C2-4 研究会プログラムのWebページからのデータ抽出

渡辺精一郎 * 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

現在 Web 上には、有益な情報を提供している文書が多数存在する。こうした文書の中には、空白や改行等で整形された文書や、XML、HTML のようにタグを用いて文書の構造を表すものが多数あり、半構造化データと呼ばれている。半構造化データの中には、その文書の持つ情報が、表のように複数のフィールドから成るレコードの繰り返しとみなせるものが存在する。Web 文書からレコードの抽出ができると、各文書からそれぞれ抽出したレコードを統合して、Web 上の情報をデータベースのように利用できるようになる。

本研究では具体的な Web の文書群について、レコードの表記形式を分析し、レコードの形式を含めたページのフォーマットの表記法を考案した。そして、ページのソースとそのページのフォーマット情報を入力として、レコードの抽出を行なうプログラムを実装した。このレコード抽出プログラムを、情報処理学会の各種研究会のプログラムを記載したページ群について適用し、レコード抽出の性能の評価を行なった。また、抽出したレコード群について統合検索を行なうシステムを設計、実装した。

1 はじめに

現在急速な勢いで普及、拡大し続けている Web 上には、様々な要求に対応した、有益な情報を提供している文書 (ページ) が多数存在する。このような Web 上の文書群から、必要な情報を抽出して活用しようとする研究が行なわれている [4]。これらの文書の中には、空白や改行、記号等で整形された文書や、XML 文書、HTML 文書のようにタグを用いて構造を表している文書がある。これらは、単なる文字列の文書と、正確に表現されたデータ構造の中間という意味で半構造化データ [3] と呼ばれる。半構造化データの中には、そのページの持つ情報が表や名簿、リストのように複数の項目から成るレコードの繰り返しで表されているものが存在する。

これらのレコードは作成者が異なることや、表現したい事象が異なることにより、それぞれの文書によってその表現方法、形式が異なる。従って、データベースのように必要とする項目に対するデータだけを取り出すことや、違う形式のページ群から共通のフィールドを取り出し統合することは容易でない。Web 文書からレコードを抽出するプログラムは一般的にラッパー [6] と呼ばれる。レコード抽出の対象とする文書に対してラッパーを用いることで、レコードの抽出が可能となる。このラッパーを用いて、それぞれの文書からレコードを抽出し、その結果を統合することで、Web 上の情報をデータベースのように扱うことが可能になる。

本稿では半構造化データの具体例として、学会の研究会プログラムが記載された Web ページを対象とし、ページ中に現れるレコードの表現形式を分析した。その分析に基づいて定めた、レコードの形式を含めたページのフォーマット表記法について述べる。また、考案した表記法で記述されたページのフォーマット情報と、Web ページのソースを入力として、レコードの抽出を行なうラッパーについて述べ、このラッパーの実験による評価を述べる。さらに、このラッパーを用いて抽出したレコードについて統合検索を行なうシステムについて述べる。

2 レコードのフォーマットとラッパー

本論文では、半構造化データの中でも特に表や名簿などのように複数のフィールドから成るレコードの繰り返しで表現されているページを対象とする。このようなページはそれ 1 つでも意味的にはデータベースとみなせる。また、このようなページを多数集め、統合し

