

日本語質問文解析における データベースセマンティクスの利用 村木一至 市山俊治 (日本電気(株) C3Cシステム研究所)

1 はじめに

自然言語によるデータベースへの質問を受理する自然言語データベース検索システム(NLQA)の研究開発は近年活発に行なわれてきた。日本語質問文を受け入れるNLQAも、島津等^[1]による地理情報質問応答システムMSSS78や、藤崎等^[2]による地理データベース検索システムや、また^[3]が既に開発されている。他方米国においてもRendezvous(Codd)^[4]、Planes(Waltz)^[5]、Ladder(Sacardoti)^[6]などの英語フロントエンドシステムを持つシステムが開発されている。

通常NLQAの入力である質問文は、与えられたデータベースが規定するサブワールドに関する記述であり、質問文中の語の意味は小さく小さくなると思われる。また、質問文は言語的に特殊な形態を持ち、文一般を扱うのと比較し、言語解析もかなり単純化されるものと予測される。

このことからやちまた、Ladderなどは、言語の形態・構文を直接データベース構造に写像する手法による質問文解析・検索文生成を行おうとしている。またやちまたは名詞表モデルを用いて、統語情報を直接関係表(名詞表)上の演算制御と似しデータ検索を遂行しようとする。

しかし、両者は統語解析中心にその処理が規定され、比較的固い質問文しか受理できないし、その解析手法を生かすための特殊な解析システムを用意する必要もある。場合によっては、既存のデータベース上にそのらの手法を用いることを困難にする。

MSSS78, Rendezvousなどのシステムは

この欠点を補うため意味をも考慮して質問文の理解を行なおうとするものがある。しかし意味処理は強化すれば処理が複雑になり過ぎ、また逆に単純化すれば、インタラクションにおける利用者からの援助を頻繁に必要とする。つまり、現状ではデータベースへの質問文解析における意味・文脈処理の適切なモデルが欠如している。

本稿は、関係データベース質問文の意味解析用モデル--- Visé(View Semantics)モデル--- の提案とその一部を検証しつつ報告する。

Viséモデルは諸章の意味とデータベースの論理構造を対応付けるVisé構造と諸章の並みの妥当性を意味的に検定し、その意味を抽出する演算--- Unify--- からなる。本モデルはワードセマンティクスとデータベースセマンティクスを直接関係付け、統語中心質問文解析の欠点を補い、関係データベース質問文の意味解析・検索文生成の基礎を提供する。

モデルの検証には、統語解析に拡張LINGOLを手直ししたものを用い、INQ DBMS上に旅行用データベースを構築し行なった。旅行用データベースは2関係表からなる。

以下、ViséモデルとViséモデル上の構成される文脈管理モデルTRIP-NETをもとに、意味解析・文脈に沿った検索文生成の基本メカニズムを説明する。最後に検証システムJAMの会話例を示す。

2 目的

自然言語理解の鍵は意味・文脈理解にあると言われ久しい。と同時に特定タスクに限定しても意味・文脈の理解はおそろしく困難な面がある。しかし、構文解析だけに頼るだけでは理解という目標にはほど遠いばかりか、そこ不明らかりなつた事柄はその対象言語内でのみ意味を持つに過ぎない。

データベース検索質問文とは、タスク限定という曖昧な制約だけでなく、諸用論的制約、会話的制約という文解析にとつて非常に好ましい環境が与えられる。Viséモデルと、その上を定式化されるTRIP-NETは、上述の必然的制約を活用し、単純かつ容易に質問文から文脈に沿った検索文を生成することを目的とする。

又、既に数多くの構文解析用システムが開発され、他方商用関係データベースが用意された現状で、既存のシステムと共存しうる質問解析システムを開発する手段の提供を目的とする。

3 Viséモデル

3.1 Visé構造

Viséモデルは言葉の意味をデータベース中の属性集合に対応付ける記述法Visé構造と、1対のVisé構造を1つのVisé構造に写像するUnification(統合)からなる。

データベース上の質問環境とは、質問文は、データベース上の或る仮想的関係(View)を選択的に特定化する。文中にある自立語は多くの場合データベースの属性、あるいは属性値を指定する役割を担う。他方、付属語は自立語の属性指定を制御する役割を果たす。こうした語のデータベース上での役割を諸の意味と考える。Visé構造の記述はこうした意味を形式的に表現する。

