C2-5 検索サイトラッパー検証のための検索結果件数推定方法

酒井 美由紀† 廣川 佐千男‡

†九州大学システム情報科学府 ‡九州大学情報基盤センター

概要

自サイト内など限られた領域を対象にした検索サイトが増大している。これらの検索サイトを効率よく利用するためには統合する必要があるが、増加する検索サイトに対して手動で統合を行うことは困難である。我々は自動的なラッパー生成による統合検索システム構築の研究を行っている。自動的に生成したラッパーに対しては、その精度を検証することが不可欠である。また、生成と同様に検証も自動的に行う必要がある。ラッパー検証の手法のひとつとして、検索サイトの返す検索結果の件数とラッパーを用いて抽出したレコードの件数の比較をすることが考えられる。本稿では、検索サイトの返す検索結果の件数を、ファイルサイズやリンク数を基にして自動的に推定する方法を提案し、複数の検索サイトに対する実験により、検索結果件数推定方法の有効性を示す。

1 はじめに

Web 上に公開される文書が膨大になるに従って、必要な情報を発見することが困難になっている。そこで、Google や Yahoo!などの一般検索エンジンは不可欠なものとなっている。しかし、これらの一般検索エンジンは検索範囲を Web 全体を対象とするので、得られる情報が多過ぎて、検索結果の品質が問題となっている。

Web 全体を検索の対象とする一般検索エンジンとは逆に、自サイト内文書などの限られた領域を検索対象とするサイトがある。これらは、Invisible Web[2, 8]、Deep Web[1] あるいは Hidden Web [5, 6] とよばれる。本論文では専門検索サイト、あるいは単に、検索サイトと呼ぶ。専門検索サイトの背後には個別のデータベースがあり、その情報は一般検索エンジンでは得られない。このようなデータベースは、そのサイトを運用している企業や組織あるいは個人が、利用者に対して積極的に責任をもって提供しているものであり、信頼できる高品質な情報と考えられる。得られるのは一ページであっても、その背景には同種のデータが潜在的にあると想定できるので、一般検索エンジンにより Web 上から得られる一ページと比較し高品質といえる。

このような専門検索サイト群を目的に応じて、統合検索できれば高品質な情報を効率よく利用することができる。一般に、複数の検索エンジンを統合する検索システムは、メタ検索エンジンと呼ばれる [9]。個々の専門検索エンジンは異なる検索形式と出力形式を用いているため、その統合には、検索エンジンごとに個別のソフトウェア部品（ラッパー）が必要となる。

これまで、数種類の一般検索エンジンを対象としたメタ検索エンジンが知られている。そこで、使われているラッパーは、人手で実装されたものである。少数の一般検索エンジンだけでなく、１０万以上あるといわれる検索サイトを統合対象とすれば、人手による実装には限界がある。このように、対象規模の増加と、各検索サイトの変化に対応するためには、ラッパーの自動生成が必須である。

また、一般の Web ページがそうであるように、検索エンジンの検索形式や出力形式も頻繁に変更されるので、一度作成したラッパーにより検索結果の抽出が正しくできているかどうかを、定期的に確認しなければならない。もし、うまくいっていないことがわかれば、再度作り直さなければならない。このようなラッパーの検証とメンテナンスにおいても自動化が必須である。いくつかの一般検索エンジンに対するラッパーは、Perl のライブラリとして CPAN で公開されているが、そのメンテナンスは個人の努力によるものであり更新には限界がある [3]。

我々は自動的なラッパー生成による統合検索サイト構築の研究を行っている [10, 7, 4]。本論文では、検索結果の一ページに含まれる検索結果の件数を自動的に

推定する方式を提案する。ラッパーにより得られる件数と比較することにより、ラッパーの検証とメンテナンスを自動化することができる。

2　統合検索とラッパーの自動生成

統合検索は、複数の検索サイトにキーワードを与え、それぞれのサイトから返って来た結果ページの中に含まれる検索結果の部分のみをラッパーを用いて切り出し、それらを組み合わせて統合検索の結果として表示する [**図 1**]。

