社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2006－IS－95（4）
2006／3／16

確率的手法を用いたWebページ推薦システム

臼井 大介†　　塚本 享治‡

東京工科大学大学院 バイオ・情報メディア研究科 メディアサイエンス専攻†

東京工科大学 バイオ・情報メディア研究科‡

概要 本研究ではユーザの Web ページ利用履歴をもとにサーチエンジンから情報を収集して、ユーザの嗜好を反映したWebページの推薦を行うWebページ推薦ブラウザを構築した．情報推薦にサーチエンジンを利用するため、情報推薦時に利用する検索用クエリをシステムがいかに生成するかが大きな課題となる．本研究ではこの検索用クエリの生成において、ユーザのWebページ閲覧履歴から収集したキーワードを組みあせることでユーザの興味を近似し、これによってサーチエンジンから情報を収集した．情報推薦の過程で、どのような検索用クエリから得られる情報がユーザにとって有益となるかを事前に予測することは困難であり、不確実性の高い問題である．そこで本研究では、この不確実性に対応するために確率的手法を用いてキーワードを構築し、システムの評価を行った．

WebPage Recommendation System Using Probabilistic Method

Daisuke Usui†　and　Mithiharu Tsukamoto‡

Graduate School of Bionics, Computer and Media Science ,School of Media Scienece, Tokyo Univercity of Technology†

Graduate School of Bionics, Computer and Media Science, Tokyo Univercity of Techonology‡

ABSTRACT In this project, a web page recommendation browser which recommend web pages for a user was developed. The system exploits user's web browsing history as user preferences and gather information from existing search engine using probabilistic method. Most difficult problem of information recommendation is to predict what type of information are useful for user. To handle this problem, this project used probability method.

１．はじめに

近年、World Wide Webの急速な普及によりWeb上には膨大な量の情報が存在するようになった．この膨大な情報空間の中からユーザが必要な情報を取得する際にはサーチエンジンを用いるのが一般的である．サーチエンジンとは、事前にＷｅｂ上から大量のＷｅｂページを収集して分類やインデキシング処理を行い、ユーザが入力したクエリに応じて、関連性の高いＷｅｂページの情報をユーザに提示するシステムである．

サーチエンジンの登場によって、インターネット上に存在する膨大な情報空間の中からユーザが必要とする情報を発見することが可能となった．現在、多くのインターネットユーザがサーチエンジンを利用して情報を取得している．現在インターネット上では、日々何百ページもの Web ページが新たに出現しており、また大量の Web ページの情報が新たに更新されている．このような状況において、自分にとって有益な情報を常に取得し続けるには、毎日何百回もサーチエンジンで検索を行えば可能かもしれないが、それは現実的には困難である．インターネットの利用に熟練したユーザであっても、まだ発見していない Web ページの中に、ユーザにとって有用なものが含まれている可能性が極めて高い．

日常生活において、新聞の書評や旅行ガイド等、膨大な選択肢の中から何かを選ぶ際に、推薦情報を利用する機会は多い．このような情報の推薦はWeb上でもAmazon.com[1]のような電子商取引サイトにおいて特定のユーザへのお勧め商品の提示といった形で実用化されている．

以上のような背景から、本研究ではユーザにとって未発見でありながら、ユーザにとって有用な Web ページを発見することを支援することを目的とした Web ページ推薦ブラウザを構築した．

Web ページの推薦の過程では、ユーザの Web ページ閲覧履歴からユーザの興味を抽出した．この興味情報をもとに、確率的手法を用いてクエリを自動的に構築し、このクエリを用いて主要なサーチエンジンの一つであるGoogle[2]から情報を収集した．推薦情報の収集にサーチエンジンを用いることで、Web 全域をカバーした情報推薦を実現した．

２．関連研究

情報推薦手法の関連研究には、ユーザと似たような行動を取る別のユーザの嗜好情報をもとに情報を推薦する協調フィルタリング(Collaborative-Filtering)手法[1,2]と Web ページの内容などのコンテンツをベースに推薦を行うコンテンツベース(Contents-Based)手法の２つに大きく分かれる．ここでは本研究と関係があるコンテンツベースの関連研究と本研究との関わりについて述べる

2.1 コンテンツベースの情報推薦手法

コンテンツベースの情報推薦手法は、情報検索手法を基盤として、ユーザの閲覧履歴等のプロファイル情報と推薦の対象となる文書(Web サイト等)の内容(コンテンツ)との比較から情報を推薦する手法であり、サーバ上で動作するシステムとユーザのローカルマシン上で動作するシステムがある．

サーバ上で動作する情報推薦システムとして、WebPersonaizer[3],WebMate[4],Web ページ推奨エンジン[5]等がある．これらは主にプロキシサーバ等を用いて、ユーザの Web ページ閲覧履歴等の情報をサーバにログと

[1] Amazon.com: http://www.amazon.com/

[2] Google: http://www.google.co.jp/

して保存し、この情報をユーザの嗜好情報として利用して情報の推薦を行う手法である．推薦情報の収集や生成等の処理はサーバ側で行うことができるため、ある程度負荷の大きな処理が可能であり、さらにユーザのブラウザに依存しないシステムの構築が可能である．一方でサーバ側にユーザの閲覧履歴等の情報を提供するシステム構成となっているため、ユーザのプライベートな情報が全てサーバ管理者に把握されてしまうといったプライバシーの問題が存在する．

