2004－NL－161　(11)
2004／5／14

トピックドリフトを支援する新聞記事ナビゲーションシステム

大熊 耕平[1]・山田 剛一[1]・増田 英孝[1]・中川 裕志[2]

[1]東京電機大学
[2]東京大学 / 社会技術研究システム 総括研究グループ

本研究では，あるトピックについての文書群から，そのトピックに関連する別のトピックの文書群へとユーザを連続的にナビゲートするシステムを開発した．本システムでは複数の新聞記事サイトの横断検索を行い，その結果得られる同一トピックの記事間の差異をユーザに提示することによりナビゲーションを行う．本システムの適用範囲は同一トピックの新聞記事群に限らず，同一トピックについて議論しているドキュメント群一般に適用することができる．例えばある問題について議論しているドキュメント群が存在したとき，本システムを用いることにより，その問題に関連のある情報や，その問題への異なる視点を見つけることができる．

CROSS-ARTICLE-SEARCH AND TOPIC DRIFTING AID SYSTEM FROM MULTIPLE NEWS SITES

Kouhei OHKUMA[1], Koichi YAMADA[1], Hidetaka MASUDA[1], Hiroshi NAKAGAWA[2]

[1] School of Engineering, Tokyo Denki University
[2] The University of Tokyo

In this research, we developed a system which navigates a user from a topic to the related other topics. This system retrieves articles of the specific topic from multiple news sites and shows differences between the articles. This system is for not only news articles but also general documents. By using this system, a user can find some related information and different points of view of the same topic.

1. はじめに

現在，インターネット上では主要な新聞社や出版社などによって記事が無料で公開されており，幅広く利用されている．これらの記事を公開しているサイト(新聞記事サイト)を横断的に検索することができれば，「複数のサイトを一度に調べたい」，「同じトピックの記事を重複して読みたくない」，「同一のトピックの記事が発信元によってどのように異なるのか知りたい」，「あるトピックの記事を読み，それに直接的，あるいは間接的に関係する記事をいもづる式に探索したい」といったユーザの要求に応えることが可能となる．本研究では，複数の新聞記事サイトを横断検索すると同時に上記の「いもづる式探索」をするシステムを作成した．

本システムで最初に各新聞社の記事の横断検索を行うと，ユーザは内容の類似する記事群を得ることになる．この類似する記事群の差異をユーザに提示することにより，ユーザは何がメイントピックで何がサブトピックなのか，あるいは情報源に固有の視点は何か，といったことを知ることができる．それらの情報をユーザが取捨選択して次回検索に反映させていくことにより，ユーザは上に述べたような「い

もづる式」に新たなトピックへとナビゲートされる．このように，本システムはトピックのナビゲータの役割を果たすよう設計されている．

本システムの適用範囲は同一トピックの新聞記事群に限らず，同一トピックについて議論しているドキュメント群一般に適用することができる．例えばある問題について議論しているドキュメント群が存在したとき，本システムを用いることにより，その問題に関連のある情報や，その問題への異なる視点を見つけることができる．コンセンサス形成過程においては，問題となっている話題だけで議論するのでは解決や妥協ができないことがあり，そのような場合に，関連する知識や意見はもちろんのこと，別の観点での意見や知見がコンセンサスを形成するために役立つものと期待される．

2. 複数新聞記事サイトのいもづる式ナビゲーションシステム

2.1. システムの概要

検索要求にマッチする記事を複数の新聞記事サイトから収集，検索を行うことで，類似記事を収集することができる．類似する多数の記事には，単独の記事に比べて，いろいろな観点からの情報，見方などが多数示されていると予想される．この類似記事を用いることで，ある話題について，単独の記事ではカバーしていなかったテーマへの糸口が見つかりやすくなると期待できる．

