Web リンク切れの自動修正における信頼度の高いリンク情報の利用

中溝　昌佳[†]　　飯田　敏成[††]　　森嶋　厚行[†††]　　杉本　重雄[†††]　　北川　博之[††††]

† 芝浦工業大学大学院 工学研究科　〒 337–8570 埼玉県さいたま市見沼区深作 307
†† 筑波大学 図書館情報専門学群　〒 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/計算科学研究センター
〒 305–8573 茨城県つくば市天王台 1–1–1
E-mail: †m103198@sic.shibaura-it.ac.jp, ††{m263,mori,sugimoto}@slis.tsukuba.ac.jp, †††kitagawa@cs.tsukuba.ac.jp

あらまし　我々は，Web のリンク切れを発見すると自動的にその修正を試みるシステムの開発を行っている．本システムの特徴は,「リンクオーソリティ」の概念を導入した点にある．Web ページ p のリンクオーソリティとは，p が移動したときに確実にリンクが更新されることが期待できる Web ページである．例えば，ある大学の学科一覧ページはその大学の学科ホームページのリンクオーソリティである．本システムでは，リンク切れの自動修正の際にリンクオーソリティの情報を利用する．本稿では実験を通じて，Web リンク切れの修正精度向上にリンクオーソリティを利用することの効果を検証する．

キーワード　WWW，リンク切れ，一貫性維持

Utilization of Reliable Links for Automatic Correction of Broken Web Links

Akiyoshi NAKAMIZO[†], Toshinari IIDA[††], Atsuyuki MORISHIMA[†††], Shigeo SUGIMOTO[†††], and Hiroyuki KITAGAWA[††††]

† Grad. Sch. of Engineering, Shibaura Inst. of Tech.,　307 Fukasaku,Minuma-ku, Saitama, 337–8570 Japan
†† Sch. of Library and Information Science, Univ. of Tsukuba,　1–2 Kasuga, Tsukuba, Ibaraki, 305–8550 Japan
††† Grad. Sch. of Library, Information and Media Studies/ Research Center for Knowledge Communities,
Univ. of Tsukuba.　1–2 Kasuga, Tsukuba, Ibaraki, 305–8550 Japan
†††† Grad. Sch. of Systems and Information Engineering/ Center for Computational Sciences,
Univ. of Tsukuba.　1–1–1 Tennohdai, Tsukuba, Ibaraki, 305–8573 Japan
E-mail: †m103198@sic.shibaura-it.ac.jp, ††{m263,mori,sugimoto}@slis.tsukuba.ac.jp, †††kitagawa@cs.tsukuba.ac.jp

Abstract　We are developing a system to automatically correct broken links in the Web. One of the feature is that the system introduces a novel concept of ”link authorities.” A link authority of an Web page p is a Web page s.t. links to p contained in the page are maintained to be kept up-to-date. For example, a Web page for a department list of a university would be a link authority of each page for the universitiy’s departments. Our system uses information on link authorities to correct broken links. This paper shows our experimental results on the automatic correction and discusses the effects of using link authorities on precision.

Key words　WWW, Broken Link, Integrity Maintenance.

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である．Web の特徴の一つに分散管理があげられる．すなわち，Web コンテンツは多くの組織・個人により独立して追加，削除，更新が行われている．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．コンテンツの一貫性が損なわれる一例として，我々は Web のリンク切れに着目している．我々はこれまで，Web のリンク切れを発見すると変更先と考えられる

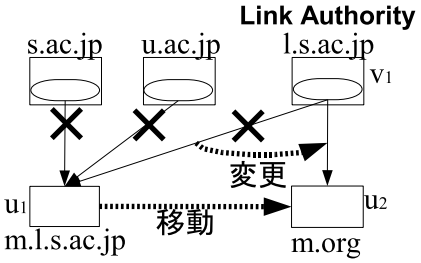

図 1　Link Authority

リンクの候補を自動的に発見しリンクの修正を試みるシステムを開発し，実験を行ってきた [1] [2]．本システムでは，Web のリンク切れはページの移動に伴って生じたものであると仮定して移動先の探索を行う．