ユーザ側のローカルマシン上で動作する情報推薦システムとしては、Latizia[6]、WebMontage[7]、PowerScout[8]等がある．本研究で提案するシステムもこのローカルマシン上で動作する情報推薦システムに当てはまる．Latizia はユーザのブックマークや Web ページ閲覧履歴情報をもとに、ユーザが閲覧している Web ページのリンク先のページを推薦するシステムである．WebMontage は利用者の時間的な Web ページの閲覧パターンをベースに情報推薦を行うシステムである．PowerScout はユーザの閲覧履歴等の情報をもとに、サーチエンジン用のクエリを自動的に生成してサーチエンジンに問い合わせを行い、その検索結果を用いてユーザに情報を推薦するシステムである．

ローカルマシン上で動作するシステムのメリットとして、ユーザの Web ページ閲覧履歴等のプライベートな情報がユーザの利用しているマシンの外に出ないため、プライバシーを保つことができるという点がある．一方で、ユーザによって利用するマシンの性能が異なるため、負荷の多い処理や手法を利用できないといったデメリットが存在する．

2.2 関連研究と本研究との関わり

本研究では、Web ページの閲覧履歴やサーチエンジンの検索履歴等の情報はユーザの興味・関心や業務に直結したプライバシー性の高い情報であり、共有したり特定サーバに提供するべき性質のものではないと考え、情報推薦システムの構築においてローカルマシン上で動作するシステム構成にした．このさい、情報推薦にサーチエンジンを利用する PowerScout[8]のアプローチを採用した．サーチエンジンを利用することで、日々Web 上から膨大な量の情報を収集、保存し、キーワードに関連付けて提供し続けるサーチエンジンという既存のメカニズムを利用することで、広大な Web 空間全域を対象として情報を提供できるというメリットがある．PowerScout は[8]で概要が紹介されているが、情報推薦にサーチエンジンを用いる手法はまだ十分には研究されていない．これが本研究でシステム構成としてクライアントアプリケーションを採用し、情報収集にサーチエンジンを採用した根拠である．

3. 情報推薦システム

3.1 システムの概要

本研究で構築した情報推薦ブラウザ(ReXBrowser)は、次の３つのモジュールからなる．これらのモジュールの関係を図 1に示す．

(1) Web Browsing Module
(2) User Monitoring Module
(3) Recommendation Engine

Web Browsing Module はユーザとシステムとのインターフェースであり、Web ブラウザ部分のモジュールである．User Monitoring Module はユーザの Web ページ閲覧行動からユーザがクリックしたリンクのテキスト(以後リンクテキストと呼ぶ)の抽出を行ったり、ユーザプロファイルをデータベースに格納するモジュールである．Recommendation Engine はユーザプロファイルをもとにクエリを生成し、サーチエンジンから情報を取得しユーザに提示し、フィードバックを行う部分までを担当するモジュールである．

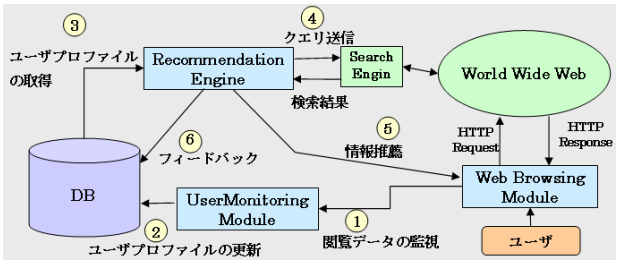

図 1　システムの概要とモジュール間の関係図

次にシステムの処理の流れを図中の番号に沿って述べる．

①閲覧データの監視

ユーザの Web ブラウジング行為によって生じる HTML データを監視する．

②ユーザプロファイルの更新

①で監視した HTML データを解析して、ユーザが閲覧した Web ページの URL，Web ページのタイトル、クリックしたリンクテキストの３つの要素によってユーザプロファイルを構成し、データベースに格納する．

③ユーザプロファイルの取得

情報推薦のためのクエリ生成に必要なユーザプロファイルをデータベースから取得する．

④クエリの送信，検索結果の取得

情報推薦に利用するクエリを生成し、サーチエンジンに送信する．検索結果として HTML 形式で取得しデータから、ユーザへの情報推薦に必要なデータのみを抽出する．

⑤ 情報推薦

ユーザプロファイルをもとに、サーチエンジンから得られた情報のランク付けを行い、ユーザに提示する．

⑥フィードバック

推薦結果をもとに、フィードバックを行う．フィードバックは推薦結果をユーザプロファイル(DB)に更新することによって行う．

3.2 システムの条件設定

本研究ではサーチエンジンから情報を収集する性質上、システムが生成する検索用クエリが情報推薦において重要な役割を果すが、この検索用クエリの生成では、次の条件設定のもと行った．

(1) ユーザの嗜好の収集を、ユーザが閲覧した Web ページのタイトルとリンクテキストに限定した．
(2) ユーザの閲覧した Web ページのタイトルとリンクテキストからキーワード候補となる単語(名詞)の切り出しを文字列処理を行うことで限定した．
(3) キーワード候補から興味単語(後述)抽出を **TF･IDF** 法を用いて行った．
(4) 興味単語２つを組み合わせることでユーザの興味

を近似した

ここで設定した条件はどれも、比較的計算コストの低い処理で実現できるものである．本研究で構築するシステムはユーザのローカルマシン上で動作するシステムであり、高い計算コストを要する手法を導入すると、ユーザのマシンによっては通常の Web ページの閲覧に支障をきたす可能性が出てくる．そこで本研究ではこのような条件を設定することで、ユーザの興味対象の収集から検索用クエリの生成までの過程における計算コストができるだけ小さくなるようにした．

