第 4 章

ジオワードを用いたウェブローカルサーチの実現

本章では，インターネット上に分散するウェブ文書を位置指向に検索する方法について述べる．

インターネット上に分散するウェブ文書を位置指向に検索するシステムを開発した．本システムでは，任意の地理的領域に属するウェブ文書を検索することが可能である．本検索システムの実現のために，3 つの手法を開発した．まず，位置指向検索に必要なウェブ文書を選択的に収集する手法，次にウェブ文書から住所を抽出し，抽出住所を緯度経度と対応づけることによる構造化手法，そして構造化された文書の緯度経度を用いた，地理的検索手法である．選択的収集手法はウェブ文書の内容を予測し，位置に関連した情報を高い割合で収集することができる．

構造化手法では，住所辞書を持った形態素解析と住所表記の正規化を用いて，ウェブ文書からの住所抽出を行った． その結果，正しい住所の抽出を保証した上で， 出現住所文字列の 92% の抽出を丁目レベルで実現した．地理的検索手法では，構造化で付与された緯度経度情報と検索領域の重なりに存在する WWW 文書の情報を提示する．この手法の評価実験を行った結果，提案手法は，検索領域として住所文字列を使用する従来のキーワード検索で少なくとも約 25% 存在していた検索もれを解消することができた．

4.1 位置指向の情報検索

情報を情報の地理的な位置とそこからの距離に基づいて検索する方法を，位置指向検索と呼ぶ．インターネットに分散するウェブ文書を位置指向に検索することができれば，文書を位置で分類したり，特定の位置に関連した情報を集めることができる．この検索は，ガイドブックやナビゲーション，地域情報案内などのアプリケーションに応用が可能で，

集められたホームページはモバイル向けを始めとする様々なサービスの有力なコンテンツにもなりうる.

現在でも一般のウェブ文書を検索するためのシステムは多数存在するが，これらのシステムの主流はキーワード検索である．キーワード検索でも住所文字列を入力することにより，特定の住所を含む文書を検索することは可能である．しかし，検索条件として与えられた位置からの距離に応じて検索することができず，位置指向の検索は困難である.

例えばこのことは，行政区界付近での検索では顕著になる．図 4.1 において，検索者(図中×で示される) が現在地の周辺の情報を検索したい場合，理想的な検索範囲は円で示されるが，検索条件として A 市を使用すると，B，C 市と円の重なり部分の情報は検索できない一方で，A 市内の円外の余分な情報も検索してしまう.

つまり，多くの場合，キーワード検索では，適切な地理的領域中の情報検索が困難である.

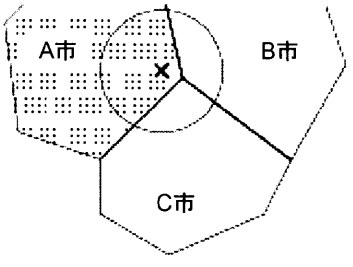

図 4.1 キーワード検索での不適切な位置指向の検索例

距離に応じて検索できないという問題は特にモバイル端末を使用して，ガイドブック，地域情報などの検索を行なうときにあらわれる．このような状況では，距離的な移動は検索者にとり負担となるため，検索者は距離的に近い場所の情報を必要とする．しかし，前述のようにキーワード検索を用いた場合，距離に応じた検索は困難なので，距離に応じた検索が可能な位置指向検索が重要となる.

適切な地理的領域中の情報を検索可能にするには, 検索対象となる情報に記述されている「地理位置」を正確に把握し, さらにその「位置」を幾何図形として表現し, やはり幾何図形で表現された検索領域との重なりを調べることが必要となる.

筆者らはインターネット上のウェブ文書を位置指向に検索するためのシステムを開発した．本システムの構成を図 4.2 に示す．本システムは大きく 3 つのモジュールからなる．本章はシステムの構成に従い，はじめに対象とする位置に関連したウェブ文書を選択的にネットワークから収集する方法とその評価について述べ，次に集めた文書から住所を抽出して，抽出した住所と緯度経度を対応づける位置指向の構造化の手法と評価を述べ，位置

指向の構造化を行った結果可能になった地理的検索と従来の検索との比較結果について説明する．さらに本システムを実装し，「モーバイルインフォサーチ実験」の中で「ここのサーチ」として試験サービスを行った結果を報告する．

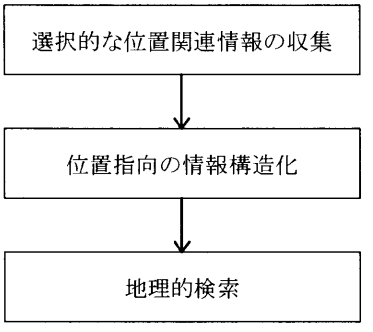

図 4.2 本システムの構成

4.2 位置関連情報の収集

4.2.1 位置関連情報

ウェブ文書の中には，ある場所に関して述べたもの (例：国立公園の紹介) や，ある場所に存在する物について述べたもの (例：レストランの紹介) などがある．これらは地理的な「位置」に関連しており，本論文では位置関連情報と呼ぶことにする．

ある情報が位置に関連しているかどうかの判定は一般に容易ではないが，今回は対象の文書中に「位置情報」が含まれているものを位置関連情報と扱う．位置情報とは，位置を特定できる情報で，緯度経度，住所，最寄り駅，ランドマーク名などがある．

位置指向の検索システムでは位置に関連した情報以外は検索対象とならない．そこで，ウェブ文書中の位置関連情報の割合を調査し，どの程度のウェブ文書が位置指向検索に使用できるか調査した．調査は，通常のウェブクローラと呼ばれる自動的にウェブ文書を収集するアプリケーションが収集した文書から約 100 ページをランダムに選択し，ページ中の住所の有無を目視で確認するという方法で行った．表 4.1 に結果を示す．表 4.1 より全ウェブ文書中の 28% 程度が位置関連情報であることが確認できた．

4.2.2 位置関連情報の選択的収集

4.2.1 節の結果から，通常のウェブクローラの収集した文書は，7 割以上が位置指向の検索システムでは検索対象とならない．そこで，これらの関連のない情報をできるだけ収集

表 4.1 位置情報を含むウェブ文書の割合

位置情報種別	割合
住所	17.1%
ランドマーク	17.9%
駅	3.5%
いずれかを含む	28.1%

せずにクローリングを行うために以下に示す位置関連情報の選択的収集法を考案した.

