

道案内文と略地図の符号化された属性による相互変換システム

山下達雄*・山縣敬一**

*奈良先端科学技術大学院大学情報科学研究科

**広島大学総合科学部

Conversion of Route Guide between Sentences and Rough Sketch via Coded Information

Tatsuo YAMASHITA* and Keiichi YAMAGATA**

**Nara Institute of Science and Technology*

***Faculty of Integrated Arts and Sciences Hiroshima University,*

Higashi-Hiroshima 739 Japan

Abstract : This paper discusses coded information of various attributes in guide map and mutual conversion between sentences and rough sketch. Practically a tool for converting linguistic route guide to visual map and a tool for converting visual map to linguistic route guide are proposed. The total system is implemented in a workstation by some computer programs; morphological analyzer, Japanese dictionary, data processor described by a programming language "Perl" and a drawing tool of map. Precise mechanism in human brains for associating linguistic sentences with images is not clear.

In this research, morphological analyzer extracts map attributes from sentences. Linguistic sentences of route guide are associated with visual map by attribute terms about direction, right/left side, crossings, buildings, qualitative distance and so on. Effectiveness of this system is estimated about some examples and discussions are also given to put this system to practical use.

Key words : route guide, map attribute, morphological analysis, rough sketch

1 はじめに

本研究では言語情報と視覚情報の関係について^{1),2)}、

- 道順を案内する時に用いられる道案内文 (言語情報)
- 略地図 (視覚情報)
- 脳の中で記録されている道順のイメージ

の3つを中心に考察した。

道順を誰かに教える時、頭の中にその道順のスタートからゴールまでの風景が時間を追って浮かんでくる。人間はその浮かんでくる道順のイメージを基に道順を言葉や略地図にして説明していると思われる。また逆にその道案内文や略地図を聞いたり見たりすると、そこへ行ったこともないのにあたかもその示された道順の通りにその場所を歩いているような連続したイメージが浮かんでくる。

この時間的に連続した道順のイメージを位置関係や距離・方向の属性値としてコンピュータの中で符号化することにより人間の思考の定性的表現の実現が可能になると思われる。本研究ではその符号化の一例として、かなり簡略化したものであるが符号化情報としての「コード」を提案し、道案内文(言語情報)と略地図(視覚情報)の相互変換の実現を試みた。

まず第2章では本研究で使用する表現法「コード」「道案内文」「略地図」について述べる。第3章では道案内文(言語情報)を「コード」に、そしてその「コード」を略地図(視覚情報)に変換するという一連の処理について述べる。第4章では略地図(視覚情報)のコンピュータへの入力と「コード」化、「コード」の言語情報化の処理について述べる。そして第5章では実際の道案内文・略地図を用いてシステムの性能を評価する。

2 表現法について

2.1 「道案内文」「略地図」とは何か

ここでは今回使用する「道案内文」及び「略地図」とはどのようなものを説明する。道案内文とは、ある場所からある場所へ行く道順を説明する文である。発話者と受話者の共通の知識の量や発話状況によってかなり様相を変えるが、今回は、その土地に詳しい発話者Aがその土地の地理について何も知らない受話者Bに発話する道案内文を、第三者にも分かるものとして取り扱う。Bが得られる情報はその道案内文(つまり言語情報)のみである。従って、Aは身振りや図入りのメモなどの視覚情報はいっさい与えることができない。電話での状況を想定すると分かりやすい。Bはその土地の地理について何も知らないだけでなく、如何なる情報もその道案内文以外から入手できない状況にある。それゆえ当然現地の地図を見ながら道案内文を聞くということとはできない。また二人だけの暗黙の了解が必要となるような文であってはならない。

本研究で言うところの略地図とは、人に、ある場所からある場所(目的地)へ行く道順を説明する地図であって、以下のように定義される。

1. 道や交差点・目印となる建物・出発地点・目的地など実際に目的地まで辿るときに必要な要素のみから構成されている。要するに目に入る建物などの目印と通り道から構成されている。
 2. 分かりやすさを優先するため実際の細かい距離の差や道のカーブなどの情報が削除されている。そのため直線と点のみで表せ、単純化できる。
 3. 出発地点から目的地までに通る道と、交差点などで通らないが目に入る道を区別している(道を太く描く)。これは自分の通る道を一目で分かるようにする必要があるためである。
- なお、本研究では処理を単純にするため、道はX軸Y軸に平行な物のみを扱っている。

2.2 コードと道案内文の構造

コードは、建物・交差点・右折など頭の中の道順のイメージから本質的な要素を抽出したものであり、道案内文と略地図との橋渡し役として用いられる。このコードにより、同じ性質の多種多様

な道案内文の語彙を一つのものに集約して扱え、処理が容易になる。また、略地図上の要素に対応づけることで地図情報がかなり圧縮できる。従って、あらゆる道順データ（言語・図）をコード化し保存できればかなりのメモリの節約が期待できる。反面その単純化によりかなりの情報が失われるという問題もある。これは言語や図の情報を失わないようなコード体系を構築するという重要な問題に通ずるが、本研究ではとりあえず単純なコード体系を定義してみて、それでシステムを構築し評価するという方針でいくので、ここではこの問題については深く立ち入らない。今回使用しているコードを表1にまとめてある。

