Webページの既読結果を用いたWeb検索結果の網羅的閲覧

厚見　悠太†　　大島　裕明††　　田中　克己††

† 京都大学工学部情報学科　〒606–8501 京都府京都市左京区吉田本町
†† 京都大学大学院情報学研究科社会情報学専攻　〒606–8501 京都府京都市左京区吉田本町
E-mail: †{atsumi,ohshima,tanaka}@dl.kuis.kyoto-u.ac.jp

あらまし　本研究では，検索結果を上位から閲覧した時の既読結果を利用し，残りの検索結果を少数に絞って提示することで検索結果全体を網羅的に閲覧させる手法を提案する．Web 情報検索エンジンで返される文書は莫大な数であることが多い．これを網羅的に見ようとすると，既存の検索エンジンではすべての検索結果を 1 つずつ見ていくしかなく，多大なユーザ負荷がかかる．本研究の目的は少数の結果を見せるだけで，検索結果全体を閲覧するのと同じ情報をユーザに与えることである．そこで本稿では既読結果と重複した内容のページを省いた後，ページの情報量に基づきランキングを行う．そして, そのランキング結果から少数のページを選択し，提示する．またシステムを実装し，結果を見る際の閲覧量の増減と情報の網羅量を評価する．

キーワード　網羅的閲覧，既読結果，ランキング

1. はじめに

近年のインターネットや検索エンジンの発展は目覚ましい．適合性フィードバック [1] や山本ら [2] による Rerank.jp(注1)などの関連研究によって，ユーザが明確な検索意図をもっている場合，望む Web ページを提示することが比較的容易となった．しかし特定のページではなく多くのページを網羅的に閲覧したい場合，既存の検索エンジンでは十分な結果を得ることは難しい．また，多くのページを網羅的に閲覧することに関する研究は十分とは言えない．既存の検索エンジンでページを網羅的に閲覧しようとした場合，検索エンジンから返ってきた結果の数は莫大であるので，それらを網羅的に閲覧するのは多大な労力を要するのが現状である．結果として，検索結果におけるランキングの上位のページのみ閲覧され，ランキング下位のページは閲覧されないでいる．

そこで本研究では既に閲覧した検索結果を用いて，まだ閲覧していない未読の検索結果を少数のページに絞り，提示する手法を提案する．ページに含まれる情報をなるべく保持しつつ，閲覧量をなるべく少なくすることが本研究の目的である．

提案手法は大きく分けて 3 つの手法で構成されている．1 つ目は対象文書の前処理である．これはクエリと文書間の距離をある手法で計算し，その距離が大きいものはノイズとして除去する手法である．この段階で除去されたページは後の手法にも影響を与えず，評価にも加えない．言わば実験対象ページを制限している．2 つ目は類似したページの除去である．これはまず既読の Web ページから特徴語を要素とするベクトルを作成する．そして同じく未読の Web ページから作成したベクトルとの類似度を計算し，その類似度に基づき，ページを除去する．3 つ目はページのランキングである．これはある指標で計算される “Cover 度 “というものに基づきランキングを行う．そしてそのランキングに基づき，あらかじめ決められた数のページを出力する．

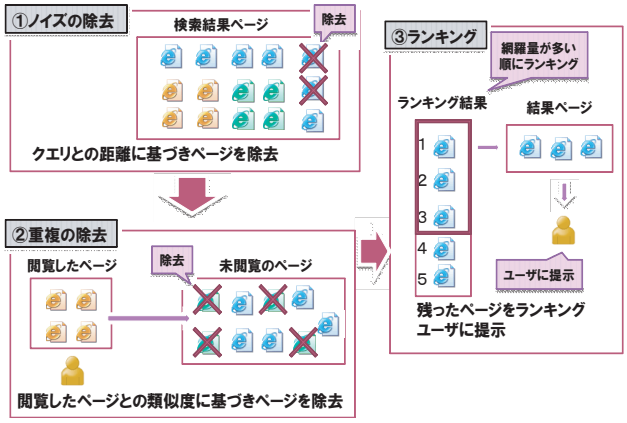

図 1　本研究の概要

また本稿ではシステムを実装し，閲覧量と網羅量という 2 つの観点から手法の評価実験を行っている．今回は 3 つの比較実験を行っている．

本研究の概要を図 1 に記す．

2 節では前処理について述べる．3 節では，類似したページを除去する手法の詳細を述べる．4 節では，ランキング手法について述べる．5 節ではシステムの実装について述べる．6 節では実験と評価を行い，考察を述べる．7 節では関連研究について述べ，8 節では本稿のまとめと今後の課題を述べる．

2. 前　処　理

本節では前処理について述べる．検索エンジンにクエリを投げて返ってくる結果の全てが，必ずしもクエリとの関連が深いページとは限らない．そのようなページを本稿では “ノイズ “と呼ぶ．このノイズを検索エンジンから文書集合を取得する時に除去する．

(注1)：Rerank.jp http://rerank.jp/

大まかな流れとしては，検索エンジンにクエリを投げ検索結果を取得する．そしてクエリと文書間の距離をはかり，その距離に従ってノイズを除去する．

2.1 クエリ-文書間の距離

本項ではクエリ-文書間の距離の計算手法について説明する．

２つの文書間の距離を測るのによく使われるものにコサイン類似度がある．しかし今回はこの手法は適用できない．理由はクエリはほぼ 1〜2 単語であり，距離を測るにはあまりにも短すぎるというものである．

