修 士 論 文

# ワールドワイドウェブからの住所録の自動生成

指導教官　佐藤理史 助教授

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

810072 津田朋樹

2000 年 2 月 15 日

## 要 旨

ワールドワイドウェブ上に存在する住所情報をカテゴリごとに自動収集し、その情報をデータベースとしてまとめあげることにより、そこからユーザの要求する形式の住所録を自動生成し、提供するシステムの作成を行なった。

本システムは (1) 住所情報収集モジュール (2) 住所情報データベース (3) 検索モジュールの 3 つの要素から構成される。住所情報収集モジュールは、住所一覧ページを情報源とすることにより、カテゴリの情報が含まれた住所情報の収集を行なう。さらに、本モジュールは、ページ内の情報を元に欠落情報の補完も行なう。検索モジュールは、データベース内の情報からユーザの指定通りの住所録を自動生成し、ユーザに提供する。

本システムはウェブ上から様々なカテゴリの住所情報を収集できるように汎用性を持たせて作成されている。任意に選択した 30 カテゴリに関してウェブ上で住所情報の収集を行なったところ、約 32,000 件の住所情報を収集することができた。その住所情報の適合率は 79%であった。また、システムのカテゴリ判定率は 90%であった。

# 目 次

# 第 1 章

# 序論

## 1.1　本研究の背景と目的

これまで特定施設や店舗などの住所や電話番号といった情報（住所情報）を知るためには、人に尋ねるか、自分でその施設の住所情報が記載されているパンフレットや広告を手に入れるか、もしくは電話帳などで探すしかなかった。また、タウンページなどで図書館や病院などの種類（カテゴリ）から、そのカテゴリに属する複数の施設や店舗の住所情報を収集することができる程度であった。しかし、近年のワールドワイドウェブ（以下、ウェブと略記）の発展に伴い、現在ではウェブ上から多くの住所情報を入手することが可能になった。誰もが簡単に情報を発信、受信できるウェブ上には、今後ますます様々な情報が増え続けていき、住所情報を取得する上でも非常に有用なメディアになっていくことが容易に推測される。

ウェブ上の住所情報が掲載されているページは、大別すると以下の 2 種類に分類することができる。

1. 施設、店舗などの紹介ページ
   特定の施設、店舗などを紹介する目的で作成されたページ。

2. 住所一覧ページ
   図書館、病院などのカテゴリによって分類された住所一覧表（住所録）が掲載してあるページ。

ウェブ上から住所情報を取得する上で、取得したい情報の種類により探さなければいけないページの種類は異なってくる。既に、自分が探している対象の名称やカテゴリが判明していて、その対象の住所が知りたい場合は、1 のようなその対象の紹介ページを探せば

よい。例えば、北陸先端科学技術大学院大学の住所が知りたい場合は、同大学の公式ページを探せばよい。しかし、この場合、公式ページ内のどこに住所情報が掲載されているかが分かっている訳ではないので、さらにそのページ内から住所情報を探す必要がある。

それとは別に、あるカテゴリに属する対象の名称と住所が知りたい場合は、2 のようなそのカテゴリに関する住所一覧ページを探せばよい。例えば、石川県内のどこに何という大学があるのかを知りたい場合は、石川県内（もしくは、北陸、中部内など）の大学住所録が掲載されているページを探せばよいのである。また、この住所録を見ることにより、北陸先端科学技術大学院大学の住所も知ることができるであろう。もっとも、このような住所一覧ページは全てのカテゴリに関して存在する訳ではないのだが、主要な施設や、利用者が多いものに関しては、殆ど作成されていると言っても過言ではないし、毎日のように増え続けてもいる。しかし、これらのページは、提供者が任意に決めたフォーマットで提供されるため、ユーザ側が必要としている形式の住所一覧表でない場合も多い。

多少の問題はあるものの、これらのページは住所情報を知る上で非常に有用なものであると言える。しかし、これらのページは、ウェブ上に点在しているため、ユーザは知りたい住所情報が掲載されているページを探すのに労力を要する。さらに、ページによって情報の記述形式が一様でないため、ユーザ側としては利用しづらい面も多い。

これらの問題点を解決するために、本研究では、あらかじめウェブ上のそれらのページから住所情報を自動収集し、一つの住所情報データベースとして統合することにより、ユーザが必要とする情報のみの住所録を自動生成して提供するシステムを提案する。これによりユーザは、必要とする住所録や住所情報を、必要なときに必要な表示形式で素早く手に入れることが可能になる。また、情報の収集も機械処理によって行なわれるため、ウェブ上で常に増え続ける大量の情報を、人手による労力を必要とせずに収集することができる。

## 1.2　本システムの情報源

本システムを作成する上で実現しなければいけない課題の 1 つとして、カテゴリ情報を持った住所情報を収集しなければいけないという点が挙げられる。これは、カテゴリ別の住所録を作成する上では必要不可欠な情報であり、ユーザが情報を検索する上でも重要な役割を果たすものだからである。カテゴリ情報を持った住所情報を収集するにあたり、どのような情報源から住所情報を収集するかを考える必要がある。住所一覧ページは、住所一覧表（住所録）の見出しとしてカテゴリ名が用いられている可能性が高く、比較的容易にカテゴリ名を抽出することが可能である。それに対し、施設、店舗などの紹介ページ

から、対象とする住所情報のカテゴリを抽出するのは非常に困難である。ウェブページ内から店舗名とそのカテゴリを抽出する研究 [1] も行なわれているが、これには非常に膨大な外部知識から構築した辞書が必要であり、実用的ではない。

以上の理由より本研究では、住所一覧ページを情報源としてカテゴリ情報を持った住所情報を収集する手法を提案する。そしてこの手法により、今まで人手で行なっていた、情報をカテゴリごとに分類する作業を容易に機械処理することが可能であることを示す。

## 1.3 本論文の構成

本論文ではまず、2 章で作成した住所情報データベース編集システムの概要を説明し、3 章と 4 章で本システムの住所情報収集処理の詳細について述べる。5 章でシステムの評価実験について述べたのち、６章で本研究の結論を述べる。

# 第 2 章

# 住所録自動生成システムの概要

本章では、作成した住所録自動生成システムの概要について述べる。

## 2.1 システムの概要

本研究で作成したシステムの構成を図 2.1に示す。本システムでは、まず、オフラインでカテゴリ別に住所情報を収集し、それをデータベースに格納する。こうして作成されたデータベースから、ユーザーの要求に従って住所録を生成する。本システムは次の 3 つの部分から構成されている。

1. 住所情報収集モジュール
   カテゴリ名を入力として、ウェブ上から住所情報を収集し、データベースに登録する。

2. 住所情報データベース
   住所情報収集モジュールで収集した住所情報を格納するデータベース。

3. 検索モジュール
   ユーザの検索要求に応じて、住所情報データベースを検索し、検索結果を一覧表の形式（住所録）で表示する。

なお、本システムは、プログラム言語 Perl5[2] を用いて実装されており、その規模は、約 2000 行である。

以下では、まず、検索モジュールとその使用例について述べる。次に、住所情報データベースの形式について述べる。最後に、住所情報収集モジュールの概要について述べる。

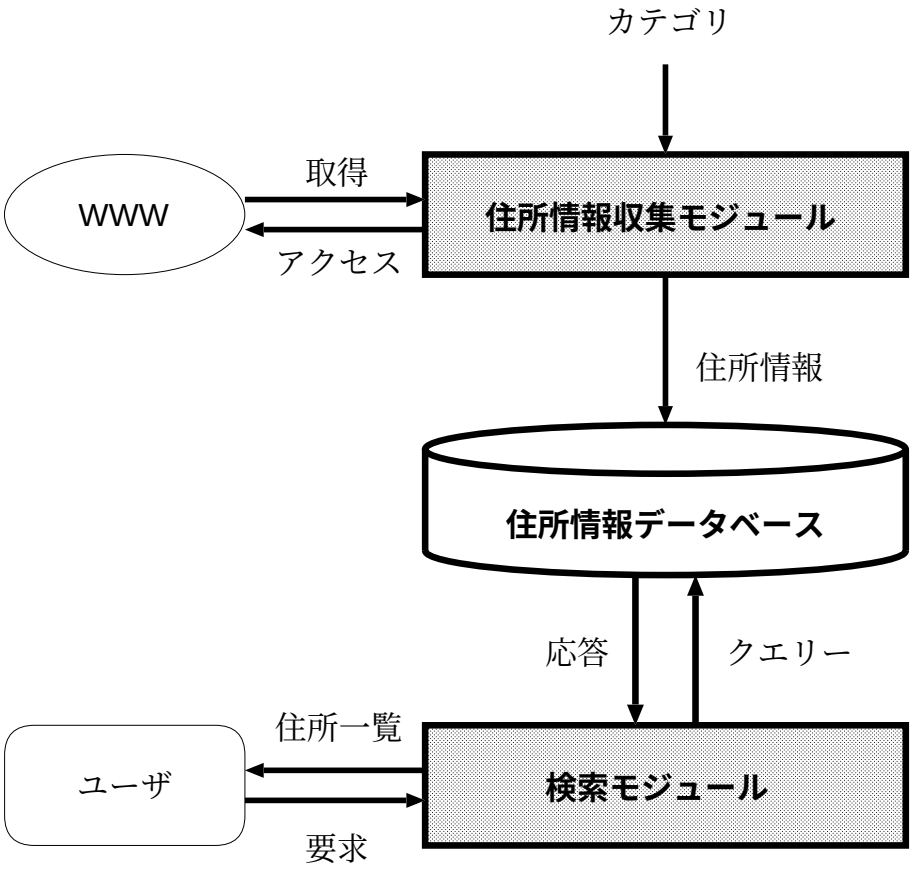

カテゴリ
取得
WWW
住所情報収集モジュール
アクセス
住所情報
住所情報データベース
応答
クエリー
住所一覧
ユーザ
検索モジュール
要求

図 2.1: システムの構成

## 2.2　検索モジュール

検索モジュールは、ユーザの要求に応じて住所情報データベースを検索し、検索結果を一覧表（住所録）として表示する処理を行う。

検索モジュールのインターフェースの画面を図 2.2に示す。この画面において、ユーザは、検索対象を限定するための条件と、検索結果の表示方法を指定することができる。

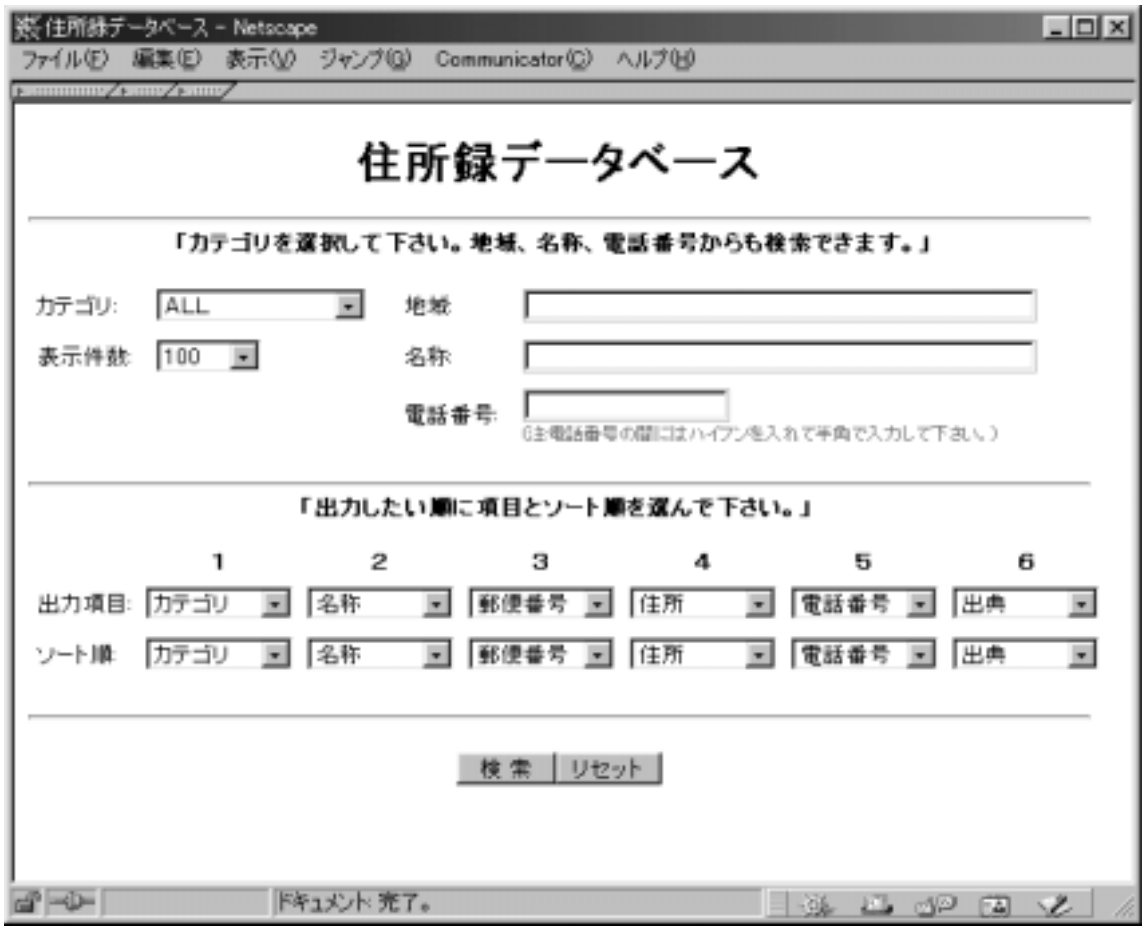



図 2.2: 検索モジュールのインターフェース画面

検索対象を限定するための条件には、以下の 3 つがある。

1. カテゴリ
   プルダウンメニューから選ぶ。カテゴリを指定した場合は、このカテゴリ名に含まれるもののみが検索される。

2. 地域
   文字列で入力する。指定した場合は、この文字列が住所に含まれるもののみが検索される。

3. 名称
   文字列で入力する。指定した場合は、この文字列が名称に含まれるもののみが検索される。

検索結果の表示方法は、以下の 3 項目が指定できる。

1. 表示するフィールド名
   以下に示す 6 フィールドから、必要なフィールド名を表示順に選択する。

   (a) 名称

   (b) カテゴリ

   (c) 郵便番号

   (d) 住所

   (e) 電話番号

   (f) 出典

2. ソート順
   ソートするフィールドの優先順位。

3. レコードの表示数
   1 度に表示するレコード数。

これらを適切に指定することにより、各種の住所録を自動生成することができる。以下に例を示す。

- 特定カテゴリの住所録
  カテゴリを指定することにより、そのカテゴリの住所録を生成することができる。例えば、病院の住所録は、カテゴリで「病院」を選択すればよい（図 2.3）。