また，我々は [3] において，信頼できるリンクを含むページである「リンクオーソリティ (Link Authority)」を求めるための仕組みと，それを実装した LA サーバ (Link Authority Server) を提案し，LA サーバを利用することにより Web ページの移動先を効率よく発見できる可能性が有ることを指摘した．ある Web ページ p のリンクオーソリティとは，p が移動したときに全 Web ページ中で十分に大きな確率でリンクが更新されるページである．

本稿では，(1)Web リンク切れの自動修正における LA サーバの利用，(2)LA サーバの設計と実装，(3) 実験を通じた LA サーバの効果の検証，について述べる．

Web のリンク切れに関する関連研究やシステムとしては，リンク切れの自動チェッカやイントラネットにおけるリンク切れ不整合の検出 [4]，Web サイト管理者のためのリンク切れ対策に関する文献 [5] などが存在するが，我々の知る限りリンク切れの自動修正に取り組んだ研究は存在しない．

2.　Web のリンク切れとリンクオーソリティ

直観的には，**リンクオーソリティ**とは「リンク先の内容が変化したときに，全 Web ページ中で十分に大きな確率でリンクが更新されるページ」のことである(注1)．

例を図 1 に示す．ここで，ある大学 A の研究室の Web ページ u_1 (m.l.s.ac.jp) が存在し，このページは複数のページからリンクされている．このうち v_1 (l.s.ac.jp) はその研究室が所属する学科の研究室一覧ページである．このとき，一般には v_1 は u_1 に関するリンクオーソリティである．従って，u_1 が u_2 (m.org) に移動したとき，u_1 へリンクしているページはリンク切れを起こすが，v_1 は u_1 へのリンクを u_2 に張り替えるはずである．

定義. ある Web ページ v は次の条件を満たすとき，別の Web ページ u のリンクオーソリティであると言う．**(1)** v が u へのリンクを持っており，かつ **(2)** u が u_{new} に移動すると，v 中の u へのリンクが u_{new} へのリンクに確実に変更されることが強く期待される．□

Web のリンク切れ修正に対するリンクオーソリティの利用のアイデアは次の通りである．すなわち，ある Web ページ中に含まれる u へのリンクがリンク切れになった場合，もし u のリンクオーソリティ v を知っていれば，v を見ることにより新たなリンク先 u_{new} を知ることが出来る．

(注1)：Google などにおける Authority ページとは異なる概念である．

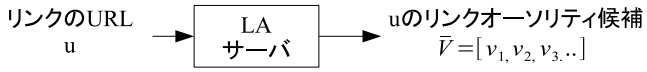

図 2　LA サーバの概要

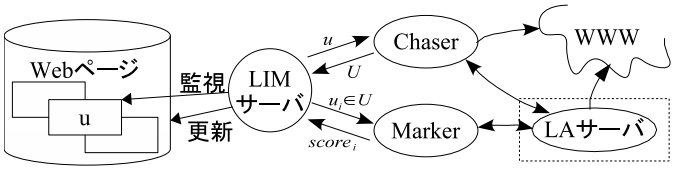

図 3　全体アーキテクチャ

3.　LA サーバを用いたリンク切れ自動修正

LA サーバとは: LA サーバ (Link Authoity Server) [3] は，ある Web ページの URL u を入力とし，u のリンクオーソリティと考えられる候補を出力する (図 2)．実際にはリンクオーソリティを一意に求めることは困難であるため，LA サーバは複数のリンクオーソリティ候補を収集し，リンクオーソリティである可能性が高いと考えられる順にランキングしたページの URL のリスト $\bar{V} = [v_1, v_2, \ldots]$ を結果として出力する．