そこで今回はクエリ-文書間の距離の計算手法に Okapi BM25[3] を用いた．これはクエリと文書間の距離を，文書がクエリを含む確率で計算している．

Okapi BM25 は以下の式で定義される．

$$score(D,Q)=\sum_{i=1}^{n} IDF(q_i)\cdot\frac{tf(q_i,D)\cdot(k_1+1)}{tf(q_i,D)+k_1\cdot(1-b+b\cdot\frac{|D|}{avgdl})} \quad (1)$$

但し，Q はクエリ全体の集合，D は対象とする文書，q_i はクエリ 1 単語，$tf(q_i,D)$ は語出現頻度を表し，文書 D の中に q_i が何回現れたかを表す．k_1 と b はフリーパラメータであり，通常は $k_1=2.0$,$b=0.75$ とする．$|D|$ は文書の長さ (単語単位) を表し，$avgdl$ は文書の平均の長さを表す．ここでの $IDF(q_i)$ は次項で定義する．

文書の長さが平均より長く，クエリがあまり現れない文書は $score(D,Q)$ が低くなるようになっている．これは長い文書の中でクエリが数回しかでない文書はクエリとの関連度が低い文書であると言う意味である．

2.2 Inverse Document Frequency の定義

本項では $InverseDocumentFrequency(IDF)$ を定義する．

t を文書内に現れる語とすると $IDF(t)$ は以下の式で定義される．

$$IDF(t)=\log\frac{N}{DF(t)} \quad (2)$$

N は全ページ数を表す．$DF(t)$ は以下の式で定義される．

$$DF(t)=\sum_{p_i\in P}\delta(t,p_i) \quad (3)$$

ここでの δ はページ p_i の中に語 t が含まれていれば 1 を，含まれていなければ 0 を返す関数である．

このようにして IDF は定義される．前項では $IDF(q_i)$ となっているので，クエリ q_i に関する IDF を計算する．

2.3 ノイズの除去

本項ではノイズの除去について述べる．2.1 項で計算した $score(D,Q)$ の値が低いものはノイズとみなし，検索エンジンから文書集合を取得する際に除去する．式で表すと以下のようになる．

$$score(D,Q)<\theta \quad (4)$$

但し，$score(D,Q)$ は 2.1 項で定義した値を表し，θ は閾値を表す．この閾値 θ によって除去されるノイズの数は変わるので，慎重に決定しなければならない．本稿では前もって実験を行い，この値を決定する．詳細は 5 節で述べる．

3. 類似したページの除去

本節では既読のページと類似したページを除去する手法について述べる．

まず既読のページ，未読のページを定義する．従来の検索エンジンで返される検索結果の上位 n 件のページをユーザが閲覧したと仮定し，その n 件のページを既読のページと呼ぶ．そして検索エンジンが返す全検索結果 N 件から既読のページを引いたもの，$N-n$ 件のページを未読のページとする．

既読のページと内容が類似した未読のページは閲覧する必要はないと仮定し，その未読のページを除去する．

3.1 検索結果ページの形態素解析

本項では検索結果ページの形態素解析について述べる．形態素解析とは自然言語で書かれた文を形態素の列に分割し，それぞれの品詞を判別する作業を指す．

形態素解析ソフトとして，今回は MeCab (注2) を使用し，これを用いて検索結果ページを形態素解析する．

この時品詞は名詞のみを扱い，他は結果から除外する．これは名詞がページの特徴を表す語であることが多いという仮定に基づいている．また一般語と呼ばれる多くの文に出現する語，“男 “や “一 “などはストップワードとして扱い，形態素解析結果から除外している．

3.2 ページの特徴ベクトルの作成

本項では前項で得られた形態素解析結果を用いて，形態素を要素とする特徴ベクトルを作成する手法について述べる．

ページ 1 つ 1 つに含まれる形態素を要素とするベクトルを作成する．あるページ p_i に対応するベクトルを v_i とし，その要素を w とすると，ページの特徴ベクトルは次式で定義される．

$$v_i(w_1,w_2,w_3,\ldots,w_n) \quad (5)$$

ここでベクトル v_i の要素 w は，前項で得られたページ内の形態素 t によって決まる．ベクトル v_i の要素の 1 つを w_j とすると，w_j は以下の式で定義される．

$$w_j=tf(t_j,p_i)\times IDF(t_j) \quad (6)$$

但し p_i はベクトル作成の対象となっているページである．ここでの $tf(t_j,p_i)$ は語出現頻度を表し，ページ 内に現れる語の出現回数を表す．

$IDF(t_j)$ は 2.2 項の (2)，(3) の式で定義される．本項では t_j に関する IDF を計算している．このようにして特徴ベクトル v_i が作成される．

3.3 重複判定手法

本項では前項で得られた特徴ベクトルを用いて類似度を計算し，その類似度に基づいて重複を判定してページを除去する手法について述べる．

前項でページの特徴を表す特徴ベクトルを作成した．この手法を既読のページと未読のページ全てに適用し，それぞれの

(注2)：“MeCab:Yet Another Part-of-Speech and Morphological Analyzer“.
http://mecab.sourceforge.jp.