3. 3 インターフェースと機能

図２は本研究で構築した情報推薦ブラウザ ReXBrowser のスクリーンショットである。

図 2　ReXBrowser のインターフェース

図２の左下の Window が情報推薦 Window であり、この Window で情報推薦を行っている．情報推薦 Window の起動は、特定のボタンをユーザが押下することにより行うようにした．情報推薦 Window には３つのクエリを用い、各クエリに対して７つの推薦情報を提示している。

システムの機能として情報推薦のほかに、推薦に利用する単語の表示、及び不必要な単語の削除機能がある。

4. 本研究の仮説

本研究では、システムが生成した検索用クエリを用いてサーチエンジンから推薦情報を収集する．サーチエンジンから得られる情報は検索用クエリによって決まるため、本研究ではユーザに対して有用な情報を提供しうる検索用クエリをシステムがいかに生成するか、が最大の課題となる．

この検索用クエリとユーザに提供する推薦情報との関係において、本研究では“ユーザが興味を持っているキーワードの組み合わせから構築した検索用クエリによりサーチエンジンから取得できる情報の中には、ユーザにとって有用な情報を含みうる”という仮説を立てる．

検索用クエリ作成の過程では、ユーザの興味のあるキーワード２つから構成される And 検索用のクエリを構築することを前提とする．

例として“ニュース”，“コンピュータ”，“料理”という３つのキーワードに興味を持つユーザを考える．上記の仮説のから、各キーワードを組み合わせた“ニュース ＆ パソコン”、“パソコン ＆ 料理”、“ニュース ＆ 料理”という３つのクエリ候補ができる．

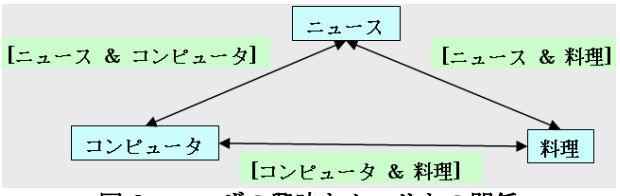

図 3　ユーザの興味とクエリとの関係

”ニュース ＆ パソコン”、”ニュース ＆ 料理”という検索用クエリは一般的に利用されうるもので、サーチエンジンからコンピュータや料理に関するニュースの情報が取得されることが考えられるため、ユーザにとって役に立つ情報を提供しうるという期待を持つことができる．しかし、この手法を用いれば“コンピュータ ＆ 料理”のように両者の間に全く関連性の無い検索用クエリをも生成しうる．一般的にユーザはコンピュータと料理の双方に興味を持っていても、両者は全く分野が異なるキーワードであるため、通常このような検索用クエリを用いてユーザが検索を行うことは極めて少ない．しかし、コンピュータと料理の両方に興味を持っているユーザであれば、一般的には利用されない検索用クエリである“コンピュータ ＆ 料理”という検索用クエリから得られる情報の中に、ユーザが興味を持つ情報を含んでいる可能性が全く無いと否定することは困難である．また場合によってはこのようなユーザによっては決して利用されないような検索用クエリから得られる情報の中に、ユーザにとって極めて役立つ情報が含まれている可能性さえあるのであるのではないか、というのが本研究の仮説である．そのような意味では、ユーザが興味を持つキーワードのどの組み合わせから、ユーザにとって有益な情報を提供しうる検索用クエリが構成されるかを事前に推測することは困難である．そこで、本研究では確率的な手法を用いてキーワードを組み合わせることで検索用クエリを構築した．

5. 情報推薦用クエリ生成理論

5. 1 クエリ生成の概要

図 4は本研究で構築したシステムの処理の流れと、用いた手法との関係を図にしたものである．

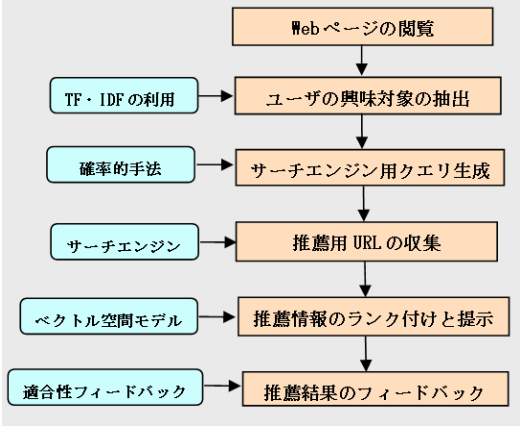

図 4　システムの処理の流れと利用する手法との関係

最初にユーザの Web ページ閲覧行動から単語を抽出し、情報検索分野において代表的な手法である **TF･IDF** 法によって各単語の重み付けを行う．それらの値をもとにユ

ーザの興味対象となる単語(以後、興味単語と呼ぶ)を選択する．次に、確率的手法を用いて、興味単語の組み合わせから検索用クエリを生成し、サーチエンジンから情報を収集する．その後、収集した情報をベクトル空間モデルを用いてユーザの興味・嗜好に合った形に再構成し、推薦情報としてユーザに提示する．最後に、提示した情報に対して、ユーザが利用した情報としなかった情報を記録しておき、それらの情報をもとに適合性フィードバック手法を用いて暗黙的フィードバックを行い、興味単語の重みを更新するというプロセスを繰り返す.

5.2 興味単語の定義

興味単語とは、ユーザの Web ページ閲覧情報から取得した単語集合のうち、単語の重みが上位 n 個の単語と定義する．単語集合の上位 n 個の単語であることから、興味単語は、ユーザが Web ページ閲覧行動において生じた情報の中でも、ユーザが特に高い興味を持っている可能性の高い単語であると言うことができる．本研究ではこの興味単語の組み合わせから検索用のクエリを構築する.