複数の類似記事はおよそ Fig. 1 のように内容がオーバーラップしている．質問と一番類似度が高い記事はユーザが全文を読むことを想定する．ここで，一番目の記事で扱っていない内容をいもづる式に手繰るという局面を想定してみよう．同じような問題設定は多文書自動要約に見られる [1]．多文書自動要約では，MMR(Maximal Marginal Relevancy)という考え方が使われる．すなわち，元の質問に類似していることと，既に選択した記事に類似していないことの両者を加味した基準によって記事を選択する．我々の目的では，いもづる式ナビゲーションの性質上，既に選択した記事に類似していないことにのみ重きをおくことになる．したがって，問題は類似度が 2 番目以降の記事に出ている内容のうち一番目の記事と重複しないものをどのようにブラウザ上に提示するかである．要約における MMR の場合，表示は記事単位であった．しかし，ここでは記事全部を提示してしまうと，いたずらに利用者に負担を強いる．そこで，2 番目以降，例えば n 番目に類似した

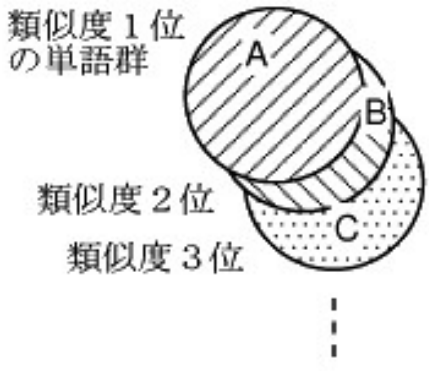

Fig. 1　類似記事群における単語の重なり

記事内容を提示する方法は，

- n 番目の記事の内容のうち，既に選択した n-1 番目までの記事に出現していない内容を表す部分を表示する．

しかし，このような部分に対応する文の集合を探し出すことは，意味理解に近いことが必要であり，現実的ではない．より軽い処理で実現でき，かつ利用者にも indicative な情報を提示できるという観点からは単語を単位とする表示が現実的である．そこで，このシステムでは，Fig. 1 の重ならない部分を単語の集合とみなすことにした．単語を提示するもうひとつの利点は，単語には TF×IDF などの方法で重要度がつけられ，その重要度の順に表示するというコンパクトな表示ができる点である．

よって，提案するシステムでは，収集した記事とその記事の固有の単語を提示する．この単語を選択し再び検索を行うことでユーザをナビゲートする．本システムはサーブレットを用いている．

このシステムの流れを以下に示す．

1. 複数新聞記事サイトの記事を収集しインデックスを作成する．
2. ユーザが検索要求を入力する．
3. 検索要求の単語を含む記事群とその記事に含まれる単語群を取得する．
4. 検索要求と各記事との類似度を求める．
5. 検索要求に最も類似した記事と，他の記事との類似度を求め，この類似度順に記事を並び替える．
6. ユーザに記事群とその各記事に固有の単語を提示する．

複数の新聞社のサイトから，検索対象となる大量

の記事をネットワーク経由で取得するのには時間がかかる．そのため，1 の記事の収集はユーザが検索を行う前にあらかじめ行っておく．この記事を収集する段階で，記事検索やナビゲーションを行う際に必要となるインデックスを作成する．なお，この複数新聞記事サイトの記事収集はサーバで自動的に行う．

ユーザがシステムに検索要求を与えたとき，その検索の対象は，その時点までにシステムが収集したすべての新聞記事となる．検索要求は，単語一つでも，複数の単語を含んでいてもよい．また，複合語を入力することも可能である．システムは 1 で構築したインデックスから，検索要求に含まれる単語群を含む記事(の ID)と，その記事に含まれる単語群を取得する．

検索要求に含まれる単語群と，各記事に含まれる単語群から，検索要求と各記事との類似度を求める．さらに，検索要求との類似度が最も高い 1 記事を新たな検索要求に見立て，この記事と他の記事との類似度を求める．この新たな類似度を使って記事を並び替えることにより，記事間の類似度の高い記事が上位に集まることになる．

ここで上位となった記事間で共通に含まれる単語をユーザに提示する．また，類似度上位となった記事は，その記事の見出し，URL，その類似度順により類似度の高い記事に含まれず，また，記事間共通の単語でもない単語群を提示する．

ユーザは提示された単語群を取捨選択し，次の検索の方向を定める．ここで 2 に戻る．このように，検索の方向をインタラクティブに変化させながら検索を進めていくのが，本システムのナビゲーション機能である．

