# 第5章

# アクセスログから地理特性を提示するためのジオワード・マイニング

## 5.1 ローカルサーチとジオワード

ローカルサーチとは，地理的な位置を条件として情報を提供する検索サービスである．ローカルサーチが広くインターネットのキラーアプリケーションとして認識されたのは，2004年に米国google，Ask Jeeves，Yahoo!などの企業が一斉に位置依存情報サービスを「local」という単語を使って開始したことが一つのきっかけとなった．2004年にローカルサーチが普及した背景には，利用者ニーズの変化，情報提供社者のニーズ変化，環境の変化などがあると考えられる．利用者ニーズの変化としては，インターネットが本格的に普及したことにより，より日常生活に密着した情報，すなわちローカル情報のニーズが増している．情報提供者側は，インターネット広告の方法がバナー広告からキーワード広告へシフトするにつれて，広告提供者の底辺が広がり，結果としてローカルサーチに関心が集まっている．さらに地図情報が身近なものとなった環境の変化も見逃すことができない．このようにローカルサーチ発展は複数の要因に支えられているので，今後も技術，産業共に発展が期待できる分野である．図5.1は代表的なローカルサーチの例である．検索したい住所と目的を入力することで情報が検索され，結果は地図上にプロット表示されている．

しかしローカルサーチは地理的な情報検索を可能にしたものの，その地域の特色を表現することには成功してない．検索条件に合致した情報を地図にマッピングするするインタフェイスは実用上の利便性もあり，見た目にもインパクトがあるが，街や地域の雰囲気といったものを表現するには至っていない．地域の特色を表現するメディアは未だ書籍が強い分野である．ガイドブックは観光地や有名店舗，あるいは歴史を写真や説明文で表現している．これは人手によるところが大きい．一方自治体白書ではその地の面積，人口，経


図 5.1　ローカルサーチの例（Google マップ BETA）

済，財政，産業，社会サービスなどといったデータがその役割を担っている．

インターネットのから統計的な知識を導きだす考え方にウェブマイニングがある．コンテンツマイニングは，ウェブ文書のコンテンツから，コミュニティマイニングはウェブページのリンク関係から，ログマイニングは利用者のアクセスログから統計的に知識を発見する方法である．本研究のアプローチとしては，ローカルサーチのサーバのアクセスログに含まれる地理的なリクエストを使って，地域の特色を取り出す方法を検討する．ローカルサーチのアクセスログには地理的なニーズが隠されている可能性がある．

図5.2に典型的なローカルサーチの検索インタフェイスを示す．ローカルサーチの利用者はキーワードと住所を入力するウィンドウを使って検索を行う．

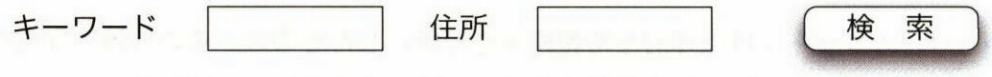



図 5.2　ローカルサーチの典型的な検索インタフェイス

ここで住所や駅名，地名などの地上の地理位置につけられた固有名詞をジオワードと呼

ぶことにする．ジオワードは普段から人が地理位置を指示するために用いている方法であり，ローカルサーチにとっても重要な方法である．

ジオワード・マイニングは，大量のジオワードを含んだデータから地理的な知識，例えば地域の特色を発見することである．本章の目的は，ローカルサーチに記録された利用者の検索ログ（アクセスログ）に残された大量のジオワードを含んだ検索語を解析する手法を明らかにすること，同ログからローカルサーチに有益な知識を発見すること，さらにはアクセスログの地理的な検索語の集合から地域ごとに存在する地理的な特性を発見することをねらいとする．

## 5.2 アクセスログのジオワード・マイニング

### 5.2.1 ジオワード・マイニングとは

ジオワード・マイニングはジオワードを含んだ大量のデータから地理に関する知識を発見することである．ジオワード・マイニングは空間データマイニング [Ester 2001] の考え方を元にジオワードを用いるものである．

空間データマイニングにおいても代表的な手法は相関ルールである．空間相関ルールは次のように定義できる [Koperski 1995].

**定義** 空間相関ルール

$P_1 \wedge ... \wedge P_m \rightarrow Q_1 \wedge ... \wedge Q_m$

但し，述語 $P_1...P_mQ_1...Q_m$ のうち少なくとも一つが地理に関する表現を持つ．

空間相関ルールでは，例えば次のようにジオワードを用いることができる．

**定義** ジオワード相関ルール

相関ルールにジオワードが含まれるもの

例：札幌 → ラーメン

**定義** 地理的一般化相関ルール

相関ルールの一般化に地理的情報を使うもの

例：上野 → 美術館，渋谷 → 美術館というルール集合から得られるより一般的な東京 → 美術館

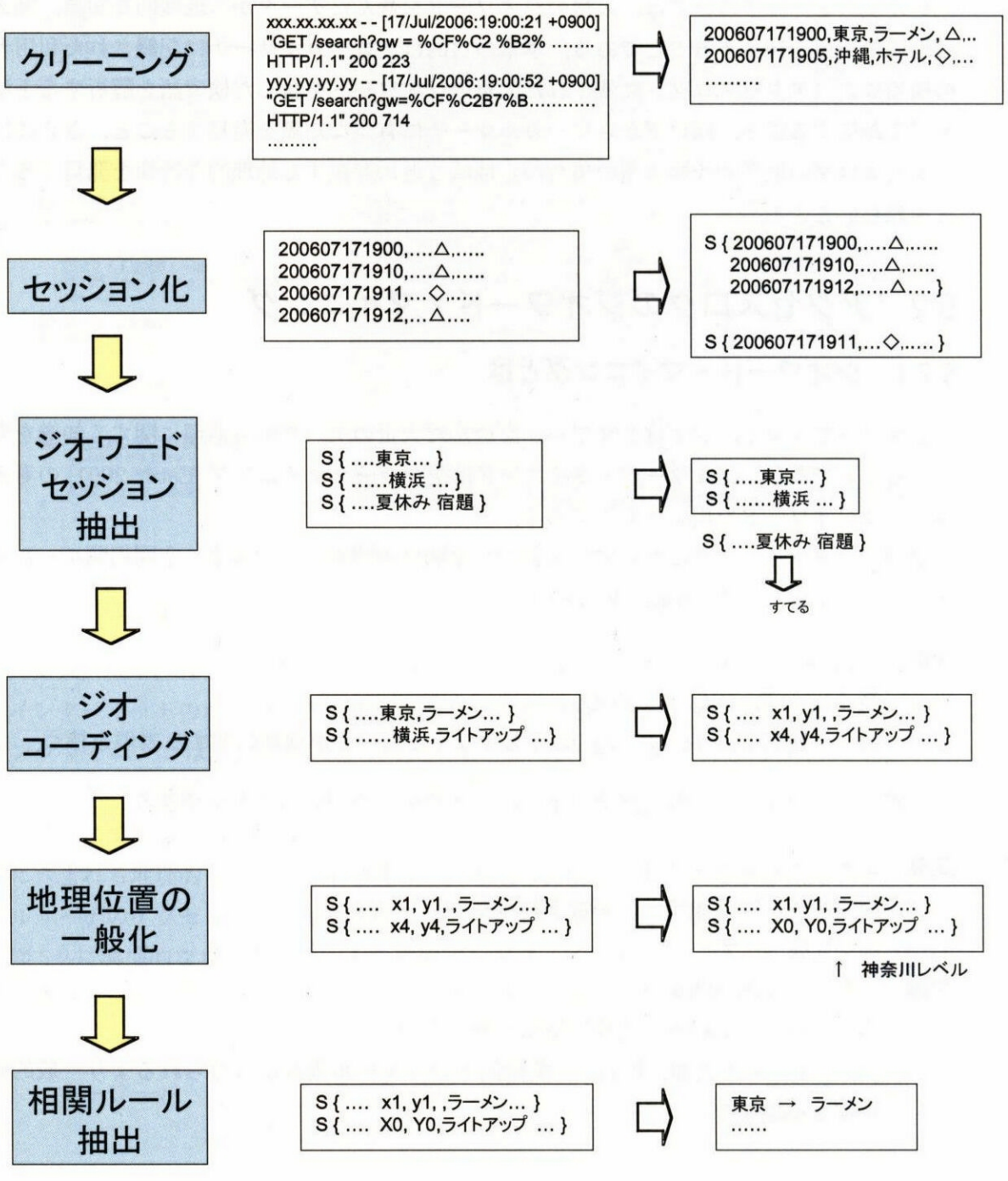



図 5.3　アクセスログのジオワード・マイニング

### 5.2.2 アクセスログのジオワード・マイニング方法

サーチエンジンのアクセスログを解析することにより，利用キーワードの多寡やランキングがわかるだけでなく，単語推薦やキーワード広告のマーケティングに用いるためのルールを得ることができる．ここではジオワード相関ルールをマイニングする方法を説明する．

**■クリーニング** 入力されたログデータから必要な情報のみを抜き出すプロセス．例えばhttp デーモンが出力するアクセスログの各アクセス（リクエストと呼ぶ）から，以下の要素を持つリクエストを抜き出して，下記フォーマットに変換する．

タイムスタンプ，リクエストワード，ID 情報（cookie）

ID 情報はそのサービスが利用者に払い出す http cookie を用いることが多いが，アクセス端末の IP アドレスを用いることもできる．なお，このプロセスで，解析に関係のないログデータ（例えば画像へのリクエスト）や非正常なログデータ（規定のフォーマットに反するもの，文字コードの判定ができないもの等）を除外することも行う．

**■セッション化** セッションとは同一利用者からの連続的なリクエストの集合をいう．本研究ではセッションの連続性を，あるリクエストから次のリクエストまでの時間間隔が30 分を超えないこと，と定義した．セッションはマイニングの処理の基本単位の一つで，同一セッション内でのリクエストを等価に考えると，セッションは通常のデータマイニングで同時購入をさす「バスケット」となる．

**■ジオワードセッションの抽出** ログデータからジオワードを含んだセッションのみを抜き出す．このプロセスは次のジオコーディングと同時に行うこともできる．なお，イエローページ等の専用のローカルサーチエンジンではあらゆる利用者リクエストが地理関連であるとみなせるのでこのステップは不要になる．

**■ジオワードの正規化とジオコーディング** ジオワードを地理位置に関連づける．ジオコーディングとはオブジェクト（この場合はジオワード）を地図上にマッピングする作業で，具体的には住所に対応する緯度経度情報を付与する作業を呼ぶことが多い．ジオコーディングはジオワードと緯度経度の対照表を参照して行う．住所対照表は正しい住所で記述されていることが前提となるので，参照の前段として，ジオワードを正規形に変換する必要がある．ジオワードの正規形とは地理位置を一意に特定できるジオワードで，かつ「正式な」表記方法を持ったものである．

**■地理位置の一般化**　地理位置をより広い大きさを持った地理位置に変換する．ルールは構成要素を一般化すれば，複数の要素が一つの要素に合併されて，より頻度の高い集合を作ることができる．例とすれば「ズボン」と「シャツ」をシソーラスを用いて「衣類」に一般化することが可能であるが，ジオワードの場合は，住所の住所階層を用いた一般化や，ジオワードの地理領域を用いて，近接するジオワードと／包含するジオワードに合併することによる一般化ができる．

**■相関ルール抽出**　ジオワードセッションの集合から相関ルールを抽出する．相関ルールの抽出は従来の手法を用いる [Agrawal 1994]．住所階層を用いてジオワードの一般化を行う場合は，一般化を同時に行うことにより処理の効率化も可能である [Srikant 1995]．

### 5.2.3　アクセスログのジオワード・マイニングの定義

アクセスログのジオワード・マイニングを前節の分類にもとづいて定義する．今回の解析では，利用者の連続的なリクエストの集合であるセッションを，データマイニングでいうバスケット（トランザクション）と扱うこととする．その考えを元に，リクエスト，セッション，タプル，相関ルールと，それぞれのジオワードとの関連を定義する．

**定義**　リクエスト $r$

$r = \{id, t, w \mid id \in \mathrm{ID}, t \in \mathrm{T}\ , w \in \mathrm{W}\}$

$ID$ を ID 集合，$\mathcal{T}$ を時間，$\mathcal{W}$ を単語集合（アイテム集合）とする．

**定義**　セッション $s$

$s = \{r_{t1}, r_{t2}, ..., r_{tm} \mid \exists id \in \mathrm{ID}\ , \forall r \in R_{id}, \exists \Delta T, \forall t_i, t_{i+1} - t_i < \Delta T\}$

$R_{id}$をある $id$ を持つリクエストの集合とする．

**定義**　ジオワードセッション $gs \in \mathrm{S}$ (S はセッション集合)

$gs = \{r_1, r_2, ...r_n \mid \exists r_k \in \mathrm{G}\ \}$

$\mathrm{G} \subseteq \mathrm{W}$ をジオワード集合とする．

**定義**　ジオワードタプル $gt$

$gt = \{g, w_k, ..., w_n \mid \exists g \in \mathrm{G}, w \in \mathrm{W}\ \}$

**定義**　相関ルール　$R : X \rightarrow Y$

$X \rightarrow Y; X \subseteq W, X \subseteq W, X \cap Y = \phi.$

支持度 $supp(R)$ は $X$ と $Y$ を同時に含むセッションの割合

確信度 $conf(R)$ は $X$ を含むセッションが $Y$ を含む割合
$conf(R) = support(R)/sup(X)$.

**定義** ジオワード相関ルール $R_g : X \rightarrow Y$
$X \rightarrow Y; \exists g \in \mathrm{G}, g \in X$ または $g \in Y$.

**定義** Co-location ルール $R_{cl} : X \rightarrow Y$
$X \rightarrow Y; \exists gt \in \mathrm{GT}, X \in gt$ かつ $Y \in gt$. GT はジオワードタプル集合.

## 5.3 ジオワード相関ルール作成実験

### 5.3.1 ローカルサーチ検索ログからの相関ルール作成実験

**■解析に使ったデータ** 本実験で解析に利用したデータは NTT 番号情報 (株) が提供しているiタウンページの検索ログデータである. iタウンページはいわゆるイエローページ（職業別電話帳）検索を提供する専門のローカルサーチエンジンである. 検索ログデータは, 表5.1に示す通り時間とフリーワード, 住所（ジオワード）を含んでいる. iタウンページ検索のユーザインタフェイス画面は, 図5.2と同様の構成を持ち, 職業名や企業名を入力できるキーワード入力フォームと住所を入力する住所フォームが分かれているため, 解析時もそれぞれの単語がフリーワードとして入力されたのか, ジオワードとして入力されたのであるかは与えられている. なお, 本検索サービスは住所を指定することが前提となっているので, 全ての検索はローカルサーチとして扱う. なお ID には http-cookie の値が暗号化された形で提供されている.

