

# 適合率と再現率を用いたWebページランキングシステムの性能評価

庭野正義\* マッキンケネス ジェームス\*\* 永井保夫\*\*

**あらまし** 本研究では、ベイジアンフィルタを利用したユーザの嗜好情報を考慮するWebページランキングシステムの提案を行った。提案システムは、Webページ検索システム利用時の大量の検索結果の中から必要なページを判断する作業を自動化し、検索効率を向上する事を目的としている。そのために、ユーザの過去のWebページの閲覧状況からユーザの嗜好情報を抽出し、ベイジアンフィルタを用いて評価の高い順番でユーザに提示するシステムを提案し、試作をおこなった。Webページの評価では、検索を行うたびに变化するユーザの興味を、興味状態として表現し、興味状態毎にユーザの嗜好情報を記録し、ユーザの検索時の興味状態に応じた嗜好情報を用いた。実験により、1度の検索作業で十分な情報が収集できず、検索語を増減して与えることによって試行錯誤しながら必要な情報を収集する場合に検索効率が上がる可能性が高いということが示された。その結果、本研究で提案したシステムを利用することで、ユーザの嗜好に合ったWebページがより上位に表示され、ユーザの検索作業の負担軽減と検索効率の向上が期待できることが明らかになった。

**キーワード** : Webページ、ランキング、性能評価、適合率、再現率

## Performance Evaluation of Web Page Ranking System by Using Precision and Recall Ratios

Masayoshi NIWANO\*, Kenneth JAMES MACKIN\*\*, and Yasuo NAGAI\*\*

**Abstract** In this paper, we proposed Web page ranking system that considers user's preference information by using the Bayesian filtering. The proposed system automates works to collect necessary pages from a large amount of retrieval results when Web page search engines are used, and has aimed to improve the retrieval efficiency. We implemented and evaluated the Web page ranking system. In the evaluation of the Web page, the interest of the user, whenever the Web pages are searched, is changed and represented as the state of interest, and the user's preferences on each state of the interest are recorded, and the preferences corresponding to the state of the interest, when the Web pages are searched, are used. The experimental result shows that it is possible to improve the efficiency of the retrieval when necessary information is collected in trial-error manner by giving the retrieval words incrementally, while enough information cannot be collected. As a result, it is clarified that the Web pages suitable for the user's preference are displayed in the higher-order rank by using the proposed system, and the improvement of the retrieval efficiency can be expected.

**Keywords** : Web page, Ranking, Performance evaluation, Precision ratio, Recall ratio

\*東京情報大学 大学院 総合情報学研究科

Tokyo University of Information Sciences, Graduate School of Informatics

2010年4月よりアイコムシステック株式会社に所属

\*\*東京情報大学 総合情報学部 情報システム学科

Tokyo University of Information Sciences, Faculty of Informatics, Department of Information Systems

## 1. はじめに

GoogleやYahoo!、goo、msnといった検索サイトで検索する場合、その検索結果は膨大であり、かつ、必ずしもユーザ個人に適した順序で表示されているとは限らない。そのため、ユーザは、大量の検索結果の中からタイトル、概要などを見て、ユーザ自身がそのページにアクセスするかどうかを判断することが必要になる。大量の検索結果に対してこの作業を繰り返すにはかなりの労力が必要であり、その問題点を解消するための研究が精力的に行われている [1] [5] [10]。

我々は、この問題点の解決のため、ベイジアンフィルタを利用したWeb推薦システムの提案と試作、評価を行ってきた [6] [7] [8]。

本研究で提案するWebページランキングシステムは、ベイジアンフィルタを用いて検索システムから受け取った検索結果の文章（タイトルと概要、ホスト名）を解析し、ユーザが興味を持つ度合いを求め、その度合いの降順に検索結果を並び替える。ユーザが興味を持つ度合いの高い検索結果を上位に並べかえることで、「ユーザが検索結果を見てアクセスするかしないか判断する」手間を省き、検索の効率化を図る。本研究では、情報検索システムを正確性と網羅性の観点から評価するために使われている適合率と再現率ならびにF値 [2] を用いた性能評価を行う。

## 2. 試作システムの概要と特徴

### 2.1 試作システムの概要

スパムメールの自動振り分けでは、メールの文章を解析し、スパムメールかどうかを判断している。この作業は「受け取った大量の文章を2つのクラスに分類する」という点で、「検索結果を見てアクセスするかしないかを判断する」作業と非常に類似している。この点に着目し、本研究ではベイジアンフィルタを応用する事により、検索対象にユーザが興味を持つかを

