「マルチメディア，分散，協調とモバイル
（DICOMO 2006）シンポジウム」平成 18 年 7 月

# Geocrawler:位置情報をもとにした個人サイト向け Web インデクサの開発

川口 誠敬　　新井 イスマイル　　藤川 和利　　砂原 秀樹
奈良先端科学技術大学院大学情報科学研究科

# Web Indexer based on Geographical Information for personal web sites

Yoshihiro Kawaguchi　Ismail Arai　Kazutoshi Fujikawa　Hideki Sunahara
Graduate School of Information Science, Nara Institute of Science and Technology

## 1.　はじめに

現在、グルメ情報を検索する場合に利用されているのがぐるなび、グルメぴあ、グルメウォーカー、などの既存グルメ検索サイトである。これらのサイトは、内装の雰囲気、料理のメニュー、写真、値段、味の評価などの情報を全国規模で収集しており、情報量が豊富である。

しかし、既存グルメ検索サイトは、広告収入の点から、特に味の評価に関して、一般受けする情報しか得ることができない。また、店舗検索に関して、広範囲のキーワード検索（店名、場所「奈良市」「生駒市」など）しかできず、地域をまたいだ検索が不可能である。例として、市の境にいるような状況では、両方の市で検索を行わない限りは適切な情報を得ることはできず、余分な検索結果を含んでしまう。つまり多くの場合、キーワード検索では地理的領域中の情報検索が困難である。今後携帯電話のようなモバイル端末が普及するなかで、現在地の周辺といった検索者に距離的に近い場所の店舗情報を距離に応じて検索が可能な位置にもとづいた検索（位置指向検索）が重要となる。

このような背景から、「損得勘定抜きの味の評価情報」（以後、評価情報）を提供でき、「位置指向検索」が可能なグルメ検索サイトが望まれている。

評価情報を提供するための情報源の一つとして、個人 HP(Home Page)やブログ(Weblog)といった個人サイトが考えられる。個人サイトとは、管理者が個人であり、インターネット上に自分の考えや趣味などを発信することができる個人スペースである。利用者は、掲示板、コメント書き込み、トラックバックを使い利用者間でコミュニケーションを取ることができる。個人サイトは、高い匿名性から、ある事柄、ある対象に対して何の遠慮もない意見を発信することができる。この個人サイトが発信する意見を収集対象にしている関連研究として、ブログを掲示板と同様の情報源と考え、評価表現抽出を利用した評判情報検索機能を持つシステムを開発している研究がある[1]。また位置指向検索の関連研究として、ランダムに HTML(Hyper Text Markup Language)ファイルを収集し、そのファイルから位置情報（郵便番号、住所、駅など）を取得して緯度・経度に変換し、HTML ファイルを地理的な位置に配置して位置指向検索を実現している[2]。

本研究では、既存サイトを取り除き個人サイトのある店舗に対しての評価情報を含む HTML ファイルを収集し、地図上に HTML ファイルへのリンクとバルーンを視覚的に配置、利用者に位置指向検索を可能にした新しいグルメ検索サイトの構築を試みる。

評価情報を提供するための情報源として、個人の利用者が多い個人サイトを収集対象にし[3]、その中でも、味に対しての評価が含まれているページを、評価情報を含むページとして選択的に収集する。個人 HP やブログは、管理者が個人であり、評価対象に何の遠慮もない評価情報を含んでいる可能性が高い。その中でもブログは、ホスティングサービスによる簡単な情報発信手段、時系列での情報確認、RSS(RDF Site Summary)[4]による更新通知、トラックバックによる関連記事との相互リンクなどから登録者も増え続けている(2006 年 3 月約 868 万人)[5]。また Web ページ内の、ある対象に対しての評価情報の書き込みを、製品開発や企業活動に反映しようという試みも見られており、個人サイトは重要な情報源として有効だと考えられる[6]。

