ユーザの閲覧履歴に基づくオンライン検索支援システム

An Online Query Suggesting System Based on User Browsing History

永井 洋一 †
Yoichi NAGAI

† 東京大学新領域創成科学研究科基盤情報学専攻
Dept. Graduate School of Frontier Sciences, University of Tokyo
nagai@logos.k.u-tokyo.ac.jp

近年の IT 技術の進展に伴い、大量のデータが Web 上に蓄積されつつある。Web 上から情報を検索する時は一般的に欲しい情報と関連が深いと思われる単語をクエリとして入力するが、ユーザによっては必要な欲しい情報に関連の深い単語がわからない場合が考えられる。自分や他のユーザの過去の履歴情報に基づいて検索支援を行う研究がなされているが、過去の履歴とはまったく違った領域に関する検索を行うときや、言語の多義性により同じクエリを入力したユーザでも欲しい情報が違っている場合などでは、うまくいかないことが考えられる。そこで我々は、ユーザのその検索内のみの履歴を基に、ユーザの必要であるページを最もよく表すと考えられるキーワードを提示するシステムを提案し、簡単な模擬実験による検証を行った。実験結果として、閲覧を進めるごとにそれなりの精度でユーザのニーズに沿った単語が提示することができたが、反面、クエリとしての単語によるユーザのニーズの表現力の問題などもあきらかになった。

1 はじめに

今日、急速なコンピュータ技術や情報通信技術の発展により計算機で扱えるデジタルデータが大量に蓄積されている。そのため、Web 上の情報は網羅性という観点からとても重要な位置を占めるに至っている。しかし情報が増えればそれだけ網羅性が増す反面、情報を検索する立場の人にとって必要な情報を膨大な Web 空間から抽出する手間が増える傾向にあり、また検索する人の検索技術の差が情報獲得の機会の差として顕著に現れると考えられる。

こうした問題に対応するために、検索エンジンを運営する Google や Yahoo などの企業は日々ユーザが使いやすいような機能を開発しており、また多くの研究者もデータマイニングや機械学習などの技術を用いることによって様々な研究がなされている [1, 3, 4, 10]。

しかし一般に人は言語について多義的に解釈するため、同じクエリでも人によって違うものを求めているような場合があり、表層的な特徴のみからなんらかの示唆をおこなうのは難しい。そのためこのようなあいまいな人間の行動を扱えるように、ユーザや他のユーザなどの閲覧履歴などを利用して各ユーザの嗜好に沿った検索結果を出せるような方法が提案されている。しかしこれらの方法ではユーザのモデルを構築するために手間や時間がかかったり、ユーザの今までの閲覧履歴を新しい検索に用いることが難しい場合もある。

そこで本研究ではユーザがその場での閲覧した履歴だけを基になるべく少ないユーザの行動履歴からユーザがページを見ている間に逐次的にユーザのニーズにそったページの提示を行うシステムを提案する。

まずユーザの Web 検索を支援する方法としてどのようなアルゴリズムがあるのかを説明した後、、それらの問題点などの考察をおこない、次にそれらの問題に対応するための提案手法について説明を行う。2 章では検索エンジンにおけるブラウジング支援についての関連研究の説明を、3 章ではそれらの問題点とそれに対する提案手法について説明を行い、4 章では提案した手法の有効性を検証する実験について述べる。

2 関連研究

ユーザの Web 検索を支援する方法には様々なものがあるが、現行の多くの検索エンジンでは最初に、ユーザが欲しい情報と関連があると思う単語（クエリ）を検索エンジンに入力する形式を取っている。しかし Web 上では大量のサイトが存在するため一般には入力したクエリにマッチするページが大量に存在する。そこでヒットしたページをユーザが知りたいページを効率良く閲覧できるように提示する必要がある。ページを重要であると考えられる順にソート

して提示する方法が一般的であるが、他にもユーザの入力したクエリにそって、検索結果をなにかの共通トピックごとにクラスタリングを行うものや、入力したクエリにとって関連の深いキーワードを提示するもの、あるいはユーザの閲覧履歴により最適なキーワードを推定、提示を行うものなどがある。

こうした検索結果のページを提示するために、主に次の 3 つの手段がよく用いられている。

1. リンク構造に基づく解析
2. ページ内容に基づく解析
3. ユーザの閲覧履歴に基づく解析

リンク構造に基づく解析は先に述べた PageRank [2] などが代表的なアルゴリズムである。あるいはリンク構造により内容的に似た Web ページの集合を抽出する研究なども行われている。