コード	対応する語の例	解説
B_名前	銀行・学校・公園	Building 目印となる建物などを表す。実際は「B_銀行」のような形で扱われる。
W_CROSS	交差点・四つ角・十字路	自分の進む道以外の道が現れ、 進路の選択を迫られる場所。
W_T	T字路	
R_名前	大通り・国道	名前を明示した道路。Wと同じ働きをする。
D_S	そのまま・まっすぐ	Direction 進行方向の変換を表す。
D_R	右・右折	
D_L	左・左折	
S_R	右側・右手	Side B が道のどちら側にあるか表す。
S_L	左側・左手	
K_C	ちょっと・少し・近く	進む距離を表す。
K_F	しばらく・だいた	C は Close、F は Far
F_P	反対側・向かい側	B と B(or E) の関係を表す。
F_V	先・越えた・ 向こう側	P は oPposit、V は oVer
N_数字	?番目・?軒目	二つ目の信号・三つ目の家などの序数詞を表す。
E_		End 目的地。目的地の名前が明示されている場合は B と同様に「E_名前」となる。

表1 コード一覧表

道案内文は目標地点（目印となるもの・出発地点・目的地）と道路の情報が交互に現れる。この道案内文のコード列の構造を正規表現で以下のように表わしている。

道案内文コード列 = $P ? (r P ?) +$

ここで、P：目印となる物（進行方向を変換する場所・出発地点・目的地など）とそれを修飾する物に関するコードの並び。

r：進行方向や距離など目印間の移動に関するコードの並び。意味的にはPとPを結ぶ道。直線で方向と距離を持つ。

ここで使われている正規表現では、[?]:直前の記号の0または1個の繰り返し、[+]:直前の記号の1個以上の繰り返し、[()]:括弧内の記号をまとめて扱うことをそれぞれ表わしている。

なぜP、rが交互に現れるかということ、もし目的地まで行くまでにどこかで曲がらねばならないとき、どこで曲がるか説明するためには具体的な目印（建物や交差点などの2本以上の道の集まる場所）が必要となり、また次に曲がるときにも同様な目印が必要となり、その目印どうしの間は当然道路でつながっているからである（以下の例1参照）。だが実際には、次の例2のように目標地点の情報が省略されている場合も多くあるので、本研究のプログラムでは省略の判断も行なっている。

例1：右に少し行くと交差点があってそこを左折すると右手に銀行...

D__R, K__C(r), W__CROSS(p), D__L(r), S__R, B__銀行(p)
 例2：右にまっすぐ行って、それから左に曲がって…

D__R, D__S(r), D__L(r)

本研究ではこの道案内文の構造を前提としてさまざまな処理を行っている。

3 道案内文から略地図への変換

3.1 処理の流れ

処理の流れは、図1の通りである。原文とは日本語で書かれた道案内文であり、《形態素解析プログラム》はICOTで配布されている既成のものを使用している³⁾。《形態素解析プログラム》の出力する形態素ごとに区切られた原文を形態素列と呼ぶことにする。《形態素列をコードに変換するプログラム》はデータ処理言語Perl⁴⁾で書かれていて、ここで形態素列の不都合を修正してコード列に変換する。そのコード列を基に《略地図を描くプログラム》(C, Xlib使用)が、略地図を描く。

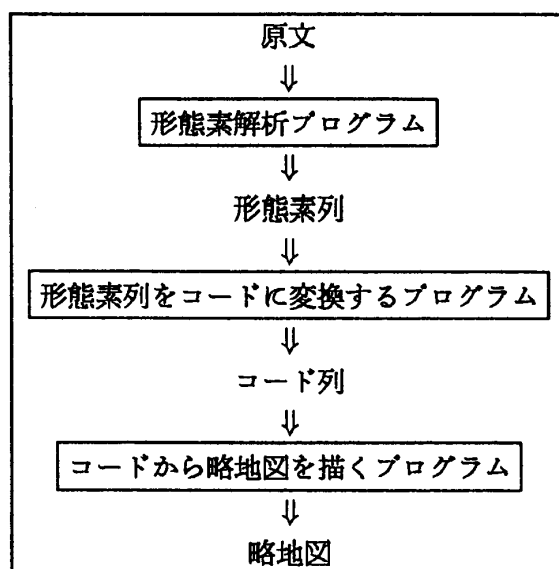


図1 道案内文から略地図への変換

(1) 形態素解析プログラム

入力された日本語文を形態素（動詞、名詞、助詞など）ごとに切り出すプログラム。

ICOTのフリーソフトウェア「形態素辞書」の単位切りプログラム*を利用している。これを利用する場合、以下のような問題点がある。

1. このプログラムは新聞記事中の日本語文を対象としているため、道案内文のような会話文の解析には少し問題がある。特に会話調語尾や省略などには対応しきれない。

2. あまりに細かく切り出し過ぎてしまう。例えば「交差点」はこのまま一語で扱いたいが、このプログラムでは「交差-点」と切ってしまう。そのほかにも「向か-い-側」「反対-側」など。そこで、次の様な対策を講じて用いている。

1. 入力文を修正する。例えば、「そしたらそこを右に行くんだよ。」という会話調表現は解析

* (C)1992 Institute for New Generation Computer Technology

ICOTフリーソフトウェアNo.33『形態素辞書』

[概要]15万語の日本語語彙に対する形態素辞書

できないので、これを、「そうしたらそこを右に行ってください。」に修正する必要がある。

2. 出力された形態素列に、細かく切り過ぎてしまった語を接合する処理を施して修正する。これは《形態素をコードに変換するプログラム》内で実現されている。

(2) 形態素列をコードに変換するプログラム

形態素解析プログラムで形態素に切り出された形態素列をコードに変換する。プログラムはデータ処理言語 Perl で書かれている。形態素を一つずつ読み込み以下の処理を行う。