定義 1

ユーザが閲覧した全てのリンクテキスト及び Web ページのタイトルから取得した単語集合を $V=\{V_1,V_2,...,V_n\}$ とし、興味単語集合を $T=\{T_1,T_2,\cdots T_n\}$ とする．$\mathrm{top}(V,n)$ を単語集合 V のうち、単語の重みが上位 n 個の単語集合と定義する．このとき $\mathrm{top}(V,n)=T=\{T_i\in V\}$ である.

5.3 単語の重み付け

単語の重み付けは、情報検索分野における索引語の重み付けにおいてよく用いられている **TF·IDF** 重み付け(**TF·IDF** weighting)を用いた．また興味単語の重み付けには、さらにフィードバック項および時間項を追加することで、推薦結果のフィードバック及びユーザの時間的な興味の変化を反映しやすくするようにした.

定義 2

１つの Web ページの閲覧から得られる情報をここでは文書と呼ぶ．$\mathrm{frequent}(V_i)$ を文書集合全体における単語 V_i の出現頻度とし、$\mathrm{num}(V_i)$ を単語 V_i を含む文書の数とする．また N を全文書数とする．このとき、$tf(V_i)$ と $idf(V_i)$ をそれぞれ単語 V_i の出現頻度と文書頻度の逆数の関数とし、次のように定義する.

$$tf(V_i) = \log(\mathrm{frequent}(V_i)+1) \tag{5.1}$$

$$idf(V_i) = \log(\frac{N}{num(V_i)}+1) \tag{5.2}$$

単語の出現頻度が重みに与える影響を少なくするために、ここでは対数を用いて計算を行う対数化索引語頻度を用いた.

単語ベクトルを **V**、興味単語ベクトルを **T** とする．$\mathrm{time}(T_i)$ を興味単語 T_i の出現時間が後になるほど大きな値を返す関数、$\mathrm{feedback}(T_i)$ を興味単語 T_i のフィードバック関数とする．また、α,β,γ を重み付け定数とする．このとき単語ベクトル **V** と興味単語ベクトル **T** の要素を次のように定義する.

$$\mathbf{V}=\begin{matrix}V_1\\V_2\\\vdots\\V_{n-1}\\V_n\end{matrix}\begin{bmatrix}v_1\\v_2\\\vdots\\v_{n-1}\\v_n\end{bmatrix} \qquad v_i=tf(V_i)\cdot idf(V_i) \tag{5.3}$$

$$\mathbf{T}=\begin{matrix}T_1\\T_2\\\vdots\\T_{n-1}\\T_n\end{matrix}\begin{bmatrix}t_1\\t_2\\\vdots\\t_{n-1}\\t_n\end{bmatrix} \tag{5.4}$$

$$t_i=\alpha\cdot\mathrm{tf}(T_i)\cdot\mathrm{idf}(T_i)+\beta\cdot\mathrm{time}(T_i)\quad+\gamma\cdot\mathrm{feedback}(T_i) \tag{5.5}$$

ここで v_i は単語 V_i の重みを、t_i は興味単語 T_i の重みである．v_i と t_i は共に単語に対する重みだが、その意味は異なる．v_i は文書集合における単語の重要度を **TF·IDF** 手法により近似したものを意味しているのに対し、t_i はこれに時間的要素とユーザのフィードバックを追加しており、ユーザにとって有益な情報を提供しうる確信度の推定値を意味している.

興味単語ベクトルを構成する要素の重みは **TF·IDF** 項、時間項、フィードバック項の３つである．これらのうち、興味単語の重みの成分として最も重要なのは **TF·IDF** であり、次に重要なのはフィードバック項であると本研究では考えた．そこで、各定数は $\alpha>\gamma>\beta$ となる値を用いた．フィードバック関数 $\mathrm{feedback}(T_i)$ の詳細は 5.6 で述べる.

5.4 確率的クエリ構築手法

定義 3

興味単語 T_i と T_j が T_i,T_j の順で構成されるクエリを $query(T_i,T_j)$ と定義する.

興味単語集合 $T=\{T_1,T_2,\cdots T_n\}$、興味単語ベクトル **T** のとき、興味単語 T_i が最初のクエリとして選択される確率 $P(T_i)$ を以下のように定義する.

$$\mathbf{T}=\begin{matrix}T_1\\T_2\\\vdots\\T_{n-1}\\T_n\end{matrix}\begin{bmatrix}t_1\\t_2\\\vdots\\t_{n-1}\\t_n\end{bmatrix} \qquad P(T_i)=\frac{t_i}{\sum_{i=1}^{n}t_i} \tag{5.6}$$

式(5.6)から、興味単語 T_i が最初のクエリとして選択される確率 $P(T_i)$ は、興味単語ベクトル **T** の要素の重みの大きさに比例した値になる．興味単語ベクトル **T** はユーザに対して有益な情報を提供しうる確信度の推定値により構成されるベクトルであるため、要素の大きさに比例した確率によって最初のクエリを選択することで、ユ

ーザが高い興味を持つと推定される単語ほど高い確率で選択されることになる.