2.2. 複数新聞社サイトからの記事収集

記事の収集は，新聞記事サイトのトップページを起点として行う．新聞記事サイトのページでは，記事本文のあるページへのリンクのほか，記事以外の様々なページへのリンクも同時に張られている．このような状況の下，サイトの中から記事本文のあるページを探し出し，そのページだけを取得する必要がある．

また，記事収集の対象とすべきサイトはユーザの好みによって異なり，さらに，記事を収集するサイトはいつでも新規に追加できることが望ましい．よって，サイト内から記事であるページを探し出す際に，特定のサイトに依存するような情報を用いて記事を収集する方法は使えない．そこで，新聞記事サイトから記事のページのみを探し出す方法として，

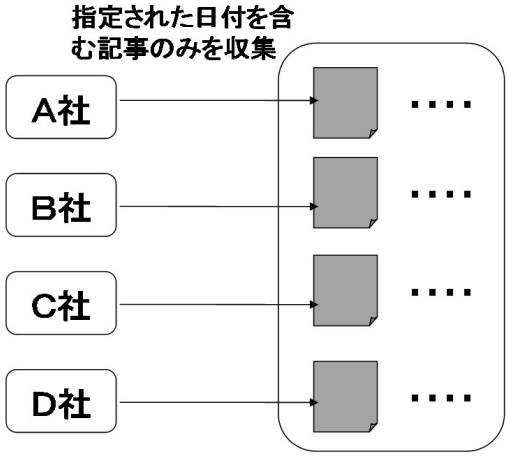

Fig. 2　複数サイトからの記事収集

新聞記事サイトに普遍的な特徴を用いて記事の収集を行う．

ここでは各ページの URL に注目した．記事本文を含むページの URL には，その記事が載った日付が含まれている．これを用いて記事ページに関する判断を行った．ほとんどの新聞記事サイトでは，記事収集を行った時点での，最新の記事へのリンク（URL に日付を含んでいる）と，各ジャンル（社会，スポーツ，政治など）の記事一覧ページへのリンクが同一ページに存在するため，これを記事ページ探索の手がかりとして利用する．

まず，トップページに含まれるリンクの URL を全て取得し，その URL が日付を含むかを調べる．この時点で日付を持つ URL があれば，そのリンクが最新記事へのリンクであると判断することができる．トップページに日付を持つ URL のページへのリンクがなければ，見つかるまでリンクをたどっていく．

前述の性質から，最新の記事へのリンクのあるページ には各ジャンルの記事一覧ページへのリンクがあるため，そこからさらに一階層リンクをたどれば各ジャンルの記事一覧ページを取得することができる．この記事一覧ページの中のリンクを調べ，特定の日付を持つ URL のページのみを取得することで，記事であろうページだけを収集することができる．記事一覧のページのリンクがわかり記事であろうページを収集した時点で，記事収集を終了する．

しかし，この中には URL に日付を持っていたとしても記事でないページが含まれている．このページを除くために，その記事へのリンクがはってある，リンク元のページの文字列を手がかりとして使用する．記事へのリンクがはってあるリンク元の文字列は，リンク先のページでは記事の見出しとして含ま

れているので，記事へのリンクがはってある文字列がリンク先のページに含まれているか否かを調べる．これによって記事であろうページ群から記事のみを収集することができるようになる．

このような記事収集方法を行うことで，与えられたサイトのページ全てを調べることなしに，そのサイトの特定の日付の記事のみを収集することが可能となる．

以上の方法を，収集対象である複数サイトに対して行う．Fig. 2 の例では，A 社～D 社のサイトから特定の日付の記事だけを収集している．その収集したすべての記事が検索の対象となる．前述の方法を使えば，A 社～D 社だけでなく未知のサイトにも対応することができる．

2.3. 収集記事からの必要な部分の抽出

記事の検索のために，収集した記事から単語とその単語の TF 値(文書内出現頻度)と URL のデータベースを作成する．データベースを作成することで，後の記事の検索を行うことができる．

