

「ヤチマタ」における擬似日本語

諸橋正幸・藤崎哲之助・鷹尾洋一・間下浩之・渋谷政昭
(日本アイ・ビー・エム(株) サイエントフィック・センター)

1. はじめに

大きなデータベースを利用し、それを参照しながら調査、意思決定、その他諸々の作業を行なうというような分野が計算機利用の分野で増大している。

データベース照会用の言語として自然言語により近い形のもの(これを我々は擬似日本語と呼んでいる)を導入する事は、今後の計算機利用者の増大に対処するにあたって非常に重要な事である。この認識の上に立って開発されたのがデータベース照会の為のプロトタイプシステム「ヤチマタ」である。これは、かなり日常語に近い流暢な表現により照会が出来るようになっている。また、意味の記述に関しては、質問の対象をデータベースに限定した事により、具体的で扱い易い形となり、比較的単純な形でかなりの成果を収めた。

このシステムの評価の為、三つの異なる適用分野を当嵌め、小規模な実験を行なっている。それらは、

- (1) 国内の航空便の時刻、空席などの照会
- (2) 倉庫にある部品の在庫などの照会
- (3) 都道府県単位の地域統計データに関する照会

である。「ヤチマタ」は現在図1のような機種構成で稼動している。この図中のA, B, Cは照会文入出力用のCRT(3270カタカナ付)の制御を行なう仮想計算機であり、実際の照会処理はYで行なう。

なお、「ヤチマタ」命名の由来は、日本語文法の開祖とも云える本居春庭著「詞八衢」(ことは"のやちまた、1808)による。

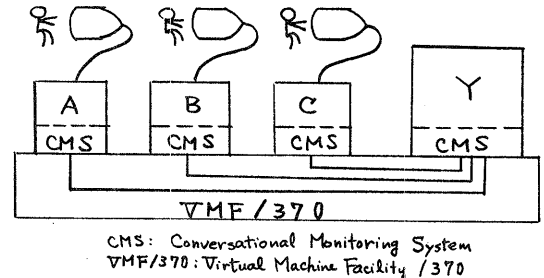


図1. ヤチマタの稼動構成図

2. ヤチマタの特徴

ヤチマタの機能上の特徴として次のようなものが挙げられる。

- (1) 日常語に近い言語(擬似日本語)による照会、表による応答。
- (2) カタカナ入力、カタカナ出力。
(漢字出力について現在検討中)
- (3) 汎用性(多くの業務に適用可能)
- (4) 関係形式データベースの利用。
- (5) 拡張機能(既知の言葉により、新しい表現を使用者が定義出来る)
- (6) 生成機能(データベースから照会に必要な辞書、表記述等を生成する道具を提供)
- (7) 分かち書きの自由さ(分かち書きの規則を使用者に強いない)
- (8) 曖昧さの処理(自動的に解消出来ない場合には、照会者との対話により解消)
- (9) 出力の編集(応答となる表に対し、ある程度の編集機能を使用者に許している)

3. テータモデル

ヤチマタで採用されているテータモデルは上述の通り関係形式モデルであるが、文章(特に句構造)との対応が明確となる様に特に考慮が払われている。しかも、文を名詞句の形で捉えるので、名詞句モデルである。

照会文は、「<名詞句>は?」なる

形で規定され、構文解析によりこの名詞句がいくつ名詞句からどの様に構成されているかを構文規則により解析する。一方、意味解析においては、各名詞句に概当する関係形式を対応させ、名詞句を繋ぎ合わせる構文規則に検索基本関数を対応させる事により、最終的に照合文に合致した関係形式を作り出す。この検索基本関数は集合演算により定式化できる。

