Webページ移動先探索処理の効率化

飯田　敏成† 　澤　菜津美† 　森嶋　厚行†† 　杉本　重雄†† 　北川　博之†††

† 筑波大学大学院 図書館情報メディア研究科　〒 305–8550 茨城県つくば市春日 1–2
†† 筑波大学大学院 図書館情報メディア研究科/知的コミュニティ基盤研究センター
〒 305–8550 茨城県つくば市春日 1–2
††† 筑波大学大学院 システム情報工学研究科 〒 305–8573 茨城県つくば市天王台 1–1–1
E-mail: †{toshi,sawa,mori,sugimoto}@slis.tsukuba.ac.jp, ††kitagawa@cs.tsukuba.ac.jp

あらまし　今日，WWW には多くのリンク切れが存在している．その一因は，Web ページの移動である．しかし，Web ページの移動先探索のための手法は確立されていない．論理的には WWW 全体を探索すれば移動先の発見は可能であるが，現実的な解ではない．我々はこれまで，リンク切れを引き起こした Web ページの移動に対して，その移動先を自動的に探索するシステムを開発してきた．我々のシステムでは,「移動先がありそうな場所」に対して局所的に探索を行うアプローチを用いており，実験の結果，Web 検索エンジンのみを用いた移動先の探索と比べ，高い発見率で移動先を発見できることが分かった．本論文では，我々のシステムによる移動先ページ発見手法に関する議論を一歩進め，高い発見率を維持したまま，移動先ページの探索に必要なページアクセス数を大幅に削減できる手法を提案する．
キーワード　WWW，リンク切れ，一貫性維持，情報検索

Efficient Search for Moved Web Pages

Toshinari IIDA†, Natsumi SAWA†, Atsuyuki MORISHIMA††, Shigeo SUGIMOTO††, and Hiroyuki KITAGAWA†††

† Grad. Sch. of Library, Information and Media Studies, Univ. of Tsukuba. 1–2 Kasuga, Tsukuba, 305–8550 Japan
†† Grad. Sch. of Library, Information and Media Studies/Research Center for Knowledge Communities, Univ. of Tsukuba. 1–2 Kasuga, Tsukuba, 305–8550 Japan
††† Grad. Sch. of Sys. and Info. Eng., Univ. of Tsukuba. 1–1–1 Tennohdai, Tsukuba, 305–8573 Japan
E-mail: †{toshi,sawa,mori,sugimoto}@slis.tsukuba.ac.jp, ††kitagawa@cs.tsukuba.ac.jp

Abstract　Today, the WWW suffers from a lot of broken links, partly because of moved Web pages. And there is no established solution to the problem of finding moved Web pages. Logically, we can find new locations of the moved Web pages if we explore the whole Web, but it is not a realistic solution. We developed a system for the automatic discovery of new locations of the moved Web pages that caused broken links. The system takes an approach of locally exploring limited spots where the moved Web pages are likely located. Our experimental results have shown that the system gives higher precision compared to the search only with Web search engines. This paper proposes a method that can dramatically reduce the number of page accesses required for the discovery of new locations of the moved Web pages, while still keeping the high precision.
Key words　WWW, Broken Link, Integrity Maintenance, Information Retrieval

1. はじめに

近年，World Wide Web (以下 Web) は社会における重要なメディアの一つとして大きな役割を果たしている．Web の特徴の一つとして分散管理が行われていることが挙げられる．この特徴は，Web を便利なツールとする一方で，Web コンテンツの一貫性の維持を困難としている要因でもある．ここでいう Web コンテンツの一貫性の維持とは，分散管理されているコンテンツ間の関係を，当初意図されていた状態のままに維持する事である．例えば，(1) あるページからリンクされている先のページの内容が当初意図していたとおりである，(2) あるページのコンテンツと別に管理されているページのコンテンツが一致している，等がある．本論文では Web リンクの一貫性の維持の問題のうち，特にリンク切れへの対処の問題に焦点を当てる．Web の

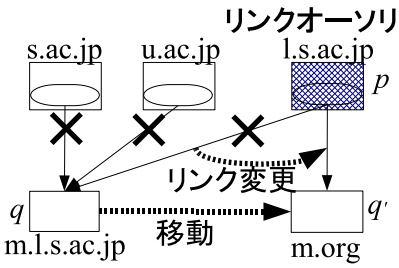

図 1　リンクオーソリティ

リンク切れは，以前から重要な問題として認識されてきた．例えば，1998 年のジョージア工科大学によるアンケート [7] では，利用者から見た Web の問題点として，リンク切れが第 3 位に挙げられている．また，2003 年の Science の記事 [15] でも，Web のリンク切れが依然重要な問題であることが指摘されている．