て検索できれば、Web 上のデータベースとして利用できる可能性がある。そのレコードの抽出に必要なページのフォーマットの表記法を提案する。

例えば、図 1 は情報処理学会の第 1 回プログラミング研究会のプログラムを記載した Web ページである。

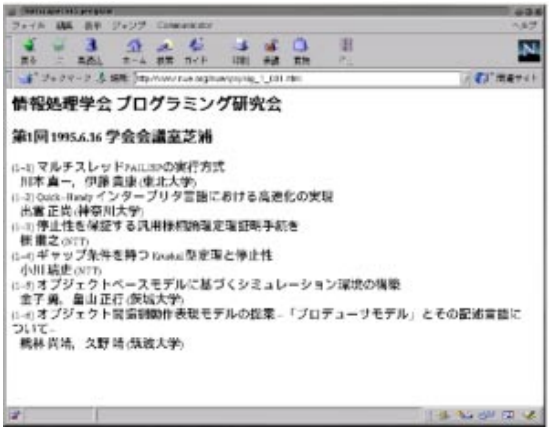

図 1: 第 1 回 PRO 研究会のプログラムのページ

このページでは、発表のタイトルと発表者、所属の 3 つのフィールドを持つレコードが繰り返し現れている。このようにレコードが繰り返し現れているページは現在の Web 上に多数存在している。また、このようなレコードが複数出現するページでは、同じフォーマットを用いてレコードの 1 つ 1 つを記述しているページが多い。そこで、このレコードの形式が予め分かっていれば、それを用いてレコードの抽出を行なうことが可能である。

このように繰り返し現れるレコードの他に、各レコードに共通の情報も重要である。この例では、“情報処理学会” や” プログラミング研究会”、“第 1 回” などの情報はこのページには 1 回しか現れないが、全てのレコードに共通の情報である。このような情報を本稿では COMMONDATA と呼ぶ。このような情報も抽出の対象とし、ページの中でどのように出現しているかを表す方法を提案する。

2.1 COMMONDATA の情報の表記法

COMMONDATA を記述する方法としてそれがページ中で出現している位置情報を用いる。

具体的には COMMONDATA の位置情報を次の 4 つで定める。

・HEAD
・TAIL
・H-NUM
・T-NUM

HEAD とは対象の文書において COMMONDATA が出現しているすぐ直前の文字列のことを指す。Web 上の文書は現在 HTML や XML を用いて記述されているものが多く、タグが HEAD となる。もし文書の先頭に COMMONDATA があり、その前に文字がない場合は HEAD は SOF(Start Of File) とする。また、HEAD は文字列の長さにはこだわらないことにする。HEAD は、その文書中で唯一出現しているものとして特定できれば理想的である。文字列の長さにはこだわらないことから、直前の文字列を長く取ることによってほとんどの場合この条件を満たすことができる。また、あまりに長い文字列を情報として持っておく事は無駄が多い。このように文字列 HEAD を指定しても、対象の文書中にこの HEAD が複数回現れる事がある。そこで HEAD の何番目の出現かを指定するために H-NUM を用いる。HEAD が SOF のときは H-NUM は 1 とする。この 2 つの情報によって COMMONDATA の出現位置を規定できる。

TAIL は COMMONDATA のすぐ直後の文字列のことを指す。もし、文書の最後に COMMONDATA があった場合は TAIL は EOF(End Of File) とする。TAIL の考え方は HEAD と同様である。違う点は文書の対象範囲である。HEAD は文書全体に対して何処にデータが存在するかを示す位置情報であったが、TAIL は HEAD および H-NUM が決まっている時点で HEAD 以前の文書について考慮する必要がない。よって、TAIL は HEAD 以降の文書中で 1 つに限定できる情報であれば良い。T-NUM は H-NUM と違い、TAIL である文字列が文書中の何番目に現れるかではなく、HEAD 以降の文書の中で何番目に現れているかを示すものとする。これらのことから、T-NUM は TAIL と同じ文字列が COMMONDATA の中に現れない限り 1 である。TAIL が EOF のときは T-NUM は 1 とする。これら 4 つの情報によって COMMONDATA を一意に定めることができる。

2.2 レコード情報の表記法

繰り返し現れるレコードの形式を以下の情報で表現する。

・R-HEAD
・R-TAIL
・RH-NUM
・RT-NUM
・DELIMITER

・ITEM

この情報の内、R-HEAD、R-TAIL、RT-NUM、RH-NUM はレコードが出現している領域を示すものである。レコードが出現している領域を指定することは大半のページでは必要ない。しかし、ページによってはレコードの形式と同じ形式で別の情報を表示しているものもある。このような場合、レコードが現れている領域を指定しておくことでノイズを抽出せずに済む。