定義1. Visé構造

Visé Structure = { Frames }

Frame = { Description-Triplets (DT) }

DT \equiv AX . RES . V

c.f. RES, V : set

定義1の如くVisé構造はDTの集合族として定式化される。各々のDTはAX(軸), RES(制約条件), V(値)からなる3つ組である。DTの集合はFrameと呼ばれ、諸要素の意味をデータベースの属性として記述する単位である。又、Frameは関係表中の属性を記述するのにも利用しよう。

検証に用いるデータベースは以下の2表からなる。

交通表 \equiv { 出発地(SE), 目的地(ME),
交通手段(KS), 料金(JR),
乗車時間(JT) }

名所表 \equiv { 名所名(MM), 所在地(BA),
入場料(KR), 閉門時(KAT),
閉門時(HEJ), 料金(KR) }

Frameはこれらの属性を記述する様である。Frameは複数個の軸をよつて1つの属性を記述するものもあり、KRL等のフレーム言語とは異なる。1つの属性の記述は、その属性名と属性値をよつて行なわれる。そのための一つの表中の属性記述の軸はATT軸, VAL軸を必ず持つ。例えば、交通表中の料金は次のように記述される。つまり、

JR \equiv { ATT . RES . JR , VAL . RES . V , ... }

1つの属性を特定化するということはその属性名と値を特定化することを他ならない。

Viséモデルでは、Frameのこうした記述を用いた諸の意味を記述する。例えば、「京都」という名前は本データベース環境では目的地, 出発地, 所在地, 観

光地の値と作り得る。語の曖昧さを記述するためRESが用いられる。

$$\text{京都} \equiv \left\{ \begin{array}{l} \text{ATT} . (\$SE, \$HE, \$HM, \$BA) . () \\ \text{VAL} . (\#SE, \#HE, \#HM, \#MM) . (\text{京都}) \end{array} \right\}$$

本例では、京都とは4つの属性のうち1つの値とあることを示す。ここはVAL軸のRESは冗長(ATTのRESと同内容)であるが、各々の軸に於いてその値Vの条件を独立に決定せよという効果を持ち、Frameに新しい軸の導入を容易にする。又、RES記述はインポートメントに於いて容易に他の構造を置きかえることができる。

名詞「京都」のように直接属性値を指示するもの以外、例として、助詞、形容詞、副詞などの語彙(付属語と総称)は、直接属性名を指定する意味を持つ。この種の語彙の意味を記述するため、本モデルでは8種の軸を設ける。

| 軸 | 役割 |
|------|-------|
| ATT | 属性名 |
| VAL | 属性値 |
| CASE | 格 |
| MOD | 係助詞 |
| MMD | 副詞類 |
| MOD | 形容詞類 |
| REF | 代名詞 |
| EMB | 埋め込み文 |

表1. 軸とその記述役割

表1は、軸とその役割を整理するものである。一般に付属語はATT, VAL軸とその語を規定する軸から作るFrameを記述される。例えば、副詞はATT, VAL, MMD, MOD軸集合によって記述される。この際、MMD軸以外のV値は全て空であり、MODのRESにはその副詞に係り得る形容詞範囲が記

述される。

動詞類(近い, 遠いなどの関係述語も含む)は、上述の名詞, 付属語類記述と多少異なる。即ち、動詞格関係を記述するため、Frame集合(Visé構造)を用いる。これは、関係表中の属性を記述するFrameの集合によって、動詞の支配する格を規定する。付録図1中の動詞「KAKAR」の意味構造に視られるように、各々のFrame中にATTのV値が与えられる、CASE軸の条件(RES)が書かれる。動詞類の意味的曖昧さについては後節で説明する。

3.2 Unification (統合)

Visé構造が7-1の記述を形式化したものに對し、Unificationは7-1向のSubstitutionを形式化する。即ち、構文規則を公理と做した時、Visé構造は構文範囲を述語とする変数であり、統合は論理Unificationである。とりもたず、係り受け関係固定の基本オペレーションである。

