デジタル・ドキュメント 17−5
（1999. 3. 12）

複数文書間のハイパーリンク自動生成とメンテナンス

石田 和生　　市山 俊治
{ishidakz, ichiyama}@hml.cl.nec.co.jp
NEC ヒューマンメディア研究所
〒630-0101 奈良県生駒市高山町 8916-47

電子図書館などのように大量のデータを蓄積したシステムからの情報検索を支援するために、関連のあるキーワード間あるいは文書間でのハイパーリンク生成などが研究されている。一般に、ハイパーリンクの生成は対象とする文書セットに依存しているため、文書の追加や削除が行われると、それに伴ってリンク再生成のために多大な時間が必要となる。本研究では、リンク元語句の前後情報を利用して精度の高いリンク生成を行うシステムを構築し、さらに、リンク生成に必要な情報を保持するインデックスファイルを用いる方式を提案し、リンク再生成に必要な時間の短縮を実現した。また、ハイパーリンクに複数のリンク先情報を埋め込む事で、文書削除によって発生する不正リンクの修復を実現した。

Generating and Maintaining Hyperlinks for Documents

Kazuo Ishida and Shunji Ichiyama
Human Media Research Laboratories, NEC Corporation
8916-47 Takayama-chou, Ikoma, Nara 630-0101, JAPAN

Hyperlink generation between the relative keywords or documents is developed to support an information retrieval from the system which accumulated a great deal of data like the digital library. Generally, when the addition and the deletion of the document are done, the great time to re-generate links is necessary, because it depends on the document set to deal with. In this paper, we suggest the methods to use the words around the link source and to use the index file which has the information necessary to generate links. As a result, it is achieved to generate high precision hyperlinks and to reduce the time necessary to re-generate links. Also, we can restore the unjust link which occurs by the document deletion to embed more than one destination of link in the hyperlink.

1. はじめに

電子図書館システムは、電子的に文書を蓄積しておきユーザが自由に検索や閲覧を行えるようにするものであるが、この電子図書館システムを構築し活用するためには、大量の文書データを電子化してシステムに蓄積しなければならない。一方、情報を単に蓄積するだけでは、ユーザが必要とするデータを迅速かつ手軽に引き出すことが出来なくなるという問題がある。このため、蓄積した情報を効率的に検索、利用するための手段としてキーワード検索をはじめとする様々な検索手法が存在する。また、文書を電子図書館システムに蓄積する際、複数の文書間で関連する情報をリンク付けすることで、ユーザは情報を閲覧中に関連する文書を次々とたどっていくことが出来るようになり、情報検索の効率が飛躍的にアップする。

そこで我々は、plain テキスト群あるいは HTML テキストファイル群からキーワードの抽出を行い、関連する語句間にハイパーリンクを自動生成するシステムのプロトタイプを作成している[1]。このシステムはテキストデータ群からキーワードを抽出し、各キーワードの重要度に応じてスコアを計算することが可能であり、また、関連する語句間にハイパーリンクを生成するものである。

ハイパーリンクの生成を、単純にリンク生成対象となっているキーワードの重みだけをもとに行ったのでは、リンク元およびリンク先語句の候補が十分に絞り切れず、膨大な量のリンクが生成されてしまうという問題がある。そこで、本研究で開発しているプロトタイプシステムではリンク適合度と呼ぶものを定義し、その計算方法を自由に定義できるようにすることでリンク生成の条件を様々にコントロール可能にしている。しかしリンク適合度の計算には、形態素解析や抽出されたキーワードの並び方に依存した計算が必要であるため、ハイパーリンクの生成に多大な時間を必要とするという問題がある。

ハイパーリンクの自動生成と同じく重要な事柄として、生成したリンクのメンテナンス作業が挙げられる。電子図書館システムでは、必ずしも恒常的な文書だけでなく、常に追加や削除が行われる文書も取り扱いの対象になると考えられるが、追加や削除が行われると生成されたリンクに不整合が生じ、リンク先が不適切、あるいはたどれないといった状況になりうる。これらのリンクを適切なものにするためのメンテナンス作業は一般に大きな労力を必要とする[2]。

