

ウェブページ間相互評価によるウェブ検索手法の提案と実装評価

荒谷 寛和[†] 藤田 茂^{††} 菅原 研次^{†††}

近年、ウェブ情報検索の分野で、PageRank に代表されるウェブのリンク構造を用いたランキング手法が主流となってきている。しかし、著名なサイトや一般的なトピックが有利に評価される傾向があり、一方で重要な情報を含むページが低く評価されることがある。本論文では、ウェブページ間で内容の類似に基づく相互評価を行うことで、セマンティクスを考慮した検索手法を提案する。次に、本提案手法に基づいて、ウェブ検索システムを設計し、評価実験のための試作システムを実装した。評価実験では、フィルタとして Google の検索結果上位 200 件を用い、本提案手法に基づく検索結果と Google の検索結果を比較した結果、提案手法が、検索者が望むランキングに近い結果であることを確認した。

Implementation and Evaluation of Search Engine Based on Mutual Evaluation among Web Pages

HIROKAZU ARATANI,[†] SHIGERU FUJITA^{††} and KENJI SUGAWARA^{†††}

In recent years, link-based ranking methods of web pages, such as the PageRanking algorithm of the Google, have been developed in order to improve the quality of searching function. The PageRanking algorithm calculates ranking of web pages based on only the structure of hyperlinks among web pages without semantic relationships among web pages. Therefore, a page which many users want to obtain from the WWW might be ranked in a low position of a retrieved list when it is not so popular. In this paper, we propose a semantic-oriented ranking method which calculates ranking of web pages based on mutual evaluation among web pages which calculate an evaluating value of the objective web page according to key words and its own content. We designed and prototyped a retrieving system based on the proposed method. The experimental system retrieved and ranked sets of 200 web pages according to given key words, and the results were analyzed comparing to the ranking result of the Google for the same key words.

1. はじめに

近年、インターネットの普及にともない、ウェブの情報量は膨大となっており、情報爆発、情報洪水と呼ばれる問題が生じている¹²⁾。このような膨大な情報から、必要な情報を高い精度で高速に検索するためのウェブ検索技術の開発が重要となっている。

第 3 世代型と呼ばれるサーチエンジンは、ウェブ

ページへの引用度の高さを、他ページからのハイパーリンクの参照数から求め、ランキングに利用している¹³⁾。代表的なサーチエンジンとして、PageRank を採用する Google¹⁾ がある。このようなサーチエンジンは、リンク元のウェブページの信頼度によってウェブページの重要度を決定するために、コンテンツの表記などによるスパムに強いことが知られている¹⁰⁾。しかし、リンクの意図が考慮されていないことによって、誰もが知っているような一般的なトピックや、著名なサイトが有利に評価されやすい一方で、検索キーワードに強く関連しているにもかかわらず、リンク数が十分でないために、下位に順位付けられるページがあり、利用者がこのページを見付けるまでに、たくさんのページを読む必要があるなどの問題があった⁵⁾。

この問題を解決するために、ウェブのハイパーリンク構造のより詳細な解析を行い、利用者の検索意図により近い検索結果を提供する方式が提案されてい

[†] 千葉工業大学大学院工学研究科情報工学専攻
Graduate School of Computer Science, Chiba Institute of Technology

^{††} 千葉工業大学情報工学科
Department of Computer Science, Faculty of Information and Computer Science, Chiba Institute of Technology

^{†††} 千葉工業大学情報科学部情報ネットワーク学科
Department of Information and Network Science, Faculty of Information and Computer Science, Chiba Institute of Technology

る^{4),6),11)}。しかしながら、これらの方式は、ハイパーリンク構造に基づくランキングであるために、リンクの意図が考慮されておらず、上記の問題の本質的な解決にはなっていない。

本論文では、上記の問題を解決するために、検索精度を向上させる手法として、ウェブページ間相互評価によるウェブ検索手法を提案し、この手法に基づいてシステムを実装し、評価実験を行うことを目的とする。ここで、評価基準として、本論文ではベクトル空間モデルを用いている⁷⁾。2章では、この手法に関する関連研究について述べ、3章では提案手法について述べる。4章では、提案手法に基づく検索システムを設計し、相互評価基準としてベクトル空間モデル VSM を採用した場合の相互評価機能の設計を5章に述べる。6章では、以上の設計に基づく並行プロセス型の実装について述べ、7章で実験結果、8章で結果についての考察を述べる。

2. 関連研究

ウェブのリンク構造を利用する HITS や PageRank に対して、ハイパーリンクの意図を考慮することで改良を試みる研究がある。Dean らによる Companion, Cocitation⁴⁾ では、同一ウェブページ内の複数のハイパーリンクに対して、近接したハイパーリンクは同一トピックを示す可能性が高いことから、注目するハイパーリンクに近接したハイパーリンクにのみ HITS を適用している。また、豊田による Companion+¹¹⁾ では、Companion に対して、注目するハイパーリンクに近接するハイパーリンクほど高い評価を与えることにより、Companion を改良している。また、ハイパーリンクに用いられるアンカテキストに着目し、アンカテキストは、リンク先の内容を比較的的確に示していることが多いことから、アンカテキストを、リンク先ページの一部として、検索質問に一致するものを Link Popularity の対象とする方法がある。同様にアンカテキストに着目した研究として、Haveliwala による Topic-Sensitive PageRank⁶⁾ がある。Topic-Sensitive PageRank では、ページに記述された内容と、そこで用いられるアンカテキストとの関連性を考慮し、その関連性が高いほどリンク先に与える PageRank 値を上げ、関連性が低いほど与える PageRank 値を下げる。以上にあげた PageRank に基づくウェブ検索システム概念図を図 1 に示す。

一方、図書館の蔵書目録、新聞記事や特許公報といった文書を対象とする一般の情報検索の分野では、伝統的な手法として、検索対象文書と検索質問を多次元ベ

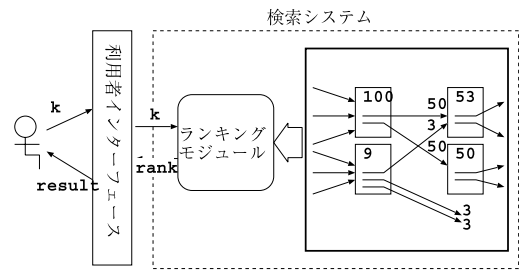


図 1 PageRank によるランキング手法
Fig. 1 Ranging system based on PageRank.