定義2. Unification (統合)

$$\text{Unify} : \{DT_{m,A}\}, \{DT_{m,B}\} \mapsto \{DT_{ec}\} \text{ s.t.}$$

$$\forall AX_{ic}. RES_{ic}. VIC \in \{DT_{ec}\}, \exists AX_{jA}. RES_{jA}. V_{jA} \in \{DT_{m,A}\}, \exists AX_{kB}. RES_{kB}. V_{kB} \in \{DT_{m,B}\} \\ \{ \{ AX_{ic} = AX_{jA} \wedge AX_{ic} = AX_{kB} \wedge (RES_{jA} \cap RES_{kB} \neq \emptyset) \rightarrow \text{JOIN}(RES_{jA}, RES_{kB}, RES_{ic}) \\ \wedge \text{MERGE}(V_{jA}, V_{kB}, VIC) \}$$

$$\forall \{ AX_{ic} \neq AX_{kB} \wedge AX_{ic} = AX_{jA} \\ \rightarrow RES_{jA} = RES_{ic} \wedge V_{jA} = VIC \}$$

$$\wedge [\{ AX_{ic} \neq AX_{jA} \wedge AX_{ic} \neq AX_{kB} \}]$$

$$\text{JOIN}(X, Y, Z) \triangleq X \cap Y = Z \text{ s.t.}$$

$$X = \emptyset \vee Y = \emptyset \text{ then } Z = Y \cup X$$

$$\text{else } Z = Y \cap X$$

$$\text{MERGE}(X, Y, Z) \text{ s.t. } Z = T \cup X$$

前述のような公理は文構造の係り受けを規定し、意味モデルは係り受けの妥当性を検証し、最後に意味構造を抽出する。Unification はこの意味妥当性判定と同時に曖昧を除去を行う。例えば付録図1中の'TOKYOKARA'という名詞句は助詞'KARA'を持つ曖昧を(CASE軸のFrom-Place, From-TimeとATT, VAL軸のSE, KAT)が名辞'TOKYO'とUnifyされる際、各々のRESを用いて解消され、NP'TOKYOKARA'には全く曖昧が残らない。

上述のことから理解されるように、RESは3つの意味を持つ。

1. 語の曖昧の記述
2. Unificationの制約
3. 語の言語的役割(他の語と係り受け関係を持つ際、他の語としてどのような意味をもつものを要求するのを記述)

節内文は動詞のまわりの意味妥当性が集り1つの意味塊を成す。これをScopeと呼ぶ。原則的には1つのVisé構造は1つのScopeを成す。

付録図1中の81に対する解析木の中では、動詞'KAKAR'が規定する属性集合をscopeと呼ぶ。本例中の動詞はこの場合2種類の曖昧を持つ。

- タ1701 Target Relationに関する
- タ1702 単-Relation中の属性(料金, 時間)に関する

2種の曖昧とは、1)複数個のVisé構造を与え、2)同一Visé構造中の曖昧な属性各々のFrameを独自に付加することにより表現される(付録図1解析木参照)。

動詞類は、他の句(統制し名詞句類)との間をUnificationされることにより、名詞句内の曖昧及び動詞類自身の

曖昧が消え去る。このUnificationは、動詞類を指定される1つのscope内にある全々の名詞句類の意味が抽出される後に行われる。Unification時、動詞類はORをprefixするVisé構造の集合を求め、他方の引数(名詞句類Frameの集合)を各々の動詞類Visé構造にUnifyし、結果としてより多くの名詞句類FrameとUnifyされるVisé構造をその単1Scope構文の意味とする。又、Unify不能を残される名詞句類Frameは、得られたVisé構造の新たな要員として追加される。

単一scope内でも「WA, GA」, 「NO」に関する特殊な処理は、名詞句類Frame集合と動詞類のUnificationの先に2行われる。

「NO」を含む名詞句類はそれが見え、且つ直後の名詞句類の意味の確定した時、直ちにUnificationを行う。これは、同格表現に対するあるいは、「NO」形容詞的句の処理に対応する。

「WA, GA」を含む句(Frame中のCMOD軸が各々CXT, NEW)が見えられた際、単一scope内名詞句類のFrameが定まるのを待つ2次の処理が行われる。「WA, GA」を含む句のFrameと後続する同一scope内の全々の名詞句類FrameがUnifyされる。即ち、「WA, GA」という助詞は提題という役割を持ち、その提題される概念と意味的に同格な概念が後続することがあるためこの処理が必要となる。付録図1中の例題81は、「JOUSHARYOUKIN WA IKURA」という句並みを持つが、語JOUSHARYOUKINとIKURAは本例文脈中同一の属性を指示する意味的に同格な概念であり、「WA」処理の起動される。

