ユーザの閲覧履歴を用いたキーワード抽出による オンライン検索支援

永井　洋一† 　近山　隆†

† 東京大学新領域創成科学研究科　〒 277–8561 柏市柏の葉 5 丁目 1 番 5 号
E-mail: †{nagai,chikayama}@logos.ic.i.u-tokyo.ac.jp

あらまし　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は数多くなされているが, 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, ユーザのその検索行動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確率を想定したスコア付けを行い提示し, それと同時にユーザに必要であると思われるページをユーザに提示するシステムを考案し実験を行った.

キーワード　情報検索, 検索支援, キーワード抽出

The Online Search Support Using Keywords Appeared in Browsed Documents

Yoichi NAGAI† and Takashi CHIKAYAMA†

† Faculty of Frontier Sciences　Kasiwanoha 5–1–5, Kasiwa-si, Chiba, 277–8561 Japan
E-mail: †{nagai,chikayama}@logos.ic.i.u-tokyo.ac.jp

Abstract　In these years, information technology has developed rapidly and there are huge digital data reserved in computer or database. So, a lot of people use Web search engine to seek Information. At the same time, we have to search applicable data from such flood of information. Therefore to search data we want, various methods are developed. In this paper, We introduce the existing method of Web searching first. Many existing search methods use machine learning. But enerically machine learning demands many train data set. It contravene the purpose to reduce Web users burden. So we propose a system that suggests query user really needs without a lot of users browsing record.

Key words　information search, retrieval support, keyword extraction

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

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, 確率的な解析を行って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やキーワード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 章では文書検索における検索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

ユーザの検索を支援するための研究として, 文書集合からその内容をよく表すキーワードを抽出する古典的な手法としては, 単語の各文書における出現頻度 (Term Freqency) と, その単語の出現する文書数 (Document Freqency) を用いた TF-IDF がある. 文書を出現する各単語で表現する場合各単語がそれぞれベクトル空間で特徴を表す次元軸となるが, その量としては各単語のそのテキストに出現した頻度をとることが一般的である.

$$\text{単語頻度 } tf_{ij} = \text{文書 } D_i \text{における語 } T_j \text{の出現数} \tag{1}$$

しかしより良く特徴を表現するように, それに加えて大域的に単語に重みをつける方法として文書頻度の逆数をとる方法がある. いま, ある N 個の文書からなる文書集合を考えると, ある語 T_j を含む文書頻度 df_j(document frequency) とはその語が出現する文書数として定義される.

$$\text{文書頻度 } df_j = \text{語 } T_j \text{を含む文書数} \tag{2}$$

この頻度が小さければ, その語が検索質問に用いられた場合に該当文書を小さな集合に絞り込むことができるので索引語としての望ましさの指標としてはこの文書頻度の逆数を用いればよい. この値とその単語の文書に出現する頻度をかけて単語の重みは

$$tf_{ji} \log \frac{N}{df_j} \tag{3}$$

として定義される. この手法は $tf-idf$(term frequency-inverse document frequency) モデルとよばれ基本的な重み付けの手法として広く用いられている.

ユーザが閲覧した文書について, ユーザが必要とした文書だったか, 必要としなかった文書だったか, といった情報を用いてユーザに必要な文書を引き出すためのキーワードを評価する手法として F-measure がある. F-measure は以下の式によって表される.

$$F\text{-}measure = \frac{(1+\beta^2) \times \text{再現率} \times \text{適合率}}{(\beta^2 \times \text{再現率}) + \text{適合率}} \tag{4}$$

$$\text{再現率} = \frac{\text{その単語が出現した必要ページ数}}{\text{全文書中の必要ページ数}} \tag{5}$$

$$\text{適合率} = \frac{\text{その単語が出現した必要ページ数}}{\text{その単語が出現した全ページ数}} \tag{6}$$

β は再現率に対する適合率の相対的な重要性である.