クトルで表現するベクトル空間モデル (VSM: Vector Space Model)⁷⁾ が広く利用されている。このモデルでは、検索質問と対象文書中の出現単語を多次元ベクトルで表し、そのコサイン尺度や内積から検索質問と文書の類似度を判定する。その類似度から得られる値を用いて文書の順位付けを行うことで、検索質問の意図を反映した検索結果を得ることができるが、VSM による情報検索システムの検索精度は十分とはいえない⁹⁾。VSM の検索精度を改善する代表的な手法として、適合性フィードバック (Relevance Feedback) がある。この手法では、得られた検索結果に対して、検索者が適合文書、不適合文書を検索システムに与えることで、検索精度の改善を行う。代表的なものとして、Rocchio の式⁸⁾ がある。

3. ウェブページ間相互評価モデルの提案

3.1 提案の背景と概要

VSM に代表される検索エンジンでは、検索キーワードとウェブページの本文との関連の強さに基づいて順位付けを行っている。この方式には、テキストの表記にともなうスパムの問題があり、ウェブページの本文を修正/工夫することで、ウェブ検索エンジンでの検索順位を上げることが比較的容易であった。この問題を解決するために、ハイパーリンク構造に基づいて順位付けを行う PageRank アルゴリズムが提案された。

PageRank アルゴリズムは参照されているリンク数により順位を計算するため、知名度の高いウェブサイトから順に内容を確認する検索意図に合致した方式である。すなわち、ウェブ空間の中で、他のウェブページ提供者より「このウェブページは重要であるのでリンクを張る」という評価があることを前提に、これに基づいて検索者に対する検索順位を計算する方法であるので、ウェブページの表記の工夫だけでは、検索順位をコントロールすることはできない。このため、上記のスパムの問題は解決されたが、検索者の検索意図をより正確に反映した検索を行うためには、下記の

問題が残っている．

- (1) 著名なサイト以外にも検索意図にマッチしたページが存在する場合、このページは下位の順位に位置づけられる．
- (2) 著名なサイトに含まれるページは、PageRankの順位リストに繰り返し表れる傾向がある．このとき、著名なサイトに含まれるページの中で、最もリンクの張られているページが検索者の求めるページとは限らず、著名サイトの中の検索者が求めるページが、リンク数が少ない場合、下位の順位に位置づけられる．
- (3) ウェブページの集団でリンクを意図的に相互に張り合うことにより順位を上げることができる．

上記の問題は、PageRankがウェブページ提供者のリンクの意図と、検索者の目的とするページの内容が必ずしも一致しないこと、著名なサイトのページの集合に対して、リンクが必ずしも適切なページに張られていないこと、ページのリンクを恣意的に張ることができるなどの理由により発生している．このため、検索者は、目的とするページを見付けるまでに、PageRankにより順位付けされたページの内容をチェックする負担が生じている．

この問題を解決するために、本論文ではPageRankアルゴリズムにより検索された上位のページを候補集合として、これらを対象に、ページの内容を反映する再ランキングを行う手法について述べる．この再ランキングでは、候補集合である上位のページの多くと、内容が一致するウェブページほど高い評価値を与える操作を行う．この操作により、検索者が必要とする内容と相関性の高いページをより上位にすることにより、検索者の負担を軽減することを目的とした、ウェブページ間相互評価モデルを提案する．

3.2 ウェブページ間相互評価モデル

ウェブページ間相互評価モデルは、ページの内容に基づくページ間の関係から、個々のページが互いに評価値を与えることでランキングを行うモデルである．

図2において、利用者の検索キーワード k に対して、適合したウェブページの集合を候補集合 C とする． C の要素であるウェブページ $wp(j)$ は、他の C の要素の検索キーワード k に対する適合度を自分を基準として計算するための評価基準関数 $E(j)$ を持つ． C の各ウェブページ $wp(i)$ は、他のすべてのウェブページ $wp(j)$ に対して、評価依頼 r をブロードキャストする．評価依頼 r を受信した $wp(j)$ は、評価基準関数 $E(j)$ に基づく評価値 v を実数で返す． $wp(i)$ は C のすべての $wp(j)$ からの評価値の加算平均を、検索キーワード

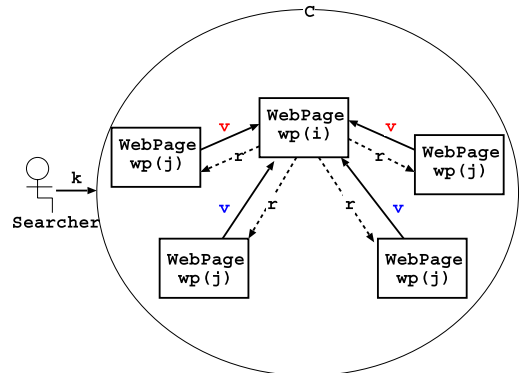


図2 ウェブページ間相互評価モデル
Fig. 2 Mutual evaluation model among webpages.