判断できると考えることにする。本研究では、ベイジアンフィルタを利用し、Webページを順位付けするシステムを提案し、試作する。試作システムでは基本的に、スパムフィルタが行う「メールがスパムか非スパムかを判断する」という処理をそのまま「Webページを閲覧するかしないかを判断する」という処理に置き換える。スパムフィルタの場合は、スパムである確率に基づき、メールをスパムか非スパムかに分類する。一方、試作システムでは、検索結果をユーザの嗜好に合わせて並び替える事により、ユーザの嗜好に合ったページが上位に表示されるようになる。その結果、より早く目的のページにたどり着く確率が上がり、検索作業効率の向上が期待できる。

### 2.2 試作システムの特徴

試作システムは、ユーザがWebページを閲覧する確率（以下、推薦度とする）を計算するために、次の情報を利用する。

- ユーザが検索時に入力した検索語
- Google AJAX Search API [3] から返された検索結果のタイトル、概要、ならびにホスト名
- 返された検索結果中のWebページにアクセスしたか/しなかったかの情報

このような情報を利用する事で、以下のような利点が得られる。

a. ユーザの作業を増やさない 検索結果への評価値として、ユーザが検索結果のWebページにアクセスした/しなかったを1と0に対応させた2値を用いる。この値を用いることで、ユーザがWebページの評価をシステムに入力するという新たな手間が発生せず、Googleなどの既存の検索システムを利用する場合と変わらない作業量で検索を行える。

b. 大量の嗜好情報が手に入る ユーザの評価を、「ページのタイトル、概要、ホスト名を見て、実際にそのページへアクセスした/しなかった」を1と0に対応させた2値とする。それにより、評価値入力のための新たな作業が発生

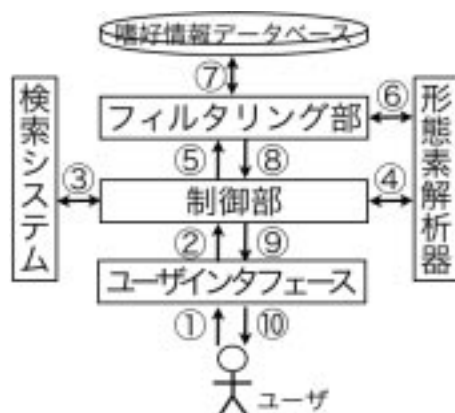


図1. システム構成

しないため、ユーザが評価したページ全てをシステムに入力できる。

c. 全く新しいWebページでも適切に評価できる検索結果のタイトル、概要、ホスト名を取得し、その文章の特徴と、ユーザの嗜好情報を比較する。そうすることにより、ユーザやシステムが初めて見たWebページでも、ある程度適切な評価を行う事ができる。

### 2.3 システム構成と処理概要

Webページランキングシステムの構成は図1のようになっている。以下では、提案したWebページランキングシステムの処理概要について説明する。

Step 1 検索ワードを受け取る Googleなどの一般的な検索システムと同じように、ユーザは、ユーザインタフェースを通してシステムに検索語を入力する(①)。入力された検索語を、制御部が受け取る(②)。

Step 2 既存検索システムでの検索 制御部は、受け取った検索語をそのままGoogle AJAX Search APIへ送信し、検索結果の集合を受け取る(③)。今回は、Google AJAX Search APIから最大32個の検索結果を取得している。

Step 3 検索興味状態の取得 制御部は、入力された検索語を形態素解析器へ送り、形態素集合を受け取る(④)。受け取った形態素集合から名詞、動詞、未知語のみを抜き出し、それらを

興味状態とする。

Step 4 推薦度の計算 制御部は、興味状態と検索結果をフィルタリング部へ送り(⑤)、検索結果毎に推薦度を取得する(⑥、⑦、⑧)。

Step 5 ユーザへの提示 制御部は、検索結果集合を、取得した推薦度の降順で並び替え、上位から順番にユーザに提示する(⑨、⑩)。

Step 6 ユーザの嗜好情報の取得 制御部は、提示された推薦結果のWebページを、ユーザが実際にアクセスしたか/しなかったかという情報を受け取り、フィルタリング部へ送る(①、②、⑤)。

Step 7 嗜好情報に基づき、データベースを更新する フィルタリング部は、受け取ったユーザの嗜好情報を基に、データベースを更新する(⑥、⑦)。

以上のStep 1 からStep 7 を繰り返す事で、検索結果の再順位付けによる推薦とユーザの嗜好情報データベースの更新が行われる。

### 2.4 興味状態の導入と取得

試作システムは、「ユーザがどのような項目を調べたいか」ということを興味状態として表現する。

検索結果の再順位付けを適切に行うために、ユーザがどのような興味状態であるかを把握した上でWebページの推薦度を求めなければならない。例えば、普段、料理について調べ、料理のレシピが記述されたページに興味を持つことがわかっていたとする。その場合に本を検索をしている時に、料理のレシピが書いてあるWebページを上位に表示するのは、検索時のユーザの興味を反映していないと考えられる。したがって、検索を行うたびに变化するユーザの興味を、興味状態として表現し、管理する必要がある。そこで、試作システムでは、ユーザの興味を興味状態として表現し、興味状態毎に別々の嗜好情報を記録する方法をとった。検索語が興味状態を表していると仮定し、「検索語の形態素の中から名詞、動詞、未知語のみを抜き出したものと、それらの中から2つの形態素