ユーザの検索支援として, ユーザの閲覧履歴に基づいてキーワードを提示するのではなく, ユーザにとって必要だと思われる文書を提示する古典的な手法として関連フィードバックがある. この手法ではユーザにとって必要であると思われるクエリを単語ベクトルの形で表現する. ユーザが閲覧を進める度に, 必要だった文書に出現した単語の重みを増やし, 不要であった文書に出現した単語の重みを減らす, といった操作を繰り返すことによって, 次第にユーザの要望をよく表した単語ベクトルクエリを生成する. 具体的には下の式によって単語ベクトルを更新する.

$$Q' = Q + \frac{1}{|R|} \sum_{D_i \in R} D_i - \frac{1}{|N|} \sum_{D_i \in N} D_i \tag{7}$$

ここで Q はユーザの要望を表した単語ベクトル.Q' は更新後の Q. D はユーザが閲覧した文書の単語ベクトル. R はユーザが必要であると判定した文書集合.N はユーザが不要と判定した文書集合である.

この他にもユーザの検索支援の手法として, 他のユーザの閲覧履歴を基にユーザの検索を支援する協調フィルタリングの手法などが実用化されている.

3. 提 案 手 法

3.1 既存の手法の問題点

既存の研究ではまとまった分量の文書やユーザの閲覧履歴からユーザの要望に沿ったキーワードや文書を提示することが多く, ユーザの閲覧履歴による手掛かりが少ない段階からユーザの検索を支援する目的に沿ったものではない. 他のユーザの過去の履歴を用いて少ないユーザの要望に関する手掛かりを補う協調フィルタリングが実用に用いられているが, それが有効に働くためにはユーザと同じ嗜好を持った他ユーザが存在しなければならない.

またユーザに必要であると考えられる文書を提示する関連フィードバックなどでは文書を単語によるベクトル空間で表したモデルを用いるが, 一般には文書には大量の単語が存在するため, 特に十分な分量の手掛かりのない状態では関係のない単語の存在などによる計算コストや時間や精度に問題が出ると考えられる.

そのため本研究ではその時のユーザの閲覧履歴を用いて, 少ない情報であってもなるべく少ない手間でユーザにとって適切な文書やキーワードを提示することを目的とした検索支援の手法を提案する.

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. 本研究ではユーザの検索支援として, ユーザの必要であると考えられる文書の提示, 及びユーザにとってより適切であると考えられるクエリの提示, の 2 つを同時にお互いを補完しながら提示する手法を提案する.

提案手法の環境として,Web 検索エンジンなどのようにユーザがクエリを入力して, その検索結果の文書のリストがユーザに提示され, ユーザはそのリストに沿って文書を閲覧する状況を想定する.

提案手法ではユーザが最初に提示された検索結果のリストに沿って文書を閲覧してゆき, 閲覧したページがそのユーザにとって必要であったかどうかをシステムに入力してもらう (閲覧しなかった文書は不必要であったと判断する). その閲覧情報を受けてシステムではユーザが閲覧を進めるのと並行して, 検索結果の文書のリストをユーザがまだ閲覧していない文書に対して, ユーザにとって必要であると思われる順序に並び替える. またユーザの要望に沿っていると考えられるキーワードもクエリ候補として同時に提示する.

3.3 キーワード提示手法

まずユーザにキーワードを提示する手法についての説明を行う. クエリ候補となるキーワードは, ユーザが最初に提示された検索結果のリストにある文書に含まれる単語に対してそれぞれスコアを付けその上位から順にユーザに提示をおこなう. 検索結果のリストについては本研究では上位 100 文書を用いている. クエリ候補となる単語について, リストにある文書に含まれる単語全てを用いると多くなりすぎるため, 出現文書数が極端に少ないか多いものや, 品詞によってあらかじめ排除することを考える.

単語に付与されるスコアとしては, 提案手法では各単語が将来どのくらい必要文書を持ってこられるかを測ることを考える. どれだけ必要文書をもってこられるかの基準としては F-measure など従来の基準をを用いることを考える. この基準を用いて各単語の将来のスコアを求めるためにベイズ推定を行う. ベイズ推定とは, ある証拠に基づいてその原因となった事象を推定するための確率論的方法である.