ページ内容に基づく解析には様々な手法が考案されている。解析する内容としては Web ページを構成する単語を基に分析を行う。代表的なページ内容に基づいた方法は、Web ページを内容に基づいて似たトピック同士をクラスタ化するクラスタリング [6] などがある。

ユーザの閲覧履歴を基づいた解析としては、アクティブユーザ自身の閲覧履歴を用いてユーザの嗜好に即した検索支援を行うパーソナライズや、他のユーザの履歴と自分の履歴と照らし合わせて自分の嗜好に似ていると思われる他ユーザの閲覧履歴から検索支援を行う協調フィルタリングなどがよく利用されている。

以下ではページ内容とユーザの閲覧履歴に基づく解析について紹介する。

2.1 自分の過去の閲覧履歴を基にした支援

同じクエリであってもユーザによって求めている情報が違う場合に対応するために、そのユーザの嗜好を抽出して利用し、検索結果をそのユーザに合わせてカスタマイズするパーソナライズ化が近年活発に研究されており、Google などではパーソナライズド検索などのサービスも実用化されている。

パーソナライズをする際にはユーザのモデルを構築するのであるが、このモデルを構築する方法にもいろいろあり、ユーザの過去の Web ページ閲覧履歴や、ユーザのブックマーク、質問形式でユーザに入力してもらったプロフィール情報をなどを基にユーザのモデルを構築する。

Web ページなどの文書検索の場合は単語のベクトルでユーザのモデルを表現する研究が多い。ここではユーザの過去の閲覧履歴を基に単語を用いたユーザモデルの構築する例を以下の式で示す。

$$P_{user} = (p_{t1}, p_{t2}, t_{tm}) \tag{1}$$

$$P_{t_k} = \frac{1}{S_N} \sum_{hp=1}^{S_n} c^{hp} \frac{tf(t_k, hp)}{\sum_{s=1}^{m} (t_s, hp)} \tag{2}$$

Sn: ユーザが閲覧したページ集合

m: ページ集合に出現した単語数

c: 一定時間以上閲覧時間が長い場合は 1, 以下なら 0

このモデルは各単語の出現するページをユーザが志向する程度の強さを表している。このモデルを用いて検索結果のページをユーザの嗜好に合ったように並び替えることを考えると各ページの単語ベクトルとユーザモデルの内積をページのスコアとして、高いスコアほどユーザの要望に一致していると考えることができる。

2.2 他人の過去の閲覧履歴を基にした支援

パーソナライズがユーザ本人のモデルのみを基に検索支援を行っていたのに対して、他のユーザのモデルを構築しておいて、ユーザの嗜好とよく一致する他ユーザの嗜好をそのユーザに当てはめる協調フィルタリングという手法がある [7]。amazon.com などでは、ユーザのこれまでの本の購買履歴と同じような履歴を持つ人の多くが買った本を推薦するシステムが存在するがこれも協調フィルタリングの例として挙げられる。ユーザのモデル構築が不完全な状態でも他のユーザモデルとの一致を調べることによって素早く有意義な示唆を提示することを可能とする。

例を説明すると、ユーザ a がページ j を選ぶ確率 $P_{a,j}$ は、他のユーザ i の過去の行動を考慮して、

$$P_{a,j} \propto \sum_{i=1}^{n} w(a,i) v_{i,j} \tag{3}$$

$$w(a,i) = \sum_{k} v_{a,k} v_{i,k} \tag{4}$$

というように表すことができる。ここで $w(a,i)$ はアクティブユーザと他のユーザ i との類似度で、$v_{i,k}$

はユーザ i が今までにページ k 選択した頻度を表している。

協調フィルタリングは他のユーザの閲覧履歴を基にするため、ユーザの閲覧履歴がそれほど十分でなくとも他ユーザのモデルを利用することで素早く提示が行えることが利点である。しかし自分の嗜好とマッチしている他のユーザがいることが前提となる。

2.3 決定木を用いたクエリ提示

ここでは機械学習の一つである決定木アルゴリズム ID3[5] を用いてユーザの閲覧履歴からクエリを作成する手順を紹介する。[4]