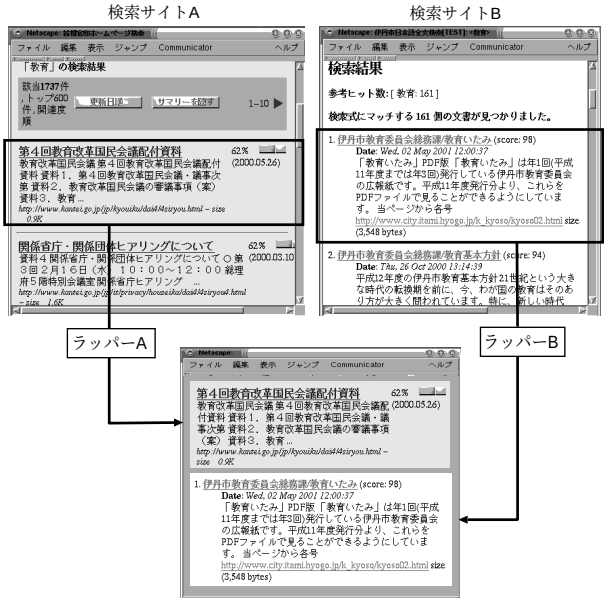


図 1: ラッパーによる検索結果切り出し

検索結果の件数が多い場合には、複数ページにわたって結果が表示される場合があるが、統合検索におけるラッパーでは、結果切り出しの対象を返って来た結果ページの最初の 1 ページ目のみとしている。複数の検索サイトによる結果を統合するため、個々のサイトからの上位数件の結果で十分であるからである。

同じ検索サイトであれば、どのキーワードに対する結果でも、常にページのフォーマットは同じである。しかし、異なるサイト間ではページのフォーマットが異なり、検索結果の表示方法や表示項目が異なってしまう。

そのため、検索サイトの結果ページから結果部分を切り出すためのラッパーは、1 つ 1 つのサイトに対応するものが必要となる。しかし、多くの検索サイトについて個々に対応するラッパーを全て人手で作ることは困難である。そこで、検索サイトの統合を行う際には自動でのラッパー生成が必須となる。

自動生成されたラッパーに対しては、1 ページ目の検索結果の全てを正しく切り出せているのか、その精度の検証を行わなければならない。

人手で、ラッパーで切り出した検索結果の件数と実際の結果ページの 1 ページ目に表示されている検索結果の件数とを比較して検証することも方法の 1 つであるが、ラッパーを 1 つ 1 つ手動で作るのと同様、非常に困難である。

そこで、返ってきた検索結果の 1 ページ目に何件の検索結果が含まれているかをページ中の記述を用いずに、自動的に推定することができれば、ラッパーによる切り出し件数との比較を行うことによってラッパーの自動検証が可能になる [**図 2**]。そのため、検索結果の 1 ページ目の件数の自動推定が必要であると考える。

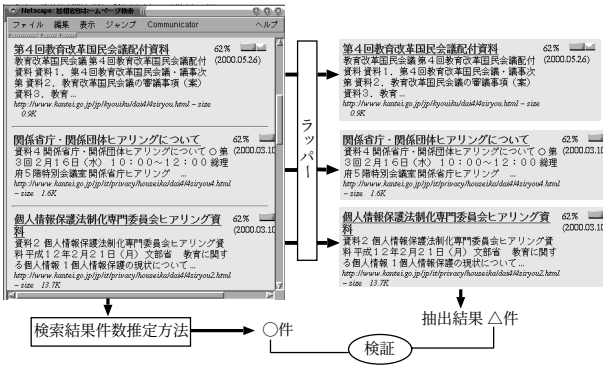


図 2: 検索結果推定方法によるラッパーの検証

3　サイズとリンク数による件数推定法

検索サイトにさまざまなクエリで検索をかけると、そのクエリに応じた検索結果が HTML ファイルとして返される。これらの検索結果のページには、検索結果のリストの他に広告等の情報が表示される。その中の、検索結果の 1 件 1 件は、タイトル、要約、結果先へのリンク等で構成されている。同じサイトであれば、検索結果のリストや 1 つ 1 つの結果のフォーマットはほとんど同じ形をしている。

