検索結果を地域で整理する百科事典テキスト検索のための地名情報抽出法

金田 泰

日立製作所中央研究所
E-mail: kanada@crl.hitachi.co.jp

あらまし 「テーマ地図検索」というテキスト情報検索法を開発した．この検索法においては，ユーザは検索のテーマを自由語入力し，地名をふくむ文の抜粋とその文へのハイパーリンクのソートされたリストをえることができる．このリストを使用してユーザはその地名の位置をしめす地図をひらくこともできる．この検索のための地名インデクスを生成するため，地名抽出法を開発した．この方法においては，地名を抽出してデータベース中の地名とマッチングし同定する．地名には数種類のあいまいさがある．あいまいさは一種の文脈解析や他のいくつかの技法によって解決する．その結果，世界大百科事典においては 96% 以上の抽出精度を実現した．情報抽出のための規則は日本語の特徴に依存しているが，その戦略は他の言語にも適用することができる．

キーワード テキスト検索，情報検索，情報抽出，地名抽出，地名同定，情報組織化，情報整理

A Method of Geographical Name Extraction from Japanese Encyclopedia for Text Search in which the Results are Ordered by Geographical Areas

Yasusi Kanada

Central Research Laboratory, Hitachi Ltd.
Higashi-Koigakubo 1-280, Kokubunji, Tokyo 185, Japan
E-mail: kanada@crl.hitachi.co.jp

Abstract A text retrieval method called the thematic mapping search method has been developed for Japanese texts. In this method, the user specifies a search theme using free words, then obtains a sorted list of excerpts and hyperlinks to sentences that contain geographical names. Using this list, the user can open maps that indicate the location of the names. To generate an index of names for this searching, a method of geographical name extraction has been developed. In this method, geographical names are extracted, matched to names in a geographical name database, and identified. Geographical names, however, often have several types of ambiguities. Ambiguities are resolved using context analysis and several other techniques. As a result, the precision of extracted names is more than 96% on average when applied to the World Encyclopædia. The rules for information extraction depends on features of the Japanese language, but the strategy and most of the techniques can be applied to texts in English or other languages.

key words Text search, Information retrieval, Information extraction, Name extraction, Name identification, Information organization

1. はじめに

インターネット, CD-ROM, DVD-ROM などのメディアが普及し, 従来のデータベース検索におけるようにプロのサーチャが検索するのではなく, エンド・ユーザが直接, 従来よりはるかに大量のテキストの全文を検索するようになってきている. 大量のテキストを検索すれば, 当然, 大量の検索結果がえられる. このような背景のもとで, 情報検索には整理された検索結果がえられることがもとめられるであろう. 検索結果がおおいとき, それが整理されていなければその全体をサーベイするには長時間を要する. しかし, 検索結果がうまく整理されていれば, ユーザは単純な検索条件でおおくの項目をもとめてサーベイし, 有用なものを選択できる. 適当な条件をかさねると検索結果をしぼりこめるばあいでも, このような組織化の機能は重要である. なぜなら, しぼりこみによって, ユーザにとって重要な一部の情報もすてられてしまうからである. ユーザは, 有用でありうる周辺的な検索結果にふれられないままになってしまう.

ユーザの意図にそって検索結果を組織化することが, この問題を解決するために重要だとかんがえられる. そこで, 組織化をともなう検索法の開発の第 1 歩として, 軸づけ検索法 [Kan 98][Kan 98a] を開発した. この方法においては, ユーザは全文検索結果を組織化するための「軸」を選択する. **テーマ年表検索** [Kan 99][Kan 99a] は「年代」を軸とする軸づけ検索である. この報告であつかう**テーマ地図検索**[1]は「地域」を軸とする軸づけ検索であり, 地理的情報をふくむテキスト集合からあるテーマに関する情報を検索し, 地域によって結果を整理する検索法である. テーマ地図検索においては, ユーザからの要求をうけとるまえにテキスト集合中の全テキストを走査して地名を抽出し, 地名データベースの登録地名とマッチングをとって地名を同定し, 地名インデクスに登録する. テキスト集合とデータベースのいずれも数種類のあいまいさをふくんでいるので, この地名抽出における最重要な仕事はあいまいさの解消である.

地名抽出もふくめ, さまざまな名詞や数値の情報抽出が研究されている (e.g, [MUC 98] [Ino 96] [Sai 98] [Tak 99] [His 97]) が, 名詞抽出の研究の大半は未知の名詞の抽出法に関するものである. 抽出された既知の固有名詞をデータベース中の名詞と比較し同定する方法は確立されていない. この論文においては, テーマ地図検索において地名を抽出し, 文脈情報をつかってあいまいさを解消し同定する方法について説明する. まずテーマ地図検索の概要を説明し (第 2 章), 地名抽出の基本方式について説明する (第 3 章). 地名のあいまいさを分析し (第 4 章), あいまいさを解消するための技法を説明する (第 5 章). さらに, あいまいさ解消の性能と地名抽出の精度とを評価する (第 6 章).

[1] http://www.hdh.co.jp/information/-net.html (日立デジタル平凡社)

2. テーマ地図検索の概要

テーマ地図検索は軸づけ検索の機能の一部を具体化したものなので, まず軸づけ検索について説明し, テーマ地図検索について補足する.

軸づけ検索においては, ユーザはメニューによって軸を選択し, 検索語を入力する. 検索語は検索のテーマを指定し, 軸が検索結果を整理する方法を指定する. 検索エンジンはその語に関する全文検索結果をえて, 軸にそってソートする. 抽象的にいえば, 検索結果は軸によって指定される空間に配置される. 検索結果を組織する基準は, クラスタリングにもとづく組織化法においてはボトム・アップにきまるが, 軸づけ検索においてはユーザによって指定されるため, ユーザの意図にそった整理が実現される. 軸上の範囲もユーザが指定することができ, 範囲内の検索結果だけが表示される.