この問題を解決するため，我々はこれまで，ページの移動によって生じたリンク切れを対象として，Web ページの移動先を自動発見し，ページ移動によるリンク切れの修正を支援するシステム (WISH システムと呼ぶ) を開発し [12] [13]，実験や公開(注1)を行ってきた．移動先の探索は，移動前のページのコンテンツや，移動前の場所に関する情報などを用いて行われる．

本システムの特徴は，Google 等の Web 検索エンジンによるページ検索とは異なる方法も用いてページの移動先発見を行う事である．一般に，Web 検索エンジンは，あらかじめ構築したインデックスを利用して必要なページの検索を行う．しかし，ページの移動先を発見するためには，移動先ページがインデックスされている必要があるため，このアプローチは次の点で不十分である．(1) 移動後でなければ移動先ページにインデックスを貼ることができないため，移動先ページがインデックスされるまでに時間を要することがある．(2) マイナーなページなど，ページによってはインデックスされない事がある．

我々が着目した点は，検索エンジンを利用せずとも，人間はページの移動先を迅速に発見できることが多い事である．人間がページの移動先を発見できる理由は，ページの移動先探索においては「移動先がありそうな場所」に偏りが存在するからである．これらの場所を計算し優先して探索することにより，人間は比較的短時間でページの移動先を発見することができる．この性質を利用して，WISH システムでは，リンク切れ発見後に「移動先がありそうな場所」を計算して探索を行うアプローチが用いられる．この「移動先がありそうな場所」の計算には，リンクオーソリティ [3] などを利用する．リンクオーソリティとは，リンク先のページが移動したときにリンクを確実に変更するページのことを指す．例えば，ある Web ページ p が，別の Web ページ q へのリンクを持っていたとする．「q が q' に移動したとき，p の中の q へのリンクを q' に確実に変更するようなページ p」をリンクオーソリティであると我々は定義している (図 1)．実験の結果，Web 検索エンジンのみを用いた移動先の探索(注2)と比べ，大幅に高い発見率で移動先ページを発見できることが分かった．

本論文では，WISH システムによる移動先ページ発見手法に関する議論を一歩進め，高い発見率を維持したまま，移動先ページの探索に必要なページアクセス数をいかに削減できるかについて議論する．具体的には，移動先ページ発見に関するこれまでの実験から得られた結果を利用し，高い発見率を維持したまま，これまでよりもページアクセス数を大幅に削減できることを示す．

本稿の構成は次の通りである．まず，2 章で関連研究について説明する．3 章では，WISH システムの概要，およびこれまでの実験で得られた移動先ページ発見に関する性質について説明する．4 章では，3 章で説明した性質に基づいた，効率のよい移動先探索について説明する．5 章では，これらの探索手段に関する実験結果を示し，提案手法が有効であることを示す．6 章はまとめと今後の課題である．

2. 関 連 研 究

リンク切れの問題が深刻な問題として認識され [1]，その修復のためのアプローチについても研究が進められてきているが [6]，リンク切れ修復のためのソフトウェアによる支援技術はまだ未成熟である．現在，既に実用レベルのものとしては，リンク切れを発見し報告するためのソフトウェア [19] がある．このようなソフトウェアが数多く存在することから，リンク切れ問題の重要性が認識されていることは間違いない．また，管理しているリンク集にリンク切れが生じると，リンク先の Web サイトの管理者に電子メールを送信するような Web サイトも存在する [8]．以上のように，リンク切れを発見し報告するソフトウェアは数多く存在するが，その後の処理は利用者に任されており，リンク切れの自動修正もしくは修正支援システムはまだ実用には至っていない．

研究レベルでは，Web のリンク切れの問題に対処するため，これまで様々な手法が提案されてきた．Peridot は IBM によって開発されたソフトウエアツールであり，彼らが特許出願した手法 [4] [5] を利用している．Peridot は WISH システムと同様に，リンク切れした Web リンクに対して移動先を発見しようと試みる．具体的には，Peridot は各 Web ページのコンテンツから求められる**フィンガープリント** (*fingerprints*) と呼ばれる情報を用いて，ページの移動先らしさを計算する．したがって，この手法では，移動先候補となる Web ページのフィンガープリントがあらかじめデータベースに格納されていなければならない．Peridot の他にも，Phelps と Wilensky は**レキシカルシグネチャ** (*lexical signatures*) を利用して移動ページを発見することを提案している [14]．これは，各 Web ページごとに用意された小さなキーワード集合であって，インデックスサーバに投入したときにその Web ページが高い確率で上位に来るよう注意深く選ばれたものである．

以上の Peridot とレキシカルシグネチャによるアプローチでは，ページの移動先を発見するために，移動先 Web ページの

(注1)：http://wish.slis.tsukuba.ac.jp/LIM-RO.html.