処理1 形態素解析プログラムの出力には、形態素の分類番号など余計な情報が付加してあるのでこれを取り除く。

例：-向か、40 -い、160 -側、10 -に、423 ⇒ -向か-い-側-に-

処理2 形態素プログラムにより細かく分割されてしまったが、処理上一語として扱いたい語を接合する。

例：-向か-い-側-に ⇒ -向かい側-に-

処理3 各形態素毎に「道案内語→コード」辞書を引き、ヒットしたらコードを出力する。

例：-向かい側-に ⇒ , F__V

処理4 形態素が、屋・店 (B) ・街・通 (R) など始まっている場合、一個前の形態素を B, R の名前に付加する。

例：-タバコ-屋 → , B__屋, ⇒ , B__タバコ屋,

例：-電車-通り → , R__通り, ⇒ , R__電車通り,

[→] は3の処理を表わす。

これらの処理において辞書のフォーマットは、例えば [まっすぐ, D__S] [先, F__V] [学校, B] のように、[道案内語, コード] の形式をとっている。また、コードBとコードR (表1参照) については、コードに道案内語を含ませる形で出力する。そのときの例外処理が上の処理4である。これは後に略地図を描くときの、名前情報として利用するための処置である。

例：B__銀行, B__駅, B__学校

例：R__国道, R__大通り

また、絶対Bが必要だと思われる所 (1つ前のコードがSかFの所と、一つ前のコードがNでかつ2つ前のコードがSかFの所) にBが現れない場合、《コードから略図を描くプログラム》でのエラーを防ぐためコード「B__?」を生成する。例えば、次の例で「ショージ」が辞書に登録されていないとき、

例：右側にショージがあって、そこを左に… ⇒ S__R, B__?, D__L...

のようになる。

(3) コードから略地図を描くプログラム

並び順に注目してコード列を略地図描画データ (以下これをMDDと略記する) に変換し、それを用いてX-Window上に略地図を描く。図2に処理の流れを示す。

コード列を頭から順に読み込んでいき、順次処理をする。最初に〈出発地点の処理〉で、出発地点が明示されているかなどを調べる。コードD, K, R (進行方向、進む距離、道路の名前など道路に関するコード) が現れている間は〈道路を描く処理〉で、道路を描く。それ以外のコード (ランドマークの位置関係や名前に関するコード) が現れている間は〈目標地点を描く処理〉で、ランドマークを描く。

上の2つの処理を繰り返し適用し続け、最終的にコードEが現れたら〈目的地の処理〉を行い目的地が明示されているかどうかを調べた上で、解析を終了する。なぜ描く処理を [道路] と [目標

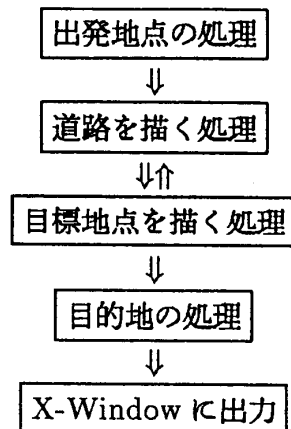


図2 コードから略地図を描くプログラムの処理の流れ

地点] の二つに分けたかについては、2. 2節「道案内文の構造」によっている。各処理では直接画面に略地図を描くのではなく、座標値などの描画データ (MDD) を生成してリストに追加していく。その後、それを元に X-Window 上に描画する。

3. 2 実行例

以下に実行例を示す。

(原文)

まっすぐ行くと、右側に教会があって、その交差点を左に曲がってまっすぐ行った右手にパン屋があり、その向かい側にある。

(形態素解析プログラムの出力 (形態素列))

まっすぐ, 24 -行, 74 -く, 181 -と, 425 -, , 3 -右側, 10 -に, 423 - 教会, 10 -が, 26 -あ, 40 -っ, 174 -て, 427 -, , 3 -そこ, 10 -の, 424 -交差, 16 -点, 10 -を, 421- 左, 10 -に, 423 -曲が, 48 -っ, 174 -て, 427 -まっすぐ, 24 -行, 74 -っ, 174 -た, 340 -右手, 10 -に, 423 -パン, 10 -屋, 10 -が, 420 -あ, 48 -り, 168 -, , 3 -その, 25 -向か, 40 -い, 160 -側, 10 -に, 423 -あ, 48 -る, 188 -。 , 2

形態素列をコードに変換するプログラムの処理2により、「-向か-い-側-」を「-向かい側-」に、「-交差-点-」を「-交差点-」にする。そして処理3により各形態素毎に辞書を引く (分かり易くするためにヒットしたものを太字で表現した)。それから処理4により、処理3で変換されたコード「B__屋」を「B__パン屋」にする。ヒットした語に対応するコードを順に出力すると次のようなコード列が得られる。

(形態素列をコードに変換するプログラムの出力 (コード列))

D__S, S__R, B__教会, W__CROSS, D__L, D__S, S__R, B__パン屋, F__P, E__

コードから略地図を描くプログラムの実行結果を図3に示す。

4 略地図から道案内文への変換

本節では、これまでの道案内文から略地図への変換とは逆に、X-Window 上にマウスを使って描かれた略地図を道案内文に変換することを考える。図4に処理の流れを示す。《略地図作成ツール》は、X-Window 上に実装され、スタート地点・ゴール地点・道・建物などの目標物・ルート (通る道) などの座標を入力とし、入力された略地図の座標データを、スタート地点からゴール地点ま