1. まずユーザの閲覧履歴としてそのページが必要だったか、必要ではなかったかの 2 値化でラベル付けをおこなう。
2. 入力された必要ページと不要ページの集合を Set_0 とする。
3. 集合 Set_0 に”未分割”の印をつける
4. ”未分割”の印がついた集合のうち任意の集合 Set_i 中の必要ページ、不要ページ中の自立語 $t_j(1 < j < N)$ について、以下のような式によって相互情報量を計算する。（未分割集合がなければ終了）

$$I(t_j) = H - H(t_j) \tag{5}$$

ここで

$p_i = Set_i$ のなかの必要ページ数

$n_i = Set_i$ のなかの不要ページ数

$s_i = p_i + n_i$

$p_i(t_j) = Set_i$ で t_j を含む必要ページ数

$n_i(t_j) = Set_i$ で t_j を含む不要ページ数

$s_i(t_j) = p_i(t_j) + n_i(t_j)$

$p_i(\bar{t_j}) = Set_i$ で t_j を含まない必要ページ数

$n_i(\bar{t_j}) = Set_i$ で t_j を含まない不要ページ数

$s_i(\bar{t_j}) = p_i(\bar{t_j}) + n_i(\bar{t_j})$

$h(a,b,c) = -(\frac{a}{c}\log_2(\frac{a}{c}) + \frac{b}{c}\log_2(\frac{b}{c}))$

$$H = h(p_i, n_i, s_i) \tag{6}$$

$$H(t_j) = \frac{s_i(t_j)}{s_i} h(p_i(t_j), n_i(t_j), s_i(t_j)) + \frac{s_i(\bar{t_j})}{s_i} h(p_i(\bar{t_j}), n_i(\bar{t_j}), s_i(\bar{t_j})) \tag{7}$$

5. 自立語 $t_j(1 < j < N)$ から $I(t_k)$ を最大にする t_k を選ぶ。$I(t_k) > 0$ の場合、t_k を持つページからなる集合を Set_i とし、それぞれに”未分割”の印をつける。i'、i” はすでに集合 $Set_i^{'}$ $Set_i^{"}$ が存在しなければ任意の数でよい。$I(t_k) = 0$ の場合は分割しない。
6. 集合 Set_i から”未分割”の印を除き、４）に戻る

このようにして作成した決定木において、必要ページを得るパスで用いた単語を演算子 AND で結合して検索式を作成する。さらに各パスで得られた検索式を演算子 OR で結合したものを最終的な検索式とする。作成される検索式によって検索されるページは、AND によって結合された各単語が共起するページとなる。そのため ID3 によって得られる検索式は、必要ページに存在し不要ページには存在しない単語の共起を表していると考えることができる。

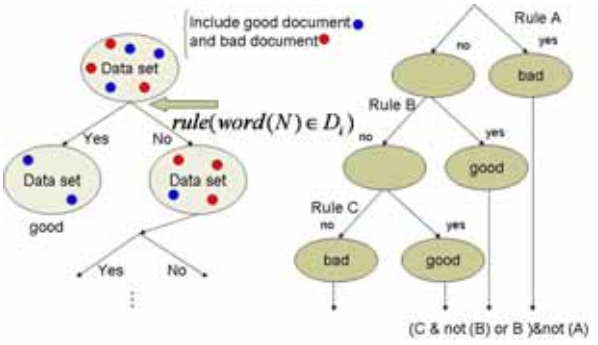

図 1: 決定木によるクエリ生成

過去のユーザの閲覧履歴を用いた手法は十分にユーザの嗜好をモデルで表現できればそれを利用してユーザの嗜好を汲み取った提示をしてくれる。しかしユーザの嗜好をモデルとして構築するためにはある程度以上のデータが必要あり、また過去に構築したモデルと関連のない検索をユーザの行う場合などモデルが常に有効であるとは限らない、などの問題点が挙げられる。

3 提案手法

3.1 既存の手法の問題点

既存の手法ではアクティブユーザの過去の履歴や他のユーザの履歴を基に、ユーザの要望を満たす検索支援を行っているが、ユーザが検索を行う時に今までとは全く違う領域に関する検索を行う場合や、ユーザの行動パターンが急激に変わる場合などでは過去の履歴が役に立つとは限らず、機械学習などの手法では一般に精度を高めるためには大量の過去の履歴データが必要となることも問題点として挙げられる。また、他のユーザとは違った嗜好で検索を行う場合や、ユーザが検索したい内容に関して、過去に他のユーザが検索した履歴がないような場合などでは、他のユーザの履歴を利用するのは難しい。