リンク切れ自動修正システム: 今回は，これまで我々が開発してきたリンク切れ自動修正システムに LA サーバを追加する．図 3 に自動修正システムのアーキテクチャを示す．点線で囲まれた部分以外が以前のシステムである．簡単化のため，ここでは，システムが監視対象とするリンクは URL u で表されるただ一つのリンクに限定する．本システムは監視下の u がリンク切れであることを発見すると，新しいリンク先 u_{new} を発見し，u_{new} に自動修正することを試みる．本システムの主要な構成要素は，対象となるリンクを監視する LIM(Link Integrity Maintenance) サーバ，移動先のページの URL である u_{new} の候補集合 U を収集する Chaser，U に含まれる発見された各候補に対して “移動先らしさ” を表すスコアを計算する Marker である．これらの動作をまとめた抽象アルゴリズムを図 4 に示す．アルゴリズムの詳細は [1] にあるため省略するが，簡単に言えば次のようになる．(1) LIM サーバは u を監視する．リンク切れを発見すると次のように Chaser と Marker を呼び出す．(2) Chaser は移動前の Web ページ u のコンテンツと URL の情報を用いて，Web サーチエンジンによる候補収集やロボットによるサイト内探索を用いた候補収集を行い，U を作成する (4 行目)．(3) Marker は各 $u_i \in U$ に対し，主に移動前の Web ページと候補リンク先の Web ページの類似度や URL の関係などに基づいてスコア $score_i$ を計算する (5〜7 行目)．(4) LIM サーバは，$score_i$ を用いて，U の中の u_i をランキングし，リスト $\bar{U}$ を計算する (8 行目)．

LA サーバを追加した場合の処理は次のようになる (図 3 点線部)．Chaser はリンクオーソリティが含むリンク群を，移動先リンク u_{new} の候補として新たに U に追加する．Maker は

```
1.LIM-Server(u) {
2.   while (true) {
3.      if (isBroken(u)) { // link is broken
4.         Set U=Chaser(u);
5.         for each u_i in U {
6.            score_i= Marker(u_i);
7.         }
8.         List Ū = the result of sorting U by score_i;
9.      }
10.     Interval();
11. }
12.}
```

図 4　LIM サーバの基本動作

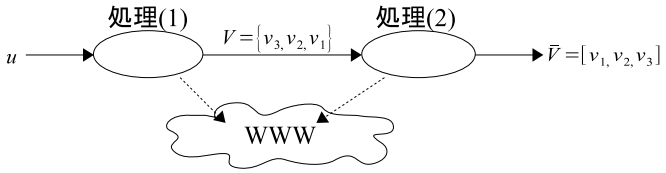

図 5　LA サーバの処理

U 中の候補ページのランキングの際，もしそのページがリンクオーソリティに指されているならば高いスコアを割り当てる．具体的には，候補ページが，LA サーバが求めたランキング上位 5 位までのリンクオーソリティ候補からリンクをされている場合，元の $score_i$ にある定数をかけることにより，スコアを高くする．

LIM サーバでは，監視対象の各リンクに対して，利用者が (1) リンクオーソリティを明示的に指定する手段と，(2) 自動修正を行うためのスコアの閾値を明示的に指定する手段を，それぞれ提供している [2]．(1) では，利用者によって明示的に指定されたリンクオーソリティを参照することにより，LIM サーバはそのリンクオーソリティに従ってリンクを自動修正する．(2) では，利用者により各リンクに対してリンクの自動修正を行うための閾値を指定することができる．LIM サーバは，この閾値を見て，もし指定された閾値より候補ページのスコアが大きい場合，発見されたリンクへ自動修正を行う．また，もし指定された閾値よりも候補ページのスコアが小さい場合，移動先候補の一覧ページを生成し，その生成されたページのリンクへ自動修正を行う．

4.　LA サーバの設計と実装

ある URL u が与えられたとき，u のリンクオーソリティの候補リスト $\bar{V}$ を計算するための LA サーバの処理は図 5 の様になる．

処理 (1): u へのリンクを持っているページ (群) V を収集． 定義により，u のリンクオーソリティは必ず u へのリンクを持つ．そこで，この処理では，u へのリンクを持っている Web ページ (の URL) 集合 $V = \{v_1, v_2, \ldots\}$ を作成する．

処理 (2): ページ群 V 中の各ページをランキング． 収集された各候補 $v_i \in V$ をある基準に基づいて評価し，リンクオーソリティと思われる順番にランキングしたリスト $\bar{V}$ を求める．

v_i の属性	値	意味
同一同一	真偽	u と同一サイトの同一ディレクトリに v_i が存在
同一上位	真偽	u と同一サイトの上位ディレクトリに v_i が存在
同一下位	真偽	u と同一サイトかつ下位ディレクトリに v_i が存在
同一その他	真偽	u と同一サイトかつその他のディレクトリに v_i が存在
上位サイト	真偽	u が属するサイトをサブドメインとして含むサイトに v_i が存在
外部サイト	真偽	u の上位サイトおよび同一サイト以外のサイトに v_i が存在
相互リンク	真偽	u と v_i の間に直接的，もしくは間接的な相互リンクが存在
index	真偽	v_i のファイル名がデフォルトファイル名 (典型的には index.html) である
#L	自然数	v_i のページに含まれているリンクの数
B	[0, 1]	v_i のページに含まれているリンクのうち，リンク切れではないものの割合