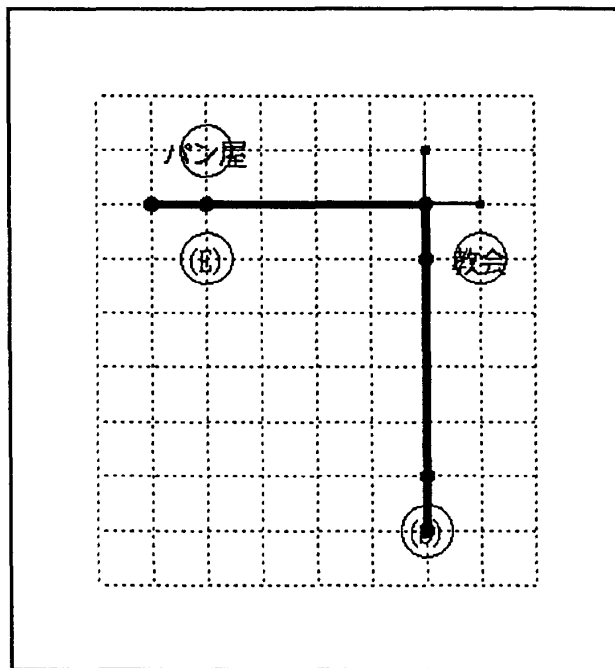


図3 コードから略地図を描くプログラムの実行結果

で辿りながら順にコードに変換する。《コードから道案内文を生成するプログラム》はそれを基に道案内文（日本語文）に変換する。

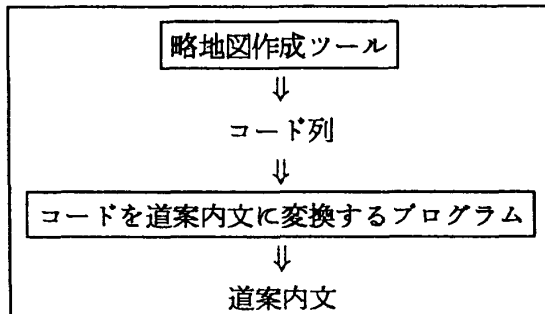


図4 略地図から道案内文への変換

4. 1 略地図作成ツール

略地図作成ツールは、X-Window 上でマウスとキーボードを用い略地図を入力するインターフェイス⁵⁾と、入力された略地図の座標などのデータ (MDD) をコードに変換する処理に分けられる。図5に処理の流れを示す。

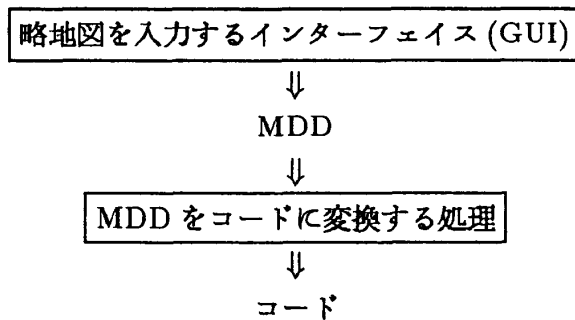


図5 略地図作成ツールの処理の流れ

(1) 略地図を入力するインターフェイス

マウスやキーボードで略地図を描き、その略地図の MDD を〈MDD をコードに変換する処理〉に引き渡す。MDD は、各要素（目印・道路など）を描き込むのと同時に生成され、MDD リストの末尾に追加されていく。ここで、あいまいさを避けるため、キャンパス上に略地図を描く時の決まりを以下のように設けている。

[略地図表現の決まり]

1. START を置いた位置から ROUTE を引き始める。
2. END、B、D は ROUTE の隣に置く。これは後に述べる〈MDD をコードに変換する処理〉で ROUTE 沿い（隣）の END と B、D を調べてコード化するという作業を行うのでそのための決まりである。
3. START から END までを結ぶ ROUTE は一筋で途切れず繋がっていなければならない、何通りもの行き方があってはならない。

(2) MDD をコードに変換する処理

略地図を道案内文に変換する第一段階として、略地図作成ツールで作成された略地図の座標などのデータ（MDD）を、コードに変換するプログラムである。まず〈暫定コード化処理〉で MDD をとりあえずコード列に変換する。その後、自然な道案内文に変換できるコード列にするため〈後処理〉を行う。図 6 に処理の流れを示す。