(注2)：WISH システムでは検索エンジンを用いた検索に加えて，ありそうな場所の探索を行う．

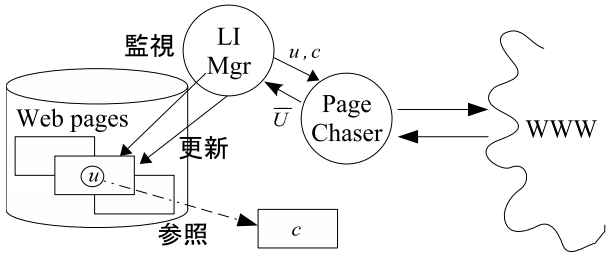

図 2　WISH システム

情報が，あらかじめ何らかのデータベースに格納されている必要がある．

これに対し，我々の開発している WISH システム [12] [9] では,「移動した Web ページがどこに存在していそうであるか」に関するヒューリスティクスを利用して，実際の Web 中からページの移動先を素早く発見することに焦点を当てており，Web 検索エンジンにインデックスされていなくても，移動先と推測されるページを収集する技術の開発を行ってきた．しかし，これまではその収集をどこまで行えばよいのかに関する議論が不十分であった．本研究は，その議論を行うものである．

3. WISH システムの概要と本論文で扱う問題

3.1 WISH システムのアーキテクチャ

図 2 は WISH システムのアーキテクチャである. 処理の流れは次の通りである．あらかじめ，ユーザがシステムに監視対象とするリンクを登録する (単純化のため，ここではただ一つのリンク u とする)．監視モジュール (LI-Manager) は，登録された u のページコンテンツのキャッシュ c，および u から別のページにリダイレクトが行われている場合にはリダイレクト先の URL ru を保存し，監視を開始する．u がリンク切れとなるまでの間，u のページコンテンツのキャッシュ c の取得を定期的に行う．また，並行して u のリンクオーソリティの収集を行う．リンク切れが発見されると，ページ探索モジュール (Page Chaser) が，移動先の探索を始める．探索の結果は「ページの移動先らしさ」を表すスコアでランキングされた URL のリスト $\bar{U}$ であり，LI-Manager に返される．最後に LI-Manager は，指定されたポリシーに基づき，結果 $\bar{U}$ を利用する．指定されたポリシーとは，例えば，スコアの閾値によりリンク切れを自動修正したり，移動先の候補をメールで利用者に連絡したりする．

3.2 移動先発見アルゴリズム概要

u の移動先の発見は，次の 2 段階で行われる．

候補収集フェーズ: クローラなど各種探索手段を利用して移動元ページ u の移動先候補となる URL の集合 U を収集する．具体的には，候補収集のために組み込んである 10 の手法 $M_i (i = 1 \ldots 10)$ を利用して，それぞれ候補集合 U_i を計算し，それらの和集合をとることにより U を求める (図 3 の 2-9 行目)．手法によって程度は異なるが，おおざっぱに言って，各 M_i は「**移動先でありそうな順**」に移動先候補 URL u_j を返すように設計されている [16].

ランキングフェーズ: U 中の各 URL に「移動先らしさ」を表すスコアをつけ，ランキングを行った結果 $\bar{U}$ を計算する (図 3 の 11-12 行目)．ランキングフェーズにおけるスコアの計算は，キャッシュ c やリダイレクト先 ru の情報だけでなく，移動元のページの位置情報や，他のページからのリンク情報などを利用して行われる．この詳細については文献 [12] [2] にある．

```
1   //候補収集フェーズ
2   for each 候補収集手法 M_i {
3     U_i = {};
4     while (M_i(u).hasNext()) {
5       u_j = M_i(u).next();
6       U_i = U_i ∪ {u_j};
7     }
8   }
9   U = ⋃_i U_i;

10  //ランキングフェーズ
11  compute score_j for each u_j ∈ U
12  return u_js with the top 3 scores.
```

図 3　移動先発見アルゴリズム

3.3 本論文で扱う問題

本稿では，候補収集フェーズにおいて，どれだけ少ないページアクセス数でページの移動先の正解が U に含まれるか，を議論する．このページアクセス数が少ないほど，ページの移動先を発見するためのコストが少なくてすむ事になる．この候補収集フェーズの問題はランキングフェーズにおけるランキングの問題とは完全に独立していることに注意していただきたい．したがって，本稿ではランキングについては議論しない．