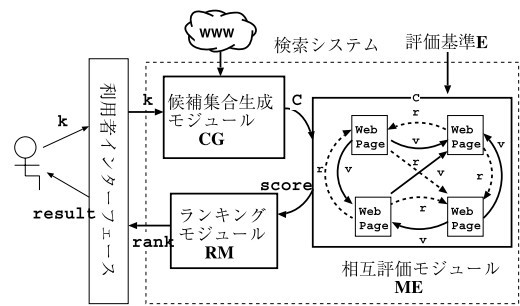


図3 提案手法に基づくウェブ検索システム
Fig. 3 Search engine based on the proposed model.

k に対する自分の評価とする．この評価値の値により、 k に対する $wp(i)$ のランキングを行う．

3.3 相互評価モデルに基づく検索システム

ウェブページ間の相互評価モデルに基づく検索システムを図3に示す．候補集合生成モジュールCGは、検索キーワード k に対して、ウェブ空間からウェブページの N 個をランキング対象の候補集合 C として選び出す機能である．相互評価モジュールMEは、 C の要素に対して相互評価を行い、評価値を決定する機能である．評価値はウェブページ間相互評価モデルと評価基準 E に基づき計算される．評価結果 $score$ は、 C の各ウェブページに対応する N 個の実数の評価値として、ランキングモジュールRMに渡される．RMは、 $score$ に基づき C を並べ替えた結果 $result$ を利用者インタフェースに渡す．

3.4 相互評価モデルによる検索精度への効果

ウェブページ間相互評価モデルでは、PageRankアルゴリズムをより一般化したランキングモデルである．すなわち、PageRankアルゴリズムでは図2に示すように、ウェブページ $wp(i)$ のランクは、他のウェブページからのリンク数に基づいて計算される．しかしながら、 $wp(i)$ の重要度をこのリンク数だけで判断

することは、前記の問題が発生する原因となる。ウェブページの間には、リンク以外にも様々な関係が存在し、この関係を基に互いに評価を行い、この評価の総合によりランクを決定することにより、PageRankのランキングの問題を改善することが期待できる。本論文では、このウェブページ間の相互評価の方式として、VSMに基づく、評価基準を採用することにより、より検索者の意図に近いページをランキングの上位に位置づけることができることを7章の実験と8章の考察で示している。

4. ウェブページ間相互評価によるウェブ検索システムの設計

4.1 候補集合生成モジュールCGとランキングモジュールRM

CGは、候補集合Cを生成するモジュールである。本提案モデルでは、個々のウェブページが、VSMを用いて、それぞれ自身の評価基準に基づいて他のウェブページを評価する特徴により、Cに属するウェブページは、検索キーワードに対して、重要度の高いウェブページの集合であることが望まれる。この重要度の高いウェブページの集合において、ウェブページ間で相互評価を行うことで、検索キーワードに内容が強く関連するウェブページを上位に順位付け、他の多くのウェブページとは内容の異なるウェブページを下位に順位付ける手法である。そこで、このCの条件を満たすために、CGには既存のウェブ検索エンジンの上位N件を使用する。現在、ウェブ検索エンジンの多くがハイパーリンク構造を利用したランキングを行っており、代表的な検索エンジンとして、goo, Excite, Infoseek, Googleなどがあげられる。本論文では、国内で最も利用者の多いGoogleをCGとして用いた。そして、重要度が高く、検索キーワードに対して関連が強いと考えられる、Googleの検索結果の上位200件をCとして、実験を行っている。またRMは、Cを評価結果scoreに基づき並べ替えるプログラムである。

4.2 相互評価モジュールMEの設計

MEは、Cを入力として、3.2節のモデルに基づいて配列scoreを戻す関数である。MEは内部に評価基準Eを持ち、Cの要素wp(i)に対して、評価関数vote[i]を生成する。評価基準Eのもとでのwp(i)に対するwp(j)の評価値は関数vote(E,i,j)で表される。MEは以下のような関数で表現される。

```
function ME(C)
  criteria E;
  eval(E,C);
```

```
for i 1 to N do
  score[i] 0;
  for j 1 to N do
    score[i] vote(E,i,j);
  end;
  score[i] sum[i]/N;
end
return score;
end function
```

ただし、NはCの要素数、eval(E,C)は、Cに対してvote(E,i,j)を構成するために必要な初期計算である。

5. VSMに基づく評価基準の実現

5.1 類似度による評価基準

wp(j)によるwp(i)への評価値vを決定するための評価基準の実現には様々な手法が考えられる。本論文では、評価基準として、ウェブページ間における文書内容の類似度を用いた。すなわち、wp(i)に評価値を与えるwp(j)は、自身の文書内容と類似するウェブページに対して良い評価値を与えるものとする。ここで、評価値の決定に類似度を用いる理由は、ウェブページwp(i)に対する評価を、wp(j)の内容を基準として与えるためである。wp(j)が、Googleの検索結果上位N件に含まれる重要度の高いウェブページであり、wp(j)に対して、wp(i)がその内容と類似しているならば、wp(i)は重要度の高いページであるとして、wp(j)は高い評価を与える。この相互評価を、Cに含まれるすべてのウェブページ間で行うことで、最も多くのウェブページから良い評価を受けたウェブページをより検索結果の上位に順位付ける。

文書間の類似度を求める代表的な方法として、ベクトル空間モデルVSMがある。VSMでは、文書を文書中に出現する個々の語の重みを成分とした多次元ベクトルで表現し、文書間のベクトルのコサイン尺度や内積によって類似度を求める。本論文では、このVSMに従って求めた文書間の類似度を用いてwp(j)によるwp(i)への評価値vを決定する。