まず，収集した記事からその記事の見出しと本文を抽出する．ここでもサイト依存性を極力排除するため，リンク以外の HTML の要素は手がかりとして使用しない．まず見出しの抽出を行う際に，各記事のリンク元のページにある，この記事へのリンクがはってある文字列を手がかりとして使う．この文字列はその記事の見出しとほぼ同一の文字列であるので，これを用いて記事の見出しを探し出す．見出しが発見された後，見出し以降にある，句点がある文を探し出す．記事の本文には句点がついているので，句点がある文を探し出すことで本文を抽出することができる．次に，抽出した見出しと本文から，名詞を取り出す．これには，形態素解析システム茶筌 [3] を用いた．抽出した見出しと本文を茶筌にかけ形態素解析を行い，名詞である単語のみを取り出し，これをデータベース作成に用いた．この時点で取り出した単語の TF 値を求める．また，取り出した単語に対応する複合語も取り出しておく．これは，それぞれの記事を特徴付ける単語を，複合語を用いて表示するために用いられる．複合語で表示することで，単語のみで表示されるよりも，意味が汲み取りやすくなる．

これにより，単語をキーとして URL を返すデータベースと，URL をキーとしてその記事の単語と TF 値を返すデータベース，単語をキーとして複合語を返すデータベースを作ることができる．これを記事の検索に用いる．

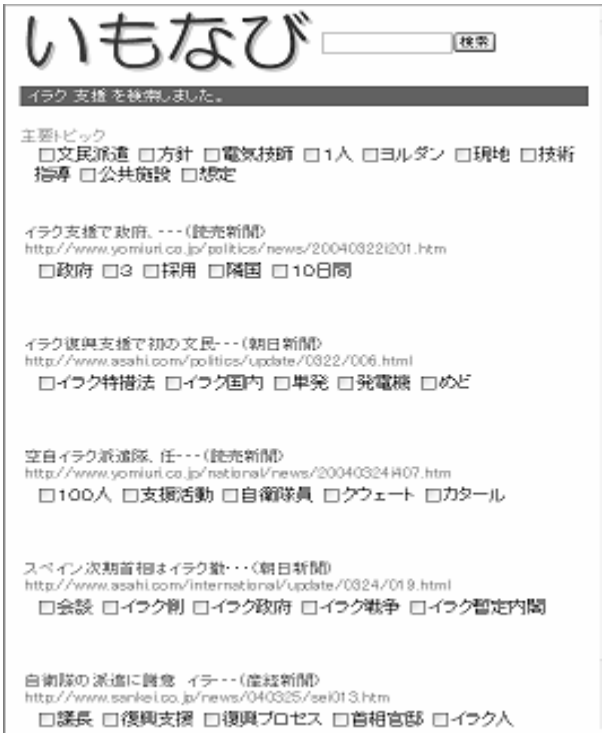

Fig. 3 「イラク支援」で検索を行った結果

2.4. 記事の検索

あらかじめ収集してある記事の中から検索要求に合う記事を検索する．検索要求の単語を見出しまたは本文中に含んでいる記事だけをユーザに提示する対象の記事とし，収集してある記事の中から検索する．単語を含んだ記事かどうかを調べるときに，2.3 節で作成したデータベースを用いる．このデータベースを用いて，検索要求の単語を見出しまたは本文中に含んでいる記事を簡単に探し出すことができる．この探し出した記事を用いて類似度の算出をし，ユーザに記事を提示する．

2.5. 類似度の算出

検索要求に最もマッチする記事とそれに関連する記事を探し出すために 2 段階の類似度計算を行う．はじめに検索要求の単語群と，検索にマッチした各記事群の間で類似度の算出を行い，検索要求に最もマッチする記事を選び出す．次に，この最もマッチした記事と，残りの記事群の間で再び類似度の計算を行う．このようにすることで，検索要求にマッチする記事と，それに関連する記事群を選び出すことができる．