そこで本研究ではユーザのその検索利用内における閲覧履歴のみを用いて、ユーザが本来必要としているであろうページを見つけることを支援するシステムを提案する。また本研究では検索エンジン自体は既存のシステムを利用し、提案手法としては、既存の検索エンジンによって提示されたページに対してユーザの取る行動を取り込んで検索を支援するシステムについて述べる。

3.2 提案手法概要

まず提案手法の全体的な概要について述べる。提案するシステムはクライアント側で動くアプリケーションを考える。まず一般の検索エンジンと同じようにユーザがクエリを入力する。システムはユーザから受け取ったクエリを既存の検索エンジンに投げ、その検索結果リストを従来の検索と同じようにユーザに提示する。ユーザには Web ページのタイトルとページの簡単な要約が載ったリストが表示され、ユーザはそのリストの情報を元に見たいページを選んで閲覧する。この時の閲覧履歴は随時システム内に貯蔵される。ユーザが閲覧を進めているのと同時にシステム側では検索結果のリストにある Web ページにアクセスし解析する。ページの解析によりキーワード候補とユーザの履歴とよくマッチするキーワードを提示する。したがってこの時に提示するキーワードは、最初のクエリによる検索結果のページの中から、検索結果のページ集合に出現する単語を用いてユーザの要望に一致するページを絞り込むことを考えている。

閲覧したページに対しては、閲覧していたページからリスト表示に戻る際に二つの戻るためのボタンがあって、そのページが必要な情報があった場合となかった場合とで使い分けることによってユーザの閲覧履歴をシステムに伝える。。ユーザの趣向を取り入れるごとに、ブラウジングページの横に最初に入力したクエリと AND で結合するのに適切なクエリの候補が提示される。その際クエリはリストで提示され、各候補クエリの下にはそのクエリによって提示されるページのタイトルや簡単な要約が 3 ページ分提示される。また似たような結果になる候補クエリはまとめて提示され、ユーザが候補クエリのとなりのボタンをクリックするとその候補クエリと近いクエリが表示される。ユーザがクエリをクリックすると、最初に入力したクエリとそのクエリとの AND 検索の結果から今まで見たページを省いたページリストが表示される。

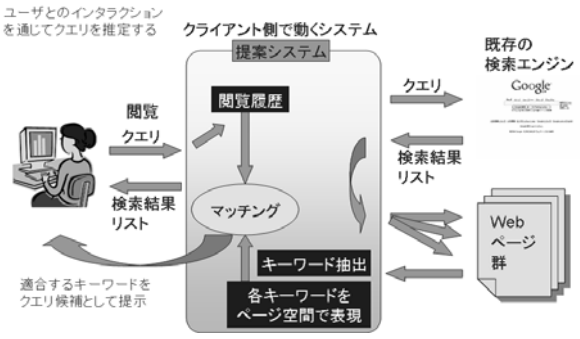

図 2: 提案システム概要

3.3 インタフェース

3.3.1 ユーザからの入力方法

本研究ではその検索利用内における閲覧履歴を用いることを考える。既存の検索エンジンによる検索結果のリストをユーザが上位から閲覧を進めるごとに、各ページが必要であったか、必要でなかったか、の 2 値情報をシステムに取り込むことを考える。

実装に関してのインタフェースのアイデアを述べると、ユーザにとって不必要であるページは閲覧をすることはないだろうから、ユーザが見なかったページは不必要であったとして取り込むことができる。しかし、ユーザが見たページに関しては、検索結果リストに表示されるページの要約だけではユーザはそのページが必要かわからず、実際に見てみないとわからなかったページであり、見た結果、必要だったか必要でなかったかをユーザに入力してもらう必要

がある。ユーザの入力の手間をなるべく小さくする必要性から、ユーザがページからもとの検索結果リストに戻る時に、多くのユーザが戻るボタンを利用していることから、戻るボタンを二つ用意して、見た結果必要だったと投票して検索結果リストに戻るボタンと、見た結果不必要だったと投票して検索結果リストに戻るボタンを用意することで、ユーザの入力の手間をできるだけ軽減することを考えることなどが考えられる。

3.3.2 ユーザへの提示方法

本研究は、ユーザの必要とするページをより少ない手間で提示することを目指したものであるが、ユーザに必要であると思われるページを直接示すと、そのページが本当にユーザが必要としているものか、すぐには判断できないことも考えられ、また仮に正しいページを提示できても、ユーザの要望がどういうものであったかは、明示的ではないため、次の検索を行う機会があった時に、その時に提示されたユーザの要望に関する知識の使いまわしができないことなどが考えられる。