R-HEAD はレコードが現れている領域の前の文字列を指定する。ただし、厳密にレコードが現れている直前の文字列である必要はなく、あくまでレコードが出現している領域が限定できれば良い。この R-HEAD の考え方は COMMONDATA を抽出する際の HEAD と同様である。よって、レコードが現れている領域が文書の先頭からであれば SOF とすることも HEAD と同じである。RH-NUM も H-NUM と同様 R-HEAD がページの文書中の何番目に出現するかを表記することにする。R-HEAD が SOF であれば RH-NUM は 1 とするのも同じである。

R-TAIL はレコードが現れている領域の直後の文字列になる。これも厳密にレコードが現れている直後の文字列である必要はない。基本的な考え方は TAIL と同じである。もし、レコードが現れている領域が文書の末尾にまで及んでいれば EOF とする。

RT-NUM も同様に R-HEAD 以降の文書の中で R-TAIL が何番目に現れているかを示すものとし、R-TAIL が EOF であればこれを 1 とする。

ITEM は各フィールドの項目名であり、各フィールドが何の種類についての情報かを記述するためのものである。各フィールドの出現順に ITEM(1)、ITEM(2)、ITEM(3)、‥‥とする。DELIMITER は 1 つのレコード中の各フィールドの間を区切っている文字情報である。DELIMITER も ITEM 同様、順に DELIMITER(1)、DELIMITER(2)、DELIMITER(3)、‥‥ とする。

2.3 フォーマット情報を入力とするラッパー

Web ページからレコードを抽出するラッパーについて述べる。先程のページのフォーマット情報は予め特定のファイルに記述しておく。ラッパーは、レコード抽出の対象ページと、そのページのフォーマット情報を記述したファイルを入力とし、対象ページのレコードと COMMONDATA の抽出を行なう。同じ形式で書かれたページ群については、同一のフォーマット情報が利用できる。出力は抽出した各レコードと COMMONDATA である。

3 実験

対象を情報処理学会に属する幾つかの研究会の Web ページとし (表 1)、実際にページのフォーマットが前述の方法で記述できるかどうか、またこのフォーマット情報を用いてレコードの抽出が可能かどうか調べた。このページより抽出したい情報は、日時、開催場所、タイトル、発表者、など計 13 の項目に対する情報である。また、この実験対象のページは全て、レコード中に必ずタイトルと発表者の 2 つのフィールドを持つ。そこで、この実験では最低でもこの 2 つのフィールドについての情報の抽出をして、さらに他の情報がある場合はその情報についても抽出した。この実験結果を表 2 に示す。

表 1: 実験対象の研究会ページ

研究会名	URL
DBS	http://www.mdbl.sfc.keio.ac.jp/IPSJ-DBS/
ARC	http://phase.etl.go.jp/sigarc/
SLDM	http://www.elc.ees.saitama-u.ac.jp/SLDM/
PRO	http://www.ipsj.or.jp/sig/pro/
AL	http://www-imai.is.s.u-tokyo.ac.jp/sigal/
MPS	http://pdap1.trc.rwcp.or.jp/sigmps/
HPC	http://phase.etl.go.jp/sighpc/
OS	http://www.ht.sfc.keio.ac.jp/SIGOS/

表 2: 各研究会のページに対するレコード抽出の結果

研究会名	ページ	成功 (完全)	失敗	定義不可
DBS	11	4 (4)	1	6
ARC	9	1 (0)	0	8
SLDM	9	7 (7)	0	2
PRO	32	32 (32)	0	0
AL	16	10 (0)	0	6
MPS	33	19 (19)	14	0
HPC	17	16 (11)	0	1
OS	6	0 (0)	0	6
計	**133**	**89 (73)**	**15**	**29**