名詞表と動詞表

ヤチマタのデータモデルは関係形式であると述べたが、より詳しく見ると二つの型の関係形式が存在する。すなわち、名詞表と動詞表である。名詞表は名詞（文法上の意味での名詞とは必ずしも一致するものではなく、ある集合を言い表す言葉ならば何でもよい。例えば「倉庫の名前」なども許される）の意味を規定し、図2の如き構造を持つ。このモデルが一般の関係形式モデルと異なる点は、各列に対して同等の扱いをしない事である。第一列を対象属性と呼び、この列に現われる要素の集合 $\{v_{11}, v_{12}, \dots, v_{1m}\}$ を示す名詞（集合語）を名詞NNで代表させる。第二列以降（もしあれば）を限定属性と称し、NNを限定・修飾する属性である。修飾にあたっては、格標識が「後置詞」として働く。ここで後置詞とは格助詞を拡張した詞辞を指し、「における」等の詞も含める。図2の例において名詞表「社員」を用いる名詞句として、「BIMの社員」がある。

一方、動詞表は動詞（または形容詞、形容動詞）を規定し、図3のような構造である。ここでは、名詞表のような対象属性は存在しない。この動詞表は連体修飾句として使われ、名詞表の補助的役割を果す。例えば、「BIMに勤める人」という句に対して動詞表「勤める」は、定義域が「人」である列を対象属性とする名詞表に変換されてこの連体修飾句に繋がる。従って以後

の構文規則の適用にあたっては、全て名詞表の形に対する検索基本関数と同じ扱いとなる。

<一般形>

NN	C ₂	...	C _n
D ₁	D ₂	...	D _n
v_{11}	v_{12}	...	v_{1n}
$\vdots$	$\vdots$		$\vdots$
v_{m1}	v_{m2}	...	v_{mn}

対象属性 限定属性

NN: 名詞
C₂...C_n: 格標識
D₁...D_n: 定義域名

<例>

社員	の
人	会社
タロウ	BIM
ジロウ	DCC
サロウ	EDC

図2. 名詞表とその一例

<一般形>

VV	C ₁	...	C _n
D ₁	D ₂	...	D _n
v_{11}	v_{12}	...	v_{1n}
$\vdots$	$\vdots$		$\vdots$
v_{m1}	v_{m2}	...	v_{mn}

VV: 動詞, 形容詞,
形容動詞
C₁...C_n: 格標識
D₁...D_n: 定義域名

<例>

勤める	
が	に、へ
人	会社
タロウ	BIM
ジロウ	DCC
サロウ	EDC

図3. 動詞表とその一例

検索基本関数

検索基本関数には次のようなものがある。

- (1) 制約演算（正順制約，逆順制約）
- (2) 集合演算（和，差，積）
- (3) 算術演算（+，-，*，/，...）
- (4) 集合関数（合計，平均，...）

制約演算とは、ある属性値を持つn個組を取出す演算であり、図4に示されるように、制約される属性が限定属性の場合を正順制約と呼び、制約を受けるのが対象属性である時を逆順制約と云う。他の演算についての説明は省略する。