軸づけ検索の実際のユーザ・インタフェースの例を図 1 にしめす. これはテーマ地図検索のインタフェースであり, Microsoft Windows 95, 98 and NT 4.0 上で動作する. ここでは説明のために単純化した図 2 のインタフェースを使用する. ユーザは「一揆」という検索語を入力し, 軸として「地域」を選択し, 範囲として「日本」を指定して百科事典を検索したとする. テーマ地図検索においては, メニューから「テーマ地図検索」を選択することにより, 軸として「地域」が選択される. 地域範囲を選択するメニューは「世界」, アジア, アフリカなどの各地域, 世界の個々の国, 日本の個々

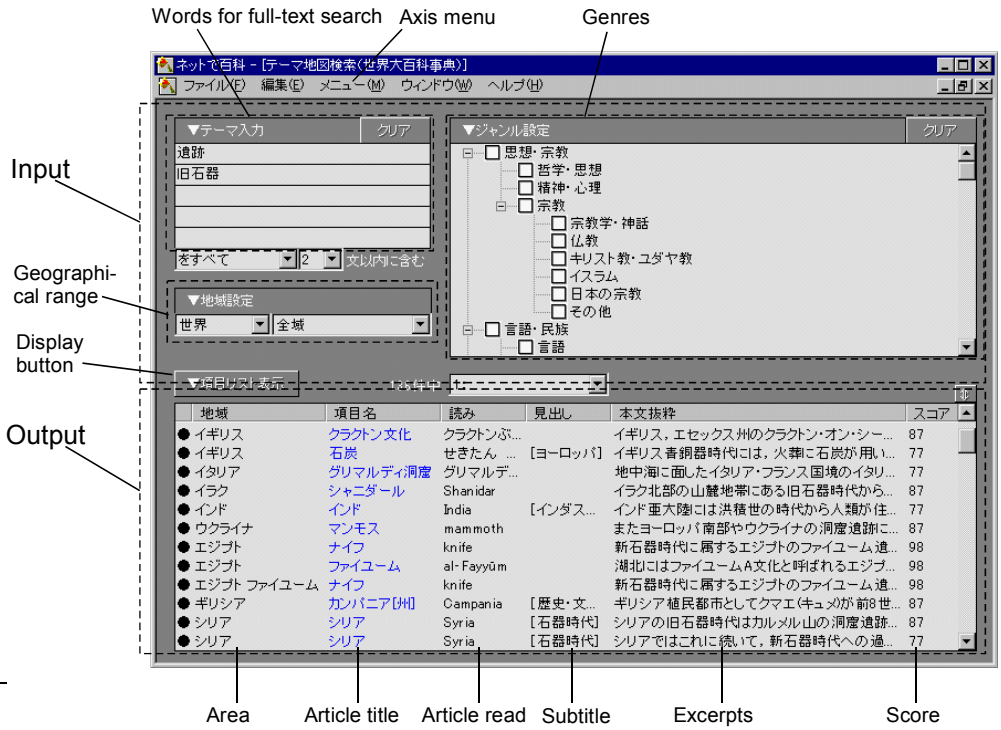

図 1 テーマ地図検索 (地域を軸とする軸づけ検索) のユーザインタフェース

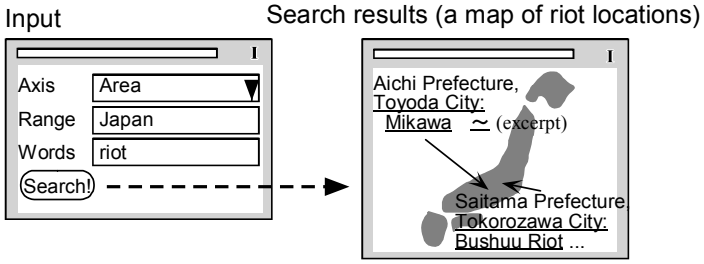

図 2 地域を軸とする軸づけ検索の機能

の県などをふくむ．検索ボタンをおすと，百科事典の各項目から検索テーマに関連する文が抜粋され，その近傍に出現する地名にもとづいてソートされて表示される．あるいは，図 2 のように各検索結果項目を地図と関連づけて表示することも原理的には可能である．図 2 においては「三河」と「武州一揆」が百科事典の項目名であり，抜粋された文がそれにつづいている．この文にはハイパーリンクがうめこまれているので，それをクリックすれば事典項目がひらき，その文の周辺や項目全体をよむことができる．また，テーマ地図検索の特徴のひとつは，当該の地名を地図上で確認できることである．

軸づけ検索のシステム・アーキテクチャ [Kan 98] [Kan 98a] を図 3 にしめす．システムは 2 つの主要な部分つまりインデクス生成部と検索エンジンとで構成される．インデクス生成部は軸ごとに存在する軸インデクス生成部と全文インデクス生成部とで構成される．軸インデクス生成部は既定のマッチング・パターンを使用して，テキスト集合から地名など，軸上にのる値を抽出して正規化し，インデクスに登録する．地名抽出法は第 3 章で説明する．全文インデクス生成部は従来の全文検索と同様のインデクスを生成するが，登録単位は文書ではなく文 (ただし長文はカンマで分割する) である．検索エンジンはユーザによって起動される．検索エンジンは，指定された範囲の地名をふくむ文を地域軸インデクスを使用してもとめる．結果は軸にそってソートし，結果項目ごとにスコアをもとめ，スコアがひくすぎる結果はすてられる．