また利用者に現在地に基づいた位置指向検索を提供するために、Web ページをその HTML ファイル内に含まれている位置情報(住所、郵便番号、電話番号など)を抽出し、それをもとに地図上に配置する。本研究では、店舗の位置を詳細に判断できる住所を位置情報として抽出する。選択的に収集した個人サイトの HTML ファイルを形態素解析にかけ、「地域」と分類された箇所を住所の可能性のあるものとして抽出する。

個人サイトは既存グルメ検索サイトと違い、管理者のサイト作成能力、価値観によるサイト構造の違いから、住所の抽出が困難である。住所の抽出方法として、パターンマッチによる抽出も考えられるが、全国規模の住所情報とマッチさせる必要があるため、汎用的な抽出法が見込める形態素解析を住所抽出方法とする。

抽出した住所を緯度・経度変換をし、Google Maps[7]上に表示させることで視覚的な位置指向検索を実現する。

以上の議論を踏まえて、本研究では、作成した 4 つのコンポーネントを用いて、「損得勘定抜きの味の評価情報」を提供でき、「位置指向検索」が可能なグルメ検索サイトのシステム構築を目的とする。目的とするグルメ検索サイト実現のために 4 つのコンポーネントを作成した。作成したコンポーネントは、評価情報を含む Web ページを選択的に収集するための HTML ファイル収集コンポーネント、HTML ファイルを形態素解析し位置情報を抽出するための住所抽出コンポーネント、住所を緯度・経度に変換する緯度・経度変換コンポーネント、緯度・経度をもとに Google Maps 上に評価情報を含む Web ページのリンクを表示する情報表示コンポーネントである。

本稿では、4 つのコンポーネントによって成り立つグルメ検索サイトを構築し、主要部分である HTML ファイル抽出コンポーネントと住所抽出コンポーネントの有効性について検証した結果を述べる。

２節では、システム概要として、システムを構成するコンポーネントの機能を説明する。３節では、本システムの主要部分である HTML ファイル抽出コンポーネントと住所抽出コンポーネントの実験を行い、その性能を既存サイトの割合や再現率、適合率を求めることで評価する。４節にまとめと今後の課題を述べる。

## 2.　システム概要

本システムは、個人サイトの HTML ファイルのみを収集し、個人サイト内に含まれる住所を抽出する。抽出した住所

を緯度・経度に変換し Google Maps 上に表示する。これにより視覚的な位置指向検索を実現する。図１に、本システムが実現する位置指向検索システムのモデルを示す。HTML ファイル収集コンポーネント、住所抽出コンポーネント、緯度・経度変換コンポーネント、情報表示コンポーネントの４つから構成されており、以下のフローに従って実現していく。各コンポーネントの説明を行う。

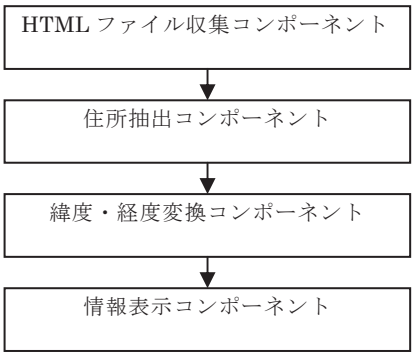


図１．システムモデル

2.1　HTML ファイル収集コンポーネント

このコンポーネントは、膨大な Web ページから評価情報を含む個人サイトの HTML ファイルを収集する機能を持つ。検索から得られる既存グルメ検索サイト(以後、既存サイト)と個人サイトの URL の中から、既存サイトの URL を除去し、個人サイトの URL をもとに HTML ファイルを収集する。