このように、検索結果 1 件分のフォーマットが一定であるとすると、検索結果が 1 件増加するごとに、ある一定の量でファイルサイズやリンク数が増加すると考えられる。従って、その 1 件分の増加量が判明すれば、検索結果ページの結果件数が推定できる。

つまり、検索結果 1 件分の増加量を a、検索結果が 0 件のページのファイルサイズ (リンク数) を b とすると、検索結果が x 件であるページのファイルサイズ (リ

ンク数) は、$y = ax + b$ で表される。a,b が求まれば、ファイルサイズ (リンク数)y から結果件数 x を求めることが出来る。

ファイルサイズ方式

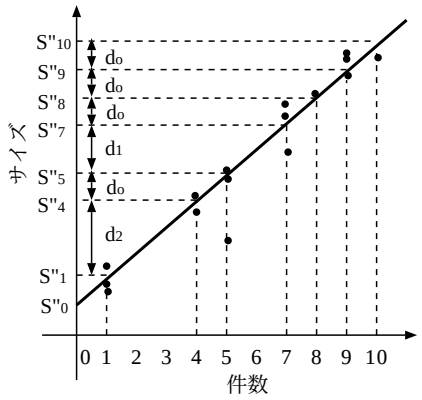


図 3: 検索結果の 1 件分のサイズの推定

まず、n 個のクエリを用いて検索を行い、n 個の検索結果のファイル F_i $(i = 1, ..., n)$ が得られているとする。各ファイルのサイズを s_i とし、これらを昇順にソートしたものを s'_i とする。ここで検索結果に含まれる誤差を考え、差分 $s'_{i+1} - s'_i$ がある域値 e より小さいものは同じ件数のファイルであるとみなし、それらのサイズの平均の値をとり昇順にソートし、順に s''_l $(l = 1, ..., m\ (1 \leq m \leq n))$ とする。ファイルサイズの差分 $s''_{l+1} - s''_l$ を昇順にソートし、d_l とする。d_l は、結果件数の差分を表し、d_l の最小値 d_0 が結果 1 件分のデータサイズである可能性がある [**図 3**]。しかし、件数の差は必ずしも 1 件ではない。これを判断するために、任意の d_l が d_0 で割り切れるかを調べる。割り切れない場合は、d_0 が k 件分の差分であると考え、d_0/k $(k = 2, 3, ...)$ として割り切れる k の値を探索する。データの種類 n を十分大きく取れば、F_i に結果件数 0 件のものが存在し、ファイルサイズの差分の最小値は 1 件分となると仮定することが出来る。このとき、$a = d_0,\ b = s''_0$ と推定できる。この式を基にファイルサイズ s_i 全ての件数を推定する。

リンク数方式

ファイルごとのリンクの数を数え、その差分をとることによってファイルサイズの時と同様な方法で結果件数を推定することが出来る。ファイルサイズ方式と異なる点は、誤差は考えず平均の値をとらないことである。

No.	URL
1	http://www.hyperdyne.co.jp/cgi-bin/namazu.cgi
2	http://www.denso.co.jp/
3	http://www.buzan.or.jp/cgi-bin/namazu.cgi
4	http://clug.linux.or.jp/ml-archive/
5	http://doc.medic.mie-u.ac.jp/mail/graduate/
6	http://info.nttls.co.jp/
7	http://www.java-conf.gr.jp/archives/
8	http://oobof.inarcadia.co.jp/ml/summary/search/
9	http://www.ehime-u.ac.jp/
10	http://www.boj.or.jp/search/search.htm

表 1: 検索サイトの URL

4 実験と問題点分析

実験

実験の対象は国内 56 件の検索サイトとし、サイトごとに 165 個のクエリを用いて検索を行い、得られたページを実験に用いた。例として、10 件の検索サイトの URL を**表 1** に示す。

表 2 では、表 1 で示した 10 件の検索サイトについて、165 個のファイルの中で実件数と推定件数の値が一致しているファイルの割合を示す。例えば、サイト 2 では、ファイルサイズ方式では 99%、リンク数方式では 73%推定が成功している。

表 3 は、方式ごとの、表 2 の推定成功の割合についての 56 件の検索サイトの分布である。例えばサイズ方式では、56 件中 5 件のサイトで 100%推定が成功し、17 件のサイトで 80〜99%の成功率であった。

