同じ条件の下において 0.7019 を上回る検索結果を得ることができた．システムに対してフィードバックを行う場合，関連のない文書が前回同様に上位に検索されないように，どのようにシステムのパラメータを更新するかが問題となる．本実験では，関連のない文書を概念ベクトルにフィードバックしたときの影響が少なかったため，これをより有効にフィードバックする手法を考慮する必要であると考えられる．これに実現することで，手法 5 のような検索質問拡張の効果に匹敵する検索性能が得られるのではないかと予想される．

また，手法 6 に示すように，手法 5 で検索質問に直接フィードバックを行った後，コンセプト・プロジェクションにより次元圧縮をした結果，表 7 や 8 に示すように，検索性能は概念ベクトルを更新する手法と比較して，あまり良い結果とはならなかった．このことは，検索質問が更新されたとしても，概念ベクトルはそのまま同じであるために，次元圧縮を行っても，ひとつの特徴軸に対してあまり大きな変化が認められなかった．すなわち，フィードバックによりある単語が拡張されたとしても，次元圧縮時

においてその単語の影響が少なくなってしまった結果であると考えられる．

5 おわりに

本稿では，次元圧縮手法であるコンセプト・プロジェクションに必要な概念ベクトルのパラメータをフィードバックさせる手法を提案し，検索精度の改善を試みた．その結果，最も一般的なフィードバック手法を用いて実験し，この手法と比較した結果，我々の提案した手法はそれより良い検索結果を得ることができなかった．しかし，検索質問拡張のような検索時のみの短期的な効果ではなく，システムに対してフィードバックを加え，システムの検索性能を長期的に高める手法としては，本手法の有効性を示すことができた．

今後の課題としては，本実験で用いたテスト・コレクションである MEDLINE には収録されているデータの少なく，さらに医学関連という特定の分野のデータであるため専門用語などの単語が強く影響を与える可能性が高いことがが問題となる．このため，日本語情報検索の評価用テスト・コレクションである BMIR-J2[10] などのような大規模で，様々な分野の内容を持ったテスト・コレクションを用いた検索実験，および評価をすることが必要であると考えている．また，検索質問と関連のある，および関連のないそれぞれの文書集合に対して，これまでに示したフィードバック手法を改良し，より概念ベクトルの性能，および検索性能の改善を行いたいと考えている．

参考文献

[1] R. I. Arriaga and S. Vempala. An algorithmic theory of learning: Robust concepts and random projection. In *Proceedings of the 40th Foundations of Computer Science*, pages 616–623, 1999.

[2] E. Chicholm and T. G. Kolda. New term weighting formulas for the vector space method in information retrieval. Technical report, Oak Ridge National Laboratory, Oak Ridge, Tennessee, 1998.

[3] I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. Technical report, IBM Almaden Research Center, 1999.

[4] D. D. Lewis. Evaluating text categorization. In *Proceedings of Speech and Natural Language Workshop*, pages 312–318, 1991.

[5] J. J. Rocchio. Relevance feedback in information retrieval. In *Salton G. (Ed.), The SMART Retrieval System. Englewood Cliffs, N.J.: Prentice Hall.*

[6] G. Salton and M. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, 1983.

[7] C. C. Vogt, G. W. Cottrell, R. K. BeLew, and B. T. Bartell. User lenses - achieving 100% precision on frequently asked questions. In *Proceedings of User Modeling '99, Banff*, pages 87–96, 1999.

[8] I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images.* Van Nostrand Reinhold, New York, 1994.

[9] Tai Xiao Ying, Minoru Sasaki, Kenji Kita, and Yasuhito Tanaka. Improvement of vector space information retrieval model based on supervised learning. In *Proceedings of the Fifth International Workshop on Information Retrieval with Asian Languages(IRAL2000)*, pages 69–74, 2000.

[10] 木谷 強ほか. 日本語情報検索システム評価用テストコレクション BMIR-J2. データベースシステム研究会, 情報処理学会, 1998.

[11] 佐々木 稔, 北 研二. ランダム・プロジェクションによるベクトル空間情報検索モデルの次元削減. 自然言語処理 (印刷中).