DEWS2006 5B-i6

Google Maps API を応用したロボット型施設検索システムの試作

長屋　務† 森本　泰貴† 藤本　典幸†† 出原　博†† 萩原　兼一††

† 大阪大学大学院情報科学研究科コンピュータサイエンス専攻
〒 560–8531 大阪府豊中市待兼山町 1–3
†† 大阪大学大学院情報科学研究科コンピュータサイエンス専攻
〒 560–8531 大阪府豊中市待兼山町 1–3
E-mail: †{t-nagaya,hiroki-m}@ics.es.osaka-u.ac.jp, ††{fujimoto,h-idehar,hagihara}@ist.osaka-u.ac.jp

あらまし 近年，インターネットの普及とともに，飲食店などの様々な施設をウェブ上で検索する機会が多くなった．しかし現在の施設検索サイトは，事前に登録されている施設の情報を返すものがほとんどであり，得られる結果が限られてしまうという問題点がある．そこで我々は，Google Web APIs と Google Maps API を応用したロボット型施設検索システムを試作した．本システムは施設の種別などのキーワードと地名を入力とし，ウェブをクロールして地域内の施設に関する情報を得る．そしてユーザに，施設の位置と各種情報を表示したスクロール可能な地図を提示する．これにより，データベースへの施設の登録という作業を省き，より多くの施設情報を得ることができる．また，様々な施設の地理的分布を視覚的に知ることも可能となる．

キーワード 地図検索，情報抽出，トピック主導型クローリング，Google Web APIs，Google Maps API

A Prototype System for Web Retrieval of Institutions with Google Maps API

Tsutomu NAGAYA†, Hiroki MORIMOTO†, Noriyuki FUJIMOTO††,
Hiroshi IDEHARA††, and Kenichi HAGIHARA††

† Graduate School of Information Science and Technology, Osaka University
Machikaneyama-machi 1–3, Toyonaka-shi, Osaka, 560–8531 Japan
†† Graduate School of Information Science and Technology, Osaka University
Machikaneyama-machi 1–3, Toyonaka-shi, Osaka, 560–8531 Japan
E-mail: †{t-nagaya,hiroki-m}@ics.es.osaka-u.ac.jp, ††{fujimoto,h-idehar,hagihara}@ist.osaka-u.ac.jp

Abstract Recently, with the spread of the Internet, various institutions like restaurants can be retrieved by Web sites. However, almost all of such Web sites return information on institutions registered in advance by human. So, such Web sites have a problem that the returned information is restricted. To solve the problem, we developed a prototype system for robot type retrieval of institutions with Google Maps API. The system crawls on the Web to get information on the institutions specified by given keywords and place names. Then, the system shows a map with the found institutions and their information extracted from the crawled Web pages. The system can reduce work for registration of information on institutions and show a geographical distribution of various institutions.

Key words Map retrieval, information extraction, topic-focused crawling, Google Web APIs, Google Maps API

1. まえがき

近年，インターネットの普及により，ウェブ上で飲食店や駅などの施設を検索することが一般的になってきている．施設を検索する方法としては様々なものが考えられるが，一般に利用されているのは，ぐるなび[1] のような施設の種別に特化したサイト，それに Google ローカル[2] のような施設に特化しないサイトの 2 つに大別される．ぐるなびでは，地域や料理の種類，価格などによって条件を絞り込むことにより，結果として登録されているサイトを得る．また Google ローカルでは，地名と施設の種別をフォームに入力することにより，それらの条件に合致する施設の示された地図を得，また地図上に示された

施設をクリックすることにより，施設についての基本的な情報を得る．しかしこれらのサイトからは，あらかじめデータベースに登録されている施設の登録されている情報しか得ることができない．

そこで我々は，ロボットを用いてウェブをクロールし，ユーザが指定した種別の施設に関して情報を収集しユーザに提示するという，ロボット型施設検索システムが有用であると考えた．本システムでは，まず入力として与えられたキーワードと大まかな地名をもとに，独自のアルゴリズムで動作するロボットを用いてウェブをクロールする．そうして得られた複数の施設に関する様々な情報を，Google Maps API を使用したスムーズにスクロール可能な地図上に反映し，ユーザに提示する．ユーザは，施設の位置を表わす地図上のマーカをクリックすることにより吹き出しを表示させ，その中に表示される文書から施設の情報を得ることができる．ここで得られる施設の情報はウェブをクロールして収集したものであり，登録式の施設検索サイトと違い，個人的に施設評価のウェブサイトを開設している人による第三者的視点からの評価といった，ウェブ上で極めて規模の小さい範囲からの情報も得ることができる．

2. 関 連 研 究

ウェブ上には，施設を検索するためのサイトが多種多様に存在している．ここでは，それらのサイトの一部を紹介し，試作したシステムとの相違点について述べる．

- **グルメ情報検索サイト ぐるなび** [1]