ページの特徴ベクトルを作成する．既読のページの特徴ベクトルと未読のページの特徴ベクトルの類似度を計算し，類似度が高ければ，それらのページは内容が重複していると見なし，ページを除去する．

類似度を計算し重複したページを除去する手法として以下の 2 つを提案する．

コサイン相関法　既読のページの特徴ベクトルと未読のページの特徴ベクトルとのコサイン類似度を計算し，あらかじめ決められた閾値以上の値を持てば，その未読のページを重複しているとみなす．

特徴語抽出法　既読のページからあらかじめ決められた閾値以上の特徴語を抽出し，未読のページがそれらの特徴語を含めば，その未読のページを重複しているとみなす．

以下でコサイン相関法，特徴語抽出法両方の手法の詳細を述べ，両手法を比較する．

3.4　コサイン相関法の詳細

本項では前項で述べたコサイン相関法の詳細について述べる．

まず類似度の計算方法について述べた後，重複したページの除去手法について述べる．

3.4.1　コサイン類似度

コサイン相関法の類似度の計算について述べる．

既読のページ集合を P_r，未読ページ集合を P_{ur} とおき，それぞれのページ集合に含まれるページを $p_i \in P_r, p_j \in P_{ur}$ と置く．そして既読のページ p_i に対応する特徴ベクトルを v_i，未読のページ p_j に対応する特徴ベクトルを v_j と置くとき，コサイン類似度は以下で定義される．

$$cos(v_i, v_j) = \frac{v_i \cdot v_j}{\parallel v_i \parallel \times \parallel v_j \parallel} \tag{7}$$

但し分子はベクトルの内積，分母はベクトルのノルムの乗算を表す．分母はベクトルの長さを正規化しており，2 つのベクトル間のなす角度で値が決定される．また，2 つのベクトル v_i，v_j が完全に同一の方向を指す場合，$cos(v_i, v_j)$ の値は 1 となり，なす角度が直角となる場合，$cos(v_i, v_j)$ の値は 0 となる．

3.4.2　重複ページの除去

次にコサイン相関法の重複したページの除去について述べる．

既読のページ集合 P_r におけるページ $p_r \in P_r$ に対応する特徴ベクトルを v_r とし，未読のページ集合 P_{ur} におけるページ $p_{ur} \in P_{ur}$ に対応する特徴ベクトルを v_{ur} とする．この時，その 2 つのページが重複しているかどうかは以下の式で判定される．

$$cos(v_r, v_{ur}) > \eta \tag{8}$$

但し cos は前項で定義されたコサイン類似度を表し，η はあらかじめ決められた閾値を表す．

この判定手法を全ての未読のページに適用する．ある未読のページと全ての既読のページと重複しているか判定し，もし一つでも重複していると判定されればその未読のページは重複していると判断し，除去する．これを全ての未読のページに適用すれば重複したページが除去される．

3.5　特徴語抽出法の詳細

本項では 3.3 項で述べた特徴語抽出法の詳細について述べる．まずページの特徴ベクトルからの特徴語抽出手法について述べ，重複したページの除去手法について述べる．

3.5.1　特徴語抽出

特徴語抽出法の特徴語の抽出について述べる．

特徴語抽出法では直接類似度を計算するようなことはせず，まずベクトルから特徴語をいくつか選択し，それに基づいて重複を判定する．

まず特徴語の抽出方法を定義する．ページ p に対応する特徴ベクトルを v とし，その要素集合を W，要素それぞれを $w_1, w_2, \ldots, w_n$ とする．この時ページ p の特徴語は以下の式で抽出される．

$$W = \{w_j \mid w_j > \zeta\} \tag{9}$$

$$w = \{w_k \mid w_k = max(w_0, w_1, w_2, \ldots, w_n)\} \tag{10}$$

但し ζ はあらかじめ決められた閾値を表し，$max(a, b, c)$ は () の中の要素の中で一番値が高いものを出力する関数である．

それぞれの式の意味は (9) はあらかじめ決められた閾値以上の特徴語をすべて抽出するという意味であり，(10) は要素の中で一番重みが高い特徴語，即ち一番ページの特徴を表している語を抽出するという意味である．

3.5.2　重複ページの除去

次に特徴語抽出法の重複したページの除去について述べる．

前項の手法を使って既読のページ全てから特徴語を抽出し，特徴語のリストを作成する．そして未読のページに含まれる語の内，1 つでもリストに含まれる語があればそのページを重複したページと見なし，除去する．

3.6　手法の比較

本項では前項までで述べた 2 つの手法を比較する．

コサイン相関法はベクトルの要素である語をすべて考慮している．よって多くの語の類似度を計算することになり，1 つ 1 つの語の重みが小さくなってしまう．また閾値を設定の方法が難しく，精度がその値に依存してしまうという問題もある．

特徴語抽出法は一部の語しか考慮されておらず，精度上の問題がある．また精度がページの特徴語の抽出の仕方に依存してしまう．しかしページにおける特徴語を上手く抽出できれば，除去されるページがわかりやすいという利点と計算時間が短いという利点がある．