いま,A および X を離散確率変数とする. ここで A を原因,X をそれに対する証拠（つまり原因によって起きたと想定される事象）とするとき,

$P(A)$ = 事象 A が発生する確率を, 事前確率（prior probability）, $P(A|X)$ = 事象 X が発生した下で, 事象 A が発生する条件付き確率を, 事後確率（posterior probability）, という. $P(A|X)$ は, ベイズの定理によって以下の式によって与えられる.

$$P(A|X) = \frac{P(X|A)P(A)}{P(X)} \tag{8}$$

分母の $P(X)$ はすべての想定される原因事象 B から

$$P(X) = \sum_B P(X|B)P(B) \tag{9}$$

と求められるため, 結局 $P(A|X)$ は以下の式で求められる.

$$P(A|X) = \frac{P(X|A)P(A)}{\sum_B P(X|B)P(B)} \tag{10}$$

ユーザが閲覧を進めたページ数を N, リストの全文書中, ユーザにとって必要である文書に単語が出現する数が G, リストの前文書中, 単語が出現する文書数が K, 現在ユーザが閲覧した N 文書中, スコア付けが行われる単語が出現する文書数が k, 現在ユーザが閲覧した N 文書中, 単語がユーザが必要であると判断した文書に出現した回数が g, とする. リスト全文書 N 中,K 回出現する単語がそのうち最終的に何回必要ページに出現するかをベイズ推定を用いて推定することを考える. $P(G|g,k)$ を全ての想定される $G, (g \leqq G \leqq K)$ について求めて G の期待値を算出する. ここで $P(G)$,$P(g,k|G)$ は次式で与えられる.

$$P(G) = {}_K C_G (P_G)^G (1-P_G)^{K-G} \tag{11}$$

$$P(g,k|G) = {}_K C_G \frac{{}_G P_g \times {}_{K-G} P_{k-g}}{{}_K P_k} \tag{12}$$

P_G はリスト全文書における必要文書の割合である. 一般にこの部分はわからないため適当な数値でも構わない. 逆に他のユーザの履歴などからあらかじめクエリとして必要である見込みの高い単語の確率を高めに設定することも可能で, ヒューリスティックな知識を事前分布の形で埋め込むことが可能である.

これらを式 10 に代入して単語の G の期待値を求める.

3.4 文書の並び替え

ベイズ推定によって推定した単語の最終的な必要文書出現数の期待値を用いて, ユーザが未閲覧である各文書に対してユーザにとって必要である確率を推定する. ここでもベイズ推定を用いて推定を行う. 文書 d に出現した単語集合を T_d とすると, 単語集合における各単語の G の期待値がわかっている時にその文書 d が必要である確率 P(必要 $|T_d)$ を求めることになる.

$$P(\text{必要}) = P_G, \qquad P(\text{不要}) = 1 - P_G \tag{13}$$

$$P(T_d|\text{ 必要}) = \frac{1}{|T_d|} \sum_{T_d} \frac{1}{1 + \frac{P_G(K_{t_d} - G_{t_d})}{(1-P_G)G_{t_d}}} \tag{14}$$

$$P(T_d|\text{ 不要}) = \frac{1}{|T_d|} \sum_{T_d} \left(1 - \frac{1}{1 + \frac{P_G(K_{t_d} - G_{t_d})}{(1-P_G)G_{t_d}}}\right) \tag{15}$$

これらを式 10 に代入して P(必要 $|T_d)$ を求める.

4. 実　　験

実験として, 今回は Web 検索におけるユーザの検索支援を行った. 最初にユーザが入力する検索エンジンに Google(注1)を, 文書の解析の実装は言語は C++, 形態素解析は Mecab(注2)を, 辞書は IPAdic(注3)を用いた. クエリ候補となる単語であるが, 今回は日本語の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.

(注1)：http://google.co.jp/
(注2)：http://mecab.sourceforge.jp/
(注3)：http://chasen.naist.jp/stable/ipadic/:単語数 23700 語

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.

