

A Method of Geographical Name Extraction from Japanese Text for Thematic Geographical Search

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 geographical search method has been developed and applied to a Japanese encyclopedia called the World Encyclopædia. In this method, the user specifies a search theme using free words, then obtains a sorted list of excerpts and hyperlinks to encyclopedia sentences that contain geographical names. Using this list, the user can also open maps that indicate the locations of the names. To generate an index of names for this searching, a method of extracting geographical names 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 by using non-local context analysis, which uses a stack and several other techniques. As a result, the precision of extracted names is more than 96% on average. This method 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.

1. Introduction

New methods of searching text through which end users can find desired information by using a simple input, and through which they can discover knowledge distributed in a text will soon be needed as interest in the Internet grows and as more CD- or DVD-ROM contents are developed. To find useful information in such media, the end-user directly searches them instead of asking professional searchers to do so, as in a conventional database search. In addition, the user searches a much larger amount of text, such as WWW pages, than before. If the user searches a larger amount of text, the quantity of search results also becomes larger.

In such situations, new information retrieval methods that organize search results will be required. If the number of search results is large, it takes too long for the user to survey all of them, unless they are properly structured. If the search results are organized well, however, the user can survey many search results and can find useful ones by using a simple search condition. The organization function is very important, because when the user can reduce the number of search results by using an elaborate search

condition, important issues related to the subject can often be discarded by this reduction. The user's exposure to the potential search results can sometimes be, therefore, too narrow.

I believe that the organization of search results based on the user's intention is necessary to overcome the above problems. As a first step toward an organizing search method, I have developed the *axis-specified search method* [Kan 98]. In this method, the user selects an axis to organize full-text search results. Thematic chronological-table searching [Kan 99] is axis-specified searching with a year axis. The thematic geographical searching described in this paper is axis-specified searching with a geographical axis, and it is a method for searching and organizing geographical information from a text collection. In a thematic geographical search, the text collection is scanned, and geographical names are extracted. The names are matched to names in a geographical name database or dictionary, and they are then identified and entered into an index. Because both the text collection and the database contains several types of ambiguities, the most important task of this name extraction is ambiguity resolution.

Although information extraction, including geographical name extraction, has been widely studied, most methods (e.g. [MUC 98] [Ino 96][Tak 99][His 97]) are used to extract unknown names. A method of *known* name extraction with identification of extracted names in relation to names entered in a database has not been established. In this paper, a method of extracting geographical names and the techniques used for resolving ambiguities in the thematic geographical search are explained. The thematic geographical search method is outlined in Section 2. The basic method of geographical name extraction used is explained in Section 3. Ambiguities of geographical names are explained in Section 4, and the techniques for resolving ambiguities are explained in Section 5. The performance of ambiguity resolution and the name-extraction precision are evaluated in Section 6.

2. Outline of the Thematic Geographical Search Method

Thematic geographical searching is part of axis-specified searching. Axis-specified searching and then thematic geographical searching are explained in this section.

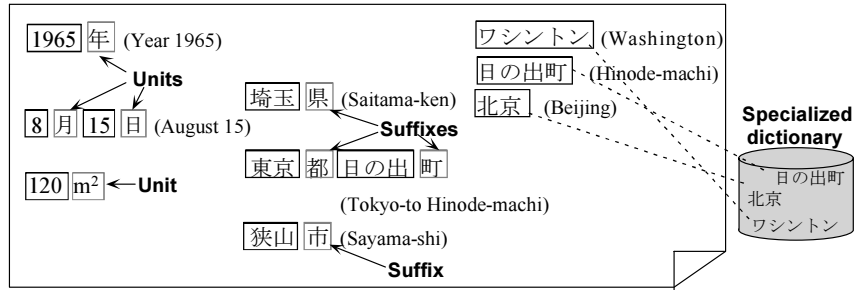
In axis-specified searching, the user selects an axis from a menu, and enters search words. The candidates of the axis are predefined by the search system. The words represent a search topic, and the axis specifies a method of organizing search results. Full-text search results for the words are obtained and sorted along the axis. Conceptually, the search results are put in a space that is specified by the axis. The range on the axis is also specified by the user. Search-results outside the range are discarded.

There are three methods for specifying an axis. Examples are shown in **Figure 1**. The following three types of axis-specified searching correspond to those examples illustrated.

1. *Quantity searching*: an axis is specified in units of a quantity (Figure 1a) [Kan 98].
2. *Subword searching*: an axis is specified by the ending or beginning of words (Figure 1b). Geographical-axis searching may be incorporated using subword searching [Kan 98].
3. *Word category searching*: an axis is specified by the category of words (Figure 1c). Geographical-axis searching may be incorporated using word category searching.

In a word category search, the database (or dictionary) contains words in a category or several categories. A thematic geographical search is a type of word category search. The database is a geographical-name database in this search. Only the words entered in the database (i.e., geographical names) are searched for.