以上のことから本研究では、リンク生成の高速化と文書の追加削除に伴うリンクメンテナンスを行う手法の提案とシステムの実装を行ったのでこれについて報告する。

以下、第2章では本システムが用いているリンク生成手法について説明し、第3章で高速化とメンテナンスの手法について述べる。第4章では試作したシステムの実行結果を示し、最後に本研究をまとめる。

2. ハイパーリンクの自動生成

2.1 既存WWWページのリンクパターン調査

plain テキスト群あるいは HTML 文書群に対してハイパーリンクの自動生成を行う場合、単純に文書中に含まれる同じキーワード間でリンクを生成したのでは膨大な量のリンクが生成されることになる。しかも生成されたハイパーリンクのリンク先文書が、リンク元文書を理解する上で補助的な情報とならないなど、ユーザにとって有益ではない可能性もある。これを改善するためには、注目しているキーワードだけでなくその前後の語句などの情報を有効に利用する必要があると考えた。そこで、既存のハイパーテキストデータに含まれているハイパーリンクのパターンについて調査を行い、リンク生成に利用できる情報の整理を行った。調査は主に

- リンク元語句の周辺情報
- リンク先文書に含まれる語句情報

を中心に行った。より具体的にいうと、リンク元語句については、リンク元語句の範囲(リンクのアンカーが設定されている部分の範囲)、語の種類(単独のキーワードなのか、文章になっているのか)、前後の情報(リンク元語句とその語句の前後に記述されている情報の関連性)の調査、リン

表 2-1 リンク元語句と前後情報の関係

リンク元ワードの説明文	275
リンク元に関連するキーワード	35
全く関連のない情報	16

表 2-2 リンク元語句のリンク先文書での出現場所

a.ページのタイトルに存在	142
b.ページの先頭に存在	227
c.見出しとして存在	26
d.箇条書きの項目として存在	3
e.説明文中に存在	43
f.画像データとして存在	28
g.次のリンクへのリンク元として存在	3
h.電子メール送信として存在	95
i.ファイルアーカイブとして存在	0
z.その他	0

ク先文書については、リンク先ページの概要(どういった内容のページであるのか)、リンク元語句が出現している場所についての調査を行った。調査する対象データとしては、ハイパーテキストデータが大量に手に入る場所ということで、YAHOO! JAPAN[3]に登録されている WWW ページの中から「コンピュータとインターネット」カテゴリの「インターネット」「オペレーティングシステム」「音楽」「雑誌」などに登録されている HTML 文書を無作為に合計 31 ファイル(文書中に含まれるハイパーリンクの総数は 329)選んだ。

調査結果のうち、ハイパーリンク生成と特に関係があると判断した「リンク元語句とその語句の前後に記述されている情報との関係」と「リンク元語句がリンク先文書中で出現している場所」についての結果を表 2-1と表 2-2に示す。これらの結果から、リンク元語句の前後の情報がリンク元と関連を持つ(説明文、あるいは関連するキーワードの場合の 310 件)のは全体の約 95%であることが分かった。さらに、リンク元語句がリンク先ページのタイトル部分、ページの先頭部分、見出しのいずれかに出現している件数は 285 件(重複分を除いて加算)で、全体に対する割合は約 86%であることも判明した。

以上の調査結果をもとに、ユーザにとって有益なリンクを生成するための指標について検討した。ここで言う「有益なリンク」とは注目しているリンク元文書の話題と関連性の高い文書がリンク先に存在しているリンクのことで、このようなリンクはユーザがリンク元文書の内容を理解するのを手助けすることが可能であると考える。

まず、文章と文章の話題の関連性の高さと、それぞれの文章中に出現する語句同士の関連度の高さに相関があると仮定する。表 2-1によれば、リンク元語句の周辺にはリンク元語句と関連の深い語句がある可能性が高いので、リンク元語句の周辺語句とリンク先文書中の語句との関連度を調べれば、リンク元文書とリンク先文書の話題の関連性を判断する指標になりうる。また、表 2-2の結果からは、リンク元語句がリンク先文書中に出現している場所に特徴があることが分かる。以上のことから

- リンク元語句の前後情報とリンク先文書との関連度
- リンク元語句がリンク先文書中で出現している場所(特にタイトルやページの先頭、見出し部分など)