4.1 実験方法

今回は実験に際して 4 つのテーマで実験を行った. 「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 28 ページを選んだテーマ, 「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 26 ページを選んだテーマ, 「阪神」というクエリによる検索結果 100 件から, 阪神タイガースに関連する 22 ページを選んだテーマ, 「Wii PS3」というクエリによる検索結果 100 件から, 両者の製品に対してどちらを買った方がよいかについて示唆を与える 27 ページを選んだテーマ, についてそれぞれ実験を行った.

それらを最初から閲覧を進めていって, 適切にキーワードを選択しできたか, 適切なページをユーザに提示できたかを調べて評価を行った.

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 3 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「阪神」による検索結果ページ全体では 2037 語, 「バレー」による検索結果ページ全体では 3073 語, 「オンライン」による検索結果ページ全体では 4406 語, 「Wii PS3」による検索結果ページ全体では 1793 語の単語を, それぞれクエリ候補として抽出した. 「Wii PS3」に関しては人の意見という曖昧な基準で分類が可能であるかを調べることを目的とし, 人の意見がたくさんありそうな Web コンテンツとしてブログを取り上げ, このテーマに関しては Yahoo!ブログ検索による検索結果リストを用いた.

実験の評価項目としては, 提示されたキーワードが各閲覧段階でどれほど有意義であるか, と, ユーザにとって必要なページを並び替えることでどれだけ早い段階でたくさん提示できたか, の 2 つを考える必要がある.

キーワードの評価方法として, ページの閲覧を進めるごとに提示された上位 10 位までのキーワードの出現するページ全てが必要ページ全体のどれくらいの割合をカバーしているかを示す再現率と, 上位 10 位までのキーワードのどれかが出現したページ集合のうち, 必要ページがどれだけの割合で含まれているかを示す適合率, を考え, 実際の評価にはそれらを総合的に評価するための指標である F-measure 値を用いた (式 4). 比較対象として, ベイズ推定を行わずにその場の情報だけを用いて算出した F-measure 値を用い, 各時点におけるそれぞれの全体での F-measure 値を比較した.

ユーザにとって必要であるページをどれだけ早く提示できたかについては, 横軸にユーザの閲覧ページ数, 縦軸に提示した必要ページ数を取ったグラフを用いて, 横軸と縦軸の最大値の積を取った面積におけるグラフに囲まれた面積の割合を評価基準とした (大きいほど良い). 本稿では以後この基準を”並び替え効率”と記述する. 比較対象としては関連研究で紹介した, 単語ベクトルとして tf-idf を用いた関連フィードバック法を用いる.

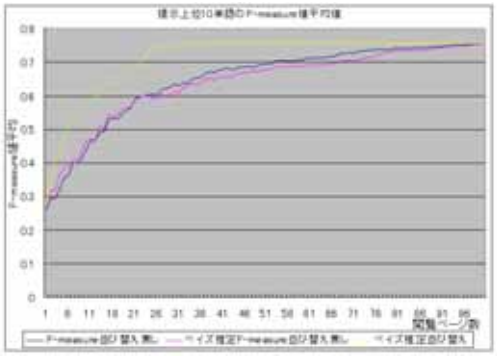

図 1　各閲覧時点におけるクエリ推定の精度 (阪神)

4.2 実験結果

4.2.1 阪神 (タイガース)

阪神と入力してタイガースのページを集めるタスクについての実験結果を示す. 阪神と入力した検索結果のうち, タイガース以外のものについてのページは, 阪神百貨店, 阪神高速, など, 阪神を地名として用いていたものであった.

最終的な F-measure 上位 10 の単語は表 1 のようなものであった.

上位 10 の単語の F-measure 値は全て 7 以上で, 直感的にもそれらがタイガースを示す単語であることがわかりやすい.

各閲覧段階での提示された上位 10 までのクエリの平均 F-measure 値を図 1 に示す.