The actual user interface in Japanese, which was developed by Hitachi Digital Heibonsha¹ and runs on Microsoft Windows and Windows NT, is shown in **Figure 2**. However, a simplified example translated into English is explained using **Figure 3**. The



(a) Quantities with specified units (b) Words with specified word tails or heads (c) Word category

Figure 1. Three types of axis specification in axis-specified searching

user searches an encyclopedia by specifying “riot” as a search word. The user selects the geographical axes and specifies Japan as a range on the axis. In the thematic geographical search, the geographical range menu contains the World, areas such as Asia or Africa, each country in the world, each prefecture in Japan, and so on. The system then displays an index map or a sorted table in which the geographical names and the excerpts from the search results are listed.² In Figure 3, “Aichi Prefecture”, “Toyoda City” in Aichi Prefecture, “Saitama Prefecture”, and “Tokorozawa City” in Saitama Prefecture are geographical names. “Mikawa” and “Bushuu Riot” are the article headings in the encyclopedia. Excerpts follow these headings in the search results. A feature of this search is that not only the encyclopedia text but also a map that contains the geographical name can be opened from a result concerning the name. By using a hyperlink, the user can see the whole article that contains the name. In Figure 3, a hyperlink is embedded in the underlined parts.

The system architecture for axis-specified searching is illustrated in **Figure 4**. This architecture consists of two main parts: a set of index generators and a search engine. Index generators generate axis indices and a full-text index from the text collection. Axis-index generators extract strings that match predefined patterns, normalize the extracted information, and enter it into the index. The full-

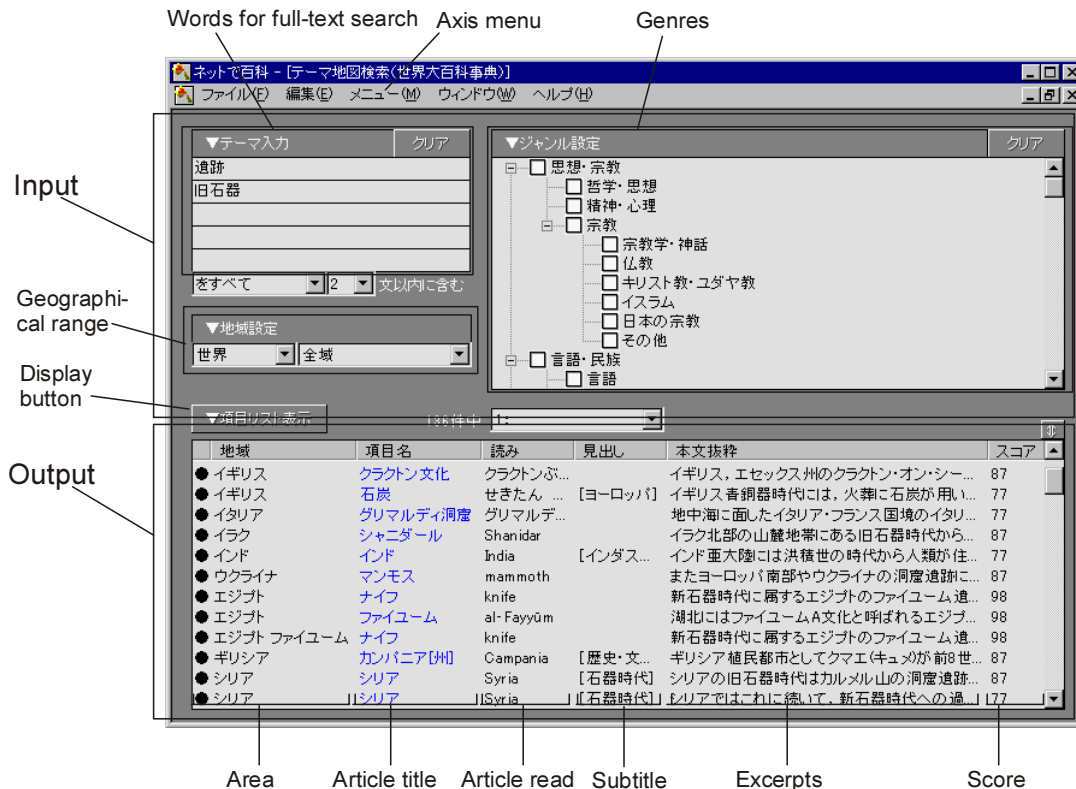


Figure 2. The interface for the axis-specified search with a geographical axis (the thematic geographical search)

¹ A Japanese company that publishes CD-ROM and networked versions of encyclopedias (<http://www.hdh.co.jp/>).

² In the current version, only an index table can be displayed. An index map cannot yet be displayed. (See Figure 2.)