表 2 において「成功」は、レコードの抽出ができたページの数である。この内、抽出した全てのレコード中のデータにノイズがないものは「完全」に成功としている。「失敗」という項目の数字は、ページのフォーマットは定めることができたが、ラッパーで抽出した中に、得たい情報と別の情報が混じってしまったものである。「定義不可能」の項目の数字は、ページのフォーマットを定めることが不可能だったページの数である。

ページのフォーマットを定めることが「定義不可能」だったページはそのほとんどが、ソースに”<pre>” タ

グを用いていたものである。この”<pre>” は、そのタグ以降に書かれた文字列をそのままでブラウザ上に見せるものである。よって、”<pre>” タグを用いたページはレコードの形式が曖昧で、その為ページのフォーマットを決めることができなかったものが多かった。このようなページを除けば、抽出成功率は 80%を越えており、中でも、DELIMITER がタグを中心として構成されているページに関しては、レコードの抽出成功率は 90%を越えていた。

4 統合検索システムの設計と実装

実装した統合検索システムについて述べる。Web をデータベースとして扱えるように、様々な種類の項目についてキーワードを入力することが可能な統合検索システムの構築を目標とした。本稿ではその前段階として、情報処理学会のプログラムのページに関しての検索を行なう CGI プログラムを実装した。

このプログラムに必要な情報はページのソースとページのフォーマット情報を記述したファイルである。この 2 つを予め保有しておく。プログラムはこれらのページに対し、前述のラッパーを用いてページ中の全てのレコードの抽出を行なう。そして、抽出した各レコードに、その対象ページの COMMONDATA を加える。こうして抽出されたレコードの中から、検索の結果として該当するものを選び出して出力する。

図 2 は実装した CGI のインターフェースである。検索は各項目に書かれたキーワードに対して and 検索を行なう。また、結果の画面は図 3 である。この図にあるように、各ページごとにテーブルを 1 つ作り、検索結果に該当するレコードを出力する。こうして得られる検索結果は、必要なデータ部分だけが抽出され、整形した形で表示される。これはキーワードを含むページを表示する一般の検索エンジンではできない。

5 関連研究

Web 上の情報を収集して活用する研究として、Craven らの Web → KB プロジェクト [1] がある。これは Web ページから知識ベースを自動的に構築しようとするもので、自動化の方法として、システムにインスタンスを与えて機械学習を行なっている。知識ベースによる推論等を行なうことができる為、より高度な検索等が期待できるが、一方で、インスタンスの生成に多大なコストがかかることや、抽出精度が問題となっている。