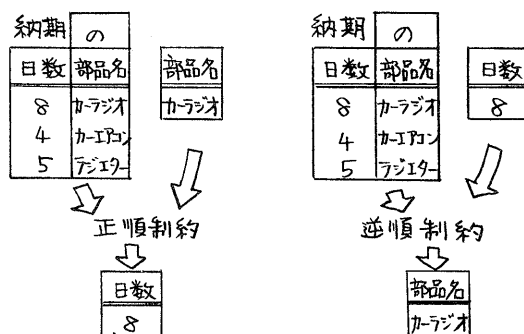


図4. 正順制約と逆順制約

4. 照会文の処理

照会文の処理過程を図5に示す。各段階における処理の詳細説明を以下で行なう。

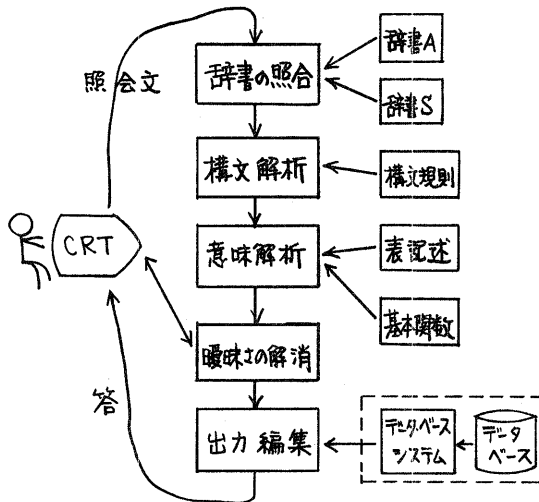


図5. 照会文の処理過程

4-1) 辞書の照合

入力された文字列に対し、空白は単語の区切りとして積極的に利用する事は、分かち書きの規則を使用者に強要する事となり、ヤチマタの設計目標に反する事になる。しかし、逆に空白を全く無視して辞書のエントリーに合致するものを捜す方法を取ると、システム側の負担が非常に大きくなる。また照会文中にアルファベットが混じる場合には、より無駄な処理が多くなる。そこで、ヤチマタでは次のような工夫を行なっている。

第一は入力文字群（空白を除く）を仮名、数字、アルファベット、特殊記号等に分類し、それらの文字の組合せに対し、間にはいる空白が省略可能かどうかを調べる。この処理に使われる表を図6に示す。この表により、入力される照会文にあって、数字間の空白、英字間の空白のみが意味を持つ（省略出来ない）としている事がわかる。例えば、「cock tail」と「cocktail」、

「32 64」と「3264」は各々区別して取扱かう。

```

' ' PRIMITIVE <BLANK>;
'0123456789' PRIMITIVE <DIGIT>;
'ABCDEFGHIJKLM' PRIMITIVE <ALPHA>;
'NOPQRSTUVWXYZ' PRIMITIVE <ALPHA>;
'アイウエオカククサシセツ' PRIMITIVE <KANJI>;
'タチツテトナニヌネノハヒフヘホ' PRIMITIVE <KANJI>;
'マミムメモヤヨラリルレロワラン' PRIMITIVE <KANJI>;
'アイエオヤユヨ' PRIMITIVE <KANJI>;
'.'@#$%&'()-/¥' PRIMITIVE <MARK1>;
'++&|'~=%.:;<>' PRIMITIVE <MARK2>;
',「」:;?' PRIMITIVE <SPECIAL>;
  
```

前	後	DIGIT	ALPHA	KANA	MARK1	MARK2	SPECIAL
DIGIT	0	1	1	1	1	1	1
ALPHA	1	0	1	1	1	1	1
KANA	1	1	1	1	1	1	1
MARK1	1	1	1	1	1	1	1
MARK2	1	1	1	1	1	1	1
SPECIAL	1	1	1	1	1	1	1

0: 二つの文字の間の空白が意味を持つ
1: " " 省略可能

図6. 空白処理指示テーブル

第二は、一つの言葉を書く場合にその途中に空白を許すか否かを辞書に記述しておく方法である。これは特に「照会文」のような合成語を記する場合に有効である。辞書への記述方法は次の辞書の説明の処で行なう。

辞書は、適用業務に依存する辞書Aと依存しない辞書Sとから成る。辞書に示される属性は、その言葉の範疇（日本語文法の品詞分類に相当するもの）、構文情報（範疇の細分類の為の情報）、意味情報（意味を記述する）の三種類から成る。図7は辞書Aの例を模式的に示したものである。エントリー中の文字列に現われる図印は、その箇所に任意個の空白を許す事を示している。逆に図印のない箇所に空白が来るような入力文字列は、辞書中の文字列と一致したとは見做さない。例えば、「納品する」、「納品 する」は辞書のエントリー「納品図する」と合致するが、「納 品する」、「納品す る」とは、合致しない。