高い確率で位置関連情報を収集するためには, 既に収集したウェブ文書から参照されているウェブ文書の内容を予測し, 位置関連情報を優先的に収集する手法が必要だと考えられる. 筆者らは, まず図 4.3 の (1) のように, 参照元のリンク文字列に位置情報が含まれている参照先は, 位置関連情報であるという仮説を立てた. この仮説の検証を, ランダムに収集したウェブ文書 20 ページに対して行ったところ, 以下に示す結果となった.

```
<a href="park.html">武蔵野市の公園</a>      ・・・(1)
<a href="agency.html">行政機関一覧</a>      ・・・(2)
```

図 4.3 位置情報を持つリンク文字列を含む HTML ファイルの例

- case1: 参照元のリンク文字列に, 図 4.3 の (1) のように位置情報を含む場合
  参照先文書が位置情報を含んでいた割合は 92.5%.
- case2: 参照元のリンク文字列に, 図 4.3 の (2) のように位置情報を含まないものの, 同一文書中に (1) のような位置情報を含む参照先がある場合
  参照先文書が位置情報を含んでいた割合は 51.3%
- case3: 上記以外の場合 (文書中のリンク文字列が位置情報を持たない)
  参照先文書が位置情報を含んでいた割合は 14.5%

検証の結果から,case1 の場合に収集優先順位を上げ,case3 の場合に収集優先順位を下げて収集すれば, 効率的に位置に関連した情報を収集できると考えられる. 実際の収集手順を以下に示す.

1. 収集の初期 URL から収集を始める
2. 収集したウェブ文書から位置情報を抽出し, 文書に含まれる各参照先 URL の収集優先順位を決定する
3. 参照先 URL を収集優先順位とともにデータベースに格納する
4. 収集優先順位の高い URL をデータベースから検索し, その URL に対応する文書を収集する
5. 一定数のウェブ文書が収集できるまで,(2) から (4) までを繰り返す

4.2.3 位置関連情報の選択的収集の評価

この手法を用いた選択的収集を行うウェブクローラおよび通常のウェブクローラ (幅優先探索法 [セジウィック 1993] を使用) が集めた文書が位置関連情報を含んでいた割合を図 4.4 に示す.

図 4.4 の横軸は, ウェブクローラが収集したウェブ文書数をあらわし, 縦軸は収集したウェブ文書中の位置関連情報の割合を示している. どちらも, 収集を開始する URL は同様のものを使用し, 日本語のウェブ文書を 10 万ページ程度収集した.

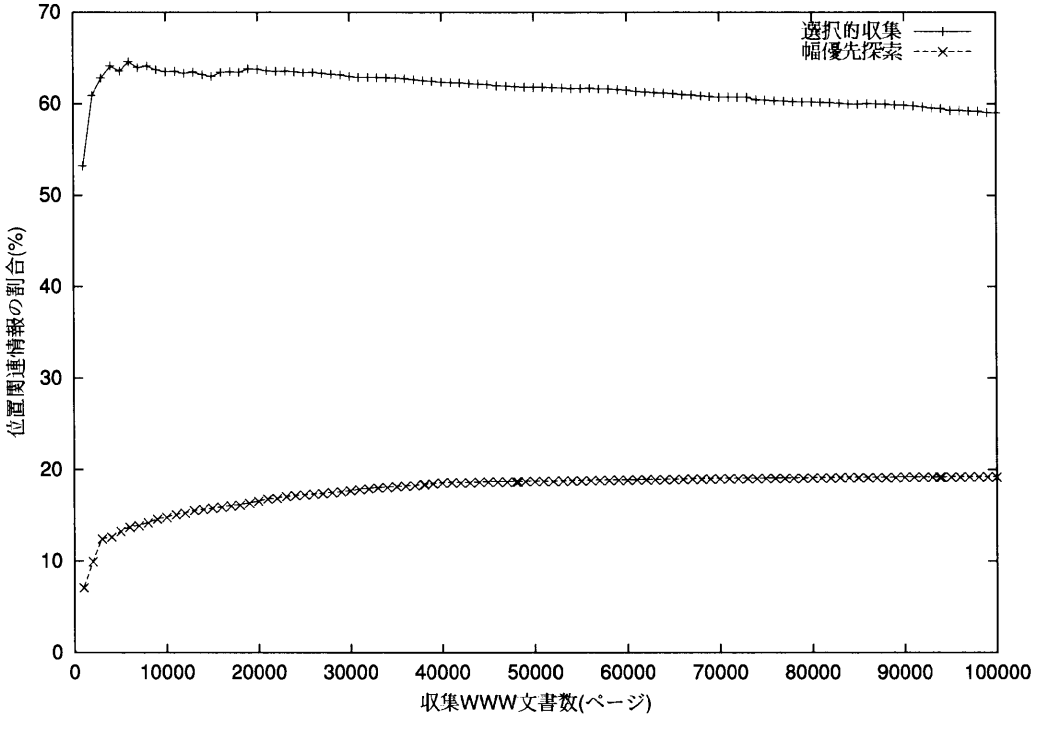

図 4.4 位置関連情報収集率の違い

図 4.4 より, 位置情報選択的収集型のクローラが収集したウェブ文書中には, 約 10 万文書収集時点においては, 通常のウェブクローラと比較して約 3 倍多く, 位置関連情報が含まれていることがわかる.

4.3 位置指向の情報構造化

この章では, 収集したウェブ文書から位置情報を抽出した上で, ウェブ文書と幾何図形情報を対応づける位置指向の構造化手法について説明する. 4.1 章で述べたように, 位置指向の検索を行う上で, ウェブ文書中位置情報の抽出はウェブ文書と幾何図形を対応づけるための前処理として不可欠である.

4.3.1 位置指向の情報構造化手法

位置情報の抽出

抽出可能な位置情報としては住所, 駅名, ランドマーク名 (例:東京タワー), 路線名, 電話番号, 郵便番号などが考えられるが, 今回は, これらのうち, 住所の抽出を行った. 住所の抽出は, 住所名を辞書に加えた形態素解析エンジン「すもも」[鷲坂 1997] で, 形態素解析を行った後に, 各形態素を住所辞書と比較し, 一致したものを住所とした.

形態素解析エンジンへ追加した辞書および比較用の住所辞書の住所数は 946,996 件である. 追加した住所の具体例を示す.

1. 都道府県 (東京都)
2. 市区町村 (東京都武蔵野市, 武蔵野市)
3. 町字 (東京都武蔵野市緑町, 武蔵野市緑町)
4. 丁目 (東京都武蔵野市緑町 3 丁目, 武蔵野市緑町 3 丁目)

位置指向の検索システムでは, 正しく住所が抽出されることが重要となる. そこで, 人名や一般名詞などの意味的に不正確なものを住所として抽出しないために, 以下のルールを用いて住所の判定を行った.