2 章のファイルサイズ方式、リンク数方式の実験結果を分析して得られた問題点やその原因を以下に挙げる。**図 4**、**図 5** は、サイト 1 におけるファイルサイズ方式、リンク数方式の実験結果のグラフである。

ファイルサイズ方式の問題点

1. 同じ件数でも大きくファイルサイズが異なる場合があるため、その件数の平均値をとるための誤差の設定が困難である
   <原因>
   - 検索結果 1 件に含まれる要約やタイトルなどの長さに差がある
   - 検索結果が 1 ページに表示される最大件数を越えていた場合次のページへのリンクが張られるため、検索結果が増えるほど、表示されている件数は同じであってもファイルサイズ

は増加する

2. 0 件の時のファイルサイズが、165 個のファイル中の最小サイズではない場合がある
<原因>
   - 検索結果が 0 件であった場合に、検索方法の詳細な説明が表示されることがある

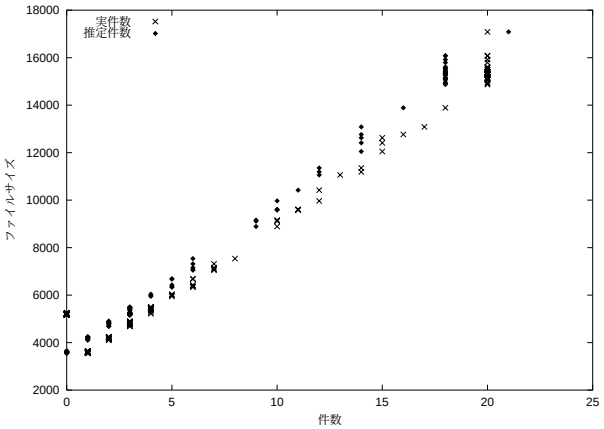


図 4: ファイルサイズ方式での推定件数と実件数のグラフ

図 4 のグラフから分かるように、これらの問題点により、実件数と推定件数の値ががほとんど一致せず、この方法では推定がうまくいっていない。また、実件数のサイズの揺れ幅が大きいことがグラフより確認できる。表 3 で示すように、推定が成功したのは 5 件のサイトのみで、9%の成功率である。

リンク数方式の問題点

1. 結果件数が 1 ページに表示される最大件数になった時リンク数が増加し、その増加量が検索結果 1 件分とは異なるため、1 件分のリンク数の増加量が正しく推定できない
<原因>
   - ファイルサイズの場合と同様に、次ページへのリンクが張られるため、リンク数が増加する

2. 0 件と 1 件のリンク数の差分が 1 件以降のリンク数の増加量とは異なる
<原因>
   - 0 件のページのフォーマットと複数の結果件数をもつページのフォーマットが異なる

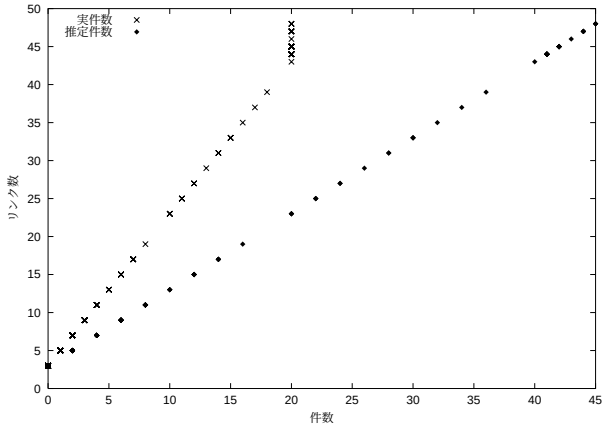


図 5: リンク数方式での推定件数と実件数のグラフ

リンク数方式においても、ファイルサイズ方式と同様に推定はうまくいかなかった。表 3 で示すように、推定に成功したサイトは無かった。

5 推定方法の改良

3 章で挙げた問題点を解決するため、ファイルサイズ、リンク数方式共に改良を加えた。

ファイルサイズ方式の改良

最初に、問題点 1 の誤差の設定についてであるが、誤差の値の取り方を改善することよりも、同じ件数のときのサイズの揺れを無くすことを試みた。そのため、誤差の原因と思われる記述部分をファイルから除去してからファイルサイズを測定して推定を行う。