- 特定地域の住所録
  地域を指定することにより、その地域に存在する対象の住所録を生成することができる。例えば、地域として「石川県小松市」を指定すると、石川県小松市にある施設の住所録が生成できる（図 2.4）。


図 2.3: カテゴリ「病院」での検索結果


図 2.4: 地域「石川県小松市」での検索結果

- 特定地域における特定カテゴリの住所録
  カテゴリと地域の両方を指定することにより、特定地域における特定カテゴリの住所録を生成することができる。例えば、地域で「石川県」、カテゴリに「書店」を指定することにより、石川県内の書店の住所録を生成することができる（図 2.5）。


図 2.5: 地域「石川県」、カテゴリ「書店」での検索結果

なお、本システムは、住所録を生成するだけでなく、名称を明示的に指定することにより、その名称を持つ施設の住所を調べる場合にも利用可能である。図 2.6に例を示す。

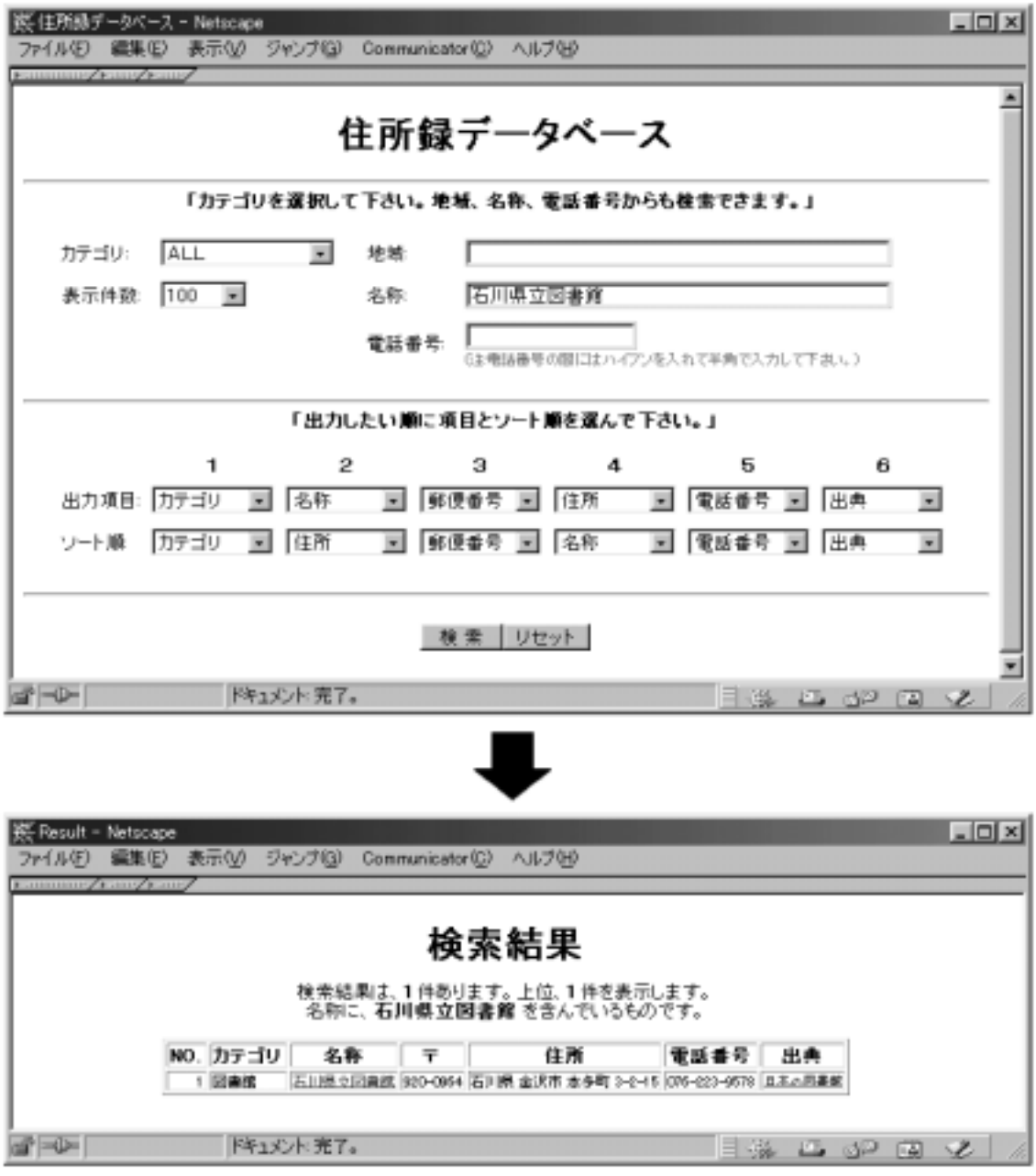



図 2.6: 名称「石川県立図書館」での検索結果

## 2.3 住所情報データベース

本システムが提供する住所情報は、住所情報データベースに格納されている。住所情報データベースは、1 つのテーブルから構成される[1]。その 1 レコードは、以下に示す 8 つのフィールドから構成され、ある特定の対象に対する 1 組の住所情報を表す。以下では、これを**住所レコード**と呼ぶ。

1. 名称
2. カテゴリ
3. 郵便番号
4. 住所
   住所フィールドは、空白によって、さらに、以下の 4 つ（あるいは、5 つ）に分割されている。
   - 都道府県レベル：都道府県名
   - 市区町村レベル：地方公共団体名（郡の場合は郡名と町村名の 2 つのレベルに分割）
   - 町域レベル：町域名
   - 小字名、丁目、番地など：町域名以降の住所
5. 電話番号
6. 抽出元 URL
   この 1 組の住所情報を抽出したページの URL。
7. 抽出元ページタイトル
   この 1 組の住所情報を抽出したページのタイトル（TITLE タグで囲まれた文字列）。
8. 詳細情報掲載ページの URL
   この対象の詳細を掲載している可能性の高いページの URL。

住所レコードの例を表 2.1に示す。

[1] 本システムではデータベース管理システムとして PostgreSQL を使用している。

表 2.1: 住所レコードの例

| フィールド名 | 値 |
|---|---|
| 名称 | 石川県立図書館 |
| カテゴリ | 図書館 |
| 郵便番号 | 920-0964 |
| 住所 | 石川県 金沢市 本多町 3-2-15 |
| 電話番号 | 076-223-9578 |
| 抽出元 URL | http://www.trc.co.jp/trc-japa/guide/jplib/ISIKAWA.htm |
| 抽出元ページタイトル | 日本の図書館 |
| 詳細情報掲載ページの URL | http://www.library.pref.ishikawa.jp |

## 2.4 住所情報収集モジュール

住所レコードは、住所情報収集モジュールによって作成される。本モジュールは、カテゴリ名を入力とし、そのカテゴリに属する対象の住所情報をウェブから抽出し、データベースに登録する処理を行う。本モジュールは、以下に示す 4 つのサブモジュールから構成される（図 2.7）。

1. ウェブページ収集モジュール
   カテゴリ名を入力とし、ウェブ上からそのカテゴリに関する住所一覧が掲載されている可能性が高いページを収集する。

2. テーブル抽出モジュール
   収集したページ内から、住所一覧テーブルの候補とそのテーブルの見出しを抽出する。ページ内に複数の住所一覧テーブルの候補が存在する場合は、その全てを抽出する。詳細は 3 章で説明する。

3. テーブル解析モジュール
   抽出したテーブルを解析し、その解析結果に基づきテーブルを標準形テーブルに整形する。詳細は 3 章で説明する。

4. 住所情報抽出モジュール
   整形したテーブルが、求めるカテゴリのテーブルである場合のみ、そのテーブルから住所情報を抽出する。このとき欠落情報がある場合、それを補完して、データベースに登録する。詳細は 4 章で説明する。

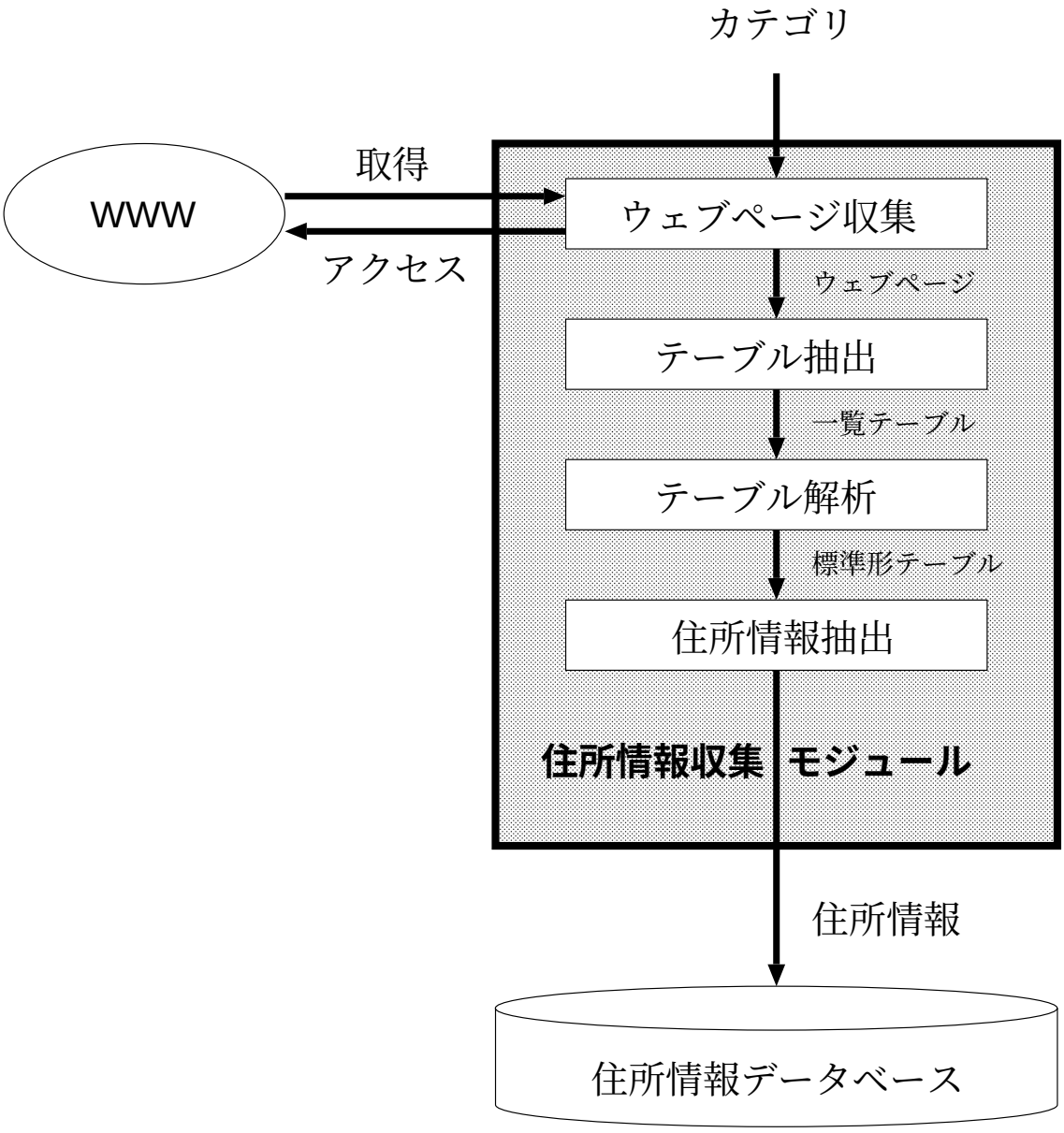



図 2.7: 住所情報収集モジュールの構成

以下では、収集の対象とする住所一覧ページとウェブページ収集について述べる。

### 2.4.1 住所一覧ページ

ウェブ上には、ある特定のカテゴリに属する施設、店舗などの住所を列挙したページが数多く見受けられる。その例を図 2.8に示す。このようなページを本論文では、**住所一覧ページ**と呼ぶ。