1. 都道府県名から省略なく正確に記述してあるものを住所とする.
   (例:東京都武蔵野市緑町 3 丁目など)
2. 都道府県, 市区は, 住所を明確に示す接尾辞「都道府県, 市区」がついているもののみ単独で住所とする. (例:宮崎県, 市川市, 渋谷区は○, 宮崎, 市川, 渋谷は人名等の可能性があるため×)
3. 町, 村は所属する郡がついているもののみを住所とする.
   (例:比企郡小川町, 邑楽郡千代田町は○, 小川町, 千代田町はさらに詳細な町字の可

能性があるため×)

4. 町字および丁目のような詳細な住所は,(2) または (3) に続いて町字, 丁目が続く場合のみを住所とする.
(例:武蔵野市緑町 3 丁目は○, 緑町 3 丁目は×)

ただし,(1) や (4) でいう, 丁目や番地の表記にはばらつきがあるため, 表記を統一, すなわち正規化を行い, 丁目を検出する必要がある. 正規化を行うためには, 正規化を行う文字列を特定する必要がある. そこで, 正規化の対象となる, 丁目表記に使用される数字および丁目と番地, 号などを接続する区切り文字の実態を調査した. 表 4.2 に結果を示す. 調査対象は 98,987 ページのウェブ文書で, その中に含まれる丁目の数は 20,310 個である. 表 4.2 にはそのうち 1% 以上の割合を占めるもののみを示す. 表 4.2 の種別は表記に用いられた

表 4.2 丁目表記のばらつき

数字	区切り文字	頻度 (%)	例
半角	-	45.64	武蔵野市 3-9-11
全角	-	19.22	武蔵野市 3—9—11
全角	丁目	17.43	武蔵野市 3 丁目 9 番 11 号
半角	丁目	5.46	武蔵野市 3 丁目 9 番 11 号
漢字	空白	4.64	武蔵野市 三 九 十一
半角	-	2.31	武蔵野市 3—9—11
漢字	丁目	1.40	武蔵野市 三丁目九番十一号

数字の種類をあらわしている. 表 4.2 を使用した丁目住所の正規化および抽出ルールを以下に示す.

1. 抽出された住所が町字か否かを調べる.
(例:東京都武蔵野市緑町)
2. 抽出された住所文字列の後に, 数字 (半角, 全角および漢数字) が続くか否かを調べる.
(例:東京都武蔵野市緑町 3-9-11)
3. 表 4.2 に示した区切り文字が, 数字の後に続くか否かを調べる.

4. 丁目について以下の正規化を行う. 数字, 区切り文字等を統一し, 丁目レベルの数字には「丁目」を区切り文字に置き換え加える. (正規化の例:東京都武蔵野市緑町 3-9-11 → 東京都武蔵野市緑町 3 丁目－９－１１)

5. 正規化後の住所が, 住所辞書中にあったら, 丁目として抽出する.

位置指向の構造化

本節では, ウェブ文書から抽出した位置情報を幾何図形 (緯度経度) に変換し,WWW 文書のメタ情報として付与する操作を位置指向の構造化と定義し, その手法について示す. 位置指向の構造化の流れを以下に示す.

1. ウェブ文書から位置情報の抽出を行う.
2. 抽出の結果, 住所と見なされた文字列を位置情報リポジトリを参照して緯度経度多角形 (または重心点) へ変換する. 位置情報リポジトリとは位置情報 (住所文字列) とそれに対応した緯度経度 (多角形または代表点) の組を定義した外部知識である.
3. 住所とそれに対応する緯度経度多角形 (または重心) を住所の組として図 4.5 のような XML を出力する.

構造化前
NTT 情報流通プラットフォーム研究所
東京都武蔵野市緑町 3-9-11

構造化後
NTT 情報流通プラットフォーム研究所
<Address>
<Name>
東京都武蔵野市緑町３丁目９－１１
< /Name>
<Polygon>
$(x_1, y_1), (x_2, y_2), \cdots (x_n, y_n)$
< /Polygon>
</Address>

図 4.5 位置指向の構造化の例

この手法を用いることにより, 検索の条件として, 緯度経度が使用できるようになり, 複数の住所にまたがる場合の検索においても, 理想的な検索範囲の検索ができる.

4.3.2 位置情報抽出の評価

本節では, 4.3.1 節に示した方法の評価について述べる. 評価に使用したウェブ文書は, 住所を含む文書をランダムに 80 ページ選択したものである. 評価はまず, 住所をその規模に応じて, 都道府県, 市区町村, 町字, 丁目の 4 つの階層に分類し, 各階層毎の正解と思われる住所の出現分布を目視によりもとめ, 更にそれぞれの階層毎の平均抽出適合率および抽出再現率をもとめた.

ウェブ文書中の住所の階層毎の出現分布を表 4.3 に示す. 出現した住所は全階層の合計で,1,059 個であった. 表 4.3 の結果より, ウェブ文書に含まれる住所のうち都道府県, 市区町村が全住所の約 7 割を占めることがわかる.

表 4.3 住所階層毎の住所の出現割合

住所種別	出現個数 (A)	出現割合 (% = A/B)
都道府県	405	38.2%
市区町村	345	32.6%
町字	165	15.6%
丁目	144	13.6%
合計 (B)	1,059	100.0%

次に, 住所階層毎の適合率と再現率の定義を 4.1,4.2 式に示す. 適合率は抽出された文字列が実際に住所であった割合, 再現率は文書中に存在した住所文字列が実際に抽出された割合である.

$$P_{(d,h)} \stackrel{\text{def}}{=} \frac{\left|\boldsymbol{Ext}_{(d,h)} \cap \boldsymbol{Rel}_{(d,h)}\right|}{\left|\boldsymbol{Ext}_{(d,h)}\right|}$$
$$\overline{P_h} \stackrel{\text{def}}{=} \frac{\sum_{d=1}^{d=n} P_{(d,h)}}{n}$$
$$(\boldsymbol{Ext}_{(d,h)} \neq \emptyset) \tag{4.1}$$

$$R_{(d,h)} \stackrel{\text{def}}{=} \frac{\left|\boldsymbol{Ext}_{(d,h)} \cap \boldsymbol{Rel}_{(d,h)}\right|}{\left|\boldsymbol{Rel}_{(d,h)}\right|}$$
$$\overline{R_h} \stackrel{\text{def}}{=} \frac{\sum_{d=1}^{d=n} R_{(d,h)}}{n}$$
$$(\boldsymbol{Rel}_{(d,h)} \neq \emptyset) \tag{4.2}$$

4.1,4.2 式において,$P_{(d,h)}$ は文書 d の階層 h における適合率であり,$R_{(d,h)}$ は同じ文書, 階層の再現率である. また,$\boldsymbol{Ext}_{(d,h)}$ は, 文書 d 中から抽出された階層 h の住所の集合であり,$\boldsymbol{Rel}_{(d,h)}$ は文書 d 中の階層 h の正しい住所の集合である. また,$\overline{P_h}$ および $\overline{R_h}$ は, 階層 h における平均適合率および再現率である.

表 4.4 階層毎の住所抽出の平均適合率および再現率

	適合率 (%)	再現率 (%)
都道府県	100.00%(301/301)	74.32%(301/405)
市区町村	100.00%(281/281)	81.45%(281/345)
町字	100.00%(145/145)	87.88%(145/165)
丁目	100.00%(133/133)	92.36%(133/144)

4.3.1 節のルールでは, 適合率を重視したため, 表 4.4 に示すように, 適合率は非常に高い.

次に再現率低下の原因を各ルール毎に示す. ルール (2) に従った抽出では, 接尾辞なし都道府県, 市区町村の抽出もれ, ルール (3) による抽出では, 郡のない町村の抽出もれ, ルール (4) による抽出では, 丁目記述の正規化によるばらつきの吸収の失敗による抽出もれが起こっている. ルール (1) による抽出もれは, 都道府県の欠如とルール (2)〜(4) による抽出もれが混在したものである.