サイズの揺れの原因となっていると思われる部分は、それぞれの検索結果での個有の情報の部分、タイトルや要約、結果先を示す URL 等である。よって、これらを除去して検索結果の構造を示しているフォーマットの部分のみを残すことができれば、サイズの揺れを押さえることができる。

そこで、次の処理を推定前の検索結果のファイルに行う。

Step1 テキスト部分を除去し、タグのみを残す。

Step2 アンカータグ内の URL を除去する。

また、問題点 2 の解決として、件数の増加を伴わないリンク数の増加によるサイズの揺れを無くすために、アンカータグを除去する。検索結果に含まれるリンクも除去することになるが、検索結果にはリンクのみでなく構造を指定するためのタグがあるため、1 件の増加分のファイルサイズには大きな影響は与えないと考

える。

Step3　アンカータグで囲まれている部分を全て除去する。

Step4　タグを属性を除いてタグ名のみにする。
例：<table border=0 cellpadding=2 cellspacing=0>
→ <table>

Step5　強調タグと
タグを除去する。

Step1 から Step5 までの操作を行ってから、推定を行う。推定方法で 2 章のファイルサイズ方式と異なる点は、0 件から 1 件への増加は例外的で必ずしも一定量の増加になるとは限らないので、サイズを昇順にソートした後、最小のサイズを 0 件と決定し、0 件のサイズを除いた他のファイルサイズに対して推定を行う点である。

表 3 で示すように、2 章の改良前のファイルサイズ方式より推定精度が向上し、40 件のサイトで推定が成功した。**図 6**(a) はサイト 1、(b) はサイト 3 についてのグラフであるが、まだいくつかの問題点がある。

リンク数方式の改良

リンク数の問題点 1 を解決するためには、検索結果ページに含まれるリンクのうち、最大件数を越えたときに現れる次ページへのリンクを表すアンカータグのみを除去しなければならない。

検索結果ページを調べると、次ページへのリンクを表す URL は”?” を含むことが多いことが分かった。このことを利用して、ファイルに含まれるリンク数を数えるときには URL に”?” を含む場合は除くように改良を行った。