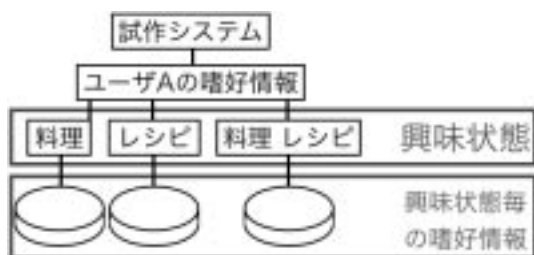


図2. 興味状態と嗜好情報

を組み合わせたものの和」を興味状態として定義する。形態素とは、文章の中で意味を持つ最小の単位である。例えば、「料理レシピ」という検索語で検索した場合の検索結果は、「料理」、「レシピ」、「料理レシピ」という3つの興味状態に属しているとみなされ、ここで取得した嗜好情報は、図2のように興味状態別に記録される。図2は、試作システムがユーザAの中でも、興味状態毎に別々に嗜好情報を記録していることを表している。このように興味状態を取得し、取得した興味状態毎に嗜好情報の記録を行うことで、検索時のユーザの興味を反映できようになるため、より適切な再順位付けを行える。

## 2.5 推薦度の計算

本節では、トークンの定義を説明した後、2.3節のStep 4において推薦度を求める手順を説明する。

### 2.5.1 トークン

ここでは、トークンを「文章を分割する単位」と定義する。今回は、2種類のトークンの取得方法を採用し、それぞれの評価を行った。1つ目は、一般的なトークン取得方法と同じで、「形態素解析器により分割された1つの形態素」を1つのトークンとしてデータベースに記録する方法である。2つ目は、ある程度トークン同士の関係に注目するようにした方法である。ここでは、「形態素解析器により分割された1つの形態素」に加え、連続する5つの形態素のうち2つの形態素を組み合わせたものを1つのトークンとして扱う。

例えば、「Web推薦システム」という文章を

分割するとき、1つ目の方法では「Web」、「推薦」、「システム」という3つのトークンが得られ、2つ目の方法では「Web 」、 「推薦」、 「システム」、「Web 推薦」、「Web システム」、「推薦システム」というトークンが得られる。

### 2.5.2 Webページの推薦度

Webページの推薦度 $P(D, w)$ は式(1)で表される。ここでは、「検索ワード $w$ を与えられた時のWebページ(ドキュメント) $D$ の推薦度」を $P(D, w)$ 、「ユーザが入力した検索ワード」を $w$ 、「検索ワード $w$ に含まれる興味状態の数」を $n$ 、「検索ワード $w$ に含まれる $i$ 番目の興味状態( $1 \leq i \leq n$ )」を $c_i$ 、「興味状態 $c_i$ が与えられた時のドキュメント $D$ の推薦度」を $P(D, c_i)$ と表す。

2.5.3で説明する式(2)により、 $P(D, c_i)$ を求め、式(1)に代入し $P(D, w)$ を求める。

$$P(D, w) = \frac{\prod_{i=1}^n P(D, c_i)}{\prod_{i=1}^n P(D, c_i) + \prod_{i=1}^n (1 - P(D, c_i))} \quad (1)$$

### 2.5.3 興味状態毎の推薦度

検索興味状態毎の推薦度 $P(D, c_i)$ は式(2)で表される。ここでは、 $c_i$ を2.5.2節で説明した式(1)と同じものとし、「興味状態 $c_i$ が与えられた時のドキュメント $D$ の推薦度」を $P(D, c_i)$ 、「ドキュメント $D$ に含まれているトークンの数」を $m$ 、「興味状態 $c_i$ が与えられた時の、ドキュメント $D$ に含まれる $j$ 番目のトークン $t_j$ の推薦度( $1 \leq j \leq m$ )」を $P(t_j, c_i)$ と表す。2.5.4で説明する式(3)により、 $P(t_j, c_i)$ を求め、式(2)に代入し $P(D, c_i)$ を求める。

$$P(D, c_i) = \frac{\prod_{j=1}^m P(t_j, c_i)}{\prod_{j=1}^m P(t_j, c_i) + \prod_{j=1}^m (1 - P(t_j, c_i))} \quad (2)$$