どちらも一長一短であるので，両手法の比較実験を行う．実験の詳細は 6 節で述べる．

4.　ランキングの作成

本節では前節で除去されなかった未読のページからランキングを作成し，そこから一定数のページを選択する手法について述べる．

前節では既読のページとの類似度を基にページを除去したが，この段階では未読のページはまだ多く残っている．そこでページ内容の情報網羅量を本稿では “Cover 度 “と定義し，こ

の “Cover 度 “に基づき，残っている未読のページを再ランキングする．最終的にランキング上位からページを選択し，ユーザが望む件数のページを取得する．

Cover 度の計算方法は以下の 2 つである．

DF 法　未読のページに含まれる単語の Document Frequency(DF) 値に基づき計算する．

単語法　全ての未読のページ中に含まれる全単語中の，対象とするページが含む単語の割合に基づき計算する．

DF 値が高い単語はより網羅的な単語であり，そのような単語を多く含むページは情報網羅量が多いと言える．またページ内により多くの単語を含むページも情報網羅量が多いと言える．これも前節の重複除去手法と同じく 6 節で比較実験を行う．

4.1 Cover 度の定義

本項では Cover 度を定義する．ページの Cover 度はページに含まれる情報が網羅性が高いものであれば値を高くするように定義する．つまり広い範囲の情報をカバーするようなページは Cover 度が高くなる．

4.1.1 DF 値による Cover 度の定義

DF 法の詳細を述べる．

p を対象とする未読のページ，t_i をページ p に含まれる語，P をランキング済みのページ集合とすると，$cov(p)$ は以下の式で定義される．

$$cov(p) = \sum_{t_i \in p} DF(t_i) - \sum_{t_i \in P} DF(t_j) \tag{11}$$

但し $DF(t_i)$ は 2.2 項の (3) 式で定義されている DF と等価である．

この式は多くの検索結果に出現する語を多く含むページは，情報網羅量が多いページであるという仮定に基づいている．また減算は既にランキングされたページが含む頻出する単語をできるだけ含まないページが高い値を持つようになっている．これは同じ頻出する単語を持つページは同じ内容を表す可能性が高いので，そのようなページに高い値を与えないようにする措置である．

ランキング作成の流れとしては，まず $cov(p)$ の値が一番高いページ p_{max} をランキング 1 位の結果とする．この時はまだ減算は行われない．次に p_{max} をランキング済みのページ集合に追加し，$cov(p)$ を再計算する．そして $cov(p)$ の値が一番高いページ $p_{max'}$ をランキング 2 位の結果とし，またランキング済みのページ集合に追加する．これを繰り返し，ランキングを作成する．

4.1.2 未出の単語数による Cover 度の定義

単語法の詳細を述べる．

p を対象とする未読のページ，$|W|$ を未読のページ全てに含まれる単語の総数，$|w|$ を p に含まれる単語で，かつ既にランキングされたページ集合 P に含まれない単語の総数とすると，$cov(p)$ は以下の式で定義される．

$$cov(p) = \frac{|w|}{|W|} \tag{12}$$

この式は全てのページに含まれる単語に対する対象とするページに含まれる単語の割合を意味している．既にランキングされたページ集合に含まれる単語を除外しているのは DF 法の場合と同じく，既にランキングされたページと同じ単語を含むページは同じ内容を表す可能性が高いので，そのようなページに高い値を与えないようにしている．

ランキング作成の流れは DF 法と同様である．

4.2 ページの選択

本項では前項でランキングされた結果からページを選択する手法について述べる．

前項までで作成したランキングから，ページを上位から順に選択する．未読のページ集合から内容の重複を除去し，ランキングを作成してページを選択することで，未読のページ集合を任意の n 件のページまで圧縮することができる．これにより，ユーザは未読のページを全て閲覧することなくほぼ全容を把握することができ，ページを見るというユーザの負荷を大幅に軽減することができる．

5. 実　　装

本節ではまず，2 節，3 節，4 節で述べた手法を実装するにあたって，調整した点について述べる．

調整点として，既存の検索エンジンの選択，ページの処理を行う範囲，手法の調整，閾値の決定，ユーザの閲覧ページにおける調整，実装言語などがある．1 つ 1 つの詳細を述べる．

5.1 検索エンジンの選択

本項では既存の検索エンジンの選択について述べる．

現在利用可能な検索エンジンは数多く存在する．有名なもので言えば，Google(注3)，goo(注4)，Yahoo!JAPAN(注5)などがある．検索エンジンによってランキングは変わり，取得するページも変わる．

今回は Yahoo!JAPAN の検索 WebAPI を使い，Yahoo の検索結果を使ってシステムを実装した．

5.2 処 理 範 囲

本項では検索結果ページの処理を行う範囲について述べる．

ページにはそれぞれタイトル，スニペット，そしてページ内容があり，順に情報量が多くなっている．本来ならば，ページ内容から文章を抜き出し，それを解析して本稿の手法を適用すべきである．しかし本稿ではタイトル，スニペットのみの解析で十分であると仮定し，ページの解析を行っている．また処理速度の面から見ても，タイトル，スニペットのみの解析は高速であり，実際の使用のことを考えるとこの利点は大きいと言える．

5.3 前　処　理

本項では前処理における $score(D, Q)$ の計算の調整について述べる．

2 節で述べた $score(D, Q)$ において，いくつか調整する点がある．まず Q はクエリ群として定義していたが，実装では

(注3)：Google http://www.google.co.jp/

(注4)：goo http://www.goo.ne.jp/

(注5)：Yahoo! JAPAN http://www.yahoo.com/.