表 5.1 実験に利用したログデータ

| 時間 | ジオワード | フリーワード | ID |
|---|---|---|---|
| 20/Sep/2003:08:19:48 | 東京都渋谷区 | ホテル | xxxxxxxx |
| 20/Sep/2003:08:20:30 | 東京都港区 | レストラン | xxxxxxxx |

**■データクリーニングの概要** データクリーニング処理の概要を説明する. 生のログデータはそのままではマイニング等の処理に用いることができないため, 必要なリクエストや属性を抜き出す必要がある. 具体的には以下の処理を行う.

- はじめに生のログデータから, 検索リクエスト以外のログデータを取り除き, さ

らに検索ログデータから解析に必要な属性のみを取り出して表5.1の形式に変換した．

- 次に同一利用者からの連続的なリクエストまとめてセッションとした．本実験では，同一利用者が前後30分を超えない間隔で行ったリクエストを同一セッションに属すると定義した．利用者の同一性に関してはhttp-cookieを用いた．
- セッション化にあたって，データ収集目的の大量リクエスト（自動収集の場合は特にクローラと呼ぶ）によるアクセスを解析から排除するために，長いセッション（今回は1セッションのリクエストが1000回を超えるもの）を削除した．なお，クローラはhttp-cookieを受け取らないものがあり，その場合は毎回のcookieが異なることがある．従ってこの方法で全てのクローラを排除することはできない．ローカルサーチのログ解析におけるoutlierの排除の議論は本論文では行わない．

**■ジオコーディングの概要**　入力されたジオワードを住所等の辞書データを参照しながらジオコーディングを行った．サーチエンジンのリクエストのジオワードは省略等が存在し，正しい形で入力されないケースが数多く存在する．ここで使った方法は，従来の住所表記のバリエーションの考え方に基づいて省略住所を補完するスタンドアロン法に加えて，スタンドアロン法では解決できなかった約1.1%（頻出上位10,000ジオワードの約3%）のあいまいな省略ジオワードを，セッション情報を用いて上位住所を推定してジオコーディングを行う方法を行った．詳しい内容は6.2に記した．

**■ログデータの概要**　上記の方法で相関ルールの作成処理を行った．はじめに元データ23億ヒットのログからクリーニングを行い，一定の形式を満たす検索リクエストを選択したところ，8,600万ヒットが取り出された．さらにこの8,600万検索が3,200万セッションに分類された．この8,600万回の検索に現れるジオワード，フリーワードはそれぞれ異なり数130万，870万である．実際に検索されている頻度の上位の単語を表5.2に示す．なおこれらのワードには，利用者の操作ミスによる，非単語入力や文字化け等で解析できない文字列も含んでおり，実際にジオワードが130万種存在するわけではない．

表5.2　ログデータ解析の概要

| | 件数 | サイズ |
|---|---|---|
| 全データ | 23億 ヒット | 700 GB |
| 解析対象の検索リクエスト | 86,522,189 ヒット | 4.844 GB |
| 同，セッション | 32,054,172 セッション | 1.195 GB |

**■相関ルール作成実験** クリーニングからジオコーディングまでを行ったデータから相関ルールを作成した．その結果を表5.3に示す．クリーニングデータから 2,000 万検索を用いて相関ルールの抽出を行った結果，最小サポート値 0.000001(1E06) の元で 100 万通りのルールを得た．

表 5.3 相関ルール導出の結果

| | 個数 | 参考 |
|---|---|---|
| ルール作成用データ（※ 1） | 20,053,210 検索 | 1.1GB |
| 作成されたセッション（※ 2） | 7,869,824 セッション | 2.4 検索／セッション |
| 作成された相関ルール（※ 3） | 1,014,388 ルール | |

※ 1 全期間の冒頭の 3 ヶ月分

※ 2 最長セッション 10，最大セッション内単語数 12 に制限

※ 3 最小サポート $1x10^{-6}$，最大ルール内単語数 3 に制限

表 5.4 作成された相関ルールの分類

| 要素数 | ルールの種類 | ルール数 |
|---|---|---|
| 2 要素間ルール | ジオワード→フリーワード | 181,052 |
| 2 要素間ルール | フリーワード→ジオワード | 181,052 |
| 2 要素間ルール | ジオワード→ジオワード | 136,458 |
| 2 要素間ルール | フリーワード→フリーワード | 66,286 |
| 3 要素間ルール | ジオワード→（ジオワード，ジオワード）等（ジオワードのみ） | 267,282 |
| 3 要素間ルール | ジオワード→（ジオワード，フリーワード）等（混成） | 137,340 |
| 3 要素間ルール | フリーワード→（フリーワード，フリーワード）等（フリーワードのみ） | 43,194 |
| 合計 | | 1,014,388 |

表 5.5 作成されたルールの例

| 種別 | 例 | サポート値 |
|---|---|---|
| ジオワード→フリーワード | 東京都 → ホテル | 0.00029 |
| ジオワード→フリーワード | 東京都 → 病院 | 0.00020 |
| フリーワード→フリーワード | ホテル → ビジネスホテル | 0.00037 |
| ジオワード (複) →フリーワード | (神奈川県, 東京都) → 埼玉県 | 0.00046 |

## 5.4　ジオワード相関ルールの興味深さに関する考察

表5.6　ジオワードからフリーワードへのルールの例（サポート値上位5件）

| 青森県 | 東京都中央区銀座 | 神奈川県鎌倉市大船 |
|---|---|---|
| ホテル | 美容院 | 居酒屋 |
| 病院 | 飲食店 | ホテル |
| 飲食店 | ホテル | ビジネスホテル |
| 居酒屋 | 書店 | 美容院 |
| 温泉 | レストラン | 飲食店 |

表5.6に「ジオワード→フリーワード」の形式のルールの例を示す．それぞれサポート値が上位，すなわち頻繁に検索される5件を掲出したものである．サポート値はコンフィデンス値と並んで相関ルールの重要性を判断するポピュラーな数値である．表5.6からわかる通り，ホテルと飲食店が3地域全ての上位5つに入ってくるほか，美容院2地域に入ってくるなど地域が異なっても同様のルールしか得られない．このことは，一つには地域を越えた普遍的な知識／ニーズの存在を示唆するが，地域ごとに特色がわかるルールの選択法が必要である．なおコンフィデンス値はルールの左項が同じであれば，サポート値と意味することは同じであるため，この場合はサポート値に代表させて議論すれば良い．

### 5.4.1　事象の地域に対する集中指数

ここでルールの地域特性に関する2つの指標を導入する．一つは事象の分布が特定の地域に集中しているか，あるいは均一の地域に分布しているかを示す集中指数，もう一つは，地域の特徴を調べるために，事象がその地域に特に多く発生しているのか否かを判断する指標，特化係数である．順に説明する．

事象が地域的に集中しているか，均一に分布しているかを示す集中指数は次のように定義される．

**定義**　事象xに対する集中指数 $Conc_x$

$$Conc_x = \frac{1}{2}\sum_i \left| \frac{x_i}{\sum x} - \frac{a_i}{\sum a} \right|$$

N個の小地域について，ある事象 $x$ が観測され，その観測値が $x_i$ であるとする．$x$ が空間的に均等に分布しているとすれば，$x_i$ は小地域 $i$ の面積 $a_i$ に比例するから両者の比

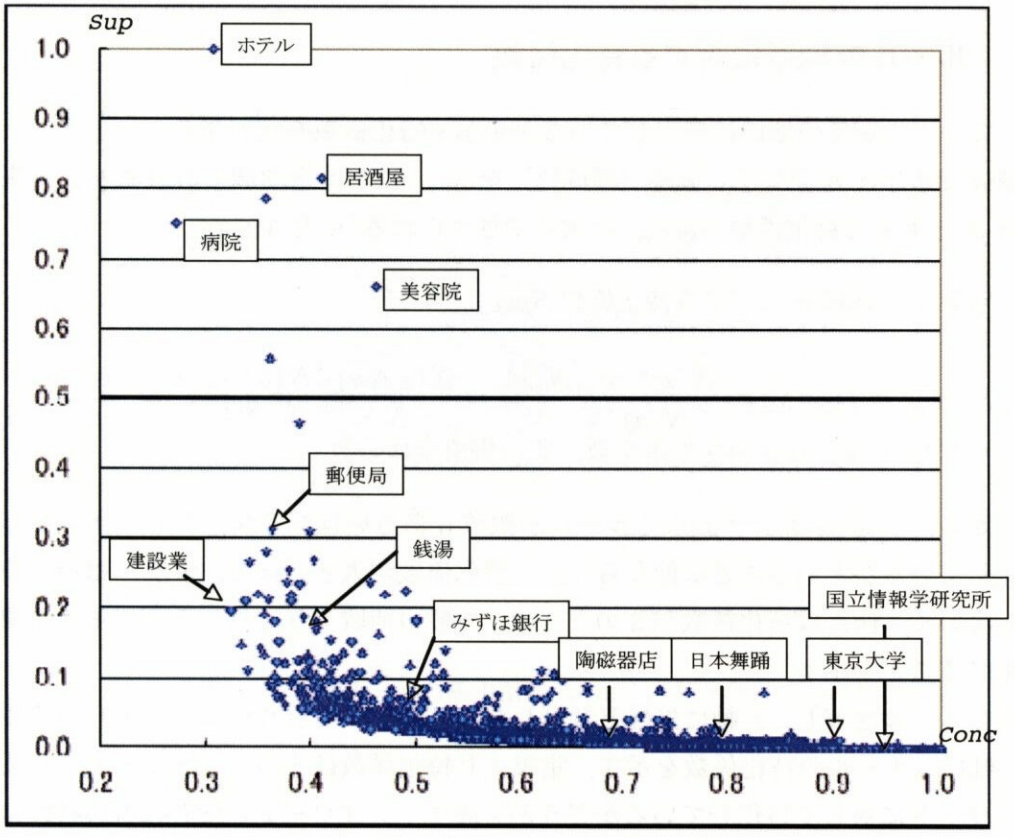



図 5.4 集中指数：検索語の地域に対する偏り

率は等しくなる．したがって次式で与えられる値を計算すれば分布の均一さを知ることができる．*Conc* は 0 以上 1 未満の値をとるが，*Conc* の値が小さければ均一に分布し，1 に近づくほど偏りがあることになる [中井 1994].

なお，本分析では，小地域をリクエストされたジオワードに，事象 $x$ をジオワードとともにリクエストされたフリーワードにした．また $a$ の値は，面積を使わずに，問い合わせの中での分布の偏りを見るためにジオワードに対するリクエスト回数（サポート値）を用いている．

図 5.4 に検索語の地域に対する偏りを示す．各点をフリーワードとし，x 軸に集中度数 (Conc)，y 軸にフリーワードの総検索回数 (Sup，相対値) を示す．定義より x 軸の値が大きいほどリクエストが特定のジオワードに集中し，一方小さいものは分散している．図 5.4 より病院やホテルが他よりもより均一にリクエストされていることがわかる．

### 5.4.2　ルールの地域に対する特化係数

続いてルールがその地域に特化しているかを示す特化係数を定義する.

地域 $a$ で事象 $x$ が発生する確率（構成比）を $p_{xa}$, 同じく全地域での事象 $x$ が発生する確率を $p_x$ とすると特化係数 $Spec_{xa}$ は次式で与えられる [中井 1994].

**定義**　事象 $x$ の地域 $a$ に対する特化係数 $Spec_{xa}$

$$Spec_{xa} = P_{xa}/P_x = \frac{N(x \wedge a)}{N(a)} / \frac{N(x)}{N} = \frac{N(x \wedge a)}{N} / \frac{N(a)N(x)}{N^2} = \frac{sup(x \wedge a)}{sup(x)sup(a)}$$

ただし $N(\alpha)$ は $\alpha$ の発生事象数, $N$ は事象全体の数

この式から, $Spec_{xa}$ は $a \rightarrow x$ なるルールが領域 a あるいはジオワード a に関してどの程度特化しているかということに他ならない. 特化係数が大きいほど, 事象 $x$ はその地域に特化している. 例えば特化係数が 2 のときは, 一般の地域より 2 倍の確率でその事象が頻繁に生じることを示す.

各点はルールを表し, x 軸に先ほど計算したフリーワード（ルールの右項）の集中指数を, y 軸にはルールの特化係数を示す. 定義より特化係数はそのフリーワードが与えられたジオワードに対して特化しているかどうかを表す. ジオワードの使われ方の特性は, ログデータが得られた環境における地域の特性であるので, 以下の地域特性がそのシステムの利用者ニーズにおける地域の特性として導かれる.

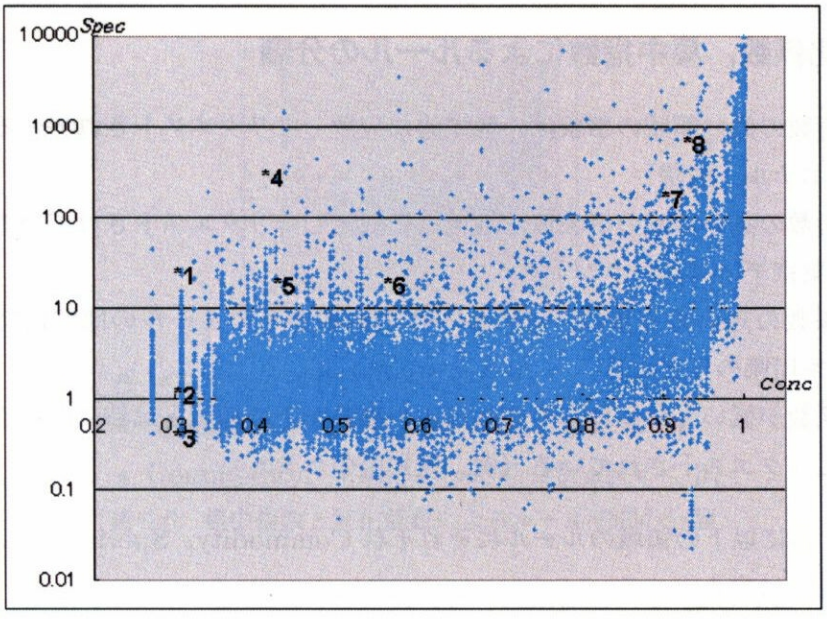



図 5.5 ジオワード相関ルールの集中指数 (Conc) と特化係数 (Spec) の関係

表 5.7 ジオワード相関ルールにおける集中指数と特化係数

| | ルール | 集中指数 | 特化係数 |
|---|---|---|---|
| 1 | 千葉県浦安市舞浜→ホテル | 0.31 | 26.63 |
| 2 | 大阪府茨木市→ホテル | 0.31 | 1.01 |
| 3 | 愛知県→ホテル | 0.31 | 0.42 |
| 4 | 大阪府大阪市北区西天満→弁護士 | 0.43 | 127.61 |
| 5 | 千葉県成田市→駐車場 | 0.44 | 12.83 |
| 6 | 香川県→うどん | 0.57 | 12.34 |
| 7 | 東京都文京区→東京大学 | 0.90 | 112.13 |
| 8 | 東京都浦安市→東京ディズニーランド | 0.93 | 537.31 |

### 5.4.3 特化係数，集中指数によるルールの分類

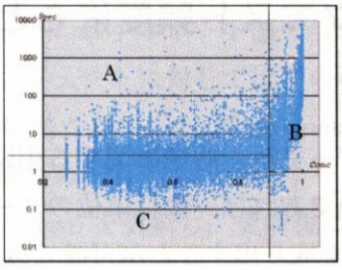
- 集中指数の低い領域の事象は，地域的に分散してリクエストされている事象である（例，ホテル，病院）
- 集中指数の高い領域の事象は，地域的に限定してリクエストされている事象である（例，東京ディズニーランド）
- 特化係数の高いルールは，そのルールの事象がそのルールの地域に特化している（例，香川県のうどん）．
- 特化係数が低いルールは，そのルールの事象がその地域とは特に関係を持たずに平均的にリクエストされている（例，愛知県のホテル）

以上から，特に以下の領域のルールにそれぞれ Commodity，Specialty，Locality と名前を付けることにする．

- Locality: 集中指数の高い事象をもつルール．そこにしかないルール
- Specialty: Locality ではないが，特定の地域に特色を持つルール
- Commodity: 上記以外．普遍的なルール，あるいは特徴を持たないルール

なお，上記以外の領域（集中指数高かつ特化係数低），あるいは上記領域の中でも外れ値に見えるものも存在しているが，これらはウェブ特有の現象によるものも含まれると考えられ（特定のルールがリンク等により極めて高いリクエストがされること／クローラ等を利用して地域網羅的にリクエストがされること等），利用の際には注意が必要なものも存在する．

本節では特化係数を用いれば一般的なルールの排除が，また集中指数を用いれば地域に排他的なルールを選ぶことができることを示した．だだし検索リクエストのログには，単純な地域の傾向を反映した検索だけでなく，クローラ等による一般的な性質に馴染まないケースや，あるいは探しているものが見つからないために繰り返し検索が起きているケース（これもニーズと実態に乖離があるという一つの特色ではあるが）などが存在することがあるため，ルールの利用目的によっては留意が必要な場合がある．




- Locality: $= \{r \mid Conc \geqq 0.9\}$
- Specialty: $= \{r \mid Conc < 0.9, Spec \geqq 3\}$
- Commodity: $= \{r \mid Conc < 0.9, Spec < 3\}$

図 5.6 集中指数と特化係数によるルール空間の分割

表 5.8 集中指数と特化係数によるルールの分類（Specialty）

| 相関ルール（Specialty） | | |
|---|---|---|
| 青森県 | 東京都中央区銀座 | 神奈川県鎌倉市大船 |
| 温泉 | 書店 | 居酒屋 |
| セメント | レストラン | ホテル |
| 肥料 | 広告代理業 | ビジネスホテル |
| 民宿 | 三越 | エルメス |
| 高校 | 美容院 | 歯科 |

表 5.9 集中指数と特化係数によるルールの分類（Localilty）

| 相関ルール（Locality） | | |
|---|---|---|
| 青森県 | 東京都中央区銀座 | 神奈川県鎌倉市大船 |
| 青森銀行 | 伊東屋 | 大船駅 |
| 青森県庁 | 博品館 | きじま |
| 青森駅 | 福家書店 | 鳥恵 |
| 東北電力 | バードランド | x |
| 八戸駅 | カルティエ | x |

## 5.5　ジオワードのリクエスト頻度にもとづくクラスタリング手法

ここまでに，アクセスログデータから地域に関する相関ルールを計算する方法を述べた．東京や大阪などの大都市圏であれば，数百単位のルールを見いだすことができるので，その中から興味深いルールを取り出すことが可能である．しかし大都市圏と一部の特定地域を除いた地域では，その地域へのリクエスト数そのものが少なく，ルールを抽出できないことがある．そのためジオワードへのリクエスト回数の頻度の多寡に関わらず，地理的な相関ルールを出力するためにはルールの一般化を行う必要がある．ジオワードは地理属性を持っているため，それを用いてよりジオワードの和集合に相当する領域を作ってやり，相関ルールが得られるようにすることを考える．

### 5.5.1　クラスタリングの定義

ジオワードの集合をクラスタリングで求める．目的は次の通りである目的：マイニングを行う地理的領域の単位を定められた最小サポート値を満たすように，地理的に近傍である地理領域を合併してより広い領域を作成する．