### 2.5.4 トークン毎の推薦度

トークン毎の推薦度 $P(t_j, c_i)$ は式(3)で表される。 $c_i, t_j$ は、2.5.3節で説明した式(2)の $c_i, t_j$ と同じものとし、「興味状態 $c_i$ が与えられた時のトークン $t_j$ にユーザが興味を持った回数」を $MC(t_j, c_i)$ 、「興味状態 $c_i$ が与えられた時のトークン $t_j$ にユーザが興味を持たなかった回数」を

表1. 嗜好情報データベース

	トークン	興味状態	選択回数	非選択回数
レコード1		web	10	25
レコード2	システム	web	3	5
	...	...	...	...

$NC(t_i, c_i)$ 、「興味状態に属するトークン全てのユーザが興味を持った回数の合計」を $MC(c_i)$ 、「興味状態に属するトークン全てのユーザが興味を持った回数の合計」を $NC(c_i)$ と表す。

$MC(t_i, c_i)$ 、 $NC(t_i, c_i)$ 、 $MC(c_i)$ 、 $NC(c_i)$ は、ユーザの嗜好情報が記録されているデータベース(2.6節参照)から取得する。データベースにどのようにユーザの嗜好情報が記録されているかは2.6節で説明する。

$$P(t_i, c_i) = \frac{\frac{MC(t_i, c_i) + 1}{MC(c_i) + 1}}{\frac{MC(t_i, c_i) + 1}{MC(c_i) + 1} + \frac{NC(t_i, c_i) + 1}{NC(c_i) + 1}} \quad (3)$$

## 2.6 ユーザの嗜好情報

ユーザの嗜好情報は、表1のデータベースに格納される。第1フィールドにトークン、第2フィールドに興味状態が格納される。この2つのフィールドが主キーとなる。第3フィールドには、「第2フィールドの興味状態に属する第1フィールドのトークン」にユーザが興味を持った回数、第4フィールドには、「第2フィールドの興味状態に属する第1フィールドのトークン」にユーザが興味を持たなかった回数を格納する。

第1フィールドであるトークンが空のレコード(表1のレコード1)には、その興味状態全体に対する嗜好情報(式(3)の $MC(c_i)$ と $NC(c_i)$ )が記録され、トークンがあるレコード(表1のレコード2)にはその興味状態に属するトークンに対するユーザの嗜好情報(式(3)の $MC(t_i, c_i)$ と $NC(t_i, c_i)$ )を記録する。

例えば、表1の場合、レコード1は、「web」という興味状態で検索された時の検索結果が合計35個であり、35個のうち10個の検索結果に興

味を持った事を表している(式(3)の $MC(c_i) = 10$ 、 $NC(c_i) = 25$ )。レコード2は「web」という興味状態で検索された結果の中に、「システム」というトークンが合計8個含まれており、その8個のうち、3個の検索結果に興味を持ったという事を表している(式(3)の $MC(t_i, c_i) = 3$ 、 $NC(t_i, c_i) = 5$ )。

このようなデータベースを作成し、ユーザの嗜好の情報を記録しておく事により、2.3節のStep4の推薦度の計算に必要なユーザの嗜好情報(式(3)で利用する $MC(t_i, c_i)$ 、 $NC(t_i, c_i)$ 、 $MC(c_i)$ 、 $NC(c_i)$ の値)を求める事ができる。2.3節のStep7では、ユーザの嗜好情報を受け取り、データベースを更新する。

## 3. 実験方法

試作システムは、Googleなどの既存Webページ検索システム利用時の大量の検索結果の中から必要なページを判断する作業を自動化し、検索効率を向上する事を目的としている。目的の達成度合いを示す指標として、情報検索システムを正確性と網羅性の観点から評価するために使われている適合率と再現率ならびにF値を用いた。適合率は、検索システムが検索結果として表示した情報の中に、どれだけユーザの要求が満たされている検索結果を含んでいるかの割合を示す。一方、再現率は、検索システムが検索する文章中全ての適合文章のうち、検索結果に表示できた文章の割合である。F値は、適合率と再現率を組み合わせて、総合的に評価するための指標である。収集したデータを用いて、ユーザの検索作業をシミュレートし、試作システムを利用し再ランク付けを行った検索結果を、再ランク付けを行わない検索結果と比べ、どれだけユーザが選んだ文章が上位に移動するかを調べる。本実験では、表示された検索結果と実際のWebページの内容が一致していない場合は考えない事とする。つまり、検索結果の内容が、Webページの内容を正しく要約しているものと仮定している。