より問題を具体化すると次のようになる．現在の WISH のアルゴリズムでは図 3 の 4-7 行目のように，各手法 M_i で収集できる候補ページが無くなるまで，移動先でありそうな順に全ての候補を数え上げていく．どれだけの候補ページを収集するかは 5 章で説明するように各 M_i 毎に規定されているが，いつ打ち切ればよいかが自明でないため，現システムでは相当数のページにアクセスを行っている．もし，これを途中で打ち切ることが出来れば，より少ないページアクセスですむことになる．問題は，ページアクセスを打ち切るための基準を決められるかどうかである．その議論に進む前に，次節でこれまでの実験から分かったいくつかの事実について説明する．

3.4 移動先ページ発見に関する性質

これまでの実験から下記の事が明らかになった．

性質 1. 各 M_i が出力する候補集合 U_i は，移動元ページ u とある程度類似しているごく少数のページと，全く類似していない大多数のページを含んでいる．図 4 は，あるページの移動先候補集合 U に含まれる各ページについて，u との類似度に関する度数分布である．類似度は，TF に基づくコサイン尺度を用いている．478 ページ中，類似度が 0.5 以上のものは 7 つ，つまりたった 1.5%しか無いことが分かる．

性質 2. 移動先ページは，高い確率で移動元ページ u と類似している (ただし，全て移動先ページが類似度が高いわけではな

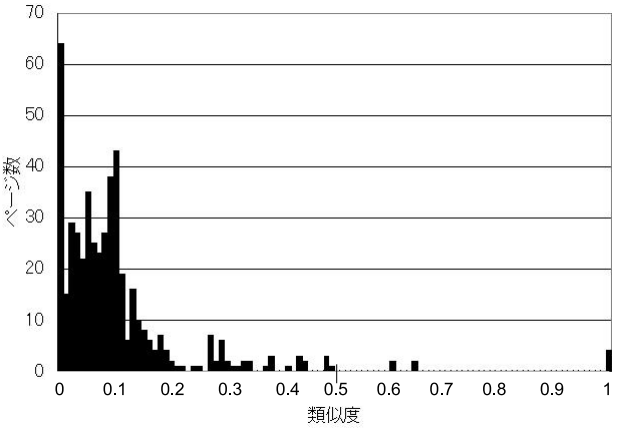

図 4　あるページの移動先候補集合 U に含まれる各ページの u との類似度に関する度数分布

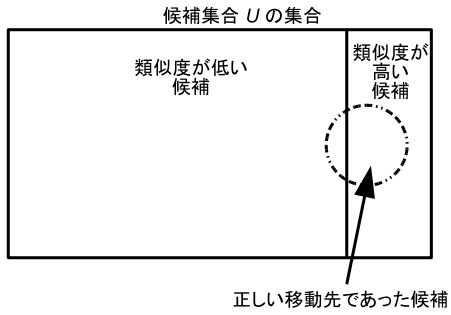

図 5　候補集合 U の集合の内訳模式図

い). 図 5 はこの事実を模式的に表したものである．候補集合のほとんどは類似度が低い．一方，移動先は高い確率で類似度が高いページである．

4. 候補収集打ち切り判定の提案

3.4 節で説明した性質 1 と 2，および,「各 M_i は移動先でありそうな順に候補 u_j を列挙する」という WISH システムの動作から，次が成立することが推測される．

性質 3. 移動元ページ u とある程度高い類似度を持つ候補 u_j が候補として初めて出現したとき，u_j が移動先である可能性が高い．かつ，もし u_j が移動先で無かった場合は，それまでに列挙された候補の中に移動先が既に入っている可能性が高い．

この性質 3 が成立するとすると，候補の収集を打ち切る基準として「移動元ページ u とある程度以上高い類似度を持つページ u_j が候補として初めて列挙された時」を利用すればよいことになる．ある程度の類似度を t として表した場合，候補収集フェーズのアルゴリズムを図 6 のように変更すればよい．変更点は 7 行目の追加のみである．

この性質 3 の導出に当たっては，性質 1 で述べている「ある程度類似度が高いページが非常に少ない」ことが重要である．候補収集のプロセスにおいて類似度の高いページが均等に現れると仮定すると，類似度が高いページが少ない場合と多い場合では図 7(a) と (b) のようになる．(a) の場合は，最初に現れた類似度の高い P_1 と 2 番目の P_2 の間隔が広いため，各 M_i は移動先でありそうな順に候補 u_j を列挙するという動作から，P_2 の「移動先らしさ」は大幅に低くなる．一方，(b) の場合，P_1 と P_2 の間隔が狭いため，P_1 と P_2 の間の「移動先らしさ」の違いが小さくなり，P_2 を候補に含めない根拠が弱くなる．我々は，以上の性質 3 が成立するとの仮説を立て，次章の実験によって検証を行う．

t: 閾値とする類似度

```
1    //候補収集フェーズ
2    for each 候補収集手法 M_i {
3      U_i = {};
4      while (M_i(u).hasNext()) {
5        u_j = M_i(u).next();
6        U_i = U_i ∪ {u_j};
7        if (sim(u_j, u) ≧ t) break;
8      }
9    }
```