VSMでは、以下の手順でwp(i)に対するwp(j)の類似度を計算する。

- (1) 各wp(i)に対して、tf-idf(Term Frequency Inverse Document Frequency)を用いてウェブページの中の語に重み付けを行い、各重みの値を成分とするベクトルV(i)を生成する。
- (2) vote(V,i,j)は内積V(i)V(j)に比例する値として計算される。

5.2 eval 関数の実現

前節の (1) の計算を行うための関数 eval(C) を以下に示す .

```
function eval(C)
    array df,tf;
    array-of-array V;
    df DF(C);
    for i 1 to N do
        tf TF(wp(i));
        V[i] tf-idf(tf,df);
    end
    return V;
```

ただし, 関数 DF(C) は, 候補集合 C のウェブページから, tf-idf に従って, 実数の配列 df (Document Frequency) を返す . また, 関数 TF(wp(i)) も同様に, tf-idf に従ってウェブページ wp(i) から, 実数の配列 tf (Term Frequency) を返す . そして, tf と df から, 関数 tf-idf によって wp(i) に含まれる語に対し tf-idf による重み付けを行い, その重みをベクトルの成分とした wp(i) の文書ベクトルを 1 次元の配列で生成し, これを V[i] の値とすることにより, 配列 V を計算し, 値として返す .

5.3 vote 関数の実現

関数 eval によって得られた, ウェブページのベクトル表現である評価基準 E を用いて, VSM に基づく評価関数 vote を実現する .

```
function vote(V,i,j)
    const T;
    t VSM(V[j],V[i]);
    rt t/360;
    if rt<=T then
        return 1;
    else
        return 0;
    endif
end function
```

ただし, T は経験的に定める閾値である . また, 関数 VSM(V[j],V[i]) は, 与えられた 2 つのベクトル V[i], V[j] から内積を求め, コサイン尺度を返す . wp(i) と wp(j) の類似度を表す rt は, [0,1] の実数値となる . 関数 eval は, rt が T 以下であるとき 1 を返し, rt が T より大きいとき 0 を返す .

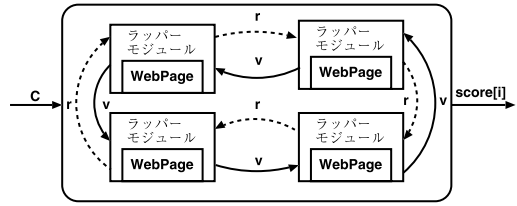


図 4 相互評価モジュール
Fig.4 CG module.

6. 検索システムの実装

6.1 ME の実装

図 3 の相互評価システムを 4 章のアルゴリズムに従って Java 言語を用いて PC に実装を行った . 本節では, 主要な機能である相互評価関数 ME の実装について述べる .

ME では, C の要素のウェブページ wp[i] が他のページを評価する vote プログラムは, 一般には, 各 wp[i] ごとに異なる . したがってシステムに与えられた評価基準 E に対して, 各 wp[i] ごとに eval(E,C) を用いて生成することが必要になる . したがって, 図 4 に示すように, C の要素である N 個のウェブページに対して, N 個のラップモジュールを生成することにより, アクティブなウェブページとして互いに評価しあう並行プロセスの実装を行った . ME は, ウェブページのラッピングプロセスを終了し, 評価基準の初期化プロセス eval を終了した後, 制御を各ラップモジュールに渡し, score[i] が完成すると, これをランキングモジュールに渡して終了する .

6.2 VSM 評価基準によるラッピングモジュールの実装

5 章で述べた VSM 評価基準に基づくラッピングモジュールの実装について説明する . この基準では, vote プログラムは各 wp[i] で一様であるため, 1 種類のラッピングモジュールだけを実装した . モジュールは Java のスレッドとして実装され, 図 4 の各スレッドモジュールは, 4.1 節で定義した ME の関数呼び出しは通信として実装されている .

7. 性能評価

本提案手法であるウェブページ間相互評価に基づく検索システムを実装し, 評価実験を行った . 評価実験ではまず, 実装したシステムを用いて, 検索時間に関する実験を行った . 次に, 検索時間の実験結果から, 実用的な時間内で検索結果を得ることができる候補集合 C のウェブページ数を定め, ウェブサーチエンジン Google を比較対象とし, 利用者の主観に基づく評価

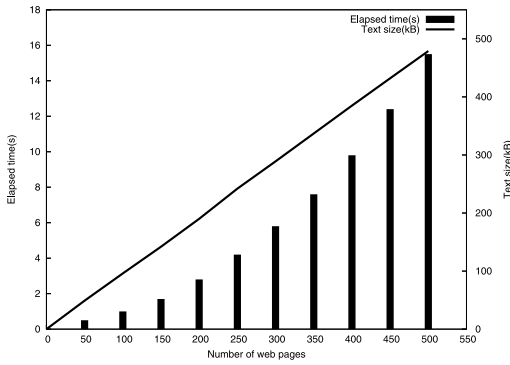


図 5 ME の計算時間

Fig. 5 Elapsed time of ME.

により求めた利用者順位に対し, Google の検索順位と提案システムの検索順位の, どちらがより利用者の意図に近い検索結果を示せたかについての評価を行った.

7.1 ME の計算時間

開発したプログラムを用いて, 本提案モデルのコア技術である, 相互評価モジュール ME の計算時間を計測した. 実験を行った環境を下記に示す.

- (1) CPU: Intel(R) Pentium(R) 4 CPU 2.80 GHz
- (2) Memory: 1,032,976 kB
- (3) OS: Linux 2.4.22

テストデータとして, キーワード (infoseek キーワードランキングの「総合」カテゴリ内上位 10 件のキーワード, 2003 年 3 月 24 日時点) を用いたときの相互評価モジュール ME の計算時間を図 5 に示す. ここで, 図中のウェブページ数は, 候補集合 C に含まれるウェブページ数を示す. また, 同時に測定に用いたウェブページのテキストサイズを示す.