Fig. 4 「政府」「首相官邸」で検索を行った結果

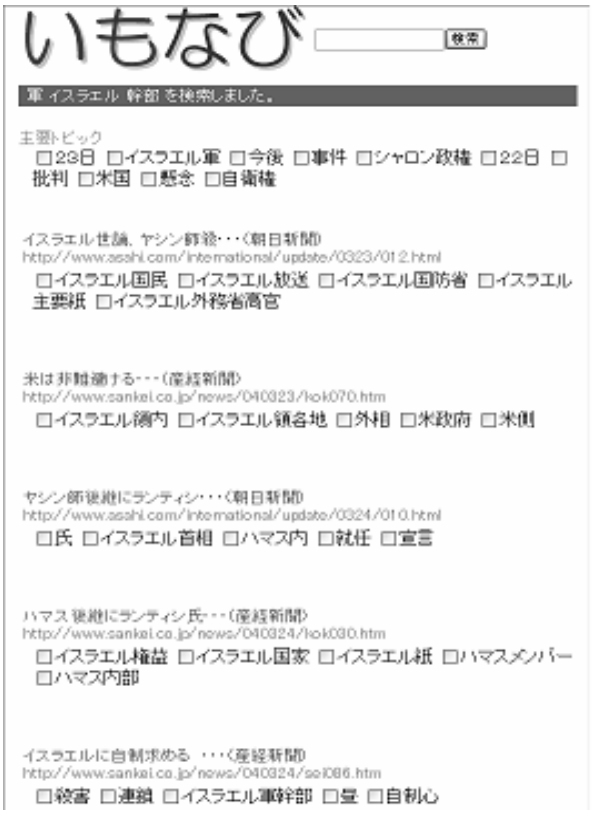

Fig. 5 「イスラエル軍幹部」で検索を行った結果

検索要求の単語群と各記事間の類似度および，最もマッチした記事とその他の各記事間の類似度には cosine 類似度を用いた．このときベクトルの次元は検索要求の単語および記事に含まれる単語の種類数となる．ここで，検索要求の単語群のベクトルの要素は，検索に用いた全記事に対する，その単語の IDF を用いている．また，記事のベクトルの要素は TF の値を用いている．

2.6. 類似度による記事の提示とナビゲーション

検索結果の例を Fig. 3 に示す．まず，それぞれの記事に共通して現れる単語を主要トピックとして提示する．次に検索要求との類似度順に，記事の見出し，URL，それぞれの記事固有の単語を提示する．ここで提示した記事固有の単語を次のナビゲーションのための検索語として用いる．この「記事固有の単語」とは，2.1 節でも示したようにその記事よりも類似度上位の記事に含まれておらず，また，各記事に共通して現れる単語も含まない，その記事の単語のことである．この単語は TF×IDF を計算し，その値の大きい順に提示している．ユーザはこの「記事固有の単語」を選択していくことでナビゲートされていく．

実際にはユーザに単語をそのまま提示するのではなく，単語が記事中で複合語として現れている場合にはその複合語を提示している．これにより，ユーザは各記事に現れる固有の概念が把握しやすくなっている．なおこの機能を実現するため，2.3 節で作成した，単語を元に複合語を得るデータベースを用いる． Fig. 3〜Fig. 5 でナビゲーションの流れの例を示す．Fig.3 では，最初に「イラク支援」という単語で検索を行った場合の例である．検索結果にはイラク支援関連の記事が得られているのがわかるが，それぞれ内容に異なりがあるため，それぞれの記事に，その差異である単語群が提示されている．ここで次の検索を行うために「政府」 「首相官邸」という単語を選択して検索を行なってみる．この検索結果を Fig.4 に示す．検索の結果としては尖閣諸島の問題関連の記事が得られていることが見出しからわかる．ここではイラク支援の話題から日本と中国の尖閣諸島をめぐる話題に変わっていることがわかる．次に「イスラエル軍幹部」という単語を選択して検索を行なう．この検索結果を Fig.5 に示す．検索結果としてはイスラエルのヤシン師殺害関連の記事が得られている．

この例では「イラク支援」→「尖閣諸島問題」→「イスラエルヤシン師殺害」とトピックが移り変わっていった．このように，検索した結果の記事の差分を選択していくことで，ユーザは新たなトピックへとナビゲートされていく．

3. 実験

本システムの最大の特徴は，類似記事間の差異の提示によるトピックドリフトの実現にある．そこで，差異を提示することによるトピックドリフトの効果を確認するため，いくつかの実験を行った．