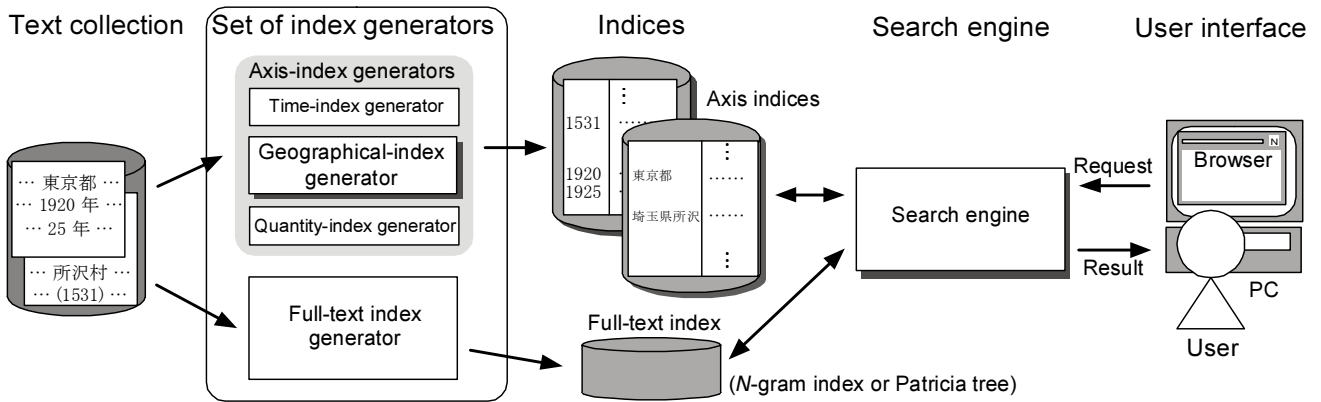


Figure 4. The system architecture for axis-specified searches

text index generator generates a full-text index of a conventional form.

The search engine is invoked by the user. The geographical axis is selected when the user selects the thematic geographical search from the axis menu (see Figure 2). The search engine searches the full text for sentences that contain geographical names in the specified range using the geographical-axis index. The results are ordered along the axis, the score for each result is calculated, and the results whose scores are too low are dropped.

3. Method of Extracting Geographical Names

The method of geographical names in thematic geographical searching is explained in this section. The basic extraction method, the structure of an extracted name, the outline of the geographical-name database (GDB), the method of name matching, and the method of handling aliases are explained.

3.1 Outline of geographical name extraction

In generating the geographical-axis index, the system scans all the documents and extracts all the strings that match geographical names contained in a GDB. Extracted names are normalized and entered into the geographical axis index. The GDB was developed at the Hitachi Digital Heibonsha for the maps in the World Encyclopædia [HDH 98]. The method of extracting geographical names has been applied to the World Encyclopædia, which is a 35-volume encyclopedia, and to Mypædia [HDH 99], which is a single-volume encyclopedia. The text volume of the World Encyclopædia is 160 MB, and the number of articles is 84,000. The number of Japanese geographical names extracted is about 130,000, and the number of foreign geographical names extracted is about 340,000 (including items with the same name).

The extraction process tests the text immediately before and after the matched string, and it decides whether the matched name

should be extracted. This local context matching is done by string matching, and so-called natural language processing, such as morpheme analysis or syntax analysis, is not used. Although the extraction task would sometimes become easier were a morpheme analysis used, our method is as powerful as the morpheme-analysis-based method in most cases. Some geographical names can be extracted using only context-free rules. However, if names with the same spelling exist in two or more areas, the names must be identified using context-sensitive rules. Matching patterns and the normalization method for matched names should depend on the nature of the text to be searched. The same spelling that occurs in different types of texts may be normalized into different names.

3.2 Structure of extracted names

In the thematic geographical search, a geographical name has a two-layered structure. The upper layer is a country name or a name of areas that correspond to countries¹ if the area range is “World”, and it is a prefecture name if the area range is “Japan”. Japan is handled differently because the World Encyclopædia is a Japanese encyclopedia. The lower layer is the name that occurred in the text. The name may be a city name, a district name, a mountain name, a lake name, and so on. The upper- and lower-layer names may be the same if the extracted name is a prefecture or country. In reality, a geographical name may have three or more layers, such as Pittsburgh, Pennsylvania, USA. However, it is sometimes difficult to correctly extract three or more layers allowing abbreviations, and a search result should be expressed in a concise format. Thus, the number of layers is a fixed number, two, in the current version. Thus, “アメリカ合衆国 ピッツバーグ” (Pittsburgh, USA) is extracted instead of “アメリカ合衆国 ペンシルベニア州 ピッツバーグ” (Pittsburgh, Pennsylvania, USA).