3. 地名抽出法

テーマ地図検索における地名抽出法について説明する．

3.1 地名抽出の概要

地域軸インデクスの生成においては，すべてのテキストが走査され，GDB に登録された地名にマッチする文字列が抽出される．抽出文字列は正規化され，地域軸インデクスに登録される．GDB は世界大百科事典 [HDH 98] の地図のために開発されたものである．これまでに世界大百科事典とマイペディア [HDH 99] にこの地名抽出法を適用する実験をおこなった．前者についていえば，テキストは 160 MB, 項目数 (文書数) は 84,000 である．そこから抽出された地名は，出現回数において日本地名は約 130,000，世界地名は約 340,000 である．

抽出プロセスはマッチした文字列の直前・直後のテキストをテストし，その文字列を地名として抽出するべきかどうかを判定する．この局所的な文脈マッチは文字列単位でおこなう．形態素解析，構文解析などの自然言語処理はおこなっていない．いくつかの地名は文脈自由な規則だけをつかって抽出できる．しかし，複数の地域に同一の地名が存在するばあいは，文脈依存の規則をつかって同定する必要がある．マッチング・パターンと正規化の規則とは検索対象のテキストの性質によってかえる必要がある．

3.2 抽出地名の構造

テーマ地図検索においては地名は 2 階層の構造をしているものとしてあつかっている．上位階層は，世界のときは国名または国名相当の地域名，[1] 日本のときは県名である．下位階層は本文に出現する地名の階層であり，都市名，地方名，山名，湖名など，さまざまな種類がある．本来，地名は，たとえば「東京都 中野区 弥生町」のように 3 階層以上の構造をしていることがおおい．しかし，多階層の地名を，省略形もゆるしたうえでただしく抽出するのは困難な仕事である．今回は表示がコンパクトでただしい (適合率がたかい) ことを優先して，抽出地名は基本的に 2 階層以下にかぎることにした．したがって，「東京都 中野区 弥生町」のかわりに「東京都 中野区」または「東京都 弥生町」が抽出される．

3.3 地名データベースの使用

テーマ地図検索においては，日立デジタル平凡社においてつくられた，地名とそのよみ，属性，その上位の地名などを登録した地名データベース (GDB) を使用している．属性としてはさまざまなものがあるが，代表的なものは「国」，「県」，「山」，「川」，「市」，「町」などである．複数の行政区域にまたがる山や島などにおいては，ひとつの下位地名に対して複数の上位地名が存在するばあいがある．GDB の一部を図 4 にしめす．GDB への登録件数は，日本の地名が約 55,000 件，外国の地名が約 41,000 件である．これらはすべて現在の地名であり，歴史的な地名は登録されていない．

GDB においては地名に識別番号がふられている．それを世界大百科事典の地図 API (application programming interface) にわたせば，地図をひらき，その地名の位置をしめすことができる．テーマ地図検索では，本文から地名を抽出するまえに GDB の内容をよみこみ，メモリ上に内部データベースを作成し使用している．

テーマ地図検索においては，GDB に登録された地名だけを百科事典から抽出している．辞書やデータベースに登録されていない未知地名の抽出が米国を中心にさかんに研究されているが，テーマ地図検索においては未知の地名は価値がひくい．なぜなら，テーマ地図検索においては抽出した地名の所属地域を特定して地図へのリンクをもとめる必要があるが，未知の地名は所属地域がわからず，それをふくむ地図は表示できないからである．また，百科事典の検索として許容できるレベルまで未知地名の抽出精度をあげるのは困難だとかんがえられる．したがって，テーマ地図検索においては GDB に登録された既知の地名だけを抽出するよう

[1] 国名相当の地域名の例としては，「南極」，「西インド諸島」，「西サハラ」，「グリーンランド」などがある．

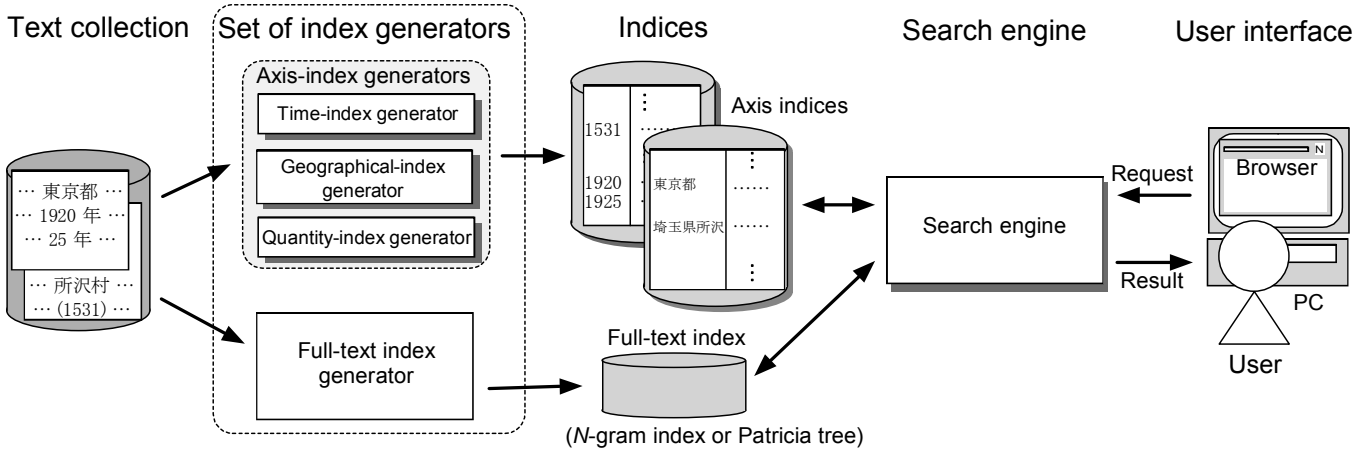

図 3 軸づけ検索のためのシステム・アーキテクチャ

にしている.