$n \times n$ の正方行列を Q とし、これを興味単語行列と定義する. 文書集合のうち、興味単語 T_i と T_j が共起する文書の数を frequency(T_i, T_j) とし、 feedback(T_i, T_j) を興味単語 T_i, T_j の共起頻度に対するフィードバック関数とする. このとき、興味単語行列 Q の要素 t_{ij} を次のように定義する. また、α, β, γ を重み付け定数とする.

$$Q = \begin{array}{c} \\ T_1 \\ T_2 \\ \vdots \\ T_{n-1} \\ T_n \end{array} \begin{array}{c} \begin{array}{cccccc} T_1 & T_2 & \cdots & T_{n-1} & & T_n \end{array} \\ \begin{bmatrix} t_{11} & \cdots & \cdots & \cdots & t_{1n} \\ \vdots & & & & \vdots \\ \vdots & & & & \vdots \\ \vdots & & & & \vdots \\ t_{n1} & \cdots & \cdots & \cdots & t_{nn} \end{bmatrix} \end{array} \tag{5.7}$$

$$t_{ij} = \alpha \cdot \log(\text{frequency}(T_i, T_j) + 1) + \beta \cdot \text{time}(T_i, T_j) + \gamma \cdot \text{feedback}(T_i, T_j) \tag{5.8}$$

ここで t_{ij} は興味単語 T_i が最初のクエリとして選択された場合に、興味単語 T_j を２つ目のクエリとして選択することで構成した検索用クエリ $query(T_i, T_j)$ が、ユーザに有益な情報を提供しうる確信度の推定値を意味している. 各係数は式(5.5)と同様の理由により $\alpha > \gamma > \beta$ となる値を用いた. また興味単語行列 Q の対角成分は 0 とした. フィードバック項の詳細は 5.6 で述べる. $\alpha > \gamma > \beta$ としているため、興味単語行列 Q の要素の重みで最も大きな貢献をするのは frequency(T_i, T_j) を含む項である.

frequency(T_i, T_j) は興味単語 T_i, T_j の共起頻度であり、２つの単語が Web ページのタイトルやリンクテキストに同時に出現すればするほど大きな値となる.

この興味単語行列 Q で表現したいことは、Web ページのタイトルやリンクテキストの文字列において同時に出現した興味単語ほど大きな値を持たせることで、単語間の距離を推定することである. ユーザの興味単語の意外な組み合わせによって構成される検索用クエリから得られる情報が、ユーザにとって役立つ可能性について 4.で述べたが、それが全てではない. むしろ同じ分野に属する単語や、単語間の関連性の高い単語の組み合わせによって構成された検索用クエリから得られる情報の方が、全く異なる分野の単語によって構成される検索用クエリによって得られる情報よりも、整合性のある情報を取得でき、その結果ユーザにとって有用な情報を提供する可能性が高いと考えるのが自然である. 一般的な文章においては、関連性の高い単語や同じ分野に属する単語ほど、共起頻度が高くなる性質があり、Web ページのタイトルやリンクテキストにおいても、この性質を有していると本研究では仮定する.

そこで、本研究では単語間の共起頻度が高いものほど、同時に検索用クエリとして選択される確率が大きくなるようする. 興味単語間の関連性が無かったり、分野が異なる興味単語の組み合わせが生成される確率は、システムを利用する最初の段階では小さな確率となるが、0 にはならないようにすることで、そのような組み合わせのクエリが構築される余地を残している.

定義 4

興味単語 T_i が最初のクエリとして選択された時に２つ目のクエリとして T_j が選択される条件付確率 $P(T_j \mid T_i)$ を以下のように定義する.

$$P(T_j \mid T_i) = \frac{t_{ij}}{\sum_{j=1}^{n} t_{ij}} \tag{5.9}$$

$query(T_i, T_j)$ を生成する確率を $P(T_i, T_j)$ とすると、式(5.6), (5.9)から $P(T_i, T_j)$ を次のように求めることができる.

$$P(T_i, T_j) = P(T_i) P(T_j \mid T_i) \tag{5.10}$$

本研究では、結果的にこの $P(T_i, T_j)$ から得られる確率によってクエリを選択するが、式(5.6)と式(5.9)で２つの興味単語によって構成される検索用クエリが生成できるため、クエリ生成過程で実際に式(5.10)を計算をすることはない.

5.5 推薦情報のランク付け

本研究ではユーザの興味・嗜好をもとに推薦情報のランク付けを行い、そのランクの高い順にユーザに提示する. その際、ユーザの嗜好情報として興味単語ベクトル **T** を用い、ランク付け手法として情報検索分野において用いられているベクトル空間モデルを用いた. サーチエンジンからは通常、膨大な量の検索結果が得られるが、その中にユーザにとって有益な情報がランクの上位にこなかった場合に、ユーザはそれを見つけることが困難になる. これはサーチエンジンを用いて情報推薦を行う場合にも問題となるため、ユーザにとって有益な情報が上位にくるように、サーチエンジンから取得した情報のランク付けを行った.

5.5.1 ベクトル空間モデルとは

ベクトル空間モデル(Vector Space Model)は Salton[9]らにより提案された手法であり、情報検索分野で広く利用されている手法である. ベクトル空間モデルでは索引語にもとづいて文書と検索質問とを多次元ベクトルで表現し、ベクトル間の類似度を求める. 検索対象となる文書 $D_1, D_2, ..., D_n$ とし、これらの文書集合に対して m 個の索引語 $k_1, k_2, ..., k_m$ があるとする. このとき文書 D_i を次のようなベクトルで表現し、これを文書ベクトルと呼ぶ.

$$\mathbf{d}_j = \begin{bmatrix} d_{1j} \\ d_{2j} \\ \vdots \\ d_{mj} \end{bmatrix} \quad (5.11)$$

ここで、d_{ij} は索引語 k_i の文書 D_j における重みである.

検索質問も索引語の重みを要素とするベクトルで表現することができ、検索質問文に含まれる索引語 k_i の重みを q_i とすると、検索質問ベクトル $\mathbf{q}$ は次のように表すことができる.