ぐるなびは，飲食店の検索に特化した登録式の施設検索サイトである．地域，予約可能日，最寄り駅，和食や洋食といった料理の種類など，豊富な条件で，あらかじめ登録されている飲食店を検索することができる．最終的に一つの飲食店を決定すると，住所やアクセス方法，営業時間，平均予算，総席数，それにメニューなど，飲食店ならではの情報を得られる．また，付近の地図を表示することもできる．

試作したシステムとの違いとして，まず検索可能な施設の種類が挙げられる．ぐるなびは飲食店に特化した検索しかできないのに対し，試作システムではどのような種類の施設も検索可能である．また，施設の情報はロボットを用いてウェブをクロールして得るため，ぐるなびのデータベースに登録されていないような飲食店も検索可能である．しかし，ぐるなびは施設に特化している分，複雑な検索条件を指定することが可能となっている．これは試作システムにはない利点である．

次に挙げられる違いとして，取得できる情報の種類がある．ぐるなびを使用すると，飲食店に必要な情報を網羅的に得られるが，逆に言えばそれ以外の情報を得ることができない．それ以外の情報としては，例えば利用者による飲食店の評判が挙げられる．ぐるなびにも「口コミ掲示板」というシステムがあるが，これは各飲食店に対して掲示板が存在するわけではなく，地域ごとに掲示板が存在しており，口コミ情報を調べるのには手間がかかる．その点試作システムでは，ウェブをクロールして情報を収集することにより，住所や営業時間といった情報から口コミ情報まで，あらゆる種類の情報を得ることができるという利点がある．また，試作システムでは，複数の飲食店をスムーズにスクロール可能な地図上に一度に表示することができる．これは複数の飲食店を比較する上で，大きな利点となると思われる．

なお，ぐるなびでは 2006 年 1 月 8 日現在，ベータ版として地図検索という機能が提供されている．これは，試作システムと同様に，スムーズにスクロール可能な地図上に，複数の飲食店を表示することができる機能である．しかし，少なくとも現段階では，和食や洋食といった飲食店での料理の種類でのみ表示・非表示の切り替えが可能となっており，あまり詳細な検索はできない．当然こちらの地図検索についても，ぐるなびでは飲食店に関する情報しか得られないため，施設の種別に依存しない検索が可能という点で，試作システムの方が優れている．なお，ぐるなびのスムーズにスクロール可能な地図は，地図サイト Mapion [5] を運営している株式会社サイバーマップ・ジャパンから技術提供を受けている．

- **楽天トラベル** [3]

楽天トラベルは，宿泊施設に特化した登録式の施設検索サイトである．このサイトは，宿泊施設の検索だけでなく直接予約を行うこともできる．試作システムはあくまで検索に的を絞っているので，予約という機能がある点においては，楽天トラベルの方が便利だと言えよう．具体的な検索方法としては，チェックイン日，チェックアウト日，利用人数，それに宿泊料金の範囲などを指定し，空室検索ができる．宿泊施設においては，検索の過程において宿泊料金や部屋の種類（ツインルームかダブルルームかなど）が重要になって来るが，それらの情報を複数の施設について表示し，比較検討することが可能となっている．最終的に一つの宿泊施設を選択し，その施設独自のページが用意されているという点ではぐるなびと同じだが，楽天トラベルにはそのページに「お客さまの声」という機能が用意されている．これは，宿泊施設の利用者が実際にその施設を利用した際の感想などを書き込み，宿泊施設の提供者がそれに返答することができる機能である．ぐるなびの「口コミ掲示板」と似た機能であるが，宿泊施設一軒一軒に対して機能が提供されているという点で，大きく異なる．また，その他の機能として，地名や最寄りの駅名・空港名から宿泊施設の検索を行ったり，高速バスや航空券，それにレンタカーといったものの予約まで可能となっている．

試作システムとの違いは，まずぐるなびと同様に，試作システムでは検索対象が施設の種別に依存しないことと，それに試作システムではロボット式で検索を行うため，登録されていない宿泊施設でも検索可能であることが挙げられる．また，複数の宿泊施設を地図上に表示し直感的に選ぶことができる点や，宿泊施設の評判を地図上のマーカを次々とクリックしていくだけで得られるという点において，試作システムの方が優れていると言えよう．ただ，宿泊施設に肝心の予約という機能に関しては試作システムの想定する機能の範囲外であるため，この点に関しては楽天トラベルの方が優れていると言える．

- **i タウンページ** [4]

i タウンページは，施設の種別に特化しない登録式の施設検索

サイトである．ジャンル（グルメや旅行・宿泊など）と地域を指定することで，複数の施設について，施設名，電話番号，住所といった基本的な情報を得ることができる．また，地図を表示することも可能である．更に「くちコミサービス」という機能があり，これは利用者が施設ごとに口コミ情報を投稿できるというものである．

試作システムとの違いは，まずｉタウンページが登録式の施設検索サイトであるのに対し，試作システムがロボット式の施設検索システムであるということが挙げられる．次に，ｉタウンページでは取得できる情報が極めて基本的なものに限られるということがある．試作システムでは，住所などの基本情報のほか,「くちコミサービス」で得られるような利用者からの評価も得られ，加えて営業時間などの施設に特化した情報も得られるため，得られる情報量はこちらの方が多い．更には，地図上に複数の施設を表示し，比較検討できるという利点もある．