一般に検索エンジンではユーザが欲しいページに関連が深いと思われるキーワードを、クエリとして入力するのであるが、そのクエリが含まれているページをリンク構造などから重要度を計算して順序付けて提示するシステムになっている。

そこで本研究では、ユーザが必要とするであろう順にページを並び替えるのと平行して、ユーザの必要とするページのみに出現するようなキーワードを、クエリ候補としてユーザに提示することを考える。ユーザの要望をクエリとして提示する場合、間接的な表現になってしまって、ユーザが必要とするページを直接提示するよりも、ユーザの要望を正確に表現する力は落ちてしまう恐れがあるが、ユーザがページ閲覧をまだあまり進めていない時期では、どちらの手法でも正確な推定を行うことが難しい。そこで推定の候補をキーワードの形で提示することでユーザの気づきを促すことで、より正確な提示が行われることが期待される。またキーワードの形でユーザの要望を表すことで、より検索を絞り込みたい時や、他の要素を含めて検索したい場合などにおいて、提示されたキーワードを組み合わせることで、提示された知識を再利用することが可能となる。

3.4 キーワードのスコア付け

上記のユーザによるページ評価を基に、ユーザが本来必要としていると考えられるクエリを、最初に検索結果として提示されたページに出現する単語の中から選ぶことを考える。クエリ候補となる単語を選ぶ際に、ユーザがページ閲覧を進める度に単語にクエリ候補としての尤もらしさとなるスコアを与えて、スコアの高い順にユーザに提示することを考える。

本研究でスコア付けのポリシーとして、ユーザが見たページの生起分布と、各キーワードの各ページでの生起分布との一致において、どれほどの確率で一致したものかを考えた。

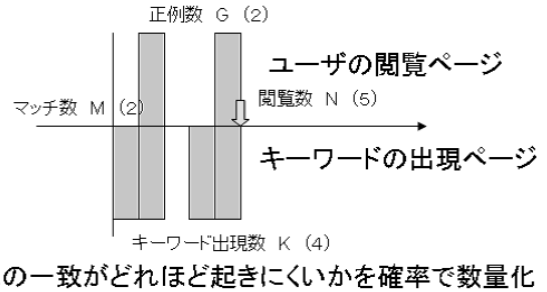

図 3: スコア付け

具体的なスコア付けについて説明を行う。図 3 において、ユーザが閲覧を進めたページ数は N, そのうちでユーザが必要だと判断したページ数が G, スコア付けが行われるキーワードが出現したページ数が K, ユーザが必要であると判断したページとキーワード出現が共起したページ数が M, である場合、クエリ候補の尤度として、ランダムにこれらの分布が与えられた場合に、ユーザが必要であると判断したページと、キーワードの出現の一致が M 以上になる確率をとった。(式 8)

$$P_k = \frac{\sum_{m=M}^{S} {}_L C_m * {}_{N-L} C_{L-m}}{\sum_{m=0}^{S} {}_L C_m * {}_{N-L} C_{L-m}} \qquad (8)$$

ここで

S は K か G の小さい方の値

L は K か G の大きい方の値

また実際には、必要ページを一つの単語でカバーできるとは限らないので、不必要ページには出現せず、部分的に必要ページをカバーする単語も評価ができるように、G>K の時には G=K とした。

この方式でいくと、序盤においては必要ページの数が少なく、偶然必要ページの分布にマッチする単

語も多いと考えられ、複数の単語に同じスコアが与えられてしまうことを避けるために、ページに出現した単語の頻度に注目して、分布が同じ単語間での順序付けを行う。

あるページにおいてよく出現する単語はそのページをよく表すと考え、同じページ分布でも、各ページにおいてより多くの頻度で出現する単語のスコアを高く評価することにする。具体的には、各ページにおいて出現する全ての単語の出現数の和における対象となる単語の頻度の割合、の和の順序によって、同じページ分布における単語間の順序付けを行う。

4 実験

4.1 実験方法

実装は言語は C++, 形態素解析は Mecab[1]を、辞書は IPAdic[2]を用いた。クエリ候補となる単語であるが、今回は名詞語のみを抽出した。タグについては一切利用していない。ページにアクセスする時は、トップページだけでは情報が少ない場合があるため、トップページからリンクの張られていてトップと同じドメインに含まれるページもアクセスして解析した。今回はインターフェースの部分は間に合わなかったため、ユーザのページ選択によりどれほど絞り込める可能性があるかを実験、検証することを考えた。