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_m \end{bmatrix} \quad (5.12)$$

情報検索においては、与えられた検索質問文ベクトル $\mathbf{q}$ と各文書ベクトル $\mathbf{d}_j$ の間のベクトル間の距離を求めることで類似度を計算し、検索質問文と類似した文書を見つけ出す. 類似度の計算手法としていくつかの手法が提案されているが、文書検索においてよく用いられるものはコサイン尺度や内積である. 類似度を S とした場合、コサイン尺度と内積の計算は以下の通りである.

コサイン尺度

$$S = \cos(\mathbf{d}_j, \mathbf{q}) = \frac{\mathbf{d}_j \cdot \mathbf{q}}{\|\mathbf{d}_j\| \|\mathbf{q}\|} = \frac{\sum_{i=1}^{m} d_{ij} q_i}{\sqrt{\sum_{i=1}^{m} {d_{ij}}^2} \sqrt{\sum_{i=1}^{m} {q_i}^2}} \quad (5.13)$$

内積

$$S = \mathbf{d}_j \cdot \mathbf{q} = \sum_{i=1}^{m} d_{ji} q_i \quad (5.14)$$

本研究では推薦情報のランク付けに、計算コストの低い内積を用いた.

5.5.2 ベクトル空間モデルを用いたランク付け

本研究では推薦情報のランク付けの過程でベクトル空間モデルを用いた. 情報検索分野で用いられるベクトル空間モデルを本研究における推薦情報のランク付けに置き換えると次のように考えることができる. まず、ベクトル空間モデルにおける索引語は本研究における興味単語に対応させることができる. またベクトル空間モデルの文書集合は本研究においてはサーチエンジンから取得した推薦情報(Web ページのタイトル文字列)に対応する. そのため、興味単語と推薦情報をベクトル空間で表現し、それらの距離を求めることで、類似度を計算できる.

本研究ではこのベクトル間の距離による類似度のほかに、サーチエンジンの検索結果に付随するランク情報と、推薦情報が既にユーザによって過去に閲覧されているかどうかの情報を追加した形で推薦情報のランク付けを行う.

定義 5

推薦情報を $L_1, L_2, ..., L_n$ とする. このとき L_i は一つの推薦情報に含まれる単語集合である. これらの文書集合に対して m 個の興味単語 $T_1, T_2, ..., T_m$ があるとする. このとき推薦情報(推薦文書)を次のようなベクトルで表現し、これを推薦情報ベクトルと呼ぶ.

$$\mathbf{l}_j = \begin{matrix} T_1 \\ T_2 \\ \vdots \\ T_m \end{matrix} \begin{bmatrix} l_{1j} \\ l_{2j} \\ \vdots \\ l_{mj} \end{bmatrix} \quad \mathrm{l_{ij}} = \begin{cases} 1 & (T_i \in L_j) \\ 0 & (T_i \notin L_j) \end{cases} \quad (5.15)$$

ここで、l_{ij} は興味単語 T_i の推薦情報 L_j における重みである. このとき、推薦情報ベクトル $\mathbf{l}_j$ と興味単語ベクトル $\mathbf{T}$ とのベクトル上における内積距離 S は次の式で表せる.

$$S = \mathbf{l}_j \cdot \mathbf{T} = \sum_{i=1}^{m} l_{ji} T_i \quad (5.16)$$

推薦情報(推薦文書)中の推薦情報 L_i の推薦度の度合いを $\mathrm{RecommendValue}(L_i)$ とする. サーチエンジンから取得した情報のうち上位 N 個の検索結果を情報推薦対象とし、推薦情報 L_i のサーチエンジンによるランクを $\mathrm{Rank}(L_i)$ とする. また、推薦情報 L_i がユーザによって過去に閲覧された回数を $\mathrm{BrowsedNum}(L_i)$ とする. このとき、$\mathrm{RecommendValue}(L_i)$ を次のように求める.

$$\mathrm{RecommendValue}(L_i) = \alpha \cdot \log(\mathbf{l}_j \cdot \mathbf{T}) + \beta \cdot \log(\mathrm{Rank}(L_i)) - \gamma \cdot \log(\mathrm{BrowsedNum}(L_i)) \quad (5.17)$$

これは、推薦情報 L_i と興味単語ベクトル $\mathbf{T}$ の内積距離が大きく、サーチエンジンのランクが上位のものほど大きな値となる. また、推薦情報 L_i をユーザが過去に閲覧している場合は、その閲覧回数が大きくなるにしたがって、$\mathrm{RecommendValue}(L_i)$ の値は小さくなる.

本研究ではシステムがサーチエンジンから取得した Web ページの $\mathrm{RecommendValue}(L_i)$ を全て求め、その値の大きな順にユーザに情報を提示した.

5.6 フィードバック

5.6.1 フィードバックモデル

本システムでは、提示した推薦結果に対するユーザの行為によってフィードバックを行う. 本システムにおけるフィードバックとは、提示した推薦結果のうち、どの情報がユーザに利用され、どの情報がユーザに利用されなかったのかをシステムに教えることにより推薦精度を向上させるためのものである. フィードバックの基本的な考え方は、ユーザにとって有益な情報を提供した興味単語の重みを大きくし、逆に役に立たなかった興味単語の重みを小さくするようにフォードバックをかけるというものである.

図 5はフィードバックの概要図である. 本研究では、

情報推薦結果をもとに、興味単語ベクトルと興味単語行列にフィードバックをかけ、値を更新する.

フィードバックにより興味単語ベクトルと興味単語行列を更新することで、検索用クエリ生成時の確率を変え、それによりフィードバック前よりもユーザにとって有用な検索用クエリを生成するようになることがフィードバックを行う目的である.