表 1　前処理における実験

クエリ \ 閾値	0.8	1.0	1.2
幕末	964	937	899
ゴルフ	973	958	932
インテリア	958	918	877

表 2　手法 [A] における実験

クエリ \ 閾値	0.1	0.2	0.3
幕末	505	754	821
ゴルフ	587	786	873
インテリア	399	706	802

表 3　手法 [B] における実験

クエリ \ 閾値	0.2	0.3	0.4
幕末	917	917	917
ゴルフ	851	851	877
インテリア	906	906	906

1 つのクエリとする．パラメータである k_1，b の値はそれぞれ $k_1 = 2.0$，$b = 0.75$ とする．また (4) 式における閾値 θ は事前に実験を行い，約 10% の文書が除去される程度に調整した．表 1 に実験結果を示す．表の値は指定された閾値でノイズ除去を行い，除去されなかった文書の数を表している．クエリは “幕末 “，“ゴルフ “，“インテリア “とした．

この結果から除去したノイズが 10% を超えていない $\theta = 1.0$ が妥当と考えられるので，以後 $\theta = 1.0$ とする．

5.4　閾値の決定

本項では残りの閾値について述べる．

(4) 式における閾値 θ 以外に，(8) 式の閾値 η，(9) 式の閾値 ζ がある．この 2 つの値は類似度の判定や特徴語の抽出に密接に関係している．よってこれも事前に実験を行う．表 2，表 3 はそれぞれの重複除去手法を用いた際の除去されなかったページの数である．既読のページ数は全て 10 件とし，クエリは前処理の時と同じく “幕末 “，“ゴルフ “，“インテリア “とした．

表 2 の結果を見ると閾値 η によって除去されるページ数が大きく変動していることがわかる．閾値 0.1 の場合はページが除去されすぎており，適切な閾値とは言えない．また閾値 0.3 の場合は多くのページが残っており，手法の影響が弱いと考えられる．よって閾値 η を 0.2 とする．

表 3 の結果を見ると閾値 ζ によらず，ほぼ一定のページが除去されている．よって閾値 ζ は手法にあまり影響を与えないと考え，閾値を中間値である 0.3 とした．

5.5　特徴語抽出法

本項では 3 節で述べた特徴語抽出法の調整について述べる．

この手法では文書から特徴語抽出を行う際に (9) 式，(10) 式という 2 つの式を定義した．本稿では文書の解析範囲はタイトルとスニペットである．タイトルは一般的に文が短いので 1 つの単語でその特徴を十分に表現できると考え，タイトルの特徴語抽出には (9) 式を用いた．また，スニペットは一般にタイトルに比べて文章が長いので，複数の単語で特徴を表現する必要があると考え，スニペットの特徴語抽出には (10) 式を用いた．

5.6　結果の閲覧

本項では検索結果の閲覧 · 未閲覧の判定について述べる．

既読のページの定義はユーザが閲覧済みのページである．そこで何をもってユーザがページを閲覧したかを定義する必要がある．本来はユーザが中身まで見たページを閲覧済みとする，もしくはユーザの行動パターンに基づき，閲覧時間やクリック率などを使ってページにどれほど関心があるかを考慮すべきであるが，今回はそのようなことは行わず，1 度システムが提示したページはすべて閲覧済みであるとする．つまり表示したページはすべてユーザが閲覧したと仮定し，表示した全てのページを既読のページとする．

5.7　実　　装

本項ではシステムの実装について述べる．2 節，3 節，4 節で述べた手法を用いてシステムを Windows 上で実装した．実装する際のプログラミング言語は C# を用いた．

6.　実　　験

本節では実装したシステムを使った実験，評価について述べる．まずどのように実験を行うかを決めた後，実験で得たデータの評価方法，評価軸を決める．そして大きく分けて 3 種類の内容の実験を行い，実験結果をグラフで表してから結果を考察する．

6.1　実験の方法

本項では実験の方法について述べる．

まず既存の検索エンジンから検索結果を取得する．今回の実験では検索エンジンは Yahoo!を使い，取得する結果は 1000 件とする．取得した結果の中から n 件を閲覧する．その n 件の結果を既読ページとし，残りの未読のページから本研究の手法を用いて一定数のページを抽出し，結果を提示する．そしてその結果ページを評価する．評価方法については次項で説明する．

6.2　評価の尺度

本項ではシステムが返した結果を評価する際の評価尺度について述べる．

閲覧量と網羅量の 2 軸で評価する．閲覧量はページ全体に対するユーザが実際に読んだ割合，網羅量はページ全体に対するユーザが実際に読んだページで内容が網羅されているページの割合である．この 2 軸を使った理由は，ページの閲覧量を増やすことによってページ全体の内容がどの程度網羅されるかを調べるためである．閲覧量が少なく網羅量が多い場合，システムは良い評価を得る．