図 6　各 v_i の属性

処理 (1) としては，u へのリンクを持つページの集合を計算できれば何でも良い．したがって，この処理はリンクオーソリティの計算においては本質的でない．例えば，クローラを用いて収集する方法，Web アーカイブなどを利用する方法，Web サーチエンジンを利用する方法などが考えられる．リンクオーソリティを求める問題として本質的なのは処理 (2) である．

4.1　リンクオーソリティ候補 V の収集

今回の LA サーバの実装では，処理 (1) の実装を次の (a)(b) によって行うこととした．(a)Google Web Service API [6] および Alexa Web Information Service API [7] を用いて，u へのリンクをもつページを検索する．(b) クローラを用いて，u へのリンクを持つページがある可能性が高いと考えられる場所を探索する．具体的には，(b) のクローラは次の場所を探索する．(1)u と同じサイト内のページ．(2)u の属するサイトをサブドメインとして含むサイトに属するページ．(3)u 中のリンクが指す先のページ．したがって，必ずしも u にリンクを持つ全てのページが V に含まれるとは限らない．

4.2　リンクオーソリティ候補 V のランキング

候補のランキングは LA サーバの本質的な処理である．本 LA サーバが，u のリンクオーソリティ候補 $v_i \in V$ をランキングし $\bar{V}$ を求める処理は，次のように設計した．まず，各 v_i に対してそれぞれ図 6 の属性を求める．次に，次のヒューリスティクスに基づいて各 v_i の「リンクオーソリティらしさ」を求める．

H1　同一ディレクトリに u と v_i が存在すれば，v_i は u へのリンクを確実に更新する可能性が高いと考えられる．

H2　v_i が u に対して論理的に上位の存在である場合，v_i はリンクオーソリティの可能性が高い．ここで論理的に上位の存在であるとは，例えば u が学科のページであるのに対して v_i が学部のページであるといった場合である．しかし確実に更新される度合いは同一ディレクトリよりはやや劣ると考えられる．

H3　v_i が u と同一サイトの上位ディレクトリに配置されてい

パターン	ランク	同一サイト				上位	外部	相互	index
		上位	同一	下位	その他	サイト	サイト	リンク	
a	1		●					-	●
b	2	●						●	-
c	2					●		●	-
d	2						●	●	-
e	3		●						
f	4	●							-
g	5		●					●	
h	6				●			●	-
i	7				●				-
j	7					●			-
k	8			●				-	-
l	8						●		-

図 7　リンクオーソリティランク

る場合，v_i は u に対して論理的に上位の存在である事が多い．

H4　v_i が u に対して論理的に上位の存在である場合，Web サイトの設計の方法によっては v_i は u と同一ディレクトリ内の index.html として配置されることがある．

H5　**H1**〜**H4** より，v_i が u と同一ディレクトリに配置され，かつ index.html である場合はリンクオーソリティである可能性が非常に高い．

H6　v_i と u が直接的，間接的な相互リンクを持つ場合，v_i はリンクオーソリティである可能性が高い．ここで間接相互リンクとは，v_i と u のサイトの間で，異なるページを介し互いにリンクをしているような関係をさす (図 8).