実験として、あらかじめ著者がクエリを選び、そのクエリによる検索結果の中で、何かしらのテーマに沿った内容のページだけを選んで必要なページ、それ以外のページを不必要ページとした模擬データを作成することで実験を行った。

実験で検証する必要のある内容は以下の 2 つである。

1. ユーザの選んだページ群をよく表現する単語が存在するのか
2. 初期の少ない閲覧履歴でユーザの要望を絞り込めるか

これらのことについて検証するために、「バレー」というクエリによる検索結果 100 件から、バレーボールに関連する 14 ページを選び、「オンライン」というクエリによる検索結果 100 件から、ゲームに関連する 25 ページを選び、検索結果ページ全体に出現する単語のうち、それらのページとよく共起する単語を抽出し、どれほど必要としたページとマッチしているのかを調べた。

全体から単語を抽出すると数が大きくなりすぎてしまうため、全体で 2 ページ以下にしか出現しない単語は抽出しなかった。その結果「バレー」による検索結果ページ全体では 1389 語、「オンライン」による検索結果ページ全体では 4406 語の単語をクエリ候補として抽出した。

実験ではページ閲覧を進めるごとに、そこでのページ情報を用いて提示された上位 5 位のキーワードで評価を行った。上位 5 位のキーワードの OR 検索で引っかかるページにおいて、必要ページ全体のうちどれほどの割合の必要ページを拾うことができるかを示す再現率と、拾ってきたページのうち、必要であるページである割合を示す適合率、参考として、これら二つの指標を総合的にみて評価する指標である F-measure で評価を行った。

4.2 実験結果

実験結果を示す。閲覧ページ数と、提示された上位 5 位までのクエリによる再現率と適合率との関係を図 4,5 に、閲覧ページ数と、提示されたキーワードの関係を表 1,2 に示す。(() 内は分母が出現数、分子がその内必要ページでの出現数)

結果をみると「バレー」、「オンライン」の双方とも、ページ閲覧を進めるごとに提示されるキーワードの精度が上がっていることがわかる。

「バレー」では再現率が 100%となったが、オンラインの例では最後まで 100%には到達しなかった。これは、「判断」という、16 出現中 9 個の必要ページでしか出現しない単語がないと 100%に到達しないためであり、適合率を犠牲にしないと完全に再現率を上げるのは難しく、単語がページクエリの最小単位としては必ずしも適当ではないという問題に直面していると考えられる。また再現率と適合率とがトレードオフとなっている関係も確認できる。「バレー」の方が「オンライン」よりも最終的な結果も良く、収束するのが早いのは、問題特有の要素もあるだろうが、出現する単語数の違いではないかと考えられる。

5 今後の課題

「実験の強化」

今回は時間が足りなく、十分な実験が行うことができなかった。より多くの実験を行うことで提案手

[1] http://mecab.sourceforge.jp/
[2] http://chasen.naist.jp/stable/ipadic/:単語数 23700 語

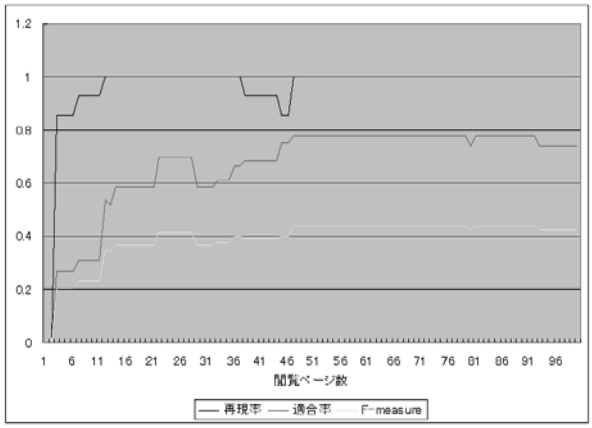

図 4: 上位 5 キーワードによる適合率と再現率 (バレー)