結果として, 適合率, 再現率は平均で 100% および約 84% であった. 今回の位置指向の検索システムでは適合率を重視しているため, 本節の評価の結果, 本抽出手法は目的を達成したと思われる.

4.4 地理的検索

4.4.1 地理的検索

4.3 章までに述べた, 位置関連情報の選択的収集と位置指向の構造化を行うことによって, ウェブ文書は幾何図形情報を持つ構造化された形になった. この幾何図形情報を検索キーに用いることで, ウェブ文書の地理的検索が可能になる. すなわち, 幾何図形として表現された利用者の検索領域 (例えば現在地の周辺は, 現在地を中心とした円) と幾何図形として表現されたウェブ文書との重なりを調べることで検索を行う. このような幾何図形同士の重なりの検索は, R–Tree[Guttman 1984] のような計算幾何学的検索アルゴリズムを用いる.

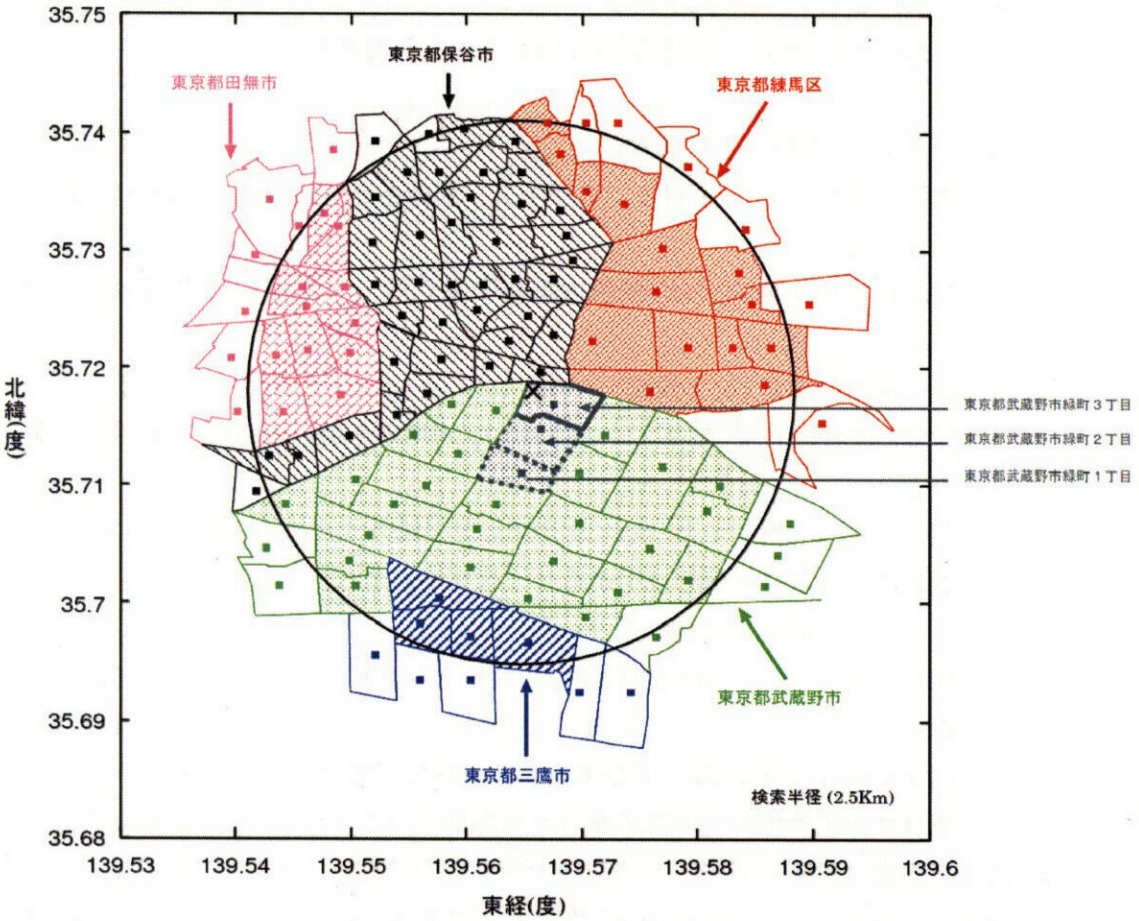

図 4.6 地理的検索とキーワード検索の例

4.4.2 地理的検索の評価

提案する位置指向検索の適切さを評価するために多角形検索による地理的検索 とキーワード検索の比較実験を行った．また参考として代表点検索による地理的検索との比較も行った．

1. 多角形検索
   検索の条件として, 検索者のいる位置と検索半径を用いるもの．データベース中では文書の位置は, 住所に対応する多角形として表現される．
   図 4.6 では, 黒, 桃, 赤, 緑, 青の実線で囲まれた領域全ての文書が検索される．
2. キーワード検索

検索の条件として, 住所文字列を用いるもの.
データベース中では文書の位置は住所文字列で表現される.

(a) キーワード検索 (丁目検索)
検索の条件として, 丁目までの住所文字列相当を用いるもの. (例: '武蔵野市緑町3丁目')
図 4.6 に示す検索では, 灰色太実線で示される 1 領域中の文書のみが検索される.

(b) キーワード検索 (町字検索)
検索の条件として, 町字までの住所文字列相当を用いるもの. (例: '武蔵野市緑町')
図 4.6 に示す検索では, 灰色太実線および灰色太点線の 3 領域中の文書が検索される（武蔵野市緑町は 1 から 3 丁目まである）.

(c) キーワード検索 (市区町村)
検索の条件として, 市区町村までの住所文字列相当を用いるもの. (例: '武蔵野市')
図 4.6 に示す検索では, 緑色実線で描かれた領域全て (実線, 塗りつぶしの両方) 並びに緑色実線の外側に広がる武蔵野市中全ての文書が検索される.

3. 代表点検索（参考）
検索の条件として, 検索者のいる位置と検索半径を用いるもの. データベース中では文書の位置は, 住所に対応する点 (重心点等) として表現する.
図 4.6 に示す検索では, 黒, 桃, 赤, 緑, 青の塗りつぶされた領域中の文書が検索される. 地理的検索として理想的な手法は, 多角形検索だと考えられるが, 現在入手できる, 住所とそれに対応する幾何図形のデータには, 多角形で住所を表現したものが少なく, 重心点で住所を表現しているものが多数あるため, 全ての住所において多角形検索はできない. したがって, 重心点による地理的検索を行う必要がある場合がある.

なおこの実験では住所の最小の単位を「丁目」としている.

本実験では, 地理的にもっとも理想的な検索手法だと思われる多角形検索の検索結果を検索の正解として用い, キーワード検索（および代表点検索）の適合率および再現率を求めた.