一般に、住所一覧ページには、住所を列挙した表あるいはリストと、どのような種類（カテゴリ）の対象の住所を列挙しているかに関する情報が記載されている。住所の列挙形式は、以下の 3 つのいずれかをとる場合が多い。

- テーブル形式（図 2.8）
- リスト形式（図 2.9）
- プレーンテキスト形式（図 2.10）

また、カテゴリの情報は、タイトルや見出しに書かれることが多い。

本研究では、この 3 つの列挙形式のうち、テーブル形式からのみ住所情報を収集する。テーブル形式のみを選んだのは、以下の 2 つの理由による。

1. 情報抽出の効率
   リスト形式、プレーンテキスト形式は、1 レコードごとの表記法の自由度が高いため、データに一貫性がない場合が多く、情報を抽出する上で効率が悪い。これに対しテーブル形式には、住所情報 1 レコードの方向と、フィールド方向が別々に存在しているという特徴がある。つまり、情報が 2 次元に存在しているということである。このため、1 つのレコードのフィールドが判明すれば、他のレコードのフィールドも判明するので、効率良く情報を抽出することができる。

2. 名称の抽出
   リスト形式、プレーンテキスト形式では、名称の判断が困難であることが多い。これは名称の前後に他の単語や文が書かれている場合が多いためである。この場合、○○図書館、××病院などの、名称の接頭語、接尾語などを見ることにより、ある程度、どこからどこまでが名称であるかを判断できる場合もあるのだが、全くそのような接頭語、接尾語を含まない名称も多い。さらに、そういった接頭語、接尾語を用いる場合には、収集するカテゴリごとに適切な接頭語、接尾語の辞書を作成しなければならない。これでは汎用性がなく、実用的でない。これに対しテーブル形


| 名称 | 郵便番号 | 住所 | 電話番号 | 救急 | 総合 |
|---|---|---|---|---|---|
| 興和会右田病院 | 192-0066 | 八王子市本町13-2 | 0426-22-5155 | ○ | － |
| 仁和会総合病院 | 192-0046 | 八王子市明神町4-8-1 | 0426-44-3711 | ○ | ○ |
| 清智会横山記念病院 | 192-0904 | 八王子市小宮町3-24-15 | 0426-24-5111 | ○ | － |
| 生活協同組合多摩相互病院 | 192-0083 | 八王子市旭町3-1 | 0426-22-7268 | ○ | － |
| 徳成会八王子山王病院 | 192-0042 | 八王子市中野山王2-15-16 | 0426-26-1144 | ○ | － |
| 東京都国民健康保険団体連合会南多摩病院 | 193-0832 | 八王子市散田町3-10-1 | 0426-63-0111 | ○ | ○ |
| 井上病院 | 193-0944 | 八王子市館町559-1 | 0426-64-5833 | ○ | － |
| 東京医科大学八王子医療センター | 193-8639 | 八王子市館町1163 | 0426-65-5611 | ○ | ○ |
| 玉栄会東京天使病院 | 193-0811 | 八王子市上壱分方町50-1 | 0426-51-5331 | ○ | － |
| 北原脳神経外科病院 | 192-0045 | 八王子市大和田町1-16-19 | 0426-45-1110 | ○ | － |
| 木下外科・胃腸科病院 | 192-0355 | 八王子市堀之内2463-1 | 0426-75-2121 | ○ | － |
| 八九十会高月整形外科 | 192-0002 | 八王子市高月町360 | 0426-92-1115 | ○ | － |
| 御殿山クリニック | 192-0375 | 八王子市鑓水428-160 | 0426-77-1500 | ○ | － |
| 親和会野後峰脳神経外科 | 192-0372 | 八王子市下柚木1974-1 | 0426-74-1515 | ○ | － |

図 2.8: テーブル形式住所一覧ページの例：
http://www.tokyo-yusei.go.jp/kurashi/kugai/31/list04.html

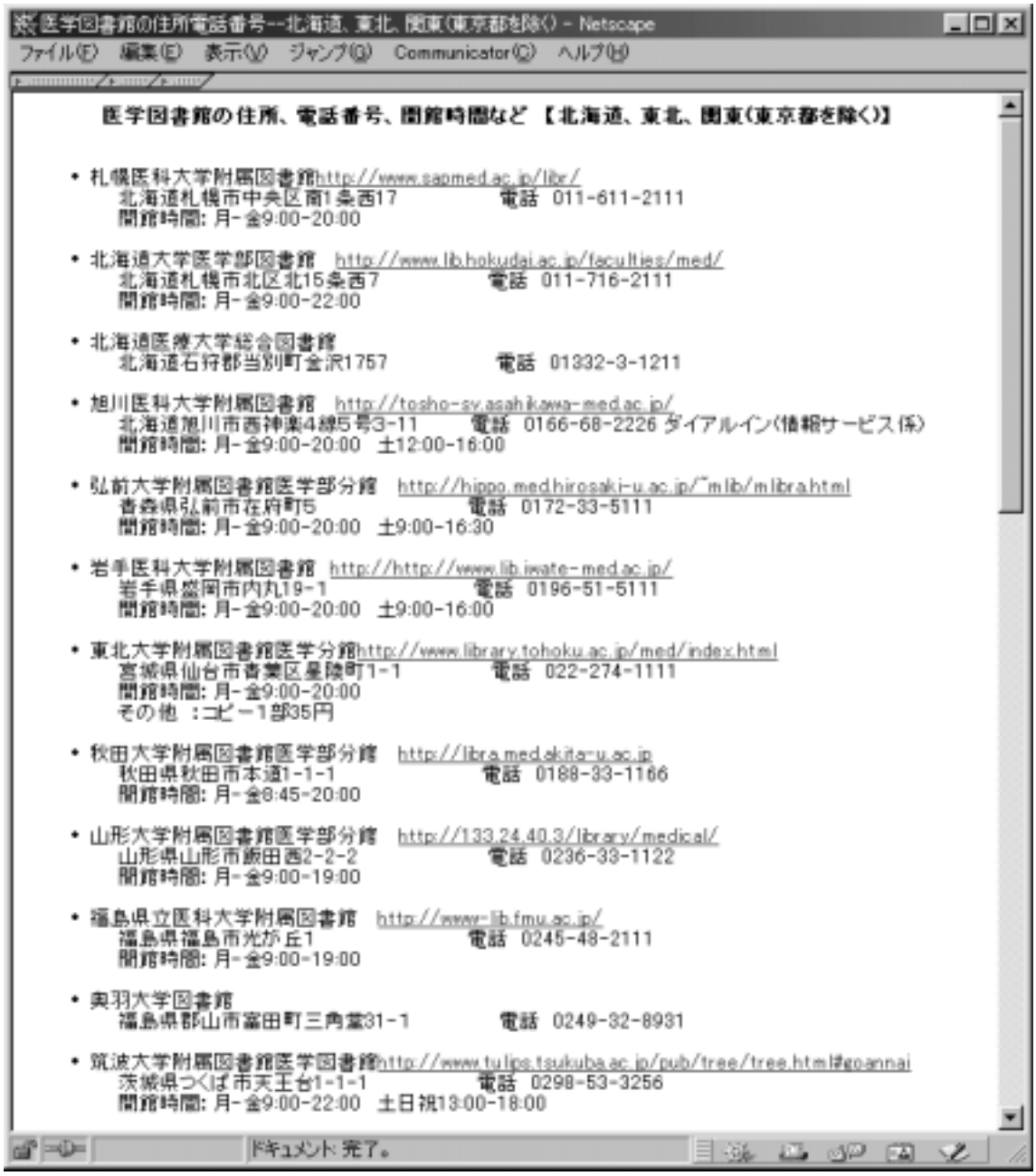



図 2.9: リスト形式住所一覧ページの例：
http://www.asahi-net.or.jp/ AL9Y-IS/hokubuD.html

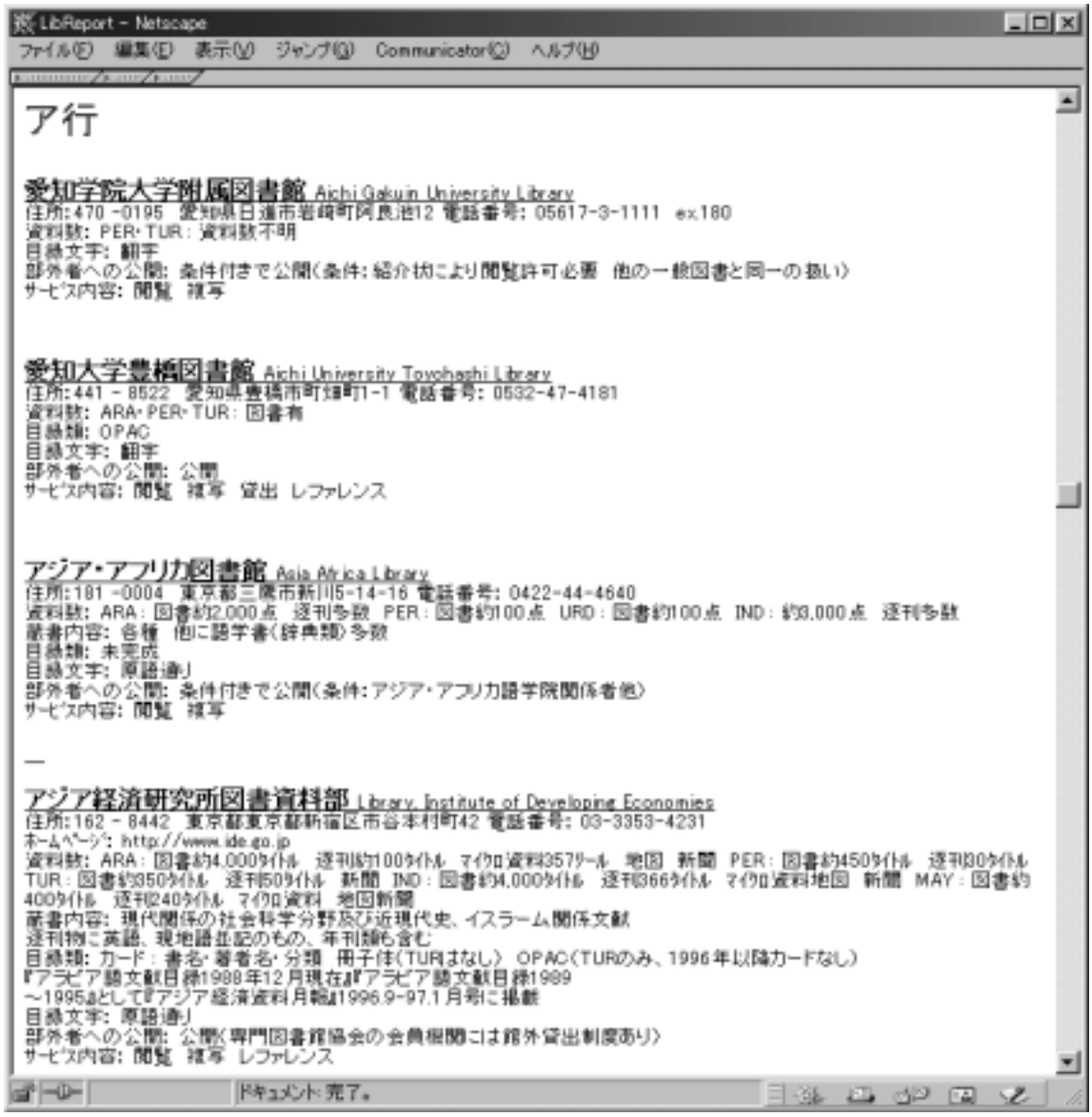



図 2.10: プレーンテキスト形式住所一覧ページの例：
http://www.l.u-tokyo.ac.jp/IAS/6-han/97-98/LibReport.html

式の場合、名称を書く場所が 1 つのセル内に限定されているため、前後に文が来ることなく名称のみが書かれる。このため、接頭語、接尾語辞書を用いることなく容易に名称を抽出することができる。

以上より本研究では、カテゴリ名を与えることのみで、効率良く住所情報を収集することが可能な、テーブル形式で書かれた住所一覧ページのみを利用して、住所情報を収集する方法をとる。

### 2.4.2 ウェブページ収集モジュール

ウェブページ収集モジュールは、カテゴリ名を入力とし、ウェブ上からそのカテゴリの住所一覧ページである可能性が高いページを収集する。

収集には、サーチエンジンを利用する。しかし、単にカテゴリ名をそのまま検索質問として使用するならば、無関係なページも多く検索されてしまうため、効率的に住所一覧ページを収集することは難しい。そこで、住所一覧ページによく現れる単語を付加することにより、収集の効率化を図る。

各サーチエンジンは、ウェブページを収集する方法として、各々が異なったアルゴリズムを用いている。そのため、収集するページに偏りが出てしまうことが多く、検索結果には同じドメインのページが多かったりするので、1 つのサーチエンジンを使うだけでは、ウェブ上の様々な場所から情報を収集することは困難である。また、検索質問に関しても同じことが言える。これらの問題を解決するには、複数のサーチエンジン、複数の検索質問を用いる必要がある。これにより、偏りなく多くのページを収集することが可能になる。以上より、本研究では 2 つのサーチエンジンと 57 種類の検索質問を用いることにする。