なお，ｉタウンページ内のｉタウンページラボというサイトでは，施設の種別は飲食店に限られるが，スムーズにスクロール可能な地図上に複数の飲食店を表示し，また各施設について基本的な情報を得ることができる．スクロール可能な地図は，ぐるなびの地図検索と同じく，株式会社サイバーマップ・ジャパンから技術提供を受けている．

- **Google ローカル**[2]

Google ローカルは，施設の種別に特化しない登録式の施設検索サイトである．地名と施設の種別を入力することにより，検索結果の複数の施設をスムーズにスクロール可能な地図上に表示することができる．更に，地図上の施設の位置を示すマーカをクリックすることにより，吹き出しが表示され，そこから施設名，住所，それに電話番号といった基本的な情報を得ることができる．吹き出しの中には，関連ページやレビューのページへのリンクが表示される場合があり，その場合はリンクをたどることによって，より詳細な情報を得ることが可能である．

試作システムとの大きな違いは，登録式かロボット式かという点である．Google ローカルでは登録されている施設の情報しか得られないが，試作システムではより多くの施設に関する，より多くの情報が得られる可能性が期待できる．

3． 試作したシステム

3.1 システムの概要

試作したシステム GMPSearch の概要について述べる．本システムのスクリーンショットを図 1 に示す．

本システムでは，施設の種別などのキーワードと地名，それにウェブをクロールする時間を入力とする．また，検索にあたって Google Web APIs か Yahoo! Search APIs を使用するため，それらの使用に必要となる識別子（Google Web APIs の場合はライセンスキー，Yahoo! Search APIs の場合はアプリケーション ID）を入力する必要がある．日本語のページのみを検索対象としたい場合は，適宜チェックボックスにチェックを入れる．

パラメータを入力した後，検索ボタンを押すと，システムがウェブをクロールし，情報を収集する．収集が終了すると，結果が視覚的な形で出力される．検索した結果，発見された施設の住所の一覧が画面右半分に表示され，左半分には施設の場所をマーカで示した地図が表示される．この地図の表示には Google Maps API を使用しており，自由な縮尺の変更やスムーズなスクロールが可能となっている．また，マーカをクリックすると，その施設についてウェブをクロールして収集した情報が吹き出しの中に表示される．

さらに本システムでは，複数の検索結果を同時に表示することも可能である．例えば，地名を入力せず全国を検索対象とし，「水田 コシヒカリ」「水田 ササニシキ」「水田 あきたこまち」という 3 つのクエリを用いて計 3 回検索を行うと，それぞれの分布状況がマーカで示された地図が表示される．そして更にある一つのマーカをクリックすると，図 1 のように吹き出しに情報が表示される．

3.2 システムの特長

本システムの特長として，以下の 4 点が挙げられる．

- **ロボット検索による情報収集**

既存の施設検索サイトでは，ほぼ全てあらかじめ登録されている情報を対象として検索を行う．そのため，得られる情報もあらかじめ登録されているものに限られる．また，ここで得られる情報は，その種類が定められているものが多く，例えば住所や営業時間といった型にはまった項目に関する情報しか得ることができない．

その点本システムでは，施設に関する情報を自力でウェブをクロールして得ているので，収集時間は必要とするものの，得られる情報の種類が制限されないという利点がある．

例外として，例えば飲食店専門の検索サイトでは，CGI などを用いてユーザからの口コミ情報を収集し，掲載しているものがある．この場合，その施設検索サイトで得られる情報はユーザからの登録が多ければ多いほど増えることになる．しかし本システムでは，そういった施設検索サイトの口コミ情報に加え，個人が作成しているウェブページに掲載されている施設の評判といった情報も得ることができ，より多くの情報を得られる可能性がある．

- **複数施設の検索可能性**

既存の施設検索サイトの中には，例えば地域やその施設で取り扱っている商品の価格，それに営業時間などによって条件を絞り込んで行き，最終的に一つの施設を見付け出すという操作を行わせるものがある．また，途中の段階までは複数の施設を一度に表示することが出来ても，施設ごとに作られた詳細な情報の掲載されたページに行き着くには，やはり最終的に一つの施設を選択する必要がある．また，例えば施設の場所を示した地図などは，最終的にたどり着く施設ごとのページにしか配置されていないことが多く，地図上に複数の施設を表示して比較検討することはできない．

本システムでは，複数の施設の詳細な情報を一度に表示できないという点では従来の施設検索サイトと変わりないが，複数の施設を同一地図上にマーカとして表示することができる．これにより，より直感的に施設の比較検討をすることができる．また，地図上のマーカを次々とクリックしていくことで各施設

図 1　試作したシステムのスクリーンショット
Fig.1　A screen shot of our system.