本研究では膨大な Web ページを収集するために、膨大な検索結果に無料で自由にアクセスでき、かつ検索エンジンとして世界中から信頼されている Google[8]の検索エンジンを利用する。このコンポーネントは Google が提供する Web API(Google Web APIs[9])を介して Google の検索技術を利用する。Google サーバに検索キーワードにオプション付きのクエリを送ることで、クエリに対するレスポンスとして、評価情報を含むと予想される URL のリストが返される。検索キーワードに－(マイナス)オプション付きのクエリを送ることで、オプション付きのキーワードを含む HTML ファイルは検索結果から除外される。マイナスオプションだけでは全ての既存サイトの URL を除去することは難しい。除去方法は、URL のドメイン名、既存サイト URL の上に階層を辿ることで検索機能、地理的に広範囲な情報提供など、3 節で定義した既存サイトを特徴づける項目に該当する URL を調べる。既存サイトと判断された URL を除き、選択的に個人サイトの URL を収集する。得られた URL のリストから、HTTP などを用いて Web からファイルをダウンロードしてくるツールである wget[10]を使い HTML ファイルを収集する。

2.2　住所抽出コンポーネント

このコンポーネントは、HTML 収集コンポーネントにて得られた HTML ファイルから住所情報を抽出する機能を持つ。住所情報を抽出する機能を実現するために、形態素解析ツールの一つである茶筌[11]を利用する。形態素解析とは、自然言語で書かれた文を形態素（言語で意味を持つ最小単位）の列に分割し、品詞を見分ける作業である。解析を行う前に HTML ファイル内のタグを除去するためにテキストファイルへ変換する。HTML ファイル内のタグを用いた住所抽出方法（<h1>奈良県奈良市三碓２－１－１０</h1>など）も考えられるが、今回は用いない。なぜなら、個人サイト（ブログ、個人 HP）の HTML ファイルを住所抽出対象にしているためである。個人サイトの HTML ファイルは、既存グルメサイトの HTML ファイルと違い、HTML ファイル作成者のプログラミングスキル、価値観によって HTML ファイルの構造が様々であり、タグの使用方法、使用箇所も様々だからである。HTML からテキストに変換したファイルに対し、形態素解析した結果の例を以下に示す。

例 1)形態素解析システム茶筌を使った解析例（一部抜粋）

| | |
|---|---|
| 全国 | 名詞-一般 |
| 店 | 名詞-接尾-一般 |
| ） | 名詞-サ変接続 |
| 八 | 名詞-数 |
| ちゃん | 名詞-接尾-人名 |
| ラーメン | 名詞-一般 |
| 住所 | 名詞-一般 |
| 福岡 | 名詞-固有名詞-地域-一般 |
| 市 | 名詞-接尾-地域 |
| 中央 | 名詞-固有名詞-地域-一般 |
| 区 | 名詞-接尾-地域 |
| 白金 | 名詞-固有名詞-地域-一般 |
| 1-1-27 | 名詞-サ変接続 |
| うどん | 名詞-一般 |
| そば | 名詞-一般 |
| TEL | 名詞-サ変接続 |
| 092-521-1834 | 名詞-サ変接続 |

本研究では、形態素解析により「地域」に分類された形態素を住所を表す単語とする。まず予備実験を行った。HTML ファイル収集コンポーネントで収集した 175 ページの HTML ファイルをテキストファイルに変換し、形態素解析を行った結果を目視により確認した。その結果、「地域」に分類される形態素が 4 つ以上連続して並んでいる箇所は、住所である可能性が高いことを確認した。そこで「地域」に分類される単語が 4 つ連続した場合は、連続して出てくる 4 つの形態素（「地域」を含んでいる）とそれより後に出てくる、4 つの形態素を連結したものを住所とする。得られた住所を次節で述べる緯度・経度変換コンポーネントに利用することにより、Google Maps 上に Web ページを位置情報に基づいて表示する。

2.3　緯度・経度変換コンポーネント

このコンポーネントは抽出した住所情報を緯度・経度情報に変換する機能を持つ。この機能を実現するために Yahoo! Maps[12]を用いた緯度・経度変換を行う。Yahoo! Maps に住所を入力し検索を行うと、ブラウザの URL 内に入力した住所に対しての緯度・経度が含まれていることがわかった。表示ページの HTML ファイルを wget コマンドで収集し、HTML ファイル内から正規表現を用いて緯度・経度を抽出する。入力した住所が Yahoo! Maps の住所データベースとマッ