初期の段階で普通の F-measure よりも高い精度でクエリの提示が行えている. またページの並び替えによってより早い段階で精度の高いクエリ提示を行えることがわかる. またページを並び替えることによってユーザに必要なページが早い段階で届き, その情報がキーワードの選別にたいして効果を発揮することで提示されるキーワードの質も良くなっていることがわかる.

次にページの並び替えの比較を図 2 に示す.

提案手法は早い段階から多くのユーザにとって必要ページを提示できていることがわかる. 各並び替え効率は, ソート無しが 0.606, 関連フィードバックが 0.74, ベイズ推定が 0.874 となっている.

4.2.2 バレー (ボール)

最終的な F-measure 上位 10 の単語は表 2 のようなものであった.

表 1　上位 10 位単語 (阪神)

順位	単語名	F-measure 値
1	ドラフト	0.807
2	赤星	0.786
3	戦	0.778
4	投手	0.758
5	藤川	0.758
6	金本	0.741
7	指名	0.741
8	鳥谷	0.741
9	岡田	0.714
10	巨人	0.714

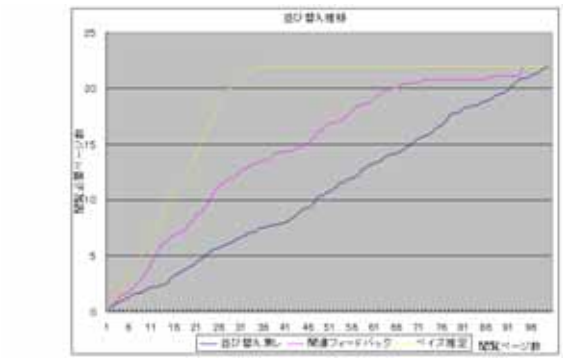

図 2　並び替えによる閲覧必要ページ数推移 (阪神)

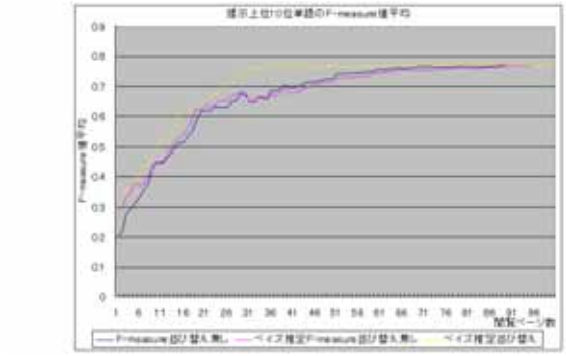

図 3　各閲覧時点におけるクエリ推定の精度 (バレー)

こちらの単語も F-measure の値が総じて高く, 直感的にもわかりやすい.

各閲覧段階での提示された上位 10 までのクエリの平均 F-measure 値を図 3 に示す.

「阪神」と同じように初期の段階で普通の F-measure よりも高い精度でクエリの提示が行えている. しかし元々簡単であるためか, どのやり方でも早い段階で高い精度に到達しているためそれほどは差がついていない.

ページの並び替えの比較を図 4 に示す.

各並び替え効率は, ソート無しが 0.557, 関連フィードバックが 0.704, ベイズ推定が 0.765 となっている.

4.2.3　オンライン (ゲーム)

最終的な F-measure 上位 10 の単語は表 3 のようなものであった.

表 2　上位 10 位単語 (バレー)

順位	単語名	F-measure 値
1	バレーボール	0.88
2	試合	0.833
3	女子	0.817
4	リーグ	0.789
5	全日本	0.781
6	男子	0.774
7	戦	0.739
8	優勝	0.729
9	選手	0.7
10	オリンピック	0.688

前回の 2 つに比べて F-measure 値がやや低めになっている. それだけ前回に比べるとオンラインというテーマからゲームだけを抜き取ることが簡単ではないということであろう. それでも上位の単語はゲームとの関連を十分連想させるものであると思われる.

各閲覧段階での提示されたクエリの精度を図 5 に示す. F-measure が元々低いため, クエリ提示では高精度の到達にあまり差がつきにくくなっている.