6.2.1　閲　覧　量

まず閲覧量を定義する．

$|P|$ を取得した全検索結果ページ数，$|p|$ を実際に閲覧したページ数とすると閲覧量は次式で定義される．

$$\frac{|p|}{|P|} \tag{13}$$

この時 $|P|$ にはノイズで除去されたページは入っていない．また $|p|$ にはユーザが最初に閲覧する既読のページとシステムが返したページ，どちらも数に入っている．

6.2.2 網 羅 量

次に網羅量を定義する．

$|P|$ を取得した全検索結果ページ数，$|p|$ を実際に閲覧したページ数，$|c|$ を未読のページの中で実際に閲覧したページ p で内容が網羅されたページ数とすると，網羅量は次式で定義される．

$$\frac{|p|+|c|}{|P|} \tag{14}$$

閲覧量の時と同じく，$|P|$ にはノイズで除去されたページは入っておらず，$|p|$ にはユーザが最初に閲覧する既読のページとシステムが返したページ，どちらも数に入っている．

$|c|$ を求めるには未読の検索結果ページをすべて閲覧し，閲覧したページ p で内容が網羅されているかを判断しなければならない．$|p|$ が少ない場合，未読のページ数 $|P|-|p|$ は非常に多くなり，多大な労力を要する．そこで代案として $|c|$ の推量を行う．

まず未読のページ $P-p$ の中から $|P^{'}|$ 件を無作為抽出する．その $|P^{'}|$ 件の中で，既読のページ p で内容が網羅されていると判断したページ数を $|c^{'}|$ とすると，次の推量が成り立つ．

$$|P|-|p| : |c| \simeq |P^{'}| : |c^{'}| \tag{15}$$

よって $|c|$ は次式で計算される．

$$|c| \simeq (|P|-|p|)\frac{|c^{'}|}{|P^{'}|} \tag{16}$$

この推量は $|P^{'}|$ が十分な量でなければ成り立たない．どのくらいの量があれば推量が可能かを次式を用いて計算した．[4]

$$|P^{'}| \geq (\frac{Z_{\alpha/2}}{x})^2\pi(1-\pi) \tag{17}$$

ここで，$Z_{\alpha/2}$ は信頼度，x は許容誤差，π は母推定比率である．今回の実験では信頼度 95% ，許容誤差 10% ，母推定比率 0.5 でサンプル数を決定した．サンプル数は 96.04 となり，$|P^{'}|$ を 96 とした．

6.3 実験の種類

本項では行った実験を 3 種類に分け，それぞれの実験の詳細を述べる．

まず，行った実験は次の 3 種類である．

1 既存の検索エンジンである Yahoo!のランキングと本手法との比較．

2 本研究で提案した手法同士の比較．重複除去手法とランキングにおける 2 つの手法をそれぞれ比較する．

3 既読ページ数の比較．最初のページを閲覧する段階でどれだけページを読んだかを比較する．

6.4 実 験 結 果

本項では実験結果について述べる．

図 2，図 3，図 4 は Yahoo での検索結果と本システムを，前項で定めた評価軸に従い評価した結果のグラフである．横軸は閲覧量，縦軸は網羅量を表している．今回の実験では全検索結果が 1000 件なので 100 件 (10%) 以上の結果を見るのは本稿の趣旨に沿わないと考え，閲覧量は 10% までとした．クエリ