利用者の検索領域が検索点 t, 検索半径 r で表される検索に対するキーワード検索の適合率 $P_{key}(t,r)$ および再現率 $R_{key}(t,r)$ は 4.3,4.4 式のように定義した.

$$
\begin{aligned}
P_{key}(t,r) &\stackrel{\text{def}}{=} \frac{\left|\boldsymbol{Result}_{(key,t,r)} \cap \boldsymbol{Result}_{(pol,t,r)}\right|}{\left|\boldsymbol{Result}_{(key,t,r)}\right|} \\
\overline{P_{key}(r)} &\stackrel{\text{def}}{=} \frac{\sum^{t} P_{(t,r)}}{n} \\
&\quad (\boldsymbol{Result}_{(key,t,r)} \neq \emptyset)
\end{aligned}
\tag{4.3}
$$

$$
\begin{aligned}
R_{key}(t,r) &\stackrel{\text{def}}{=} \frac{\left|\boldsymbol{Result}_{(key,t,r)} \cap \boldsymbol{Result}_{(pol,t,r)}\right|}{\left|\boldsymbol{Result}_{(pol,t,r)}\right|} \\
\overline{R_{key}(r)} &\stackrel{\text{def}}{=} \frac{\sum^{t} R_{(t,r)}}{n} \\
&\quad (\boldsymbol{Result}_{(key,t,r)} \neq \emptyset) \\
&\quad (\boldsymbol{Result}_{(pol,t,r)} \neq \emptyset)
\end{aligned}
\tag{4.4}
$$

$\boldsymbol{Result}$ は検索結果の集合をあらわし,key はキーワード検索を,pol は多角形検索をあらわす. 例えば, 検索点 t, 検索半径 r における多角形検索結果の集合は $\boldsymbol{Result}_{(pol,t,r)}$ のように表現される. また,$\overline{P_{key}(r)}$ および $\overline{R_{key}(r)}$ は適合率および再現率の平均をあらわし,n は検索を行った点の数である. なお代表点検索の適合率と再現率 $P_{(poi,t,r)}$ と $R_{(poi,t,r)}$ も同様に定義できる.

キーワード検索が 3 種類に別れている理由は, キーワード検索では, 地理的検索と異なり, 任意の地理的範囲における検索ができないためである. そこで,「丁目」,「町字」,「市区町村」という住所を 3 段階に単純拡大し, その中で, 最良の結果をキーワード検索の結果として用いるという条件下での比較を行うことを考える.

検索半径が小さい時は, キーワード検索では「丁目」検索が, 大きい時は「市区町村」検索が最良の結果が期待できる. 実際に検索半径を変化させた時, どの段階のキーワード検索が最良の結果になるかは Rijsbergen 尺度 [Rijsbergen 1979] を用いて決定した. この尺度は適合率と再現率の混成尺度であり, この値が大きい程, その手法の検索性能が良いとされる. 式 4.5 に Rijsbergen 尺度の式を示す.

$$
E = 1 - \frac{1}{\alpha\left(\frac{1}{P}\right) + (1-\alpha)\frac{1}{R}} \tag{4.5}
$$

P は適合率, R は再現率, α は重みである. 今回は, 適合率と再現率を等分に評価するために, $\alpha = 0.5$ とした. この値は F 値において適合率と再現率を同等に扱うケースと同一である.

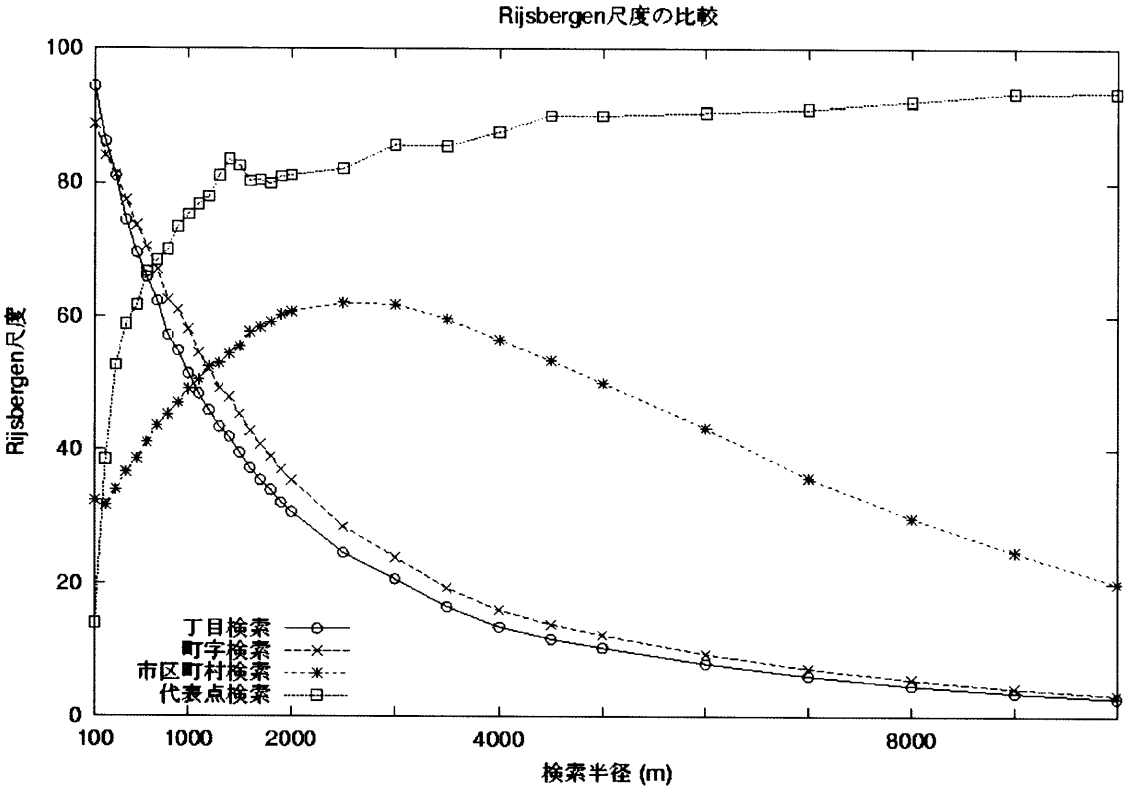

図 4.7 Rijsbergen 尺度の比較

ここで実際に検索を行った結果を報告する．検索の対象となるウェブ文書は 4.2 章に示したウェブクローラが収集した文書 61,265 ページを用い, 検索点はランダムに選択した 150 点, 検索半径は 100m〜10km とした．実験結果を図 4.7 に示す．

このように半径が 200m 以下の時は「丁目検索」が，続いて 1,100m までは「町字」検索が，それ以上では「市区町村」検索の性能が高いことが分かる．またこの図から，キーワード検索は，ほとんどの検索範囲において，3 手法のうちで，最も性能が悪いことが確認された．

半径–適合率，再現率曲線を示す．なお多角形検索の適合率，再現率は今回の定義から，どの半径においても 100% である．

図 4.8 に示すように，キーワード検索の適合率は，階層が切り替わる境界で，40% 程度まで低下することもある．

また，図 4.9 から，キーワード検索の再現率は，高い部分でも 75% 程度であり，さらに検索半径を拡大するに従って減少する．すなわちキーワード検索では最も良い条件下で