以上の議論は単一scope内での意味処理である。次の処理は本文を例として複数Scopeを持つ節内文処理について説明する。

理め込め文処理とは、前節に属げ下
表1中のEMB軸記述及びその生成が中
心となる。

理め込め文からの意味抽出とは、理
め込め文中の動詞類語と、それらに接続
(理め込め文を受け取る)する名詞E1の
scopeと考える。下標式図は理め込
め文のscopeを示す。

名詞類1*--*名詞類k*動詞類*名詞

意味抽出(構文則適用判定の際)とは
動詞類Visé構造に訂し、名詞類kから
名詞類1と1と名詞の順にUnification
を行なわれる。この際、名詞類i
($1 \leq i \leq k$)はUnifiableとなくともよい。
しかし、名詞は必ずUnifiableとあり必
要がある。Unifyするに生成するに
新しいVisé構造は、当該理め込め文の
意味を抽出したものである。理め込
め文処理とはもう1つの特別な処理を
行なう。これは、名詞に訂するFrame
を複製し、そのFrameに新しい軸EMB
を創り、そのV値と1と理め込め文の
Visé構造を与えることである。

又、意味抽出時に統合する行なうた
名詞句類i ($1 \leq i \leq k$)のFrameは、名詞
Frameと同じレベルの役割を持つものと
考え以後の処理を受ける。

さて、上述の処理中統合する行ない
残る名詞類Frameは、構文則適用時に
完全な意味制御を行なわれなければこ
とに起因する。一般に意味妥当な理
め込め文(scopeの固定とを考慮される)
を構文解訂時の発見すること容易
な行ない。よって、上述のように
設けた理め込め文要員と判断するた
めに対し、意味処理時にその誤りを訂
正する手法を用意しなくてはなら
ない。

2次節では、意味抽出するにVisé
構造から文脈に沿った残原文を生成可
るためのモデルTRIP-NETについて説明

する。

4. TRIP-NET

4.1 TRIP-NETの構造

質問文中には省略、代名詞指示等
があり、質問文中の文脈、会話文脈か
ら省略情報、代名詞指示対象の発見を
行う必要がある。

TRIP-NETは、Frame構造を基本単位
とするネットワークであり、会話文脈
中の場所移動、焦点移動の情報を持
つものである。検索文生成時の文
脈情報処理は、全2つのTRIP-NETに
よって行なわれる。

TRIP-NETは各々の節からFrameの2つ
のネットワークからなる意味網である。図1は
一連の質問Q1~Q3後に生成される
TRIP-NETである。

- Q1. 東京から京都まで何時向か
りますか。
- Q2. そこから金閣寺までバスで
いかりますか。
- Q3. 何時に開きますか。

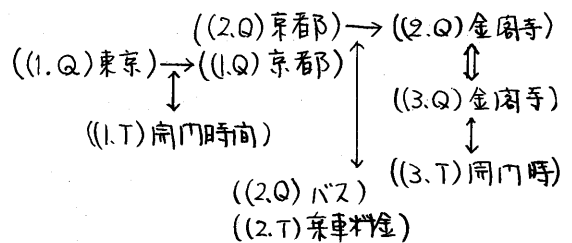


図1 Q1~Q3から生成されるTRIP-NET

図中水平矢印は場所移動を示し、矢印
左側は出発地、右側は目的地を示す。
垂直二重矢印は、JOINを示す。
現在JOINキーは、出発地、目的地、
場所とする(INQ DBMSのimplicit JOIN
機能を用いる)。垂直矢印は関連す
る属性を示す。TRIP-NET中のFrame

(図1 2, 3は簡略化, 2, 3)は質問番号と識別符号を特許化する。識別符号は, 上記にある質問文中のTarget (T) であるの Qualification (Q) を記述する。

TRIP-NET は会話の進展に従って新たな Frame が付加して成長する。例えば図1 2, 3, 質問 Q1 が行なわれる後には, 質問番号が1 である要素は行から成るが, Q3 後は図1 全体である。

4.2 文脈処理

一連の質問応答において, 前の質問を前提として省略を含む質問や, 前の質問あるものは前の質問結果に対する応答を指定する指示代名詞を含む質問がある。

本モデルでは省略・代名詞処理は全2 TRIP-NET を基に行なわれる。

代名詞処理の対象となる代名詞は, 上記(4); (5); (6); (7); (8); (9); (10); (11) である。代名詞は, 2 類に分類される。1) 場所を特定する (REF1), 2) あるいはものを指示する (REF2)。2 分類は, 代名詞意味構造 Frame 中の REF 軸の V 値を区別される。この Frame は, ATT, VAL 軸を持ち, 各々 RES 記述が行なわれる。REF1 は REF 軸値として Frame の ATT, VAL 軸に於ける RES は, BA, SE, ME, MM の曖昧さを持つと規定され, REF2 は, RES は何をも指示し得るという意味を空しくしておく。又, 指示代名詞は, REF 軸のみを持つ Frame を与える。