実際に使用する 57 種類の検索質問 (query) を表 2.2に示す。表に示したように、住所一覧ページによく現れる、住所や一覧表に関連する語とカテゴリ名で検索することにより、効率良く住所一覧ページを収集する。また、これらの一覧ページは、地域単位でまとめられていることが多く、ページ内に県名が多数現れるので、都道府県名とカテゴリ名で検索を行なう場合も効率良く住所一覧ページを収集することができる。ここでは、goo[2]、Infoseek Japan[3]の 2 つのエンジンを使用し、それぞれの検索質問に対して、最大 100 件の URL を収集し、それらをダウンロードする。つまり、最大では、11400 ページを収集する。

[2] http://www.goo.ne.jp/

[3] http://www.infoseek.co.jp/

表 2.2: 検索質問

| NO. | 検索質問 |
|---|---|
| 1. | カテゴリ名 |
| 2. | カテゴリ名 AND 所在地 |
| 3. | カテゴリ名 AND 住所 |
| 4. | カテゴリ名 AND 一覧 |
| 5. | カテゴリ名 AND リスト集 |
| 6. | カテゴリ名 AND 案内 |
| 7. | カテゴリ名 AND 所在地 AND 一覧 |
| 8. | カテゴリ名 AND 所在地 AND リスト集 |
| 9. | カテゴリ名 AND 住所 AND 一覧 |
| 10. | カテゴリ名 AND 住所 AND リスト集 |
| 11〜57. | カテゴリ名 AND （47 都道府県名） |

# 第 3 章

# テーブル抽出とテーブル解析

本章では、住所一覧ページからテーブルを抽出し、そのテーブルを住所情報が抽出しやすいように、標準形テーブルに整形するまでの一連の処理について詳しく述べる。

## 3.1 テーブル抽出モジュール

ウェブページ収集モジュールによって収集したページ内から、そのページ内に存在する全ての住所一覧テーブルの候補を抽出する。なお、本論文では、テーブルタグを用いて作成された住所一覧のことを**住所一覧テーブル**と呼ぶことにする。本モジュールでは、まず、抽出処理を行ないやすい形にページを整形してから、住所一覧テーブルの抽出を行なう。

ウェブページからテーブルを抽出する際に問題となるのは、テーブルが入れ子構造になっている場合である。この場合、内外どちらのテーブルが一覧テーブルであるかの判断が難しく、そのままの構造だとテーブル解析の処理の妨げになる。何らかの方法でどちらが一覧テーブルであるかを判定し、一覧テーブル以外のテーブルを解体する必要がある。

そこで、どのような場合にテーブルが入れ子構造になるかの調査を行なった。入れ子構造になっている住所一覧テーブルを人手により 30 件収集したところ、その内 28 件がページのレイアウトのためにテーブルを使用し、その中に一覧テーブルがあるという構造であった。残りの 2 件は、一覧テーブル内のある特定セル内にのみ複数の値を書くために使用されていた。そして、これらのテーブルの罫線[1]の使用法に着目したところ、表 3.1に示す関係が観察された。

以上の調査結果は、次のようにまとめられる。

[1] ここでは TABLE タグで BORDER を指定している場合、罫線ありとする。

表 3.1: 入れ子構造の住所一覧テーブルの罫線使用法

| 外側が住所一覧テーブル | | |
|---|---|---|
| 外側テーブルの罫線 | 内側テーブルの罫線 | 件数 |
| 有 | 有 | 0 |
| 有 | 無 | 2 |
| 無 | 有 | 0 |
| 無 | 無 | 0 |
| 内側が住所一覧テーブル | | |
| 外側テーブルの罫線 | 内側テーブルの罫線 | 件数 |
| 有 | 有 | 0 |
| 有 | 無 | 0 |
| 無 | 有 | 22 |
| 無 | 無 | 6 |

- 外側のテーブルに罫線が無い場合は、内側のテーブルが住所一覧テーブルである可能性が高い。
- 内側のテーブルに罫線が有る場合は、内側のテーブルが住所一覧テーブルである可能性が高い。
- 外側のテーブルに罫線が有り、内側のテーブルに罫線が無い場合は、外側のテーブルが住所一覧テーブルである可能性が高い。

以上の結果を踏まえ、本モジュールでは、この関係を利用して住所一覧テーブルを判別し、入れ子構造のテーブルを解体する手法を採用する。

その他の問題としては、テーブルの見出しをどのようにして見つけるのかという問題があるが、テーブルの見出しは多くの場合、何らかの強調タグによって目立つように記載されているので、テーブル上部の強調タグに着目することによって、そのテーブルの見出しを見つける手法を採用する。

以下に、本モジュールで実際に行なう 4 つの処理を示す（図 3.1）。

1. ページ整形
   収集したページ内から住所一覧テーブルを抽出しやすいように、ページを整形する。

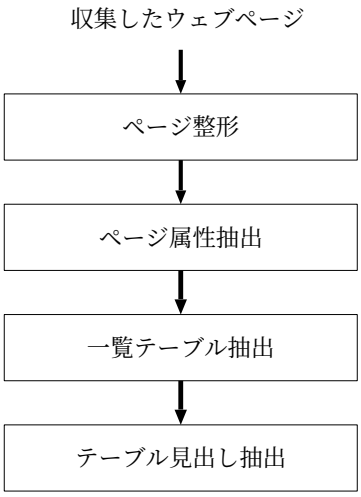



図 3.1: モジュールの処理手順

2. ページ属性抽出
   整形したページ内からページタイトルを抽出する。もしそのページが市区町村レベルでまとめらたものである場合、その地域名も抽出する。本論文では、これを**ページの地域属性**と呼ぶ。

3. 一覧テーブル抽出
   整形したページ内から、TABLE タグで囲まれた文字列を住所一覧テーブルの候補として抽出する。複数存在する場合は全て抽出する。

4. テーブル見出し抽出
   住所一覧テーブルの見出しを抽出する。そのテーブルが市区町村レベルでまとめらたものである場合、その地域名も抽出する。本論文では、これを**テーブルの地域属性**と呼ぶ。ここで、地域名が抽出できなかった場合、2 で抽出したページの地域属性をテーブルの地域属性とする。

以上の４つの処理により、住所一覧テーブルの候補及び、ページタイトル、テーブルの見出し、テーブルの地域属性を抽出する。以下では、各処理内容について具体的に説明する。

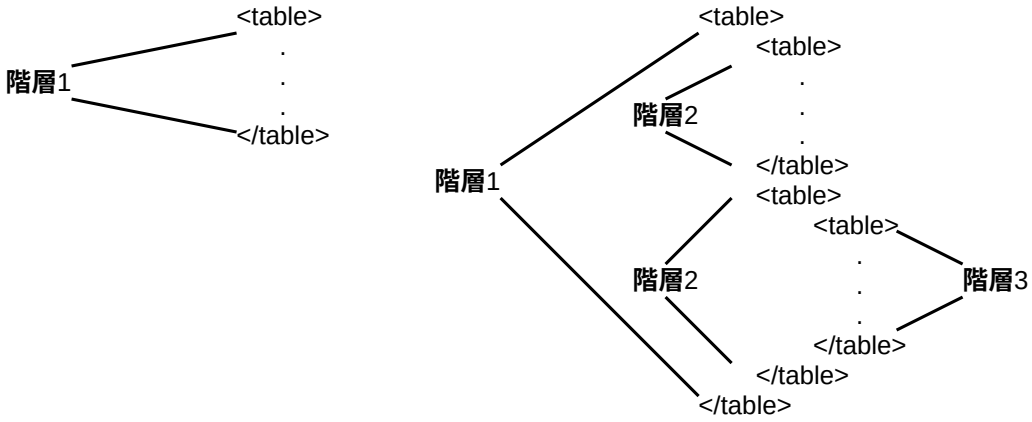



図 3.2: 入れ子テーブルの例

### 3.1.1 ページ整形

収集したページから住所一覧テーブルを抽出しやすいように、ページを整形する。一覧テーブルを抽出する上で問題になる入れ子構造のテーブルを解体し、その他の一覧テーブルになり得ないテーブルも解体する。以下にアルゴリズムを示す。

1. ページ内の全てのテーブル開始タグ、終了タグの位置関係を調べ、入れ子構造になっている場合は、その入れ子の層の深さを調べる。入れ子テーブルの例を図 3.2に示す。

2. 最下層のテーブルと、そのテーブルを含んでいる一つ上層のテーブルの罫線の有無を調べる。

3. 上層と下層の罫線の関係を見て、上層が罫線を使用し下層が罫線を使用していない場合のみ、下層の TABLE タグを削除する。その他の場合は、上層の TABLE タグを削除する。これにより TABLE タグの入れ子構造が解消される。

4. 入れ子構造が何層にもなっている場合は、入れ子構造が解消されるまで、この処理を繰り返す。

5. 次にページ内から１行もしくは１列しかないテーブルを探す。これらのテーブルは一覧テーブルには成り得ないので、TABLE タグ及び、テーブル内の TR、TD、TH タグを削除して解体する。但し、１行１列のテーブルがあった場合のみ、そのテー

ブルは直下のテーブルの見出しである可能性があるので、テーブル内の文字列を見出しタグ CAP-T で囲んでから解体する。

以上の処理により、ページ内には２行２列以上の入れ子構造になっていないテーブルのみが残る。

### 3.1.2 ページ属性抽出

整形したページ内からページタイトルとページの地域属性を抽出する。地域によって分類されているページでは、ページ上部、もしくは下部で市区町村名を記載し、一覧テーブル内の住所には町域名からしか記載しない場合がある。こういったテーブルから住所情報を抽出するための補足情報として、ページの地域属性を抽出する。以下にアルゴリズムを示す。

1. ページ内の TITLE タグで囲まれた文字列をページタイトルとして抽出する。
2. ページの最初から 1 番上にあるテーブルの開始タグまでに市区町村名が書かれていないか調べ、最初に見つかった市区町村名をページの地域属性として抽出する。
3. ページ上部に市区町村名が無い場合は、1 番最後のテーブル終了タグからページの最後までに市区町村名が書かれていないか調べ、最後に見つかった市区町村名を抽出する。

### 3.1.3 一覧テーブル抽出

整形したページ内から、TABLE タグで囲まれた文字列を住所一覧テーブルの候補として抽出する。複数存在する場合は全て抽出する。このとき、テーブル開始タグの上部の文字列も抽出する。これは次の、テーブル見出し抽出時に使用する。テーブルが複数存在する場合は、一つ上のテーブルの終了タグ以降の文字列を抽出する。

### 3.1.4 テーブル見出し抽出

抽出したテーブルの見出しと、地域属性を抽出する。以下にアルゴリズムを示す。

1. テーブルの見出しは、テーブルの上部に見出しタグや各種強調タグを用いて書かれている場合が多いので、テーブル見出し抽出用に抽出した文字列内から、強調タグで囲まれた文字列をテーブル見出しＡとして抽出する。強調タグで囲まれた文字列

が複数存在する場合は、一番下部にある文字列を抽出する。このパターンマッチングに用いる強調タグは「H」「FONT SIZE」「B」「STRONG」「EM」と、テーブル抽出モジュールで付与する「CAP-T」の 6 種類である。

2. 抽出したテーブル内に CAPTION タグが存在するか調べる。CAPTION タグが存在した場合、タグで囲まれている文字列をテーブル見出しＢとして抽出する。

3. テーブル見出しＢ、Ａ、ページタイトルの順で市区町村名が存在するかどうか調べていき、存在した場合、それをテーブルの地域属性として抽出する。これは、そのテーブルが何らかの地域によってまとめられたものである場合、その地域名をテーブルの見出しに書いている場合が多いからである。存在しない場合は、3.1.2 で抽出したページの地域属性をテーブルの地域属性とする。

以上の処理により、住所一覧テーブルの候補と、ページタイトル、テーブル見出しＡ、Ｂ、テーブルの地域属性が抽出される。

## 3.2 テーブル解析モジュール

本モジュールは、住所一覧テーブルから住所情報を容易に抽出するために、テーブルを標準形テーブルに整形する処理を行なう。標準形テーブルとは、図 2.8のような、1 行に 1 レコードづつ書かれていて、1 行目にはテーブルヘッダ[2]があり、複数のセルにまたがっているセルが 1 つもないテーブルのことである。

テーブルを標準形テーブルに整形するに際し、本研究では、まず、テーブルの属性を解析し、それからその解析結果に基づきテーブルを標準形テーブルに整形する 2 ステップを踏む方式を採用する。これは、解析しながらテーブルを整形する方式では対応が困難な、複雑なテーブル形式に対応するためである。

これらの処理を行なうに際し、以下の 3 つの問題がある。

1. テーブルヘッダの無いテーブルが存在する
   通常、一覧テーブルにはテーブルヘッダがあるはずなのだが、中には無いテーブルも存在し、この場合、各フィールドの属性を判断することが困難になる。

2. 1 つのセルが複数のセルにまたがっている
   1 つのセルが複数のセルにまたがっていると、実際にそこにあるはずのセルの値が何であるか分からないため、情報の抽出が困難になる。

[2]本論文では、テーブル内の各フィールドの属性名が書いてある行（列）を、テーブルヘッダと呼ぶ。

3. HTML の表記が正しくない
   テーブル内の「TD」「TH」タグなどの閉じタグが書かれていないなどの HTML の表記ミスがある場合、機械処理の妨げになる。

1 に関しては、住所一覧テーブルには 1 レコードに住所が 1 つ必ずあり、電話番号や、郵便番号が書かれている場合が多いという点や、1 レコードの項目の並び順は住所より先に名称があるなどのヒューリスティックスを利用することにより、各フィールドの属性を推測し、仮のテーブルヘッダを作成してテーブル上部に付与するという手法を用いることで対応する。