の詳細情報を得られるので，より高いユーザビリティを提供できると考える．

- **施設の分布の可視化**

既存の検索サイトは，地図上に表示する施設の数に制限があるものが多い．施設ごとの詳細情報を示したページにしか地図がないようなサイトでは，1 枚の地図上には 1 つの施設しか表示しない．また，1 枚の地図上に同時に複数の施設を表示するサイトでも，その数に制限があり，検索結果に含まれる施設を全て表示できない場合が多い．よって，施設の分布を把握することが困難である．

本システムでは，検索結果に含まれる施設全てを，1 枚の地図上に同時に表示できる．このため，検索結果に含まれる施設

の分布状況を地図上で可視化できる．

また，本システムで複数回の検索を行った場合，本システムは複数の検索結果のそれぞれに含まれる施設を全て同時に 1 枚の地図上に表示できる．このとき，各検索結果ごとに異なるマーカで施設を表すため，どの検索結果に含まれる施設がどこにあるのかを判別できる．この機能により，ユーザは様々な種類の施設について，その分布の違いを 1 枚の地図上で比較できる．例えば図 1 では,「水田 コシヒカリ」のキーワードで検索した結果が新潟県に集中しているなど，米の品種による分布の違いを知ることができる．

- **低コストでの実現可能性**

本システムは，1 つの Windows アプリケーションと，1 台のウェブサーバからなる．アプリケーションは，利用者からの施設検索要求があるたびにロボットを用いてウェブをクロールし，情報を収集する．そのため，検索要求ごとに収集時間を要してしまうという代償は払う必要があるものの，登録式の施設検索サイトのように巨大なデータベースを保持および維持する必要がなく，小規模な構成のマシンでも十分に利用が可能である．またウェブサーバが必要な理由は，Google Maps API のライセンスの「インターネットからアクセス可能なページにしか同 API を使用してはならない」という規約によるものであり，ウェブサーバは Google Maps API を利用した検索結果の地図表示に利用するのみである．そのため，特に高性能なサーバを用意する必要はない．以上の理由により，本システムは低コストな環境で実現可能となっている．

4. アルゴリズム

4.1 システムのアルゴリズム的構造

ユーザが入力したクエリと収集時間に対して，試作したシステムは，まず指定された収集時間の間，トピック主導型クローリングを行う．次に集めたウェブページを解析し，ウェブページ中に出現する住所表記の認識と，認識した住所を説明する記述の認識を行う．最後に認識した住所の緯度・経度を求めて，その緯度・経度に対して認識した説明記述を対応づけた地図を Google Maps API を用いて作成する．ただし，入力された地名を含まない住所は検索結果に含めない．また，抽出した住所の説明記述のどこにもキーワードが現れない場合，その住所は検索結果に含めない．以上の動作をウェブページの収集と解析に関してはマルチスレッド化して並行に行う．同一サイトへのアクセスは 2 秒以上の間隔を空ける．なお 5 節の評価実験時に用いたスレッド数は 32 である．以降では本システムが用いている制限時間付きトピック主導型クローリングアルゴリズム，住所表記の認識アルゴリズム，住所の説明記述の抽出アルゴリズム，それに住所の緯度・経度への変換アルゴリズムについて説明する．

4.2 制限時間付きトピック主導型クローリングアルゴリズム

クローリングについては我々が以前に開発したアルゴリズム [7], [8] を用いた．このアルゴリズムは与えられた収集時間の間，与えられたクエリに適合するウェブページの収集を行う．以下ではこのアルゴリズムの特徴についてのみ述べる．詳細については文献 [7] を参照されたい．

我々のアルゴリズムはクローリングを開始する種ページの集合を Google Web APIs [9] を用いて取得する．そしてこれらの種ページに加えて，種ページからリンクされているウェブページをクロールする．提案アルゴリズムの新規性は以下の点にある．提案アルゴリズムは，アンカーテキスト以外の領域にキーワードを含まないウェブページのリンクはたどらない．これは，アンカーテキストはそのウェブページの内容を表すのでなく，リンク先のウェブページの内容を表すからである．このためアンカーテキスト以外の領域にキーワードを含まないウェブページはクエリに適合しない傾向がある．この傾向はニュースサイトやブログサイトで特に強い．もしアンカーテキスト以外の領域にキーワードを含まないウェブページのハイパーリンクもたどると，1 ページにそのようなリンクが 2 つ以上ある場合，クロールされる不適合ページの数は指数関数的に増大する．このためインターネット接続の貴重なバンド幅を不適合ページの収集で使い尽くしてしまう．