本研究ではフィードバック手法として情報検索分野で広く用いられている適合性フィードバック手法を用いた.

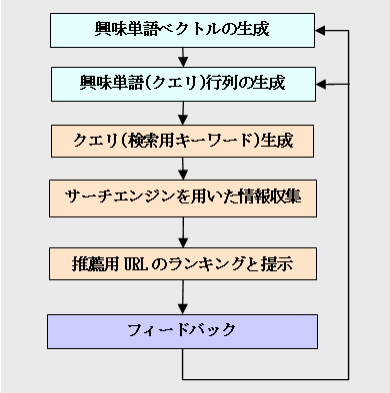

図 5　フィードバックの概要図

5.6.2 適合性フォードバックとは

適合性フィードバックとはユーザが得た検索結果のうち、どの文書が検索意図に適合し、どの文書が適合しないかを検索システムに教えることでシステムの検索精度を向上させる手法である.

適合性フィードバックでは、適合文書に含まれる検索語の重みを大きくし、逆に不適合文書に含まれる検索語の重みを小さくするように検索質問中の索引語の重みを調整する. 適合性フィードバックにはいくつかの手法があるが、次の式は適合性フィードバックの初期の研究者であるロッチオにより考案されたロッチオの式[10]である.

$$\mathbf{q_m} = \alpha\mathbf{q} + \frac{\beta}{|D_r|}\sum_{\forall \mathbf{d_j} \in D_r} \mathbf{d_j} - \frac{\gamma}{|D_n|}\sum_{\forall \mathbf{d_j} \in D_n} \mathbf{d_j} \quad (5.18)$$

この式は、文献検索に用いられるもので、$\mathbf{q}$ は前回の検索質問ベクトル、$\mathbf{q_m}$ は修正された検索質問ベクトル、D_r は適合文書、D_n は不適合文書、$\mathbf{d}_i$、$\mathbf{d}_j$ は文書ベクトルである. 第２項で適合文書に対するポジティブフィードバック、第３項では不適合文書に対するネガティブフィードバックを行っている. また α,β,γ は 0 以上の定数であり、それぞれ元の検索質問、適合文書、不適合文書をどの程度重視するかを表している.

5.6.3 適合フィードバックの本研究への適用

本研究でのフィードバック対象は興味単語ベクトル $\mathbf{T}$ と興味単語行列 Q である. 本研究のフィードバック手法はロッチオの式のように、直接 $\mathbf{T}$ や Q という記号を用いてベクトルとして(もしくは行列として)フィードバック式を構成するよりも、各成分のフィードバックに着目した方が説明しやすいため、ここでは各成分のフィードバックに着目して述べる.

定義 6

$\mathrm{usedFreq}(T_i)$ を興味単語 T_i から取得した推薦情報を、ユーザが利用した頻度を返す関数とする. このとき、興味単語 T_i から取得した推薦情報の結果を返す関数を $\mathrm{userResult}(T_i)$ とし次のように定義する.

$$\mathrm{userResult}(T_i) = \begin{cases} \mathrm{usedFreq}(T_i) & (\mathrm{usedFreq}(T_i) > 0) \\ -1 & (\mathrm{usedFreq}(T_i) = 0) \end{cases} \quad (5.19)$$

ここでは、ユーザが興味単語 T_i から取得した推薦情報を１度でも利用した場合は、$\mathrm{userResult}(T_i)$ 関数の値に利用頻度を用いたポジティブフィードバックを行う準備をしている. また、ユーザが興味単語 T_i から取得した推薦情報を１度も利用しなかった場合は利用頻度は 0 になるが、この場合は $\mathrm{userResult}(T_i)$ 関数の値を-1 とすることで、ネガティブフィードバックを行う準備をしている.

FT_i を興味単語 T_i のフィードバック用の重みとすると、FT_i は次の式によりフィードバックを行う.

$$newFT_i = FT_i + \mathrm{userResult}(T_i) \quad (5.20)$$

ここで $newFT_i$ はフィードバック後の FT_i を表している. 結果として FT_i の値は、推薦結果として興味単語 T_i が利用された場合には大きな値となり、利用されなかった場合にはネガティブフィードバックの影響で小さな値となる.

FT_i を用いて式(5.5)の興味単語の重み付けの際に利用した $feedback(T_i)$ 関数を次のように定義する.

$$feedback(T_i) = \begin{cases} \log(FT_i + 1) & (FT_i \geq 0) \\ -\log(-FT_i) & (FT_i < 0) \end{cases} \quad (5.21)$$

また、興味単語行列 Q のフィードバック手法は興味単語ベクトル $\mathbf{T}$ のフィードバック手法と考え方は全く同じであるため、ここでは説明を省く.

6. 評価

6.1 評価方法

情報検索分野ではテスト・コレクション等を用いた客観的な評価手法が確立しているが、情報推薦システム分野の研究は情報検索分野に比べると登場してまだ日が浅く、評価手法が確立されていない. そのため、情報推薦システムの評価は研究者ごとに異なる方法が用いられている. そこで本研究では、実際に数名のユーザにシステムを利用してもらい、その利用結果を用いてシステムの有用性の評価を行った.

6.2 評価結果

研究評価として、7 名のユーザに２週間程度、本研究で構築した ReXBrowser を利用してもらい、その利用データを解析することで研究評価を行った. 7 名のユーザは同じ研究室のメンバーや大学院生であり、全員が豊富なインターネット利用経験があり、一般的な利用者より

もコンピュータに関する知識もインターネットに関する知識も高いユーザである.