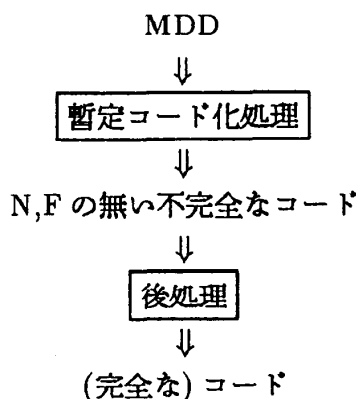
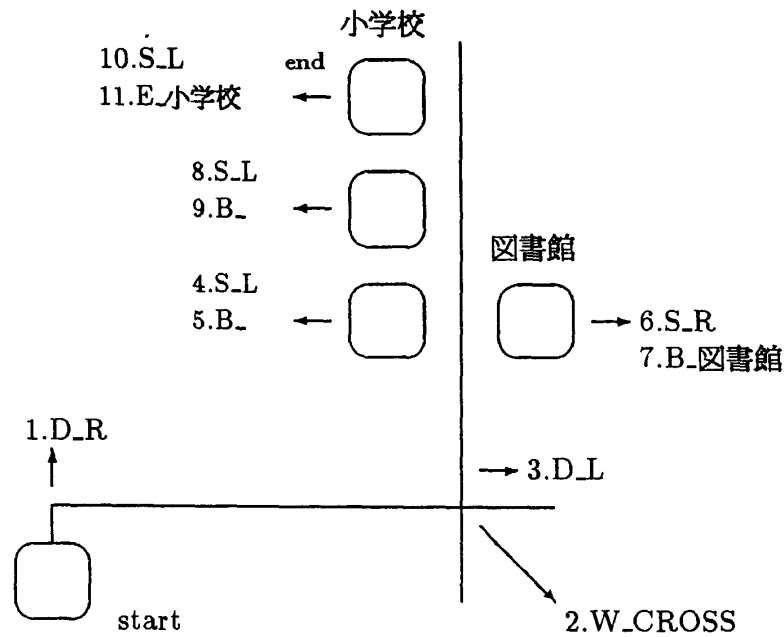


図 6 MDD をコードに変換する処理

〈暫定コード化処理〉では、出発地点からスタートし、目的地まで ROUTE を辿っていく。移動しながら ROUTE の両サイドを調べ、コード B か END が見つかったら全て「S __?, B __名前」の形で出力する。出力するコードは S, B, D, W, E のみである。図 7 を参照。

〈後処理〉では、前の処理で出力されたコード列の中の連続する S の列を必要に応じて N, F を用いたコード列に変換することによって、そのコード列が《コードから道案内文に変換するプログラム》で自然な道案内文に変換されるようにする。

例えば、図 7 の「..., S __L, B, S __R, B __図書館, S __L, B, S __L, E __小学校」というコード列は、敢えてそのまま道案内文にすると「... 左側に建物があって右側に図書館があって左側に建物があって左側に小学校がある。」となる。しかし「... 右側の図書館の向かい側の三軒目に小学校がある。」と言った方が自然な表現であると思われる。そのような理由で〈後処理〉では先のコードを「..., S __R, B __図書館, F __V, N __3, B __小学校」と変換する。



⇓
D_R, W_CROSS, D_L, S_L, B_-, S_R, B_図書館, S_L, B_-, S_L, E_小学校

図7 暫定コード化処理の実行例 (数字の順にコード化)

4. 2 コードを道案内文に変換するプログラム

略地図からコードに変換するプログラムから出力されたコードを、日本語の道案内文に変換する。このプログラムはデータ処理言語 Perl でかかっている。処理の手順は以下の通りである。

1. 「コード→道案内語」辞書を引きコード列を道案内語列に変換する (辞書引き)
2. コードの並び順に注目して自然な文になるように各道案内語の語尾を調整する (語尾調整)

(1) 辞書引き

コードから道案内語への辞書の内容は以下の通りである。

- D_L 左に
- D_R 右に
- D_S まっすぐ
- K_C 少し
- K_F しばらく
- W_CROSS 十字路
- W_T T字路
- S_R 右側
- S_L 左側
- F_V 向こう側
- F_P 反対側

変換例：D_L, W_CROSS, D_R, S_L, E_ ⇒ 左に十字路右に左側

しかし、このままでは日本語になっていないので次の処理で調整する。

(2) 語尾調整

コードの並び順に依り、分岐処理で語尾を決定している。分岐処理のルールの一部を以下に示す。

- S 右側 ・左側の処理
 - もし前のコードがNならば、語尾は「の」
 - そうでなければ、語尾は「に」
 例：N__3, S__R, B__銀行 ⇒ 3つ目の右側の銀行...
 例：S__L, B__公園 ⇒ 左側に公園が...
- F 向こう側 ・反対側の処理
 - もし次のコードがNならば、語尾は「の」
 - そうでなければ、語尾は「に」
 例：F__P, N__3, B__銀行 ⇒ 反対側の3つ目の銀行...
 例：F__V, B__公園 ⇒ 向こう側に公園が...
- E 目的地の処理
 - もし前のコードがNならば、語尾は「です。」
 - そうでなくてもしEが無名ならば、語尾は「ある。」
 - そうでなければ、語尾は「がある。」
 例：N__3, E__ ⇒ 3つ目です。
 例：S__L, E__ ⇒ 左側にある。
 例：F__P, E__駅 ⇒ 反対側に駅がある。
- N 序数詞の処理
 - もし次のコードがEで無名ならば、語尾は「つ目」
 - そうでなくてもし次のコードがEで名前ありならば、語尾は「つ目が」
 - そうでなければ、語尾は「つ目の」
 例：N__3, E__ ⇒ 3つ目です。
 例：N__3, E__銀行 ⇒ 3つ目が銀行です。
 例：N__3, B__ ⇒ 3つ目の建物...