4.3 住所表記の認識アルゴリズム

本システムでのウェブページ中の住所表記の認識処理は，全国の全ての住所をキーワードとするウェブページ中の文字列マッチングとして実現している．

全国の住所の文字列データは，国土地理院が無料で提供しているデータファイル [10] から抽出した．このデータファイルには地番までの細かさ（X 丁目 Y 番 Z 号なら X 丁目 Y 番まで）で全国の住所が記録されている．本システムは 30 秒程度の実行時間でも実用的な量の検索結果を出せることを目標としている．この目標を実現するためには，全国の住所の文字列データは全て主記憶上（32 ビット CPU の場合で高々2 GB 程度）に収める必要がある．地番までの細かさの住所文字列の総数は 1400 万強，サイズの総和は 336 MB となり，一般的なコモディティ PC の主記憶容量が 512 MB であること，住所文字列を管理するデータ構造が消費するメモリ量（住所 1 つあたりポインタ 1 つだけを消費するとしてもポインタだけで約 56 MB 必要）を考えると，ナイーブな実装では住所文字列は主記憶上に収まらない．このため住所文字列から「丁目」以降を削除した文字列をキーワードとして文字列マッチングを行うこととした.「丁目」以降の文字列の認識については，“1-3-5”，“１丁目 3-5”，“一丁目三番五号”，“１ノ３ノ５”，“１－３－５”などの文字列を構文解析により認識する．この場合，住所文字列数は 12 万弱，サイズの総和は 3 MB 弱となる．

キーワード数 12 万の文字列マッチングを単純に BM 法 [11] などの 1 つのキーワードの文字列マッチングアルゴリズムを用いて実現すると，キーワード数に比例する時間がかかり，本システムの目標は到底達成できない．そこで多数のキーワードを線形時間で同時に検索する文字列マッチングアルゴリズムとして知られている AC 法 [11] を用いた．AC 法は複数のキーワード群を前処理して状態遷移表を構築し，その状態遷移表を用いて複数のキーワードの文字列マッチングを同時に行うアルゴリズムである．状態遷移表はキーワード群のみに依存し，検索対象のテキストには依存しないので，キーワード群が固定であれ

ば状態遷移表の構築は最初の 1 回だけでよい．AC 法の計算量は以下の通りである．キーワード群の文字数の総和を m，検索される文字列 T の長さを n，T 中にキーワードが出現する回数を k とすると，

- 前処理（状態遷移表の構築）の時間計算量： $O(m)$
- 検索自体の時間計算量： $O(n+k)$

となる．今回の住所検索のように，前処理は 1 回だけでよい場合は，$O(n+k)$ 時間で検索できることになる．$k \leqq n$ なので，総文字数 n の 1 ページ中の住所文字列を全て認識するのにかかる時間は $O(n)$ となる．前処理を 1 回だけすることによってキーワード数に関わらず，1 キーワードの場合と同じ線形時間のオーダで複数キーワードの文字列マッチングが行えることに注意されたい．12 万件の住所文字列に対する前処理にかかる時間は Intel Pentium 4 - 2.8 GHz の場合で 2.7 秒程度である．

4.4 住所の説明記述の抽出アルゴリズム

住所表記を含むウェブページは，ほとんどの場合その住所に関する何らかの説明記述（施設の住所であれば施設名やその解説など）も含んでいる．住所とその説明記述は表やリストの形で書かれることもあれば 1 つの文章中に現れる場合もある．しかし普通に書かれたウェブページであれば，どのような形式で書かれる場合でも，住所とその説明記述はウェブページ中のどこかに固まっていっしょに記述されるはずである．HTML はタグを用いて構造的に記述されるので，この固まりは HTML 構文木の観点では，1 つの部分木に対応する（図 2 参照）．そこで本システムではまず住所表記を含む TEXT ノードを 4.3 節のアルゴリズムを用いて認識し，次に住所表記を 1 つだけ含む TEXT ノードを含む部分木の極大集合を求める（図 2 参照）．そして最後にこの極大集合中の各部分木を用いて次のように説明記述を抽出する．1 つの部分木に含まれる全ての TEXT ノードはその部分木に含まれる唯一の住所表記に対する説明記述と判断する．

なお，相良ら [6] も同様に HTML 構文木から部分木を構築し説明記述を得ているが，本手法はボトムアップに構文木を構築するのに対し，[6] は HTML 構文木全体を TEXT ノードを 1 つしか含まないようにトップダウンに分割していく手法を取っている点で異なっている．しかしながら，両手法によって得られる結果に差異はないものと思われる．

4.5 住所の緯度・経度への変換アルゴリズム

Google Maps API は，住所の緯度・経度への変換（geocoding）機能は提供していない．REST を用いて Google サイトに geocoding を行わせることは Google Maps API のライセンスに違反する．このため我々は，国土地理院が無料で提供している全国の住所と緯度・経度の対応を記録したデータファイル [10] を用いてデータベースを作成し，このデータベースを用いて geocoding を実現している．提供されているデータファイルは地番までの細かさで全国の住所と，その住所の代表点の緯度・経度などを記録したものである．本システムでは地番までの住所をインデックスとして，住所，緯度，経度をレコードとする FoxPro 形式のデータベースを用いている．総レコード数は 1400 万強，ファイルサイズはインデックスファイルとデータファイルを合わせて 1.6 GB 弱，1 回の geocoding にかかる時間は Intel Pentium 4 - 2.8 GHz の場合で 0.13 ミリ秒程度である．