(指示代名詞処理は, 名詞句類処理が終了し, 各々の句の意味が整った時に起動される。

REF1 処理

名詞句類の REF 軸値が REF1 であるものを発見した時, TRIP-NET 上の最も

大なる質問番号を持つ全2の Frame 集合 (Current-context: CC と呼ぶ) 中から Unifiable な Frame を集めて, 上記をこの句の意味とする。この際, CC から TRIP-NET 上の水平文印を含む時は, この句の VAL 軸の V 値を, CC 中の目的地の VAL 軸中の V 値を強制的に入れ替える。更に, この値か? (前質問文の Target である) であるか, REF1 の値を持つ Frame に EMB 軸を創り, CC を埋める。

REF2 処理

Unifiable な CC 部分集合を V 値 REF2 を含む句の意味とする。

両処理中, CC の中の目的地, 出発地, 名所名と規定された Frame があれば特殊な処理を行なう。TRIP-NET 中から CC を見つけ出す際, 目的地, 出発地, 名所名と規定された Frame があれば, この ATT, VAL 軸の RES に, ME, SE, MM を追加し, ATT 軸の V 値を空くする。こうすることによって, 2 表上の JOIN キー情報が CC 中に埋め込まれる。

この2, 通常 REF を持つ名詞句類は上述の処理によって曖昧さを持つ。前3.2 節では, 但々の名詞句類に曖昧さがない場合について説明したが, 1つの名詞句類は複数個の Frame を持つ曖昧な場合 (通常この特殊ケースは, REF 軸をもつ名詞句類だけに限られる), 動詞との Unification は, 全2の意味を順次調べ, Scope 全体として最もよく Unify される意味をもつ Frame をこの句の意味とする。

以上説明した手法は, 代名詞処理の意味抽出時に TRIP-NET を参照しダイナミックに行なわれるとこそこの有効性がある。しかし, この生成される曖昧さは動詞類の Visé 構造に

よって解消される。しかし日本語領内文に限らず動詞類が省略されることがある。そのような場合には、CCからその情報を埋める必要がある。このCC利用は、質問文の意味を抽出した後に行われる。我々のモデルでは、下の仮定を用いる。

Proposition

質問文では通常主文以外の動詞類は省略される

上記の仮定はかなり自然なものである。日本語に限らず、省略とは、何らかの場が設定された後に生起するものである。従属文中の動詞類が省略されることは場が設定されたものであることを意味するからである。

そこで、省略の理め込みを行うためには、動詞類省略時に残る曖昧さを何らかの手段で解消する必要がある。我々は、この曖昧さを有くために、基底関係の Visé 構造を用いる。これは全く動詞と同じ構造を持つ。各々の Frame による各属性が記述される。前節、本節に既に述べたように、質問文の意味は、Visé 構造として表現され、ある場合には各々の Frame は今般曖昧さを残して置く。それに対し、基底関係の各々は ATT 軸の V 値は確定して置く。

単 1 scope 文では、その scope 内の全ての Frame に対し、より多く unifiable な基底構造（これは動詞類とその scope 内他の Frame との演算と同じ）を用いて、個々の Frame の ATT 軸 V 値を定める。即ち、単 1 scope 内の動詞類の 2 つの曖昧さ（文通表 Visé 構造、名所表 Visé 構造）をもつと同じ意味をもつ。

上記演算は理め込み文に対して行う必要はない。本演算後には選ばれた基底構造には、その scope 内の意味が含まれることに留意する。

4.3 TRIP-NET, 検索文生成

動詞類省略に対する基底関係 Visé 構造を用いた処理の後得られる Visé 構造から TRIP-NET を生成し、その CC から検索文生成を行なう。

TRIP-NET の生成は以下の手順で行なう。上記の手続で得られる基底構造を PQ (Partial Query) とし、新しい CC を CQ (Complete Query) とする。