H7　同一ディレクトリ内で v_1 から v_2 へリンク $v_1 \rightarrow v_2$ があり，逆方向に存在しない場合，v_2 が v_1 に対して論理的に上位の存在であるとは考えにくい．

H8　(**H7** と比較して)v_1 と v_2 が同一ディレクトリに配置され，お互いに相互リンクが張られている場合，これらの間の論理的な上位下位の関係は何とも言えない．なぜなら「戻る」などのリンクが存在するからである．

H9　リンク切れが多いページはリンクオーソリティとは考えにくい．このヒューリスティクスは他のヒューリスティクス **H1**〜**H8** とは独立していると考えられる．

我々は，他とは独立していると考えられる **H9** を除いたヒューリスティクス (**H1** から **H8**) を考慮して図 7 の表を作成した．表の各項目は v_i の属性を表しており，その説明は図 6 にある．図において黒丸はその属性が真であることを表し，空白は偽であることを表す．ハイフンはどちらでも良いことを表す．本 LA サーバのランキング処理では，v_i が与えられると，まずこの表を用いて v_i の**リンクオーソリティランク**を求める．一般に，同じランクを持つ v_i は複数存在する．次に，**H9** を反映させるため，各ランクの中で，そこに属する v_i を，リンク数とリンク切れではないものの割合の相乗平均 $\sqrt{\#L \times B}$ をキーとして降順に並べる．以上により並べられた結果を，$\bar{V}$ とする．

間接相互リンク発見の実装. v_i が与えられたとき，図 6 の相互リンク属性を求めるためには，u と v_i との間に相互リンクがあるかを知らねばならない．直接的な相互リンクを持つことは簡単に調べられるが，図 8 のような間接的な相互リンクを発見するためには，多くのリンクを探索する必要がある．

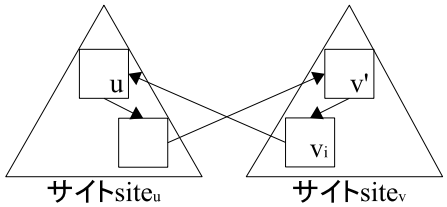

図 8　間接相互リンク

監視対象ページ数	49750
リンク切れ総数	146
内 移動に伴うリンク切れ	47

図 9　実験期間内に発生したリンク切れの数

本来，この処理では図 8 の $site_u$ と $site_v$ の両サイトの内部を探索しなければならないが，本 LA サーバの実装では，あらかじめ処理 (1) によって v_i から u へリンクの存在が保証されていることを利用する．具体的には $site_u$ の探索だけを行い，$site_v$ 中のいずれかのページに対するリンクを発見すると，u と v_i 間に間接的な相互リンクが存在すると見なす．これは厳密に言えば $site_v$ の v' と v_i 間のリンクの存在を保証しない近似的な処理であるが，これにより探索処理量を半分にしている．

5. 実　　験

実装した LA サーバを用いた実験を行い，LA サーバによって発見されたリンクオーソリティを利用することにより，リンク切れ自動修正システムの移動先発見精度はどのように変わるのか検証を行った．

本実験ではまず，筑波大学，芝浦工業大学，北海道大学，東北大学，東京大学，名古屋大学，京都大学，大阪大学，九州大学の計 9 大学のドメインに属するサイトの中に含まれるリンクを収集した．本実験で監視対象とするリンクは，これらの収集したリンクのうち，リンク元とリンク先が異なるサイトであるようなリンク (合計 49750 個) である．したがって，リンクが指している先は学内のサイトとは限らない．監視対象をこれらのリンクに絞った理由は，リンク先とリンク元が異なるサイトであればリンク切れが発生する確率が高いのではないかという我々の予測による．

5.1 実 験 結 果

これらのリンクを対象に 2005 年 1 月 9 日より実験を行っている．2005 年 2 月 4 日時点で，LIM サーバは監視対象のうち 146 個のリンク切れを発見した (図 9)．このうち，Web ページ移動によって生じたと考えられるリンク切れは 47 個存在した．ここで，ページ移動によるリンク切れとの判断は次のように行った．つまり，LIM サーバの探索ログなどをもとに，様々な方法で手作業で移動先の探索を行い，移動先と考えられるページが発見できたものを移動したページとした．また，移動先発見が成功したとは，LIM サーバが出力した結果 $\bar{U}$ の上位 3 位以内に正しい移動先が含まれていることとした

まず，移動先の探索結果を図 10 に示す．図における “LA サー

	成功	失敗	成功率
LA サーバなし	28	19	59.6%
LA サーバあり	34	13	72.3%

図 10　移動先の探索結果

分類	詳細	個数
a	$V = \phi$	18
b	$V \neq \phi$ だが，LA らしきページは含まれていない	14
c	V に LA と思われるページが含まれているが，$\bar{V}$ の上位ではない	0
d	$\bar{V}$ の上位 5 位以内で発見されたが，まだリンクが更新されていない	3
e	$\bar{V}$ の上位 5 位以内で発見され，正しくリンクが更新された	12