本提案手法で検索を行った場合の時間計算量は, 候補集合 C の要素数 n に対して n^2 のオーダーである.

7.2 検索性能の評価方法

本論文では, 本検索システムによって得られた検索順位と, Google によって得られた検索順位とを比較し, どちらが, 利用者の主観で与えられた検索順位に近似するかで検索性能の評価を行った. 利用者の主観に基づく評価は, 関連コミュニティ群を発見する Companion+¹¹⁾ の評価実験にも採用されており, 本実験でもこの評価方法に基づいて, 以下の基準で評価実験を行った.

- (1) 評価実験で検索対象としたウェブページは, Google を用いて獲得した上位 200 件のウェブページである. この件数は, 利用者が結果を 1 分以内に知ることができるという条件を満たすことのできる候補集合のサイズを図 5 の結果よ

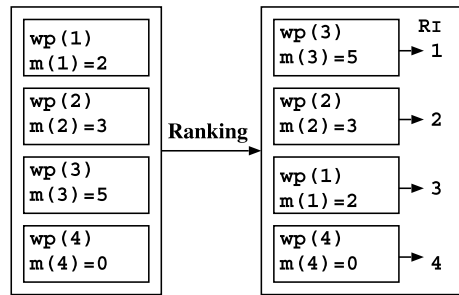


図 6 被検者順位 R_I の決定

Fig. 6 Example of R_I .

り求めたものである.

- (2) 獲得したウェブページを, 情報工学科に所属する学生 20 人を被験者として, アンケートフォームを通して与えられた検索キーワードに対して表示されるウェブページが, 検索結果としてどの程度適切であるかを 0 (悪い) から 5 (良い) の 6 段階で評価してもらった.
- (3) 使用する検索キーワードは, infoseek²⁾ で公開されている, infoseek キーワードランキングの「総合」カテゴリ内上位 10 件のキーワード (2003 年 3 月 24 日時点) を用いた.

被験者によって与えられた 0 から 5 の 6 段階の評価値を E_u とし, E_u を基に被験者順位 R_I を下記の手順に従って求めた.

- (1) infoseek のキーワードランキング上位 10 個を用いて, Google により得られた検索結果上位 200 件である候補集合 C のウェブページに対し, 被験者が, 評価値 $E_u = 0$ から $E_u = 5$ を与える.
- (2) ウェブページ wp(i) に対する 20 人の被験者の評価値 E_u の平均値を $m(i)$ とする. 図 6 に示すように, ウェブページを $m(i)$ によってランキングして得られた順位を被験者順位 R_I とする.
- (3) wp(i) の $m(i)$ が同値である場合は同順位とする.

したがって, ウェブページ wp(n) は, 下記に示す 5 つの値が割り当てられる.

- (1) 固有の識別番号 n
- (2) 被験者の評価値 E_u の平均値 $m(n)$
- (3) 被験者順位 $R_I^{(n)}$
- (4) Google による検索順位 $R_G^{(n)}$
- (5) 本検索システムによる検索順位 $R_P^{(n)}$

被験者順位 R_I を基準とし, Google によって得られた検索順位 R_G の示すばらつきを表す 2 乗平均を S_G , 本検索システムによって得られた検索順位 R_P の示すばらつきを表す 2 乗平均を S_P として, S_G と

S_P は、それぞれ下記の評価式で求める． N は評価対象とするウェブページ数である．

$$S_G = \frac{1}{N} \sum_{n=1}^N (R_G^{(n)} - R_I^{(n)})^2 \quad (1)$$

$$S_P = \frac{1}{N} \sum_{n=1}^N (R_P^{(n)} - R_I^{(n)})^2 \quad (2)$$

S_G と S_P は、被検者順位 R_I に検索結果 R_G, R_P が近づくほど、0 に近づく．以上から得られる S_G と S_P から、Google と本検索システムの検索性能比 P を次式により求める．

$$P = \frac{S_G}{S_P} \quad (3)$$

したがって、性能比 P が 1 より大きいとき、本検索システムが Google の検索結果よりも R_I に近い結果を示したことを表す．

7.3 実験結果

infoseek のキーワードランキング上位 10 個とは上から（無料，攻略，2 ちゃんねる，壁紙，ホテル，大阪，写真，東京，掲示板，ゲーム）である．検索キーワードの上位 5 個に対する P の値を表 1 に、6 位から 10 位までに対する P の値を表 2 に示す．

分析をさらに詳しく行うために、 C を 6 つのカテゴリ $C^{(0)} \sim C^{(5)}$ に分割した．

$$C^{(q)} = \{wp(i)|m(i) \text{ の四捨五入した値が } q \text{ である} \}$$

たとえば、 $C^{(5)}$ は被験者が平均的に最上位にランクしたウェブページの集合であり、 $C^{(0)}$ は被験者がキーワード k に対し最も望ましくないと評価したウェブページの集合である．各ランクのウェブページ集合 $C^{(q)}$ に対して、前節で述べたものと同じ評価法を行った．すなわち、 q をカテゴリの番号として、

$$S_G^{(q)} = \frac{1}{N^{(q)}} \sum_{E_u^{(q)} \in n} (R_G^{(n)} - R_I^{(n)})^2 \quad (4)$$

$$S_P^{(q)} = \frac{1}{N^{(q)}} \sum_{E_u^{(q)} \in n} (R_P^{(n)} - R_I^{(n)})^2 \quad (5)$$

$$P^{(q)} = \frac{S_G^{(q)}}{S_P^{(q)}} \quad (6)$$

さらに、 $N=20$ について、すなわち、Google の検索結果上位 20 件と本検索システムの検索結果上位 20 件に関する、各キーワードに対する性能比 $P_{[20]}$ の値を求めた．各キーワードに対する、 $P^{(0)} \sim P^{(5)}, P_{[20]}$

表 1 性能比較 (1)
Table 1 Performance comparison (1).

