2005－DBS－137（Ⅰ）（16）
2005／7／13

Webリンク切れ自動修正のための公開実験システムの開発

飯田敏成[†]　澤　菜津美[††]　森嶋厚行[†]
杉本重雄[†]　北川博之[†††]

近年，Web は社会における重要なメディアの一つとなっている．Web の特徴の一つとして分散管理が挙げられる．この特徴は，Web を役立つメディアにしている一方で，Web コンテンツの一貫性維持を困難にする要因ともなっている．これまで，我々は Web ページの移動によるリンク切れを自動的に修正することを試みるシステムを開発し，実験を行ってきた．しかし，これまでの実験は人工的に集めたリンク集合を対象にしたものであり，実際の利用状況とは異なっていた．そこで我々は，実際の利用状況での実験結果を取得すること，そして実験結果に基づいてシステムの精度向上を行うことを目的として公開実験を行う．本論文では，この公開実験とそのために必要なシステムの開発について説明する．

Development of A System for Open Experiments on Automatic Correction of Broken Links in the Web

Toshinari Iida,[†] Natsumi Sawa Atsuyuki Morishima,[††,†]
Shigeo Sugimoto[†] and Hiroyuki Kitagawa [†††]

Recently, the Web has become an important medium for our society. A characteristic of the Web is its framework for the distributed management of Web pages. Although the characteristic makes the Web a useful medium, it is also a reason why it is difficult to manage the integrity of Web contents. So far, we developed a system for automatic correction of broken links and performed experiments. However, the experiments were done with a fixed set of Web links we manually constructed. Therefore, we plan to perform open experiments whose purposes are to get experimental results in more practical situations and to use the results to improve our algorithms. This paper explains the open experiments and the system to be used in the experiments.

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．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 Web のリンク切れがあり，我々はこれに着目している．

我々はこれまでに次のような 2 つのシステムの開発，および実験を行ってきた．一つは，Web のリンク切れを発見すると変更先と考えられるリンクの候補を自動的に発見してリンクの修正を試みる LIM(Link Integrity Management) サーバである[3)5)]．このシステムは，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．もう一つは，移動した Web ページを発見するための強力な手がかりとなるリンクオーソリティを提供する LA(Link Authority) サーバである[1)2)4)]．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

† 筑波大学大学院 図書館情報メディア研究科
Grad. Sch. of Info. and Media Studies, Univ. of Tsukuba.
†† 筑波大学 図書館情報専門学群
Sch. of Lib. and Info. Science, Univ. of Tsukuba.
††† 筑波大学大学院 システム情報工学研究科
Grad. Sch. of Sys. and Info. Eng., Univ. of Tsukuba.

☆ Google などにおける Authority ページとは全く異なる概念である．

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．

我々はこれらのシステムを利用した実験を行ってきた．しかし，これまでの実験は人工的に集めたリンク集合を対象にしたものであり，実際の利用状況とは異なっていた．そこで我々は公開実験を行う．本公開実験では，実際の利用状況での実験結果を取得すること，そして実験結果に基づいてシステムの精度向上を行うことを目的とする．本稿では，我々のシステムを利用した公開実験および公開実験システムの開発について述べる．

関連研究 リンク切れの自動修正を試みるシステムは，我々の知る限り IBM の Peridot[6][7] だけである．このシステムは，リンク切れなどによって一貫性が損なわれたリンクを自動的に別のページへのリンクに修正することを試みる．他の関連研究としては lexical signunature[8] がある．この 2 つに共通している点は，Web コンテンツから抽出した特徴的なキーワードを利用してページの同定を行うアプローチを採用していることである．つまり，既にインデクシングされた Web ページ集合の中からリンクの修正候補として最適な Web ページを発見するという状態を想定している．それに対して，我々はインデクシングされているという前提が存在しない中で，Web 中からリンクの修正候補として最適な Web ページを発見することを試みる．具体的には，我々のシステムは「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リンクの修正候補の探索を行う．