1. 現状に於いて、CC が空ならば、PQ を CQ とする
2. CC の条件 (Q と識別される Frame) がなければ PQ を CQ とする。
3. PQ 中の Frame の VAL 軸に具体的値 (? 以外) の値を持つ (条件情報を持つ) ものがなければ、

i) PQ のターゲット情報 (各々の Frame の CMOD 軸が CXT or NEW の VAL の V 値が空であるか、それ以外では、VAL 軸 V 値が ? の Frame) の 1 部が CC に含まれるか、あるいは、その意味的キーが CC に含まれるか、PQ と CC の条件部を合わせて CQ とする。

ii) i) 以外ならば、PQ を CQ とする。

4. CC と PQ の各々の条件部、条件情報を共通属性があれば、PQ を CQ とする。

5. PQ のターゲット情報がなければ CC のターゲット部と、CC, PQ の条件部、条件情報をマージしたものを合わせて CQ とする。

6. 上記以外には PQ のターゲット情報と、CC, PQ 各々の条件情報、条件部を Merge したものを合わせて CQ とする。

上記手続は主文に対し2のたあある。理め込み文(PQ中のEMBをもつFrameの、EMB軸V通)は、全2が条件部として扱われる。この際PQ中では、主文のVisé構造とEMBの通のVisé構造が同一関係系Visé構造に統合しうる場合は、統合したものを上述PQとし、それ以外はEMB内の上記16の処理が行われる。

この結果検索文は複数理め込み関係として生成されることがあるが、この検索文構文はSQL^{[5][8]}とほぼ同じ構造として与えられる。

22, CQはSQL表現を生成する基となるが、CQの条件部は、Qualification (SQLのwhere句)、ターゲット部はTarget (SQLのselect句)に写像され、理め込みCQは、理め込みselect句、where句に写像される。

最後にCQは、TRIP-NET中に追加され、この際主select句以外は全てTargetと識別される。

5 会話例

現在Viséモデル、TRIP-NETは1部インプリメント工山稼働している。付録図2はこの会話例を示す。本モデルインプリメントでは、既存のDBMSに手を入れないで日本語フロントエンドを付加することを目的としている。そのため各種の名辞を言語処理部が辞書に登録(コア上に持つ)している。しかし、何百万件の名辞を辞書に登録することは無理であり、将来的には形態素処理時にデータベース内で探索し名辞の同程を行うことが必要となる。

現状では、全2の名辞を辞書登録しないが、言語解析時に未定義語として登録されている名辞を発見し、文構造からその未定義語名辞の意味を予測することにより上記問題に対処している。付録図中2. Bはこの例を示

す。現在のインプリメントは、単文と、代名詞による複雑な構文に処理できるように2000語、未定義語を含む5文(句数7位)の解析は2~5秒、未定義語を1語含む文は5~7秒の処理速度を得ている。なお、インプリメントは、1MIPS機上のLispインタープリターを用いている。

6 まとめ

ViséモデルとTRIP-NETは意味処理文脈処理の1つの形式化手法であり、特にViséモデルは非常に簡潔な表現法とこの上の推論を完全に形式化している。

本モデルの特徴は、その簡明さと同時に有効なことがある。モデル検証を目的とした開発工山JAM (Japanese Access Method)は、かなり高速にその箇文解析・検索文生成を行なうことを実証している。現在、JAM下では次の機能が実現されている。

1. 解析から視た機能
 - スパーリングエラー訂正
 - 未定義語処理
 - 曖昧語処理
 - マジック処理
2. 会話管理から視た機能
 - パラフレーズ
 - yes, no, what 質問タイプ同定
 - 代名詞(1部)の処理
 - ターゲット・条件省略処理
3. 推論機能
 - コスト最小・最小道程出力
 - ?制約下での最適道程出力(開発中)
4. 検索・出力機能
 - 出力件数プロット
 - 動的表結合(implicit Join)

Viséモデルは意味処理の形式化手段である。従来は構文情報とデータセマンティックスを関係付けるモデル

が提案しているが、言語に依存し、
 又、データベースに依存することは必
 然であった。Viséモデルはこうした
 タスク依存性をかなり小さくするとい
 う期待がたれ、将来多言語、デー
 タベース上を検証する予定である。

最後に本研究の機会を与えられた当
 研究所コンピュータシステム研究部三
 上部長、箱崎課長に感謝するとともに、
 データベース検索用言語FOL^[8]を使用
 させた頂いた日吉代に感謝します。