本研究も現状では各ページに対しレコードのフォー

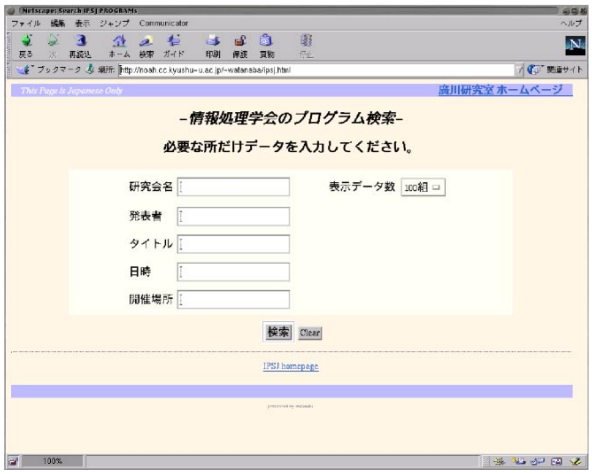

図 2: CGI プログラムのインターフェース

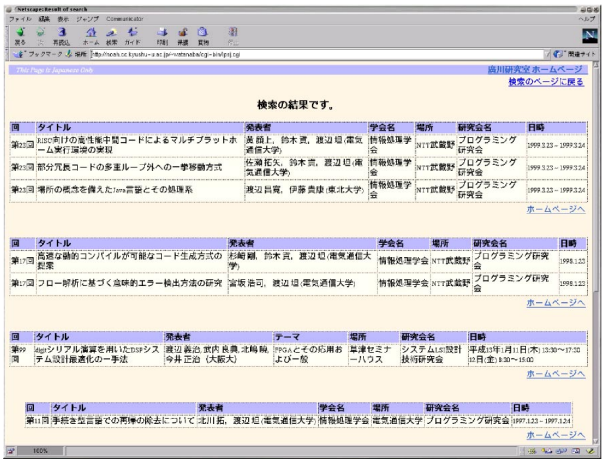

図 3: 検索結果の画面の一部

マット情報を記述するためコストがかかる。しかし、その解決策としてレコード情報の自動的発見が考えられる。既にそのような研究は盛んに行なわれており、その一つとして Kushmerick ら [2]LR ラッパーがある。この LR ラッパーは、HTML 文書のタグを対象として、レコードの区切りとなっているものを発見する。また、我々の研究室でのラッパーに関する研究 [6] においても、HTML 文書のタグの繰り返しパターンに着目してレコード抽出を行なっている。本研究の実験対象のページのうち、レコードのフォーマット情報がタグで表されているものについては、既にこのラッパーを用いて自動的にフォーマット情報抽出ができることを確認している。しかし、これらのラッパーではレコードの区切りが HTML のタグである。本研究で対象としたデータでは、区切りがその他の文字列である場合があり、タグを対象とするこれらの手法が単純に適応

できるわけではない。

レコードの区切りの自動的発見の他に、レコード中の各フィールドの内容を特定する必要がある。Web 上の特定の情報を収集、統合する研究として、杉田ら [7] がある。[7] では、書誌情報、特に書評を Web 上から収集し、自動編集を施した上で利用者に提案する。Web ページからの書誌の情報の抽出には、著書名、出版社名などそれぞれに応じたデータベースを予め持っていてそれを利用している。また、梅原ら [5] では、HTML 文書から XML 文書への半自動変換を行なっている。[5] では、レコードの内容の特定をする為に、HTML 文書のタグの木構造を利用しており、また、レコード中の文字列の類似度の比較も行なっている。これらの研究にあるように、抽出したレコード部分の内容を検証することが本研究の課題の一つに挙げられる。

6 まとめ

本稿では、半構造化データの具体例として、研究会のプログラムを記載した Web ページ群を対象とし、半構造化データ中のレコードを抽出、利用する為の方法について述べた。

まず、レコードを抽出する為に必要となるページのフォーマット表記法を提案した。そして、この表記法で記述されたページのフォーマット情報と、ページのソースを入力として、レコードを抽出するラッパーを実装し、その評価を行なった。また、作成したラッパーを利用して、情報処理学会のいくつかの研究会のプログラムを記載した Web ページに対し、レコード情報の統合検索が可能な CGI プログラムを作成した。

提案したページのフォーマット表記法および実装したラッパーは単純なものであるが、その分ある程度の汎用性があることを実験で示した。また、実装した CGI はまだ初期段階のものであったが、十分使えるものであった。

参考文献

[1] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, S. Slattery, “Learning to Construct Knowledge Bases from the World Wide Web”, Artificial Intelligence vol118, p69-113 (2000)

[2] N. Kushmerick, D. Weld, R. Doorenbos, “Wrapper induction for Information Extraction”, IJCAI’97, p729-737 (1997)

[3] S. Abiteboul, P. Buneman, D. Suciu, “Data on the Web”, Morgan Kaufmann Publishers (2000)

[4] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, J. Widom, “The TSIMMIS Project:Integration of Heterogeneous Information Sources”, In Proceedings of IPSJ Conference, p7-18, Tokyo Japan (1994)

[5] 梅原雅之, 岩沼宏治, 永井宏和, “事例に基づく HTML 文書から XML 文書への半自動変換”, 人工知能学会論文誌 16 巻 5 号 B, p408-416 (2001)

[6] 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 (2001)

[7] 杉田茂樹, 江口浩二, “目録データベースと Web コンテンツの統合的利用方式”, 情報処理学会研究報告 vol2001 No20, p153-158 (2001)