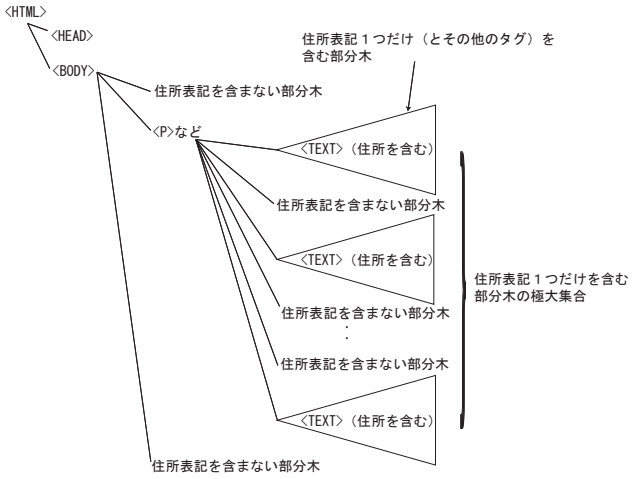

図 2　住所の説明記述の抽出方法

Fig. 2　The idea behind our information extraction algorithm.

5. 評価実験

5.1 実験環境

本システムの実験に用いた環境は以下の通りである．

CPU　Intel Pentium 4 - 2.8 GHz

メモリ　1 GB

OS　Microsoft Windows Server 2003

ネットワーク環境　Gigabit Ethernet

5.2 実験内容

本システムの評価実験で求めたいのは，本システムが収集した住所がクエリに適合しているか，本システムが住所の説明記述と判断した文章が正しく住所を説明するものであるかの 2 点である．そのため，実際に本システムで情報の収集を実行して，実行結果に対してこの 2 点について評価を行った．本システムが出力した各住所について，情報元のウェブページを参照しながら住所とクエリの適合性を評価する．次に，クエリと適合していると判断した住所について，システムがその住所の説明記述と判断した文章が，その住所を説明するものであるかを評価する．なお，これ以降システムが住所の説明記述と判断した文章を「関連文章」と呼ぶことにする．

住所とクエリの適合性についての評価基準は，住所に入力した地名が含まれており，かつその住所が表す施設が入力したキーワードに適合するものならば適合とする．例えば，地名に「大阪」，キーワードに「神社」と入力した場合,「大阪」という文字列を含み，かつ神社の場所を表す住所を適合とする．また，地名を指定せず，キーワードに「南アフリカ ワイン」と入力した場合，南アフリカ産のワインを取り扱う商店，飲食店などを表す住所を適合とする．

関連文章がその住所を説明するものであるかどうかについては，関連文章の適合率と再現率によって評価する．関連文章の適合率とは，関連文章に含まれる，実際に住所を説明する文の

表 1　実験に使用したクエリとシステムの実行時間
Table 1　Queries used in our experiment and running time of our system

クエリ			指定収集時間 30 秒		指定収集時間 180 秒	
No	地名	キーワード	総実行時間 (秒)	収集ページ数	総実行時間 (秒)	収集ページ数
1	大阪	神社	34.01	322	198.92	1566
2		竪穴式住居	33.90	365	200.58	1763
3	大阪	救急病院	47.57	277	202.88	1467
4	神奈川	野球場	55.88	323	198.23	1598
5	東京	茶道教室	33.98	234	197.95	1218
6		南アフリカワイン	34.12	297	198.06	1627
7		酒造	38.71	365	197.95	2258
8		五重塔	33.89	356	197.84	1854

表 2　住所とクエリの適合性についての評価結果
Table 2　Precisions of retrieved addresses

クエリ	指定収集時間 30 秒			指定収集時間 180 秒		
No	収集数	適合数	適合率	収集数	適合数	適合率
1	68	59	86.76 %	198	148	74.75 %
2	9	7	77.78 %	23	13	56.52 %
3	13	10	76.92 %	58	35	60.34 %
4	33	17	51.52 %	92	46	50.00 %
5	26	23	88.46 %	65	46	70.77 %
6	37	36	97.30 %	197	187	94.92 %
7	27	21	77.78 %	310	176	56.77 %
8	20	9	45.00 %	100	22	22.00 %

表 3　関連文章の適合率と再現率についての評価結果
Table 3　Precisions and recalls of retrieved information on relevant addresses

クエリ	指定収集時間 30 秒			指定収集時間 180 秒		
No	適合数	適合率	再現率	適合数	適合率	再現率
1	59	96.42 %	72.25 %	148	94.83 %	96.40 %
2	7	100.00 %	100.00 %	13	88.04 %	98.90 %
3	10	72.51 %	100.00 %	35	69.48 %	100.00 %
4	17	88.39 %	78.23 %	46	68.67 %	88.49 %
5	23	90.11 %	92.58 %	46	76.84 %	94.07 %
6	36	97.40 %	93.07 %	187	96.85 %	97.20 %
7	21	97.34 %	96.59 %	176	95.06 %	88.19 %
8	9	100.00 %	97.38 %	22	94.92 %	97.97 %