### 3.1 収集データ

被験者 8 名に、ユーザの閲覧記録収集のために新たに作成した実験用Webページを利用し、自由に検索を行わせ、閲覧記録を収集した。2009年11月16日から2009年11月29日までの14日間データを収集し、その間に143回の検索作業が行われた。ユーザ毎の検索回数は、7回から25回となった。閲覧記録は、被験者が入力した検索語、Google AJAX Search API から返ってきた検索結果（タイトル、概要、ホスト名）、被験者が選択した検索結果で構成される。閲覧記録を用いて、Google AJAX Search API から受け取った順番をそのままユーザに提示した場合と、ユーザの嗜好情報を用いて検索結果の順番を入れ替えた場合の比較を行い、システムを評価する。

### 3.2 評価方法

試作システムは、Google AJAX Search API から受け取った検索結果の並び替えを行うだけであるため、前述の適合率と再現率の定義をそのまま適用すると、Google AJAX Search API と全く同じ値になってしまう。そのため、本研究では、新たに適合率' と再現率' をそれぞれ式 (4)、式 (5) として定義し、その尺度を用いて評価を行った。今回の実験では、試作システムの上位に表示される検索結果 8 件の情報を用いて評価した。

#### a. 適合文章

今回の実験では、収集したデータに含まれる検索結果の中で、ユーザが実際にアクセスした検索結果を適合文章と定義した。例えば、ある検索結果集合に対して、ユーザが表 2 のような評価を行ったとする。表 2 は、検索結果集合とユーザの評価情報を表している。この場合、ユーザは順位 1 位と 3 位の検索結果を選択しているため、1 位と 3 位の検索結果を適合文章とし、その他の文章は非適合文章とする。この定義を用いて、以下で説明する適合率'、再現率' ならびにF値で評価を行う。

#### b. 適合率'

表2. ユーザの評価情報例

順位	1	2	3	4	5	6	7	8
選択したか	YES	NO	YES	NO	NO	NO	NO	NO

適合率' を検索結果上位  $x$  件中の適合文章の数と定義する。適合率' は式 (4) で求められる。表 2 を例に具体的な計算例を示す。表 2 のように検索結果の上位 8 件をユーザが評価し、そのうちの 1 件目と 3 件目の検索結果を有益だと判断し、クリックしたとする。このときの上位 2 件の適合率は、 $Precision' = \frac{1}{2} = 0.5$  となる。このシステムを用いた場合、上位  $x$  件中にどれくらいの割合で適合文章が含まれているかを知ることができる。

$$\begin{aligned} & \text{上位 } x \text{ 件までの } Precision' \\ &= \frac{\text{検索結果上位 } x \text{ 件中の適合文章の数}}{x} \quad (4) \end{aligned}$$

#### c. 再現率'

再現率' をユーザが評価した検索結果中の適合文章の割合と定義する。再現率' は式 (5) で求められる。表 2 を例として説明すると、上位  $x$  件までの  $Recall' = \frac{1}{2} = 0.5$  となる。この値から、ユーザが選択した適合文章のうち、上位  $x$  件に入っている適合文章の割合を知ることができる。

$$\begin{aligned} & \text{上位 } x \text{ 件までの } Recall' \\ &= \frac{\text{上位 } x \text{ 件中の適合文章数}}{\text{ユーザが評価した適合文章数}} \quad (5) \end{aligned}$$

#### d. F値

適合率と再現率には、トレードオフの関係がある。適合率と再現率を総合して評価する値として、F値を用いる。F値は式 (6) によって求められる。表 2 を例として説明すると、上位 2 件の  $F\text{値} = \frac{2}{\frac{1}{0.5} + \frac{1}{0.5}} = 0.5$  となる。

$$F = \frac{2}{\frac{1}{Recall'} + \frac{1}{Precision'}} \quad (6)$$

### 3.3 実験手順

被験者は、実験用のWebページを用いて検

索を行った。被験者は、実験用のWebページに配置されているテキストフィールドに検索語を入力し、sendボタンを押す事で、検索を行う。試作システムは、中央に検索結果が4つずつ表示され、推薦度による並び替えは行わない。さらに、nextボタンやprevボタンを押す事で、前後の検索結果を表示する。記録した情報に基づき、ユーザが選択した検索結果を適合文章とし、再ランク付けを行った場合とそうでない場合の適合率' と再現率' ならびにF値を求め、比較する。再ランク付けを行った場合とそうでない場合の比較は、以下の手順で行われる。

Step 1 被験者の閲覧記録を収集する

Step 2 収集したデータから検索1回分の記録を読み込む

Step 3 再ランク付けを行わない場合の上位  $x$  件の適合率' と再現率' を記録する ( $1 \leq x \leq 8$ )