まずクラスタリングを行う単位ジオワードオブジェクト（gwo）をジオワード，空間図形，サポート値からなる組と定義する．ジオワードに対応する図形は一般に閉曲線であるが，ここでは円とする．具体的に今回は，ジオワードの集合を GW，ジオワードの空間図形を中心座標 (x, y) と半径 r，サポート値 s を全検索セッションにおけるその gwo が使われた回数の割合とし次のように定義する．

**定義**　$gwo$

$$gwo = \{gw, x, y, r, s \mid gw \in GW, x > 0, y > 0, r > 0, s = support(gw)\}$$

また gwo 間の距離 D を次のように定義する

**定義**　gwo 間の距離 D

$$D(gwo_1, gwo_2) = (gwo_1 \text{と } gwo_2 \text{の中心座標間の直線距離}) + r_1 + r_2$$

この距離の定義は（gwo を円と仮定した場合）2つの gwo 間の最遠点間距離を与える．これは半径の大きい gwo に他の gwo が吸収一般化されることを防ぐことを目的としている．クラスタリングのアルゴリズムを以下に示す．

アルゴリズム

1. ジオワードオブジェクトの最小サポート値を与える
2. 全てのジオワードオブジェクトに対して以下の処理を行う
   (a) 最小サポート値に満たない $gwo$ を $D(gwo, gwo_x)$ を最小とする $gwo_x$ に合併する（最寄りの $GWO$ への合併）
   (b) 合併を行うごとに gwo を更新し，更新後の全ての gwo サポート値が最小サポート値を超えるまで処理を続ける
3. 最小サポート値を更新して最初に戻る

$gwo$ の更新アルゴリズム

1. 中心座標は，サポート値の重みを付けた 2 中心座標の重心点
2. 半径は，更新後の中心座標から 2 つの $gwo$ の和集合の中の最遠点までの距離
3. サポート値は 2 つの $gwo$ のサポート値の和
4. 便宜上ジオワードはサポート値の大きい $gwo$ のジオワードを使う

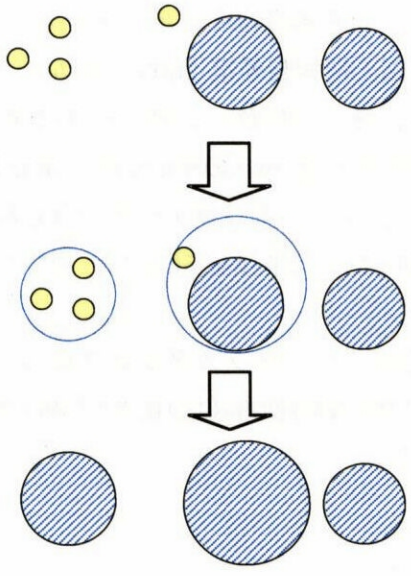


図 5.7 クラスタリングの概念図

この手法の基本的考え方は，最長距離法による階層型クラスタリングである．一般にこのクラスタリング方法は，最小距離法のような局所的な弱さ（細長い連結空間ができる）はないが，一方で同一のクラスタに入るべきであるが，偶然離れてしまったクラスタが，

なかなか合併できないという欠点がある．このような場合は，結果が不安定になるが，本手法は，クラスタ（gwo）のサポート値を段階的に上げながら処理を行っているので，gwoの地理的大きさ（半径）とサポート値にある程度関連があれば，合併は半径の小さいものから始まり，逆に半径の大きいもの同士が早めに合併してしまうことがないと考えられる．

以上の操作で，gwoをノードとするツリーが作成される．リーフノードはクラスタリング開始時のgwo，中間ノードはgwoが合併して作成されたものである．上記の作成アルゴリズムでは合併のルーチンが繰り返されるが，rootから同一の階層（n階層目）に属するgwoは，同じ回のルーチンで出力されている．従って，各階層に属するgwoのサポート値には下限が存在して，その値が各ルーチンで設定された最小サポート値になる．このサポート値をクラスタ階層最小サポート値（CRMS: Cluster Rank Minimum Support）と呼ぶことにする．

### 5.5.2　クラスタ作成実験

実験に用いるgwoは，先にクリーニングを行った9,000万検索から取り出したものである93,515種のジオワードを用いた．このgwoデータに対して外部データを使って，ジオワードに空間図形情報，すなわち緯度経度と半径を付与した．使ったデータは街区レベルの位置参照情報であるが[国土情報整備室 2003]，細かい地域では丁目番地の点データの緯度経度が定義されている．そこで任意のジオワードに対して，該当する住所の点データをあつめ，その重心（単純平均）をgwoの中心点に，求めた重心から各点までの距離の標準偏差をgwoの半径とした．また詳細な点データが得られず，（市町村の）重心点座標しか入手ができない市町村に関しては半径を5kmに，同様に駅に対しては半径を2kmに一律に設定した．

上記の93,515個のgwoを使って，クラスタリングを行った．その結果を表5.10に示す．与えたサポート値（CRMS）は1E06（$1x10^{-6}$）を初期値とし，最終的には1まで段階的に変化させた．

表 5.10 ジオワードクラスタ作成結果

| 最小サポート値 (CRMS) | クラスタ数 | 最小サポート (CRMS) | クラスタ数 |
|---|---|---|---|
| なし | 91,389 | 2E04 | 1,217 |
| 1E06 | 70,402 | 3E04 | 881 |
| 2E06 | 25,286 | 5E04 | 626 |
| 3E06 | 18,463 | 1E03 | 399 |
| 5E06 | 13,714 | 2E03 | 244 |
| 1E05 | 9,401 | 3E03 | 162 |
| 2E05 | 5,968 | 5E03 | 109 |
| 3E05 | 4,261 | 1E02 | 67 |
| 5E05 | 3,032 | 1E01 | 10 |
| 1E04 | 1,955 | 1 | 7 |

さらに，図5.8は，作成されたクラスタの一部を可視化したものである．図5.8のパラメータ類は以下の通りである．

START: 開始状態 (gwo 数 (n)=91389)
1E05: 最小サポート (CRMS) = $1x10^{-5}$ (n = 9401)
1E04: CRMS = $1x10^{-4}$ (n = 1995)
1E03: CRMS = $1x10^{-3}$ (n = 399)
3E03: CRMS = $3x10^{-3}$ (n = 162)
1E02: CRMS = $1x10^{-2}$ (n = 67)
円は gwo を，円の半径は gwo のサポート値 (相対値) を示す．

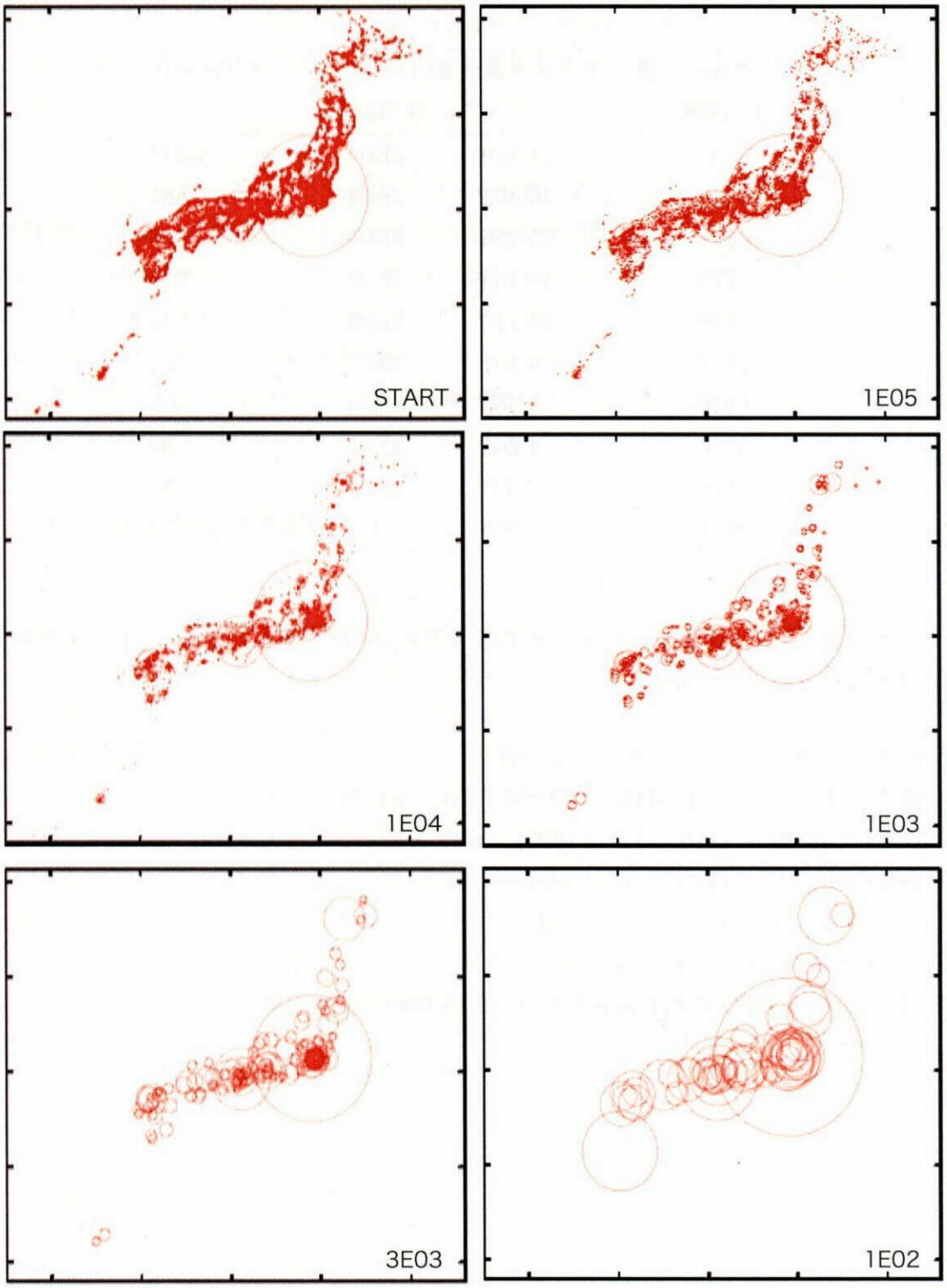



図 5.8　作成されたジオワードのクラスタの可視化例

図5.9はgwo「東京都港区新橋」関連のgwoを示す．図から新橋に合併されるgwoは33種ある．

クラスタリングの各段階で指定するクラスタ階層最小サポート値(CRMS)を1E06に設定すると，「霞ケ関駅（東京都千代田区）」が「東京都千代田区内幸町」に，「桜田門（東京都千代田区）」が「東京都千代田区霞が関」に合併される．CRMSを3E05に設定すると，新橋に合併されるgwoは7つになり，「東京都港区芝浦」(サポート値0.00034)，「新橋駅（東京都港区）」(0.00037)，「東京都港区虎ノ門」(0.00043)，「東京都港区新橋」(0.00046)，「東京都中央区築地」(0.00051)，「銀座駅（東京都中央区）」(0.00066)，「東京都中央区銀座」(0.00142)のgwoにまとめられる．「中央区築地」のクラスタは元々は0.0002程度のサポートしか持たないが，「築地駅」，「勝どき」，「豊海町」，「浜離宮庭園」，「月島」，「月島駅」，「晴海」，「東雲」，「辰巳駅」を加えた10のgwoの和集合として倍以上の0.005のサポートを持つgwoを形成することができる．

このように本クラスタリング法を使うと，検索頻度の最小値を保証した形で，ジオワードをクラスタに分けることができる．このクラスタは住所階層を用いていないので，住所階層より細かい粒度でクラスタのサイズを調節することができ，また近接していても都道府県が違うので同一のクラスタを形成しないということがない．また，異なるレベルの住所同士や，非住所ジオワードも同一の枠組みで扱うことができる．

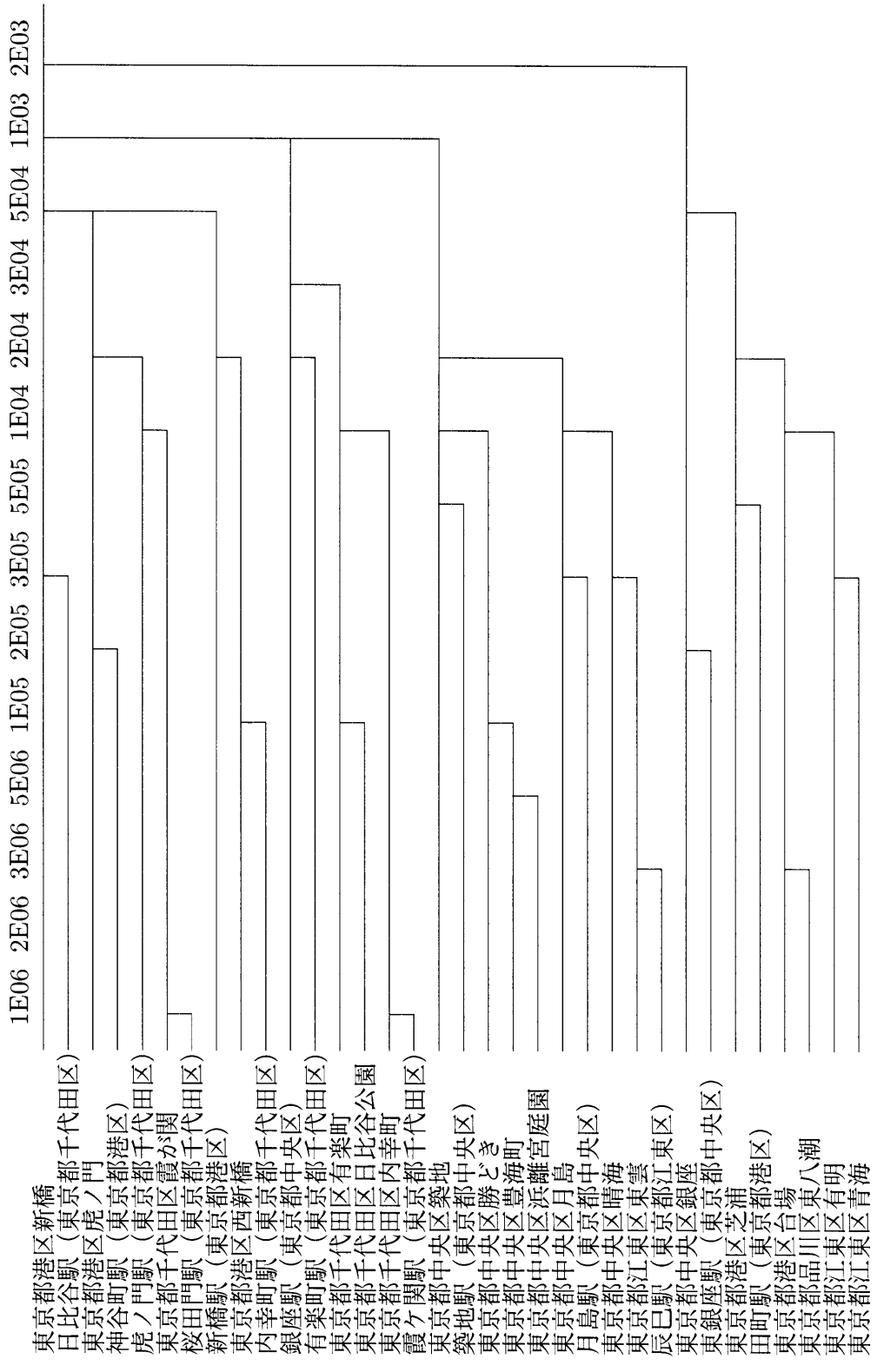



図 5.9　作成クラスタの例（デンドログラム）

## 5.6 頻度に応じたクラスタリング手法を用いた一般化相関ルール

前節で述べた通り，リクエスト回数が多くない地域でも近隣のジオワードをまとめることにより，より大きい，すなわちリクエスト回数が多いジオワードオブジェクト（GWO）を作ることができる．GWO を統合して，より大きい GWO を作ることを一般化と呼ぶ．GWO を排他的に一般化する方法には，住所階層を使った一般化がある（例えば市町村を県単位にそろえる）．作成したクラスタは任意の GWO に対して，あるサポート値を満たす GWO への対応を与えるので，GWO の一般化が可能である．ここでは住所階層を使った一般化とクラスタを使った一般化により，得られる相関ルールにどのような影響が出るかを調べる．

相関ルールの作成結果を，作成されたルールの数と作成できた地域の数で評価する．一般に出力された相関ルールには最小サポート値が存在して，それを満たさないルールは採用されない．地理的な相関ルールにおいて，地域を排他的に取る限り，ルール数とルールが作成できる地域数はトレードオフの関係にある（地域を「全国」とすれば各ルールのサポート値は高くなる，また地域を小分割して地域数を増やせば増やすほど，最小サポート値を満たすルールの数は少なくなる）．ここで分析の詳細に入る前に，本実験に現れたジオワード数を示す．左から解析に使った全ジオワード数，全解析ログデータの中から見つかったジオワード数，ある最小サポート値 S を満たすルールを作成できたジオワード（ラージジオワード (S) と呼ぶ）の数を示す．なお，辞書中のジオワード数は市区郡町村レベルまではほぼ網羅的であるが，それ以下はジオコーディング用データの存在／入手の困難性のため，必ずしも網羅的ではない．またログのジオワード，およびマイニングを行う対象のジオワードは，最小レベルを町大字に統一した．このようにログ中には 8 万余のジオワードが存在するがラージジオワード (1E06) は 3,401 に限られる．