2 に関しては、1 つのセルが複数のセルにまたがるのは、並列している複数のセルの値が同じなので 1 つのセルにまとめた場合と、テーブルの見出しとして 1 行をまとめている場合（以降ではこの行のことを見出し行と呼ぶ）の 2 つが考えられる。これより、1 行がまとめられている場合は、見出し行としてその行を抽出し、それ以外の場合は、複数のセルにまたがっているセルを分割し、各セルに同じ値を与える方法を用いる。

3 に関しては、テーブル内の各開始終了タグの関係を調べ、不足しているタグがある場合はそれを付与することにより HTML の表記ミスを正す。

以下では、テーブル解析と標準形テーブルへの整形の各処理について詳しく説明する。

### 3.2.1 テーブルの各属性の解析

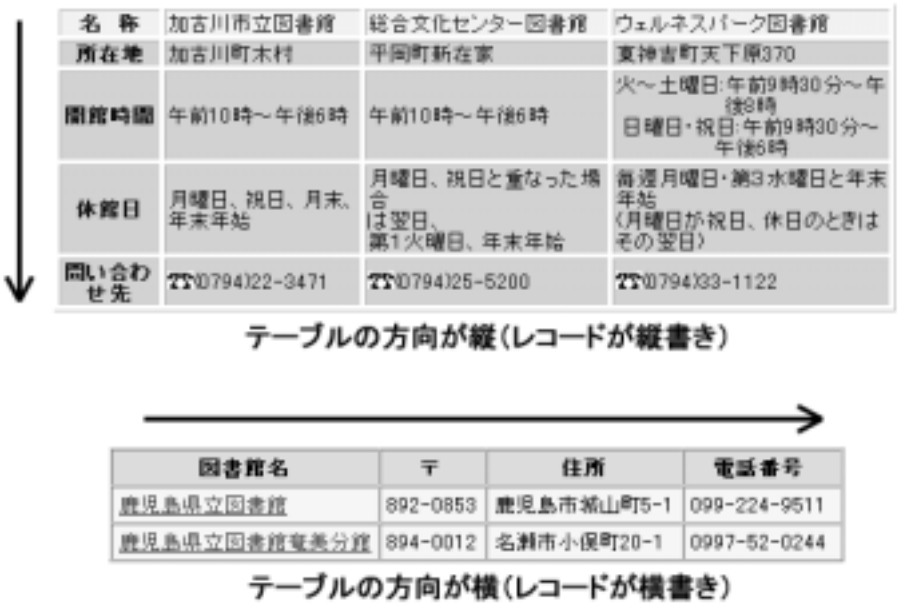
ここではテーブル抽出モジュールで抽出したテーブルの 6 つの属性を解析する。解析には、一覧テーブルによく用いられるテーブルレイアウトの特徴、テーブルヘッダの内容、テーブル内の住所の位置関係などを利用する。

まず、ウェブ上に存在する様々なテーブルの特徴を表現するために、以下の 6 つの属性を決定した。

1. テーブル方向
   1 レコードが書かれている方向（図 3.3）。

2. テーブルヘッダの有無
   フィールド名を示すテーブルヘッダの存在の有無（図 3.4）。

3. テーブルヘッダの行（列）数
   テーブルヘッダに用いられている行（列）数（図 3.5）。



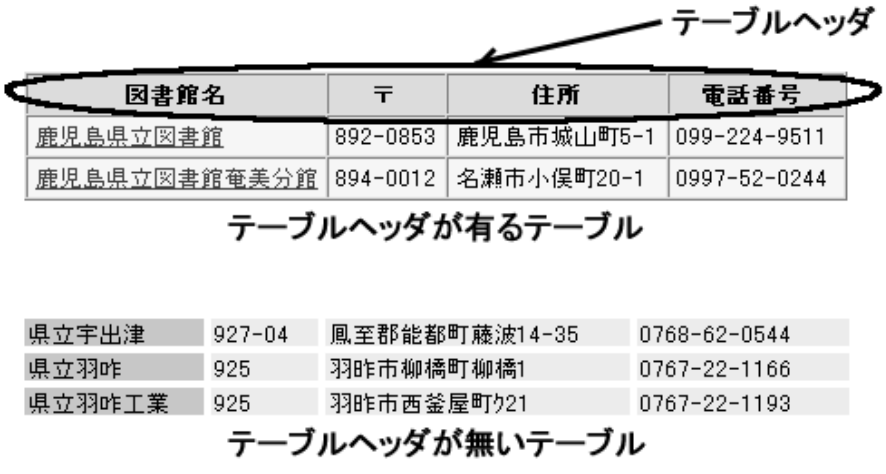
| 名　称 | 加古川市立図書館 | 総合文化センター図書館 | ウェルネスパーク図書館 |
|---|---|---|---|
| 所在地 | 加古川町木村 | 平岡町新在家 | 夏神吉町天下原370 |
| 開館時間 | 午前10時～午後6時 | 午前10時～午後6時 | 火～土曜日:午前9時30分～午後8時<br>日曜日・祝日:午前9時30分～午後6時 |
| 休館日 | 月曜日、祝日、月末、年末年始 | 月曜日、祝日と重なった場合<br>は翌日、<br>第1火曜日、年末年始 | 毎週月曜日・第3水曜日と年末年始<br>(月曜日が祝日、休日のときはその翌日) |
| 問い合わせ先 | TT0794)22-3471 | TT0794)25-5200 | TT0794)33-1122 |

テーブルの方向が縦(レコードが縦書き)

| 図書館名 | 〒 | 住所 | 電話番号 |
|---|---|---|---|
| 鹿児島県立図書館 | 892-0853 | 鹿児島市城山町5-1 | 099-224-9511 |
| 鹿児島県立図書館奄美分館 | 894-0012 | 名瀬市小俣町20-1 | 0997-52-0244 |

テーブルの方向が横(レコードが横書き)

図 3.3: テーブル方向




| 図書館名 | 〒 | 住所 | 電話番号 |
|---|---|---|---|
| 鹿児島県立図書館 | 892-0853 | 鹿児島市城山町5-1 | 099-224-9511 |
| 鹿児島県立図書館奄美分館 | 894-0012 | 名瀬市小俣町20-1 | 0997-52-0244 |

テーブルヘッダが有るテーブル

| | | | |
|---|---|---|---|
| 県立宇出津 | 927-04 | 鳳至郡能都町藤波14-35 | 0768-62-0544 |
| 県立羽咋 | 925 | 羽咋市柳橋町柳橋1 | 0767-22-1166 |
| 県立羽咋工業 | 925 | 羽咋市西釜屋町ク21 | 0767-22-1193 |

テーブルヘッダが無いテーブル

図 3.4: テーブルヘッダの有無


図 3.5: テーブルヘッダの行数

4. １レコード当たりの行（列）数
   １レコードに用いられている行（列）数（図 3.6）。

5. 見出し行の有無
   テーブル内の見出し行の有無。

6. 表の数
   複数の表が見出し行によって連結され、１つのテーブルになっている場合の表の数（図 3.7）。

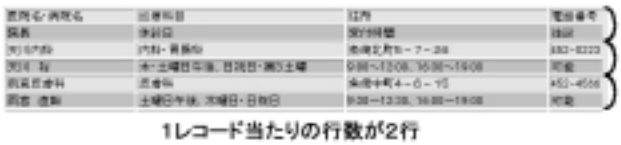
以下の手順によって、一覧テーブル内から、これら 6 つの属性の値を決定する。

### (１) 見出し行の探索

見出し行の数を調べる。見出し行の判定方法は、まず、テーブルの列数を調べ、それから TD、TH タグ内で COLSPAN を用い列数と同数を指定しているセルが存在する行を探す。この方法で見つかった行が 1 行でも存在する場合、見出し行ありと判定する。見出し行が 2 行以上ある場合は、そのテーブルはその行数分の表が結合したものであると判定し、それを表数とする。1 行の場合は表数 1 と判定、見出し行なしの場合は、見出し行なし、表数 1 と判定する。判定後、全ての見出し行を削除する。

### (２) テーブル内の複数セルにまたがっているセルを分割

まず、テーブル内の TD、TH タグで COLSPAN を用いているセルを探し、見つかった場合、そのセルをその COLSPAN で指定している数のセルに分割する。各セル内の値は分割する前のセルと同じものを入れる。次に ROWSPAN を用いているセルを探し、見つ



1レコード当たりの行数が2行

1レコード当たりの行数が1/2行

図 3.6: 1 レコード当たりの行数


図 3.7: 見出し行と表の数

かった場合、COLSPAN と同様にセルを分割する。以上の処理により、複数のセルにまたがっているセルは全て、1 行 1 列のセルに分割される。

### (3) テーブルヘッダの探索

テーブルの 1 行 2 列目から 1 行目の各セル内をテーブルヘッダパターンと照合する。テーブルヘッダパターンとは、住所一覧表のテーブルヘッダでフィールド名として用いられている語のことである。これらの語を表 3.2に示す。

表 3.2: テーブルヘッダパターン

| |
|---|
| 施設名、名前、名称、<br>住所、所在地、アドレス、ADDRESS、ＡＤＤＲＥＳＳ、<br>電話、TEL、ＴＥＬ、連絡先、問い?合わ?せ、<br>郵便番号 |

この照合により、テーブルヘッダの有無とテーブル方向を判定する。これらの語が見つかった場合、そのテーブルにはテーブルヘッダがあり、横書きであると判定する。見つからなかった場合は、次に 1 列 2 行目から 1 列目の各セル内をパターンと照合し、見つかった場合は、そのテーブルにはテーブルヘッダがあり縦書きであると判定する。この場合、処理の簡単化のために、行列を反転して、表を横書きの状態にする。見つからなかった場合は、テーブルヘッダなしと判定する。

### (4) テーブルヘッダの行数の調査

(3) でテーブルヘッダありと判定された場合、2〜4 行目も同じようにパターンと照合する。2 行目にパターンが存在しない場合、テーブルヘッダ行数は、1 行と判定、2 行目にパターンが存在し、3 行目に存在しない場合、テーブルヘッダ行数は 2 行と判定、3 行目に存在し、4 行目に存在しない場合は、テーブルヘッダ行数は 3 行と判定。4 行目にも存在する場合は、その表は解析不能とする。

### (5) テーブルの方向の調査

(3) でテーブルヘッダなしと判定された場合、各セル内に住所が書かれているか調べ、その住所の位置関係より、テーブルの方向を判定する。どこにも住所が書かれていない場合、そのテーブルは住所一覧テーブルでは無いと判定する。

**(６)1 レコード当たりの行数の調査**

テーブル内の各セル内に住所が書かれているか調べ、その住所の位置関係より、テーブル内の 1 レコード当たりの行数を調べる。例えば、1 行に 2 レコード存在する場合は、1/2 行、2 行で 1 レコードの場合は 2 行と判定する。

### 3.2.2 テーブルの標準化

解析した 6 つのテーブル属性を元に、テーブルを標準形テーブルに整形する。処理手順を以下に示す。

1. 見出し抽出とテーブル分割
   見出し行が存在する場合、その値をテーブル見出しＣとして抽出する。表数が複数ある場合は、テーブルを見出し行ごとに分割する（図 3.8）。

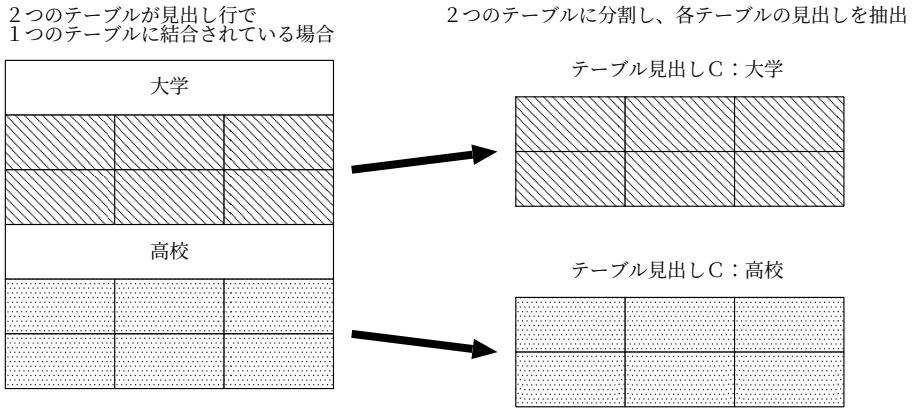



図 3.8: 見出し抽出とテーブル分割

2. 複数セルにまたがるセルの削除
   テーブル内に複数のセルにまたがっているセルがある場合、そのセルがまたがっているセル数分のセルに分割して同じ値を持つ複数のセルに変換する（図 3.9）。