図 6　類似度の閾値を用いた候補列挙の打ち切り

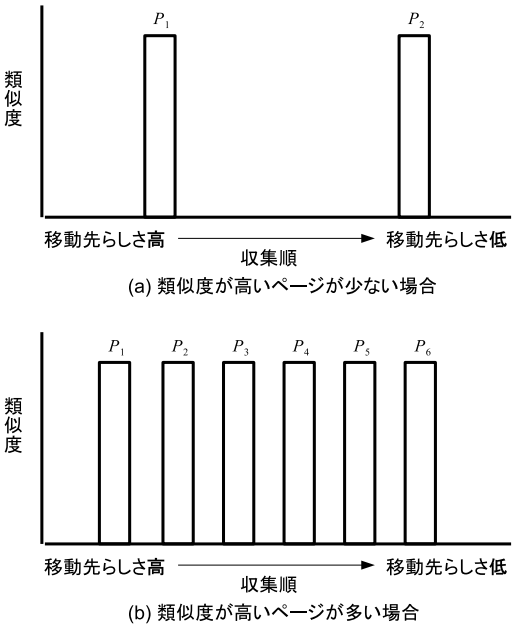

図 7　類似度が高いページが多い場合と少ない場合の比較

5. 実　　験

本実験では，実 Web 環境において，4 章の性質 3 を用いた候補収集の打ち切りによって，移動先ページの発見率を下げることなくページアクセス数を削減可能かどうかの検証を行う．実験の説明に入る前に，まず，各候補収集手法 M_i を簡単に説明する．

5.1 具体的な移動先候補収集手法

本実験では，手法 M_1〜M_{10} の 10 種類の手法を用いて移動先候補の収集を行う．移動先候補の収集方法は様々であるが，下記の点が共通である．(1) 手法によって程度は異なるが，収集されたページの上位 (最初の方) に，移動先ページである可

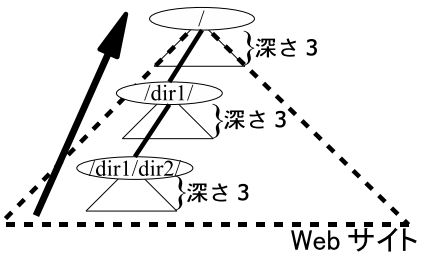

図 8　手法 M_6 によるサイト内探索

能性が高いページが存在する．(2) 各手法ごとに，収集ページの終了条件が決まっている．各手法の概要を次に示す．詳細は [12] [2] [16] に記載されている．

手法 M_1: 保存していたキャッシュ c (3 章参照) から TFIDF の値が上位のキーワードを 10 件抽出し，Google で検索する．検索結果の上位 10 件を順に収集する．結果が存在しない場合 (検索結果が 0 件だった場合) は，キーワードを一つずつ減らして検索する．

手法 M_2: M_1 と同様であるが，Yahoo 検索を利用する．

手法 M_3: M_1 と同様であるが，MSN 検索を利用する．検索結果の上位 50 件を順に収集する．Google や Yahoo と収集数が異なる理由は，API を利用する際にデフォルトで収集できる数が異なるからである．

手法 M_4: キャッシュ c の title タグに書かれた文字列を Yahoo で検索する．検索結果の上位 10 件を順に収集する．

手法 M_5: M_4 と同様であるが，MSN で検索を利用する．検索結果の上位 50 件を順に収集する．

手法 M_6: 移動元ページ u が存在した Web サイト内をクローリングして収集する．その際，移動元ページ u が存在した場所の周辺，およびサイトルートから u にたどる経路の周辺に移動先が存在しやすいとの考えに基づき，図 8 のように，u から開始し，u の URL を構成する下位ディレクトリから上位ディレクトリに向かってディレクトリ構造を辿ったときの各ディレクトリを起点として，それぞれのディレクトリから深さ 3 リンクまで順にアクセスする．これらのアクセスが終了したとき，候補収集が終了する．

手法 M_7: M_6 と同様に，移動元ページ u が存在した Web サイト内をクローリングして収集する．ただし，M_6 に比べて，より「移動先がありそうな順番」を意識したクローリングを行う．手法の詳細は [16] に説明されている．

手法 M_8: 保存していたリダイレクト先 ru (3 章参照) を収集する．

手法 M_9: リンクオーソリティ(1 章参照) の持つリンク先を収集する [12]．WISH システムでは，リンクオーソリティを計算し，最もらしいリンクオーソリティの持つリンクから順にアクセスしていく．WISH が発見したリンクオーソリティ全てのリンク先を収集した時点で候補収集を終了する．