範疇（カテゴリー）の列に現われる

VTRAIN, PTRAIN, CTRAIN, ITRAIN は各々動作・状態語(用言にあたる)、後置詞、集合詞(名詞)、その要素の表現(固有名詞)に対応している。ここで後置詞は、名詞表あるいは動詞表の格標識に対応する格助詞で、名詞表に使われるか動詞表に使われるかの違いを構文情報として持つ(NVSW)。用言に対しては、その活用形がすべて登録される。例えば「納品する」に対して「納品 図さ」(未然)、「納品 図せ」(未然)、「納品 図し」(未然)、「納品 図し」(連用)、「納品 図する」(終止)、「納品 図する」(連体)がそれぞれ登録されることになる。

辞書についても同様が構造を持つており登録されている詞辞は、助動詞、補助動詞(だ、いる)、助詞(は、を)、接続詞(および)、代名詞(これ、あれ、これら、あれら)、コマンド用辞句(LIST、?)等である。このうち活用するものについては辞書Aと同様、すべての活用形が登録されている。

エントリ	範 疇	構文情報	意味情報
納品 図する	VTRAIN	活用のかた=サ変、 活用形=連体、 動詞表の構造記述	DB内でのキー
納品 図し	VTRAIN	活用のかた=サ変、 活用形=未然、 動詞表の構造記述	DB内でのキー
。			
所在地	CTRAIN	名詞表の構造記述	DB内でのキー
倉 庫	CTRAIN	名詞表の構造記述	DB内でのキー
。			
に 図 け る	Ptrain	格標識=13 連体/連用の区別(NVSW) =連体	—
が	Ptrain	格標識=14 連体/連用の区別(NVSW) =連用	—
。			
港に 図 せ	ITRAIN	定義域=5	—

図7. 辞書Aの構造

数字の場合は、起こり得る終ての数を登録する訳にはいかないので、文字の並び規則を与える。(図8)

```

<INTEGER>:=<DIGIT>M6;
<DECIMAL>:=<INTEGER><'.'><DIGIT>OM5;
<FLOAT> :=<DECIMAL><'E'><'+'>,<'-'>>0
               <DIGIT>M2;
<NUMBER> :=<INTEGER,DECIMAL,FLOAT>;

```

図8. 数連系の並び規則

4-2) 構文解析

構文解析はREL⁽¹⁾、MIND⁽²⁾システムで用いられた手法に基づく上昇型シンタックス、ディレクティッドな方法を用いている。

構文規則について

辞書の照合処理により、入力された照会文は、辞書の合致したエントリから得られる範疇、構文情報、意味情報を含む箱(ノード)を付加される(図9)。

このようにして得られた有向グラフを用いて構文解析が行われる。ここで特に構文情報が参照され、その適用順序を制御するのに利用される。

解析に用いられる構文規則は、通常の文脈自由文法と同じ記法により書かれるが、左辺として複数個の項が許される。又右辺の各項に、構文情報の制約条件を与えることができ、この条件を満足する場合にのみその構文規則の適用がうけられる。一方左辺に現われる項に構文情報を与え機能を持ち、これは規則適用後にできるノードに構文情報を設定する役目を果たす。

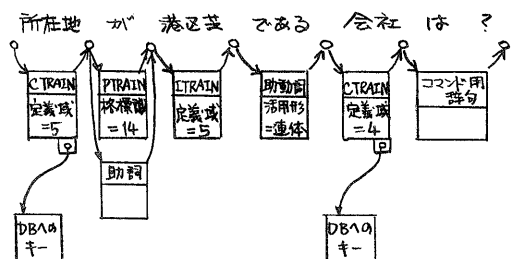


図9. 辞書照合処理

図10において、入力文の一部に<C>と<P>の二つの連続する箱が付加されている時、図中に示される構文規則が与えられると、<C>の箱の構文情報 DOMAIN の持つ値が規則の右辺第一項にある制約条件を満足するので、この規則の適用により箱<A>ができる。この箱は左辺の構文情報設定の指示に従い、右辺第一項の DOMAIN の値が与えられる。