チングする場合は緯度・経度変換作業を行い、正しい緯度・経度を含む HTML ファイルを返す。しかし、入力した住所文字列が長すぎたり、短すぎたりする場合は、Yahoo! Maps が自動的に近似(最長一致)の住所に入力を変え、近似の住所に対しての緯度・経度変換が施され緯度・経度が返される。その結果、正しい住所でない住所入力の場合でも、近似の住所に緯度・経度変換されることが起こる。つまり、緯度・経度変換コンポーネントの精度は、住所抽出コンポーネントの住所抽出の精度に依存している。

### 2.4 情報表示コンポーネント

このコンポーネントは、緯度・経度変換コンポーネントで得られた緯度・経度をもとに Google Maps 上にバルーン(マーカー)を表示する機能を持つ。緯度・経度変換コンポーネントで得られた数値（緯度・経度）を Google Maps API[13]に渡し、数値をもとに地図上にバルーンと個人サイトの URL を表示させる。これによって、位置指向検索が可能な個人サイトのグルメ情報提供するグルメ検索サイトが実現する。実際にこのコンポーネントを利用した結果の表示画面を図 2 に示す。各個人サイトに含まれる位置に基づいて地図上にバルーンをプロットしている。バルーンをクリックすると、表示されている位置の店舗について記述のある個人サイトへのリンクが表示される。

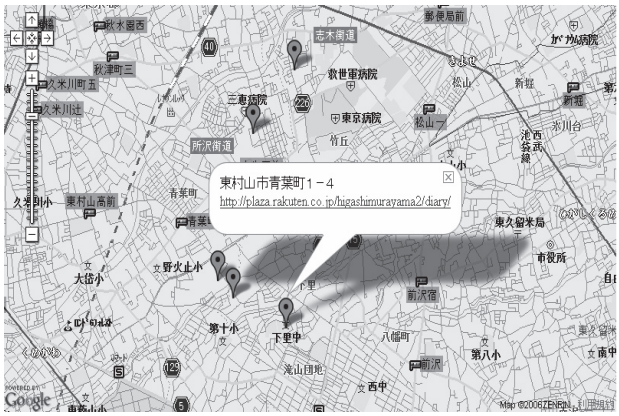


図 2. 情報表示の様子

## 3. システムの実験と結果

本システムは、2 節で示したシステムモデルに基づき、4 つのコンポーネントが実装されている。実験目的は、「HTML ファイル収集コンポーネントの個人サイト HTML ファイルを収集する精度」と「住所抽出コンポーネントの正しい住所を抽出しているかを評価する再現率と適合率」を確認することである。実験を行い、2 つのコンポーネントの精度を求め検討する。実験環境は、CPU(Pentium 4 3.06GHz)、メモリ(1GB)、OS(Windows XP SP2)、言語(Perl5.8.2,JavaScript)、Web ブラウザ(Internet Explorer 6)である。

### 3.1 HTML ファイル収集コンポーネントの実験と結果

HTML ファイル収集コンポーネントの目的は、検索から得られる既存サイトと個人サイトの URL リストの中から、既存サイトの URL を除去し、個人サイトの URL をもとに HTML ファイルを収集することである。実験では、特定の検索語(Google に検索語「ラーメン　住所」を渡す)に対して、検索オプションをつけない場合と検索オプションをつける(HTML ファイル収集コンポーネント)場合の検索を行う。検索語に含まれる「住所」に関しては、位置指向検索を可能にするために必要な要素として考えられる。検索で得られた URL から HTML ファイルの内容を目視により確認し、個人サイト、既存グルメ検索サイトに分類する。