3.3 Use of a geographical-name database

A GDB that contains geographical names, their identification numbers (INs), their readings, their types, and their upper-layer names, and so on, has been developed along with the World Encyclopædia Maps at Hitachi Digital Heibonsha. The type can be country, prefecture, mountain, river, city, town, world-famous city, and so on. There may be two or more upper-layer names for a lower-layer name because a geographical feature, such as a mountain or island, may spread along two or more areas. Part of

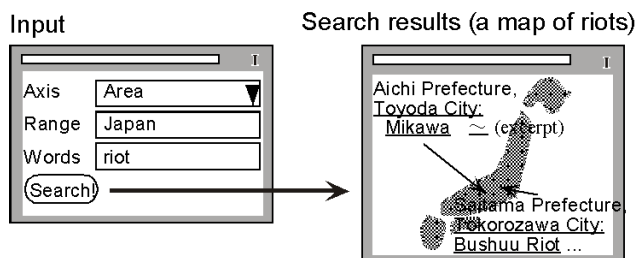


Figure 3. Geographical axis search

¹ Examples of names of areas that correspond to countries are the South Pole, the West Sahara, and Greenland.

this database is shown in **Figure 5**. The number of Japanese names entered in the GDB is about 55,000, and the number of foreign names is about 41,000. The GDB does not include historical names.

IN	Name	Reading	Attributes	Upper-area code	Priority
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	なんせいしよとう	諸島	46	47
10	九州	きゅうしゅう	島	101	

Figure 5. Part of the World Encyclopædia map database

All the names have their INs in the GDB. A map that shows the location of the name can be opened by passing the IN to the map application programming interface (API). However, not all the names entered in the GDB can be located in the maps. So the existence of a geographical name in a map is also recorded in the GDB. Just before the name extraction, the system inputs the contents of the GDB and converts them into an internal form in the memory. The GDB contains other information, such as latitude, longitude, and so on, but they are not used in our method.

Only the geographical names that are entered in the GDB are extracted from the encyclopedia text. Recently, the extraction of unknown names, i.e., names that are not entered in a database, has been widely studied. However, it is impossible to obtain locations or maps of the places from unknown names, although it is important that the thematic geographical search identifies the location, and obtains the link to a map. In addition, the precision of unknown name extraction is not sufficiently high for encyclopedia searches, whose results must be reliable. So unknown names are much less valuable than known names for the thematic geographical search. Thus, only the names entered in the GDB are extracted.

3.4 Matching geographical names

The algorithm for matching geographical names is illustrated in **Figure 6** and explained below. The *extract* function inputs a text and the GDB, and outputs a list of pairs (I_i, S_i) , where I_i is the IN of a geographical name, and S_i is the sentence number in which the geographical name occurs. The context stack C and its handling in the *identify* function will be explained in Section 5.2.

This algorithm is designed according to the features of the Japanese language. Japanese words are written in three types of characters: Kanji (Chinese characters such as 西, 鳥), Hiragana (と, の, etc.), and Katakana (プ, マ, etc.). The Roman alphabet is sometimes used too. There are no delimiters between words. Morpheme analysis is thus required to recognize word boundaries exactly.

When candidate strings for matching are extracted from the text, the probability of extracting erroneous names becomes high if the text matching is allowed to start anywhere in the text. It is possible to introduce morpheme analysis, and the matching starting points (MSPs) are restricted to the beginning of morphemes. However, a result of morpheme analysis usually contains errors of several percent; it also has a high computing cost, and it does not greatly increase precision. Thus, it is not used in this method, and the positions where the type of character changes, for example, from Hiragana to Katakana or Kanji, or from a special character (symbol) to Kanji, are used for MSPs. The positions where ex-

```

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 IN (identification number) of a geo-
    graphical name, and  $S_i$  is the sentence number 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 MSP 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 IN.
                if  $I$  is not nil then -- The identification succeeded.
                    add  $(I, S)$  to index  $X$ ;
                end if;
            end if;
        end loop;
    end loop;
end;

```

Figure 6. Geographical name extraction procedure (main part)

tracted names end are also used as MSPs. However, sometimes the character type does not change at word borders, so MSPs are put immediately after the characters that often occur immediately before a geographical name, such as 年 (year), 月 (month), 市 (city), 町 (town), 村 (village), 前 (before), or 後 (after).

There is no character type change when the name begins with Hiragana if the word immediately before the name is a particle, and there are few geographical names that are commonly written in Hiragana. Thus, every position before Hiragana is used as an MSP. For example, the MSPs of the sentences for part of the encyclopedia article “Ibo River” (揖保川) are shown by bars:

|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|く|川。|幹川|流路延長 70km、|全流域面積
 810km²。

Dots (“.”, called Nakaguro in Japanese) are usually used as delimiters within a geographical name in Katakana characters, as in プエルト・リコ (Puerto Rico), so they are handled similarly to Katakana characters. Dots are also sometimes used to juxtapose names in Kanji or Hiragana characters, as in 埼玉・東京 (Saitama and Tokyo). However, if MSPs are inserted immediately after dots, erroneous geographical names are often extracted, so no MSPs are inserted after them.