構文規則:

<A: DOMAIN=DOMAIN(1)> ← <C: DOMAIN=3>
<P>;

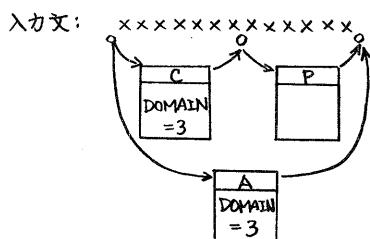


図10. 構文規則の機能

構文規則の左辺には、さらに規則適用の際に呼び出される関数を指定することができる。この関数は二種類存在し、呼び出される時点により区別される。適用前に呼び出される関数を直前割出し手続と呼び、規則適用後に呼び出されるものを直後割出し手続と呼ぶ。この二種の関数は、構文規則では引数を指定せず、そのかわり右辺の項に対応する箱が、自動的に引数として渡される。従って規則では記載できない構文情報、意味の樹の検査をプログラムにより行なうことができる。

構文規則の適用

最初に適用される規則は、辞書により見つけた文字連系<CTRAIN, ITRAIN, VTRAIN, ...>をより一般的な形でまとめ、のちの規則を当飲め易い形に昇格させるための規則である。図11で示されるように、辞書A(適用業務に依存する辞書)で見つかった詞辞は、<NP>

(Noun Phrase, 名詞句)、<VP> (Verb Phrase, 動詞句)、<PP> (Post-Position, 後置詞) の名でまとめられる。

R1. NUMBER

<NP: +NUM, +SING...> ← <NUMBER>;

R2. ITRAIN

<NP: +SING, NUM=NUM(1)...> ←
<ITRAIN>;

R3. CTRAIN

<NP: -SING, NUM=NUM(L)...> ←
<CTRAIN>;

R4. PTRAIN

<PP: NTSW=NTSW(1)...> ← <PTRAIN>;

R5. VTRAIN

<VP: INF1YP=INF1YP(1), INFLX=INFLX(1)
...> ← <VTRAIN>;

図11. 最初に適用される規則群

この他の規則はほとんど以上で出て来た<NP>、<PP>、<VP>を組み合わせてより大きな<NP>にし、最終的には<SS> (Sentence) に組みあげてゆく為の規則である。以下でそれらのうち代表的なものをいくつか紹介する。

名詞句に関する構文規則

名詞句に関する構文規則の代表的なものとしては、

(1) <NP: DMB=DMB(2)> ←

<NP: DMB=DMB(2), -SING>

<NP: +SING>;

これは「部品カーラジオ」のような句を処理する規則であり、この場合、右辺における構文情報の制約条件としては第一項、第二項とも同一の定義域を持つ (DMB=DMB(2)) ことと、第一項は<CTRAIN>から<NP>になつたもの、すなわち集合名詞であり (-SING) 第二項は逆に<NUMBER>、<ITRAIN>から<NP>になつたもの、すなわち固有名詞か数詞である (+SING)。

(2) <NP:DMB=DMB(1),-SING> ←

<NP>< DANTEI>

<NP:DMB=DMB(1),-SING>;

「マネージャーである人」のような句を扱う規則であり、< DANTEI> は断定の助動詞(だ、です)、補助動詞(ない、ある、いる)、打消しの助動詞(ない、ぬ)、助詞(は)などを組合せたものを示す。

(3) <NP:DMB=DMB(5),-SING> ←

<NP:-SING>< GA>

<NP:DMB=DMB(1),+SING>

< DANTEI>< NP>;

「所在地が港区芝である会社」に代表される句を処理する規則で< GA> は格助詞(が、の)を表わす。またこの規則に関連して、次のような書き換え規則がある。簡單の為に略式で表わすと、