研究評価の最後の段階で、各ユーザに実際に役に立った推薦情報の数を質問し、この値を有用 URL の数とした.

評価結果は次の通りである.

表１　７名のユーザの利用結果

	推薦 Window 合計	推薦 URL 合計	推薦 Window の利用回数	推薦 URL の利用	有用 URL の数
User 1	103	2163	24	24	5
User 2	21	441	10	23	5
User 3	27	567	11	18	6
User 4	38	798	6	10	5
User 5	89	1869	11	23	8
User 6	45	945	9	16	6
User 7	54	1134	10	18	6

推薦情報の有効性の評価方法として本研究では、推薦 Window 利用率、推薦 URL 利用率、有用 Window 情報率、有用 URL 情報率の４つの指標を考案し、これを用いて評価を行った.

推薦 Window 利用率は推薦 Window の表示回数に対する、ユーザが利用した推薦 Window の数の割合とした. 推薦 URL 利用率は、推薦 URL の合計回数に対する、ユーザが利用した推薦 URL の数の割合とした. 有用 Window 情報率は、情報推薦 Window の表示回数に対する、ユーザにとって有用だった情報推薦 Window の割合とした。有用 URL 情報率は、推薦 URL の合計回数に対する、有用な推薦 URL の数の割合とした 。ただし、有用 Window 情報率計算において、ユーザにとって有用な情報がどの情報推薦 Window に含まれていたかまでは分からなかったため、ここでは各有用情報はそれぞれ１つの情報推薦 Window に含まれていたものと仮定して、その割合を求めた.

表２は７名のユーザの利用率、有用情報率の平均を表にしたものである。

表２　利用率,有用情報率の平均値

	平均値
推薦 Window 利用率	0.254755
推薦 URL 利用率	0.021805
有用情報率(Window)	0.139253
有用情報率(URL)	0.006631

図 6は７名のユーザの推薦 Window 利用率のグラフである.

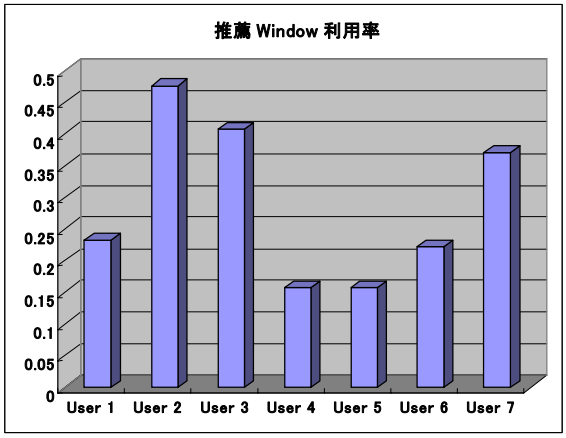

図 6　推薦 Window 利用率のグラフ

ユーザによる評価であるため、各ユーザによって利用状況にばらつきが見られたが、平均推薦情報 Window 利用率が 25. 4%と 4 回の提示に対して最低一つはユーザに利用される情報を提示できている. また平均有用 Window 情報率が約 14%であるから、7 回の情報推薦 Window の提示に対して最低１つはユーザにとって有用な情報が推薦できた. これにより、本研究で提案した手法によって、ユーザにとって有用な情報を推薦できることが確認できた.

７．おわりに

本稿では確率的手法を用いて構築したクエリにより情報を収集して、ユーザに情報を推薦する Web ページ推薦手法について提案を行った。評価実験を行った結果、提案手法によってユーザに対して有用な情報を提供できることが確認できた。

今後の課題として、本研究では採用しなかったブックマーク情報の利用、単語の重み付け計算手法の改良、クラスタリング手法等の利用による推薦精度の向上等がある。さらに情報推薦システムは評価手法が確立されていないため、客観的な評価手法の確立も課題である。

参考文献

[1] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon and J. Riedl. *GroupLens: Applying Collaborative Filtering to Usenet News.* In Communications of the ACM,volume 40, pp. 77-87, 1997.

[2] 森幹彦,山田誠二:ブックマークエージェント:ブックマークの共有による情報検索の支援,電子情報通信学会論文誌, J-83-D-I, pp. 487-494,2000.

[3] B. Mobasher. WebPersonalizer：A Server Side Recommender System Based on Web Usage Mining, Technical Report TR-01-004, 1991.

[4] Liren Chen and Katia Sycara.WebMate: A personal agent for Browsing and Searching. Proceedings of the 2nd International Conference on Autonomous Agents, pp. 132-139, ACM Press, 1998.

[5] 佐藤 健吾, 確率モデルによる Web ページ推奨エンジン,情報処理推進機構 平成 13 年度未踏ソフトウェア創造事業, 2001.

[6] Henry Lieberman, Letizia: An Agent That Assists Web Browsing, The Proceedings of the International Joint Conference on Artificial Intelligence, 1995.

[7] Corin R. Anderson, Eric Horviz, Web Montage: A Dyanmic Perosnalized Start Page ,WWW2002, ACM, 2002.

[8] Henry Lieverman, Chistopher Fry, and Louis Weitzman, Exploring the Web With RECONNAISSANCE AGENTS, Communications of the ACM August 2001/val. 44, No. 8, 2001.

[9] Salton, G. Wong, A. and Yang, C. S.:A Vector space model for automatic indexing,Communications of the ACM, Vol.18, No.11 pp.613-620, 1975.

[10] J. J. Rocchio. Relevance feedback in information retriebal. G. Salton, editor, The SMART Retrieval System – Experiments in Automatic Document Processing, Prentice Hall Inc., Englewood Cliffs, NJ,1971.