表 5.13 にジオワードの一般化手法ごとのルール数を示す．一般化手法としては従来方式（住所階層利用）と提案方式（クラスタ）に大きくわけ，それぞれの一般化粒度を変えて，試している．得られたルール数であるが，最小サポート 1E06 を得たい場合，いずれのケースにおいても 20 万種程度のルールが得られるが，実際に Specialty に分類できるものはその 1/5 程度であり，住所方式で「正規化（町大字）」の 38830 件のところクラスタリング方式では C3E04 の 54252 が最大で約 1.4 倍のルールを与えた．また最小サポートを 1E05 に高くすると，住所方式では 813 ルールしか取り出せないところ，クラスタ方式では最大で約 1.7 倍となる 1358 ルールを取り出している．

次に最小サポート値を満たすルールが得られるラージジオワード数を分析する．同様に

表 5.11　ジオワードの数

ラージジオワード（1E06）とは
最小サポート値 1E06 を満たす相関ルールが存在するジオワードのこととする

| 種別 | 辞書 | 全ログ | ラージジオワード数 (1E06) |
|---|---|---|---|
| 都道府県 | 47 | 47 | 47 |
| 市区郡町村 | 3,917 | 3,887 | 1,694 |
| 町大字 | 208,447 | 73,098 | 1,319 |
| 駅 | 8,789 | 3,817 | 341 |
| 合計 | 221,200 | 80,849 | 3,401 |

表 5.12　ジオワード一般化手法ごとのルール数

手法の C3E05 は最小サポート値を $3x10^{-5}$ としたクラスタを使ったことを意味する.

| 一般化方法 | 最小サポート値 1E06 のルール数 | | | 最小サポート値 1E05 ルールの数 | | |
|---|---|---|---|---|---|---|
| | 全て | Specialty | Locality | 全て | Specialty | Locality |
| 無変換 | 177,971 | 42,026 | 31,870 | 3,767 | 641 | 216 |
| 正規化（町大字） | 192,827 | 38,830 | 38,731 | 5,355 | 812 | 305 |
| 市区町村に統一 | 198,673 | 30,394 | 47,128 | 7,082 | 888 | 374 |
| 都道府県に統一 | 158,064 | 6,920 | 72,887 | 18,684 | 822 | 2,014 |
| C1E06 | 192,955 | 38,932 | 38,730 | 5,355 | 813 | 305 |
| C3E06 | 193,862 | 39,743 | 38,743 | 5,355 | 813 | 305 |
| C1E05 | 196,284 | 41,688 | 38,790 | 5,362 | 815 | 305 |
| C3E05 | 201,682 | 44,933 | 38,969 | 5,417 | 835 | 305 |
| C1E04 | 221,050 | 53,443 | 39,515 | 5,672 | 946 | 307 |
| C3E04 | 246,373 | 54,252 | 41,843 | 6,469 | 1,143 | 311 |
| C1E03 | 256,913 | 37,997 | 48,349 | 8,736 | 1,358 | 336 |
| C1E02 | 203,967 | 7,397 | 71,662 | 18,712 | 790 | 577 |

表 5.13 ジオワードの一般化手法ごとのラージジオワード数

| GWO の一般化方法 | ラージジオワード (1E06) の数 | | | ラージジオワード (1E05) の数 | | |
|---|---|---|---|---|---|---|
| | 全て | Specailty | Locality | 全て | Specailty | Locality |
| 無変換 4735 | 4225 | 1797 | 323 | 199 | 73 | |
| 正規化 (町大字) | 3401 | 3133 | 1384 | 343 | 236 | 87 |
| 市区町村に統一 | 2340 | 2152 | 1076 | 294 | 200 | 88 |
| 都道府県に統一 | 47 | 47 | 47 | 47 | 47 | 47 |
| C1E06 | 3507 | 3243 | 1380 | 343 | 236 | 87 |
| C3E06 | 4171 | 3906 | 1387 | 343 | 236 | 87 |
| C1E05 | 4571 | 4351 | 1397 | 348 | 239 | 87 |
| C3E05 | 3673 | 3506 | 1402 | 373 | 259 | 87 |
| C1E04 | 1795 | 1771 | 1284 | 459 | 327 | 89 |
| C3E04 | 854 | 854 | 836 | 550 | 377 | 89 |
| C1E03 | 358 | 358 | 358 | 358 | 272 | 100 |
| C2E02 | 56 | 56 | 56 | 56 | 56 | 55 |

住所方式では最小サポートが 1E06 の時は 3133 件のところ，クラスタリングでは約 1.4 倍の 4351 件，最小サポートを 1E05 にすると，それぞれ 236，377 件で，クラスタリング手法が約 1.6 倍多くの Specialty を出力した．

## 5.7 クラスタによる一般化相関ルール作成の評価

解析により特徴のあると考えられる下記の空間を比較して分析する

- S1 ベースライン《正規化 (町大字)》
  Specialty ルール数 38830，ラージジオワード数 3133
- S2 比較 1，小さいクラスタ《C1E05》
  Specialty ルール数 41688，ラージジオワード数 4351
- S3 比較 2，大きいクラスタ《C3E04》
  Specialty ルール数 54252，ラージジオワード数 854

ただし上記の Specialty ルールとラージジオワードは，最小サポート値 1E06 に対してのもの．

- 比較 1 小さいクラスタを作ったケース (S2)

クラスタリングで得られたS2は，S1の約1.4倍の地域でルールが作成可能になる．S1になくて，S2に存在するジオワードには次のようなものがある．

千葉県勝浦市勝浦→民宿

神奈川県足柄下郡箱根町湯本→温泉

このクラスタは市区町村より下の複数の町大字が合併してできている．クラスタへのリクエスト回数は最小サポートを満たしているが，そこから新たに発見できる知見はあまり多いとは言えない．

- 比較2 大きいクラスタを作ったケース (S3)

  ルールの例：愛媛県宇和島市→鮮魚店

  このケースでは，クラスタリングが進み，クラスタ数は《市区町村》よりも少ないが，より多くのルールを出力できている．このことは，一つのジオワードに対応するルールが多いということを示している．以下に示したのは，S1とS3の差分の例である．

  – 大きいクラスタではじめて見いだせたルールの事象

    役場 官公庁 喫煙具 製材業 小売業 農協 贈答品店 料理仕出し日用品雑貨店 コンビニ 調味料 鮮魚店 敷物 民宿 暖房装置理容店 石工品 木製品 牛乳 かわら（S3に存在して，S1に存在しなかったルールの右辺から上位20件）

  – 大きいクラスタには残らなかった，あるいはSpecialtyではなくなったルールの事象

    居酒屋 ホテル 飲食店 ビジネスホテル 美容院歯科 病院 旅館 皮膚科 眼科（S1に存在して，S3に存在しなかったルールの右辺から上位10件）

このようにルールを増やしているものは，頻繁ではないが必需なサービス，あるいは産業類である．これらはジオワードがクラスタリングされた結果，ルールとして見いだすことができた．一方ルールが減じた側の事象は，極めて一般的な事象である．これらはより細かい地域単位でも相関ルールとして最小サポートを満たしていたためであり，地域数が減ることによりルール数も減ることとなった．

## 5.8　作成されたジオワード相関ルールの考察

表5.15から表5.19に作成されたジオワード相関ルールの考察の例を示す．それぞれの表はジオワード（青森県，東京都中央区銀座など）に対して本章で述べた3つの方法で一般化相関ルールを作成し（青森県→ホテル），それをルール3つの興味深さの指標で分類し，都合9つのセグメントにわけて表示したものである．

各セグメント中のルールはルールのサポート値が上位のもの5つを表示した．ルールの最小サポート値は1E06であり，x印は最小サポート値を満たすルールが見いだせなかった項目である．

**ジオワードの選び方**

ジオワードの出現頻度（サポート値）を本研究で用いたログデータにおける分布を調べて表5.14に示す．表のようにジオワードの分布には極めて偏りがあり，例えば頻出上位の164種のジオワードで全体の約50%の検索が行われている．ルールを作成する実験においては様々なサポート値を持つジオワードに対する一般化の影響を調べるために，表の各行に相当するサポート値を持つジオワードを取り出して相関ルールを作成した．今回例としてあげたジオワードは青森県と東京都中央区銀座がサポート値1未満0.001以上のグループから，さらに以下順に，長崎県大村市，神奈川県鎌倉市大船がグループ2からと各グループから2ジオワードの例を掲示している．

**ジオワードの一般化方法**

一般化相関ルールは，本章の提案に従い，一般化を行わないものと，ジオワードを住所階層を利用して一般化するもの（代表として市区町村レベルの一般化を），ジオワードを提案のクラスタを利用して一般化するもの（代表としてC3E04を）用いた．

**ルールの興味深さの分類**

全体集合（すべて）とSpecialtyとLocalityの3つの集合にわけて掲示した．なお，Conmodity相当のルールはおおむね「すべて」の集合と近いので省略している．

**考察**

まず，一般化の効果を考察する．グループ3以降のジオワードでは，一般化を行わない場合，最小サポートを満たすルールを発見することができなかった．このため，一般化を行うことによってルールが作成できる最低限のメリットがあきらかになる．

次に一般化手法を住所階層方式とクラスタ方式を比べてみよう．例えば，神奈川県鎌倉市大船において，SpecialtyとLocalityルールには違いがあることがわかる．住所階層方式では高級住宅地，もしくは寺院を中心とした観光地である鎌倉市の性質が現れているが，クラスタ方式では鎌倉全体の影響を受けず，下町の様子が現れている（Localityに現れるルールは駅と飲食店である）．

また徳島県那賀郡木沢村では，市区町村レベルでも最小サポート値を満たすルールが出てこない．このため次の一般化を行うと都道府県レベルにまで持ち上がることになる．逆にクラスタリング法では適切な範囲でクラスタを形成してルールが計算できる状況になっ

ている（木沢村は近隣の町村と合わせて，徳島市の一部とクラスタを形成した）．

また京都府京都市中京区畳屋町などサポート値が小さい地域では，市町村レベルの一般化とクラスタがほぼ同等になるケースも見られた．

本研究では全体の多様な地域でより多くの興味深いルールの抽出が取り出せるクラスタとしてCeE04を，またSpecailtyを与える特化係数の値を3以上と定めたが，これらは対象によってより効果を高められる値が存在すると考えられる．このことは生活圏，文化圏として連続性のある適切な地域を探し出すことと同等と考えられる．

また，今回はクラスタを見つけるためのジオワード間の距離を，地理的な距離で定義したが，ログのセッションにおける距離を定義することも可能である．図5.10はジオワード間の相関ルールでサポート値の高いものを可視化している．各点はジオワード（ジオワードはC3E04のクラスタに一般化），線分はジオワード間の相関ルールを示し，太い程サポート値が高い．

ジオワード間の相関ルールは利用者が同じ検索セッションの中で，ジオワードの置き換えを行った結果が反映されている．このことは，検索をを行う上でのジオワード間の距離を示していると考えられる．日常生活の買い物を考えると徒歩圏内で入手ができなければ，車，あるいは鉄道の利用を考慮する．このようにジオワード間の距離は直線距離以外の要素でも定義できるはずである．インターネットにおける情報検索においては，より地域の再選択の自由度が増すため，インターネットにおける地理を表現できる距離体系が必要であり，これは今後の研究の課題である．

表 5.14 ジオワードの出現頻度ごとの分布

| サポート値 | | ジオワード数 | サポート計 (%) |
|---|---|---|---|
| (1, | 0.001] | 164 | 49.5 |
| (0.001, | 0.0001] | 882 | 25.7 |
| (0.0001, | 0.00001] | 5,811 | 15.9 |
| (0.00001, | 0.000001] | 37,011 | 8.5 |
| (0.000001, | 0) | 36,981 | 0.4 |

表 5.15 作成されるルールの例 1

| | 青森県 | | | 東京都中央区銀座 | | |
|---|---|---|---|---|---|---|
| | 一般化手法 | | | 一般化手法 | | |
| | なし | 市町村 | C3E04 | なし | 市町村 | C3E04 |
| すべて | ホテル | ホテル | ホテル | 美容院 | ホテル | ホテル |
| | 病院 | 病院 | 病院 | 飲食店 | 飲食店 | 飲食店 |
| | 飲食店 | 飲食店 | 飲食店 | ホテル | 金券ショップ | 美容院 |
| | 居酒屋 | 居酒屋 | 居酒屋 | 書店 | 歯科 | 歯科 |
| | 温泉 | 温泉 | 温泉 | レストラン | 居酒屋 | 居酒屋 |
| Specialty | 温泉 | 温泉 | 温泉 | 書店 | 三越 | 書店 |
| | セメント | セメント | セメント | レストラン | 広告代理業 | レストラン |
| | 肥料 | 肥料 | 肥料 | 広告代理業 | デパート | 広告代理業 |
| | 民宿 | 民宿 | 民宿 | 三越 | 三井住友銀行 | クリーニング |
| | 高校 | 高校 | 高校 | エルメス | みずほ銀行 | カラオケ |
| Locality | 青森銀行 | 青森銀行 | 青森銀行 | 伊東屋 | 伊東屋 | 伊東屋 |
| | 青森県庁 | 青森県庁 | 青森県庁 | 博品館 | 博品館 | 博品館 |
| | 青森駅 | 青森駅 | 青森駅 | 福家書店 | 東京駅 | 福家書店 |
| | 東北電力 | 東北電力 | 東北電力 | バードランド | 八重洲ブックセンター | バードランド |
| | 八戸駅 | 八戸駅 | 八戸駅 | カルティエ | ざくろ | カルティエ |

表 5.16 作成されるルールの例 2

| | 長崎県大村市 | | | 神奈川県鎌倉市大船 | | |
|---|---|---|---|---|---|---|
| | 一般化手法 | | | 一般化手法 | | |
| | なし | 市町村 | C3E04 | なし | 市町村 | C3E04 |
| すべて | 病院 | 病院 | 病院 | 居酒屋 | ホテル | 居酒屋 |
| | ホテル | ホテル | ホテル | ホテル | 居酒屋 | 歯科 |
| | 居酒屋 | 飲食店 | 工芸品 | ビジネスホテル | 病院 | 飲食店 |
| | 飲食店 | 保育園 | 陶磁器製造 | 美容院 | 飲食店 | ホテル |
| | 保育園 | 居酒屋 | 民芸品 | 飲食店 | ビジネスホテル | 病院 |
| Specialty | 病院 | 保育園 | 工芸品 | 居酒屋 | 写真館 | 居酒屋 |
| | ホテル | 消防機関 | 陶磁器製造 | ホテル | 動物病院 | 歯科 |
| | 居酒屋 | 市区町村機関 | 民芸品 | ビジネスホテル | 寺院 | 飲食店 |
| | 飲食店 | 国立病院 | 旅館 | 美容院 | 神社 | ビジネスホテル |
| | 保育園 | 駐車場 | 保育園 | 歯科 | 三菱電機 | 不動産取引 |
| Locality | 大村市役所 | 大村市役所 | 在宅介護支援センター | 大船駅 | 鎌倉市役所 | きじま |
| | 鳥千代 | 九州教具 | 大村市役所 | きじま | 大船駅 | 大船駅 |
| | 親和銀行 | 鳥千代 | 親和銀行 | 鳥恵 | 珊瑚礁 | 鳥恵 |
| | 大村郵便局 | 親和銀行 | 大正屋 | x | 豊島屋 | x |
| | 長崎空港 | 大村郵便局 | 九州電力 | x | 鳥恵 | x |

表 5.17　作成されるルールの例 3

| | 天王寺駅（大阪府大阪市天王寺区） | | | 東京都大田区東矢口 | | |
|---|---|---|---|---|---|---|
| | 一般化手法 | | | 一般化手法 | | |
| | なし | 市町村 | C3E04 | なし | 市町村 | C3E04 |
| すべて | x | ホテル | 居酒屋 | x | 病院 | 病院 |
| | x | 病院 | 飲食店 | x | ホテル | 歯科 |
| | x | 飲食店 | ホテル | x | 歯科 | ホテル |
| | x | 居酒屋 | 美容院 | x | 美容院 | タクシー |
| | x | 郵便局 | 産婦人科 | x | 飲食店 | 美容院 |
| Specialty | x | 三井住友銀行 | 産婦人科 | x | フラスチック加工 | ホームセンター |
| | x | ヨドバシカメラ | イズミヤ | x | ゴルフ練習場 | フラスチック加工 |
| | x | ソフマップ | 金券ショップ | x | 金型 | ゴルフ練習場 |
| | x | コーナン | 喫茶店 | x | 図書館 | 金型 |
| | x | 関西電力 | ホームセンター | x | ダイソー | 図書館 |
| Locality | x | 大阪中央郵便局 | ホームズ | x | 大田区役所 | 大田区役所 |
| | x | 大阪市役所 | 西成区役所 | x | 羽田空港 | 羽田空港 |
| | x | 大阪駅 | 近鉄百貨店 | x | 歓迎 | ダイシン百貨店 |
| | x | 新大阪駅 | くるま | x | ダイシン百貨店 | 歓迎 |
| | x | 近鉄百貨店 | x | x | 機材 | 機材 |