<NP>が<NP>であり<NP>が

<NP>である<NP> ←

<NP>が<NP>であり<NP>である

<NP>;

ここで、右辺の<NP>がの部分が見返されることにより、(3)を適用できる形へ書き換えが行なわれる。

(4) <NP:DMB=DMB(3),-SING> ←

<NP><PP:NVS=1><NP:-SING>;

「倉庫の部品」の例で示される形を扱う規則である。「1970年の東京の人口」のような例でみられるように、この規則を繰り返して何回も適用する場合には、右から左へ適用されていくように構文情報を利用した制約条件が付けられている。従って上記の例では、

(1970年の(東京の人口))

という具合に解析が行なわれる。

なお、<PP>の構文情報として現われている NVS は、連体/連用修飾のどちらで使われるべき後置詞であるかを指定する情報で、NVS=1とは連体修飾をするのに使われる後置詞に対してのみ(4)が適用されるべきであることを示している。

用言を扱う構文規則

連体修飾句として現われる用言を扱う代表的な規則には次のものがある。

<NP> ← <NP><PP:NVS=0><VP>

<NP>;

「カーラジオを納品した会社」で示される形を扱う。<VP>には、動詞、形容詞、形容動詞の用言が単独で存在する場合だけではなく、「ない」、「ある」、「ぬ」、「いる」などの助動詞、補助動詞が付くこともある。

接続詞の取扱い

接続詞、接続助詞には、および、または、と、か(これは副助詞として分類されている)、並びに、等があるが、これらを使う場合、次のような問題が起きる。「AとBかC」なる句に対し、

((AとB)かC)

(Aと(BかC))

の二種類の解析が可能となる。どちらのケースが日本語として自然であるかは一概には云えないし、なおかつ、両者の意味は異なっているので照会者の判断を仰ぐ以外に方法はなさそうである。

こうした形で起きる曖昧さを除く為に、ヤチマタでは and, or の混在を許さないという制約を設けている。この制約により、

((AとB)とC)

(Aと(BとC))

は、どちらの解析を行なっても結果は同じとなり、右から左への解析だけを行なえば充分となる。

代名詞の取扱い

ヤチマタの照会文においては、「それ」、「これ」、「これら」などの代名詞、「その」、「この」等の連体詞を用いて、前の照会文に対する答の表を参照する事が出来る。この機能を使う為の基本文型は、

<代名詞,連体詞>のうちで<NP>のものは?

<代名詞,連体詞><NP>の句を含む文

数詞、その他の取扱い

演算 $+$, $-$, $*$, $/$, $($, $)$ 等も照会文中に記する事が出来る。これらの算術演算は、通常の順序で演算が行なわれるような構文規則が用意されている。

以上で述べた規則を適用する事により、照会文の解析が進み、それに従って作り出される<NP>は照会文中のより大きな範囲を示すようになる。これを最終的な文に作りあげるのが文に対する規則である。ヤチマタでは照会文の文型として、「<名詞句>は？」の形だけを取扱っているのので、文を認識する為の規則は簡単な形をしている。

所在地が 徳島県 である 会社 は？

```
graph TD
    CTRAIN1[CTRAIN] --> J1(( ))
    PTRAIN[Ptrain] --> J1
    ITRAIN[ITRAIN] --> J1
    J1 --> DANTEL[DANTEL]
    DANTEL --> J2(( ))
    CTRAIN2[CTRAIN] --> J2
    J2 --> ENDMARK[ENDMARK]
    ENDMARK --> J3(( ))
    NP1[NP] --> J3
    J3 --> SS1[SS]
    SS1 --> SS2[SS]
```

この時点で実際には意味の樹も作られ、これが次の段階である意味解析処理の入力となるのであるが、構文解析の樹と同時に説明を行なうと煩雑になるのでここでは省略した。次節で詳しく触れる事にする。