リンク切れに関連する他の研究としては，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不整合の検出[9]，Web サイト管理者のためのリンク切れ対策に関する文献[10] などが存在する．

2. リンク一貫性維持支援システム

本章では，我々が今までに設計・開発した LIM サーバおよび LA サーバについて述べる．

2.1 LIM サーバ

LIM(Link Integrity Management) サーバとは，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である (図 1)．

LIM サーバの働きについて，表 1 に示す記号を利用して説明する．話を簡単にするために，ここでは，システムが監視対象とするリンクを 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，u の移動先 u_{new} を発見し，u へのリンクを 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監視・更新サーバ，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 の各候補に対して “移動先らしさ” を表すスコアを計算する Marker である．アルゴリズムの詳細は[3] にあるため省略するが，簡単に言えば次のようになる．(1) 監視・更新サーバは u を監視する．u がリンク切れであること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 $P(u)$，URL から分かる情報，LA サーバによって提供される u のリンクオーソリティを用いて，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．(3)Marker は各 $u_i \in U$ に対し，$P(u)$ と $P(u_i)$ の類似度や，u と u_i の URL から分かる関係などに基づいてスコア $score_i$ を計算する．(4) 監視・更新サーバは，$score_i$ を用いて各 $u_i \in U$ をランキングし，リスト $\bar{U}$ を計算する．

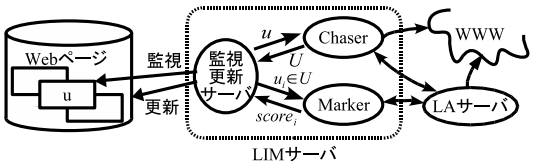

図 1 LIM サーバ アーキテクチャ

記号	説明
u	LIM サーバが監視している URL
u_{new}	u の移動先の URL
$P(u)$	ページ u のコンテンツ
w, w_{new}	u の存在するサイト，および u_{new} の存在するサイト

表 1 記号

LIM サーバの Chaser および Marker は，次に示すヒューリスティクスに基づいて候補収集およびスコアリングを行う．

H1 $P(u_{new})$ と $P(u)$ は似ている可能性がある．

H2 u から u_{new} にリダイレクトが行われている可能性がある．

H3 w 内で移動する可能性がある．

H4 w 内の u 以外のページの移動先サイトに u_{new} が存在する可能性がある．

H5 w から w_{new} に対してリンクが行われている可能性がある．

H6 u のリンクオーソリティから u_{new} にリンクが行われている可能性がある．

2.2 LA サーバ

LA(Link Authority) サーバとは，ある Web ページ

図 2 LA サーバの概要

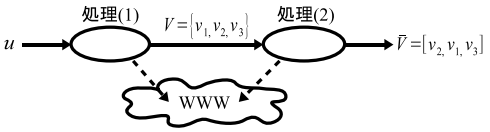

図 3 LA サーバの処理

の URL u を入力とし，u のリンクオーソリティであると考えられる候補 v を出力するシステムである (図 2)．実際にはシステムが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番にランキングしたリンクオーソリティ候補リスト $\bar{V}=[v_1, v_2, \ldots]$ を出力する．この $\bar{V}$ を計算するための処理は図 3 のようになる．

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) の集合 $V=\{v_1, v_2, \ldots\}$ を作成する．

処理 (2): ページ群 V 中の各ページ v 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．

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であるのは処理 (2) である．

処理 (2) のランキングを求める処理は，次のように行う．まず，各 $v_i \in V$ に対してそれぞれ表 2 の属性を求める．次に，LA サーバが利用する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．このヒューリスティクスの詳細は[1)2)] にあるため省略するが，簡単に言うと v_i と u の位置関係，およびファイル名などを考慮する．