Names are matched using the longest coincidence method. If there are two or more geographical name spellings in the GDB, which match the string in the text, the longest one is selected. Then the quantifiers are tested. This test is explained in Section 5.4. When a geographical name spelling is found, it is normalized by the *normalize* function (see the next subsection). The geographical name is then identified with a name entered in the GDB by the *identify* function. The IN is obtained by *identify*. The identification process, or ambiguity resolving process, is explained

in Section 5.2. If the IN has been found, it is entered into the axis index.

3.5 Normalizing aliases

Aliases often exist for geographical names. For example, China is an alias for the People's Republic of China, the USA is one for the United States of America, and the Aleutian Islands is one for the Aleut Islands. In Figure 6, occurrences of aliases are replaced by normalized names by using the *normalize* function and they are entered into the axis index.

4. Ambiguity of Geographical Names

It is impossible for an encyclopedia to be completely free of ambiguity because it is written in a natural language. The GDB also has some ambiguity. Geographical names thus have various ambiguities described in this section.

4.1 Different locations with the same spelling

There are many geographical locations whose names have the same spelling. In particular, there are many cities with the same name in the U.S., and between the U.S., the U.K., and Australia. For example, there are at least eight Columbuses in the U.S., and three of them, in Georgia, Ohio, and Indiana, occurs in the encyclopedia. As well-known examples, New York is both a city name and a state name, and Washington is both the capital name and a state name.

Three example texts are shown. These were derived from the World Encyclopædia, but the texts are shortened and rewritten in English. The first example is:

Ohio is a state in the United States. More presidents have been born in Ohio than in any other state except Virginia, where an equal number have been born. Columbus, Cincinnati and Dayton are major cities there. (derived from the article "Ohio [State]")

The state where Columbus is located, i.e., Ohio, occurs here. The ambiguity concerning Columbus can be resolved using this. However, another state name, Virginia, occurs between Ohio and Columbus, and this may confuse the ambiguity resolution process.

The second example is:

A psychoanalyst born in Hungary. He moved to the United States in 1930. He taught psychoanalysis in Boston, Chicago and Los Angeles. (derived from the article "Franz Alexander")

The state name for Boston does not occur here. Thus, there is a chance of confusing this Boston with Boston in the U.K. However, the ambiguity can be resolved by "the United States", because there is no Boston in the U.S. other than the one in Massachusetts included in the GDB.

The third example is:

A ball game invented by a student in America. The Rose Bowl, which is held in a suburb of Los Angeles, Pasadena, has the longest history. (derived from the article "American Football")

The ambiguity concerning Pasadena is not resolved by "the United States" here, because there are two Pasadenas in the U.S. However, if the state for Los Angeles, i.e., California, is used, it can be identified as Pasadena, California.

4.2 Ambiguity with proper or common nouns

The spellings of some geographical names are the same as those of

a proper noun for non-geographical names. For example, many geographical names, such as Washington, come from human names. Also, the spellings of some names are the same as those of a common noun. For example, there are geographical names such as 平和 (peace), 運河 (canal), 東西 (east and west), and 東方 (east direction) in China. There are cities called Newtown and Poole, whose Japanese spellings are the same as those of common nouns (i.e., new town and pool), in the U.K. If these names are extracted only by testing the spellings, most of the extracted results would actually be non-geographic proper nouns or common nouns.

4.3 Incompleteness of GDB

A geographical-name database is a large database inputted by humans, and it will reflect the complexity of real-world politics and geographical conditions. Thus, it is unlikely to be far from complete, and is likely to be error free. Also, because the GDB was not built specifically for extracting names, it does not necessarily have the ideal attributes for extraction. For example, a geographical name may appear two or more times in the GDB, and each may have different attributes in each case. This may happen, for example, if a city appears on two different types of map. The name extraction process should be tolerant of such incompleteness.

4.4 Other errors

There may be other errors and ambiguities in a geographical name extraction process and in a GDB. For example, a fragment of a word may be regarded as a geographical name because of word analysis failure. Because morpheme analysis is not used, such errors may occur when a decision concerning word borders based on character types is not correct.

5. Resolving Ambiguity

Methods for resolving ambiguity in extracting names and for reducing the occurrence of extraction errors are explained in this section. The identity of names (Section 5.1) is described, and rule-based (5.2-5.3) and fact-based or dictionary-based (5.4) ambiguity resolution techniques are explained.

5.1 Identity of geographical names

As mentioned in Section 4.3, there may be two or more records that represent geographical names with the same spelling in the GDB. If there are such names and their upper-layer names are equivalent, and if they do not contain contradictory information, they are regarded as the same locations. Two different locations are rarely identified as the same in this technique in the World Encyclopædia.