手法 M_{10}: Web サーバが出力するエラーページ (404 ページ等) にかかれたリンク先のページを収集する．エラーページの先は，移動先ページが存在する Web サイトのトップページであることが多いため，リンク先を起点に深さ 3 リンクまで順にアクセスする．

5.2 実験方法

実験は次のように行う．まず，WISH システムによって実 Web 環境におけるリンクの監視を行い，リンク切れが発生した際に移動先の探索を行う．この際，最初は各候補収集手法 M_i の打ち切りは行わず (図 3 のアルゴリズムを用いる)，各手法で規程の候補収集が終了するまで探索を行った場合の発見率を求める．次に，図 6 のアルゴリズムを用いて，性質 3 を利用した探索の打ち切りを行った場合の発見率を求める．その際，打ち切りのための類似度の閾値 t を変化させ，発見率への影響を調査する．

発見率を議論するためには，WISH システムが発見したリンク切れに対して，原因がページの移動によるものかどうか，もしそうならば正しい移動先となるページはどこか，を調査する必要がある．これらは人手によって行った．まず，移動前のページの URL やコンテンツを利用してページの探索を行う．探索は，検索エンジンによる探索など，人がページの移動先を探すために行うと考えられる方法で行った．ここで，明らかに移動先と思われるページが発見された場合に，このリンク切れはページの移動によるものであると判断し，発見したページを正しい移動先とした．

5.3 実験結果

2007 年 2 月 20 日時点で，図 3 のアルゴリズムにより WISH システムはページの移動によるリンク切れ 45 件に遭遇し，そのうち 35 件の移動先を発見した．次に，探索を打ち切る閾値となる類似度を 1 から 0.1 まで変化させたとき，35 件のうち何パーセントが候補収集フェーズで収集されるか (発見されるか) を調査した．その結果を図 9 に示す．横軸は類似度の閾値 t である．この結果から分かるとおり，類似度の閾値を 0.5 にしても，ほとんど発見率に変化が無いことが分かる．

提案手法では，類似度の閾値を下げると，候補収集フェーズで必要なページアクセス数が削減される．類似度の閾値と発見率，必要なページアクセス数の関係を表したグラフを図 10 に示す．ここから，類似度の閾値を下げることにより，発見率を維持したままで必要なページアクセス数が大幅に削減可能なことが分かる．

以上の結果から，性質 3 が成立している可能性が高いことが推測できる．

次に，より詳細に結果を検証するため，各実験結果の詳細をまとめたものを図 11 に示す．各図には，各候補収集手法 M_i ごとの貢献，すなわち各 M_i がどれだけの正解ページを収集できたかと，それぞれの手法で 35 ページの移動先を発見するのに必要であったページアクセス数を示している．さらに，ページアクセス数を各手法の移動先発見成功数で割った平均ページアクセス数を示す．これは，その候補収集手法について，1 ページの移動先を発見するために必要な平均ページアクセス数を表している．各手法の移動先発見成功数の括弧内の数字は，他の手法では発見出来なかった移動先を内数で表している．

これらから，各手法 M_i とも，打ち切りのための閾値 t を下げても，発見ページ数は若干低下するだけであり，結果に大幅

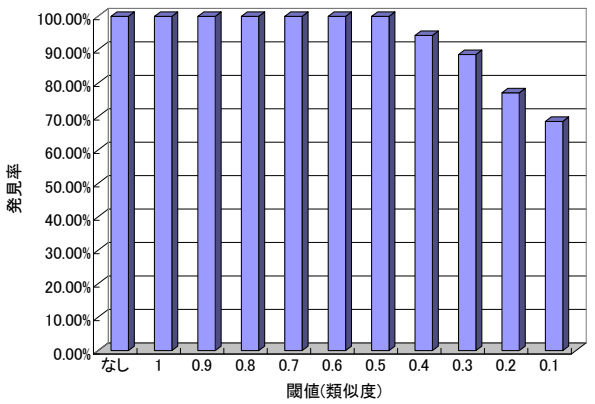

図 9　閾値の変化による発見率の推移

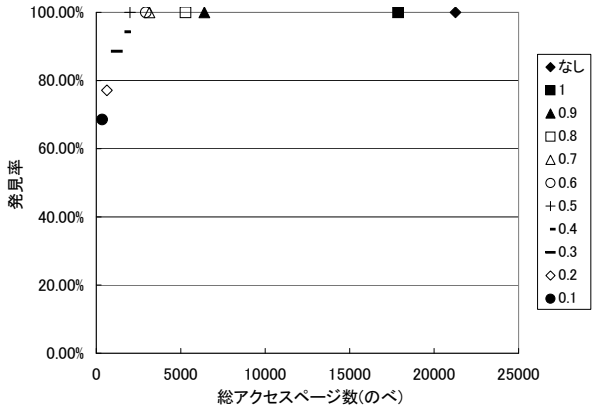

図 10　総アクセスページ数と発見率の関係

な影響がないことがわかる．また，各手法毎には若干発見されなくなるものがあるものの，図 9 のようにトータルで 100%を維持している理由は，複数の手法でオーバラップして発見される移動先ページが相当数あるため，互いに補い合っているからであることがわかる．