我々はそれらのヒューリスティクスを考慮して表 3 を作成した[1)2)]．各項目は v_i の属性を表しており，その説明は表 2 にある．表 3 において黒丸はその属性

v_i の属性	値	意味
同一同一	真偽	u と同一サイトかつ同一ディレクトリに v_i が存在
同一上位	真偽	u と同一サイトかつ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リンクの割合

表 2 各 v_i の属性

ランク	同一サイト				上位サイト	外部サイト	相互リンク	idx
	上位	同一	下位	その他				
1		●					-	●
2	●						●	-
2					●		●	-
2						●	●	-
3		●						
4	●							-
5		●					●	
6				●			●	-
7				●				-
7					●			-
8			●				-	-
8						●		-

表 3 リンクオーソリティランク

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，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リンクの割合の相乗平均 $\sqrt{\#L \times B}$ 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．

3. 公 開 実 験

本章では，公開実験を行う目的，実験の概要，実験結果の解析方法について述べる．

3.1 目　　的

我々は，実際の利用状況での実験結果を取得すること，そして実験結果に基づいてシステムの精度向上を行うことを目的とする．

3.2 公開実験概要

我々が今までに行ってきた実験と，公開実験の相違について説明する．(1) 今までは我々が一定の規則に従って収集したリンクの集合をシステムの監視対象として実験を行ってきた．公開実験では監視対象とするリンクを指定するのは利用者であり，利用者の要求に

応じて監視対象とするリンクがダイナミックに増減する．(2) 今までは高々数万のリンクの集合をシステムの監視対象として実験を行っていた．公開実験ではより多くのリンクを監視対象とする．(3) 今までは実験結果を手作業で解析していた．公開実験では結果を自動的に解析することを目的とする．(4) 今までは，リンク切れが起きた際にシステムが発見したページの移動先候補は解析のためだけに利用されていた．公開実験では，リンク切れとなったリンクの修正候補として利用者に提供する．(5) 今までは公開を前提としていなかったのでユーザビリティを考慮していなかった．公開実験では多くの利用者に利用してほしいので，簡単にシステムの操作および結果の閲覧ができるようにする．(6) 今までは，著しく大きくない探索コストを無視してきた．公開実験ではリンク切れが発生してから短時間で探索を行い情報を提供するために，探索コストを最小限にする．

3.3 公開実験のログ

公開実験では，LIM サーバおよび LA サーバが監視対象のリンクについて監視および探索を行う過程で得ることができる様々な情報をログに記録する．これらのログとして記録する項目を表 4 に示す．

3.4 フィードバックとして受け取る情報

公開実験では，システムが提供したリンクの修正候補に対する利用者からのフィードバックを受け取る．ここでは，システムが受け取るフィードバックについて説明する．

- システムによって提供されたリンクの修正候補の中に正しい移動先があった場合には，その移動先をフィードバックとして受け取る．正しい移動先が提供されなかった場合にも正しい移動先が提供されなかったというフィードバックを受け取る．
- システムによって提供されたリンクの修正候補の中に正しい移動先がなかったが，利用者が自分で正しい移動先を発見することができた場合にはその URL をフィードバックとして受け取る．
- 利用者の登録したページの中でリンク切れとなったリンクは利用者にとってどのような種類のリンクであるかをフィードバックとして受け取る．(例えば企業のページ，大学のページ，友人のページ，他人 (面識がない) のページなど．)
- 移動先の探索の際にキーワード検索に利用されたクエリが適切にページの内容を表していたかどうかをフィードバックとして受け取る．