3. テーブル方向変換
   テーブル方向が縦の場合、テーブルの行列を反転させる（図 3.10）。

4. テーブルヘッダ抽出
   テーブルヘッダが存在する場合、テーブルヘッダを抽出し 1 行のテーブルヘッダに

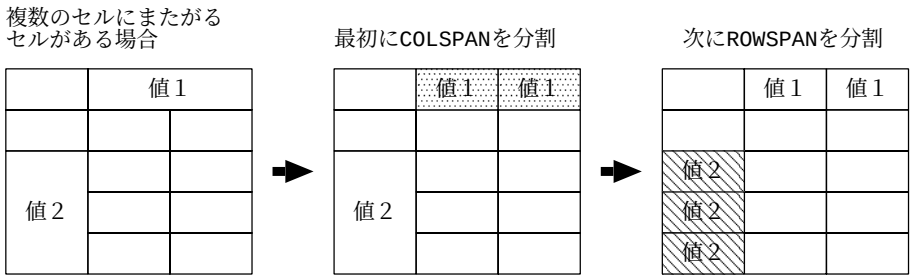



図 3.9: 複数セルにまたがるセルの削除

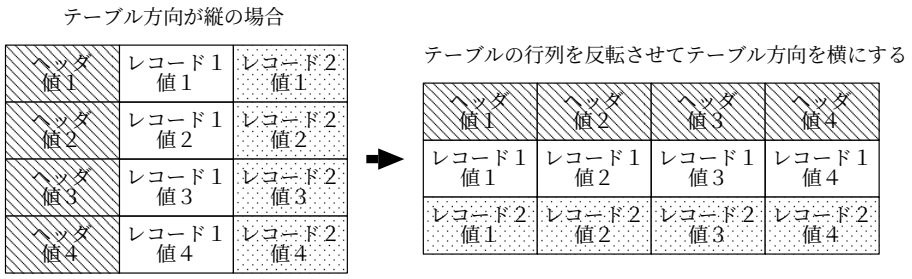



図 3.10: テーブル方向変換

変換する。

このとき、テーブルヘッダが複数行で 1 レコード当たりの行数も複数行の場合は、テーブルヘッダを横に展開して 1 行にする。テーブルヘッダが複数行で 1 レコード当たりの行数が 1 行の場合、各行の同じ列の値をスラッシュで区切って統合し、1 つのセルに入れることにより 1 行にする。この処理により、テーブル展開後のレコード行との整合がとれる（図 3.11）。

また、見出し行により複数のテーブルが 1 つのテーブルに結合されているものに関しては、テーブル分割後の 1 番上のテーブルからのみテーブルヘッダの抽出を行なう。そして、このテーブルヘッダを、分割された全てのテーブルのテーブルヘッダとして扱う。この処理は、見出し行で複数のテーブルが結合されている場合、最初のテーブルにしかテーブルヘッダが書かれていないことがあるので、その場合に対応するために行なう。

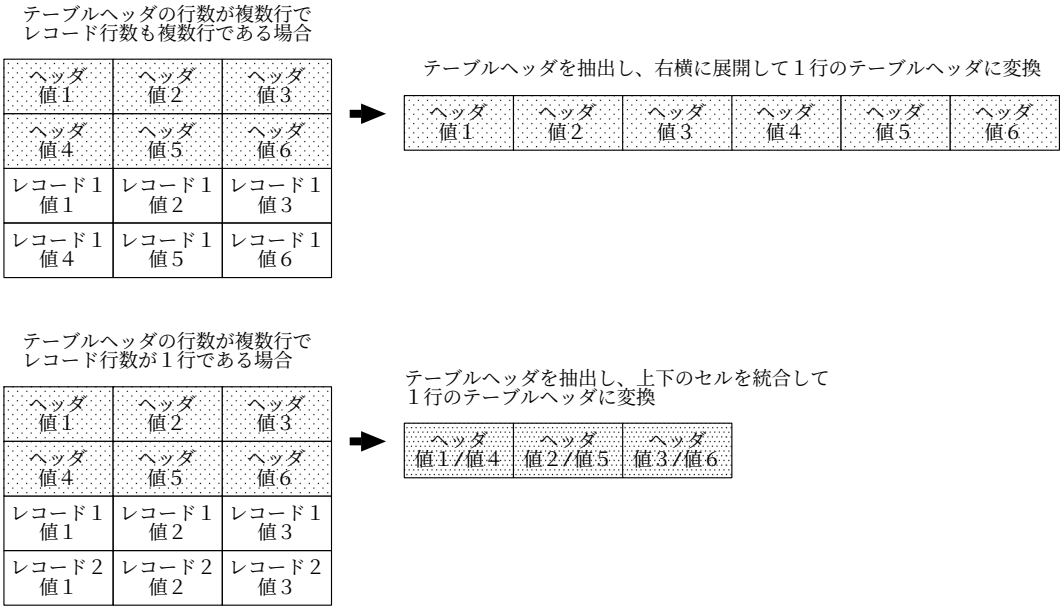



図 3.11: テーブルヘッダ抽出

5. テーブル展開

テーブルを 1 レコード 1 行のテーブルに展開する。1 レコード当たりの行数が複数行の場合、2 行目以降を 1 行目の右横に展開して 1 行にする。1 レコード当たりの行数が 1 行未満の場合、横に複数のテーブルが接続されているものと考えて、2 つめ

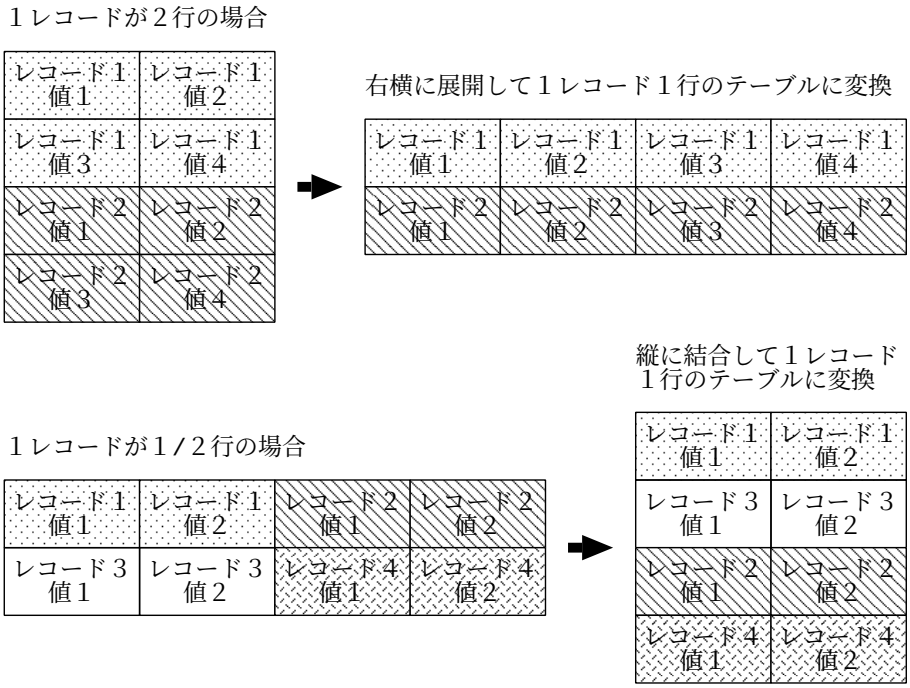



図 3.12: テーブル展開

以降のテーブルを 1 つ目のテーブルの下に結合して 1 レコード 1 行の形にする（図 3.12）。

6. 仮テーブルヘッダ作成
   テーブルヘッダが存在しない場合、1 レコード目の値から各フィールドを推測し、そのフィールド名をヘッダ行として生成する。郵便番号、住所、電話番号は、パターンマッチングによって推測し、フィールド名とする。他にも数字のみのフィールドには「番号」、ひらがな、かたかな一文字のフィールドには「50 音分類」、地方、地区、地域と書かれたフィールドには「地域分類」というフィールド名をつける。これらのフィールド以外で、住所フィールドに一番近い左側にある列を名称フィールドとする。以上のフィールド推測を元にテーブルヘッダを作成する。

7. テーブルとテーブルヘッダの連結
   展開したテーブルとテーブルヘッダを連結し、標準形テーブルにする（図 3.13）。

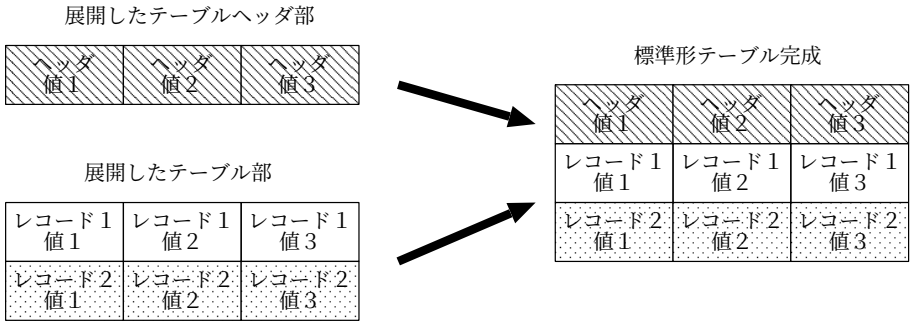



図 3.13: テーブルとテーブルヘッダの連結

## 3.3 テーブル抽出とテーブル解析の実行例

テーブル抽出とテーブル解析の実行例を図 3.14に示す。この実行例の処理手順を以下に示す。

1. ページ整形
2. ページ属性抽出（ページタイトル：与野市学校一覧、ページの地域属性：与野市）
3. 一覧テーブル抽出
4. テーブル見出し抽出（テーブル見出しA：与野市内国公立学校一覧、テーブル見出しB：なし、テーブルの地域属性：与野市）
5. テーブルの各属性の解析（テーブル方向：横、テーブルヘッダの有無：あり、テーブルヘッダの行数：1、1 レコード当たりの行数：1/2、見出し行の有無：あり、表の数：1）
6. テーブルの標準化（テーブル見出しC：●高等学校）

以上のような手順で図 3.14のページから、標準形テーブル、ページタイトル、テーブル見出しA、B、C、テーブルの地域属性を得る。


図 3.14: テーブル抽出とテーブル解析の実行例

# 第 4 章

# 住所情報の抽出

本章では、住所情報抽出モジュールについて説明する。標準形テーブルからの情報抽出方法と、欠落情報の補完方法について詳しく述べる。

## 4.1 住所情報抽出時の問題点

住所情報抽出モジュールは、まず、そのテーブルが対象とするカテゴリのテーブルであるかを判定し、それから住所情報を抽出、データベースに登録する処理を行なう。

住所情報を抽出するにあたっての 2 つの課題を以下に示す。

1. 住所一覧テーブルのカテゴリの判定法
   その一覧テーブルが対象とするカテゴリのテーブルであるかどうかを何らかの方法で判定する必要がある。

2. 欠落情報の補完
   住所が町域レベルからしか表記されていない場合、同じ名称の町域が複数あるので正確な住所の判定は困難である。また、市外局番のない電話番号、郵便番号が 3 桁や 5 桁の旧郵便番号で表記されている場合も多い。このように情報の一部が欠落している場合、何らかの方法でこの欠落を補完する必要がある。

1 に関しては、カテゴリの情報は、タイトルや見出しに書かれているか、テーブルヘッダの名称部に用いられていることが多いので、その中にカテゴリ名が書いてあるかどうかによって判定を行なう方法をとる。

2 に関しては、ページが市区町村単位でまとめられている場合に多く見られる表記法で、その場合、ページの上部か下部、もしくはテーブルの見出しに市区町村名が書いてあるこ

とが多い。本研究では、それらの場所から市区町村名を探しだし、町域レベルの住所と統合することにより、正確な住所を得る手法を採用する。電話番号、郵便番号に関しては、住所を元に市外局番と 7 桁の郵便番号を得る方法をとる。

## 4.2 住所情報抽出アルゴリズム

本モジュールでは、まず、そのテーブルが対象とするカテゴリのテーブルであるかどうかを調べる。対象カテゴリのテーブルであることが判明したら、テーブルヘッダから各フィールド属性を決定し、それを元に 1 レコードづつ住所情報を抽出し、住所レコードとしてデータベースに登録する。本モジュールの処理手順を以下に示す。

1. 住所一覧テーブルのカテゴリ確認
   標準形テーブルから住所情報を抽出するに先立ち、そのテーブルが収集対象とするカテゴリの住所一覧テーブルであるか確認する。今までに取得した、テーブル見出しＡ、Ｂ、Ｃ、ページタイトル及び、そのテーブルのテーブルヘッダ内のいずれかに、収集対象のカテゴリ名が存在するか調べる。存在した場合のみ 2 へ進む。

2. フィールド判定
   標準形テーブルのテーブルヘッダから各フィールド属性を決定する。フィールド属性の決定方法は表 4.1のパターンとテーブルヘッダのパターンマッチングによって行なう。表のパターンは、サンプルとして収集した 79 件の住所一覧ページを参考に決定した。

表 4.1: フィールド属性名パターン一覧表

| フィールド属性 | パターン |
|---|---|
| 名称部 | (カテゴリ)、(カテゴリ) 名、施設名、名前、名称、店名、店舗名、企業名、事業所名、学校名、館名 |
| 郵便番号部 | 郵便番号、〒 |
| 住所部 | 住所、所在地、アドレス、ADDRESS、ＡＤＤＲＥＳＳ |
| 電話番号部 | 電話番号、電話、TEL、ＴＥＬ、連絡先、問い?合わ?せ |

また、名称フィールドのヘッダ部のみが空白になっているテーブルが存在するので、名称フィールドが見つからない場合は、住所部の左側から空白のヘッダを探し、見つかった場合、そのフィールドを名称フィールドとする。