5.2 Name identification using non-local context

Context must be analyzed to identify ambiguous geographical names. For example, if "Columbus" occurs in a description on Ohio, it is probably Columbus, Ohio. If it occurs in a description on Georgia, it is probably Columbus, Georgia. However, neither the syntax nor the semantics of natural language is analyzed in this implementation. It is possible to analyze them partially, but complete analysis is not possible with the current technology. Instead, a much simpler method is used to analyze context and to try to identify ambiguous names. This method is sufficient for resolving ambiguity in the cases described in Section 4.1.

The identification method is described in **Figure 7**. When

```

function identify(N, var C) return I;
input N: a name spelling;
output I: an IN 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  $I_1, I_2, \dots, I_m$ ,
        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 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;

```

Figure 7. Geographical name identification procedure

scanning the text from left to right to extract names, a country or prefecture name (or state names in a U.S. case) are stored in a stack called the *context stack*.¹ If a prefecture or country name occurs in the text, it is entered into the context stack. If a geographical name whose layer is lower than that of a prefecture or country name occurs, the upper-layer prefecture or country name is entered into the context stack. The depth of the stack is limited to around five, and when the number of names entered exceeds the depth, the oldest name is discarded.

The GDB contains a type of priority value or the relative importance of each geographical name. If an extracted name is ambiguous and the candidates have different priorities, the candidate with the highest priority is selected.

If there is no ambiguity in a name in the text, the upper-layer name is identified without referring to the context. If the name in the text is a prefecture or country name, the context is not tested. This means that prefecture and country names are context-free. If an ambiguous name occurs, the candidates for its upper-layer names are compared with the names stored in the context stack or their upper-layer names, and an equivalent name is selected. The order of comparison is controlled by the *context_stream* function.

¹ The method for context handling does not strongly depend on the language, but the word order does. The upper-layer geographical name (e.g., the U.S.) comes *before* a lower-layer name (e.g., Pittsburgh) in Japanese, but the order is reversed in English. Because the text is scanned from left to right in this method, this order of stacking names must be modified when applying this method to English texts.

This function returns a stream (or list) of geographical names and their upper-layer names in the context stack. If a candidate matches two or more names returned from the *context_stream* function, the first one is selected.

The *context_stream* function illustrated in **Figure 8** is explained here. If a geographical name in *C* is American, its state name is put into the stream. If a geographical name in *C* is Japanese, its prefecture name is put into the stream. However, if the name itself is a state or prefecture name, it is not put into the stream. Then, the country name is put in, and finally, the global area name, such as North America or Asia, is put in. In the *identify* function, the name *N* is compared with prefecture or state names first if the names in the context stack are in Japan or the U.S., respectively, then it is compared with the country names. Finally, it is compared with the global names. This order of comparison may have room for improvement, but most cases of erroneous identification can be avoided by using this order.

The ambiguity resolution process in the example of the state of Ohio, the first example in Section 4.1, is explained here. The names extracted when processing “Columbus” are listed:

(the most recent) *Virginia [State], Ohio [State], the United States, Ohio [State]* (the least recent).

The state names obtained from these names are:

Virginia [State], Ohio [State], Ohio [State].

These are put into the stream (or list) *S*, which will be the result of *context_stream*. Then, the country names and the global names are put in. Because duplication is avoided, the value of *S* becomes:

Virginia [State], Ohio [State], the United States, North America.

The order of matching is from the most recent (nearest) to the least recent. So whether there is a Columbus in Virginia is tested first. Because there is none, whether there is a Columbus in Ohio is then tested. Because there is, the geographical name is identified as “Columbus, Ohio”. The identification result does not depend on the order of matching in this case. However, if the context stack contained Georgia, where there is another Columbus, the result would depend on the order. If Georgia comes before Ohio, i.e., it is farther from “Columbus”, it is still identified as

```

function context_stream(C) return S;
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 U.S. 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 S;
end;

```

Figure 8. Test context stream generation procedure

(1) Succeeding qualifiers

A, B, ..., Z, O, 1, ..., 9, 》, >, “ ”, 語 (language), 人 (people), 家 (family), 氏 (Mr. or Ms.), 法 (law), 属 (genus), 目 (order), 派 (school), 党 (party), 賞 (prize), 大学 (university), and suffixes that consist of 両 (both), 各 (each), 諸 (various), or numbers and the above items. 大統領 (president), 首相 (prime minister), 総督 (governor-general), [大]司教 (bishop), [大]主教 (bishop), 男爵 (baron), 子爵 (viscount), 内閣 (cabinet), 政權 (administration), [-]族 (group or family), 兄弟 (brothers), 姉妹 (sisters), 主義 (-ism), 時代 (era), 報告 (report), [会]社 (company), 銀行 (bank), 商会 (company), 商店 (store), [街]道 (street), [大]聖堂 (cathedral), 記念 (memorial), 變動 (movement), 広場 (square), 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, 館, 区, 星, 期, 師, 銃, 鉢, 屋, 々.