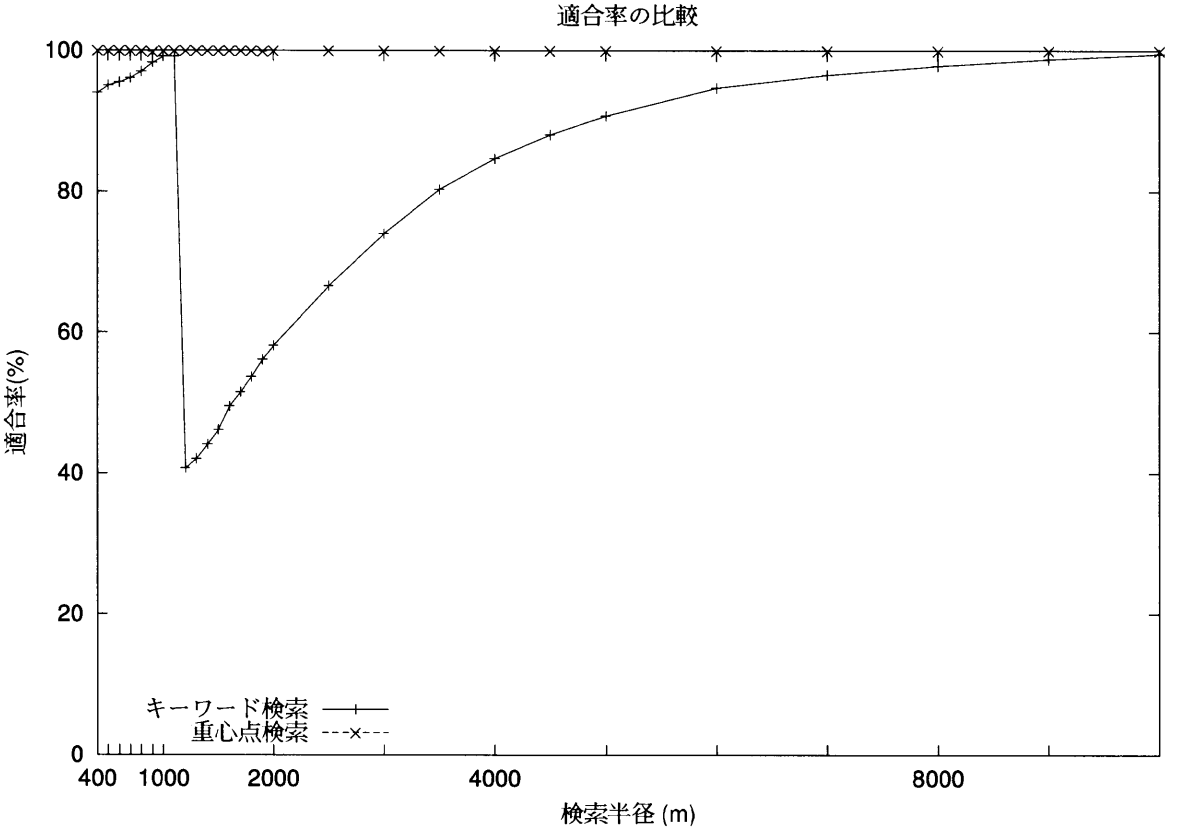

図 4.8 適合率の比較

も約 25% の検索もれを起こしており，提案する地理的検索手法はこの検索もれを解消することができる．

なお，代表点検索に関しては，検索範囲が小さいときは，検索範囲中に，代表点が含まれない場合があり，再現率が低くなっているが，検索範囲の拡大とともに，再現率が高くなる．

以上に示した通り，位置指向の検索において，キーワード検索は検索範囲の拡大と共に性能が悪化する．提案する地理的検索手法は，キーワード検索で少なくとも約 25% 存在していた検索もれを解消することができた．

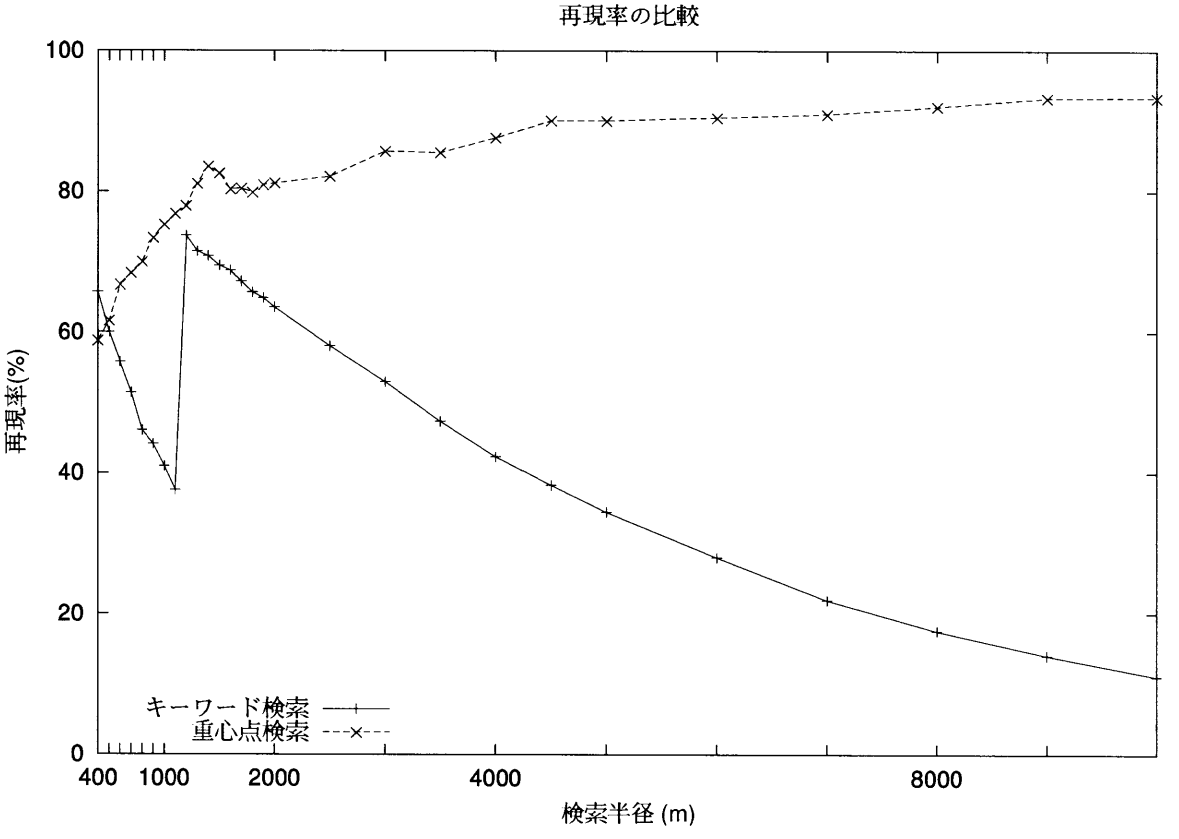

図 4.9 再現率の比較

4.5 位置指向のサーチエンジン「ここのサーチ」

4.5.1 モーバイルインフォサーチ

著者らのプロジェクトでは，1997 年 9 月からインターネット上の情報を位置で統合し，検索可能にすることを目標とした，モーバイルインフォサーチ公開実験 [三浦 1997][高橋 1998] を行っている．1997 年 9 月〜1998 年 6 月にかけて行った実験では，インターネット上で公開運用されている異なる位置関連検索サーバ（電話帳，地図，タウン情報，天気予報など）に対して，統一されたインターフェイスで位置指向検索することを可能にし，データベースの異種性を解消するシステム，いわゆるデータベースラッパー [Lewis 1991] を実装し評価実験 [三浦 1998][高橋 2000] を行った．この実験により，多様な既存のデータベースに対し，位置指向検索が可能であることが確認出来たが，一般のウェブ文書の位置指向検索は不可能であった．そこで本章で紹介した位置指向

の検索手法を実装し,「ここのサーチ」として公開実験を開始した.