をリンク生成の指標として定めた。しかし、前者の関連度計算のためには語句と語句の関連性を表す指標(例えばシソーラス)などを用いたり、後者の場所の判断には文書構造の解析が必要となるので、どちらも計算量が大きくなりがちである。そこで本研究では上記の基準を簡略化し、

- リンク元語句の前後の語句とリンク先文書に出現する語句との一致度
- リンク元語句がリンク先文書に出現している頻度

という基準を用いることにする。

2.2 ハイパーリンク生成手順

HTML 文書で使用されているハイパーリンクは通常、語句から文書全体、あるいは語句から文

書の特定部分へはられていることが多い。しかし、リンク先となる文書が論文や物語り全体のように、長さがあまり短いものでない場合、文書全体へのリンクを生成してもリンク先の全体像がつかめないのであまり有用であるとは言えない。そこで本研究では主に、語句、あるいは段落といった比較的小規模なブロックをリンク先の対象として扱うことにする。

ハイパーリンクの生成は、大まかには以下のような手順で行う。

1. 入力文書に対し形態素解析を行い、文書を語句に分割
2. 分割された語句の中から出現頻度などを利用して重要な語句を選択し、キーワードとして抽出
3. リンク元文書中に存在するキーワードとリンク先文書中に存在するキーワード、あるいはブロックとの間にリンクを生成するかどうかを判別する評価値(以下では、これをリンク適合度と呼ぶ)を計算
4. リンク適合度が閾値を超えていればリンクを生成

生成されたリンクは、入力文書中に HTML のアンカータグ形式(<A>と</A>)で埋め込む。このような形式で埋め込むことのメリットは、生成されたハイパーテキストを既存の HTML ビューアで閲覧することが可能となる点である。

以上で説明した手順の中で、生成されるリンクの質を決めるものがリンク適合度の計算方法である。この計算は、2.1節で述べた既存 WWW ページのリンクパターン調査結果から得られた知見をもとに行う。すなわち、リンク元語句がリンク先文書中に出現している頻度とリンク元語句の前後情報とリンク先文書の関連度を利用して計算する。しかし基本的に、どのような入力文書に対しても最適となるようなリンク適合度の計算方法というものは存在しないと考えている。これは、対象文書の読み手としてどういった層をターゲットにしているのかによって、生成すべきリンクが異なってくることからも容易に想像がつく(例えば、初心者向けの文書であれば、専門用語などからその説明文へのリンクを生成することが有用であるが、専門家向けの文書で同じことを行うと、読み手にとってあまり有用でないリンクが多数存在することになる)。このため、対象とする文書の種類や対象ユーザ毎にリンク適合度の計算方法をカスタマイズできるほうが利用価値が高い。そこで本研究ではリンク適合度の計算方法をリンク辞書と呼ばれる定義ファイルに記述しておき、その内容に従ってリンクを生成する[1]。

リンク辞書に記述できるリンク適合度計算式は、コンピュータ言語のひとつ perl で記述出来る演算式の範囲で自由に定義できる。例えば、リンク辞書に

```
score=sum(*i,-2,2,1,"(src_w(0) eq des_w(¥$i))")
```

のように記述すると、リンク先語句の前後 2 ワード中にリンク元語句と同じ語句がいくつ含まれているかによってリンク適合度を計算することを表す。ここで、src_w(n)と des_w(n)はそれぞれ、現在注目しているリンク元語句、あるいはリンク先語句から n 番目にある語句を返す関数で、リンク辞書の表現を簡単にするために用意した関数である。

本方式によると、単純に同じ語句同士でハイパーリンクを生成した場合に比べ、生成されるリンク数を 1/5 にまで低減できた。さらに、本方式によって生成されなかった残りの 4/5 のリンクのうち、92%は生成されないことが妥当であると判断された。すなわち、本方式により精度の高いハイパーリンクの自動生成が実現できた。

3. ハイパーリンクのメンテナンス

3.1 メンテナンスの必要性

一般に、ハイパーリンクの自動生成は、対象となる文書群に依存しているため、文書群に対し文書の追加あるいは削除が行われた場合には、ハイパーリンクの生成をやり直す必要がある。例えば、ある文書が追加された場合には、その文書から他の文書、あるいは他の文書からその文書へのリンクを設定してやらなければ、追加された文書