Step 4 検索結果の推薦度を求める

Step 5 検索結果を推薦度の降順に並び替える

Step 6 並び替えた結果の上位  $x$  件の適合率' と再現率' を記録する ( $1 \leq x \leq 8$ )

Step 7 閲覧記録の情報を基に、嗜好情報を更新する

Step 8 閲覧記録がなくなるまでStep 2 から Step 7 を繰り返す

Step 9 Step 1 からStep 6 までで求められたそれぞれの上位  $x$  件の適合率' と再現率' およびF値を比較する

上述のStep 2 からStep 9 を4章の実験結果に示されるように、条件を変えて2回行った。1回目は実験1として、2.3節で説明したStep 1 からStep 7 を繰り返すことで推薦度の計算と学習を行う。検索語と検索結果およびユーザの評価情報は、実験手順のStep 1 で収集したデータを用いる。さらに、試作システムの興味状態の取得方法を改善し、ユーザの過去の嗜好情報を効率よく使用できるようにすることで、適合率' と再現率' およびF値が改善すると考える。そこで、2回目は実験2として、興味状態

表3. 実験1で使われていた検索語

検索された順番	検索語
1	物理演算
2	物理演算について
3	物理演算とは
4	物理演算とは
5	物理演算
6	物理演算とは
7	物理演算について
8	物理演算とは何か
9	物理演算って何
10	物理演算とは 何

表4. 実験1の結果比較

$x$	再順位付けあり			再順位付けなし		
	適合率'	再現率'	F 値	適合率'	再現率'	F 値
1	0.27	0.18	0.22	0.09	0.05	0.06
2	0.18	0.23	0.20	0.09	0.09	0.09
3	0.15	0.25	0.19	0.12	0.20	0.15
4	0.11	0.25	0.16	0.09	0.20	0.13
5	0.09	0.25	0.13	0.07	0.20	0.10
6	0.08	0.25	0.12	0.07	0.25	0.12
7	0.06	0.25	0.10	0.06	0.25	0.10
8	0.06	0.30	0.11	0.05	0.25	0.09

の取得方法(2.3節で説明したStep 3)を改良し、関連する検索語を解析させ、適切な興味状態を取得できるようにして、推薦度の計算を行う。このようにすることで、より効率よくユーザの嗜好情報を再順位付けに使用できると考える。この実験では、人手により興味状態の指定を行った。

## 4. 実験結果

### 4.1 実験1

表3は、検索語「物理演算」を固定し、それ以外の検索語を増減して与えることで「物理演算」についての情報を収集している例を示している。この一連の検索を行ったときの上位  $x$  件での適合率' と再現率' およびF値の平均値を、再順位付けを行った場合と再順位付けを行わなかった場合で比較した結果を表4に示す。

適合率', 再現率', F 値

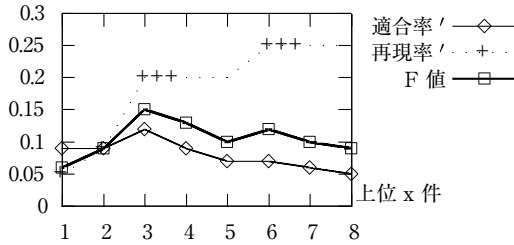


図3. 再順位付けなしの結果 (実験1)

適合率', 再現率', F 値

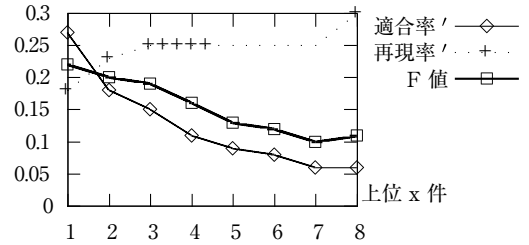


図4. 再順位付けありの結果 (実験1)

表5. 実験2で同じ興味状態と判断した検索語

検索された順番	検索語
1	レゴ 開発環境
2	レゴ 開発環境 lisp
3	レゴ lisp
4	lisp
5	lisp 研究
6	lisp 自律
7	自律システム
8	自律
9	自律 プログラム
10	自律
11	自律とは
12	自律 ロボット
13	自律 ロボット 研究
14	LISP レゴ 研究

表6. 実験2の結果比較

x	再順位付けあり			再順位付けなし		
	適合率'	再現率'	F 値	適合率'	再現率'	F 値
1	0.79	0.35	0.49	0.71	0.33	0.45
2	0.61	0.45	0.51	0.54	0.40	0.46
3	0.55	0.54	0.54	0.52	0.51	0.51
4	0.52	0.66	0.58	0.52	0.66	0.58
5	0.43	0.67	0.52	0.49	0.74	0.59
6	0.39	0.72	0.51	0.42	0.75	0.54
7	0.35	0.73	0.47	0.36	0.75	0.48
8	0.35	0.80	0.49	0.35	0.80	0.49