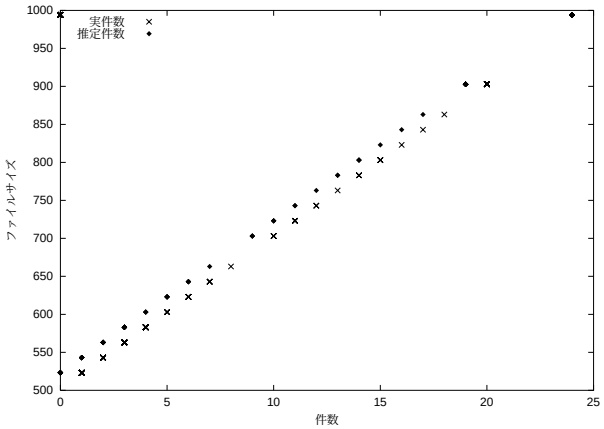
図 7(a) はサイト 10、(b) はサイト 1 についてのグラフである。

表 3 で示すように、改良により精度が向上し、53 件のサイトが推定に成功している。95%の成功率である。

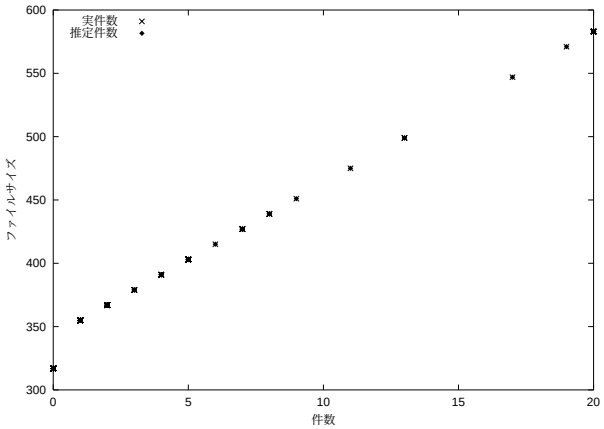
ファイルサイズおよびリンク数を合わせた合同方式

上述の 2 つの方式の改良により推定精度は向上したが、新たに出現した問題点や解決できなかった問題点のため正しく推定出来ない場合があった。

- ファイルサイズ方式で 0 件が最小サイズでない
- ファイルサイズ方式で 0 件の次が 1 件ではない
- ファイルサイズ方式で 0 件と 1 件の差が非常に小さく、平均を取るときに同じサイズとみなされてしまう



(a) 推定出来ていないグラフ



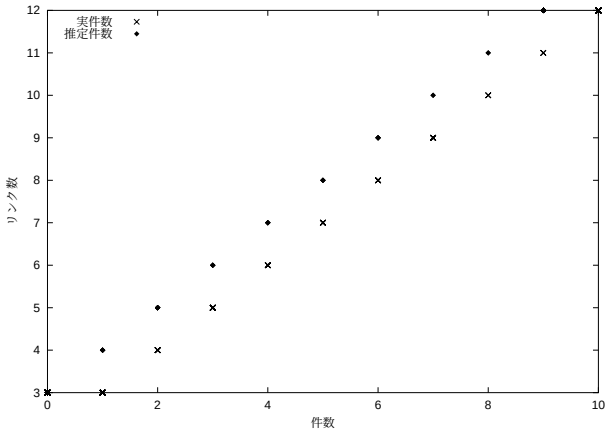
(b) 推定が成功したグラフ

図 6: 改良後のファイルサイズ方式での推定件数と実件数のグラフ

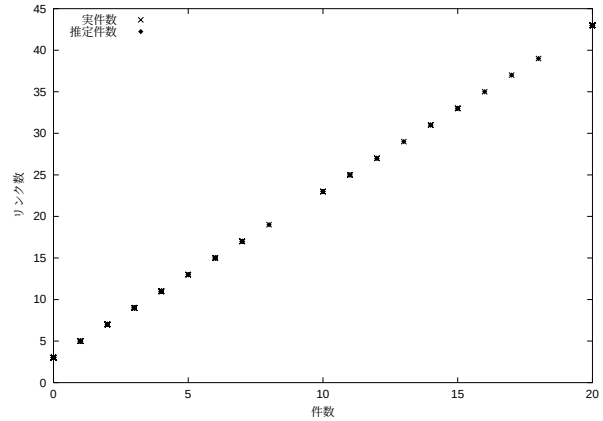
- リンク数方式で 0 件と 1 件が同じリンク数になる
- 0 件の検索結果がない

これらを改善するために、2 つの方式を利用して組み合わせて推定を行うように改良する。

0 件の判定がファイルサイズ方式でもリンク数方式でも難しいが、合同方式では、リンク数が最小のもののうちサイズが一番小さいものを 0 件とする。また、リンク数方式ではアンカータグのうち次ページへのリンクを示すものとして”?” を含むタグを除いたが、検索結果のリンクに”?” が含まれる可能性もあるため、より一般性を持たせるために、合同方式ではリンク数を数えるときにはページに含まれる全てのリンクを数えることにする。また、検索結果 1 件分の増加量の推定は、サイズとリンク数の両方を用いて求める。



(a) 推定出来ていないグラフ



(b) 推定が成功しているグラフ

図 7: 改良後のリンク数方式の推定件数と実件数のグラフ

表 3 で示すように、53 件のサイトが件数推定に成功した。95%の成功率である。

6 精度評価

3 章で示したデータに対して、全ての実験を行った。

表 3 により、改良後のリンク数方式と合同方式が最もよい推定精度であることが分かる。だが、リンク数方式の方は、「次ページへのリンクには”?” が含まれる」ことを前提とした方法のため、検索結果を表すリンクに “?” が含まれるという対処できない場合が起こることも考え得るので、より一般性の高い合同方式の方が件数推定に有効であると思われる。