図 11　リンクオーソリティの探索結果の分類

バなし” の行は LA サーバを利用せず，“LA サーバあり” の行は LA サーバを利用して探索を行った結果である．LA サーバを利用せずに移動先の発見に成功したものは 28 個，失敗したものは 19 個，成功率は 59.6%であった．それに対して，LA サーバを利用した場合は，移動先の発見に成功したものは 34 個，失敗したものは 13 個，成功率は 72.3%となった．この結果から見てわかるとおり，LA サーバを利用することにより，ページの移動先発見率が約 12.7 ポイント向上した．

一方で，LA サーバを利用しても移動先の発見ができなかったものが 13 個存在した．そこで，実験において発見された Web ページ移動に伴うリンク切れ 47 個について，LA サーバが出力するログなどから，LA サーバの探索結果についての分析を行った (図 11).

その結果，LA サーバがリンクオーソリティと考えられるページを発見でき，正しく更新されているページは 47 個中 12 個 (図 11 e) であった．残る 35 個については，図 11 の a〜d のように分類した．以下に，分類ごとの原因の検証を行う．

分類 a　この分類に当てはまるものは，18 個存在した．これは今回の実験で利用したリンクオーソリティ候補収集の手法が原因であると考えられる．本来，リンクオーソリティ候補群 V としては，u に対してリンクを張っている全てのページが収集されるべきである．しかし，今回我々が用いたリンクオーソリティ候補の収集手法では不十分であったと考えられる．この分類に当てはまるものへの対応としては，リンクオーソリティ候補の収集手法の改良などが考えられる．

分類 b　この分類に当てはまるものは，14 個存在した．これは上述の b と同様の理由も考えられる．しかし，全ての Web ページにリンクオーソリティが存在するわけではないとも考えられる．つまり，リンクオーソリティ候補の収集手法の追加や改良を行っても，このパターンに該当するものを限りなく減少させることは不可能であると考えられる．

分類 c　今回の実験では，このパターンは存在していない．本実験の結果より，我々の考えたリンクオーソリティ候補ランキングのヒューリスティクスが適切だったと考えることができる．この分類に当てはまるものが増加するようであれば，リンクオーソリティ候補のランキングヒューリスティクスの見直しが必要である．しかし，現在の実験結果からはその必要性はないと考えられる．

効果	内容	個数
あり	リンクオーソリティが発見されていなければ，正しい移動先を発見できなかった	4
	リンクオーソリティが発見されていなければ，正しい移動先を上位 3 位以内に評価できなかった	2
なし	リンクオーソリティは発見されたが，正しい移動先を上位 3 位以内に評価できなかった	1
	リンクオーソリティは発見されたが，移動先の発見に特に影響を与えなかった	5

図 12　移動先探索におけてリンクオーソリティを利用した影響

分類 d　この分類に当てはまるものは，3 個存在した．これは，Web ページの移動が発生してから，リンクオーソリティが持つリンクを参照するまでの時間が短かったためだと考えられる．正しいリンクオーソリティは，Web ページの移動に伴いリンクを正しく修正すると考えられるが，それがページ移動が起こった直後に行われるとは限らない．この分類に当てはまるものは，ページ移動後，ある程度時間が経過した後にリンクオーソリティが有効に機能するのではないかと考えられる．

本実験では，特に分類 a と分類 b に該当するものが多く存在した．この原因は，本実験で利用した LA サーバのリンクオーソリティ候補 V の収集手法に問題があったものと考えられる．LA サーバの現在の実装では，リンクオーソリティ候補 V を効率的に収集するために，先に述べた候補収集の手法を次に示す手順で実施した．

手順 1　すべての監視対象の URL に対して Alexa Web Information Service API [7] を用いて，u へのリンクをもつページを収集.

手順 2　すべての監視対象の URL に対して Google Web Service API [6] を用いて，u へのリンクをもつページを収集.

手順 3　すべての監視対象の URL に対してクローラを用いて，u へのリンクを持つページがある可能性が高いと考えられる場所から収集.