受け取ったフィードバックは，実験結果の解析に利用する．

監視対象の各 URL に関するログ		
名前	型	説明
URL	String	監視を行う URL
登録日時	Date	監視対象として登録された日時
リンク切れ日時	Date	リンク切れになったことをシステムが発見した日時
解除日時	Date	監視対象から除外された日時
各リンク切れ URL に対する LIM サーバの探索結果に関するログ		
名前	型	説明
URL	String	移動先の探索を行った URL
キーワード	String	H1 に基づいてキーワード検索を行うために利用したクエリ
H1 探索日時	Date	H1 に基づいたキーワード検索を最後に行った日時
H2 探索日時	Date	H2 に基づいた探索を最後に行った日時
H3 探索日時	Date	H3 に基づいた探索を最後に行った日時
H4 探索日時	Date	H4 に基づいた探索を最後に行った日時
H5 探索日時	Date	H5 に基づいた探索を最後に行った日時
H6 探索日時	Date	H6 に基づいた探索を最後に行った日時
LIM サーバが発見した各移動先候補に関するログ		
名前	型	説明
URL	String	移動先候補の URL
類似度	float	オリジナルの Web ページと候補の類似度
H1 探索結果	boolean	H1 に基づいたキーワード検索によって発見されたのかどうか
H2 探索結果	boolean	H2 に基づいた探索によって発見されたのかどうか
H3 探索結果	boolean	H3 に基づいた探索によって発見されたのかどうか
H4 探索結果	boolean	H4 に基づいた探索によって発見されたのかどうか
H5 探索結果	boolean	H5 に基づいた探索によって発見されたのかどうか
H6 探索結果	boolean	H6 に基づいた探索によって発見されたのかどうか
各 URL に対する LA サーバの探索結果に関するログ		
名前	型	説明
URL	String	リンクオーソリティの探索を行った URL
Google 探索日時	Date	Google を利用した検索を最後に行った日時
Google 探索時間	int	Google を利用した検索に要した時間 (ミリ秒)
Alexa 探索日時	Date	Alexa を利用した検索を最後に行った日時
Alexa 探索時間	int	Alexa を利用した検索に要した時間 (ミリ秒)
同一サイト探索日時	Date	同一サイト探索を最後に行った日時
同一サイト探索時間	int	同一サイト探索に要した時間 (ミリ秒)
上位サイト探索日時	Date	上位サイト探索を最後に行った日時
上位サイト探索時間	int	上位サイト探索に要した時間 (ミリ秒)
相互リンク探索日時	Date	相互リンク探索を最後に行った日時
相互リンク探索時間	int	相互リンク探索に要した時間 (ミリ秒)
LA サーバが発見した各リンクオーソリティ候補に関するログ		
名前	型	説明
URL	String	リンクオーソリティ候補の URL
Google 探索結果	boolean	Google を利用した探索で発見されたのかどうか
Alexa 探索結果	boolean	Alexa を利用した探索で発見されたのかどうか
同一サイト探索結果	boolean	同一サイト内探索によって発見されたのかどうか
上位サイト探索結果	boolean	上位サイト探索によって発見されたのかどうか
相互リンク探索結果	boolean	相互リンク探索によって発見されたのかどうか
リンクの数	int	リンクの数はいくつか
デッドリンクの数	int	デッドリンクの数はいくつか
デッドリンクの割合	float	デッドリンクの割合はいくつか
最終発見日	Date	最後にこの候補を発見した日時はいつか
ページチェック	Date	最後にこの候補をチェックしたのはいつか
ページ状況	boolean	最後にこの候補をチェックしたときにページが存在したかどうか
ランク	int	候補のランクはいくつか

表 4 公開実験でログに記録する項目

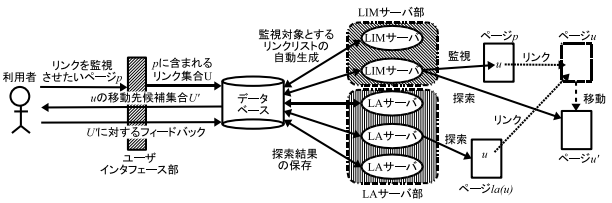

図 4 公開実験システム アーキテクチャ

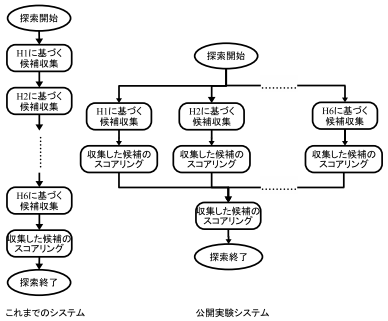

図 5 探索手順の比較

3.5 結果の解析

公開実験ではフィードバックを利用して次の解析を行う．

- ユーザから得たフードバックの総数 $feed\_all$，発見に成功した場合のフィードバックの総数 $feed\_success$ から，システムが Web ページの正しい移動先の発見に成功した割合がどの程度であるかを計算する．