本システムでは、個人サイト（ブログ、個人 HP）と既存グルメ検索サイト（既存サイト）に関して以下のように定義した。

ブログ：
・RSS フィードを持つもの
・トラックバック機能を持つもの
・アーカイブによる過去ログ参照機能を持つもの
・時系列に日記が参照可能であるもの

個人 HP：
・ブログ、既存サイトの定義に該当しないもの
・店舗に対する評価情報を持つもの(★の数、数値の大小(4.5 点、8.0 点)、言葉の強弱(普通、旨い、激旨)など、他店舗との違いを明確に表記していること)

既存サイト：
・店舗検索機能を持つもの
・全国規模の情報提供範囲を持つもの
・会員登録制機能を持つもの
・サイト管理者が旅行会社、地域情報局、テレビ会社、新聞社であるもの
・グルメ(ラーメン)以外の情報を提供しているもの(生活情報、コスメ、天気、交通など)
・上記のいずれかの内容を階層構造を辿ることにより確認できるもの

その他：
・ネット通販、PDF、RSS 情報、リンク切れ

表 2．集計結果

| | 既存サイト | 個人サイト | | その他 | 既存サイトの割合 |
|---|---|---|---|---|---|
| | | ブログ | 個人 HP | | |
| 検索オプションなし | 323 | 35 | 29 | 13 | 0.81 |
| 検索オプション付き | 188 | 151 | 24 | 51 | 0.47 |

表 2 に上記の定義に従って分類を行った結果を示す。

検索語：「ラーメン　住所」で検索を行った場合、検索結果上位 400 件中 64 件が個人サイト（ブログ 35 件、個人 HP29 件）であった。また既存サイトが 323 件であった。この結果は、既存サイトを除去し個人サイトの HTML ファイルを収集するという目的にそぐわない。本来求める個人サイトの HTML ファイルを収集するために、検索結果の URL に出現頻度が高かった既存サイトに対して、以下のようなキーワードにマイナスを付けることで検索結果から除去する。

検索語：「ラーメン　住所 －ぐるナビ －Yahoo!グルメ －グルメウォーカー －all about －MSN グルメ －livedoor グルメ －ラーメンバンク －タウン －NAVITIME」の場合、上位検索結果 400 件中 175 件が個人サイト（ブログ 151 件、個人 HP24 件）であり、188 件が店舗情報、ラーメン総合案内サイトという集計結果が得られた。この集計結果を用いて

HTML ファイル収集コンポーネントを使用した場合の既存サイトの割合を求め実験結果とする。表 2 からも読み取れるように、HTML ファイル収集コンポーネントを用いることで、HTML ファイル 400 件に対しての既存サイトの割合を 0.81 から 0.47 に減少させることに成功した。

3.2　住所抽出コンポーネントの実験と結果

住所抽出コンポーネントの目的は、正規化した住所を抽出することである。実験は、まず HTML ファイル収集コンポーネントで収集した個人サイトの HTML ファイル 175 件を目視で住所確認し、正しい住所数を調査する。正しい住所の定義は、収集した個人サイトの HTML ファイル内に記述された住所が番地以降(「生駒市高山町 8916-5」、「三碓 1-5-10」など)まで記述された住所である。正しい住所の定義に従って、収集した個人サイトの HTML ファイルを目視した結果、720 個の正しい住所を確認できた。また住所抽出コンポーネントを用いて HTML ファイルから住所を抽出した結果、729 個の住所抽出に成功した。そのうち正しい住所は 94 個抽出することができた。正しい住所数と住所抽出コンポーネントを用いて HTML ファイルから抽出した住所数を用いて、住所抽出コンポーネントの再現率と適合率(精度)を以下に示す式を用いて求め、このコンポーネントの評価を行う。

A: 目視により確認した正しい住所数
B: 住所抽出コンポーネントで抽出した住所数

$$\text{再現率} = \frac{A \cap B}{A} \quad \text{.....(1)} \qquad \text{適合率} = \frac{A \cap B}{B} \quad \text{.....(2)}$$