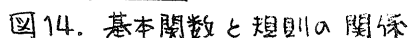
意味の樹を作っていく作業は、構文規則を順次当て嵌めていく手順の中で行なわれて来た。こうして作られた意味の樹を評価し、データベース探索の手順を生成するのが意味解析の仕事である。意味の樹の評価について述べる前に、ここで意味の樹の生成についてまず説明する事にする。

左辺に現われる項に FPE-〈関数名〉
(〈引数〉, ..., 〈引数〉) と書く事により、
〈関数名〉を根とし、引数の数だけ枝を持つ (引数がない場合もある) 樹が意味の樹としてこの項に付加される。
ここで引数は、右辺の指定した項の意味の樹あるいは構文情報を持つ値である。
図 13 に FPE 機能を用いた構文規則とその適用に関する例を示す。

入力文: 港区の倉庫

[7]

規則: $\langle NP: DMB=DMB(3), -SING: FPF-FWR(3,1,2) \rangle$
 $\leftarrow \langle NP: \dots \rangle \langle PP: \dots \rangle \langle NP: \dots \rangle:$



自動的に解消できない曖昧さが生ずる時は、意味解析を終った時点で二つ以上の意味の樹が評価される。この場合、ヤチマタは質問者との間で対話を行ない、最終的にどの樹を選ぶかを決定する。こうした例は主として係り受けの場所に関して起こる。例えば、「渋谷区にありラジエーターのある倉庫に納入している業者は？」という質問文に対し、「渋谷区にあり」という修飾句が「倉庫」に係る解釈の仕方と、「業者」に係る解釈の仕方と二通りの解析が行なわれる。この場合に質問者に対し、両方の解釈の違いを説明する為、 $\langle NP \rangle$ を作り上げていく規則のうちで、後に曖昧さが残りそうな規則には、あらかじめ対象となる修飾句と、それが係る句の候補とを、有向グラフ

拡張機能は、照会文の入力と同時に新しい名詞、動詞などを、すでにヤチマタが知っている言葉により定義する事を可能にする機能である。

6. おわりに

ヤチマタの機能に関して主に構文規則を中心に紹介を進めて来たが、最後にまとめとして、今後の問題点及び議論し残した点を列挙してみる。

ーシステム全体として

- 。レスポンス・タイム減少の為の工夫
- 。漢字仮名混り文による出力
現在、TEKTROを用いた漢字出力を実験中である。

ー照会文の機能として

- 。限定詞の取扱い
「総ての」、「少なくとも一つの」等の限定詞を導入する事を考慮中。

。推論

都道府県単位の地域統計データに関する照会において、「宅地の割合」を「宅地の総面積に対する割合」という具合に書き換える事を拡張機能を用いて行なっている。

。照会文の文型

ヤチマタはWHAT型の質問しか許していないが、例えば“Yes/No”型の質問を許すなど、より多様な質問を受け入れるような文法を開発する必要がある。

。入力文の誤まりの指摘

この事に関しては、現在タイプミスを発見する事がほぼ出来る状態にある。また、意味解析の際に発見される表記述と照会文との矛盾の指摘も可能である。

ーデータベースに関して

- 。Q-Aシステムにおける意味として
関係形式モデルとSemantic Netとどちらが有利かについての検討。
- 。上位・下位概念
名詞を集合として捉えているので、その部分集合を拡張機能により定義する事で実現されると考える。

以上三つの観点からいくつかの話題をとりあげ題目だけ並べて見たが、かなり大きな問題も含まれているので、今後とも検討を続けていきたいと思っている。

「ヤチマタ」は当センターにおいて約3年間に渡り開発されてきたプロトタイプである。開発にあたり、有力な助言を賜わった方々に深く感謝する次第である。

[参考文献]

- (1) F.B.Thompson, et.al. "REL: A Rapidly Extensible Language System", 24th ACM National Conf., 1969
- (2) M. Kay "Experiment with a Powerful Parser", 2nd International Conf. on Comput. Linguistics, 1967