3.1. 実験の方針

トピックドリフトの様子を定量的に扱おうとする場合，いくつかの着目点が考えられる．

(a) システムが表示する記事の変遷
(b) システムが表示する単語群の変遷
(c) ユーザが選択する単語群の変遷

(a), (b) はシステム側，(c) はユーザ側に着目しているが，システムの表示はユーザの単語選択によって決まるため表裏一体である．ここでは (a) に着目し，大きなトピックとも考えることのできる記事という単位で，ドリフトの様子を見ていくことにした．記事をベクトル空間モデルにより単語ベクトルとして表し，検索が進むにつれ，1 位の記事の単語ベクトルがどのように変化していくのかを類似度計算により求める．ドリフト回数(すなわち検索回数)をパラメータとし，検索を重ねるごとに，当初の記事から内容が離れていくかどうかを調査した．

類似記事間の差分を見せるか否かによってドリフトの様子がどう異なるかを調査するのが基本であるが，類似記事の集まり方は 2.5 節で述べた 2 段階の記事の並べ替えを行うか否かに依存するため，この 2 段階の並び替えを行う場合と行わない場合の両方について実験を行った．

3.2. 実験方法

類似記事間の差分を取る場合と取らない場合の二つの方法で検索を行い，どの程度ドリフトするかを調査した．2.5 節で述べた 2 段階の類似度計算による記事の並べ替えを行なって記事を表示する場合と，2 段階の類似度計算を行なわず，検索要求にマッチした順位で記事を表示する場合の 2 通りで実験を行なった．検索結果の内容がどの程度ドリフトしているかを見るために，最初に検索した結果 1 位となった記事と，それ以降の検索で 1 位となった記事との間で類似度計算を行い，その値の大きさをどの程度ドリフトしているかの判断に用いた．検索は最初に決められた検索要求を入力し，その後，チェックボックスにランダムにチェックを入れて（チェック率は 8%），7 回連続で検索を行った．これを 1 セットとし，5 セット行い，平均を求めた．検索要求には「イラク派遣」，「北朝鮮拉致問題」，「大リーグ」の 3 つの単語を用い，それぞれ 5 セットずつ行なった．合計として，15 セット分のデータを平均して結果を求めた．ここで，ランダムにチェックを入れるのはそれぞれの記事の特徴語のみとし，差分を取る方法で表示を行なう共通トピックの語は選択しないものとした．

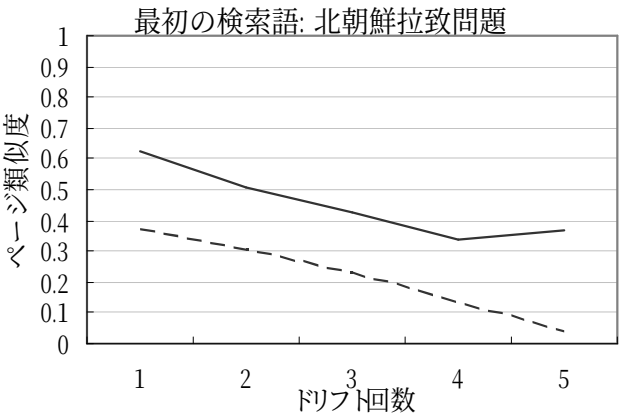

Fig. 6 ドリフトによるトピックの拡散度
（並べ替えなし，「北朝鮮拉致問題」）

3.3. 実験結果

実験の結果を Fig. 6 から Fig. 9 に示す．実線が差分を取らなかった場合，破線が差分をとった場合である．グラフの縦軸の値は記事間の cosine 類似度の値である．また，グラフの横軸は，類似度を求めた記事と記事との距離である．たとえば，1 回目の検索結果の記事と，3 回目の検索結果の記事で類似度を求めた場合，検索回に 2 回分の差があるので，記事の距離 2 となる．同様に 2 回目の検索と，4 回目の検索も距離 2 となり，以下，3 回目と 5 回目，4 回目と 6 回目，5 回目と 7 回目の場合も距離 2 となる．これらの類似度を平均したものが，グラフの数値となる．