3. 住所情報抽出

判明したフィールド属性を元に各行から住所情報を抽出する。抽出する住所情報は以下の 5 つである。

(a) 名称

対象とする住所情報の名称を抽出。

(b) 詳細情報掲載ページの URL

名称にリンクが張られている場合、そのリンク先がその名称の詳細情報を掲載しているページである可能性が高いので、それを詳細情報掲載ページの URL として抽出する。

(c) 郵便番号

ハイフンで区切られた 7 桁の数字の場合のみ抽出する。数字、ハイフンが全角で書かれている場合は、数字、ハイフンともに半角に変換する。

(d) 住所

市区町村レベル以上から書かれていて、町域レベル以下まで書いてあるもののみ抽出する。これを、都道府県レベルから書かれた正確な住所に変換してからデータベースに登録する。

(e) 電話番号

市外局番から書かれている番号のみ抽出する。市外局番や市内局番を括弧で囲ってある場合はそれをハイフンに変換する。数字、ハイフンが全角で書かれている場合は、数字、ハイフンともに半角に変換する。

4. 欠落情報の補完

上記の住所情報抽出時に、各セル内の値が不完全なものである場合、他の情報を利用して、その値を完全なものにする処理を行なう。この処理は以下の 3 つについて行なう。

(a) 住所の補完

住所が町域レベルからしか書かれていない場合、その上のレベルの都道府県、市区町村名は一つに定まらないので、都道府県名からの正確な住所を取得することができない。市区町村名が判明すれば都道府県名も判明するので、正確な住所を取得するためには最低でも市区町村名を知る必要がある。そのため、あらかじめ一覧ページの上部、下部、タイトル、テーブル見出しなどの市区町村名が書かれている可能性の高い場所を調べおき、市区町村名が書かれている場

合はそれを取得しておく。そして、その市区町村名と町域レベルの住所を統合することにより、正確な住所情報を取得する。

(b) 郵便番号の補完
郵便番号が 3 桁、5 桁、もしくは存在しない場合、先に抽出した住所を用いて全国郵便番号データーベースを検索し、その住所の郵便番号を抽出する。全国郵便番号データーベースは郵政省のホームページ[1]内の「住所の新郵便番号ダウンロードサービス」からダウンロードしたファイルを利用して作成したものである。

(c) 電話番号の補完
電話番号に市外局番が無い場合、先に抽出した住所を用いて全国市外局番データベースを検索し、市外局番を取得して、それを電話番号の前部にハイフンで区切って追加する。全国市外局番データベースは、http://www.gs.niigata-u.ac.jp/ naga/kensaku.html 内の市外局番検索のデータを元に作成したものである。

5. 住所情報登録
抽出した住所情報に、名称と住所の両方が存在する場合のみ、これに、カテゴリ名、ページタイトル、抽出元 URL を追加したものを 1 レコードの住所レコードとしてデータベースに登録する。このときデータベース内に、名称、カテゴリ、住所の 3 つが同一のレコードが存在した場合、その住所レコードが同一情報であると判断し、登録は行なわない。但し、既に登録されているレコードに、郵便番号、電話番号、詳細情報掲載ページの URL のいずれかが無く、登録しようとしていたレコードにそれらが存在する場合は、情報統合としてそれらの値を既存レコードに追加する。このとき、郵便番号、電話番号に関しては、その情報の抽出元ページの URL を付加しておき、検索モジュールで情報を提供するときに、番号にその URL のリンクが張られるようにする。

## 4.3 住所情報抽出の実行例

住所情報の抽出例として、まず、一般的な例を図 4.1と表 4.2に示す。図 4.1のページから表 4.2の情報が抽出される。

[1] http://www.postal.mpt.go.jp


図 4.1: 抽出元ページ 1：http://chukakunet.pref.kagoshima.jp/pref1/tosyokan/kenritu.htm

表 4.2: 図 4.1 のページから抽出される住所レコードの表

| フィールド名 | 値 |
|---|---|
| 名称 | 鹿児島県立図書館 |
| カテゴリ | 図書館 |
| 郵便番号 | 892-0853 |
| 住所 | 鹿児島県 鹿児島市 城山町 5-1 |
| 電話番号 | 099-224-9511 |
| 抽出元 URL | http://chukakunet.pref.kagoshima.jp/pref1/tosyokan/kenritu.htm |
| 抽出元ページタイトル | 県立図書館一覧 |
| 詳細情報掲載ページの URL | http://chukakunet.pref.kagoshima.jp/pref1/index.html |

次に情報が補完される例を図 4.2と表 4.3に示す。図 4.2のページから表 4.3の情報が抽出される。このとき補完される情報は、郵便番号の 112-0002、住所の文京区、電話番号の市外局番の 03 である。


図 4.2: 抽出元ページ 2：http://www.city.bunkyo.tokyo.jp/benri/shussan/hoikuen.html

表 4.3: 図 4.2 のページから抽出される住所レコードの表

| フィールド名 | 値 |
|---|---|
| 名称 | 久堅保育園 |
| カテゴリ | 保育園 |
| 郵便番号 | 112-0002 |
| 住所 | 東京都 文京区 小石川 5-27-7 |
| 電話番号 | 03-3811-0712 |
| 抽出元 URL | http://www.city.bunkyo.tokyo.jp/benri/shussan/hoikuen.html |
| 抽出元ページタイトル | 便利帳／出産・子ども・教育／子ども/保育園一覧 |
| 詳細情報掲載ページの URL | なし |

# 第 5 章

# 実験と検討

本章では、作成した住所録自動生成システムの抽出精度と有効性を評価するために行なった 2 つの実験の結果を述べ、その結果について考察する。

## 5.1 カテゴリ「図書館」に対する実験

情報源を限定することにより、本システムが、ウェブ上に存在する住所情報をどの程度の割合で収集してくるのかを評価する実験を行なった。以下でその実験について述べる。

### 5.1.1 実験方法

実験方法は、カテゴリ「図書館」に関する住所一覧ページを人手で収集し、それを情報源として本システムで住所情報の抽出を行なわせる方法をとった。実験に使用するデータは、本システムのテーブル解析能力を知るために、できる限り多くの種類の住所一覧ページを収集する必要があるので、以下の方法を用いて収集した。

- Goo と InfoseekJapan の 2 種類のサーチエンジンを用いてウェブページを収集
- 各サーチエンジンに対し、「カテゴリ名」「カテゴリ名 AND 住所」「カテゴリ名 AND 一覧」の 3 つの検索質問を使用
- 各検索質問に対し、上位 50 件のウェブページを収集

以上の方法で収集したウェブページ 300 件中、同一ページを削除した 272 件について、人手でテーブルタグを用いて住所情報が記載されているページを選別した。選別条件は以下の 3 つである。

- 最低でも名称と町域レベルの住所（この場合ページ内に市区町村名が存在すること）がテーブル内に存在
- 2 行 2 列以上のテーブル
- ページ内にカテゴリ名が書いてあること（名称に使われているものは除く）

以上の全ての条件を満たしているページから人手によって収集した住所レコードと、272 件のウェブページからシステムが抽出した住所レコードを比較することにより、システムの抽出精度 [3] を調べた。評価尺度には、再現率と適合率を用いる。

$$\text{再現率} = \frac{\text{抽出した正しい住所情報}}{\text{全ページ中の正しい住所情報}} \tag{5.1}$$

$$\text{適合率} = \frac{\text{抽出した正しい住所情報}}{\text{抽出した住所情報}} \tag{5.2}$$

### 5.1.2 実験結果

この実験の結果を表 5.1に示す。

表 5.1: カテゴリ「図書館」に対する実験の結果

| | 人手 | システム |
|---|---|---|
| 抽出レコード数 | 268 | 226 |

システムが抽出した 226 件中、1 件の住所レコードが、人手で抽出したデータ以外のものであり、全く別のカテゴリの住所レコードであった。それ以外の住所レコードに関しては、全て人手で抽出したものと同じものであった。

次に、システムが抽出した住所レコード 226 件について、各項目の内容を評価した。評価は人手で抽出した住所レコードを元に、その内容と同一であれば正、違っている、もしくは情報を抽出できなかった場合を誤と判定することにより行なった。判定結果を表 5.2 に示す。

この結果で住所の抽出に失敗したレコードは全て、電話番号でも失敗していたため、実際に何らかの誤りがあったレコード件数は 9 件であり、全ての情報が正しかったレコード件数は 217 件であった。以上の結果、システムの再現率と適合率は以下のようになる。

表 5.2: カテゴリ「図書館」に対する実験によってシステムが抽出した住所レコードの判定結果

| 項目 | 正 | 誤 | 合計 |
|---|---|---|---|
| カテゴリ | 225 | 1 | 226 |
| 名称 | 225 | 0 | 225 |
| 郵便番号 | 224 | 1 | 225 |
| 住所 | 220 | 5 | 225 |
| 電話番号 | 167 | 7 | 174 |
| 住所レコード | 217 | 9 | 226 |

$$\text{再現率} = \frac{217}{268} = 81\% \tag{5.3}$$

$$\text{適合率} = \frac{217}{226} = 96\% \tag{5.4}$$

まず、再現率は 81%であり、本システムの収集能力が十分実用的なレベルであるということを示す結果になった。また、適合率は 96%と非常に高い数値であり、本システムの抽出精度の高さを示す結果を得ることができた。

次に、住所情報の抽出に失敗したレコードについて、その原因を調査した。まず、一覧ページから全く住所情報が抽出できなかったレコードについて調査した。抽出失敗の原因には、次の 5 種類のものがあった。

1. 住所一覧ではない。
2. システムがレコード行数を誤って判断した。
3. 記載されている住所に不備がある。
4. 住所に略字が使用されている。
5. 町域レベルが数字のみである。

1 に関しては、本システムでは、ある一つの対象の住所情報を表記するためにテーブルタグを使用しいている場合でも、表記方法が住所一覧と同様であればそのテーブルから情報を抽出するのだが、中には、レイアウトに凝るあまり、表記方法が非常に複雑になっているものがあり、そのようなテーブルから情報が収集できなかった場合であった。

2 に関しては、テーブルヘッダのないテーブルで、実際は 1 レコード当たりの行数が 1 行であるのに、1 行目の住所が正しく記載されていなかったため、システムが 1 レコード当たりの行数を 3 行と誤って判断してしまい、3 行に 1 レコードの割合でしか住所情報を抽出しなかったという場合であった。

3〜5 に関しては、住所の表記に関する問題で、3 は、町域名の最後に「町」が必要な住所であるにも関わらず、それが省かれていた場合であった。4 は、その住所の正式な漢字ではなく略字が使用されて書かれていた場合であった。5 は、町域レベルの住所が数字のみの住所である場合にシステムがそれを町域と判断することができない場合であった。3、4 の場合に関しては、情報自体に不備があるのだが、いずれの場合も本システムの住所判定方法をもう少し強力なものにすれば対応可能な問題である。

以上のように抽出失敗の原因の多くは、一覧ページに記載されている住所に何らかの不備があった場合であった。よってこれらの情報を抽出しなかったのは、システムとして正しい処理を行なったと言える。また、本システムは、ユーザへ提供する情報の信頼性を高めるために適合率に重きをおいて作成しているので、再現率が 81%というのは、高い数字であると言える。

次に、抽出した住所レコードの内容の一部に誤りがあったものについて、その原因を調査した。その結果、抽出失敗の原因には、次の 3 種類のものがあった。

1. 住所と電話番号が同一セル内に書かれている。

2. システムがレコード行数を誤って判断した。

3. 郵便番号の補完失敗。

1 に関しては、一つのセル内に住所と電話番号の両方が書かれていて、住所の丁目、番地、号の数字と電話番号の数字が連続して書かれていたため、システムが間違った場所でその数字を区切ってしまい、住所と電話番号ともに誤ったものになってしまったという場合であった。

2 に関しては、テーブルヘッダのないテーブルで、実際は 1 レコード当たりの行数が 1 行であるのに、1 行目と 3 行目の住所が正しく記載されていなかったため、システムが 1 レコード当たりの行数を 2 行と誤って判断してしまい、1 行目から電話番号を、2 行目からは名称と住所を抽出してしまったという場合であった。

3 に関しては、システムのバグであると思われる。

以上のような抽出の失敗はあったものの、適合率は 96%と非常に高く、システムとして活用する上で十分な抽出精度を得ることができた。

## 5.2　30 カテゴリに対する実験

本システムの有効性と汎用性を評価するために、複数のカテゴリに対して実際にウェブ上から住所情報を収集する実験を行なった。以下でその実験について述べる。

### 5.2.1　実験方法

本システムの有効性を評価するために、表 5.3に示す 10 分野 30 種類のカテゴリに対して、実際にウェブ上から住所情報を収集する実験を行なった。収集した住所情報の中からランダムに 300 件の住所情報を抽出し、その住所情報の適合率を評価した。また、カテゴリの種類により、どの程度収集量に差がでるかも評価した。なお、このカテゴリは、本システムの汎用性を評価するためにインターネットイエローページ [4][5] などを参考に様々な分野から選定した。

表 5.3: 収集カテゴリ一覧表

| 分野 | カテゴリ名 |
|---|---|
| 教育機関 | 大学、高校、中学校、小学校、幼稚園、保育園 |
| 各種施設 | 図書館、美術館、博物館 |
| 医療機関 | 病院、薬局 |
| 礼拝施設 | 神社、寺院 |
| 宿泊施設 | ホテル、民宿、旅館 |
| 飲食店 | レストラン、飲食店、ラーメン店、居酒屋 |
| スポーツ | スキー場、ゴルフ場 |
| 販売店 | スポーツショップ、釣具店、書店、パソコンショップ、ブティック |
| 自動車 | 自動車販売、自動車整備 |
| サービス | 美容室 |