表 5.18　作成されるルールの例 4

| | 徳島県那賀郡木沢村 | | | 愛知県豊田市住吉町 | | |
|---|---|---|---|---|---|---|
| | 一般化手法 | | | 一般化手法 | | |
| | なし | 市町村 | C3E04 | なし | 市町村 | C3E04 |
| すべて | x | x | 居酒屋 | x | 居酒屋 | 飲食店 |
| | x | x | ホテル | x | 飲食店 | 病院 |
| | x | x | 飲食店 | x | ホテル | 美容院 |
| | x | x | スナック | x | トヨタ自動車 | ホテル |
| | x | x | 病院 | x | ビジネスホテル | 居酒屋 |
| Specialty | x | x | 居酒屋 | x | トヨタ自動車 | ラーメン店 |
| | x | x | スナック | x | トヨタ | ジャスコ |
| | x | x | 理容店 | x | 自動車部品・用品製造 | 北海道 |
| | x | x | 小売業 | x | タイヤ・チューブ製造業 | 映画館 |
| | x | x | ガソリンスタンド | x | 神戸屋 | ヤマダ電機 |
| Locality | x | x | 阿波銀行 | x | 禅 | 愛知学院大学 |
| | x | x | x | x | 豊田市役所 | 日進市役所 |
| | x | x | x | x | メグリア | 三好町役場 |
| | x | x | x | x | アラコ | アイム |
| | x | x | x | x | さくらダイニング | 赤池タイヤ |

表 5.19　作成されるルールの例 5

| | 北海道帯広市 | | | 京都府京都市中京区骨屋町 | | |
|---|---|---|---|---|---|---|
| | 一般化手法 | | | 一般化手法 | | |
| | なし | 市町村 | C3E04 | なし | 市町村 | C3E04 |
| すべて | x | ホテル | ホテル | x | ホテル | 京都大学 |
| | x | 薬局 | 薬局 | x | 飲食店 | 京都中央信用金庫 |
| | x | 居酒屋 | 病院 | x | 病院 | 京都銀行 |
| | x | ビジネスホテル | ビジネスホテル | x | 居酒屋 | 京都市役所 |
| | x | 病院 | 居酒屋 | x | 染色工業 | 京都駅 |
| Specialty | x | ホテル | 薬局 | x | ホテル | 染色工業 |
| | x | 薬局 | 生コンクリート | x | 染色工業 | 神社 |
| | x | ビジネスホテル | 肥料 | x | 飲食店 | ソフマップ |
| | x | 古本 | ケーキ店 | x | 病院 | 呉服卸 |
| | x | 下宿 | 燃料店 | x | 居酒屋 | 仏壇・仏具 |
| Locality | x | 帯広市役所 | 帯広市役所 | x | 京都大学 | 京都大学 |
| | x | 六花亭 | 六花亭 | x | 京都中央信用金庫 | 京都中央信用金庫 |
| | x | クランベリー | クランベリー | x | 京都銀行 | 京都市役所 |
| | x | 北海道ホテル | 北海道ホテル | x | 京都市役所 | 京都駅 |
| | x | ポスフール | ポスフール | x | 京都駅 | 京都銀行 |

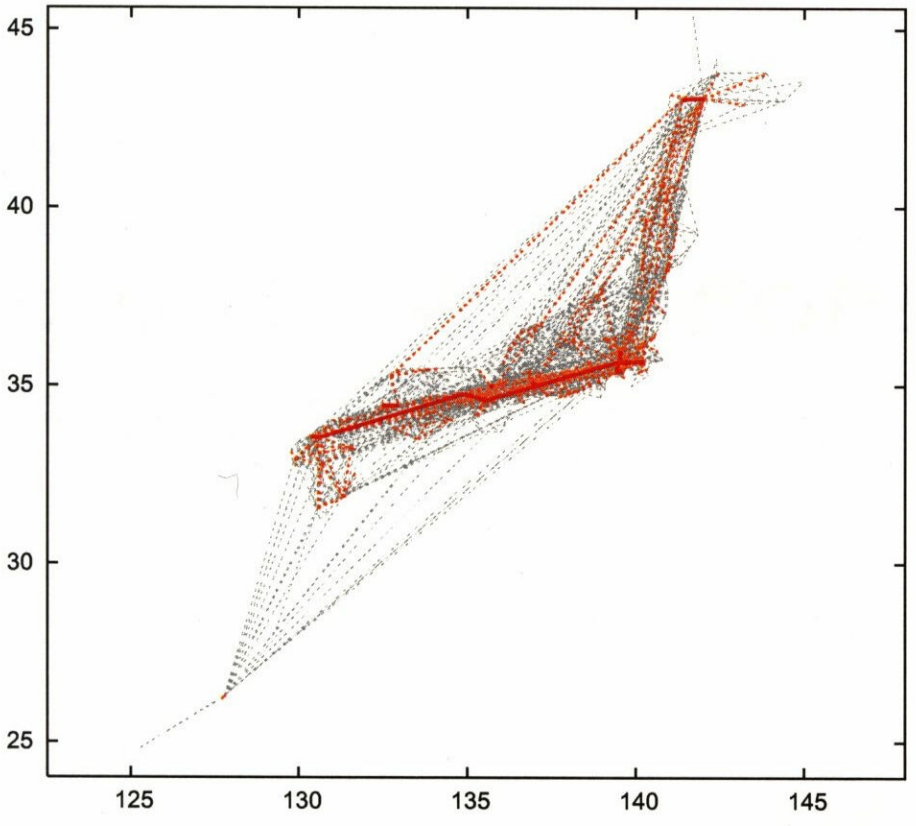



図 5.10 ログセッションにおけるジオワードのクラスタ間のつながり

# 第 6 章

# ジオワード・マイニングのための固有名詞解析手法

## 6.1 セッション情報を用いた省略のあるジオワードのあいまい性解消法

ローカルサーチシステムは，ユーザが住所やランドマークなどのジオワードを入力して，情報を検索するシステムである．ジオコーディングとは位置に関連したオブジェクトに緯度経度等の地理情報を付与することで，ここでは文字列であるジオワードに緯度経度を付与することを行う．基本的なジオコーディングのプロセスは以下の通りである．

1. ジオワードをジオワード正規形に変換する
2. ジオワード正規形の緯度経度をジオワードに付与する

ジオワードはユーザに手入力されることがあり，必ずしも正しい表記をされるとは限らない．ここで「正しい」表記を正規形と呼ぶことにする．一般に住所等ジオワードは複数の表記のバリエーションが慣例として流通していて，どの表記が正しいかを決めることは容易ではない場合もあるが，本章では JIS で定められた住所表記などを正規データとする．本章の課題は，入力されたクエリの中のジオワードをジオワード正規形に変換することである．

### 6.1.1 住所表記のゆれ

相良の研究によれば，日本語住所のゆれは省略，アラビア数字の代用，異体字等の文字の差異があると報告されている．

この報告の中で省略された住所を高速にジオコーディングする方法が評価されている

表 6.1 住所表記のゆれの種類と出現確率 [相良 2003] より

| ゆれの種類 | 含まれる確率 |
|---|---|
| 上位住所要素の省略 | 11.3% |
| 詳細地名の略記 | 88.2% |
| アラビア数字による代用 | 92.0% |
| 異体字の使用 | 0.1% |
| その他 | 0.1% |

が，本質的に情報量が不足していれば正規形を特定できない．すなわち，サーチエンジンリクエストに気軽に入力される文字列は，省略が多く，その省略による住所のあいまい性を解消するには，何らかの他の情報でその省略住所の帰属を推定する必要がある．下記はその一例である．府中は有名な例で，全国の市で唯一異なった都道府県同士で名称の衝突が起きている

- 府中 → 東京都府中市？広島県府中市？
- 上野 → 三重県上野市？上野駅？
- 中野 → 長野県中野市？東京都中野区？
- 福岡 → 福岡県？宮城県仙台市泉区福岡？
- 港区 → 東京都港区？名古屋市港区？大阪市港区？

### 6.1.2 不完全住所の上位住所補完

セッション情報を利用して，あいまいな不完全住所を補完する．例えば，中央区は複数の政令指定都市にあり，それだけでは上位の都道府県を判定することができない，しかし同じ検索セッションで東京都がリクエストされていたら，中央区も東京都とみなすことができると仮定する．

アルゴリズム
あるジオワードが
R1 if 自明住所である（正式住所候補がひとつしかない） then 正式住所確定□
R2 if 同一セッション内に自明住所が存在し上位住所が同じである
if 一つの自明住所と上位住所が同じ then 正式住所確定□
else（二つ以上の自明住所と上位住所が同じ） then あいまい性は解消できない□
R3 if 同一セッション内に上位住所が同じであるジオワードが存在する
if 一つと上位住所が同じ then 正式住所確定□
else あいまい性は解消しない□

### 6.1.3 上位住所補完実験

#### スタンドアロン法

従来の手法を用いて，ログデータのジオコーディングを行う．基本的な住所照合の考え方は以下の通りである．

入力ジオワードを単独で正規データ辞書と照合し，正規住所を見つける．照合する際の照合規則としては以下の自明と判断する規則用いる．複数の規則に合致する場合は，若いルールを優先して適用する．但し県間でルール該当衝突が起きる場合は表欄外の細則（※甲）に従う．

ジオワードを自明住所と判断する規則
E1 住所の完全表記とただ一つ一致する（東京都文京区本郷）
E2 住所語尾を含む都道府県，市区町村とただ一つ一致する（横浜市）
E3 上位住所を省略した町大字の表記とただ一つ一致する（文京区本郷）
E4 都道府県と一致しかつ市区郡町村語幹とただ一つ一致する（神奈川県横浜）
E5 都道府県と一致し，かつ町大字とただ一つ一致する（東京都銀座）
E6 都道府県語幹とただ一つ一致する（千葉）
E7 市区郡町村語幹とただ一つ一致する（横浜）
E8 駅名語幹とただ一つ一致する（原宿）
E9 町大字名とただ一つ一致する（本郷）

スタンドアロン法で実験に用いたデータを以下の表に示す．ジオワードは全体のリクエストログ全体から，最も頻繁にリクエストされる上位 10,000 種のジオワードを用いた．全体の 8,600 万クエリには約 130 万のジオワードが存在するが，このうち上位 1 万のジオワードは全体のリクエスト総数の 79.3% を占めている．

表 6.2 実験データの定義

| 解析対象のクエリ数 | ジオワードの種類 | ジオワード上位 10,000 種 |
|---|---|---|
| 86,522,189 ヒット | 1,286,634 (100%) | 10,000 (79.3%) |

実験 1 のスタンドアロン法での正規形とのマッチングの結果を表に示す．その結果，何もせずにコーディングができたジオワードが約 70% あり (E1 に該当)，残りの約 20% も上記の自明規則で一意に住所を絞り込むことができた（※ 1，E2〜E9 が単独で合致）．また一意にしぼれなくても，都道府県を優先させるなどの処理を行って，解決できたものが約 4% 存在した（※ 2，※甲）．また従来から指摘されている表記のゆらぎ等の原因によるものも合計 3% 程度存在することがわかった．そこで残ったのが部分文字列であいまい性が解決できない 328 種のジオワードである．数にして 3% は多くはないが，これが解決されない場合は，そのジオワードが示す地域の検索ができなかったり，解析ができなかったりと問題が想定される．しかしユーザはなんらかの解決を図っているはずであり，その根拠はログデータにも残されているはずである．

**セッション法**

データの一部 3,303,406 セッションを解析した結果，従来手法で解析不可能であった 328 単語のいずれかが用いられたセッションが全体の約 1% 37,643 存在した．このセッションに対してアプローチに記したセッション法でジオコーディングを行った．その結果，約 92% に相当する 34,776 セッションで解決できた．解決できた例を示す．

解析できた例

- 銀座，築地
- 銀座→東京都中央区銀座？栃木県鹿沼市銀座？・・・
- 築地→東京都中央区築地？岩手県宮古市築地？・・・
- 結果： 東京都中央区銀座，東京都中央区築地
- 府中市，小金井
- 府中市→東京都府中市？広島県府中市？
- 小金井→東京都小金井市？小金井駅（栃木県）？・・・
- 結果： 東京都府中市，東京都小金井市

日本語ジオワードの表記のバリエーションに従って，サーチリクエストに含まれるジオワードの表記方法を分類した．その結果正しい住所表記が 7 割，衝突なく解決できる省略表記が 2 割，文字表記のゆれが約 9% 存在するなどの入力の傾向が明らかになった．その

表 6.3 解析結果

| ジオワード解析結果 | ジオワード数 |
|---|---|
| 完全住所表記 | 6,967 |
| 部分住所表記（衝突なし）※1 | 2,023 |
| 部分住所表記（衝突あり，解決可能）※2 | 427 |
| 部分住所表記（衝突あり，解決不能）※3 | 328 |
| 文字表記のゆらぎ ※4 | 92 |
| 大町 ※5 | 62 |
| 俗地名・自然地名 ※6 | 49 |
| その他 ※7 | 52 |
| 合計 | 10,000 |

※1 上位住所の省略，例：京都市（1,585）
中間住所の省略，例：東京都銀座（271）
住所接尾語の省略，例：愛媛（134），その他（33）

※2 埼玉（埼玉県？栃木県黒磯市埼玉？）

※3 府中市（東京都府中市？広島県府中市？），港区

※4 諫早市（諫早が正しい），横浜市保土ヶ谷（同，保土ケ谷）

※5 東京都千代田区神田（神田○○町の集まりが神田）

※6 東京２３区，大阪市内，淡路島

※7 埼玉県浦和市（市町村合併で現存せず）
大阪府伊丹市（兵庫県伊丹市が正しい）

※甲 同一都道府県内で省略ジオワードの衝突細則
都道府県，市区郡町村，駅，町大字の優先順位と仮定する
例：千葉：千葉県？千葉市？千葉駅？→千葉県
他県間で衝突ジオワードの衝突
都道府県 対 町村，都道府県 対 町大字，市 対 町大字
の関係時のみ左項を優先する

上で，従来の手法ではスタンドアロンで解決できなかった「中央区」，「中野」などのジオワードが頻出ジオワード 10,000 種のうちの約 3%，全体のセッションの中では約 1.1% 存在していたが，セッション情報を用いて上位住所を補完することで正規化をできるようにした．この技術を用いて 5 章ではジオワード・マイニングを行っている．

## 6.2 人名のかな表記のゆれに基づく近似文字列照合法

日本人名のかな表記にゆれとよばれる変形が存在し，日本語情報検索システムの問題となっている．本節では人名のかな表記にゆれが存在してももれのない検索を可能とする近似文字列照合法を提案する．ゆれの問題に対処するためには表記を統一して検索を行なうことが一般的であるが，現在かな表記を統一する基準は明らかではなく，そのため統一すべきゆれが多種になった場合の対策も明らかになっていない．本文では日本人名約 3,000 万件を解析し，姓のゆれのデータを収集分析する．その結果，姓は 9 万種の姓のゆれ単位に分類できること，実データ上で 58% の姓に何らかのゆれが存在すること，ゆれの原因は連濁などの接続部の変化が大部分を占めることを明らかにする．さらにこのゆれの関係に基づいた正規化による照合を提案する．すなわち，実際にすべてのゆれを 21,276 組の文字列の等式関係で記述し，そこから自動的に 15,841 の正規化規則を作成して照合する方法を提案する．この正規化規則を使った照合法を人名の分布にしたがった検索に適用し，再現率と適合率の観点から評価を行った．その結果，93% の適合率を達成したうえで，完全一致検索では 1 検索あたり 15% 存在していたゆれによる検索もれを解消した．人名についてかな表記のゆれが存在してももれのない検索が可能となった．

### 6.2.1 日本人名のかな表記のゆれ

日本人名のかな表記にはゆれとよばれる変形が多数存在している．かな表記のゆれとは，同一の語に対して複数のかな表記が許容されている現象をいう [国語研 1983]．表記のゆれは一般に観察される現象であるが，特に人名には「なかだーなかた (中田)」，「あめみやーあまみや (雨宮)」など多数の例が存在する．この現象は日常会話では見逃されることがあっても，情報検索では問題になる．電話番号検索や文献検索などのように人名から情報を検索する時，検索者は登録されている人名を正確に入力する必要がある．しかし対象に表記のゆれがある場合は，問い合わせの表現と登録情報の表現に不一致が生じ情報の検索もれが生じる場合がある．この問題を解決するために，検索を行なう際に「完全に」一致したものばかりでなく「おおまかに」一致するかどうかを判定する処理，近似文字列照合 (approximate string matching) が必要である．