6. ま と め

本論文では，計算機による Web ページ移動先探索について，発見率を保持したまま探索に必要なページアクセス数を削減する手法について議論を行った．具体的には，移動先候補ページの探索の際，類似度を閾値とすることにより，発見率を維持したまま大幅にページアクセス数を削減可能であることを示した．今後は，より大規模な実験によって，本論文の結果をさらに検証する必要がある．

謝　　辞

ゼミなどでご議論いただきました筑波大学大学院図書館情報メディア研究科の阪口哲男助教授，永森光晴講師に感謝致します．本研究の一部は，文部科学省科学研究費補助金（#18049005）による．

文　　献

[1] ASHMAN, H. AND DAVIS, H. 1998. Panel missing the 404: link integrity on the world wide web. *Computer Networks 30(1-7)*, 761–762.

[2] Atsuyuki Morishima, Akiyoshi Nakamizo, Toshinari Iida, Shigeo Sugimoto, Hiroyuki Kitagawa. Automatic Correction of Broken Web links: Ideas, Experiments, and Lessons Learned. (submitted)

[3] Akiyoshi Nakamizo, Toshinari Iida, Atsuyuki Morishima, Shigeo Sugimoto, Hiroyuki Kitagawa: A Tool to Compute Reliable Web Links and Its Applications. International Special Workshop on Databases for Next Generation Researchers (SWOD2005), pp.146-149, April 2005.

[4] BEYNON, M. AND FLEGG, A. 2004. Hypertext request integrity and user experience. US Patent Application Publication, US 2004/0267726 A1.

[5] BEYNON, M. AND FLEGG, A. 2005. Guaranteeing hypertext link integrity. US Patent Application Publication, US 2005/0021997 A1.

[6] DAVIS, H. C. 1999. Hypertext link integrity. *ACM Comput. Surv. 31*, 4es, 28.

[7] GVU's WWW User Surveys. http://www.gvu.gatech.edu/user_surveys/

[8] HUXLEY, L., PLACE, E., BOYD, D., AND CROSS, P. 2002. Planet sosig - a spring-clean for sosig: a systematic approach to collection management. http://www.ariadne.ac.uk/issue33/planet-sosig/.

[9] 飯田敏成, 澤菜津美, 森嶋厚行, 杉本重雄, 北川博之, Web ページ移動先発見のための公開実験システム. 日本データベース学会 Letters, Vol.4, No.2, pp.21-24, 2005 年 9 月.

[10] M.Beynon, A.Flegg: Guaranteeing Hypertext Link Integrity. US Patent Application Publication, US 2005/0021997 A1, Jan, 2005.

[11] M.Beynon, A.Flegg: Hypertext Request Integrity and User Experience. US Patent Application Publication, US 2004/0267726 A1, Dec, 2004.

[12] 中溝昌佳，森嶋厚行，杉本重雄，北川博之, WWW リンク一貫性維持支援システムにおけるリンク切れ自動修復. 日本データベース学会 Letters, Vol.3, No.3, 2004 年 12 月.

[13] 中溝昌佳，森嶋厚行，有山智洋，杉本重雄，北川博之, WWW コンテンツ一貫性維持のためのリンク更新機構の提案. 日本データベース学会 Letters, Vol.2, No.2, pp.65-68, 2003 年 10 月.

[14] PHELPS, T. A. AND WILENSKY, R. 2000. Robust hyperlinks: Cheap, everywhere, now. In *Proc. of DDEP/PODDP 2000*. 28–43.

[15] Robert P. Dellavalle, Eric J. Hester, Lauren F. Heilig 他, Going, Going, Gone: Lost Internet References. Science, Vol.302, 2003 年 10 月 31 日.

[16] 澤菜津美, 森嶋厚行, 飯田敏成, 杉本重雄, 北川博之, Web ページ移動先発見のための効率的なクローリング手法. (投稿中)

[17] 澤菜津美, 飯田敏成, 森嶋厚行, 杉本重雄, 北川博之, Web ページ移動先発見のためのクローリング手法の提案. 情報処理学会研究報告, Vol.2006, No.78(2006-DBS-140(II)), pp.437-442. 電子情報通信学会技術研究報告, Vol.106, No.150, pp.91-95.