本稿で示した実験結果では，**手順 1** を用いた収集しかできていない．そのため，候補の収集が不十分であったと考えられる．

また，LA サーバによってリンクオーソリティと考えられるページが発見されていた 12 個については，ページ移動先の発見に及ぼした影響についても調査を行った (図 12)．その結果，“リンクオーソリティが発見されていなければ，正しい移動先を発見することができなかった” というものが 4 個，“リンクオーソリティが発見されていなければ，正しい移動先を上位 3 位以内にランキングできなかった” というものが 2 個存在した．つまり，LA サーバを利用することで，これまでの LIM サーバだけでは発見できない移動先を発見することができた．

その一方で “リンクオーソリティは発見できたが，正しい移動先を上位 3 位以内に評価できなかった” というものが 1 個存在した．しかし，これは，LIM サーバによる移動先候補の評価

手法の問題であり，LA サーバの問題ではなかった．

5.2 代表的な事例の分析

ここでは，これまでの LIM サーバだけでは発見することができなかったが，LA サーバが求めたリンクオーソリティを利用することで移動先を発見することができたものを，事例を用いて説明する．5.1 節で述べたとおり，LIM サーバと LA サーバをともに利用することにより，発見することができた移動先のページは 6 個増えた．これらを，以下の通り 2 つのグループに分類した．

グループ 1: これまでの LIM サーバの結果には正しい移動先が候補にすら入らなかったが，LA サーバが求めたリンクオーソリティを利用することで，移動先を発見することができたもの．

グループ 2: これまでの LIM サーバの結果には正しい移動先が候補に入っていたが，LA サーバが求めたリンクオーソリティを利用することで，移動先候補の上位 3 位以内として評価されたもの．

本稿では，各グループについて，二つずつ具体的な事例を用いて説明する．

グループ 1 このグループは，先に述べたとおり，LA サーバを利用することで，初めて移動先が発見できたものである．このケースに当てはまるリンク切れについて，以下に具体的な事例を 2 つあげる．

事例 1-1:**順天堂大学大学院 医学研究科 講師の日本語トップページが移動したことにより生じたリンク切れ**

これは，属していたサイトのすべて Web ページが，別のサイトへと移動をしたためにリンク切れが発生した．ここで，移動元のページを u，移動先のページ u_{new} として説明する．

本事例において，移動先 u_{new} は，これまでの Chaser によって収集された移動先候補 U の中には含まれていなかった．このリンク切れの特徴は，Web サイトが丸ごと移動したという点である．したがって，ページの移動先が移動元と異なるサイトであったために，これまでの Chaser では u_{new} を発見することができなかった．

しかし，LA サーバは u のリンクオーソリティ v を発見した．つまり，u が u_{new} へ移動した際，リンクオーソリティ v は u へのリンクを u_{new} へと修正したため，LIM サーバは v の持つリンクをたどることにより u_{new} を発見することができた．

ここで，リンクオーソリティ v は u の属していたサイトのルートページであり，移動元サイトから移動先サイトへと移転したことを示す内容が記述されていた．また，v は移動先サイトのトップページへのリンクを持っていた．

事例 1-2: **京都大学 ウイルス研究所 ヒトがんウイルス研究分野のページが同一サイト内で移動をしたためリンク切れが生じた．**

これは，京都大学 ウイルス研究所のサイトのリニューアルに伴い，サイト内の別のディレクトリにページが移動したためリンク切れが生じた．ここで，移動元のページを u，移動先のページ u_{new} として説明する．

本事例において，移動先 u_{new} は，これまでの Chaser によって収集された移動先候補 U の中には含まれていなかった．移動先ページ u_{new} は，これまでの Chaser ではたどり着くことができない場所であった．

しかし，LA サーバは u のリンクオーソリティ v を発見していた．つまり，u が u_{new} に移動した後，リンクオーソリティ v は u へのリンクを u_{new} へと修正したため，LIM サーバは v を利用することによって u_{new} を発見することができた．

ここで，具体的に u のリンクオーソリティ v は京都大学 ウイルス研究所の各研究分野の紹介ページであった．

グループ 2: このグループは，LA サーバを利用することで，正しい移動先がランキングの上位 3 位以内として評価されるようになったものである．このケースに当てはまるリンク切れについて，以下に具体的な事例を 2 つあげる．