その他に

- K 距離の処理
- D 進行方向の処理
- R 道路の処理
- B 建物などの処理
- W 交差点などの処理

などがある。

4. 3 実行例

図8に示すような略地図作成ツールで描かれた略地図をまずツール自身がコードに変換する。次にコードを道案内文に変換するプログラムが、表2に示すように道案内文：

私の家からそのT字路を右に行って、十字路で商店街を左に行って、左側に本屋があってそのT字路を左に行って、右側に郵便局があってそのT字路を右に行って、右側に図書館がある。

を出力する。

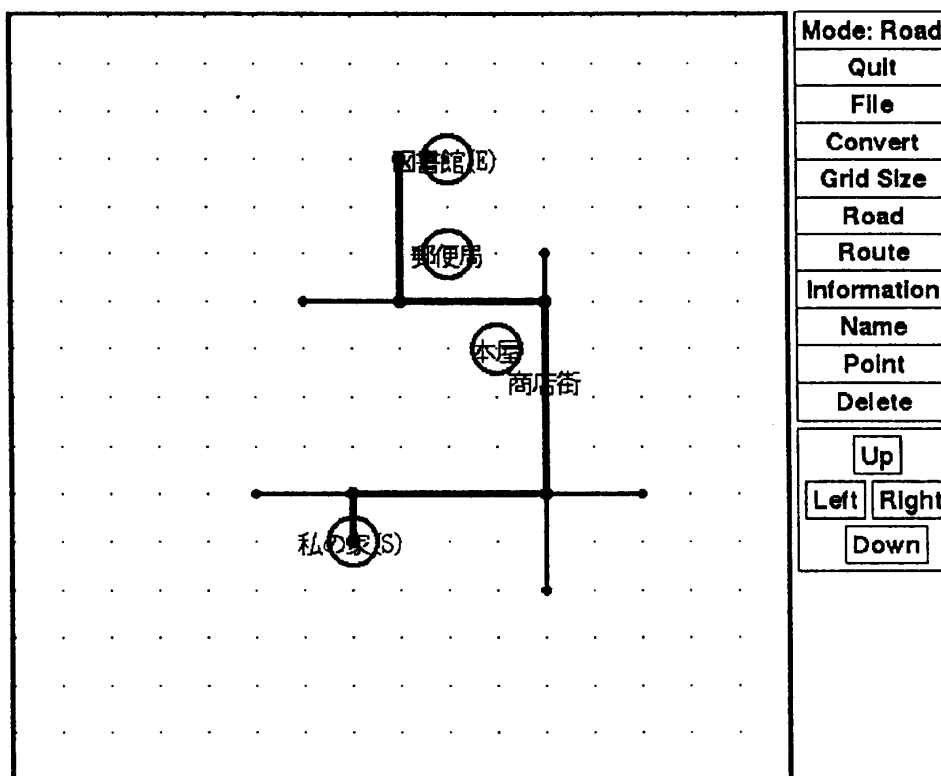


図8 略地図作成ツール

認識されたコード	出力された道案内語+語尾
B_私の家	私の家から
W_T	そのT字路を
D_R	右に行って、
W_CROSS	十字路で
R_商店街	商店街を
D_L	左に行って、
S_L	左側に
B_本屋	本屋があって
W_T	そのT字路を
D_L	左に行って、
S_R	右側に
B_郵便局	郵便局があって
W_T	そのT字路を
D_R	右に行って、
S_R	右側に
E_図書館	図書館がある。

表2 コードから道案内文への変換

5 実際的な応用例による評価

これまでに述べてきたシステムについて、実用に耐え得るかどうかテストしてみた。

5.1 道案内文から略地図への変換

以下の道案内文は、“国語入試問題必勝法”（清水義範，講談社文庫，1992年11月13日，pp. 52）

からの引用である。

駅を降りて右へ出て、その街道を道なりに三分ほど歩きますと角に時計屋がありますから、そこで左に曲がってそのまま進んで、銀行を越したところにあるタバコ屋の向かい側の生け垣の家です

これをこのまま《形態素解析プログラム》に引き渡すとエラーがでてうまくいかないで、エラーのでないような必要最低限の変更（下線の部分）を行う（原文変更）。

駅を降りて右へ出て、その街道を道なりに三分ほど歩きますと、角に時計屋がありますから、そこで左に曲がってそのまま進んで、銀行を越したところにあるタバコ屋の向かい側の生け垣の家です。　

これを再び《形態素解析プログラム》に通し、その出力を《形態素列をコードに変換するプログラム》に引き渡す。以下がその結果である。

B__駅, D__R, R__その街道, B__時計屋, D__L, D__S, B__銀行, F__V,
B__タバコ屋, F__P, B__家, E__

これを《コードから略地図を描くプログラム》に通した結果が図9である。家・銀行・タバコ屋の位置があきらかにおかしい。原因はB__時計屋・B__銀行の前にSが無く、位置が確定できないからである。時計屋の場合はともかく、銀行の場合は後にFが続くので位置の確定が不可欠になる。とりあえずB__銀行の前にSを挿入してみる（コード変更）。

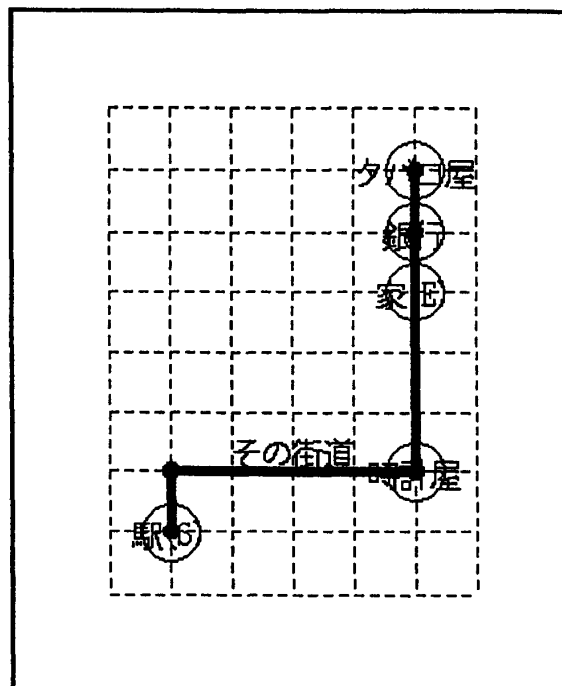


図9 正しくない略地図

..., D_S, S_R, B_銀行, ...