$$\text{移動先ページ発見成功率 (\%)} = \frac{\text{feed\_success}}{\text{feed\_all}} \times 100$$

- フィードバックから得たリンクの種類を基に，リンクの種類に応じた発見成功数，発見失敗数の内訳がどのようになっているかを分析する．
- フィードバックから得た正しい移動先 URL と移動元 URL の比較などから，Web ページの正しい移動先が提供できなかった場合の原因が何かを考察する．
- キーワード検索で利用するためにシステムが選択したキーワードは適切であったかを考察する．

フィードバックと公開実験によって取得するログから次の解析を行う．

- Web ページの正しい移動先の発見に成功した場合の，各ヒューリスティクスの貢献度を分析する．
- Web ページの移動先の探索およびリンクオーソリティの探索に要する平均時間を分析する．
- リンクオーソリティの平均発見数を分析する．

4. 公開実験システムの設計と実装

3.2 節で述べたように，これまでの実験で利用していたシステムと公開実験システムでは利用方法が異なる．よって，これまでのシステムをそのまま利用するには次のような問題がある．それらの問題を解決するための工夫を施し，それによって設計される公開実験システムのアーキテクチャを図 4 に示す．

問題 1: 今までの実験では，我々が一定の規則に従って収集したリンクの集合からリンクリストを生成し，そのリンクリストを対象として実験を行っていた．リストの作成は一度だけで，以後リストは変化させなかった．また，数多くのリンクに対して短時間で探索するためにサーバを複数台用意して実験を行っていた．そしてリストが固定だったので，リストを単純に 3 分割して各サーバで探索を行っていた．しかし公開実験システムでは対象とするリンクが利用者の要求に応じてダイナミックに増減するため，今までのリストの分割手法では各サーバに対して対象となるリンクを適切に分配することができない．

問題 2: 今までの実験では，リンク切れが発生した際に正しいリンクの修正候補を発見することだけが目的だった．しかし公開実験システムではリンク切れが起きた際にリンクの修正候補を利用者に提供する必要があり，リンク切れが発生してからできるだけ短い時間で探索を行わなければならない．

問題 1 および問題 2 に対する工夫: 今までの実験では，各ヒューリスティクスに基づく探索の処理を順番に行い，最後にスコアリングを行っていた．公開実験システムではそれぞれの探索の処理を平行して行う．これによって時間あたりの探索効率を向上させる．今までのシステムと公開実験システムの違いを図 5 に示す．

また，今までの実験では，我々が LIM サーバおよび LA サーバに対してリンクリストを与えていたが，これでは監視対象ページのリンクの増減に対応できない．公開実験システムでは，図 4 に示すような複数の LIM サーバおよび LA サーバが自律的にリンクリストを作成する仕組みを用意する．この仕組みの例として，LIM サーバがリンク切れしたリンクに対して H1 に基づく探索を行う場合を示す．疑似コードによって示すと図 6 および図 7 のようになる．リンク切れしたリンクの集合 (コード中での Target) を表 5 に示す．そして過去に探索したリンクと探索日時 (コード中での status) を表 6 に示す．LIM サーバが H1 に基づく探索を開始するとき，まず sort メソッドを呼び出し，Target(表 5) と status(表 6) から表 7 のように