評価対象	検索キーワード				
	無料	攻略	2 ちゃんねる	壁紙	ホテル
P	1.18	1.09	1.24	0.94	1.13
$P^{(5)}$	2.13	0.53	0.06	1.04	0.52
$P^{(4)}$	1.21	1.33	1.38	0.70	0.94
$P^{(3)}$	1.09	0.92	1.24	1.03	1.15
$P^{(2)}$	0.75	1.03	1.44	0.75	0.85
$P^{(1)}$	1.22	1.36	1.58	1.10	2.91
$P^{(0)}$	3.42	1.10	1.32	2.60	1.62
$P_{[20]}$	1.73	1.34	4.27	3.34	2.26

表 2 性能比較 (2)
Table 2 Performance comparison (2).

評価対象	検索キーワード				
	大阪	写真	東京	掲示板	ゲーム
P	0.87	2.27	0.81	0.99	1.43
$P^{(5)}$	3.81	0.31	0.46	0.11	26.6
$P^{(4)}$	0.65	1.93	0.57	1.04	1.34
$P^{(3)}$	0.92	3.10	1.24	1.30	1.32
$P^{(2)}$	0.87	1.36	0.76	0.73	1.38
$P^{(1)}$	0.51	0.86	0.71	0.74	0.84
$P^{(0)}$	14.46	0.76	0.89	2.14	4.60
$P_{[20]}$	1.33	0.31	0.48	0.44	9.02

を表 1 および表 2 に示す．また、 P の値を求めるために用いた S_G, S_P の値を、表 3、表 4 に示す．

8. 考 察

性能実験の結果から、6 個の検索キーワード { 無料，攻略，2 ちゃんねる，ホテル，写真，ゲーム } で、Google よりも本検索システムの方がより被検者順位 R_P に近い検索結果が得られた．また、上位 1 位から 20 位の検索結果においては、7 個の検索キーワード { 無料，攻略，2 ちゃんねる，壁紙，ホテル，大阪，ゲーム } で、Google よりも期待順位に近い結果が得られた．しかしながら、下記に示す課題も見付かった．

8.1 汎用的なキーワード

今回、実験で用いた検索キーワードは、それ単独で用いると、非常に広範囲のウェブページを対象とするキーワードであった．そのため「東京」や「大阪」という検索キーワードを用いた場合、その地域で人気のテーマパークや、銀行、企業、また天気予報といった様々なコンテンツのウェブページが候補集合 C に存在した．このような候補集合に対して、今回の試作で用いた VSM による相互評価を行うと、ウェブページの内容が分散することで、互いに評価値 $v=0$ を示すウェブページが多く存在した．このような問題によって、被験者にとって重要と判定されるようなウェブページ

表 3 性能比較 (3)

Table 3 Performance comparison (3).

評価項目	検索キーワード (S_G/S_P)				
	無料	攻略	2ちゃんねる	壁紙	ホテル
S_G/S_P	6352/5514	6883/6290	6040/4871	4593/1377	10760/9482
$S_G^{(5)}/S_P^{(5)}$	12116/5683	7927/15001	900/13953	5967/5739	1305/2534
$S_G^{(4)}/S_P^{(4)}$	9788/8072	12071/9101	7641/5542	7786/11201	8269/8770
$S_G^{(3)}/S_P^{(3)}$	6830/6286	4829/5263	8298/6711	6894/6721	11038/9625
$S_G^{(2)}/S_P^{(2)}$	2308/3087	3227/3146	3520/2450	3177/3645	12334/14438
$S_G^{(1)}/S_P^{(1)}$	6568/5366	6321/4664	4156/2627	8990/8156	13282/4562
$S_G^{(0)}/S_P^{(0)}$	11892/3480	8510/7746	6513/4941	9915/3815	14733/9095
$S_{G[20]}/S_{P[20]}$	6389/3699	6075/4517	6392/1489	4593/1377	3383/1497

表 4 性能比較 (4)

Table 4 Performance comparison (4).

評価項目	検索キーワード (S_G/S_P)				
	大阪	写真	東京	掲示板	ゲーム
S_G/S_P	5936/6840	6874/3024	5644/6947	7896/7907	5877/4112
$S_G^{(5)}/S_P^{(5)}$	4416/1158	240/769	2316/5028	1226/11487	18849/708
$S_G^{(4)}/S_P^{(4)}$	6479/9997	9559/4947	7722/13522	12887/12403	6557/4883
$S_G^{(3)}/S_P^{(3)}$	7852/8504	7622/2460	6584/5303	3661/2820	7175/5422
$S_G^{(2)}/S_P^{(2)}$	3177/3645	3631/2665	2692/3543	6157/8463	2854/2065
$S_G^{(1)}/S_P^{(1)}$	4127/8108	2773/3207	7145/10050	5577/7564	2754/3298
$S_G^{(0)}/S_P^{(0)}$	21866/1512	1223/1613	17319/19368	5120/2388	8757/1903
$S_{G[20]}/S_{P[20]}$	5953/4482	871/2815	2514/5213	3913/8835	2839/314