は他の文書とのつながりを持たない孤立したものとなる。また、文書が削除された場合には、その文書をリンク先として持つハイパーリンクがジャンプ不可能なリンクとして残存することとなり、ユーザに無意味な情報(たどれないリンク)を与えることとなる。従って、文書の追加や削除に伴い、ハイパーリンクの再生成を行う必要がある。

しかし、本研究の自動生成手法は形態素解析とリンク適合度の計算部分が比較的重い処理のため、対象文書群の規模が大きい場合には、文書の追加や削除のたびにリンクの再生成を行わせるのはあまり実用的ではない。次節以降では、文書の追加と削除の際に効率的にリンクの再生成を行う手法について述べる。

3.2 インデックスファイルによるリンク生成の高速化

文書の追加に伴うハイパーリンクの再生成は、文書群全体に対してではなく、「追加された文書からその他の文書へ」と「その他の文書から追加された文書へ」の 2 通りについてのみリンクの生成を行うことで時間の短縮が可能となる。また、形態素解析やキーワード抽出などは各文書毎に独立したものなので、ハイパーリンクの生成に先立ち予め実行させておくことが可能である。前者のリンク生成対象の限定については、ハイパーリンク生成時に対象ファイルを指定するだけで実現される事柄なので、以下では、後者の独立した情報保持の実現のために新たに規定したインデックスファイルとそれを用いたリンク生成の高速化について述べる。

リンク適合度の計算では、リンク元とリンク先にある語句とその重要度(以下ではスコアと呼ぶ)を使い、リンク辞書に記述された内容に従ってあるブロック単位で演算を行うので、以下の項目についての情報が必須となる。

- キーワード、およびその品詞とスコア
- キーワードの出現場所
- ブロックの区切りの場所

従って、これらの情報を予め用意しておけば、リンク適合度の再計算をする際に元の文書へのアクセスや形態素解析などが必要なくなり、リンク生成が高速化できる。本研究では、これらの情報を蓄えておくファイルをインデックスファイルと呼び、図 3-1に示すようなフォーマットで記述することにした。インデックスファイルの各項目について以下で説明する。

```
[Keyword]
サービス:サ変:1.101800
通信:サ変:1.109000
コンピュータ:名詞:2.209000
ビジネス:名詞:2.303600
情報:名詞:4.510800
                (中略)
[Stream]
8,6,8,8,8,9,1,2,4,7,8,6,3,10,3,3,2,2,3,2,6,8,8,6,…

[Block]
block=¥n
0:1,1:36,1:37,3:58,3:59,4:94,6:329,23:1040,…
```

図 3-1 インデックスファイルの例

インデックスファイルは大きく分けて Keyword, Stream, Block の 3 つのセクションに分かれている。Keyword セクションには、抽出されたキーワードの語、品詞、スコアを記述している。Stream セクションは Keyword セクションで記述されたキーワードが文書中にどのような順序で出現しているかを表している。それぞれの数字は Keyword セクションの何番目のキーワード(図 3-1の例でいうと、1 は「サービス」、2 は「通信」を表す)であるかを示している。Block セクションはブロックの区切りの場所を記述する。ひとつの場所は n:m という形式で表現され、n 番目のキーワードと(n+1)番目のキーワードの間に区切りが存在することを表している。m はその区切りがファイルの先頭から m バイト目の位置にあることを示しており、n で区切りの場所が特定できない場合に使用される。例えば、ブロックの区切りが「。」である場合に

電話です。分かりました。通信します。

といった文書を処理して「電話」と「通信」というキーワードが抽出できたとすると、この文書のキーワードとブロック区切りの並びは

```
DES_FILE=D:/data/file1.hyp
DES_FILE=D:/data/file2.hyp
DES_FILE=D:/data/file3.hyp
```

図 3-2　リンク情報ファイルの例

```
宅配便会社の急成長は
<A HREF = "ugoku.hyp#LNK0001" link =
"ugoku.hyp#LNK0001, tsuushin.hyp#LNK0004"
>電話</A>のおかげである。
```