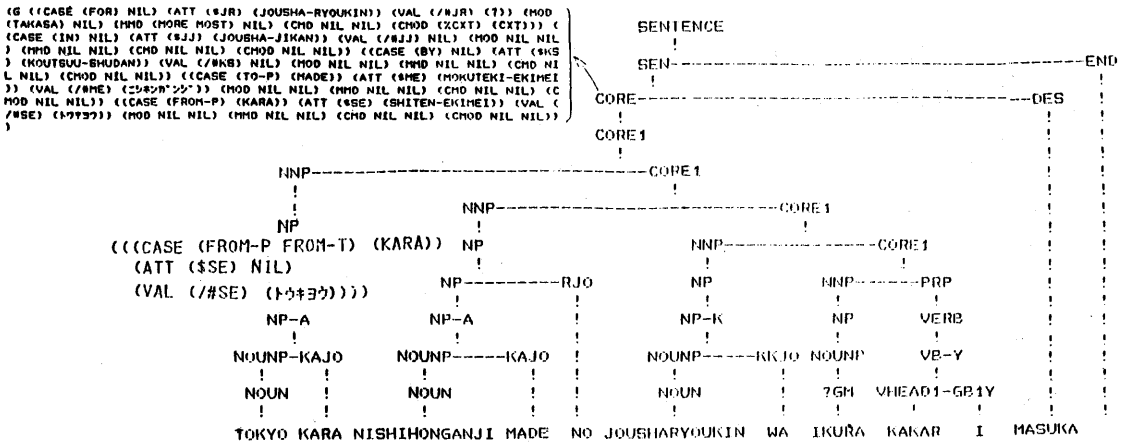
付録 図1 意味抽出例

参考文献

- [1] 島津明 「日本語の構文意味解析-構句依存性」
MSSS78 への試み, 信学会 AL78-94
- [2] Waltz D.L. 「An English Language Question Answering
System for a Large Relational DataBase」
,CACM Vol. 21, No. 7 1978
- [3] COBB E.F. 「DATABASES: Improving Usability and
Responsiveness」 Proc. '78 International
Conf. on Databases
- [4] 田中稔穂, 他 「拡張 LINGOL マニュアル」電産研, 1978
- [5] Chamberlin, D.D. 他 「SEQUEL: A structured English Query Lan-
guage」, PROC. ACM SIGARIBET, 1974
- [6] 藤崎, 他 「データベースシステム「マーマ」とその問い合わせモデル」
信学誌. 卅. Vol. 20, No. 1.
- [7] Sawadai, 「A LADDER USER'S GUIDE」, SRI, TN. 163, 1978
- [8] 日吉 他 「日本語用文庫データベース検索システム」
信学会, 情報システム部7連合大会予稿. 81

シツエン ナ ナニ ?

* TOKYO KARA NISHIHONGANJI MADENO JOUSHARYOUKIN WA IKURAKAKARIMASUKA?



```

(TOKYO NOUN ((#SE) 0.))
((ATT (#SE) ()))
(KARA KAJO (NIL 0.))
((CASE (FROM-P FROM-T) (KARA))
 (ATT (#SE #KAJ) ()))

(KAKARI VHEAD1 (NIL 0.))
(OR ((CASE (FROM-P) ()))
 (ATT (#SE) (SHITEN-EKIMEI))
 (VAL (#SE) (I)))
(CASE (TO-P) (I))
(ATT (#HE) (MOKUTEKI-EKIMEI))
(VAL (#HE) (I))
(CASE (BY) (I))
(ATT (#KS) (KOUTSUU-SHUDAN))
(VAL (#KS) (I))
(CASE (IN) (I))
(ATT (#JJ) (JOSHUA-JIKAN))
(VAL (#JJ) (I))
(CASE (FOR) (I))
(ATT (#JR) (JOSHUA-RYOUKIN))
(VAL (#JR) (I)))

```

((SELECT DATA ((JOSHUA-RYOUKIN) (SHITEN-EKIMEI) (MOKUTEKI-EKIMEI) (KOUTSUU-SHUDAN))) (WHERE (SHITEN-EKIMEI EQ トウキョウ AND MOKUTEKI-EKIMEI EQ ニシホウガンジ)))

シツエン ナ ナニ ?
 * TOKYO KARA NISHIHONGANJI MADE IKURA KAKARI MASUKA?

シツエン ナ ナニ ?
 * TOKYOKARA NISHIHONGANJI MADENO JOUSHARYOUKIN WA?