ID	地名	よみ	属性	上位地域コード	優先度
1	竹島	たけしま	島	32	
2	日本海	にほんかい	海洋	101	
3	宗谷海峡	そうやかいきょう	海峡	1	5
…	…	…		…	…
8	南西諸島	なんせいしょとう	諸島	46 47	
10	九州	きゅうしゅう	島	101	

図 4 世界大百科事典地図データベースの一部

3.4 地名のマッチング法

地名をマッチするアルゴリズムを図 5 にしめし, 以下で説明する. 関数 *extract* はテキストと GDB を入力して (I_i, S_i) という対のリストを出力する. ここで I_i は地名の ID であり, S_i は地名が出現する文の ID である. コンテクスト・スタック C と関数 *identify* におけるそのあつかいかたについては 5.2 節において説明する.

地名をテキストから抽出するとき, テキスト上の任意の位置から GDB 中の地名とのマッチングを開始することをゆるすと, 誤抽出確率がたかくなる. 形態素解析を導入して形態素の先頭からだけマッチすることも可能だが, 形態素解析の結果は通常数%のあやまりをふくみ, 計算コストもかかる. また, 百科事典に多出する外字を形態素解析においてあつかうのはやっかいである. 形態素解析をつかわないとマッチング・パターンはやや複雑になるが, 総合的にはそれをつかうのと同程度の精度をより低コストで実現できると判断した. そこで, テーマ地図検索においては, 形態素解析をつかわず, 字種間の繊維点, たとえばひらがなからカタカナや漢字に, あるいは特殊文字 (記号) から漢字に文字種が変化する点をマッチング開始点とした. 抽出地名の末尾もマッチング開始点として使用する. しかし, 語の境界で文字種が不変なときもあるので, 地名の直前にしばしば出現する「年」,「月」,「市」,「町」,「村」,「前」,「後」などの文字の直後もマッチング開始点とした.

先頭がひらがなである地名の直前に助詞があるばあいなどは, 語の境界で文字種がかわらない. ひらがなで記述される地名はすくないため, ひらがなについては文字種の変化がなくてもマッチング開始点としている.

例として, 世界大百科事典の「揖保川」という項目の一部におけるマッチング開始点を縦棒でしめす.

> |兵庫県|西部, |鳥取|と|の|県境付近|に|発|し, |姫路市|網干|で|播磨灘|に|そ|そ|ぐ|川。|幹川流路延長 70km, |全流域面積 810km^2。

```
function extract(Text) return X;
input   Text: the text of an encyclopedia article;
output  X: the geographical axis index that contains pairs (I_i, S_i),
where I_i is the identification number of a geographical name, and S_i
is the sentence identifier in which the geographical name occurs;
global  GDB: the geographical name database;
begin
   make context stack C empty;
   make index X empty;
   for each sentence S in Text (from first to last) loop
      for each matching starting point in S
                (from left to right) loop
         N := the name spelling that matches to a name in
               GDB by the longest coincidence method
            using GDB (if no name matched, N becomes nil);
         if N is not nil and
               the suffix or prefix of N indicates that N is
               not a geographical name then
            N := nil;
         end if;
         if N is not nil then
            N := normalize(N);
               -- Normalize the spelling.
               -- N will be the normalized name.
            I := identify(N, C);
               -- Identify the name.
               -- I is the identification number.
            if I is not nil then
                        -- The identification succeeded.
               add (I, S) to index X;
            end if;
         end if;
      end loop;
   end loop;
end;
```

図 5 地名抽出手続き (主部)

地名は最長一致法によってマッチしている. もしテキスト中のある文字列にマッチする地名が GDB に 2 個以上登録されていると, ながいほうの地名が選択される. 抽出された地名に対しては接尾辞・修飾語句等のテスト (5.3 節参照) をおこなう. このテストに合格すると関数 *normalize* (次節参照) によって地名は正規化される. そして, 関数 *identify* によって GDB にふくまれる地名と同定される. 同定とは, 抽出された地名に対して GDB における識別番号 (ID) がふられるということを意味している. ID がふられると図 6 においては結果リストに登録されるが, 実際には高速アクセス可能なインデクスに登録される.

3.5 別名の正規化

地名にはしばしば別名が存在する．たとえば，「中華人民共和国」に対する「中国」，「アメリカ合衆国」に対する「米国」，「イギリス」に対する「英国」，「大韓民国」に対する「韓国」などがある．また，本来の意味の別名ではないが，それに準じるものとして「プエルト・リコ」に対する「プエルトリコ」，「米領北マリアナ諸島」に対する「アメリカ領北マリアナ連邦」などがある．テーマ地図検索においては百科事典本文にこのような別名があらわれたときも抽出し，正規の名称に変換してあつかう．図 5 においては，関数 *normalize* において別名が登録され，その出現時には正規化された地名でおきかえられる．