図 3-3　複数リンクの埋め込み例

電話 。 。 通信 。

のようになる。これをインデックスファイルに記述すると、ブロック区切りの場所を示す n はそれぞれ、1, 1, 2 となり、1 つめと 2 つめの区切りの順序が特定できなくなる。こういった場合に順序を特定するため、先頭からのバイト数も同時に記録しておく。

以上で述べたようなインデックスファイルを予め用意しておけば、リンク適合度計算時に必要な形態素解析とキーワード抽出、ブロックの判別などが元の文書にアクセスせずとも非常に簡単に行えるようになるため、リンク生成の高速化が実現される。これにより文書追加時のメンテナンス作業に要する時間の短縮が可能となる。

3.3 文書削除への対応

文書が削除された場合には、リンク先がなくなったハイパーリンクの検出と新たなリンク生成を行う必要がある。この節ではこれらの作業を高速に行うための手法について説明する。

リンク先がなくなったハイパーリンクの検出は、基本的には、全文書の中身を全て探索すれば実行できるが、あまり効率的ではない。これは文書の長さが長くなればなるほど探索に要する時間が増大するからである。しかしリンク先として存在しうる文書の数は高々文書群全体のファイル数である。そこで予め、ある文書中に存在するハイパーリンクのリンク先文書の名前をファイルに保存しておくことにする。以下ではこのために用いるファイルをリンク情報ファイルと呼ぶ(図 3-2 参照)。文書が削除された場合には、リンク情報ファイル中に削除された文書の名前が含まれているかどうかをチェックして、含まれている場合のみ元ファイルの中身を探索する。このようにすれば削除されたファイルに無関係なファイルの中身を探索する必要がなくなり、メンテナンス対象となるリンクの探索に要する時間を短縮することが可能となる。

次に、リンク先のないハイパーリンクを発見した場合の対処について述べる。リンク先がないハイパーリンクはそのまま放置するとユーザに無意味な情報を提示することとなり、あまり好ましくない。対処法としては

- そのリンクを削除
- 新しいリンク先を見つける

の 2 通りが考えられるが、単純にリンクを削除してしまったのではユーザに提示すべき情報が減ることになるため、新しいリンク先を用意するほうが望ましい。新しいリンク先の探索方法としては、リンク先が削除されたハイパーリンクのリンク元語句から他の文書へのリンク生成を通常通り行ってやれば良いが、ここではより処理の簡潔な方法で対処することにする。すなわち予め、リンク先の候補を複数用意しておきリンクのアンカー部分に埋め込んでおくというものである。図 3-3 の例で説明すると、アンカータグの link 属性に設定されている値が、埋め込まれた複数のリンク先候補である。このとき、文書 ugoku.hyp が削除されると、代替リンクとして tsuushin.hyp#LNK0004 が用いられることになる。また、この手法では、候補のリンク先をハイパーリンクの自動生成時に、リンク適合度が閾値を超えたものをすべて、あるいはリンク適合度の大きい順にいくつか選択することで用意することが出来るため、リンク生成時に、新たな大きな処理が加わることもないという利点もある。

以上のようにすれば、新リンクの探索に要する時間はほとんど必要なくなり、メンテナンス作業の効率も非常に良くなる。ただしこの方法だと、

```
link=:名詞@0.5+ block@3+
cond="src_w(0) eq des_w(0)"
score="kanren(2,blk_b(0),blk_e(0))"
```

図 4-1　使用したリンク辞書(一部)

文書の削除が何度か行われると、予めアンカー部分に埋め込んであるリンク先候補を全て使い果たし、新たなリンク先を選択することが出来なくなるという問題点もある。このような場合には、最初に述べたように、そのリンクのリンク元語句から他の文書へのリンクを通常通り生成することで対処が可能である。

4.　システムの実行例

前章までで述べたインデックスファイルを用いたハイパーリンク生成の高速化と文書の削除に伴うリンクメンテナンスを実現するシステムを今回試作したので、その実行例について述べる。なお、開発したシステムのプラットフォームは CPU: Pentium200MHz, メモリ: 96MB, OS: Windows NT4.0 である。また、入力対象文書については、文庫本を選択した。これは、文庫本は基本的に文字がベースであるため、テキスト処理を行う本システムの性能評価に適していると考えられるからである。使用する文書として C&C 文庫から、お互いにある程度関連のありそうな、「動く電話」(森島 著)、「電話が変わる」(榎本 著)、「パソコン通信入門」(小村 著)の 3 冊を選び、それぞれの文献の先頭から約 2000 文字分をテキストデータとして手で打ち込み、入力データとして用いた。