本節では，かな表記を使う人名検索において，ゆれのある文字列の同一性を判断する近似照合法について述べる．近似照合を実現するための第 1 の課題は，どのような人名に，どのようなゆれが存在するかを明らかにすることである．日本の人名は種類が多く，検索のためにはゆれの類型化が必要である．そのため 3,000 万件の人名に対してゆれの解析を行ない，ゆれを具体的に求めてみる．第 2 の課題は，求めたゆれの情報に基づいて，

効率の良い検索ができることである．これに対しては，表記を統一して照合を行なう正規化による照合法を行う．正規化規則を人手で作成する作業は一般に容易ではないために，本節ではゆれの関係から機械的に正規化規則を求める手法を提案し，実際に正規化規則を作成する．このうえで，明らかにしたゆれに対して，検索もれがないことを保証し，無駄な名前を検索しないことを目標とする．この目標に対しては，提案する照合法を，現実に想定される問い合わせに対する検索の再現率 (適合すると判断される総レコード中，実際に検索された割合) と適合率 (検索されたレコード中，適合すると判断される割合)[Cleventon 1962] の期待値から評価する．さらにこの照合法を 10,000 人のデータベースにおける検索処理に適用し，ゆれによる検索もれを解消したことを検証するとともに適合率に関する課題を論じる．

### 6.2.2 情報検索における表記のゆれの問題

情報検索における課題として，キーワードの表記の不一致への対策の必要性が認識されている [藤澤 1993]．本研究も表記の不一致への対策を論じる．まず関連する既存の研究を説明しながら，本研究の位置付けについて説明する．

#### 人名のかな表記のゆれ

本研究で対象とするかな表記を検索キーとする検索法には，漢字に変換する手間がかからない，漢字が分からなかったり入力できない場合でも検索可能，などの利点がある．この検索では，音声の聞き違い，同一漢字の使用などの様々な程度や要因で表記の不一致が起きている．このなかでも「ゆれ」すなわち同じ名前を指しているにもかかわらず複数の表記が認められている現象を第一に解決する必要がある．

前節で第 1 の課題として，ゆれの起こる姓とその類型を明らかにすることをあげた．人名のかな表記に関して文献 [田中 1975][田中 1977][高橋 1992][宇土 1993] の報告があり，かな表記の変形には，かなづかい，清音濁音，長音の扱いの違いなどがあることが明らかになっている．しかしその全体像は明らかになっていないため，次節で 3,000 万人のデータをもとにかな表記のゆれの分析を行なう．

#### 正規化

第 2 の課題である効率良い検索を行なうために，正規化を使った照合を行う．一般に，文字列の正規化を使った検索は，問い合わせ文字列と登録情報の文字列の正規形を作って比較するために，1 回の照合で複数の異なった表記のレコードを検索することができる．この性質はシステムの負荷を減じるので，電話番号検索のシステムなどで使用されている．

正規化を使った検索について，Hall と Dowling が同値性問題として述べている [Hall 1980]．以下同値性問題について要約する．文字列の集合 $S$ における関係 $R$ が，文字列 $a, b, c \in S$ に対して，反射律，対称律，推移律すなわち，

$$
\begin{aligned}
&(1) \quad aRa \\
&(2) \quad aRb \quad \Rightarrow bRa \\
&(3)\ aRb,\ bRc \Rightarrow aRc
\end{aligned}
$$

を満たすとき，$R$ を同値関係と呼ぶ．このとき集合 $S$ は $R$ によって，同値類 $S_1$, $S_2$, $S_3$,.., の直和に分解することが可能である．問い合わせ文字列 $q$ に対して，同値関係にある文字列を検索する処理は，$q$ の属する同値類 $S_q$ に含まれる全ての文字列 $a \in S_q$ を求めることと言い換えることができる．このことは，同値類ごとに正規形と呼ばれる文字列を定め，同値類に属する文字列をその正規形に変換する正規化規則を与えることができれば，文字列 $q$ の正規形と同じ正規形を持つレコードを探すことと定義することができる．これを人名の検索に適用したのが本研究である．文字列照合節では人名のかな表記のゆれから同値関係 $R$ を求める．

さらに正規化処理をするためには，具体的に正規化規則を求めなければならない．まず現在行なわれている正規化手法を紹介する．

英語の代表的な類音誤りの正規化手法に，現在でも ISODE のディレクトリ [Robbins 1991] など使われている Soundex システム [Odell 1918][Odell 1922] がある．このシステムはアルファベット 26 文字を定義した 7 つのグループに分類し，1 文字ずつ各グループに対応したコード表6.4に変換したうえで (先頭文字はそのまま残す) 照合を行なう．

表 6.4 Soundex Codes

| | | | |
|---|---|---|---|
| 0 | A E I O U H W Y | 4 | L |
| 1 | B F P V | 5 | M N |
| 2 | C G J K Q S X Z | 6 | R |
| 3 | D T | | |

一方日本語のシステムでとられている正規化の代表は清音と濁音の対立などの扱いである．表6.5に番号案内のシステム [宮部 1983][戸部 1990] で行なわれている対策を示す．表 2 以外にもゆれは存在するが，それらは漢字に依存するなどの理由で普遍的な規則を記述しにくい．実際に番号案内のシステムでは，人名ごとに随時派生テーブルを用意して

表 6.5 番号案内におけるかな表記のゆれの対策の例

| 問題点 | 対策 |
|---|---|
| 濁音・半濁音の間違いやすさ | 濁音・半濁音を清音に変換 |
| 拗音・促音の書き方 | 小書きを大書きに変換 |
| 「じ・ぢ」「ず・づ」の使い分け | 「ぢ」を「じ」に「づ」を「ず」に変換 |

ゆれの対処を行なっている．現在カタカナの異表記を正規化する手法が提案されているが [獅々堀 1994]，漢字のかな表記のゆれのための規則が求められている．

表6.5の正規化規則は単純であるが，精度を増すために，「むかいだ」「むかだ」「むこうだ」「こうだ」を等しく扱いたいなどと，正規化に盛り込む関係を増やすと規則を人手で作成することが困難になる．規則が正確に作成されないと，本来一つの形になるべきものが，異なる形に変形されてしまうことがある．ここで必要な処理は，文字列の等しいという関係から，等しい文字列に対してはその関係を保ったまま正規形への変換を行なう正規化規則を求める処理である．本節ではこの処理に Knuth-Bendix の完備化アルゴリズム [Knuth 1970] を適用し，文字列照合の節で求めた $R$ から正規化規則を作成する．この正規化規則はもれのない検索を保証する．これはアドホックに正規形を求めた検索ではできない性質である．

もれのないことの副作用として，意図しなかった名前が同じ正規形に変換されること，すなわち適合率が低下する可能性がある．本手法は再現率を重視したものであり，適合率に関する問題は検索節で実際のデータにより検証する．

### 6.2.3 かな表記のゆれの解析

**調査の対象**

本調査は電話帳に掲載されている漢字表記を持つ姓を対象として行った．日本人の姓は出版されたものだけでも 13 万種強報告されているが [丹羽 1985]，電話帳に掲載されている漢字姓は 16 万 4 千種におよぶ．調査対象の母集団「姓データベース」は，全国全ての電話帳の掲載情報の中から姓の漢字情報とかな表記情報を取り出し，出現度数を付して作成した．1993 年 7 月の電話帳に基づいている．このうち頻度が全国で 3 以上のものを調査対象とした (表6.6)．従って対象の「姓」の種類 (漢字とかなの異なり語数) は約 10 万 8 千，のべ語数約 3,200 万の集合である．表6.7に例を示す．使用字種は「漢字」は JIS 漢字 (JIS X 0208) のみ 2,957 種，「かな表記」は「を」を除く直音全て 69 音 (「ぢ」「づ」を含む) と撥音「ん」の 70 種である．「ゐ」「ゑ」は含まれない．拗音，促音に用いられる「ゃ・ゅ・ょ・っ」は小書きされていない．

表 6.6 姓データベースの度数

| 漢字長 | 異なり語数 | | | のべ語数 | 姓の例 |
|---|---|---|---|---|---|
| | 漢字 + かな | 漢字 | かな | | |
| 1 | 2,344 | 1,464 | 1,261 | 1,098,963 | 林，森，原 |
| 2 | 86,413 | 60,386 | 48,074 | 29,281,304 | 佐藤，鈴木 |
| 3 | 19,336 | 15,047 | 15,116 | 1,313,876 | 佐々木，長谷川 |
| 4 | 357 | 273 | 326 | 6,164 | 勅使河原 |
| 全体 | 108,450 | 77,170 | 60,283 | 31,700,307 | |

表 6.7 姓データベースの例 (上位 10)

| 順位 | 漢字 | かな表記 | 出現度数 | 順位 | 漢字 | かな表記 | 出現度数 |
|---|---|---|---|---|---|---|---|
| 1 | 佐藤 | さ／とう | 486,347 | 6 | 伊藤 | い／とう | 272,248 |
| 2 | 鈴木 | すず／き | 430,176 | 7 | 山本 | やま／もと | 271,651 |
| 3 | 高橋 | たか／はし | 357,501 | 8 | 中村 | なか／むら | 266,947 |
| 4 | 田中 | た／なか | 337,310 | 9 | 小林 | こ／ばやし | 256,734 |
| 5 | 渡辺 | わたな／べ | 280,624 | 10 | 加藤 | か／とう | 216,225 |

**ゆれの判断基準**

ゆれの関係にある姓のかな表記は，同一の姓を指すものである．例えば「やまざき／やまさき (山崎)」は同一と判断し，「しょうじ／とうかいりん (東海林)」や「ほんたに／もとや (本谷)」は別の姓という立場をとることとする．姓の同一か否かの判断を 3,000 万のデータ全てに対して行なうための具体的な基準が必要である．そこで漢字単位に人名に用いられるかな表記を整理した．

漢字単位のゆれの判断の基準を初期には「漢和辞典の見出しで音と訓のように別項目扱いのものはゆれではない」と考えた．しかし常用漢字表 [国語課 1992] を見ても「あめ，さめ (雨)」や「わける，わかれる，わかる，わかつ (分)」などの派生して生まれたものがそれぞれ見出しになっている例が存在した．また辞典に全く記載がないかな表記も存在し，慣用音まで含んだ音の体系化は容易ではない [湯沢 1987] のが現状であり，独自の判断基準が必要になった．

そこで全ての姓を漢字単位に分解して，漢字ごとにかな表記を集め，全てのかな表記の対立を (1) 変形位置，(2) 音種，(3) 形態の観点から 35 のカテゴリーに分類して，そのカテゴリーごとにゆれとみなせる関係が存在するかを調べた．その結果から漢字単位にかな

表記をゆれのグループに分類する「漢字かな表記辞書」を作成し，これに従って姓のゆれの判定を行った．

**姓のかな表記のゆれ**

表 6.8 姓のゆれ単位の例

| 漢字 | かな表記 |
|---|---|
| 神代 | くましろ |
| 神代 | こうしろ, こうじろ, こおしろ |
| 神代 | じんだい, しんだい |
| 神代 | かみしろ, かじろ, かしろ |
| 神代 | かくみ, かこみ |
| 神代 | かみよ |
| 神代 | こうだい |
| 水谷 | みずたに, みづたに, みつたに, みすたに |
| 水谷 | みずがい |
| 水谷 | みずや, みずのや |
| 水谷 | すいたに |
| 角田 | つのだ, つのた |
| 角田 | かくた, かくだ, かどた, かどだ |
| 角田 | すみだ, すみた, すまだ, すだ |

6.2.6 節の処理を行なって全ての姓を「姓のゆれ単位」(以後 NVU と略す) に分類した (表6.8)．NVU は「漢字表記」と「かな表記」の組を単位とする「姓」の集合で，1つの NVU に属する姓は，漢字表記が同じで，かな表記が同一の名前をさす，すなわちゆれの関係にあるという性質を持つ．ゆれの判定は，前節で作成した「漢字かな表記辞書」に従った．

例えば「神代」には 13 種類のかな表記がある．このうちかな表記が「こうしろ」「こうじろ」「こおしろ」の姓を同じ名前と判断し 1 つの NVU に，同様に「かみしろ」「かじろ」「かしろ」を別の NVU に，つごう「神代」姓を 7 つの NVU に分類した．こうして姓全体 108,450 種を 87,630 の NVU に分類した．このうち 16,754 の NVU には 2 つ以上のかな表記が対応し，ゆれがあることが分かった．これは実データの 58% に何らかのゆれが存在することを示している．

次にゆれの起きた原因を表6.9に示す．表6.9は NVU ごとに存在するゆれの原因を調べ集計した．連濁 (連接部の後項語頭の清音が濁音化する，詳しくは文献 [佐藤 1989] を

参照) が全 NVU の 6 割以上に存在する原因である．しかしそれ以外にも「あめみや-あまみや (雨宮)」といった母音のゆれや，「ふじわら-ふじはら (藤原)」といったの子音のゆれが存在している．これらは従来の正規化規則では対処できていないゆれである．

表 6.9 姓のかな表記のゆれの原因

比率はゆれのある 16,754 の NVU に対する百分率で示した．

| 変形位置 | 種別 | 例 | 件数 | 比率 (%) |
|---|---|---|---|---|
| 連接部前項 | 母音 | あめみや-あまみや (雨宮) | 1,498 | 8.9 |
| | 助詞 | やまうち-やまのうち (山内) | 753 | 4.5 |
| | その他 | かみざき-かんざき (神崎) | 1,186 | 7.1 |
| 連接部後項 | 連濁 | やまさき-やまざき (山崎) | 10,725 | 64.0 |
| | その他 | ふじわら-ふじはら (藤原) | 963 | 5.7 |
| 語頭 | 子音 | おやま-こやま (小山) | 786 | 4.7 |
| | その他 | しかの-かの (鹿野) | 194 | 1.2 |
| 末尾 | 母音 | じんぼ-じんぼう (神保) | 813 | 4.9 |
| | その他 | たかはた-たかはたけ (高畠) | 419 | 2.5 |
| その他 | | やぎぬま-やなぎぬま (柳沼) | 293 | 1.7 |
| ぢ/じ，づ/ず | | みづたに-みずたに (水谷) | 410 | 2.4 |
| おう/おお | | とうやま-とおやま (遠山) | 217 | 1.3 |
| 複数原因の混在 | | こうじろ-こおしろ (神代) | 2,747 | 16.4 |

### 6.2.4 ゆれのある情報に対する文字列照合

本章では，目標とする近似照合を行なうために，前節で明らかにした姓のゆれ単位 (NVU) から正規化規則を作成する．始めに「ゆれに基づく一致」を定義し，次に正規化に用いる同値類を定義し，その上で正規化規則を作成する．

#### ゆれに基づく一致

目標とする照合は，人名の検索において問い合わせまたは登録情報のかな表記にゆれがあっても，もれのない検索を行なうことである．この検索を「ゆれに基づく一致」と呼ぶことにする．定義を次に示す．

姓のかな表記の集合を $S$ とするとき，$S$ における関係 $V_\delta$ を，$a,b \in S$ に対して，$a$ と $b$ が同じ $NVU$ に属すとき $aV_\delta b$ が成り立つとする．すなわち，

$\exists\ \kappa, \exists u; (\kappa, a) \in u \wedge (\kappa, b) \in u \Leftrightarrow aV_\delta b.$

ただし $\kappa$ はある姓の漢字表記，$u$ はある NVU.

このとき「ゆれに基づく一致」は，問い合わせ文字列を q とすると，q に対して，

$\forall\ x \in A(q)$，ただし $A(q) = \{x \mid x \in S, qV_\delta x\}$

なる x を求めることである．

A(q) は，NVU から具体的に求めることができる．例えば問い合わせが「かくた」であれば，「角田」「額田」などの NVU から，

A("かくた") = かくた, かくだ, かどた, かどだ, がくた

となる．この集合は問い合わせに対してゆれの関係にある姓のかな表記を網羅している．この集合を次の集合を比べてみよう．

X("かくた") = { かくた, かくだ, かどた, かどだ, がくた, すだ, すまだ, すみた, すみだ, つのた, つのだ, ぬかた, ぬかだ, ぬがた }

X("かくた") は，"かくた"の「漢字表記」と同じ「漢字表記」を持つ「かな表記」を集めたものであるが，ゆれを求めるためには精度が低い．NVU を明らかにしたことによって「ゆれに基づく一致集合」A(q) が求められ，ゆれの関係取り出すことが可能になった．以下 A(q) に基づいて検索を行う方法について述べる．

**正規化に用いる同値類の定義**

姓のかな表記の集合 S における関係 V を次のように定義する．

$\exists\ h_1, h_2, ..., h_n \in S;\ aV_\delta h1 \wedge h_1V_\delta h_2 \wedge ... \wedge h_nV_\delta b \Leftrightarrow aVb.$

V は推移律を満たすので同値関係である．またこの定義から $aV_\delta b \Rightarrow aVb$ であることがわかる．

関係 V から次の検索が定義できる．問い合わせ文字列を q とすると，

$\forall\ x \in B(q)$，ただし $B(q) = \{x \mid x \in S, qVx\}$.