割合である．また，関連文章の再現率とは，情報元ページに含まれる住所の説明記述の全ての文に対する，関連文章に含まれる，実際に住所を説明する文の割合である．システムが出力した住所のうち，クエリに適合する各住所に対して，住所の関連文章の適合率及び再現率を求め，その平均値を算出することで評価する．

5.3　実 験 結 果

評価実験で用いたクエリは 8 件である．また，ウェブページの収集時間は各クエリに対して 30 秒と 180 秒の 2 通りで実験を行った．その実験結果を表 1，2，3 に示す．

表 1 は，評価実験で用いたクエリと，システムの検索処理結果を示している．総実行時間はウェブページの収集も含めた全ての検索処理にかかった処理時間を，収集ページ数は検索処理においてシステムが収集したウェブページの総数を，それぞれ表す．なお，表 2 及び表 3 におけるクエリ No は，表 1 のクエリ No と対応している．

表 2 は，住所とクエリの適合性を評価した結果を示している．収集数はシステムが出力した住所の数を，適合数はそのうちクエリに適合した住所の数を，適合率は収集数に対する適合数の割合を，それぞれ表す．実験の結果，ウェブページの収集時間が 30 秒の場合には，クエリに適合する住所の収集数は平均して 22.8 となり，適合率の平均は 75.19 %となった．また，ウェブページの収集時間が 180 秒の場合には，クエリに適合する住所の収集数は平均して 84.1 となり，適合率の平均は 60.75 %となった．このことから，ウェブページの収集時間を長くすると，適合率は一般的に減少するものの，収集する適合住所の絶対数は増加することが分かった．

表 3 は，関連文章の適合率と再現率を評価した結果を示している．適合数はシステムが出力した住所のうちクエリに適合した住所の数を，適合率及び再現率は各住所について算出した関連文章の適合率及び再現率の平均値を，それぞれ表す．実験の結果，関連文章の適合率及び再現率の平均値は，ウェブページの収集時間を 30 秒とした場合にはそれぞれ 92.77 %，91.26 %となり，ウェブページの収集時間を 180 秒とした場合にはそれぞれ 85.53 %，95.15 %となった．これらはいずれも高い数値と言える．

5.4　考　　察

評価実験を通じて，システムが出力した住所及び関連文章と，データの情報元となるウェブページを照らし合わせる作業を繰り返し行った．その作業を元にして，住所のクエリに対する適合性及び関連文章の適合率と，データの情報元となるウェブページの特徴との関係について考察する．なお，以下の文章では評価実験に用いたクエリを，表 1，2，3 にて記述した No を用いて参照する．例えば，「地名指定なし，キーワード “竪穴式住居”」というクエリを参照する場合，クエリ 2 と記述する．

5.4.1　住所を 1 件しか含まないウェブページ

本システムが認識できる住所を 1 件しか含まないウェブページを情報元とした場合，本システムはそのウェブページの全体をその住所の関連文章と判断する．このようなウェブページは，含まれる住所の説明記述がほぼページ全体を占める場合が多く，そしてその場合は住所がクエリに適合する上，関連文章の適合率も高くなる．

しかし，ページ全体としてはクエリに関する情報を記載しているが，含まれる住所はそれらの情報とは無関係というウェブページも存在する．例えば，クエリ 8 を与えた実行においての，観光会社が作成し自社の住所を記載している五重塔紹介ページなどである．このようなウェブページを情報元とした場合，取得した住所はクエリに適合しない．

その他，複数の施設などに関する情報を記載しているが，住所は 1 件しか含まれないようなウェブページも存在する．例えば，クエリ 4 を与えた実行においての，神奈川に位置する野球場を複数紹介しているが，そのうちの 1 つしか住所を記載していないウェブページや，クエリ 1 を与えた実行においての，大

阪に位置する神社や寺院を複数紹介しているが，そのうち 1 つの寺院のみ住所を記載しているウェブページなどである．このようなウェブページを情報元とした場合，前者のような場合は住所がクエリに適合するが関連文章の適合率が低くなり，後者の場合は住所がクエリに適合しない．いずれにせよ，よいデータが得にくいウェブページとなる．

5.4.2 複数の住所を含むウェブページ

本システムが認識できる住所を複数含むウェブページを情報元とした場合，情報元のウェブページの構造によって結果にある程度の傾向が見られる．

本システムの情報元として都合がいいのは，HTML を構文解析して木構造で見たときに，住所とその関連情報との距離が近くなるようにまとめられているウェブページである．具体的には，1 つの div タグに 1 つの住所とその関連情報を含むような div タグが複数存在するようなウェブページや，table タグによって 1 行，即ち 1 つの tr タグに 1 つの住所とその説明記述をまとめているようなウェブページである．このような構造のウェブページを情報元とした場合，住所と同一の div タグあるいは tr タグ内に含まれる情報のみを関連文章として出力することが多く，その場合は住所がクエリに適合する上，関連文章の適合率も 100 %かそれに近い値となる．ただし，取得した住所を含む div タグあるいは tr タグの兄弟ノードに，認識できる住所を含まない div タグあるいは tr タグが存在する場合，それらのタグに含まれる内容も取得した住所の関連文章に加えられる可能性があり，その場合は関連文章の適合率が低くなる．また，クエリに適合する文書を含まないが認識できる住所を含む div タグあるいは tr タグの兄弟ノードに，クエリで指定したキーワードを含むが認識できる住所を含まない div タグあるいは tr タグが存在した場合，後者の div あるいは tr タグが含む情報を関連文章として，前者の div あるいは tr タグが含む住所を取得する場合がある．そのようにして取得された住所はクエリに適合しない．