でも、検索結果の下位にランキングされるウェブページがあった。

8.2 マルチメディアコンテンツ

ウェブページの大部分が画像により構成されていたり、Flash³⁾などを用いたマルチメディアコンテンツに対して、本検索システムでは、相互評価を行うのに十分なテキストを取得することができず、適切に相互評価を行うことができなかった。代表的な例としては、検索キーワード「掲示板」において、被験者の多くが最も高い評価値 $E_u=5$ を与えたウェブページは、2ちゃんねる掲示板のエントランスページであった。しかしながら、このページのほとんどは画像で構成されており、取得できるテキストはごくわずかであった。このエントランスページは、Googleの検索結果では、上位に出現するが、本検索システムでは、最下位にランクされてしまった。このようなマルチメディアコンテンツを扱う際に生じる問題は、他の検索キーワードでも全般的に生じており、 S_P を大きくする要因となった。

8.3 追加実験

本検索システムでは、ウェブページの内容が分散するような汎用的な検索キーワードを用いた場合、また画像やFlashを中心に構成されたウェブページに対して、適切な相互評価が困難であった。そこで、性能評価で用いた検索キーワードが、広く意味を解釈できる

表 5 「Java, 学習」「LAN, 構築」

Table 5 Additional experiment with “Java, study” and “LAN, structure”.

評価対象	検索キーワード	
	Java, 学習	LAN, 構築
P	2.25	2.22
$P^{(5)}$	12.68	7.63
$P^{(4)}$	2.27	3.16
$P^{(3)}$	2.11	1.79
$P^{(2)}$	3.23	1.52
$P^{(1)}$	1.85	2.76
$P^{(0)}$	1.70	5.14
$P_{[20]}$	1.58	2.52

汎用的なキーワードであったことを考慮し、よりコンテンツの対象を限定するような検索キーワードを用いた追加実験を行った。追加実験として用いた検索キーワードは、「Java 学習」「LAN 構築」のように2つのキーワードを組み合わせたものを使用した。この2つのキーワードに対して、同様の手順で実験を行った結果を、表5に示す。

表5に示す検索キーワードを用いることで、候補集合Cのウェブページは、その内容が検索キーワードに関連したページが多く、また、画像で構成されるウェブページ、マルチメディアコンテンツが比較的少なかった。このようなウェブページを対象として本検

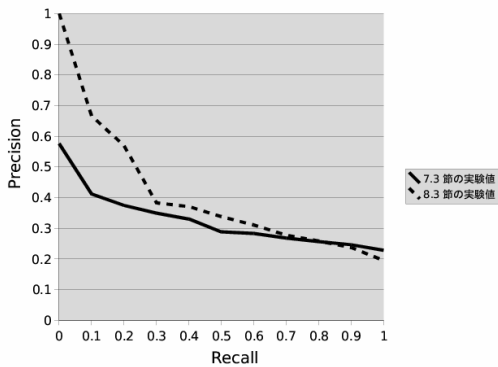


図7 本提案モデルの再現率・適合率曲線

Fig. 7 Recall-precision curve for proposed model.

システムで検索した場合、表5で示すように、すべての評価対象で、Googleよりも R_I に近い検索結果を得ることが確認できた。

8.4 検索性能

図7に、本提案手法による再現率・適合率曲線を示す。図中の実線は、7.3節の表2および表3を得るために用いた実験結果を基に、10個の検索キーワードに対して得られた実験結果それぞれに対して、再現率0から1.0までの11点での適合率を求め、その適合率を平均した値を示している。また、適合文書の判定は、被験者によって与えられた0から5の評価値のうち、4以上のものを適合文書、3以下のものを不適合文書としている。また、図中の点線で示した値は、8.3節の表4を得るために利用した実験結果を基に、上記と同様の手順で、再現率11点における適合率を求め、その適合率を平均した値を示している。

図7では、7.2節で使用した10個の検索キーワードがいずれも、広く意味がとれるキーワードを用いたことで、候補集合中のウェブページの内容がばらつき、検索者の意図しないウェブページも含まれたこと、またマルチメディアコンテンツや画像などが多く含まれ十分なテキストを獲得できなかったことが原因となって、全体的に低い検索精度を示している。一方、8.3節で行った追加実験では、検索キーワードを複数用いて検索者の検索意図がより的確に検索結果に反映されることで、候補集合中のウェブページの内容のばらつきが少なく、また解説記事に代表されるようにテキストが十分に獲得できるような場合、本提案手法が適切に機能することで、全体的に高い検索精度を示す特徴が見られる。

8.5 検索時間の問題

7.1節で示すように、本提案手法の中心技術である、相互評価モジュールMEの計算時間は、本論文が想定

する候補集合のサイズ、ウェブページ200個に対し、2.8秒であり、これは実用的な計算時間であると考えられる。しかしながら、本提案手法では、余弦尺度の計算を、対象とするウェブページ数を N としたとき、 $N \times (N - 1)$ 回実行するため、ウェブページ数の増加にともない、余弦尺度の計算回数も n^2 のオーダーで増加する。現在は、文書ベクトルの次元数が約3000次元であり、これが原因となって余弦値の計算量が増加している。文書ベクトルの次元数をより低く抑えることで、MEの計算時間をより減少するなどの改善方法を考慮する必要がある。

8.6 クラスタリングとランキングの議論

本論文では、Googleの検索結果の上位200件に対して、ウェブページ間相互評価を行うことで、ウェブページをランキングする手法について述べた。一方、検索結果に対して意味解析を行いクラスタリングを行うことが、検索者にとって効果的な検索結果となることがある。たとえば「スポーツ」や「政治」といった、広い分野を包括するようなキーワードによって得られる検索結果に対してクラスタリングを行うことで、検索者は分類されたカテゴリを選択することで、自分の求めるウェブページにいち早くたどり着くことができる。また、検索結果の全体をカテゴリとして把握することができ、検索者にとって重要なウェブページを見落とすことが少なくなる利点が考えられる。