4. 地名情報のあいまいさ

地名情報がふくむあいまいさと分類し，その例をしめす．

4.1 つづりがひとしい，ことなる地名

ことなる地名だがつづりがひとしいものが存在する．日本ではたとえば「荒川」という川が関東南部以外に北海道，青森，福島，新潟などに存在する．アメリカには多数の同名のまちが存在し，さらにイギリス，オーストラリアとのあいだにも同名のまちが存在する．たとえば「プリンストン」という地名は全米で 6 個，存在する．また，よく知られた例として，ニューヨークは都市名として存在すると同時に州名としても存在する．また，ワシントンは首都名として存在するとともに州名としても存在する．

あいまいさをふくむ 3 つの文例をしめす．これらは世界大百科事典における記述をちぢめたものである．最初の例は「オハイオ [州]」という項目に由来する．

> アメリカ合衆国中西部の州．バージニア州と並んで最も多くの大統領を生み出した．同州にはコロンバス，シンシナティ，デートンなどの大都市がある．

この例においては，コロンバスが属する州つまりオハイオ州が項目名として出現している．コロンバスに関するあいまいさはこの情報をつかえば解消することは可能である．しかし，バージニアという州名が出現するため，それがあいまいさ解消をじゃまする可能性がある．

第 2 の例は「アレクサンダー (Franz Alexander)」という項目の一部の記述を短縮したものである．

> ハンガリー生れの精神分析者．1930 年アメリカに渡り，ボストン，シカゴ，ロサンゼルスなどで精神分析を教えた．

この例ではボストンが属する州名が出現しない．そのためこのボストンと英国のボストンとを誤認する可能性がある．しかし，「アメリカ」という国名が出現し，GDB には米国にあるマサチューセッツ州以外のボストンは登録されていないため，このあいまいさは解消可能である．

第 3 の例は「アメリカン・フットボール」という項目の一部の記述を短縮したものである．

> アメリカの学生がサッカーやラグビーをもとに考案したチーム競技．バラ祭で知られるロサンゼルス郊外パサデナの〈ローズ・ボウル〉はもっとも歴史が古い．

この例では，アメリカには 2 つのパサデナがあるため，パサデナに関するあいまいさは「アメリカ」という国名の出現によって解消されない．しかし，ロサンゼルスの所属州であるカリフォルニアを文脈としてつかえば，カリフォルニアのパサデナと同定することが可能である．

4.2 地名以外の固有名詞や普通名詞とひとしい地名

地名が地名以外の固有名詞とひとしいばあいがある．地名が人名をもとにしてつけられたとき，たとえば「ワシントン」がそうである．また，普通名詞とひとしい地名も存在する．たとえば，中国には「平和」，「運河」，「東西」，「東方」などの地名が存在する．また，イギリスには「ニュータウン」，「プール」といった地名が存在する．これらをつづりだけにもとづいて抽出すると，地名として抽出された名詞のおおくが地名以外の固有名詞や普通名詞という結果になる．

4.3 GDB の不完全さ

地名データベースは人手で入力された大規模データベースであり，実世界の政治や地理的条件の複雑さを反映している．したがって，完全なものからはほどとおく，あやまりをなくすことは不可能にちかい．また，世界大百科事典の GDB の開発目的は地名抽出ではないので，地名抽出のために理想的な性質をもったものだとはいいがたい．たとえば，ひとつの地名が GDB に複数回出現するばあいもあり，それぞれが完全には一致しない属性をもっている．このようなことは，たとえば 2 個のことなるタイプの地図の両方にある都市が出現するようなばあいにおこる．このような不完全性があるばあいには，地名抽出のアルゴリズムはそれにたえるものでなければならない．

4.4 他のあやまり

地名抽出や GDB における他の種類のあやまりやあいまいさもありうる．たとえば，語の境界の解析不良のために語の一部が地名とみなされることもありうる．すなわち，形態素解析をおこなうかわりに字種判定などによって語の境界をきめているので，判定が不適切なためにあやまった地名を抽出することがありうる．

5. 地名のあいまいさの解消

地名のあいまいさの解消と誤抽出低減の技法についてのべる．まず地名の同一性についてのべ，規則ベースおよび事実ベース (辞書ベース) の方法をしめす．

5.1 地名の同一性

4.3 節でのべたように，GDB においては同一の地名をあらわす複数のレコードが存在するばあいがある．テーマ地図検索においては，おなじつづりの地名をふくみ上位地名もひとしく，矛盾する情報をふくまない GDB のレコードは，同一の地点をあらわすものとみなしている．この方法でことなる地点が同一視されることがないとはいいきれないが，すくなくとも現在の GDB に関してはそのようなあやまった同定がおこることはまれである．

5.2 非局所的なコンテクストにもとづく地名同定

あいまいな地名を同定するためには文脈を把握する必要がある．たとえば，「一宮町」という地名が山梨県に関する記述のなかにあらわれれば，それは山梨県一宮町を意味している確率がたかい．また，兵庫県に関する記述のなかにあらわれれば，それは兵庫県一宮町を意味している確率がたかい．しかし，テーマ地図検索においては自然言語の意味や構文の解析はおこなっていない．それらを部分的に解析することは可能だが，完全な解析は現在の技術では不可能である．したがって，はるかに単純な方法によって文脈を把握し，あいまいな地名を同定している．この方法は 4.1 節でのべた例題におけるあいまいさを解消するのに十分である．

その方法は図 6 のとおりである．地名抽出のためにテキストを左から右へ走査するが，そのとき国名，県名または州名 (米国のとき) を文脈スタックとよぶ配列 *C* にスタックする．[1] スタックのふかさは 5 程度に制限し，それをこえると左端の地名をすてる．県名や国名がテキストにあらわれると *C* に格納する．県名や国名より下位の地名があらわれると，上位の県名や国名を *C* に格納する．