事例 2-1: **新潟大学理学部の英語のトップページが，サイトのリニューアルのために生じたリンク切れ**

これは，新潟大学理学部のサイトのリニューアルに伴って，ページがサイト内の別のディレクトリに移動したためにリンク切れが生じた．ここで，移動元のページを u，移動先のページを u_{new} として説明する．

本事例では，u の移動先 u_{new} は，これまでの Chaser にって，すでに発見されていた．しかし，ページのコンテンツが大幅に変更されており，そのため u と u_{new} の類似度が低く，$\bar{U}$ の上位 3 位以内として評価されなかった．具体的には，ランキング第 9 位として評価されていた．

この u について，LA サーバはリンクオーソリティ v を発見した．そして，v が持っていた u へのリンクは u_{new} に更新されていた．このため，LIM サーバは v からリンクされている u_{new} をより高く評価し，リンク移動先候補の第 1 位として出力した．

ここで，LA サーバが発見したリンクオーソリティ v とは，新潟大学理学部のトップページであった．

事例 2-2: RIKEN BNL Research Center **のトップページが，同一サイト内で移動をしたためリンク切れが生じた．**

これは，RIKEN BNL Research Center のサイトのリニューアルに伴って，このページがサイト内の同一ディレクトリの中で移動したためリンク切れが生じた．具体的には，このページのファイル名は default.htm であったが，ファイル名が変更され default.asp となった．ここで，移動元のページを u，移動先のページを u_{new} として説明する．

このリンク切れの特徴としては，ページの移動先は同一のディレクトリ内であり，ただファイル名が変更されただけである．そのため，移動先 u_{new} は，これまでの Chaser によってすでに発見されていた．しかし，**事例** 2-1 と同様に，ページのコンテンツが大幅に変更されており，そのため u と u_{new} の類似度が低く，$\bar{U}$ の上位 3 位以内として評価されなかった．具体的には，ランキング第 6 位として評価されていた．

この u について，LA サーバはリンクオーソリティ v を発見した．また，この v は，u へのリンクを u_{new} へと更新をしていた．そのため，LIM サーバは v からリンクされている u_{new} をより高く評価し，リンク移動先候補の第 1 位として出力した．

ここで，LA サーバが発見したリンクオーソリティ v とは，RIKEN BNL Research Center のサブコンテンツ内で「トップ

ページへ戻る」リンクを持つページであった．

6. おわりに

本稿では，信頼できるリンクをもつ Web ページである「リンクオーソリティ」が，リンク切れ自動修正のためのページ移動先の発見にどのような効果があるかを検証した．具体的には，リンクオーソリティを計算するソフトウェアである LA サーバの設計と実装について述べ，実装した LA サーバを用いた実験とその実験結果について説明した．実験結果を分析した結果，リンクオーソリティを利用すると，利用しない場合と比較して，移動先発見率が約 12.7 ポイント向上したことがわかった．

今後の課題としては，LA サーバのリンクオーソリティ候補の発見率向上のための候補の収集手法の検討などがあげられる．

謝　　辞

ゼミなどでご議論いただきました芝浦工業大学工学部情報工学科の古宮誠一教授に感謝致します．本研究の一部は日本学術振興会科学研究費補助金若手研究 (B)(課題番号 15700108) による．

文　　献

[1] 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.

[2] 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.

[3] 中溝昌佳，森嶋厚行，杉本重雄，北川博之, WWW における信頼度の高いリンクの発見. 情報処理学会研究報告 Vol.2004, No.72(2004-DBS-134(II)), pp. 397-402. 電子情報通信学会技術研究報告 Vol. 104, No.177 (DE2004-63), pp. 87-92, 2004 年 7 月.

[4] 河合英紀, 河野泉, 石黒義英, 福島俊一, サイト品質管理のためのリンク不整合検出. 日本データベース学会 Letters, Vol. 3, No. 1, pp. 45-48, 2004 年 6 月

[5] Hugh C. Davis: Hypertext link integrity. ACM Comput. Surv. 31(4es): 28 (1999)

[6] Google Web APIs: http://www.google.com/apis/

[7] Alexa Web Information Service: http://pages.alexa.com/prod_serv/WebInfoService.html