表4の左側の列は、適合率の取得時に使用した上位 x 件の値を示す。x の列を除いた左半分の適合率' と再現率' ならびにF値は、試作シス

テムを用いてランキングを行った場合の値である。表4の右半分の適合率' と再現率' ならびにF値は、Google AJAX Search API を用いて取得した検索結果の順位をそのまま利用した場合の値である。図3および図4は、表4をグラフとして表現したものである。図3および図4では、それぞれ縦軸が適合率' と再現率' ならびにF値の平均値を示し、横軸が適合率の計算に使った上位 x 件の x の値を示している。以上のような場合には、再順位付けを行った場合の方が再順位付けを行わなかった場合と比較し、適合率' と再現率' ならびにF値が改善している。

#### 4.2 実験2

実験2では、表5に示すように一連の検索作業で得られた検索結果を、「LISP 処理系を搭載したLEGO MINDSTORMS で、自律ロボットを作成する」という1つの興味状態とみなした。表5の一連の検索作業で用いられた検索語である「レゴ」は、他に使用されている検索語である「開発環境」、「lisp」、「ロボット」などから、「LEGO MINDSTORMS」の事であると考えられる。LEGO MINDSTORMS とは、レゴブロックとコンピュータを用いてロボットプログラミングを体験できるキットである。このキットでは、lisp言語で制御ができる環境が提供されている。これらの事から、この一連の検索作業は、lisp 処理系を搭載したLEGO MINDSTORMS で自律ロボットを作成するために必要な情報を収集するために行われたと判



適合率', 再現率', F 値

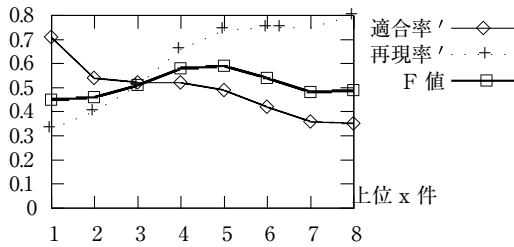


図5. 再順位付けなしの結果 (実験2)

断できる。現在の試作システムの興味状態の取得方法では、「レゴ開発環境」と「lisp研究」のように、検索語の中に同じ形態素が使われていない場合、「レゴ開発環境」で取得したユーザの嗜好情報は「レゴ」や「開発」、「環境」、「レゴ開発」、「レゴ環境」、「開発環境」という興味状態用の嗜好情報として記録されてしまう。そのため、「lisp」や「研究」、「lisp研究」という興味状態に属する検索結果の推薦度計算に使用することができない。これに対して、表5に示す一連の検索作業で得られた検索結果を、「LISP処理系を搭載したLEGO MINDSTORMSで、自律ロボットを作成する」という1つの興味状態として処理を行う場合、「レゴ開発環境」で取得したユーザの嗜好情報を、「lisp研究」での推薦度計算に使用することができる。

そこで、実験2では、表5に示す一連の検索作業を、1つの興味状態として認識させ、この一連の検索の上位  $x$  件の適合率' と再現率' ならびにF値の平均値を求めた。ここで求めた値と、再順位付けを行わない場合の上位  $x$  件の適合率' と再現率' ならびにF値を比較した結果を表6に示す。図5および図6は、表6のグラフ表現である。 $x$  が1から3までの適合率' を見ると、再順位付けありの値が再順位付けなしの値と比べて高いことがわかる。このことから、試作システムの再順位付けにより、適合文章が上位へ移動したと考えられる。この結果から、試作システムの興味状態の取得方法を改善し、より適切なユーザの興味状態を把握すること

適合率', 再現率', F 値

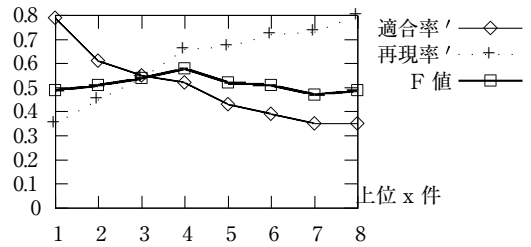


図6. 再順位付けありの結果 (実験2)