また、図 6(a) は、実件数と推定件数の一致という点では推定は失敗しているが、グラフの実件数と推定件数の傾きは一致している。この方式では、実験を行った 56 件の検索サイト全てで実件数と推定件数のグラフの傾きが同じであった。本論文での検索結果件数推定の目的はラッパーの検証なので、完全な件数推定でなくてもこのように実件数のグラフの傾きが得られれば、ラッパーのレコード抽出数のグラフとの比較をラッパー検証のひとつの指針とすることができる。よって、グラフの傾きの一致のみでも有効な検証手段となると思われる。

No.	サイズ方式		リンク数方式		合同
	改良前	改良後	改良前	改良後	方式
1	0.00	0.00	32.12	100.00	100.00
2	99.39	100.00	73.94	100.00	100.00
3	85.45	100.00	47.88	100.00	100.00
4	70.91	100.00	33.33	100.00	100.00
5	100.00	100.00	66.67	100.00	100.00
6	61.82	100.00	33.94	100.00	100.00
7	63.64	100.00	27.88	100.00	100.00
8	72.12	100.00	29.70	100.00	100.00
9	36.97	100.00	7.27	100.00	100.00
10	25.45	100.00	19.39	19.39	100.00

表 2: 検索サイトごとの実件数と推定件数の一致の割合

割合	サイズ方式		リンク数方式		合同
(%)	改良前	改良後	改良前	改良後	方式
0〜19	14	13	12	3	1
20〜39	6	2	18	0	2
40〜59	2	1	14	0	0
60〜79	12	0	7	0	0
80〜99	17	0	5	0	0
100	5	40	0	53	53

表 3: 評価のまとめ

7 まとめと今後の課題

本論文では、検索サイトラッパーの検証に用いるための検索結果推定方法について、いくつかの方法を提案した。56 サイトについて実験を行い、改良後のリンク数方式と、ファイルサイズとリンク数方式を組み合わせた合同方式が最も推定精度がよい結果が得られ、合同方式の方が件数推定に適した方式であることを示した。

また、完全一致の件数推定でなくても、推定件数のグラフの傾きが実件数のグラフの傾きと一致することを利用して、ラッパー検証を行うことも出来る。

このように、本論文で提案した検索結果推定方法はラッパー検証に使用可能な精度を持つことを確認し、ラッパーの検証に有効であることが示された。

今回は国内のサイトを実験対象とし、高い精度の結果を得たが、英語の検索サイトにおいては、検索結果ごとの揺れ幅が大きく、これらの方式でもあまり推定の精度があがらなかった。よって、今後の課題として、英語サイトにも適用できるような方式を検討する必要がある。

また、提案した方式は、1 つのサイトにつき推定対象となる複数の検索結果のページが、多岐の件数にわたることを前提としているので、そのような検索結果のページを得るためのクエリの選択も非常に重要な課題である。

参考文献

[1] BrightPlanet, The Deep Web:Surfacing Hidden Value, BrightPlanet White Paper, 2000 http://maya.cs.depaul.edu/ classes/cs589/papers /deepweb.pdf

[2] Chris Sherman, Gary Price, The Invisible Web, Infomation Today, Inc. 2001.

[3] CPAN,http://www.cpan.org/

[4] S. Hirokawa, S. Watanabe, Y. Koga, T. Taguchi, Automatic Feature extraction of Search sites, Proc.SSGRR2001

[5] P. Ipeirotis, L. Gravano, M. Sahami, PERSIVAL Demo: Categorizing Hidden-Web Resources, JCDL2001,2001

[6] P. Ipeirotis, L. Gravano, M. Sahami, Probe, Count, and Classify:Categorizing Hidden-Web Databases, ACM SIGMOD 2001,2001

[7] 古賀 康則, 田口 剛史, 廣川 佐千男：検索サイト統合のためのラッパー生成法, DEWS2001 CD-ROM:6b-1, 2001.

[8] Paul Pedley, The invisible web, ASLIB, 2001.

[9] E. Selberg, O. Etzioni, The MetaCrawler Architechture for Resource Aggregation on the Web, IEEE Expert, 11–14, 1997.

[10] T. Tagchi, Y. Koga, S. Hirokawa, Integration of Search Sites of the World Wide Web, Proc.CUM, vol.2, pp.25-32,2000.