これを通した結果を図10に示す。

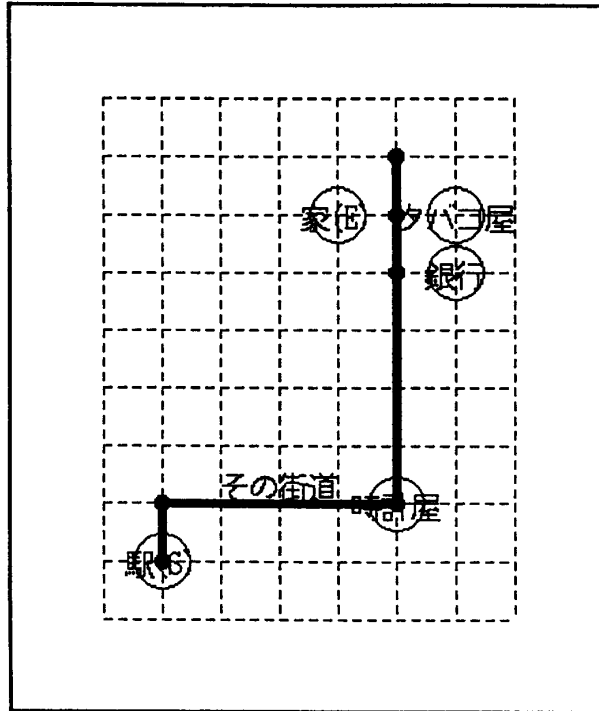


図10 正しい略地図

実際に数人の人に、アンケート形式でチラシなどに載っている略地図を見せて道案内文を書いてもらった。これらの道案内文をUNIX上のファイルに打ち込んで処理を行なった。全部で12文である。

目標は、原文を人が見て理解できる略地図（分かる略地図）に変換することである。結果は表3の通りとなった。原文変更の内訳は、英語子文字を大文字に（5箇所）、平仮名を漢字に（辞書にヒットしないため）（3箇所）、アラビア数字を漢数字に（2箇所）、形態素の接続上の問題（1箇所）となっている。このうち形態素の接続上の問題とは、「...を越えてすぐの左側...」が《形態素解析プログラム》に受けつけてもらえず、「すぐ左側」に変更したというものである。コード変更しなければならない原因は、「広島駅に向かってまっすぐ」などの「～に向かって」の処理がなされていないことによるものと、方角の処理の不備（北上する＝D_Sと認識できなかった）があげられる。

そのままの文で「分かる略地図」に変換される	4
原文変更すれば「分かる略地図」に変換される	6
コード変更しなければならない	2

表3 変換の結果

その他の問題点として、固有名詞（特にローカルなもの）に対する辞書の語彙の少なさがあげられる。今回ヒットしなかったものに、日通・ローソン・広銀・広総・西条プラザ・ブルーバールなどがある。また必要な箇所でSが無いことが多い。全12文の中で9箇所あった。これは、人は普通「現

地へいけば分かる」という考えで道案内をするからだと思われる。

5. 2 略地図から道案内文への変換

本研究のアルゴリズムを知らない人に《略地図ツール》を使って実際に略地図を描いてもらった。ツールの使い方については口頭で教えた。

[総合科学部から理学部までの道順]

実際に描かれた略地図は図11、最終的に得られる道案内文は、

総合科学からT字路を右に行つて、左側に西体育館があつて向こう側に西グランドがあつてその十字路をまっすぐ行つて、左に行つて、右に行つて、右側に学校教育があつて右に行つて、右側に教育があつて左に行つて、左側に北体育館があつてそのT字路を右に行つて、T字路をまっすぐ行つて、左側に駐車場があつてその十字路を右に行つて、左側に理学部がある。

となった。

ツール上での略地図の描き方・表し方にはかなり制約があるので、思った通りに変換してくれる。しかし出力する道案内語が固定されているので文章としては少しくどく感じてしまうのが難点である。