で、適合率' と再現率' ならびにF値の改善が期待できる。

## 5. 考察

実験1の結果から、検索語を固定し、それ以外の検索語を増減して与えることで情報収集を行う場合を考える。これは、1度の検索作業で十分な情報が収集できず、検索語を増減して与えることによって試行錯誤しながら必要な情報を収集する場合に該当する。そこでは、適合率' と再現率' ならびにF値の改善がみられた。その理由は、連続した検索語を用いて1つの対象について検索を行うため、データベース中の嗜好情報と検索時点でのユーザの嗜好情報のずれが少なくなり、以前の検索作業で収集した嗜好情報が再順位付けに効果的に影響し、適合率' と再現率' ならびにF値の改善につながったためだと考えられる。しかしながら、上述の場合以外の場合、つまり、前回の検索と今回の検索に全く関係がない場合には、適合率' と再現率' ならびにF値が改善しない場合がみられた。試作システムは、検索語の意味を考慮せずに興味状態を取得するため、検索語同士の概念的な包含関係や、同義語、類義語を考慮できず、ユーザの興味状態とのずれが生じてしまうため、ユーザの興味状態とのずれが発生し、適切でない学習が行われなかったためだと考えられる。

実験2の結果では、興味状態の取得方法を改善し、学習を行うことで、適合率' と再現率'

ならびにF値の改善が見られた。その理由は、興味状態を人手によって指定することにより、ユーザの興味状態とのずれを減少させたからであると考えられる。実験2で行った興味状態の取得方法をシステム化することが今後の課題である。

## 6. まとめと今後の課題

本論文では、ベイジアンフィルタを利用したWebページランキングシステムの試作と実験による評価および考察を行った。その結果、1度の検索作業で十分な情報が収集できず、検索語を増減して与えることによって試行錯誤しながら必要な情報を収集する場合に、適合率'と再現率'ならびにF値の改善がみられた。さらに、検索語の意味を考慮してより適切なユーザの興味状態を取得することにより、適合率'と再現率'ならびにF値が改善する可能性があることを確認できた。

今後の課題としては、次の2点が考えられる。

### a. 興味状態の取得方法の改良

試作システムでは、検索語の意味を考慮せずに分かち書きを行い、その組み合わせから興味状態を取得するため、適切なユーザの興味状態を取得する事ができない場合がある。この問題を解決するためには、検索語の意味まで考慮した興味状態の取得方法を考える必要がある。また、興味状態や検索語の意味を正しく認識するためには、同義語、類義語辞書や、シソーラス辞書などを用意することも必要である。

### b. 実験方法の見直し

ユーザの嗜好情報を考慮したランキングや推薦を行うシステムの評価を適合率'と再現率'ならびにF値で行う場合、どのWebページを適合文章するかが非常に重要となる。試作システムでは、普段の検索作業の効率を向上を目的としているため、今回の実験では、ユーザに自由に検索をさせ、その検索の中で実際にクリックされたWebページを適合文章とみなした。一方、類似研究 [5] では、検索語を指定し、検

索語に関連する検索結果を、被験者に全て選択させ、そのWebページを適合文章としており、比較的高い適合率と再現率、ならびにF値となっている。類似システムと比較する場合、今回の実験方法で適合文章を取得するのでは無く、類似システムと同じ方法で適合文章を取得することが必要となる。

## 【文献】

- [1] 天野環, 中里秀則, 中村隆史: ベイズ推定を用いたWebマイニング, 電子情報通信学会技術研究報告Vol.104, No724 (2005), pp.43-48.
- [2] D.M. Christopher, R. Prabhakar, and S. Hinrich: Introduction to Information Retrieval, Cambridge University Press, New York, 2008.
- [3] Google AJAX Search API - Google Code : <http://code.google.com/intl/ja/apis/ajaxsearch/>.
- [4] 石川徹也, 宇田隆幸: 情報フィルタリングの利用システム: 情報推薦システム (<特集>情報のフィルタリング), 情報の科学と技術Vol.59, No10 (2006), pp.458-463.
- [5] 國貞暁, 山本けい子, 田村哲嗣, 速水悟: 要約情報の類似度を用いたWEB検索支援システム, 第21回人工知能学会全国大会, Miyazaki, JSAI, 2007.
- [6] 庭野正義, Kenneth James Mackin, 永井保夫: ベイジアンフィルタを利用したWeb推薦システムの試作と評価, 電子情報通信学会2009総合大会D-8-13, Ehime, IEICE, 2009.
- [7] 庭野正義, K.J.Mackin, 永井保夫: ベイジアンフィルタを利用したWeb推薦システム, 日本ソフトウェア科学会第26回大会3A-2, Shimane, JSSST, 2009.
- [8] 庭野正義, K.J.Mackin, 永井保夫: ベイジアンフィルタを利用したWebページランキングシステム, 社会システムと情報技術研究ウィーク, Hokkaido, SIG-AI, 2010.
- [9] POPFile - Automatic Email Classification - Trac : <http://getpopfile.org/>
- [10] 高須賀清隆, 丸山一貴, 寺田実: 閲覧履歴を利用した協調フィルタリングによるWebページ推薦とその評価, 電子情報通信学会技術研究報告Vol.107, No131 (2007), pp.115-120.