### 5.2.2　実験結果

実験の結果、総収集レコード数は 32071 件であった。各カテゴリごとの収集件数を表 5.4に示す。

この結果より、カテゴリごとに収集量のばらつきはあるものの、全てのカテゴリから情報を収集していることから、本システムは汎用性の高いシステムであることが示された。

表 5.4: 各カテゴリごとの収集件数

| カテゴリ名 | 収集件数 | カテゴリ名 | 収集件数 |
|---|---|---|---|
| 大学 | 859 | 旅館 | 548 |
| 高校 | 3482 | レストラン | 814 |
| 中学校 | 832 | 飲食店 | 418 |
| 小学校 | 251 | ラーメン店 | 204 |
| 幼稚園 | 1530 | 居酒屋 | 126 |
| 保育園 | 1744 | スキー場 | 290 |
| 図書館 | 984 | ゴルフ場 | 1066 |
| 美術館 | 536 | スポーツショップ | 144 |
| 博物館 | 2692 | 釣具店 | 47 |
| 病院 | 5339 | 書店 | 2363 |
| 薬局 | 1759 | パソコンショップ | 18 |
| 神社 | 57 | ブティック | 103 |
| 寺院 | 522 | 自動車販売 | 53 |
| ホテル | 3317 | 自動車整備 | 19 |
| 民宿 | 1741 | 美容室 | 213 |

次に、各カテゴリごとの収集量について考察する。教育機関や各種施設などの公的な機関や医療関係などの人間が生活する上で必要な施設に関しては、多くのレコードを収集することができた。また、ウェブ上での閲覧頻度の高いホテルなどの宿泊施設なども多く収集している。これらの収集量が多かったものの共通点として、企業や公的団体が住所一覧ページを作ることが多いカテゴリである点が挙げられる。それとは逆に、店舗関連の情報はあまり収集することができなかった。この原因として考えられるのは、これらのカテゴリの一覧ページを作成するのは、その殆どが個人であるためだと思われる。個人が作るページでは、記載方法が適切なものでなかったり、情報が不足していたりするので、あまり収集されなかったのではないかと思われる。その他、個人の作るページでは、テーブルヘッダの内容があまり一般的でないものが用いられている場合が目についた。本システムでは、システムに汎用性を持たせるために、テーブルヘッダの判定には一般的なものしか想定しなかったが、これらのページに対応するためには、カテゴリごとにテーブルヘッダのマッチングパターンを考えればよい。

次に、収集した住所レコードからランダムに 300 件を抽出し、そのレコードの内容を評価した。評価は各レコードの抽出元ページの内容を元に、その内容と同一であれば正、違っている、もしくは抽出できなかった場合を誤と判定することにより行なった。判定結果を表 5.5に示す。

表 5.5: 30 カテゴリに対する実験によってシステムが抽出した住所レコードの判定結果

| 項目 | 正 | 誤 | 合計 |
|---|---|---|---|
| カテゴリ | 270 | 30 | 300 |
| 名称 | 264 | 6 | 270 |
| 郵便番号 | 263 | 7 | 270 |
| 住所 | 262 | 8 | 270 |
| 電話番号 | 239 | 19 | 258 |
| 住所レコード | 237 | 63 | 300 |

この結果で複数の項目の抽出に失敗したレコードが 7 つあったため、実際に何らかの誤りがあったレコード件数は 63 件であり、全ての情報が正しかったレコード件数は 237 件であった。以上の結果、本システムの適合率は以下のようになる。

$$\text{適合率} = \frac{237}{300} = 79\% \tag{5.5}$$

実験結果は、カテゴリ「図書館」に対する実験より、かなり低い結果になってしまった。しかし、失敗した件数の内、30 件がカテゴリの判定ミスによるものであり、カテゴリの判定が正しい場合の住所レコードの適合率は 88%と、システムとしは十分実用レベルにある結果を得ることができた。また、このことより、本システムのカテゴリ判定率は 90%であることもわかった。

次に、ランダムに抽出した 300 件の住所レコードで内容の一部に誤りがあったものについて、その原因を調査をした。調査結果を各項目ごとに示す。

1. カテゴリ
   これは、全く違うカテゴリのものが抽出されているという訳ではなく、幼稚園と保育園、ホテルと民宿と旅館、博物館と美術館などの、分野としては同じものの中で間違われている場合がほとんどであった。この原因は、住所一覧が同じ分野の複数のカテゴリについて、まとめて書かれている場合があるからである。例えば、宿泊施設一覧という形で住所一覧としてまとめられている場合などがこれに当たる。この場合、各レコードにカテゴリの内容が書かれている場合と書かれていない場合がある。書かれていない場合は、名称から推測するしかないのだが、推測できない名称である場合もある。このことを踏まえると、本システムを実用化するにあたり、どのようなカテゴリで分類するのかは非常に重要になってくると言える。

2. 名称
   誤った名称を抽出してしまった場合が 6 件あった。原因は以下の 2 つであった。

   (a) 住所一覧ではない。

   (b) テーブルヘッダがない一覧テーブルで間違ったフィールドから名称を抽出した。

   (a) に関しては、カテゴリ「図書館」に対する実験時にもあった問題であるが、住所一覧ではないのにテーブルを使用している場合に、レイアウトに凝るあまり表記方法が非常に複雑になっているものがあり、そのため、誤った名称を抽出してしまった場合である。このような住所一覧ではないものに関しては、何らかの方法で住所一覧と判別して、別途対応する必要があるのかもしれない。(b) に関しては、仮テーブルヘッダ作成時のフィールド推測のパターンを増やすことで対応できる。

3. 郵便番号
   郵便番号の補完ミスが 7 件あった。これは、システムのバグであると思われる。

4. 住所

   住所の末尾にゴミが付着してしまった場合が 7 件あった。これは、住所の後に連続で別のものが書かれている場合に、システムが誤って余分なものを抽出してしまった場合である。この点に関しては、もう少し精度を高める必要がある。

   残りの 1 件は、テーブルヘッダのないテーブルで、実際は 1 レコード当たりの行数が 1 行であるのに、1 行目の住所が正しく記載されていなかったため、システムが 1 レコード当たりの行数を 3 行と誤って判断してしまい、1 行目から電話番号を、2 行目から住所を、3 行目から名称を抽出してしまったという場合であった。

5. 電話番号

   電話番号が記載されているのに、それを抽出できなかったことが 15 件あった。原因は以下の 3 つであった。

   (a) 電話番号が住所や名称と同じセル内に書かれているのに、テーブルヘッダには住所や名称のみしか書かれていない。

   (b) 電話番号の市外局番、市内局番、お客様番号が空白で区切られている。

   (c) システムが新市外局番に対応していない。

   (a) に関しては、電話番号部を表すヘッダがない場合は、住所や名称のフィールドから電話番号を探す機能をシステムに追加すれば解決するものと思われる。(b) に関しては、表記方法に問題があるので対応する必要はない。(c) に関しては、市外局番を補完する際にシステムが古い市外局番を補完してしまった場合である。これには現在のシステムで用いている全国市外局番データベースに変更前と変更後の両方の市外局番を登録して対応する必要がある。

   また、誤った電話番号を抽出してしまった場合が 4 件あった。その原因は以下の 2 つであった。

   (a) システムがレコード行数を誤って判断した。

   (b) システムがテーブルヘッダ行数を誤って判断した。

   (a) に関しては、テーブルヘッダのないテーブルで、実際は 1 レコード当たりの行数が 1 行であるのに、1 行目の住所が正しく記載されていないなどの理由で、システムが 1 レコード当たりの行数を複数行と誤って判断してしまい、名称とは別の行から電話番号を抽出してしまった場合であった。(b) に関しては、実際のヘッダ行の 1 つ下の行にヘッダ項目が記載されていたため、システムがヘッダ行を 2 行と誤って

判断してしまったのと、そのテーブル内に 1 レコードが 2 行のものと 1 行のものが混在しているという状況が重なったために起こってしまった失敗であった。

実験の結果、適合率は 79%とやや低いものになったが、郵便番号に関しては、抽出失敗件数全てにおいてデータを登録しているわけではなく、また、電話番号に関しても抽出失敗件数 19 件中 15 件がデータを登録しているわけではないので、実際のデータの信頼性を落しているわけではない。また、この場合、他の一覧ページからその情報を補完する可能性もあるので、それほど大きな問題ではない。住所に関しても、全く異なる住所をデータとして登録した件数は 1 件であり、その他の場合は住所末尾にゴミがついているだけなので、実際にそのデータを使用する上では、さほど問題にはならない。これらのことより、実際にシステムを使用する上での情報の信頼性はもう少し高いものであり、システムとして運用するのに問題のないレベルであると言える。

また、カテゴリ「図書館」に対する実験時より、適合率が大幅に下がった理由としては、カテゴリの種類によって、カテゴリの判定をしやすいものとそうでないものがある点が挙げられる。図書館は、一覧ページとして独立で存在する場合が多いので、カテゴリを判定しやすかったものと思われる。

## 5.3 検討

本システムの再現率は 81%と高く、住所情報を収集する能力は十分実用レベルにあることを示すことができた。また、適合率は 79%という多少低い結果に終ったが、これはカテゴリの判定に失敗したものが多かったためであり、カテゴリが正しく判定されたものに関しては、88%という結果を得ることができた。さらに、カテゴリが正しく判定されたときの失敗のほとんどは、データの信頼性を損なう失敗ではなかったことから、本システムは、十分実用レベルに達していると言える。また、本システムのカテゴリ判定率は 90%という高い値を得ることができた。

今後、本システムの精度をより高めるためには、以下の 3 つの方法が考えられる。

1. テーブルヘッダのマッチングに用いる項目を各カテゴリごとに用意する。
2. テーブルヘッダがない場合の、仮テーブルヘッダ作成時のフィールド推測のパターン数を増やす。
3. 住所一覧でなく、1 つの対象のみの住所情報が記載されているテーブルに関しては、別処理にして対応する。

その他、本システムでは、名称の異表記については取り扱わなかったが、この問題を解決することにより、より精度の高いシステムを作ることが可能になる。異表記への対応法に関しては、佐藤ら [6] の研究のように、住所、電話番号などから同一の対象であると思われる住所情報を複数集め、その中から多数決をとる方法が考えられる。

# 第 6 章

# 結論

本論文では、ウェブ上に存在する住所情報を自動的に収集して、ユーザの要求する形式の住所録として自動編集し提供するシステムについて述べた。

本システムは、住所情報収集モジュール、検索モジュール、住所情報データベース、の 3 つの要素から構成される。住所情報収集モジュールは、住所一覧ページを情報源とすることにより、カテゴリの情報が含まれた住所情報の収集を行なう。住所情報データベースは、住所情報収集モジュールで収集した住所情報を格納するデータベースである。検索モジュールは、住所情報データベース内の情報から、ユーザの指定通りの住所録を自動生成し、ユーザに提供する。

本システムの大きな特徴として、カテゴリごとに住所情報を収集する点が挙げられる。通常、ウェブ上に存在する住所情報のカテゴリを機械的に判断するのは非常に困難であるのだが、カテゴリごとにまとめられている住所一覧ページを情報源とすることにより、それを実現した。また、ページ内の情報を元にした欠落情報を補完する方法を提案し、実現した。

本システムはウェブ上から様々なカテゴリの住所情報を収集できるように汎用性を持たせて作成されており、任意に選択した 30 カテゴリに関してウェブ上で住所情報の収集を行なったところ、約 32,000 件の住所情報を収集することができた。その住所情報の適合率は 79%であった。また、システムのカテゴリ判定率は 90%であり、正しくカテゴリ判定された住所情報の適合率は 88%であった。

本システムの作成により、ウェブ上から必要な情報のみを効率良く収集する方法と、収集した情報をユーザの要求する形式に編集して提供する方法を示した。今後、さらにインターネットが発展するに伴い、このような収集した情報をユーザのニーズに合わせて編集し、提供するシステムの需要は、ますます増えてくるものと思われる。その実現方法を示

した点で本研究は意義のあるものであったと言える。

今後の課題としては、名称の異表記への対応が挙げられる。これにより、より精度の高いシステムを作成することが可能になる。

# 謝辞

本研究を進めるにあたり、多くの御教示を賜りました佐藤理史助教授に深く感謝致します。そして、日ごろから技術的にも精神的にも支援してくださいました知識工学講座の皆様に心から感謝の意を表します。

# 参考文献


[1] 井上香織, 横路誠司, 高橋克巳, 広告の自動構造化, 情報処理学会研究報告, 99–NL–132, pp34–39, 1999.

[2] Larry Wall, Tom Christiansen, Randal L. Schwartz 共著, 近藤嘉雪 訳, プログラミング Perl 改訂版, オライリー・ジャパン, 1997.

[3] 長尾眞, 黒橋禎夫, 佐藤理史, 池原悟, 中野洋, 言語情報処理, 岩波書店, 1998

[4] 日本経済新聞社編, 日経インターネットイエローページ 1999, 日本経済新聞社, 1998.

[5] [厳選] 日本のホームページ 10 万 '99 年版, 株式会社アスキー, 1998.

[6] Satoshi Sato and Madoka Sato: Toward Automatic Generation of Web Directories. *Proc. of International Symposium on Digital Libraries 1999(ISDL'99)*, pp.127-134, 1999.