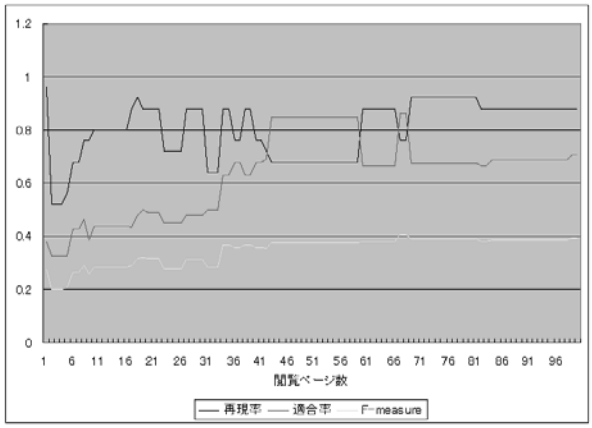

図 5: 上位 5 キーワードによる適合率と再現率 (オンライン)

法の有効性を検証する必要がある。また、取り組む問題を理解しやすい形にモデル化することで、実験、検証を行いやすい環境を整えることも優先事項として考えられる。

「クエリ表現力の強化」

クエリをキーとして必要ページを持ってくることを考えると、単純に AND と OR による単語の結合では表現できるページ集合には限界がある。そこで、NOT や入れ子上にクエリを構造化することでクエリの表現力を増やす工夫が考えられる。また、単語数の増加が問題の複雑化につながっていると考えると、対策として予めフィルタリングを行って単語数を絞り込むことなどが考えられる。

「ページの閲覧順序の入れ替え」

提示されるキーワードの精度は閲覧するページの順序に依存する。また、そもそもの目的がユーザの必要なページを小さい手間で持ってくることなので、ユーザに提示するページの表示順序を工夫することが考えられる。今回は述べなかったが、ユーザが必要とするページを選んだときに、必要ページに含まれている単語を多くの割合で含むページを上位にもってくるようにページ順序を変える実験を行い、以前よりもユーザの必要ページを早く持ってくることに成功したが、逆に提示されるクエリの精度は低下してしまった。これについては現在原因を考察中であるが、ユーザの必要とするページと、クエリの選り分けに必要なページとは違ったものとして対応することを考慮する必要がある。

「情報の取り込みの強化」

今回はユーザの閲覧履歴として検索結果のページが必要であったか、必要でなかったか、の 2 値を取り込んだが、もっとユーザの要望を表している情報を取り込むことが課題として考えられる。例えば、ユーザが同じ不必要と判断したページでも、検索結果リストにある要約を読んで不必要と判断した場合と、要約だけ読んでも内容についてはわからず、実際にページを見てから不必要と判断したページとでは、検索結果リストにある要約情報に含まれる意味合いが異なってくると考えられ、こうした情報を利用してさらに詳しい分析を行うことが考えられる。

6 おわりに

本発表では Web 検索エンジンについての現状について説明し、ユーザの過去の閲覧履歴から検索支援を行う既存の手法について説明を行い、それらの問題点として、それらの手法が過去の履歴がある程度以上あることを想定しているため、自分が今まで検索したことのないような新しいテーマについて検索を行うときや、他人もあまり検索しないような内容や、他人とは違った切り口で検索を行う場合などでは、うまく検索できないことを挙げた。そこで提案手法として、既存の検索エンジンを用いて、ユーザがその時の閲覧履歴のみを用いて検索支援を行う手法を提案した。今回実験を行った模擬データは非常に単純ではあったものの、閲覧を進めるごとに有効なクエリを提示できることがわかった。今後の課題として

はより厳密な実験を行い、そのためにモデルを構築することや、クエリの構成方法、ユーザに見せるページの順序の並び替え、ユーザからより多くの情報を取り込むこと、などが挙げられる。

参考文献

[1] Gary.W.Flake, Eric.J.Glover, Steve Lawrence, "Extracting Query Modifications from Nonlinear SVMs", *Proceedings of the Eleventh International World Wide Web Conference*May,(2002)

[2] Lawrence Page, Sergey Brin, Rajeev Motwani, Terry Winograd, "The PageRank Citation Ranking: Bringing Order to the Web", *Stanford Digital Libraries Working Paper,* (1998)

[3] Hiroyuki Kawahara, Toshiharu Hasegawa, "Mondou: Interface with Text Data Mining for Web Search Engine", *IEEE Thirty-First Annual Hawaii International Conference on System Sciences,* Vol5, pp.275(1998)

[4] 中島浩之, 木谷強, 岡田守, "検索語間における共起関係の特定によるレレバンスフィードバックの高精度化", 情報処理学会論文誌,Vol.40,No.3,pp.1236-1244(1999)

[5] Quinlan, J.R, "C4.5", *Programs for machine learning, Morgan Kaufman,*(1993)

[6] GrokkerSearchEngine, http://www.grokker.com/ (2001)

[7] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filetring of netnews. *Proceeding of 14th Uncertainty in Artificial Intelligence*, pp.43-52, 1998.