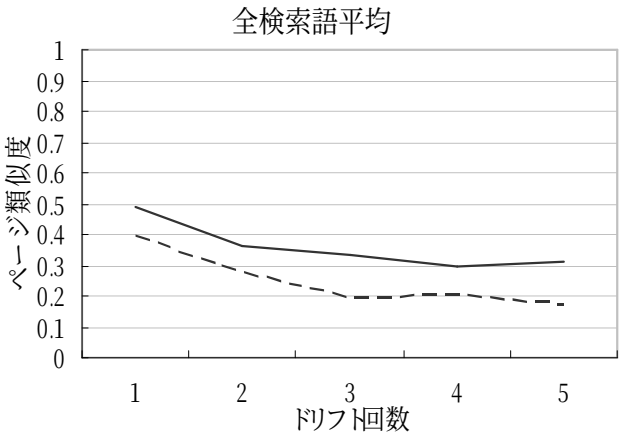

Fig. 7　ドリフトによるトピックの拡散度
（並べ替えなし，平均）

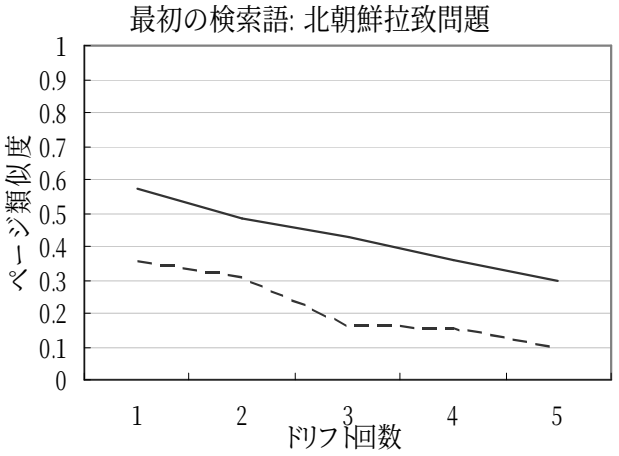

Fig. 8　ドリフトによるトピックの拡散度
（並べ替えあり，「北朝鮮拉致問題」）

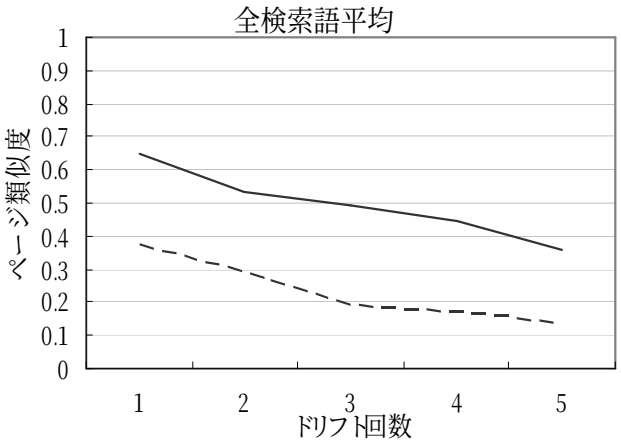

Fig. 9 ドリフトによるトピックの拡散度
（並べ替えあり，平均）

2 段階の類似度計算による並べ替えを行なわない場合の評価

はじめに 2 段階の類似度計算を行なわない方法で，差分を取る，取らないによって，どの程度ドリフトに影響があるのかを調査した．Fig. 6 に「北朝鮮拉致問題」を検索要求として検索を行った場合の結果を，Fig. 7 に 3 つの語での検索結果を平均した結果を示す．

結果として，検索をした記事間の距離が離れるほど類似度が低くなっている．これは，検索を重ねることにより，記事の内容の関連が薄くなっていることを示している．また，差分を取った場合の類似度は，差分を取らなかった場合の類似度が低い．このことから，差分を取らなかった場合よりも，差分を取った場合のほうが，より発散していることがわかる．

2 段階の類似度計算により並べ替えを行なう場合の評価

次に，2 段階の類似度計算を用いて記事を並べ替えた後に，差分を取る，取らないによって，どの程度ドリフトに影響があるのかを調査した．Fig. 8 に「北朝鮮拉致問題」を検索要求として検索を行った場合の結果を，Fig. 9 に 3 つの語での検索結果を平均した結果を示す．