なる $x$ を求める．この $B(q)$ は $S$ における $q$ の同値類であり，冒頭で述べた同値性問題の枠組で正規化処理を行なうことができる．

同値類集合 $B(q)$ と，ゆれに基づく一致集合 $A(q)$ の間には $B(q) \supseteq A(q)$ なる関係があるため，正規化をして検索を行った場合，ゆれの関係にあるかな表記は全て検索される．ただし同じ NVU に属さないかな表記も検索される可能性がある．このことについては検索節で分析する．

**正規化規則の作成**

NVU は数万あるので，NVU から正規化の変換規則を正しく作成する作業は容易ではない．それゆれ正当性がある操作で，変換規則が具体的に求められることが必要である．

同値関係から変換規則を求めるアルゴリズムは Knuth-Bendix の完備化アルゴリズムとして知られている．このアルゴリズムの基本は，1 つの文字列が異なる 2 つの文字列に書き換えられる場合，この 2 つの文字列が等しくなるような規則の追加を繰り返すことである．この完備化アルゴリズムを使って，姓のかな表記のゆれの全ての変形から正規化規則を作成した．なお完備化アルゴリズムについては，6.2.7 節で説明する．

はじめに述べた通り，提案する正規化法は，姓のかな表記を V による同値類に分けることであり，同じ同値類のかな表記が一つの正規形に変換される変換規則を求めることが必要である．扱いたいゆれの関係は，ゆれの解析節で明らかにした全ての NVU(姓のゆれ単位) の任意の 2 かな表記を関係 V で結んだものである．例えば，表 5 の 2 番目の NVU は，次のように表現される．

"こうしろ"V"こうじろ",
"こうしろ"V"こおしろ",
"こうじろ"V"こおしろ"

こうして全てのゆれの関係 21,276 組から正規化規則を求める処理を行った．その結果，もともと Knuth-Bendix の完備化アルゴリズムは停止性が保証されていないが，V については処理は停止し，ゆれの正規化規則 15,841 組を得た．処理する際に一つの文字列全体を一つのシンボルとして扱った ("こうしろ"と"こうじろ"を別個のシンボルと扱う)．与えたゆれの関係の集合を図 6.1 に，完備化アルゴリズムによって作成したゆれの正規化規則を図 6.2 に示す．

### 6.2.5 検索処理への適用

**検索の定義**

文字列照合節で求めた正規化規則を使った検索を以下のように定義する．ゆれの正規化規則集合を C とする．

**ゆれの正規化規則を使った検索** 問い合わせ文字列の C による正規形に対して登録情報の文字列の C による正規形が一致するものを探す．

**正規形** 文字列 $w_1$ に対して変換規則の集合 R を適用して変換文字列 $w_2$ を得る．この操作を $w_2$ に適用する規則がなくなるまで繰り返す．$w_2$ を $w_1$ の $R$ による正規形と呼ぶ．

{
“あいいそ” V “あいそ”,　“あいうら” V “あいのうら”,　“あいかさ” V “あいがさ”,
“あいかみ” V “あいがみ”,　“あいかわ” V “あいがわ”,　“あいかわ” V “あわかわ”,
“あいかわ” V “かいがわ”,　“あいか” V “あきか”,　“あいか” V “あきしか”,
“あいきよう” V “あゆきよう”,　“あいき” V “あいぎ”,　“あいくち” V “あいぐち”,
“あいさか” V “あいざか”,　“あいさき” V “あいざき”,　“あいさわ” V “あいざわ”,
“あいざわ” V “あゆさわ”,　“あいざわ” V “あゆざわ”,　“あいしま” V “あいじま”,
“あいず” V “あいつ”,　“あいず” V “あいづ”,　“あいず” V “かいづ”,
“あいせき” =“あいぜき”,　“あいせん” V “あいぜん”,　“あいた” V “あいだ”,
“あいた” V “かいた”,　“あいた” V “かいだ”,　“あいだ”=“かいた”,
“あいだ” V “かいだ”,　“あいつき” V “あいづき”,　“あいつ” V “あいづ”,
.....
}

図 6.1　かな表記のゆれの等式集合

{
“あいのうら”⇒“あいうら”,　“あいがさ”⇒“あいかさ”,　“あいがみ”⇒“あいかみ”,
“あいがわ”⇒“あいかわ”,　“あわかわ”⇒“あいかわ”,　“かいかわ”⇒“あいかわ”,
“かいがわ”⇒“あいかわ”,　“かゆかわ”⇒“あいかわ”,　“かゆがわ”⇒“あいかわ”,
“あきか”⇒“あいか”,　“あきしか”⇒“あいか”,　“あゆきよう”⇒“あいきよう”,
“あいぎ”⇒“あいき”,　“あいぐち”⇒“あいくち”,　“あいざか”⇒“あいさか”,
“あいざき”⇒“あいさき”,　“あいざわ”⇒“あいさわ”,　“あゆさわ”⇒“あいさわ”,
“あゆざわ”⇒“あいさわ”,　“あいじま”⇒“あいしま”,　“あいつ”⇒“あいず”,
“あいづ”⇒“あいず”,　“かいず”⇒“あいず”,　“かいつ”⇒“あいず”,
“かいづ”⇒“あいず”,　“あいぜき”⇒“あいせき”,　“あいぜん”⇒“あいせん”,
“あいいそ”⇒“あいそ”,　“あいだ”⇒“あいた”,　“かいた”⇒“あいた”,
.....
}

図 6.2　かな表記のゆれの正規化規則

### 評価

ゆれの正規化規則を使って，レコード数 3,000 万件の姓データベースを検索する実験を行った．本節では，実験結果を適合率と再現率の期待値から評価する．

問い合わせの入力は，データベースに存在するかな表記全てが，その出現頻度に応じて現れる (例えば「さとう」は全問い合わせの 1.5% になる) と仮定した．文字列 $q$ が入力される確率を $P_q$，ある照合一致基準が定義された時，問い合わせ $q$ に対する検索レコード数を $M_q$ とすると，検索レコード数の期待値は次の式で表される．

ただし，入力に関して前記の仮定をおいたので，$P_q$ は $q$ をかな表記とするレコードのデータベース中の存在比率として求めることができる．

実験は照合一致基準として，(1) 提案する「ゆれの正規化法」，(2) 表 2 に示した濁音の清音化などを行なう「従来の手法」，(3) 問い合わせと同じ文字列のみを照合する「完全一致法」について行った．その結果，検索レコード数の期待値は，順に 274，222，216(10 万人あたりの検索に換算) となった．

この結果に対して，ゆれによる見落としを防いだ効果と，それに伴って無駄なレコードが検索された程度の 2 点を調べる．この 2 点は，「問い合わせに対してゆれの関係にあるものであれば，適合する (relevant)」と判断する場合，再現率，適合率となる．この場合，前節求めた「ゆれに基づく一致集合 (A(q))」は，問い合わせ q に対して，「適合する」と判断される集合である．従って，入力 q に対して，それぞれの照合一致基準による検索レコードの集合を R(q) とすると R(q) と A(q) から，

$$\text{再現率} = \mathrm{N}(\mathrm{R}(\mathrm{q}) \cap \mathrm{A}(\mathrm{q})) \,/\, \mathrm{N}(\mathrm{A}(\mathrm{q})).$$
$$\text{適合率} = \mathrm{N}(\mathrm{R}(\mathrm{q}) \cap \mathrm{A}(\mathrm{q})) \,/\, \mathrm{N}(\mathrm{R}(\mathrm{q})).$$

として計算することができる (ただし N(S) は，集合 S の要素のデータベース中の存在数とする)．これらの期待値は，検索レコード数の期待値と同様の計算で求めることができる．以上の結果を表 6.10 に示す．完全一致法では再現率は 85% であり，適合するレコードの 15% が検索にもれていたことがわかる．検索もれの解消は，再現率を 91% まで上げている従来の手法でも不十分であり，本手法の有効性を確認した．この時の適合率は 93% であった．実際のシステムにおいては，いったん検索したレコードから不要なものを落とす方法はたくさんある．一方，一度落ちてしまった情報については対処の方法がない．それゆえ適合率より再現率を重要視する方法が有用である．

### 姓のかな表記の同値類

正規化は姓のかな表記の集合を互いに交わりのない同値類に分解する処理である．この処理で全 60,283 種の姓のかな表記が 44,442 の集合に類別された．「やまもと」，「あおき」

表 6.10　ゆれの正規化規則を使った検索の期待値

検索レコードは 10 万人あたりの件数に換算して示す.

| 照合一致基準 | 検索レコード数 | 適合率 | 再現率 |
|---|---|---|---|
| ゆれの正規化法 | 274 | 0.93 | 1 |
| 従来の手法 | 222 | 0.99 | 0.91 |
| 完全一致法 | 216 | 1 | 0.85 |

など 34,500 種の姓は単独で同値類を作っており，ゆれに対して安定した姓であるといえる．次に「やまさき」と「やまざき」などのように 15,042 種の姓が，2 つのかな表記で同値類を作っている．一方 10 を越えるかな表記が連結してできた同値類が 81 種，1,564 語存在した．かな表記数が最大となった正規化の同値類は正規形が「ゆ」となる集合で表記数は 109 である ( 図6.3 )．これらの存在が適合率を下げる原因となっている．

**実データベースへの適用**

提案する照合法を実際のデータベースに適用する時の影響を述べる．姓データベースから，姓の実データ上の重みづけを行った上で無作為に抽出して，レコード数 10,000，かな表記の異なり語数 3,540 種の試験データベースを作成し，これを用いてゆれの正規化規則を使った検索の実験を行った．本電話帳データベースの場合，統計的には母集団が 10,000 を越えると 1 検索あたりの検索レコード数の期待値が 20 を越えるため住所などの他の条件を付加して対象を少なくとも 10,000 以下にする必要がある．検索の例を表6.11 に示す．

表 6.11　検索例

| 入力 | 出力 |
|---|---|
| おはら | おおはら，おはら，おばら，こはら，こばら，こばる |
| とみやま | とうやま，とおやま，とみやま，とやま |
| わたなべ | わたなべ，わたべ |
| あがづま | あがつま，あずま，わがつま |
| かしはら | かしはら，かしわばら，かじはら，かじわら |

このように検索結果はゆれに基づく一致集合に該当するかな表記を全て含み，ゆれによる検索もれを解消した．この検索の検索レコード数の期待値は 27.6 件で，完全一致の場合は 20.1 であり，この差 7.5 件の中にゆれの存在によって検索からもれていたレコード

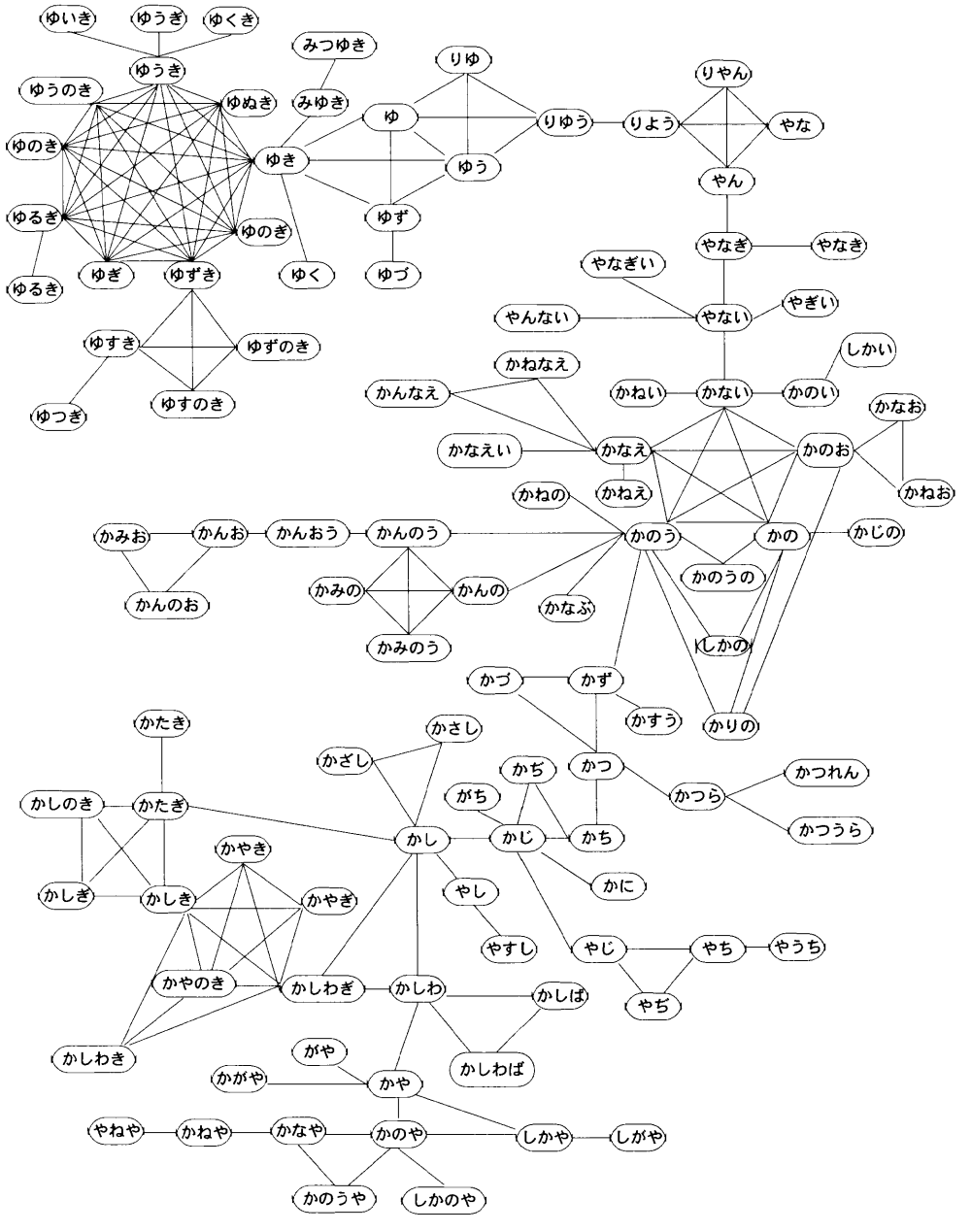



図 6.3 ゆれの同値類の例

が含まれている．照合一致されるゆれの文字種が多種へ拡張されているだけでなく，「かしはら」に対する「かしわばら」などの長さの変化を伴うゆれへの対処も可能となっている．

最後に正規化による悪影響の可能性について検索レコードの数と種類の観点から述べる．最大の検索レコード数を示すものは正規形が「さとう」となる検索で 232 件である．これは表記が「さいとう」なる「佐藤」姓の存在を介して「さとう」(153 件) と「さいとう」(79 件) が合流したためである．一方最多種の姓の合流は図 3 で示した正規形が「ゆ」となる検索で，実際に検索された姓は 24 種，検索レコード数は 49 件にすぎない．この他の例でも多数の合流による検索レコード数の爆発的増加はおきなかった．もしこれらの正規化による合流を避けたいならば，頻度の少ないかな表記を持つ姓に対して，データベース上で代表的なかな表記を別名 (alias) として登録しておき，その姓にかかわる変換を正規化規則から外すことが可能である．こうすることによってデータベースの規模は大きくなるものの正規化規則数および望ましくない合流を減じることができる．これは具体的なシステム設計時の最適化の課題である．

### 6.2.6 姓に使われる漢字のかな表記に現れるゆれの調査

以下では，ゆれの解析節で述べた，姓を「姓のゆれ単位」(NVU) に分類する手順について述べる．

#### 手順

1. 漢字とその語中の出現位置単位に全てのかな表記を使用度数 (使われている姓の種類) を付与して集める．
2. 漢字単位にかな表記間の対立を変形位置，音種，形態のカテゴリーに分類する．
3. カテゴリー単位にゆれの関係が含まれているかを調べる．
4. 漢字単位のかな表記のゆれ関係を導く (「漢字かな表記辞書」)．
5. 姓を NVU に分類する．

#### 漢字単位のゆれの判定

(1) から (3) までの調査の結果を表 6.12 に示す．以下で説明する基準に基づき，判定の項目に○印を付したカテゴリーにゆれの関係が存在すると判定した．

それぞれのカテゴリーでゆれと判断した基準を具体的に説明する．まず漢字の基本的な音訓の枠組，訓の活用の有無を，常用漢字表や漢和辞典 [藤堂 1987] などで判断した．その上でゆれはかなづかいの対立 1) や音韻の変化 [小松 1981]，読みあやまり [国語研 1955] などの典型的パターンが定着したものであると考え，ゆれが起こるケースとして次の 3 項