ページの並び替えの比較を図 6 に示す.

ページの並び替えの比較を図 6 に示す. 各並び替え効率は, ソート無しが 0.593, 関連フィードバックが 0.658, ベイズ推定が 0.801 となっている. 関連フィードバックの成績が良くないが, このテーマは単語数が多いため, 単語ベクトル空間のスパースネスの影響を受けたのではないかと考えられる.

4.2.4　Wii PS3 (購入参考)

人の意見という曖昧なものを扱うため, このテーマに限って名詞以外の品詞も利用した.

最終的な F-measure 上位 10 の単語は表 4 のようなものであった.

F-measure の値が 4〜5 台であることから, 比較検討のためになるページだけをきれいに抽出する単語はなかなか存在しないことがわかる. 直感的にはわかりにくいが,www などは比較を詳しく書く人は外部にあるソースを参照にするためブログの記事内にリンクを張る割合が多いからであると考えられる. 「感」については, してやられた感, 爽快感など, ゲーム個別の感想に結びついていた.

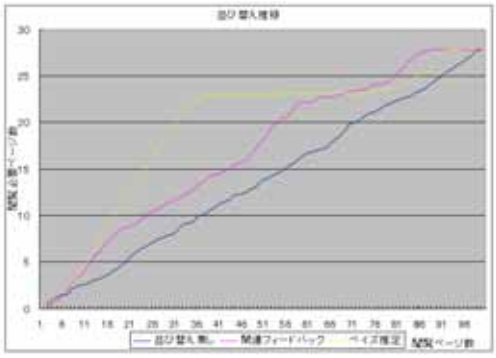

図 4　並び替えによる閲覧必要ページ数推移 (バレー)

表 3　上位 10 位単語 (オンライン)

順位	単語名	F-measure 値
1	アップデート	0.644
2	プレイ	0.641
3	ラグナロク	0.614
4	アイテム	0.598
5	戦闘	0.584
6	クエスト	0.574
7	英雄	0.567
8	不具合	0.562
9	カフェ	0.556
10	キャラクター	0.556

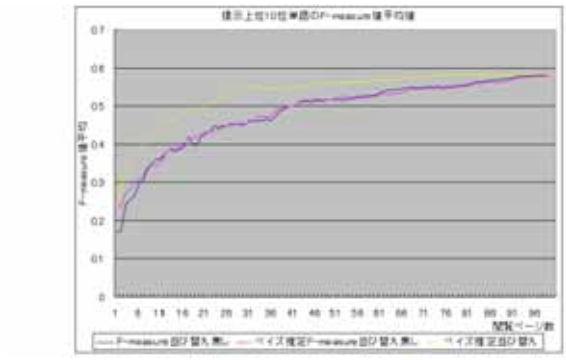

図 5　各閲覧時点におけるクエリ推定の精度 (オンライン)

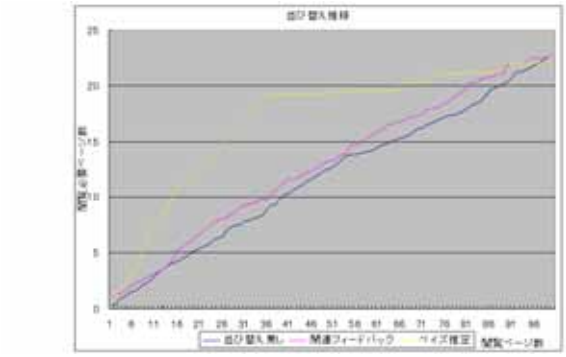

図 6　並び替えによる閲覧必要ページ数推移 (オンライン)

各閲覧段階での提示された上位 10 までのクエリの平均 F-measure 値を図 7 に示す．ページの並び替えによるクエリの F-measure 値の上昇はかろうじて多少認められるものの, その他はほとんど変わっていない．もともと必要ページ集合に存在しうる F-measure 値が低いため難しいと考えられる．