4.5.2 ここのサーチの検索手順

ここのサーチは現在地などの位置を検索条件としてウェブ文書を検索し，URL，タイトル，文書の抜粋，文書に含まれる位置情報などに，検索条件として与えられた位置からの距離を加えて出力する位置指向のサーチエンジンである．出力の順は距離の近い順である．なお検索条件の半径はシステムが自動的に決定する方法を取っている．「ここのサーチ」の検索手順を以下に示す.

1. 利用者は, 緯度経度 (GPS,PHS, 手動入力), 住所, 駅名, 郵便番号等の位置情報をWWW ブラウザ経由で指定する.
2. 緯度経度はそのまま, 住所, 駅名, 郵便番号は緯度経度に変換されて, 検索が開始される.
3. 最小半径 (100m) で検索を行う.
4. 指定件数 (50〜100 件) に達しなかった場合は, 指定件数へ達するまで検索半径を広げる. 最大半径 (初期値:2000km) まで拡大して, 件数が足りない場合は処理を (6) へ移す.
5. 拡大した検索半径で検索結果が指定件数に達した場合は (6) へ処理を移す. 指定件数を越えた場合は, 検索半径を指定件数に達する直前の検索半径と現在の検索半径の平均値へ設定し, 最大半径を現在の検索半径へ変更して処理 (4) へ戻る. 但し, 繰り返しが上限 (3 回) を越えたら (6) へ処理を移す.
6. 利用者に検索結果を返す.

上述のように, 本手法では, 検索結果が適切な数となるように，検索範囲を調節できるため, キーワード検索を行った場合よりも，より適切な件数の結果を検索者に提示できる.

4.5.3 実験結果から

住所のまたがり

4.4 章に示したように，地理的検索は，キーワード検索と比較して検索もれが少ない.

そこで，実際の検索での地理的検索の有効性を定量的に求めるために,「ここのサーチ」で実際に行われた検索から複数の住所にまたがる検索の割合を調査した．調査には「ここのサーチ」を開始した，1998 年 9 月〜1999 年 5 月までの検索ログに記録された，82,136 検索を用いた．分析に使った検索では 1 検索あたり平均 162 文書が検索された.

結果を表 4.5 へ示す．全検索数:82,136 は，平均検索半径:2450m であった.

調査の結果，例えば複数の町字にまたがる検索は全体の検索の 67.5% 程度あることが

表 4.5　「ここのサーチ」における複数住所検索の割合

またがり住所階層	検索回数 (回)	割合 (%)
都道府県	10243	12.47
市区町村	33635	40.95
町 字	11587	14.11

確認できた．これらの検索では特に地理的検索が有効になっている．

図 4.10 にここのサーチの検索例を示す．この図では利用者は東京都と千葉県と埼玉県の 3 県の境界近くにいて，ここのサーチを行っている．その結果，右側のフレームに示されている通り，葛飾（東京都），三郷（埼玉県），松戸（千葉県）の 3 県にまたがる結果が出力されている．

図 4.10　ここのサーチの検索例

ウェブコンテンツの地域分散

また,「ここのサーチ」で検索可能な地域の調査を行った.

調査は以下の手順で行った.

- ウェブクローラが収集したウェブ文書から 4.3.1 節に示した手法により，住所を抽出する.
- 抽出した住所を同じ手法で緯度経度へ変換する.
- 変換された緯度経度を白地図上にプロットする.

図 4.11 は上記の手順に沿ってプロットしたもので，青色から順に赤色に近付くにしたがって，その緯度経度に対応するウェブ文書の件数が多いことを示している．図 4.11 をみると，位置情報を持つウェブ文書は日本全土に分散していることがわかった.

図4.12のグラフは，ウェブコンテンツと都道府県の関係を示している．収集した情報で位置情報の構造化ができたページの数を都道府県に対応づけて表示したものである．東京が人口，Web ページ共に多いが，北海道や長野（オリンピックの時期であった)，京都といった観光地は人口に比べて Web ページが多い．一方東京のベッドタウンとしてしられている埼玉と千葉が人口に比べてページが少ないのが興味深い． これは Web ページに地理位置を付与してできる解析の一例である．Web 情報の地理情報を用いた可視化は単純な地図表現を超えて色々なものがあるべきである.

4.5.4 実装

ここのサーチの実装に関して説明する．ここのサーチは UNIX 上のサーバアプリケーションで，Apache ウェブサーバと，Informax RDBMS および SpatialWare を用いて実装した．データ構成を図4.5.4に示す．PK, FK, Ix は主キー，外部キー，インデックスを示す.

検索は検索条件（円）と位置情報データ（重心）の重なりによって行った．重心点への距離が同じものが複数存在した場合は，住所階層が深いものを優先した．投入するデータ量を削減するために，重心点データへの距離による検索を行ったが，重心点が対象図形の閉曲線内に存在しない場合（例えば凹図形）が存在する．この場合は重心点が検索条件に含まれていても，対称図形の閉曲線が円と交わらない場合がある．このような場合でも正しく検索を行うためには，検索条件と検索対象図形の交わりの面積で評価する必要があり．このような検索の実装も行っている.