[8] G.Salton, C.Yang, "On the Specification of Term Values in Automatic Indexing", *Journal of Documentation*29(4), December, pp.351-372(1973)

[9] Chien.L.F, "PAT-Tree-Based Adaptive Keyphrase Extraction for Intelligent Chinese Information Retrieval", *In proceedings of the 20th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*SIGIR'97, Phliadelphia, pp.50-58(1997)

[10] Hua-jun.Zeng, Qi-Cai.He, Zheng.Chen, Wei-Ying.Ma, Jinwen.Ma, "Learning to Cluster Web Search Results",

[11] Hasutie.T, Tibshirani.R, Friedman.J, "The Elements of Statistical Learning", *New York: Springer-Verlag,*(2001)

[12] Smola.A.J, Schlkopf.B.A, "Tutorial on Support Vector Regression", *NeuroCOLT2 Technical Report Series,*NC2-TR-1998-030.October(1998)

表 1: 提示された上位 5 位単語 (バレー)

ページ数	1 位	2 位	3 位	4 位	5 位
5	月 (10/38)	世界 (6/22)	検索 (3/18)	全日本 (8/10)	選手 (7/11)
10	月 (10/38)	大会 (10/17)	バレーボール (12/14)	選手 (7/11)	チーム (9/14)
20	バレーボール (12/14)	選手 (7/11)	女子 (11/13)	チーム (9/14)	男子 (7/10)
30	バレーボール (12/14)	女子 (11/13)	高校 (5/11)	選手 (7/11)	男子 (7/10)
40	女子 (11/13)	試合 (10/13)	リーグ (6/7)	出場 (6/9)	男子 (7/10)
50	予選 (6/7)	バレーボール (12/14)	女子 (11/13)	試合 (10/13)	選手権 (6/7)
60	予選 (6/7)	バレーボール (12/14)	女子 (11/13)	試合 (10/13)	選手権 (6/7)
70	バレーボール (12/14)	女子 (11/13)	試合 (10/13)	選手権 (6/7)	リーグ (6/7)
80	バレーボール (12/14)	女子 (11/13)	試合 (10/13)	全日本 (8/10)	選手権 (6/7)
90	バレーボール (12/14)	女子 (11/13)	試合 (10/13)	大山 (6/6)	恵 (5/5)
100	バレーボール (12/14)	女子 (11/13)	大山 (6/6)	試合 (10/13)	全日本 (8/10)

表 2: 提示された上位 5 位単語 (オンライン)

ページ数	1 位	2 位	3 位	4 位	5 位
5	今後 (6/16)	ソフト (8/17)	デザイン (4/19)	稼動 (2/3)	後 (13/34)
10	メンテナンス (10/19)	アイテム (12/17)	定期 (9/23)	ユーザ (9/19)	株式会社 (6/21)
20	ユーザー (9/19)	アイテム (12/17)	定期 (10/13)	条件 (10/22)	ゲーム (22/31)
30	ユーザー (9/19)	プレイ (12/12)	インストール (22/31)	遊び方 (7/7)	スロット (7/7)
40	プレイ (12/12)	インストール (10/13)	クエスト (6/6)	マウス (6/8)	アップデート (6/12)
50	プレイ (12/12)	インストール (10/13)	プレイヤー (10/10)	クエスト (6/6)	答え (7/7)
60	プレイ (12/12)	インストール (10/13)	プレイヤー (10/10)	ゲーム (22/31)	クエスト (6/6)
70	プレイ (12/12)	インストール (10/13)	プレイヤー (10/10)	正常 (8/9)	ゲーム (22/31)
80	プレイ (12/12)	インストール (10/13)	プレイヤー (10/10)	正常 (8/9)	ゲーム (22/31)
90	プレイ (12/12)	プレイヤー (10/10)	ゲーム (22/31)	戦闘 (6/7)	クエスト (6/6)
100	プレイ (12/12)	プレイヤー (10/10)	ゲーム (22/31)	遊び方 (7/7)	スロット (7/7)