一方，本システムの情報元として都合が悪いのは，住所とその説明記述とが同一の TEXT ノードに記述されているウェブページである．本システムは認識できる住所を含む TEXT ノードの内容を関連文章に含めることがないため，このようなウェブページを情報元とすると，本システムは住所の説明記述を関連文章に含めることができない．このような例として，クエリ 4 を与えた実行において情報元となった，複数の野球場を紹介しているウェブページが挙げられる．このウェブページではある野球場の説明記述全てを，その住所と同一の TEXT ノードに記述している．その周辺にキーワード「野球場」を含むが住所が記述されていない TEXT ノードがあるため，その TEXT ノードの内容を関連文章としてその野球場の住所が出力されたが，関連文章の適合率は 0.00 %となった．

他に，タイトル部分など，ページ内に記述された住所を含むノード全ての先祖となるノードに，クエリで指定したキーワードが記述されているウェブページも都合が悪い．このようなノードに記述された内容は，そのノードに最も近い住所の関連文章とみなされるため，それ以外のノードの関連文章とはならない．例えば，キーワードがタイトルにのみ含まれていて，クエリに適合するような住所を複数含んでいるようなウェブページを情報元とした場合，実際に本システムが出力する住所は，タイトルに最も近い位置に記述された住所のみである．また，クエリ 7 を与えた実行において情報元となった，タイトルにキーワード「酒造」が含まれているが，タイトルに一番近い住所が酒造会社ではなく，その酒造会社の酒が飲める店の住所であったウェブページのように，タイトルにキーワードがあったためにクエリに適合しない住所を出力してしまう場合もある．

6. まとめと今後の課題

Google Maps API を応用したロボット型施設検索システムを試作した．本システムは，既存のウェブ上の施設検索サイトと異なり，施設の情報をデータベースではなくロボットにウェブをクロールさせることで取得する．そのため，既存のデータベースに登録されていない施設に関する情報も取得でき，また低コストな環境での利用が可能である．

本システムが出力する住所のクエリに対する適合性と，本システムが出力する住所の関連文章が正しくその住所を説明している割合の，2 種類の評価実験を行った．その結果，ウェブページの収集時間を 180 秒とした場合，住所のクエリに対する適合率は平均して 60.75 %となり，住所の関連文章の適合率及び再現率は平均してそれぞれ 85.53 %，95.15 %となった．また，ウェブページの収集時間を長くすると，住所のクエリに対する適合率は減少するものの，収集する適合住所の絶対数は増加することが分かった．さらに，評価実験を通じて，住所の情報元となるウェブページの構造によって，両適合率にある程度のパターンが現れることが見て取れた．

今後の課題としては，現在のアルゴリズムでは適合率・再現率が低くなってしまうクエリに対して，高い適合率・再現率が得られるようにアルゴリズムを改善することが挙げられる．また，できるだけウェブページの構造に依存せずに高い適合率・再現率を得られるようなアルゴリズムの改善も行っていきたい．

文　献

[1] グルメ情報検索サイト ぐるなび，http://www.gnavi.co.jp/
[2] Google ローカル，http://maps.google.co.jp/
[3] 楽天トラベル，http://travel.rakuten.co.jp/
[4] i タウンページ，http://itp.ne.jp/
[5] Mapion，http://www.mapion.co.jp/
[6] 相良毅，有川正俊，“ジオパースによる Web からの空間コンテンツ獲得”，第 15 回データ工学ワークショップ（DEWS2004），I-11-01, 2004
[7] 藤本典幸，萩原兼一，“ウェブマルチメディア検索のためのパーソナルシステム”，夏のデータベースワークショップ（DBWS2005），電子情報通信学会技術研究報告，DE2005-117，pp. 61–66，2005
[8] N. Fujimoto and K. Hagihara, “A Personal System for Web Image Retrieval”, Preceedings of the 4th Winter International Symposiumon Information and Communication Technologies (WISICT 2005), pp. 209–216, 2005
[9] Google Web APIs (beta), http://www.google.com/apis/
[10] 国土地理院，街区レベル位置参照情報ダウンロードサービス，http://nlftp.mlit.go.jp/isj/
[11] D. Gusfield, “Algorithms on Strings, Trees, and Sequences : Computer Science and Computational Biology”, Cambridge University Press, 1997