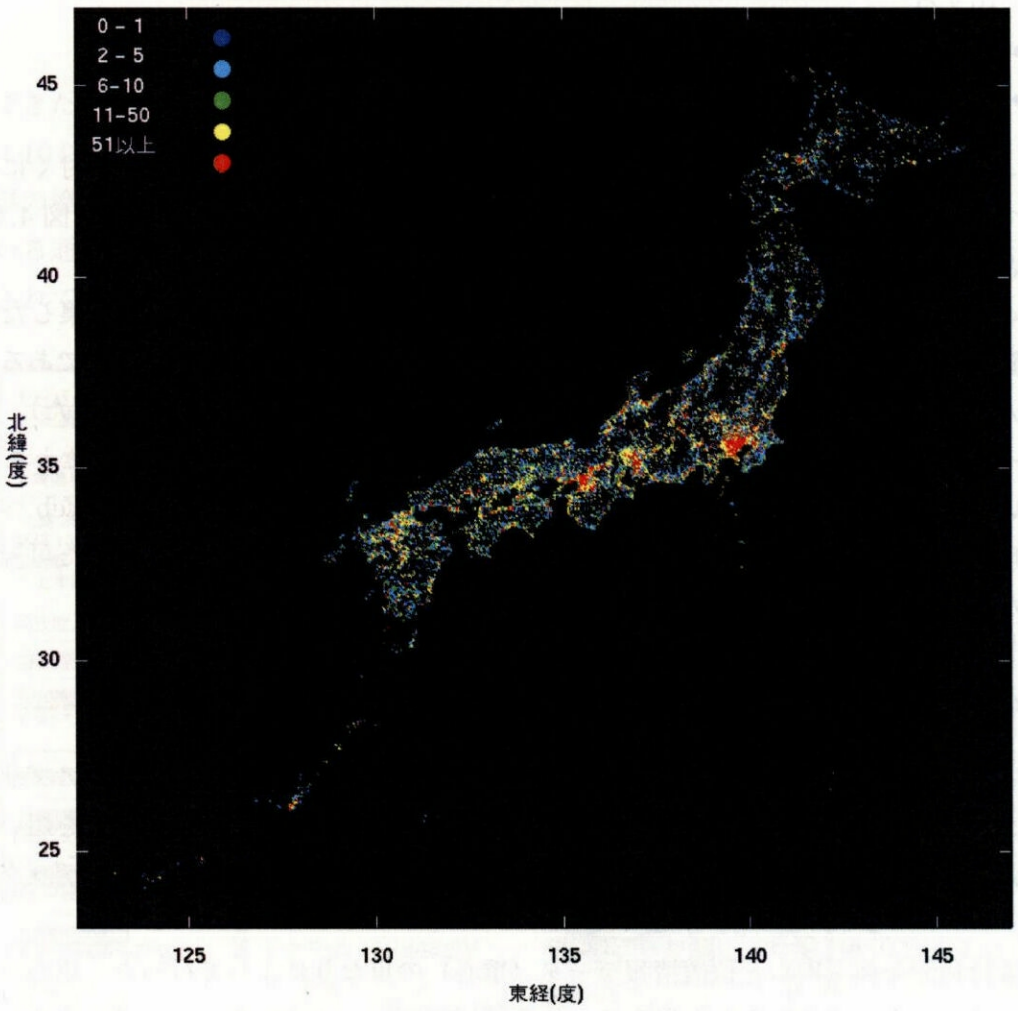

図 4.11 ウェブ文書の地理的分散

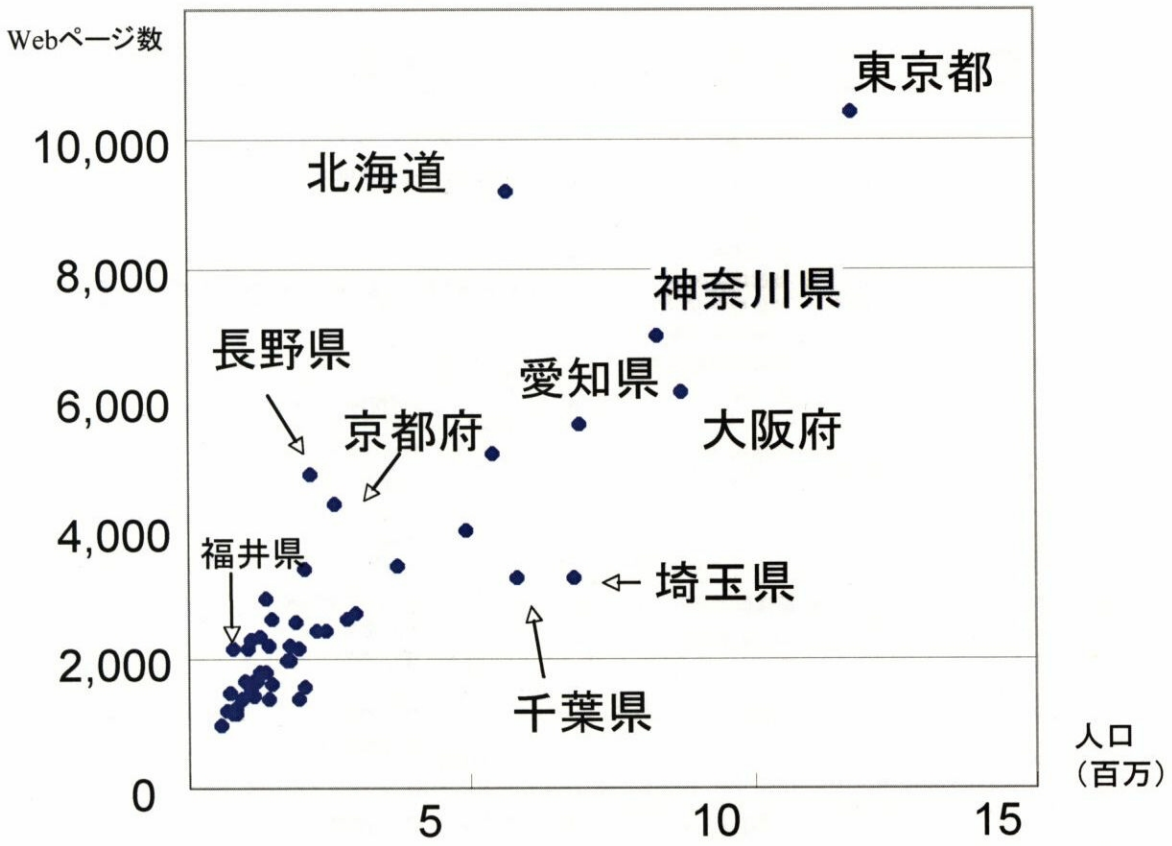

図 4.12　実験で収集したウェブページの都道府県とその人口との関連

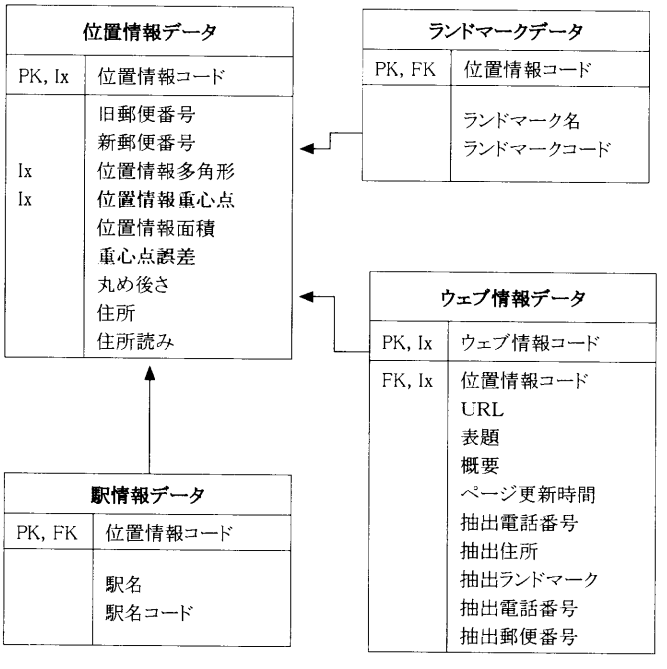

図 4.13 ここのサーチのデータの連携