もしテキスト中の地名にあいまいさがなければ，上位の地名は文脈を参照せずに同定できる．県名や国名が抽出されたときには，文脈はテストされない．あいまいな地名が出現したときは，それと同定するべき地名候補の上位地名が文脈スタックに格納された地名やその上位地名と比較され，ひとしい地名が選択される．比較の順序は関数 *context_stream* によって制御されている．この関数は上位地名をストリーム (またはリスト) としてかえす．もし上位地名のひとつが 2 個以上の地名とマッチするときには，右端のもの (スタック中でより最近格納したもの) が上位地名として選択される．

関数 *context_stream* の定義を図 7 にしめす．もし文脈スタック *C* 中の地名が米国地名ならそれが属する州名をストリームにいれ，その地名が日本地名ならそれが属する県名をストリームにいれる．しかし，もしその地名が州名か県名ならば，それはストリームにいれない．つぎに国名をストリームにいれ，最後に「北アメリカ」,「アジア」というような大域名をストリームにいれる．関数 *identify* においては，もし文脈スタックがふくむ地名が日本や米国のものであれば，地名 *N* をまず県名や州名と比較する．つぎにそれを国名と比較する．最後にそれを大域名と比較する．この比較順序は改良の余地があるが，この順序にしたがえば，すくなくとも世界大百科事典においては，まちがった同定はほとんどさけられる．

4.1 節の最初にしめしたオハイオ州の例に関するあいまいさ解消のプロセスを説明する．文中の「コロンバス」を処理する直前の抽出地名をリストする:

(スタック下位) **オハイオ [州]**, **アメリカ [合衆国]**, **オハイオ [州]**, **バージニア [州]** (スタック上位).

```
function identify(N, var C) return I;
input    N: a name spelling;
output   I: an identification number of the name;
input/output C: the context stack;
global   GDB: the geographical name database;
begin
   if N denotes a unique name (i.e., it is context-free) then
      I := the only identifier;
   else       -- Ambiguity exists.
      I := nil;
      for each element A in context_stream(C) loop
         G := the set of name identifiers I1, I2, …, Im,
              whose spelling is N and whose upper-layer
              names include A (using GDB).
         if the number of elements of G is 1
              (i.e., there is no ambiguity) then
            I := the element;
            exit loop;
         end if;
         if only one of the names specified by G has the
              highest priority value then
            I := the identifier of the name;
            exit loop;
         end if;
         -- (1)
         if G is not empty then  -- Ambiguity not resolved.
            exit loop;  -- I is nil.
         end if;
      end loop;
   end if;
   if I is not nil then
      push each upper-layer name of I into C
         only when it is not duplicated;
   end if;
   return I;
end;
```

図 6 地名同定手続き

```
function context_stream(C) return L;
input       C: a context stack;
output   S: a stream or list of upper-layer names;
begin
   S := empty;
   for each element A of C (from top to bottom) loop
      if A belongs to the US then
         put the state of A into S when it is not duplicated;
      else if A belong to Japan then
         put the prefecture of A into S
            when it is not duplicated;
      end if;
   end loop;
   for each element A of C (from top to bottom) loop
      put the country of A into S when it is not duplicated;
   end loop;
   for each element A of C (from top to bottom) loop
      put the global area of A into S
         when it is not duplicated;
   end loop;
   return L;
end;
```

図 7 テスト文脈ストリーム生成手続き

ここからえられる州名は:

オハイオ [州], **オハイオ [州]**, **バージニア [州]**.

これらをストリーム *S* にいれ，*context_stream* の値の一部とする．つぎに，国名と大域名とを *S* におく．重複は排除されるので，*S* の値はつぎのようになる:

(バージニア [州], **オハイオ [州]**, **アメリカ [合衆国]**, **北アメリカ)**

マッチングは最近 (右端) のものから過去 (左端) のものへという順序にしたがう．そのため，まずバージニア州にコロンバスという地名があるかどうかをしらべる．それは存在しないので，オハイオ州にあるかどうかをつぎにしらべる．それは存在するので，この地名は「オハイオ州コロンバス」と同定される．同定の結果はこのばあいはマッチングの順序に依存し

[1] 文脈処理の方針は言語につよく依存してはいないが，語順には依存している．上位の地名 (たとえば「米国」) が下位の地名 (たとえば「ハワイ」) より日本語ではさきにくるが，英語では逆順になる．現在はテキストはほとんど厳密に左から右に走査しているので，この方法を英語のテキストに適用する際にはスタックする順序を変更する必要がある．

ない．しかし，もし文脈スタックが「ジョージア」をふくんでいれば，そこにはコロンバスが存在するので，結果は比較順序に依存する．もしジョージアがオハイオより左 (現在位置からとおい位置) にあらわれれば，やはり「オハイオ州コロンバス」と同定される．しかし，逆順であれば「ジョージア州コロンバス」と同定される．4.1 節における他の 2 例における各地名も，この方法でただしく同定される．

GDB は地名間の相対的な重要性をしめす優先度をふくむ．もし抽出地名があいまいでその候補がことなる優先度をもっていると，より優先度のたかい候補を選択する．

5.3 接尾辞・修飾語句等によるフィルタリング

地名と人名・組織名とのくべつがつかないとき，抽出された名詞の前後につく接尾辞，接頭辞や修飾語句をしらべることによって，くべつできるばあいがある．たとえば，「大統領」，「党」，「兄弟」などの直前の語は地名ではないと判定できる．このように固有名詞の直前・直後にくることばのリスト (網羅的ではない) を図 8 にしめす．しかし，地名以外の固有名詞につねにこのようなことばがつくとはかぎらないので，補足的な方法である．