[18] Seung-Taek Park, David M.Pennock, C.Lee Giles, Robert Krovetz: Analysis of lexical signatures for improving information persistence on the World Wide Web. ACM Trans. Inf. Syst. 22(4): 504-572 (2004)

[19] XENU'S LINK SLEUTH. http://www.cs. washington.edu/lab/sw/LinkSleuth.html.

打ち切り無し

手法	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}
移動先発見成功数	13 (3)	3 (0)	5 (0)	16 (2)	15 (0)	14 (4)	2 (1)	12 (2)	9 (0)	3 (0)
ページアクセス数	136	43	235	151	356	11757	2	12	4448	4125
平均ページアクセス数	10.5	14.3	47.0	9.4	23.7	839.8	1.0	1.0	494.2	1375.0

閾値 t: 1

手法	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}
移動先発見成功数	13 (3)	3 (0)	5 (0)	16 (2)	15 (0)	14 (4)	2 (1)	12 (2)	9 (0)	3 (0)
ページアクセス数	97	40	199	115	252	9946	2	12	3076	4125
平均ページアクセス数	7.5	13.3	39.8	7.2	16.8	710.4	1.0	1.0	341.8	1375.0

閾値 t: 0.9

手法	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}
移動先発見成功数	12 (3)	3 (0)	5 (0)	16 (2)	15 (0)	12 (4)	2 (1)	12 (2)	8 (0)	3 (1)
ページアクセス数	86	23	155	90	189	3496	2	12	1975	372
平均ページアクセス数	7.2	7.7	31.0	5.6	12.6	291.3	1.0	1.0	246.9	124.0

閾値 t: 0.8

手法	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}
移動先発見成功数	12 (3)	3 (0)	4 (0)	16 (2)	14 (0)	12 (4)	2 (1)	12 (2)	8 (0)	3 (1)
ページアクセス数	69	23	115	81	167	3496	2	12	943	372
平均ページアクセス数	5.8	7.7	28.8	5.1	11.9	291.3	1.0	1.0	117.9	124.0

閾値 t: 0.7

手法	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}
移動先発見成功数	12 (3)	3 (0)	4 (0)	15 (2)	13 (0)	11 (4)	2 (1)	12 (3)	8 (0)	3 (1)
ページアクセス数	69	23	111	71	153	1695	2	12	943	69
平均ページアクセス数	5.8	7.7	27.8	4.7	11.8	154.1	1.0	1.0	117.9	23.0

閾値 t: 0.6

手法	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}
移動先発見成功数	12 (3)	3 (0)	4 (0)	15 (2)	11 (0)	11 (4)	2 (1)	12 (3)	8 (0)	3 (1)
ページアクセス数	63	23	111	71	125	1521	2	12	935	69
平均ページアクセス数	5.3	7.7	27.8	4.7	11.4	138.3	1.0	1.0	116.9	23.0

閾値 t: 0.5

手法	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}
移動先発見成功数	12 (3)	3 (0)	4 (1)	15 (2)	11 (0)	9 (4)	2 (1)	12 (4)	8 (0)	2 (1)
ページアクセス数	63	23	111	71	116	618	2	12	935	44
平均ページアクセス数	5.3	7.7	27.8	4.7	10.5	68.7	1.0	1.0	116.9	22.0

閾値 t: 0.4

手法	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}
移動先発見成功数	12 (3)	3 (0)	4 (1)	12 (2)	9 (0)	7 (2)	2 (1)	12 (4)	7 (0)	2 (1)
ページアクセス数	63	23	111	54	100	520	2	12	768	44
平均ページアクセス数	5.3	7.7	27.8	4.5	11.1	74.3	1.0	1.0	109.7	22.0

閾値 t: 0.3

手法	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}
移動先発見成功数	9 (2)	2 (0)	4 (1)	10 (2)	7 (0)	6 (2)	2 (1)	12 (6)	6 (0)	2 (1)
ページアクセス数	48	20	111	43	42	317	2	12	573	44
平均ページアクセス数	5.3	10.0	27.8	4.3	6.0	52.8	1.0	1.0	95.5	22.0

閾値 t: 0.2

手法	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}
移動先発見成功数	4 (3)	1 (0)	2 (0)	8 (2)	6 (0)	5 (2)	2 (1)	12 (6)	4 (0)	2 (1)
ページアクセス数	32	11	72	31	31	237	2	12	159	44
平均ページアクセス数	8.0	11.0	36.0	3.9	5.2	47.4	1.0	1.0	39.8	22.0

図 11　詳細な実験結果