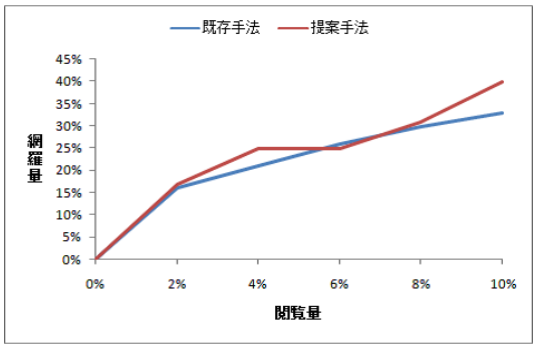

図 2 既存のランキング結果と提案手法との比較（クエリ “幕末 “）

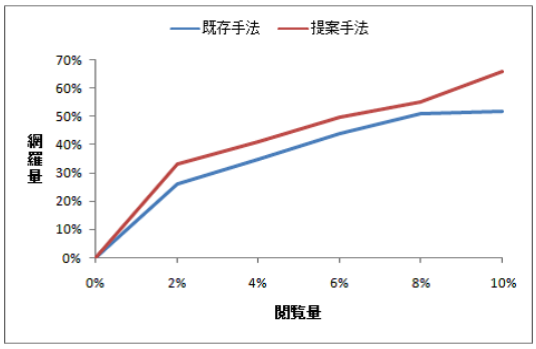

図 3 既存のランキング結果と提案手法との比較（クエリ “ゴルフ “）

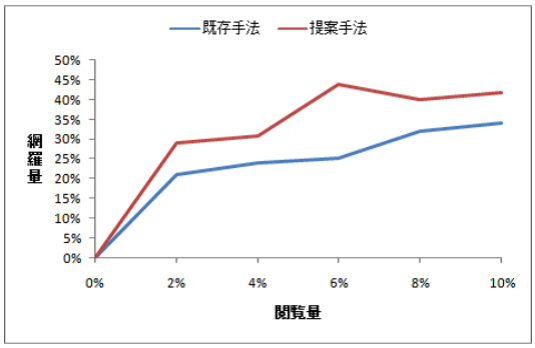

図 4 既存のランキング結果と提案手法との比較（クエリ “インテリア “）

はそれぞれ “幕末 “，“ゴルフ “，“インテリア “であり，重複除去手法は特徴語抽出法，ランキング手法は DF 法である．また既読ページ数は 10 件とした．

図 5，図 6 は 3 節，4 節で述べた手法同士を比較したグラフである．これも前項で定めた評価軸に従い評価した結果のグラフである．クエリは “幕末 “であり，図 5 のランキング手法は DF 法を，図 6 の重複除去手法は特徴語抽出法を用いた．また既読ページ数は 10 件とした．

図 7 は既読ページ数の比較である．検索エンジンで返されたページをランキング上位から順に閲覧する際の閲覧ページ数を変化させて比較している．クエリは “ゴルフ “であり，重複除去手法は特徴語抽出法を，ランキング手法は DF 法を用いた．

6.5 結果の考察

本項では結果の考察を述べる．

まず図 2，図 3，図 4 の結果を考察する．全ての結果におい

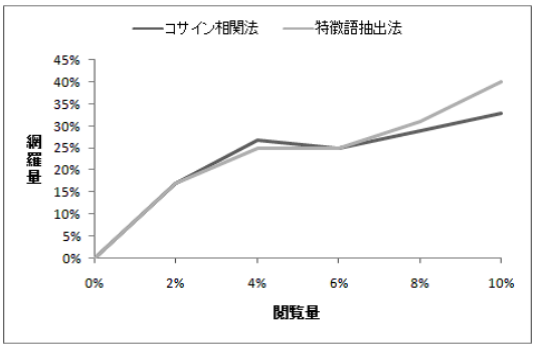

図 5　コサイン相関法と特徴語抽出法との比較（クエリ “幕末 “）

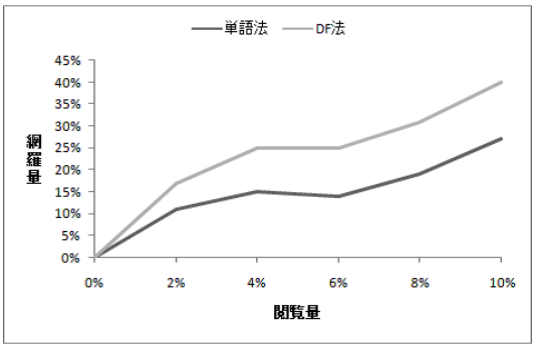

図 6　DF 法と単語法との比較（クエリ “幕末 “）

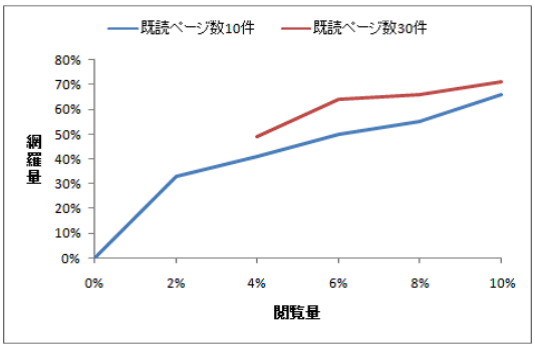

図 7　既読ページ数の比較（クエリ “ゴルフ “）

て既存のランキングによる結果より提案手法による結果の方がいい結果が出ている．特にクエリが “インテリア “の場合に非常に良い結果が出ている．逆にクエリが “幕末 “の場合にはあまり良い結果が出ていない．クエリが “ゴルフ “の場合は既存のランキングによる結果との差異はあまりないが，全体的に網羅量が高くなっている．理由はいくつか考えられる．

まずクエリが “インテリア “の場合，既存の検索エンジンでは同じような内容のページが上位に固まってランキングされていた．このような場合本手法は有効に働き，上位のランキングにない内容のページを上手く補完することができた．クエリが “幕末 “の場合，結果ページで返される内容が多岐に渡っていたことが精度の悪化の原因だと考えられる．クエリが “ゴルフ “の場合，既存の検索エンジンのランキング上位の結果が比較的網羅的な内容のページを含んでいたため，本手法と結果があまり変わらなくなってしまったと考えられる．

次に図 5，図 6 の結果を考察する．図 5 を見ると特徴語抽出法の方がコサイン相関法よりも若干良い結果を残しているが，顕著な違いは見られない．しかし計算時間の面で言えば特徴語抽出法の方が優れているので結果として特徴語抽出法の方が良い結果が出たと言える．図 6 を見ると DF 法の方が単語法よりもかなり良い結果を残している．この結果から 3 節で述べた重複除去手法よりも 4 節で述べたランキング手法の方がより本研究の手法に影響を及ぼしていることがわかった．

最後に図 7 の結果を考察する．30 件のグラフが 4% からしかないのは 30 件見ることを前提としているため，2% の結果がとれないからである．図を見ると明らかに既読ページを 30 件としたほうが良い結果が出ている．これは既存の検索エンジンの結果を多く見れば重複除去が有効に働くからであると考えられる．