リンク辞書は、2.1節のリンクパターン調査結果を参考にして、リンク元語句の周辺情報とリンク先文書の関連度をもとにリンク適合度を計算するように定義したものを用いた。参考までにリンク辞書の一部を図 4-1に示す。このリンク適合度計算式は、リンク元の前後 2 ワード分の語句がリンク先のブロック(段落)中にいくつ出現しているかを計算している。

電話線の両端が<A HREF = "tsuushin.hyp#LNK0001" link = "tsuushin.hyp#LNK0001, ugoku.hyp#LNK0001" >電話</A>機につないであるか、<A HREF = "ugoku.hyp#LNK0002" link = "ugoku.hyp#LNK0002, tsuushin.hyp#LNK0001" >通信</A>線の回路のなかに<A HREF = "tsuushin.hyp#LNK0002" link = "tsuushin.hyp#LNK0002" >コンピュータ</A>が割り込んでいるかどうかの違いだけである。あとで詳述するが、ＶＡＮ（付加価値通信網）などと呼ばれ、さまざまなデータ、<A HREF = "tsuushin.hyp#LNK0001" link = "tsuushin.hyp#LNK0001" >情報</A>が数字に変えられて電話線を通っていく。

ハイテクの技術によって、ふつうの<A HREF = "tsuushin.hyp#LNK0001" link = "tsuushin.hyp#LNK0001" >電話</A>で、

「もしもし、こんにちは」

といっている間に、このＶＡＮの回線だと、驚くほどの情報が“詰め込まれて”送れる、ということだけを知っておいていただこう。つまり、クロネコやペリカンの会社は、あの膨大な量の荷物をさばくというビジネスをしながら、そのための情報代は格安になっている。

図 4-2　リンク生成実行結果(リンク元文書)

<A NAME="LNK0002">ところが、そのようなコンピュータも通信と結びつくことで、より日常的で人間的な世界をつくりだせるようになった。それがパソコン通信／ワープロ通信と呼ばれるものである。コンピュータが単なるデータ処理マシンとしてではなく、人間同士の双方向のやりとりを媒介するためのコミュニケーション・メディアとなったわけだ。パソコン通信ではコンピュータの向こう側に確かに人がいる。本当の高度情報化というのは情報装置の高度化だけではなく、人間が主役になって情報を高度に活用することであり、それをつうじて人間性を高め、人間同士の社会的な結びつきが強められるものでなければならないだろう。その意味で、パソコン通信のもつ双方向メディアとしての役割は社会的にも非常に注目を集めている。
</A>
時間と地球を超えるコミュニケーション
<A NAME="LNK0001">では、なぜ双方向メディアというのだろうか。パソコン通信の仕組みを簡単に説明しておこう。パソコン通信というのは、…

(以下省略)

図 4-3　リンク生成実行結果(リンク先文書)

4.1　インデックスファイルによるリンク生成

以上で述べた条件で、ハイパーリンクの生成を行った結果を図 4-2と図 4-3に示す。またインデックスファイルを用いたことによる高速化の実証のため、インデックスファイルを用いずにリンク生成を行った場合とインデックスファイルを用いて生成した場合との実行時間の比較を行った。この結果、インデックスファイルを用いない場合はハイパーリンクの生成に約 36 秒かかっていたのが、インデックスファイルを用いることにより約 23 秒に短縮された。このことからインデックスファイルによる約 36%の高速化が実証された。

4.2　ファイル削除に対するリンクメンテナンス

次に、ファイルを削除したときのハイパーリンクのメンテナンス実行結果を示す。入力対象である 3 冊の文庫本に対し、ハイパーリンク生成を行った結果(図 4-2と図 4-3)が得られている状