しかしながら、検索者が複数のキーワードを用いて、限られた分野の中から有益な情報を含むウェブページを、即座に求めるとき、ランキングによって検索結果を示す方が、検索者にとっては効果的であると考えられる。ランキングによって検索者に結果を示すことは、クラスタリングによるカテゴリの中から探すよりも、上位数件程度の中から目的とするページを見付けられるときは、即座にそのウェブページにアクセスができる。クラスタリングとランキングによる検索結果の提示は、いずれも重要な技術であり、適切に使い分けが必要がある。

8.7 提案手法の効果対候補集合のサイズ

本提案手法は、候補集合中のウェブページが、互いに文書間の類似度を計算することで、他ページの評価値を決定している。このとき、ウェブページ $w_p(j)$ は、自身の本文と類似した本文のウェブページ $w_p(i)$ に対して高い評価値を与える。そのため、候補集合中のウェブページ $w_p(j)$ はそれぞれ、検索キーワードに対して適合度の高いウェブページである必要がある。この条件を満たすために、本論文では、Googleの上位200件が適合度の高いウェブページの集合であるとして、

候補集合に利用している．そのため，200 件以上のより多くのウェブページを候補集合とすると，候補集合には適合度の低いウェブページが多く含まれ，そのようなウェブページが，自身と類似するウェブページに対しても高い評価値を与える現象が生じるために，本提案手法による検索精度は低下すると考えられる．

したがって，より多くのウェブページを候補集合とし，また検索性能を維持するためには，適合度の高いウェブページの条件を適切に設定し，その条件を満たすウェブページが示す評価値に対して，高い重みを与える方法が，解決策の 1 つとして考えられる．

9. ま と め

ハイパーリンクの参照関係を利用したランキング手法である PageRank には，リンクの意図が十分に考慮されていないことによる問題が指摘されている．本論文では，ウェブページの内容の評価から，推薦を相互に行うウェブページ間相互評価によるランキング手法を提案した．次にこの手法に基づくウェブページ検索システムの設計と試作実験を行った．本論文での試作では，相互評価法として VSM 法を採用し，候補集合として Google の上位 200 件を用いることにした．

評価実験では，200 個前後が実用的に用いることができる最大ページ数であること，また，本提案手法が，ウェブページの本文を基準に他のウェブページに評価を与えるため，ウェブページ自体が適合度の高いページである必要があることから，ウェブページを 200 個に絞るためのフィルタとして，Google の検索結果上位 200 件を用いた．infoseek の検索キーワードランキング上位 10 個を用いて検索精度の評価を行った結果，半数以上の 6 個のキーワードで，Google よりも検索精度が向上し，またよりウェブページの対象を制限するような複数の検索キーワードを組み合わせて用いた場合には，Google の検索結果よりも高い検索精度を示すことが確認できた．今回の実験では，相互評価対象ページを絞り込むためのフィルタとして Google の検索結果を用い，相互評価法として VSM 法を用いたが，より本提案手法に適したフィルタの設定，および，精度の高い相互評価方式の設計が今後の課題である．

参 考 文 献

- 1) Google. <http://www.google.com/>
- 2) infoseek. <http://www.infoseek.co.jp/>
- 3) Macromedia. <http://www.macromedia.com/>
- 4) Dean, J. and Henzinger, M.R.: Finding related pages in the World Wide Web, *The 8th Inter-*

- national World Wide Web Conference* (1999).
- 5) Bharat, K., et al.: Improved algorithms for topic distillation in a hyperlinked environment, *Proc. 21st ACM SIGIR Conf.*, pp.104–111 (1998).
- 6) Haveliwala, T.H.: Topic-sensitive pagerank. online manuscript.
<http://www.stanford.edu/taherh/papers/topic-sensitive-pagerank-tkde.pdf>
- 7) Salton, G. and McGill: *Introduction to Modern Information Retrieval*, McGraw-Hill Book Company (1983).
- 8) Schapire, R.E., Singer, Y. and Singhal, A.: Boosting and rocchio applied to text filtering, *Proc. SIGIR '98*, pp.215–223 (1998).
- 9) 柘植 覚, 獅々堀正幹, 黒岩眞吾, 北 研二: サポートベクタマシンによる適合性フィードバックを用いた情報検索, 情報処理学会論文誌, Vol.44, No.1, pp.59–67 (2003).
- 10) 風間一洋, 原田昌紀: Web サーチエンジン技術の高度化, 人工知能学会誌, Vol.16, No.4, pp.503–508 (2001).
- 11) 豊田正史: Www における関連コミュニティ群の発見, 情報処理学会研究会報告 2000-DBS-122-40, pp.307–314 (2000).
- 12) 北 研二, 津田和彦, 獅子堀正幹: 情報検索アルゴリズム, 共立出版株式会社 (2002).
- 13) 有吉祐介, 福島俊一: 目的および個人に特化したサーチエンジンの開発, 人工知能学会誌, Vol.16, No.4, pp.520–524 (2001).

(平成 16 年 5 月 20 日受付)

(平成 16 年 11 月 1 日採録)



荒谷 寛和

1977 年生．2002 年千葉工業大学大学院工学研究科博士前期課程修了．現在，同大学院同研究科博士後期課程在学中．情報検索，エージェント，セマンティックウェブに興味を持つ．



藤田 茂 (正会員)

1968 年生．1997 年千葉工業大学大学院博士後期課程情報工学専攻期間満了退学．現在同大学情報工学科講師．博士 (工学)．エージェント分散処理システムに興味を持つ．



菅原 研次（正会員）

1950年生．1980年東北大学大学院博士課程中退．同年千葉工業大学助手．現在同大学情報ネットワーク学科教授．工学博士．エージェント，サイバー社会に興味を持つ．