(2) Preceding qualifiers

家 (-ist) (except 国家 (nation)), 大統領 (president), 首相 (prime minister), 総督, 提督, 監督, 將軍, [大]司教 (bishop), [大]主教 (bishop), 民族 (folk), 諸族 (folk), 大將 (general), 中將 (lieutenant general), ..., 大佐 (colonel), ..., 小尉 (second lieutenant), 艦, 党, 者, 長, 人, 公, 機, 師, 夫, 妻.

Figure 9. Suffixes and prefixes that follow proper nouns, excluding geographical names, in Japanese

“Columbus, Ohio”.¹ However, if Georgia comes after Ohio, it is identified as “Columbus, Georgia”. The name in the other two cases in Section 4.1 are also identified correctly using this method.

In the current version, the prefecture or country names in article titles or subtitles are handled in the same method as geographical names in the body text. However, it would be better to hold them longer in the context stack.

5.3 Filtering by qualifiers

When a geographical name and a human or organization name are not distinguishable, it may be possible to filter out the latter by checking for qualifiers before and after the name. For example, a word that is preceded by President or a followed by Party, Brothers or Company can be judged as other than a geographical name. A list of such suffixes or prefixes in Japanese is shown in **Figure 9**. However, this is a supplementary technique though, because qualifiers do not always exist.

5.4 Dictionary-based techniques

The rule-based techniques explained in Sections 5.2 to 5.3 are not sufficient for resolving all ambiguities concerning geographical information written in a natural language. There are also three methods of implementing dictionary- or fact-based techniques.

1. Modifying the GDB. Records in the GDB are modified, added, or removed. This is a very straight-forward method, but is difficult if the original purpose of the GDB was not thematic geographical searching.
2. Building a temporal database from the GDB and small supplementary dictionaries or lists (patches) just before the name extraction. This method may complicate the structure of the database, but it does not affect other projects in which the text

¹ In the *context_stream* function, if the context stack contains Georgia and Virginia in that order, and the country name of the most recently occurring name is pushed before the second most recently occurring name, the result of *context_stream* will be:

*Virginia [State], the United States, Georgia [State],
North America.*

Then, if state capitals are given a higher priority, Columbus will match “Columbus, U.S.”, which means Columbus, Ohio. This search result is unlikely.

and/or the GDB are used.

3. Adding XML or SGML tags to the text. An identified name can be enclosed in a tag and added to ambiguous geographical names in the text. However, if the text is used for other purposes, this may be difficult.

We use the second method, mainly by adding the following three supplementary databases:

1. A list of spellings that may occur in the GDB but that are not entered into the specialized database. Ambiguity with proper or common nouns may be resolved by using this list.
2. A list of record identifiers in the GDB. The records specified by this list are not entered into the specialized database. If a name in the GDB only acts as noise in the name extraction, the record can be removed using this list.
3. A collection of records or parts of records that fully or partially replace records in the GDB. If the priority of a name in the GDB is not appropriate for the name extraction, it may be replaced using this collection.

Sometimes it is possible to greatly improve the extraction precision by using these techniques. However, note that these techniques are ad hoc. A text update, which adds new geographical names to the text, may degrade the application result of these techniques.

6. Evaluation

The ambiguity-resolution performance and precision have been evaluated.

6.1 Ambiguity-resolution performance

The results of the ambiguity-resolution performance evaluation are shown in **Table 1**. The following method was used. Name spellings that occurred five times or more in the GDB were listed. Two test collections of Japanese and foreign name spellings (listed in **Figure 10**), were generated from the spellings. All occurrences of these names entered into the geographical axis index are retrieved using a thematic geographical search. All the search results were checked manually and the number of errors were counted.

Table 1. Results of evaluating ambiguity resolution

Type	Number of names	Total extracted names	Number of errors	Precision
Japanese names	48	633	103	0.84
Foreign names	34	1139	98	0.91

Table 1 shows that the precision for the foreign names was good (91%), but that the precision for the Japanese names was lower (84%). Three reasons for the lower precision are that the correct place was sometimes not entered in the GDB, that human names were sometimes wrongly extracted as geographical names, and that two or more names with the same spelling were often found in a prefecture and it was sometimes hard to distinguish between them using the context. However, both precision values were much better than random selection from ambiguous names. A random selection would lead to precision of less than 20%, because there were at least five ambiguous names for each name used in the evaluation.