(1) 直後につくことば
A, B, …, Z, 0, 1, …, 9, 》, 〉, “ ”, 語, 人, 家, 氏, 法, 属,
目, 派, 党, 賞, 大学, およびこれらの前に「両」，「各」，「諸」，数値がついたもの,
大統領, 首相, 総督, [大]司教, [大]主教, 男爵, 子爵, 内閣, 政権,
[一]族, 兄弟, 姉妹, 主義, 時代, 報告.
[会]社, 銀行, 商会, 商店, [街]道, [大]聖堂,
記念, 変動, 広場, 流, 的, 科, 宗, 教, 寺, 学, 炉, 病, 様, 伯, 卿, 公,
朝, 号, 著, 邸, 荘, 殿, 廷, 司, 院, 塔, 塚, 軍, 隊, 群, 角, 川, 章, 座,
飴, 区, 星, 期, 師, 銃, 鉱, 屋, 々.

(2)直前につくことば
家 (「国家」をのぞく), 大統領 , 首相,
総督, 提督, 監督, 将軍, [大]司教, [大]主教,
民族, 諸族, 大将, 中将, …, 大佐, …, 小尉,
艦, 党, 者, 長, 人, 公, 機, 師, 夫, 妻.

図 8 地名以外の固有名詞の前後につくことば

5.4 辞書にもとづく技法

5.2～5.3 節でしめした規則にもとづく技法だけでは自然言語で記述された地名情報のあいまいさを解消するのに十分ではない．そこで，つぎの 3 つのような辞書 (事実) にもとづく技法を併用する．

1. GDB 情報の修正．GDB がふくむ情報の一部を修正，追加，あるいは削除することをかんがえる．
2. テキスト中の地名に XML や SGML の タグを付加する．タグのなかに同定された地名の識別子をいれる．
3. GDB と小規模の補足的データベースまたはリスト (一種のパッチ) とから，地名抽出の直前にメモリ上にデータをコピーして専用のデータベースをつくる．

1. は非常に直接的な方法だが，GDB はもともとテーマ地図探索を目的として開発されたものではないので，それに手をいれるのは困難である．2. はテキストが他の目的にもつかわれるときは適用困難である．3. はデータベースの構造を複雑化させるが，おなじテキストや GDB をつかう他のプロジェクトに影響をあたえないという利点がある．我々は第 3 の方法をつかっている．おもにつぎのような 3 つの補足的なリストをつかっている．

- GDB に出現する可能性があるが専用データベースには登録しないつづりのリスト．このリストをつかって登録地名を削除することにより，固有名詞や普通名詞を地名と誤認することがある程度ふせげる．
- GDB 中のレコード識別子のリスト．このリストによって指定されたレコードは専用データベースに登録しない．もし GDB 中の地名がノイズとしてだけふるまうばあいは，このリストをつかってレコードを削除する．
- GDB 中のレコードを完全に，または部分的におきかえるレコードやその一部の集合．もし GDB における地名の優先度が地名抽出に適当でなければ，この集合をつかうことによってかきかえることができる．

これらの方法によって抽出精度をおおきく改善できるばあいもあるが，これらはアドホックである．これらにたよりすぎると，検索対象のテキストにあらたな地名が登録されたときには，インデクスの質を低下させるであろう．

6. 評価

2 つの評価の結果をしめす．

6.1 あいまいさ解消の性能

あいまいさ解消の性能評価結果を表 1 にしめす．評価方法はつぎのとおりである．GDB に 5 回以上出現する地名つづりから日本と外国の地名つづりを選択して 図 9 のリストを生成した．地域軸インデクスに登録したそれらの地名の全出現をテーマ地図検索によって検索し，全検索結果を人手でチェックしてあやまりをかぞえた．

表 1. あいまいさ解消の評価結果

地名分類	地名数	全抽出数	誤抽出数	精度
日本地名	48	633	103	0.84
外国地名	34	1139	98	0.91

(1) 日本地名
愛宕山, 一番町, 烏帽子山, 横島, 観音崎, 吉野町, 境川, 錦町, 月山, 原町, 御岳, 広瀬川, 荒川, 高森山, 高島, 黒岳, 黒川, 今町, 三国岳, 三和町, 山田町, 若松町, 春日町, 小川町, 松山町, 焼山, 新川, 清水町, 赤川, 相生町, 大岳, 大手町, 大川, 大峠, 大和町, 茶臼山, 中央区, 中津川, 天狗岳, 南田町, 日の出町, 鉢伏山, 平島, 弁天島, 明神山, 野島, 矢筈山, 有明町

(2) 外国地名
アーリントン, アバディーン, アレクサンドリア, ウィルミントン, ウィンチェスター, オールバニー, キングストン, ケンブリッジ, コロンバス, コロンビア, サン・カルロス, サン・フェルナンド, サン・ペドロ, サン・ルイス, サンタ・クルス, ジャクソン, スプリングフィールド, セーレム, チャールズタウン, ニューカスル, ニューポート, バーリントン, ブラック川, フランクリン, プリマス, プリンストン, ベルビル, ポーツマス, マリオン, マンチェスター, ラ・パス, ランカスター, リッチモンド, レバノン

図 9 評価に使用した地名のリスト

表 1 によれば，世界地名の精度がたかい (91%) が，日本地名の精度はややひくい．精度がひくい理由は，正解である地名が GDB に登録されていないばあいがあること，地名の