(1),(2)式を用いて、再現率と適合率を求めた結果、住所抽出コンポーネントの再現率は 13.0%、適合率は 12.9%になった。この結果より、個人サイトの住所記述の方法が様々であり、このコンポーネントで利用したアルゴリズムでは個人サイトの住所抽出は困難であることがわかった。今後の研究では、住所を抽出するアルゴリズムを改良して高い再現率と適合率を得ることを目指す。

## 4.　まとめと今後の課題

本研究では、評価情報を含む HTML ファイルを選択的に収集し、検索者の現在地にもとづき、地図上に店舗情報を視覚的に配置、位置指向検索を可能にした新しいグルメ検索サイトの構築を試みた。本稿では、システムモデルの主要部分である HTML ファイル収集コンポーネントと住所抽出コンポーネントの有効性について検証した。

HTML ファイル収集コンポーネントについては、このコンポーネントを使用する場合、使用しない場合の既存サイトを収集した割合を求め比較評価した。その結果、既存サイトが含まれている割合が 0.81 から 0.47 に変化していることがわかった。これはこのコンポーネントを用いることで既存サイトの件数が減少したことを意味し、目的である既存サイトを除き、個人サイトをある程度選択的に収集できた。

住所抽出コンポーネントについては、個人サイトの HTML ファイル 175 件から、このコンポーネントを用いて住所抽出を行い再現率と適合率を求め評価した。HTML ファイル 175 件から目視により得た住所数が 720 個、住所抽出コンポーネントを用いて抽出した住所数が 729 個、そのうち正しい住所数は 94 個であった。これらの住所数を用いて、再現率と適合率を求めた結果、再現率 13.0%、適合率 12.9%という結果になった。実験で用いた個人サイトの HTML ファイルは、管理者が個人であるため住所記述が様々であり、このコンポーネントで使用した住所を抽出するアルゴリズムでは対応しきれていないことが明らかとなった。

住所抽出コンポーネントは、形態素解析で「地域」に分類された単語だけを連結するだけでは正しい住所を抽出することはできない。なぜなら丁目以降を解析したとき、「地域」に分類されない、何丁目を表す数字や漢数字が必ず現れるからである(例:２丁目５－１２、三丁目など)。そこで、「地域」が４つ以上連続して現れた場合、４つ連続して現れた「地域」に分類された単語に加えて、それ以降連続して現れた「地域」に分類される単語も抽出する。これによって丁目より前に記述される住所を抽出することが可能である。丁目以降(丁目、番地、号)に関しては、既存の住所データベースを参考にし、記述方法を分析し正規表現によるパターンマッチによって抽出する方法を検討している。

## 5.　参考文献


[1] 鈴木泰裕,高村大也,奥村学, ”Weblog を対象とした評価表現抽出,” 人工知能学会研究会資料 SIG-SW&-ONT-A401-02.
[2] 横路誠司,高橋克巳,三浦信幸,島健一,”位置指向の情報の収集,構造化および検索手法,” 情報処理学会論文誌 vol.47 No.7,pp_1987-1998, July 2000.
[3] 武田英明, “Weblog 研究の現状,” 人工知能学会研究会資料 SIG-SWO-A402-06 ,July 2004.
[4] RDF Site Summary (RSS) 1.0. http://web.resource.org/rss/1.0/
[5] ブログ及び SNS の登録者数（平成 18 年 3 月末現在）. http://www.soumu.go.jp/s-news/2006/060413_2.html#b2
[6] 松村真宏, “チャンス発見のためのコミュニティマイニングに関する研究,” 平成 14 年度東京大学大学院工学系研究科電子工学専攻博士論文,2003.
[7] Google Maps. http://maps.google.co.jp/
[8] Google. http://www.google.co.jp/.
[9] Google Web APIs. http://www.google.com/apis/.
[10] GNU Wget. http://www.gnu.org/software/wget/.
[11] 茶筅. http://chasen.naist.jp/hiki/ChaSen/.
[12] Yahoo! Maps. http://map.yahoo.co.jp/.
[13] Google Maps API. http://www.google.com/apis/maps/.