(1) Japanese names

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) Foreign names

アーリントン (Arlington), アバディーン (Aberdeen), アレクサンドリア (Alexandria), ウィルミントン (Wilmington), ウィンチェスター (Winchester), オールバニー (Albany), キングストン (Kingston), ケンブリッジ (Cambridge), コロンバス (Columbus), コロンビア (Columbia), サン・カルロス (San Carlos), サン・フェルナンド (San Fernando), サン・ペドロ (San Pedro), サン・ルイス (San Luis), サンタ・クルス (Santa Cruz), ジャクソン (Jackson), スプリングフィールド (Springfield), セーレム (Salem), チャールズタウン (Charlestown), ニューカスル (Newcastle), ニューポート (Newport), バーリントン (Burlington), ブラック川 (Black River), フランクリン (Franklin), プリマス (Plymouth), プリンストン (Princeton), ベルビル (Belleville), ポーツマス (Portsmouth), マリオン (Marion), マンチェスター (Manchester), ラ・パス (La Paz), ランカスター (Lancaster), リッチモンド (Richmond), レバノン (Lebanon).

Figure 10. List of evaluated geographical names

6.2 Precision

The results of evaluating the precision of name extraction using five retrieval tasks are shown in **Table 2**. The first three columns show the search conditions, and the last three show the results. In this evaluation, the correctness of geographical information was judged by a human for all the search results. The precision of the searches for “revolt” or “paleolith and ruin” was more than 98%. This was quite satisfactory. However, that of two other searches was about 95%, which was not satisfactory for the encyclopedia. This is caused by identification errors on only a few words: モバイル (Mobile) and 日本 (Japan) for “beer”, and 津 (Tsu) for “tea”.¹ If these words are excluded, the precision is as high as others. The precision was much better than that for the ambiguity resolution because name ambiguity was not deliberately introduced.

7. Conclusion

A text retrieval method called the thematic geographical search method has been developed. In this method, geographical names in the text are extracted, identified using a geographical name database, and entered into an index.

The most important problem in the identification is ambiguity

Table 2. Results of evaluating precision of name extraction

Search words	Range of area	Distance in sentences*	Number of search results	Number of errors	Precision
一揆 (revolt)	Japan	2	641	13	0.980
茶 (tea)	Japan	0	376	20	0.947
ビール (beer)	World	3	583	29	0.950
コンピュータ (computer)	World	5	568	16	0.972
旧石器, 遺跡 (paleolith and ruin)	World	5	525	7	0.987
Total	-	-	2693	85	0.968

* See Kanada [Kan 98].

¹ Mobile is used both as a proper and a common name. Japan is often used as part of a proper name such as Japan Beer (日本麦

resolution. The application of several ambiguity resolution techniques, including analysis of non-local contexts using a context stack, has enabled extraction precision of more than 96% on average. Although this extraction method was developed for an encyclopedia, and the rules used for extracting geographical names and resolving ambiguity include language-specific rules, the strategy and most of the rules are generic. Thus, they can be applied to other types of texts and to texts in English or other languages.

Future work on the method of extracting names will include **1**) improving precision and recall (Context handling should be improved so that both the precision and recall can become better) and **2**) application to other types of text (This method should be applied to other types of text, such as newspapers, Internet mail and newsgroups, and the WWW) and application to texts in other languages.

Acknowledgments

I am grateful to Yasufumi Fujii of Hitachi Digital Heibonsha for allowing me to use the text and the GDB of the World Encyclopædia. I thank Toshiyuki Oda, Tomomitsu Inoue, Toru Adachi, and the other members of the Hitachi Digital Heibonsha, and Yuji Ogihara and Yoshiaki Hirano from the Information Systems Division, Hitachi Ltd., for helping me improve the extraction and sorting methods. I also thank Yukio Hoshi and other members of the Software Division, Hitachi Ltd., for improving the full-text search engine so that it could be used for thematic geographical searching.

References

- [HDH 98] *DVD/CD-ROM World Encyclopædia, version 2*, Hitachi Digital Heibonsha, 1998.
- [HDH 99] *CD-ROM Mypædia 99*, Hitachi Digital Heibonsha, 1999.
- [His 97] Hisamitsu, T., and Niwa, Y.: Acquisition of Person Names from Newspaper Articles by Lexical Knowledge and Co-occurrence Analysis, *SIG on Natural Language Processing*, Information Processing Society of Japan, 118-1, pp. 1–6, 1997 (in Japanese).
- [Ino 96] Inoue, Y., et al.: Template-based Products Information Extraction from Newspaper Articles, *SIG on Natural Language Processing*, Information Processing Society of Japan, 96-NL-115, pp. 83–90, 1996 (in Japanese).
- [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, 1998.
- [Kan 99] Kanada, Y.: Methods of Extracting Year References for Chronological-table-generating Text Searching, *Int'l Symposium. on Digital Libraries 1999*, Univ. of Library and Information Sci., Tsukuba, 1999.
- [MUC 98] *Proceedings of the Seventh Message Understanding Conference (MUC-7)*. SAIC, 1998.
- [Tak 99] Takao, Y., Nagai, H., Nakamura, S., and Nomura, H.: Information Extraction from Newspaper Articles of Multiple Products — classification of expression patterns —, *SIG on Natural Language Processing*, Information Processing Society of Japan, 129-17, pp. 117–124, 1999 (in Japanese).