結果として，並べ替えを行なわない場合と同様に，検索をした記事間の距離が離れるほど類似度が低くなった．また，差分を取った場合の類似度は取らなかった場合の類似度よりも低く，この場合にも差分をとった場合のほうが，より発散することがわかる．

3.4.　まとめ

2 つの実験結果より，差分を取らない場合よりも，差分を取って検索を行うことで，類似度が低くなることがわかった．類似度が低くなるということは，元の記事との関連が薄くなったということであり，トピックのドリフトが起きやすいことを示している．ここで，Fig. 7 と Fig. 9 の差分を取らない場合を比較して見ると，Fig. 9 の結果のほうが類似度が大きいことがわかる．これは，並べ替えを行なうと，共通のトピックの記事が集まることとなり，差分を取らないと同じような単語が特徴語として提示されてしまうためである．ここでも差分を取ることの有効性が示されていると言える．

4. 関連研究

本研究のシステムはナビゲーションを主眼としているが，そのナビゲーションの方向がユーザの選択する単語群により決定づけられることから，関連性フィードバック(relevance feedback)[2]に似ている面がある．ただし，関連性フィードバックによるクエリー拡張は，検索開始時のユーザの意図に近づくようクエリーを修正していく仕組みであるのに対して，本研究のシステムでは，初回の検索結果と(関連はしているが)異なる方向へとナビゲートしていく．よって，ユーザからのフィードバックは，その内容も目的も異なっている．

本研究のシステムは類似記事群とその差異をユーザに提示するため，一般的なドキュメント空間の可視化システム[4]，あるいはWebの可視化システムとの類似性がある．それらに比べると本研究のシステムは単純なインタフェースであるが，ほぼ同一内容のドキュメントが多数存在するという状況下において，その相違点に着目しユーザに提示する場合には，単語そのものによって差異を提示するのが明解でわかりやすい．ただし，目的がナビゲーションではなく差異そのものの閲覧である場合には，異なる単語群を含む段落といった単位で抽出することが望ましいといえる．なお，Web全体の拡大に伴い，「ほぼ同一内容のドキュメントが多数存在する」という状況も拡大しており，本研究のシステムの適用範囲も拡大の方向にあると考えている．

本研究のシステムはナビゲーションシステムであるため段階的にトピックがドリフトしていくが，その一回一回におけるドリフトの度合いは，すでに2.1節で指摘したように，多文書自動要約においてMMRにより定量化されている．本システムの目的は要約ではないが，トピックのドリフトを評価する際の参考としたい．

5. おわりに

本研究のシステムの大きな特徴はトピックのドリフト支援であるが，このドリフトの履歴を可視化することにより，個々のユーザの視点に立ったトピック・サブトピックのマップが得られる．これはトピック・サブトピックの関連性の分析に役立つほか，ユーザが自分の履歴を鳥瞰することによって，ユーザの頭の中を整理することも可能となる．

本研究のシステムはまだ試作段階である．これから運用実験を行うことにより，トピックのドリフトがどのような条件下で効果を発揮するのか確認していきたい．また，このシステムがどのような目的・場面で役立つのか未知数な部分もあるため，トピックのドリフト履歴を可視化する機能を用いながら，このシステムの新たな可能性を明らかにしていきたい．

参考文献

1) 奥村学，難波英嗣(2002)「テキスト自動要約に関する最近の話題」『自然言語処理』9(4)，97-116.

2) William, B, F., and Ricardo, B., (1992). *Information Retrieval --- Data Structures & Algorithms*.

3) 松本裕治，北内啓，山下善隆，松田寛浅，原正幸(1999)「日本語形態素解析システム 茶筌 version 2.0 使用説明書　第二版」『NAIST Technical Report』NAIST-IS-TR99012.

4) Earl, R. (1994). Galaxy of news: an approach to visualizing and understanding expansive news landscapes. *Proceedings of the 7th annual ACM symposium on User interface software and technology*，3.-12.

本研究は，社会技術研究システム ミッション・プログラム「安全性に係わる社会問題解決のための知識体系の構築」（2001～2002 年度は日本原子力研究所の事業，2003 年度からは科学技術振興事業団の事業）の研究として行われた．