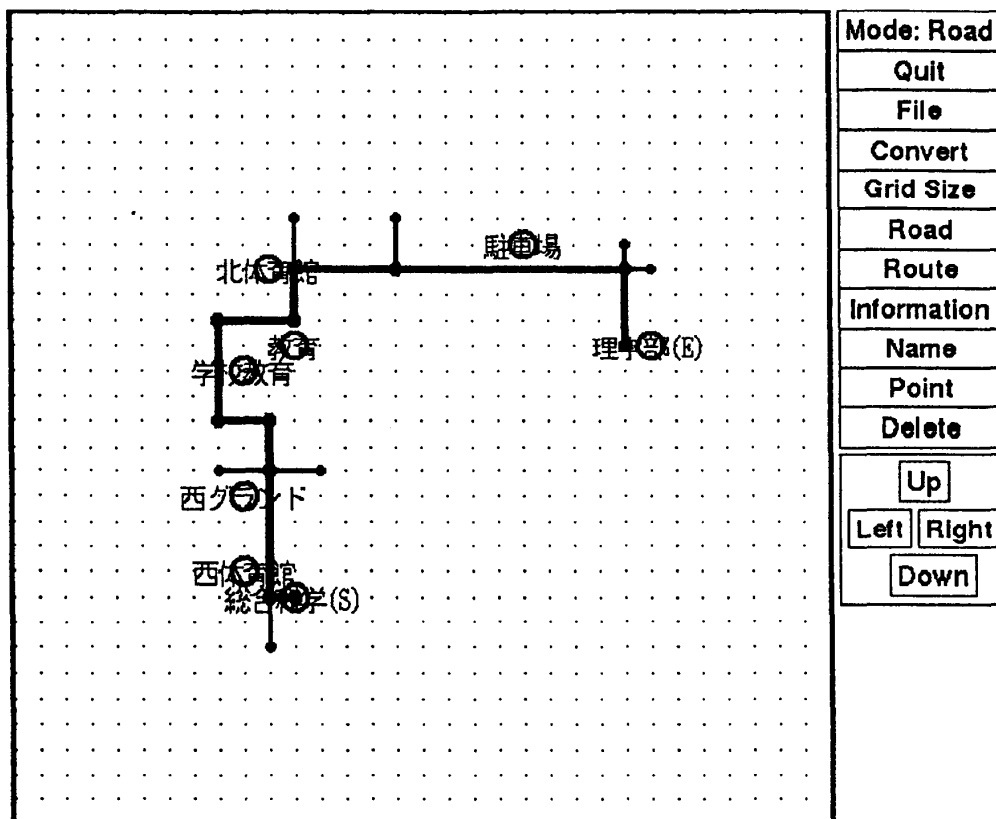


図11 総合科学部から理学部まで

6 おわりに

本研究では、道案内文（言語情報）と略地図（視覚情報）とをコード（道順のイメージ）を通して結び付けて、道案内文を略地図へ、また略地図を道案内文へ変換するプログラム群を開発した。様々な問題を抱えているが、かなりの割合で意図した結果が得られた。以下、問題点について述べる。

このコード体系ではまだまだ不満な点が多い。もっと厳密に定義する必要があるだろう。今回苦しめられた「橋を渡る」「線路に沿って」「駅に向かって」といった表現に対応できるコードが必要であろう。

形態素解析ソフトウェアに関しては、今回はとり合えず手近な ICOT のものを用いたが、ICOT 以外にも様々なものが出回っているので、それらを試してみても本研究に最適なものを捜す必要がある。

コードへの変換に関しては、本研究のようにキーワード抽出だけでは限界がある。本格的な日本語の解析（特に意図の解析）が必要である。

辞書に関しては、ローカルな地名や店の名前や目印に用いられる抽象的なものなどにどう対処すればいいのかという問題が残された。これらの語ををいちいち登録していったのでは辞書が膨大になってしまう。それらの語を識別するアルゴリズムが必要だろう。この点に関して、査読者から有益な助言を頂いた。「Xがある」あるいは「Xがあつて」のような目印の存在構文のパターンを、そのまま辞書に登録しておく方法である。本研究はその後も発展させていっており、次の段階として品詞連接パターンを用いる方法を検討している。存在構文のパターンもこの中に取り込んで行きたいと考えている。査読者からの提案に感謝します。

略地図ツールに関しては、まだまだ使いにくい点がある。いちいち略地図を描くよりも、実際の地図を取り込んで通り道をなぞる方法が理想である。

コードを日本語に変換する方法に関しては、今のようないきあたりばったりでは、長い文章のとき同じ語が何度も出てくどくなってしまう。一つのコードに色々な語を与える・ある程度の長さになったら「。」で文を区切るというような処理が考えられるが本質的な改善にはならない。やはり本格的な文章生成システムが必要だと思われる。

どんなに正確な略地図や道案内文も、実際に目的地まで導くことができなければ役に立たない。また逆にどんなに曖昧な略地図・道案内文でも現地に行ってみると役立つ。しかし曖昧であろうと正確であろうと、目的地につれていってくれる略地図・道案内文とそうでないものとは何かしら違いがあるだろう。それを知るために実際の地図のデータをワークステーション上に入力しておき、様々な略地図・道案内文を用いて目的地に辿れるかどうかのシュミレーションが必要である。これは今後の課題である。

参 考 文 献

- (1) 山田, 西田, 堂下, 2次元平面におけるポテンシャルモデルを用いた位置関係の推定, 情報処理学会論文誌, vol.29, No.9, pp.824-834, 1988.
- (2) 山田, 網谷, 星野, 西田, 堂下, 自然言語における空間描写の解析と情景の再構成, 情報処理学会誌, vol.31, No.5, pp.660-672, 1990.
- (3) TRIE 構造辞書リリースのためのドキュメント, ICOT Free Software No.33, 1992.
- (4) Larry Wall and Randal L. Schwartz (著), 近藤嘉雪 (訳), Perl プログラミング, ソフト

バンク株式会社, 1993.

- (5) 安居院猛, 永江孝規, Xアプリケーション・プログラミング(2)—Athena ウィジェット編—, 株式会社新紀元社, 1992.