電話線の両端が<A HREF = "tsuushin.hyp#LNK0001" link = "tsuushin.hyp#LNK0001" >電話</A>機につないであるか、<A HREF = "tsuushin.hyp#LNK0001" link = "tsuushin.hyp#LNK0001" >通信</A>線の回路のなかに<A HREF = "tsuushin.hyp#LNK0002" link = "tsuushin.hyp#LNK0002" >コンピュータ</A>が割り込んでいるかどうかの違いだけである。あとで詳述するが、ＶＡＮ（付加価値通信網）などと呼ばれ、さまざまなデータ、<A HREF = "tsuushin.hyp#LNK0001" link = "tsuushin.hyp#LNK0001" >情報</A>が数字に変えられて電話線を通っていく。

ハイテクの技術によって、ふつうの<A HREF = "tsuushin.hyp#LNK0001" link = "tsuushin.hyp#LNK0001" >電話</A>で、

「もしもし、こんにちは」

といっている間に、このＶＡＮの回線だと、驚くほどの情報が“詰め込まれて”送れる、ということだけを知っておいていただこう。つまり、クロネコやペリカンの会社は、あの膨大な量の荷物をさばくというビジネスをしながら、そのための情報代は格安になっている。

図 4-4　メンテナンス実行結果(リンク元文書)

態で、「動く電話」(ファイル名 ugoku.hyp)のファイルを削除する。この状態では、図 4-2のリンク元文書 4 行目にある「通信」という語句からはられているハイパーリンクのリンク先がたどれない。ここでリンクメンテナンスを実行させた結果を図 4-4に示す。この結果を見れば、たどれなくなっていた「通信」からのリンク先として別ファイル(tsuushin.hyp)へのリンクに変更されていることが分かる。また、2 行目の「電話」に設定されていたリンク先候補(アンカータグの link 属性値)の中からも削除されたファイルへのリンクが正しく消去されていることが分かる。

4.3 考察

以上の実行結果から、今回試作したシステムを用いることで

- 複数文書間での精度の高いハイパーリンクの自動生成を実現
- インデックスファイルを用いたリンク生成の高速化(約 36%)
- 文書の削除によってたどれなくなるリンクの修復

が実現できた。しかし本システムのリンク生成はまだ実用に耐えうるほど高速化されたわけではないので、今後更なる高速化が必要であろう。

速度の低下に最も影響するのがリンク適合度の計算である。現状では語句と語句、あるいは語句とブロック全ての組み合わせについて計算を行っているため時間がかかっている。しかも、同じ語句同士の組み合わせについても繰り返し計算を行っている。この点を改善し、一度計算した組み合わせパターンと同じパターンが出てきたときには計算をスキップするなどの工夫を行えば、更なる高速化が実現できると考えている。

リンクのメンテナンスにおいても予め用意しておくリンク候補の選択にはまだ問題が残っている。例えば図 4-2の文書において、ファイル tsuushin.hyp が削除されたとすると 5 つあるリンクのうち 3 つが代替リンクの選択に失敗することになってしまう。これは、文書中の接近した場所にある語句は互いに関連があることが多いため、そのリンク先も同じファイルに偏ってしまう傾向があるためと思われる。これについては、リンク先候補をなるべく複数のファイルに分散させて選択するなどの工夫が必要であると考える。

5. おわりに

本研究では、複数のテキスト文書群に対してハイパーテキストの自動生成を行うシステムにおいて、前後の語句情報を利用した、リンク生成の精度向上と、インデックスファイルを用いた実行時間の短縮を実現した。また、文書削除に伴い発生する不正なリンクの修復を実現する手法の提案とシステム実装を行った。これによりテキスト文書のハイパーテキスト化、および生成されたハイパーテキストのメンテナンス作業が大幅に軽減されると期待される。

本研究は日本情報処理開発協会(JIPDEC)による次世代電子図書館システム研究開発事業の一環として、次世代電子図書館システム実現のための個別技術とその実装技術の開発のために行っている。

参考文献

[1] 石田 他: 検索情報抽出の研究開発, 次世代電子図書館システム研究開発事業論文集, pp. 89-92, 1998.

[2] M. S. Ackerman, R. T. Fielding, Collection Maintenance in the Digital Library, http://csdl.tamu.edu/DL95/papers/ackerman/ackerman.html, 1995.

[3] YAHOO! JAPAN: http://www.yahoo.co.jp/