かわりに人名が誤抽出されているばあいがあること，そしてつづりがひとしい複数のなまえがしばしばひとつの県内にあらわれて，しかもそれをかんたんな文脈解析でくべつすることがむずかしいことである．しかし，世界地名，日本地名のいずれも，あいまいな地名をランダムに選択するのにくらべるとはるかに高精度である．すくなくとも 5 個はつづりがひとしい地名があるので，ランダムに選択すれば精度は 20% 以下になる．

6.2 抽出精度

5 個の検索タスクを使用して地名抽出精度の評価をおこなった結果を表 2 にしめす．この評価においては，地名情報のただしさは人手によって全条件の全検索結果を調査することによって判定した．「一揆」と「旧石器，遺跡」の検索に関しては精度は 98% 以上であり，ほぼ満足すべき結果である．しかし，「茶」と「ビール」の検索条件については 95% であり，百科事典の検索結果としては十分な精度だとはいえない．これらを平均すると精度は 96% 以上である．この精度低下は，「ビール」の検索においては「モビール」と「日本」，「茶」の検索においては「津」という，わずかな数の語によってひきおこされている．[1] もしこれらの語を除外すれば，精度は他と同様の値となる．なお，抽出精度があいまいさ解消の精度よりはるかにたかいのは，あいまいさ解消の評価においては故意にあいまいさを導入しているので，それにくらべると抽出精度の評価においてはあいまいさがすくないからである．

表 2. 地名抽出精度の評価結果

検索語	地域範囲	距離 (文単位)*	検索結果数	誤抽出数	精度
一揆	日本	2	641	13	0.980
茶	日本	0	376	20	0.947
ビール	世界	3	583	29	0.950
コンピュータ	世界	5	568	16	0.972
旧石器, 遺跡	世界	5	525	7	0.987
Total	-	-	2693	85	0.968

* 金田 [Kan 98] 参照.

7. 結論

テーマ地図検索の地名抽出・同定においてもっとも重要な問題が，あいまいさの解消である．文脈スタックを使用した非局所的な文脈の解析をふくむいくつかのあいまいさ解消法を適用することによって，平均で 96% 以上の抽出精度を実現した．これは百科事典の検索のために開発した方法であり，地名抽出規則やあいまいさの解消法においては言語依存の規則や方法をつかっているが，その戦略やおおくの規則は汎用性がある．したがって，これらの方法は他の種類のテキスト，すなわち新聞や，英語や多言語のテキストにも応用可能である．また，用途は検索に限定されない．

今後の課題として，地名抽出法の適合率・再現率の向上，他の種類のテキストへの適用などがある．

謝辞

以下の方々に感謝する．藤井氏ほか日立デジタル平凡社の方には世界大百科事典の GDB と世界大百科事典およびマイペディアの使用許可をいただいた，日立デジタル平凡社の織田，井上，足立各氏，日立製作所情報システム事業部 (当時) の荻原，平野両氏には地名情報抽出法の改良に協力していただいた．日立東北ソフトウェアの山崎，澤田両氏には開発した専用クライアントをつかわせていただいた．日立製作所ソフトウェア事業部の星氏には全文検索エンジンを改良していただいた．

参考文献

[Fra 92] Frakes, W., and Baeza-Yates, R., ed.: *Information Retrieval: Data Structures & Algorithms*, Prentice Hall, 1992.

[HDH 98] *CD-ROM/DVD-ROM 世界大百科事典 第 2 版*, 日立デジタル平凡社, 1998.

[HDH 99] *CD-ROM マイペディア 99*, 日立デジタル平凡社, 1999.

[His 97] 久光 徹, 丹羽 芳樹 : 辞書と共起情報を用いた新聞記事からの人名獲得, *情報処理学会 自然言語処理研究会*, 118-1, 1–6, 1997.

[Ino 96] 井上 裕二 他 : テンプレートを用いた新聞記事からの製品情報抽出システム, *情報処理学会 研究報告* 96-NL-115, 83–90, 1996.

[Kan 98] Kanada, Y.: Axis-specified Search: A New Full-text Search Method for Gathering and Structuring Excerpts, *3rd Int'l ACM Conf. on Digital Libraries*, pp. 108–117.

[Kan 98a] 金田 泰 : 軸づけ検索法 — 文書からの抜粋を抽出・整列して出力する全文検索法, *情報処理学会 情報学基礎研究会*, 98-FI-50-4, pp. 25-32, 1998.

[Kan 99] 金田 泰, 澤田 瑞穂, 山崎 幹夫, 平野 義明, 藤井 泰文 : 「ネットで百科」における「テーマ年表検索」の機能と実現法, *情報処理学会 第 58 回全国大会* 1J-03, 1999.3.

[Kan 99a] 金田 泰 : 百科事典から動的に年表を生成するテキスト検索法のための年代情報の抽出法と表現法, *情報処理学会 情報学基礎研究会* (予定), 1998.7.

[MUC 98] *Proceedings of the Seventh Message Understanding Conference* (MUC-7). SAIC, 1998.

[Sai 98] 斉藤 公一, 迫田 昭人, 中江 富人, 岩井 禎広, 田村 直良, 中川 裕志 : 数値情報をキーとした新聞記事からの情報抽出, *情報処理学会 自然言語処理研究会*, 125-6, pp. 63–70, 1998.

[Tak 99] 高尾 宜之, 永井 秀利, 中村 貞吾, 野村 浩郷 : 複数製品の紹介記事からの製品情報抽出 — 製品記述パターンの分析 —, *情報処理学会 自然言語処理研究会*, 129-17, 117–124, 1999.

[1] 「モビール」は固有名詞と普通名詞の両方で出現する．また，「日本」はしばしば他の固有名詞の一部として出現する．