監視対象のリンク	リンク切れ日時
http://u1.jp	2005/6/1 12:00
http://u2.jp	2005/6/5 10:00
http://u3.jp	2005/6/11 20:00
http://u4.jp	2005/6/9 20:00
http://u5.jp	2005/6/4 10:00
http://u6.jp	2005/6/10 1:00
http://u7.jp	2005/6/10 12:00
http://u8.jp	2005/6/2 16:00
http://u9.jp	2005/6/8 23:00
http://u10.jp	2005/6/3 12:00

表 5 リンク切れしたリンク一覧

探索を行った URL	最終探索日時
http://u1.jp	2005/6/2 0:00
http://u2.jp	2005/6/6 0:00
http://u5.jp	2005/6/5 0:00
http://u8.jp	2005/6/3 0:00
http://u10.jp	2005/6/4 0:00

表 6 過去に LIM サーバが H1 に基づく探索を行ったリンク一覧

探索順序	探索のためのリンクの集合
1	http://u3.jp
2	http://u7.jp
3	http://u6.jp
4	http://u4.jp
5	http://u9.jp
6	http://u2.jp
7	http://u5.jp
8	http://u10.jp
9	http://u8.jp
10	http://u1.jp

表 7 ソートされたリンク一覧

```
1. List sort(List status) {
2.    List target = Target と status を結合した結果;
3.    target = 過去に探索を行っていないリンク，探索日時が古いリンクの
      順にソートした結果;
4.    return target;
5. }
```

図 6 擬似コード: 共通のソートメソッド

リンクをソートする．そして上位 N のリンクを取得し，このリンクをリンクリストと見なす．そしてリンクリストの各リンクについて探索を行う．擬似コードでは LIM サーバの H1 に基づく探索について示したが，LIM サーバおよび LA サーバのその他の探索においても同様である．

問題:3 今までは，我々以外がシステムを利用することを想定していなかったのでユーザビリティを考慮していなかった．しかし公開実験システムではできるだけ簡単に操作できるようにして多くの人に利用してもらいたい．

問題 3 に対する工夫: 公開実験システムでは，監視を行う Web ページの登録およびシステムの探索結果の閲覧のためのユーザインタフェースを提供する．これによってユーザビリティを向上させる．ユーザとシステムの間のインタラクションは以下の通りである．

```
1. Class LIM-H1 {
2.    main() {
3.       while() {
4.          List status = getH1StatusFromDB();
5.          List target = Top-N(sort(status));
6.          for(i = 0; i < N; i++) {
7.             target(i) について H1 に基づいた探索;
8.          }
9.       }
10.   }
11. }
```

図 7 擬似コード: LIM サーバの H1 に基づく探索

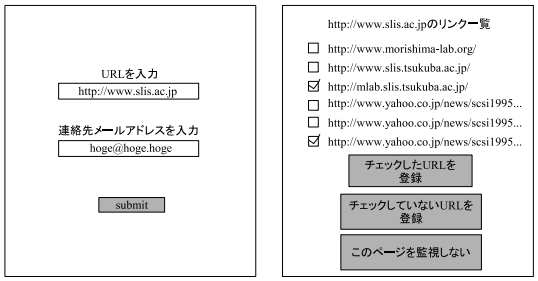

図 8 システムへの登録時の画面例

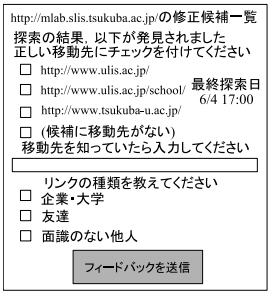

図 9 システムによる修正候補提供時の画面例

システムへの登録時:

(1) リンクの監視を希望する Web ページの URL p を送信する (図 8 画面例 1)

(2) p のページに含まれるリンク一覧が表示されるので，監視を行うリンクを指定する (図 8 画面例 2)

(3) システムへの登録が完了し，リンクの監視が行われる

監視していたリンクが切れたとき:

(1) p においてリンク切れが発生

(2) LIM サーバが p の移動先候補を計算

(3) システムから利用者に結果表示用の URL をメールで送信し，利用者は提示された結果をもとに p の移動先 p' を発見

(4) 利用者は結果に対するフィードバックを送信 (図 9)

問題 4: 今までの実験では，LIM サーバおよび LA

サーバはデータの受け渡しを行わず，人手によって行っていた．つまり，LIM サーバのヒューリスティクス H6 でリンクオーソリティを利用しているが，今までは LIM サーバが直接参照していたのではなく，間に人手が介入していた．何故なら，サーバを複数台利用していたがそれぞれが連携するような仕組みを用意していなかったからである．しかし公開実験システムでは利用者に対してできるだけ早く情報を提供する必要があり，いちいち人手を介入させるわけにはいかない．

問題 4 に対する工夫: 今までの実験では，各サーバの探索結果は各サーバごとにファイルに保存していた．このため，各サーバ同士が通信する仕組みがなかったのでデータの共有 (リンクオーソリティの参照など) ができなかった．公開実験システムでは各サーバの探索結果を一つのデータベースに格納することによって，各サーバ同士が通信を行うような特別な実装を行わなくてもデータの共有を行えるようにする．それによって LIM サーバの全ての探索を自動化することができる．

図 4 に示すように，公開実験システムの主要な構成要素はユーザインタフェース部および LIM サーバ部，LA サーバ部である．

ユーザインタフェース部: 利用者がシステムを利用するためのインタフェース．リンク切れを監視する Web ページの URL の登録，リンク切れ修正候補の提供，フィードバックの送信などを行う機能を提供する．ユーザインターフェース部は PHP によって実装され，Web ブラウザから利用することができる．

LIM サーバ部: 通常はリンク切れが発生していないかどうかをチェックしている．リンク切れを発見すると，リンク先の Web ページの移動先を探索する．その際，リンクオーソリティの情報も利用する．LIM サーバ部は Java によって実装を行う．また，探索は複数のサーバを利用して行う．探索の過程で利用する検索エンジンは，Google[11)]，および Yahoo[12)] である．

LA サーバ部: 常に各リンク先のページに対するリンクオーソリティを計算する．LA サーバ部は，Java によって実装を行う．また，探索は複数のサーバを利用して行う．探索の過程で利用する検索エンジンは，Google[11)]，Alexa[13)] である．

5. おわりに

本稿では，我々が開発しているリンク一貫性維持支援システムの公開実験について述べた．特に，公開実験の目的，結果の解析方法，公開実験用システムの開発における問題点と工夫について説明した．今後は公開実験を実施し，実験結果を基にしてリンク切れの修正精度の向上をはかる予定である．

謝　辞

システムに関するご助言をいただきました中溝昌佳氏に感謝致します．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田畑孝一教授，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．

参考文献

1) 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.
2) 中溝昌佳, 飯田敏成, 森嶋厚行, 杉本重雄, 北川博之, 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. 電子情報通信学会第 16 回データ工学ワークショップ (DEWS2005), 7 pages, 2005 年 3 月.
3) 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.
4) 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.
5) 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.
6) M.Beynon, A.Flegg: Guaranteeing Hypertext Link Integrity. US Patent Application Publication, US 2005/0021997 A1, Jan, 2005.
7) M.Beynon, A.Flegg: Hypertext Request Integrity and User Experience. US Patent Application Publication, US 2004/0267726 A1, Dec, 2004.
8) 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)
9) 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. 電子情

報通信学会第 15 回データ工学ワークショップ
(DEWS2004), 2004 年 3 月.
10) Hugh C. Davis: Hypertext link integrity. ACM
Comput. Surv. 31(4es): 28 (1999)
11) Google Web APIs:
http://www.google.com/apis/.
12) Yahoo! Search Web Services:
http://developer.yahoo.net/.
13) Alexa Web Information Service:
http://pages.alexa.com/prod_serv/WebInfoService.html.