このように本手法はクエリ，ランキング手法によってかなり結果が変わることがわかった．また適切なクエリ，既読ページ数を選べば本手法は有効に働くことがわかった．

7.　関 連 研 究

本節では関連研究について述べる．

湯本ら [5] はユーザの求める情報の全容を表すページ集合を発見する全容検索を提案している．この研究ではクエリが決まればそれに対する解が一意に決まり，その解にユーザが満足していない場合のことを考慮していない．本研究ではユーザの既読のページを考慮しており，結果に満足していない場合も新たにページ候補を提示することが可能であり，それが独自性である．

松生ら [6] は検索結果ページの概要をキーワード集合で表現する手法を提案している．この研究では概要をキーワード式と呼ばれるキーワードを論理式で表した式によって表現している．本研究では概要をキーワードで表すことはしておらず，ページ集合で表現している．その点が本研究との差異である．

クラスタリングの研究で言えば Hearst ら [7] の Scatter/Gather がある．クラスタリングではクエリが決まると結果が一意に決まる．しかしクラスタリングされた結果に満足しないユーザはそれ以上のアプローチをとることができない．また既に読んだページと似た内容のページを読む可能性も多い．本研究では結果をまとめるのではなく結果を絞ることを目的としており，この研究とは目的が違う．

類似の文書を除去する手法に関連する研究では Zhang ら [8] の novelty/redundancy ditection がある．これはクエリとは関連 (relevant) があるが他の文書と冗長でなく新規性 (novelty) がある文書を発見する手法を提案している．本研究では冗長な文書を除去するだけでなく，情報の網羅性という概念を導入し，冗長な文書を排除した後により網羅的な内容の文書を提示している．

8.　まとめと今後の課題

本節ではまとめと今後の課題について述べる．

本稿ではユーザの既読ページを用いて未読のページをなるべく情報量をおとさず少数のページに絞り，提示する手法を提案

した．また実験・評価を行い，手法の妥当性を検証した．

今後の課題としては既読のページの定義であるユーザがページを閲覧したかどうかを決める際に，ユーザの行動履歴 (UserBehaviorData) を使うことが考えられる．ページの滞在率やクリック率などを考慮して閲覧・未閲覧の判断をしていく予定である．

ページの解析範囲についてはページのタイトルとスニペットだけでなく，ページの内容にまで踏み込み多くの文章の解析を行う予定である．またページの内容を解析した場合とタイトル・スニペットのみで解析した場合との比較実験を行い，内容を解析する効果を検証する予定である．

今回の研究では再帰性に関しての議論がなされていなかったが，この研究の延長として再帰性を導入することは比較的容易である．つまりシステムが提示したページを閲覧したページ（既読のページ）としてから本手法を適用すれば，再びシステムがページを提示することが可能である．再帰性を導入する意味を考え，今後導入していく予定である．

評価に関しては，今回の実験では評価する人数が少なく正確さにかけているのが課題である．今回は実験のタスクが膨大なので調査人数を増やすことができなかった．今後はタスクを少数に絞り，評価する人数を増やすことによって，より正確なデータをとる予定である．

謝辞　本研究の一部は，京都大学グローバル COE プログラム「知識循環社会のための情報学教育研究拠点」，文部科学省科学研究費補助金特定領域研究「情報爆発時代に向けた新しい IT 基盤技術の研究」，計画研究「情報爆発に対応するコンテンツ融合と操作環境融合に関する研究」（研究代表者：田中克己、課題番号 1809041），および，文部科学省研究委託事業「知的資産の電子的な保存・活用を支援するソフトウェア技術基盤の構築」，異メディア・アーカイブの横断的検索・統合ソフトウェア開発（研究代表者：田中克己）によるものです．ここに記して謝意を表すものとします．

文　　献

[1] Rocchio,J. et al,: Relevance feedback in information retrieval, The SMART Retrieval System:Experiments in Automatic Document Processing, pp.313-323(1971).

[2] 山本岳洋，中村聡史，田中克己: Rerank-By-Example: 編集操作に基づく検索結果の網羅的閲覧，DBSJ Letters, Vol. 6, No. 2, pp, 57-60(2007).

[3] Robertson, S. and Walker, S.: Okapi/Keenbow at TREC-s, NIST SPECIAL PUBLICATION SP, pp. 151-162(2000).

[4] 東京大学教養学部統計学教室編: 基礎統計学　自然科学の統計学，東京大学出版会.

[5] 湯本高行，田中克己: Web ページ集合を解とする全容検索，情報処理学会論文誌（データベース），Vol. 48, No. 11, pp. 83-92(2007).

[6] 松生泰典，是津耕司，小山聡，田中克己: 検索結果の概要を表すキーワード式生成による質問修正支援，電子情報通信学会第 16 回データ工学ワークショップ (DEWS2005)1C-i9(2005).

[7] Heast, M. A. and Pedersen, J. O.: Reexamining the cluster hypothesis: scatter/gather on retrieval results, SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM Press, pp. 76-84(1996).

[8] Zhang, Y., Callan, J. and Minka, T.: Novelty and redundancy detection in adaptive filtering, ACM Press, pp. 81-88(2002).