を設けた．(1) かなづかいの変化．(2) 末尾の変化．(3) 漢字接続部の変化．

**■かなづかい** 表6.12のカテゴリー31，32(以下C31などと記する)．かなづかいのゆれとして「ぢ/じ，づ/ず」と二重母音「おう/おお」の書き分けが存在する．この2つは語中のあらゆる位置で現れた．

**■末尾の対立** C1-6，C19-24．これらは語幹が同じでありこのカテゴリーに属する関係をゆれと考えた．ただしC3に分類される「き-こ (木)」の例は語幹が同じではないが母音交替として知られた現象であるのでこれもゆれと扱う．

**■連接部の対立** C1-6，C7-12．連接部後項の変化は語幹の変化が起こる．このうちC7，C8，C10，C12には語幹の対立のうち一般語彙でも存在する変化であり順に説明する．

連接部の音の交替としては，先頭の子音が清音から濁音に変化する連濁が知られており，ゆれの代表的なものである (C7の一部)．それ以外の子音の交替では派生の認められるものに「は，わ，ば，ぱ」(はら-わら (原)) の対立などがある．他方辞書で別個の音として扱われているものに「ぶ-む (武)」などがあるが，カテゴリー全体としてはゆれを含んでいる．同様のことが子音の脱落/添加にもあてはまる．この連濁以外のC7とC8を条件つきでゆれに分類する．しかしC9，C11には派生関係が見い出せなかった．

同じく音の脱落/添加には「の・が・つ・な・て・ん」の添加が多く見受けられる．これらは助詞の役割を果たしていると考えゆれとみなす (C12の一部)．その他の脱落/添加にも純粋な音の省略が含まれておりC10，C12の残りを条件つきでゆれとみなす．

なおこれらの連接部の対立は転音，音便，音韻の添加，脱落，融合などの複合語の変音現象として国語研究所の文献 [国語研 1985] [国語研 1988] などに詳しい．

**■その他の対立** 先頭部の対立は語幹の変化であるため原則としてゆれと認めない (C15，C17)．ただし清音と濁音の対立をゆれと認めた (C13の一部)．他は連接部後項の先頭と同様に考えて，C13の残りとC14，C16，C18を条件つきでゆれとみなす．

それ以外の関係では，語幹が変化しない内部での1文字の交替，脱落/添加の全て (C25-30) と，頭部で1文字以上の共通文字列があるもの (C33) のみをゆれと考え，他は全て除外した (C34，C35)．

#### 漢字かな表記辞書

前節で判断した関係に次の規則を入れて実際に派生関係として機能しているものをとりだした．この規則は，「漢字には複数の代表的な表記があり，それから音の変化や脱落，活用などで派生が生まれている」と仮定して作成した．派生関係を取り出す処理を図6.4に示す．

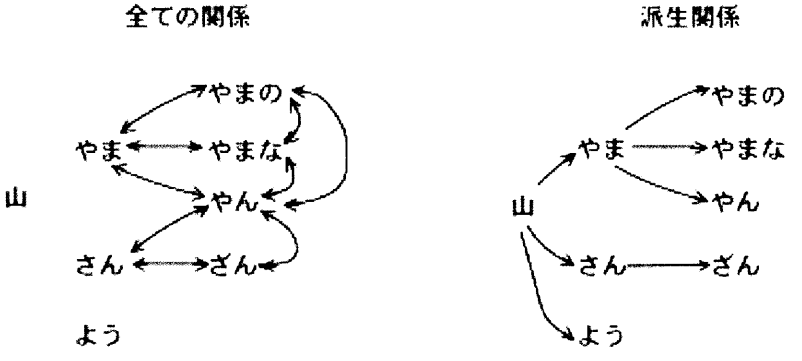



図 6.4 派生関係の抽出

1. 使用度数の多いものから少ないものへ派生する.
2. 2種類以上の親表記から派生を受けることはない.
3. 2種類以上の親表記の候補が存在する時は，語幹が変わらないもの，使用度数の多いものを優先する.

「やま」「さん」「よう」のように派生の先頭に位置するものが，漢字の代表的な表記であり (見出し表記と呼ぶことにする)，それ以外の表記は見出し表記に対する派生と扱った. このことから表6.13に示す漢字かな表記辞書を得た.

**姓のかな表記のゆれ**

漢字かな表記辞書を使って任意の姓がゆれの関係にあるかを調べることができる. すなわち同一の漢字を持つ姓のうちで，かな表記を漢字単位に見出し表記に変換したとき，見出し表記が同じになるものがゆれの関係にある. この結果得られた NVU を本文の表6.13に示す.

表 6.12 かな表記の対立のカテゴリーとゆれの判定

| | 変形位置 | 音種別 | 形態 | 例 | 判定 |
|---|---|---|---|---|---|
| 1 | 連接部 | 子音 | 交替 | かわ-かは (川-) | ○ |
| 2 | 前項末尾 | | 脱落/添加 | ほう-ほく (北-) | ○ |
| 3 | | 母音 | 交替 | あめ-あま (雨-) | ○ |
| 4 | | | 脱落/添加 | そう-そ (曽-) | ○ |
| 5 | | その他 | 交替 | こえ-こし (越-) | ○ |
| 6 | | | 脱落/添加 | かわ-か (川) | ○ |
| 7 | 連接部 | 子音 | 交替 | はら-わら (-原) | ○ |
| 8 | 後項先頭 | | 脱落/添加 | おん-のん (-音) | ○ |
| 9 | | 母音 | 交替 | ざい-ぜい (-西) | × |
| 10 | | | 脱落/添加 | うみ-み (-海) | ○ |
| 11 | | その他 | 交替 | かん-じん (-神) | × |
| 12 | | | 脱落/添加 | みなみ-なみ (南) | ○ |
| 13 | 先頭 | 子音 | 交替 | しち-ひち (七-) | ○ |
| 14 | | | 脱落/添加 | わが-あが (吾-) | ○ |
| 15 | | 母音 | 交替 | きん-こん (金-) | × |
| 16 | | | 脱落/添加 | いで-で (出-) | ○ |
| 17 | | その他 | 交替 | はぎ-おぎ (萩-) | × |
| 18 | | | 脱落/添加 | むこう-こう (向) | ○ |
| 19 | 末尾 | 子音 | 交替 | ばた-ばな (-端) | ○ |
| 20 | | | 脱落/添加 | かけい-かけひ (筧) | ○ |
| 21 | | 母音 | 交替 | たて-たち (-館) | ○ |
| 22 | | | 脱落/添加 | のう-の (-能) | ○ |
| 23 | | その他 | 交替 | かど-かく (-角) | ○ |
| 24 | | | 脱落/添加 | たいら-たい (-平) | ○ |
| 25 | 内部 | 子音 | 交替 | つむら-つぶら | ○ |
| 26 | | | 脱落/添加 | いおり-いほり (庵) | ○ |
| 27 | | 母音 | 交替 | えびす-えべす (胡) | ○ |
| 28 | | | 脱落/添加 | おぎ-おおぎ (扇) | ○ |
| 29 | | その他 | 交替 | はりのき-はんのき (榿) | ○ |
| 30 | | | 脱落/添加 | やぎ-やなぎ (柳) | ○ |
| 31 | 任意 | かなづかい | ぢ/じ, づ/ず | ふじ-ふぢ (藤-) | ○ |
| 32 | | | おう/おお | とうり-とおり (通) | ○ |
| 33 | 上記以外の 1 文字共通 | | 先頭文字が共通 | あがり-あげ (上) | ○ |
| 34 | | | その他 | こく-くろ (黒) | × |
| 35 | 共通文字列なし | | さん-やま (山), たか-こう (高) | | × |

表 6.13 漢字かな表記辞書

下線を付した語は見出し表記

| 漢字 | 位置 | 表記 |
|---|---|---|
| 田 | 先頭 | た，たの，たん，たな，たつ，だ |
| | | でん |
| | | とう，とお |
| 上 | 末尾 | がみ，かみ |
| | | うえ，え，う |
| | | じょう，しょう |
| | | あげ，あがり |

### 6.2.7 完備化アルゴリズム

完備化アルゴリズムの定義を示す

R は変換規則集合，G は等式集合，GT は定義済みの順序とする．
R を空とする．G が空でないならば以下のことを繰り返せ．

1. G から一つの要素 $wg_1 = wg_2$ を取り出す．$wg_1$ と $wg_2$ が同じ文字列ならばそれを捨て，1. にいく．$GT(wg_1, wg_2)$ ならば $wg_1 \rightarrow wg_2$ という変換規則を作る．$GT(wg_2, wg_1)$ ならば $wg_2 \rightarrow wg_1$ という変換規則を作る．GT の性質から，このいずれにも該当しない場合はない．ここで求まる変換規則を $w_1 \rightarrow w_2$ とおく．
2. R の要素であるすべての規則 $w_3 \rightarrow w_4$ について，$w_3$ が $w_1 \rightarrow w_2$ により変形できるならば，$w_3 \rightarrow w_4$ を R から取り除き，G に $w_3 = w_4$ を追加する．
3. $w_1 \rightarrow w_2$ と R の全ての規則 $w_3 \rightarrow w_4$ について，Critical Pair の $w_5$ と $w_6$ をもとめ，G に $w_5 = w_6$ を追加する．
4. $w_1 \rightarrow w_2$ を R に追加する．R の要素である全ての規則 $w_3 \rightarrow w_4$ について $w_4$ の R による正規形 $w_7$ を求める．そして，$w_3 \rightarrow w_4$ を取り除き $w_3 \rightarrow w_7$ の規則要素を追加する．
5. R の要素である全ての等式集合 $w_3 = w_4$ について，$w_3$ の R による正規形 $w_8$ と $w_4$ の R による正規系 $w_9$ を求める．G から $w_3 = w_4$ を取り除いて $w_8 = w_9$ を加える．

Critical Pair の定義

ある変換規則集合 R が存在するとき，そこから任意の 2 つのルールを取り出す（同一のルールであることを含む）．長さが N 以下の文字列集合 SN の要素の文字列がこの 2 つのルールで．それぞれ重なりのある部分が変形され，2 つの変形結果を導いたとする．重なりの部分が同一である制約のもとで，2 つの変換規則の Pair の中の長さ最小の Pair を Critical Pair と呼ぶ．

Critical Pair の求め方

$w_1 \rightarrow w_2$，$w_3 \rightarrow w_4$ という二つの変換規則があったとき，$w_1$ と $w_3$ の共通文字列に注目して，両方の変換規則が適用できる文字列を作る．$\alpha$，$\beta$ は空文字列を含む任意の文字列，$\xi$ は空文字列でない任意の文字列とする．以下の 4 つの場合に適合する条件を探し，$\alpha\,\xi\,\beta$ の形式の文字列に 2 つの変換規則を作用させる．それぞれの規則で変形した文字列の結果の $w_5$ と $w_6$ は Critical Pair である．

$$(w_1, w_3) \in \{(\alpha\,\xi, \xi\,\beta), (\alpha\,\xi\,\beta, \xi), (\xi\,\beta, \alpha\,\xi), (\xi, \alpha\,\xi\,\beta)\}$$

# 第 7 章

# 結論

本章では本論文をまとめるとともに，今後の課題について述べる．

## 7.1 本論文のまとめ

本論文では，はじめにコンテンツ中のジオワードを用いて地理的な検索を実現する方法を提案した．まずイエローページデータにデジタル地図データを統合することにより，イエローページデータに緯度経度を持たせる手法について述べ，さらにウェブ文書とイエローページデータを統合することにより，ウェブ文書に緯度経度を持たせる手法を提案した．前者はイエローページデータを地理的検索と地図表示可能にしたもので，この仕組みは実システムでも永く用いられ有効性が確認された技術である．後者で明らかにしたウェブ文書，イエローページ，地図 3 者間の関係は，続いて述べたウェブ文書の地理的検索手法を通じて，現在の一般的なローカルサーチの基盤的手法となった．

続いて，ウェブ文書に対して地理的検索を行うためのより一般的な方法として，文書中のジオワードを用いて地理的検索を実現する方法を提案した．ここでは文書中のジオワードを取り出して緯度経度に変換し，文書に緯度経度を付与して空間索引付けをし，空間問い合わせを可能とする方法を明らかにした．この検索手法を住所境界における検索の問題解消という観点から評価すると，従来の住所方式で存在した 25% 程度の検索もれを解消できることを明らかにした．この手法はモーバイルインフォサーチ実験として，1998 年 6 月から 2003 年までインターネット上で公開され，ローカルサーチを先駆的に可視化した．

論文の後半では，検索履歴データから地理特性を取り出す手法を提案した．この手法は履歴に含まれるジオワードを地理的に処理して空間的相関ルールを導く方法を定義し，地域に対して単語を推薦するものである．この提案は，検索回数が少ない地域に対しても，ジオワード間の地理的距離を用いることによりクラスタリングを行いサポート値の調整を可能とする一般化空間相関ルールのをマイニングできることを特徴としている．提案手法

は9,000万件の実検索ログを用いた実験により，従来の単純な住所階層を用いる手法に比べ，多様な地域でより多くの興味深いルールの抽出が可能であることを明らかにした．

さらに本論文ではジオワード・マイニングを行うための2つの固有名詞解析手法を提案している．一つは住所の省略によるあいまい性の解消である．従来困難であった同名異地住所問題が，検索履歴においてはセッション情報を用いることで，省略された上位住所の推定でき，解消できることを示す．本手法は評価実験により，頻出ジオワード10,000の3%に達する同名住所の問題を解決できることを報告した．もう一つは固有名詞の表記のゆれの解消である．3,000万の日本人姓のかな表記を分析した結果，姓は約9万通りのゆれ単位に分類可能であることを明らかにし，ゆれの単位の同値規則から自動的に正規化規則を作成する方法を考案し，この正規化を用いた検索では完全一致検索時に1検索あたり15%存在していた検索もれを，93%という高い適合率を達成しつつ解消できることを示した．

## 7.2 今後の研究課題

### ローカルサーチに関する課題

ここまでの本研究を含めた全般的な取り組みから，一般のウェブ文書を収集して地理的な検索システムを構築するためには，情報統合の単位を定める固定的なデータ（目録情報）の存在が必要であることは明らかである．現在の商用のローカルサーチは目録情報として，イエローページデータを用いている．彼らの考えを最も積極的に表現する意見には「イエローページはウェブコンテンツを地理的に検索可能とするだけでなく，インターネット情報に信頼や公平さをもらたす」[Himmelstein 2005]がある．

一方，イエローページを使うアプローチの限界を指摘する声も少なくない．「ローカルサーチはウェブ上のシティガイドに過ぎない．多くのウェブ上の有用なローカル情報が見落とされている」[Tezuka 2006]．「ローカルサーチは職業別電話帳に依存しており，電話帳情報にない情報は扱われない，郡や州といった広い地域に対する情報を扱うモデルがない」[Markowetz 2005]などの指摘がある．

ウェブ情報の統合の目録にイエローページを使う利点は，正確に情報を整理できることであり，このことによって一般消費者レベルでも満足できる地図付きのサービスが実現されたといえる．その上でこのアプローチの有効な領域は次のように整理できる．

1. データの実体が目録に存在するものに限られる
2. データ表現構造が目録が持つスキーマに限られる

ローカルサーチが有効な範囲を拡張して行くためには，上記の特徴を補強する必要があ

る．前者に対しては，著者は目録データをさらに拡充して行くべきであると考えている．拡充の自動化は重要な研究テーマであるが，検索の品質を担保するために，目録データの管理には人手を用いることも必要である．法人が人手で行う場合は，手間に対する対価を与えるビジネスモデルが必要である．またネットワークでコミュニティベースに目録データを構築する試みも存在する．Wikipedia はインターネットで参加者が自発的に情報を登録し，時には他の参加者と議論を交わしながら情報の質を高めるプロジェクトである [wikipedia]．現時点ではどのアプローチが適切であるかは議論のわかれるところであるが，ローカル情報を流通させる基盤としての目録情報の更なる構築と，それをベースとしたより高度な情報統合手法の開発が必要である．

後者は，扱う情報が店やサービスといった電話帳的な法人格だけでなく，より一般的な事象を扱うための課題である．例えば「東京の秋葉原でメイドが流行っている」という事象をみても，「秋葉原」という地域の定義にも幾通りもの範囲がありうる上に，「メイド」という話題を予め目録として登録しておくことも困難で，一般的な事象を目録ベースで解決することには限界がある．著者は本課題をウェブマイニングによって発見的に解決ことができると考えている．幸い現在ブログ等の個人情報発信インフラも整ってきており，情報源は充実している．本論文の第 5 章で述べた地域のクラスタリング手法と，ウェブコンテンツのマイニング，ウェブコミュニティのマイニングを組み合わせることにより，活きたローカル情報を取り出す仕組みの検討を進めたい．

## ジオワード・マイニングに関する課題

本研究で提案した地理的なサポート値の調整を可能とする一般化空間相関ルールマイニング手法の実システムへの適用を行いたい．本手法は検索システムの単語推薦だけでなく，他の機能にも用いることができると考えている．ローカル情報はいわゆるロングテールと表現される問題，すなわち多種の情報がそれぞれ少ない回数利用される状況にあり，その中で適度な閲覧回数が保証できる情報提供先のセグメントを地理性を考慮して定義できることが必要である．本マイニング手法の有効性を見極め，広告などの情報を提示するシステムへの適用を検討したい．さらにログデータのマイニングは時系列に飛び込む大量のデータの解析である．各地域のローカル情報が頻繁にストリーム的に配信される状況も遠い未来のことではない．このようなストリーム的なローカル情報に関しても空間的に知識を取り出すためのマイニング手法への一般化を検討していきたい．