ページの並び替えの比較を図 9 に示す．

各並び替え効率は, ソート無しが 0.492, 関連フィードバックが 0.624, ベイズ推定が 0.61 となっている．

こちらのテーマでは並び替え効率で関連フィードバックが提案手法を上回っているが, このテーマでは出現単語数が少ないため単語ベクトル空間においてスパースネスの影響が少ないということも理由の一つとして考えられる．

表 4　上位 10 位単語 (ゲーム機)

順位	単語名	F-measure 値
1	マンガ	0.593
2	www	0.533
3	だっ	0.522
4	counter	0.508
5	考え	0.506
6	らしい	0.493
7	一般	0.493
8	感	0.49
9	星	0.476
10	てる	0.468

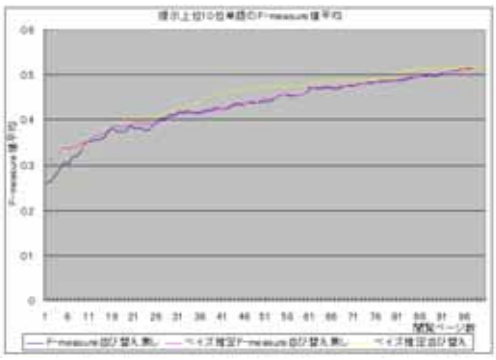

図 7　各閲覧時点におけるクエリ推定の精度 (ゲーム機)

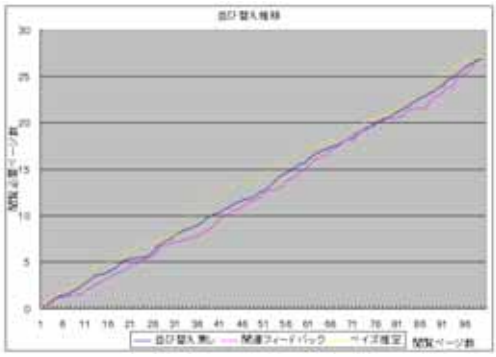

図 8　ページ並び替えの評価 (ゲーム機)

5. おわりに

本論文ではユーザの検索を支援するために, ユーザが閲覧を進める度に確率的な基準によってユーザに必要であると思われるキーワードと文書の 2 つを提示することでお互いが補い合うことでより良い成果を上げる手法を提案した．実験として簡単な Web 検索を想定して評価を行い, 簡単なタスクであれば良い成果を上げられると考えられるものの, 人の意見など曖昧な目的に沿って分類するためには単純に一単語を単位とした検索支援では難しいことが考えられる．今回は文書における単語の出現頻度や, 各単語間における相関などの情報は用いておらず, そうした情報をうまく用いることができればればさらに使い易いシステムを考案できる可能性があると考えられるため, 今後は確率的なフレームワークを拡張してそれらの情報を柔軟に取り込めるような仕組みつくりについて考案していきたい．

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

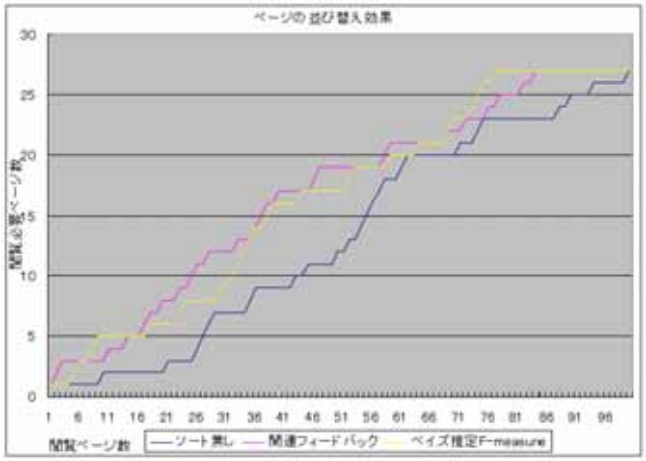

図 9　並び替えによる閲覧必要ページ数推移 (ゲーム機)

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

[8] 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)

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

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

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