付録 図2 会話例

a. 代名詞処理会話例

ニョウゴ モトノ？
 A ケイシキ ナンゴ
 B ニホゴ
 ? A
 シツメン ノ ナニ ?
 \$ TOKYO KARA KYOTO MADE IKURA KAKARIMASUKA?
 アタリ シツメン
 シテン イキメイ カ トウキヨウ ナ
 カツ モツキ イキメイ カ キヨウト ナ 7L
 ヨウ
 ショウシ リョウキン ト
 シテン イキメイ ト
 モツキ イキメイ ト
 コウツウ シュタン
 ヲ オシエテクダサイ
 ナスガ? カクニンシテ クダサイ
 A ナイ ソウチス
 B イイ チカイマス
 C モウ ナマス
 ? A
 コゴ
 2. ケン カイトウ シマシク
 ヒョウシ シチホシイ ケンズウ ヲ ケーイン スルカ マリ
 シタ A-D ヨリ イランチ ケーイン シタクダサイ
 A センフ ヒョウシ スル
 B ショウケンチ ノ アタイ ヲ カエテ シツメン ヲシタイ
 C ハツリ シツメン ヲ シタイ
 D シュウリョウ シタイ
 ? A
 シテン-イキメイ ナゴヤ
 モツキ-イキメイ キヨウト
 コウツウ-シュタン シンカンセン
 ショウシテ-リョウキン 9500.
 シテン-イキメイ トウキヨウ
 モツキ-イキメイ キヨウト
 コウツウ-シュタン ナス
 ショウシテ-リョウキン 7100.
 ツキニ トウシマスカ?
 A ニホゴ ナ シツメン ヲ シタイ
 B ケイシキナンゴ ナ シツメン ヲ シタイ
 C シュウリョウ シタイ
 ? A
 シツメン ノ ナニ ?
 \$ NAGOYA KARAWA?
 アタリ シツメン
 シテン イキメイ カ ナゴヤ ナ
 カツ モツキ イキメイ カ キヨウト ナ 7L
 ヨウ
 ショウシ リョウキン ト
 シテン イキメイ ト
 モツキ イキメイ ト
 コウツウ シュタン
 ヲ オシエテクダサイ
 ナスガ? カクニンシテ クダサイ
 A ナイ ソウチス
 B イイ チカイマス
 C モウ ナマス
 ? A

コゴ
 2. ケン カイトウ シマシク
 ヒョウシ シチホシイ ケンズウ ヲ ケーイン スルカ マリ
 シタ A-D ヨリ イランチ ケーイン シタクダサイ
 A センフ ヒョウシ スル
 B ショウケンチ ノ アタイ ヲ カエテ シツメン ヲシタイ
 C ハツリ シツメン ヲ シタイ
 D シュウリョウ シタイ
 ? A
 シテン-イキメイ ナゴヤ
 モツキ-イキメイ キヨウト
 コウツウ-シュタン シンカンセン
 ショウシテ-リョウキン 3600.
 シテン-イキメイ ナゴヤ
 モツキ-イキメイ キヨウト
 コウツウ-シュタン ナス
 ショウシテ-リョウキン 1600.
 ツキニ トウシマスカ?
 A ニホゴ ナ シツメン ヲ シタイ
 B ケイシキナンゴ ナ シツメン ヲ シタイ
 C シュウリョウ シタイ
 ? A
 シツメン ノ ナニ ?
 \$ KOKOKARA SANZENIN EWA NANIDE IKIMASUKA?
 アタリ シツメン
 シテン イキメイ カ キヨウト ナ
 カツ モツキ イキメイ カ ワンセン ナ 7L
 ヨウ
 コウツウ シュタン ト
 シテン イキメイ ト
 モツキ イキメイ
 ヲ オシエテクダサイ
 ナスガ? カクニンシテ クダサイ
 A ナイ ソウチス
 B イイ チカイマス
 C モウ ナマス
 ? A
 コゴ
 1 ケン シカ アリマセン
 シテン-イキメイ キヨウト
 リカエ-イキメイ ナンチ
 モツキ-イキメイ ワンセン
 コウツウ-シュタン-1 ナス
 コウツウ-シュタン-2 トホ
 ツキニ トウシマスカ?
 A ニホゴ ナ シツメン ヲ シタイ
 B ケイシキナンゴ ナ シツメン ヲ シタイ
 C シュウリョウ シタイ
 ? A
 シツメン ノ ナニ ?
 \$ SOKOJA NANJI NI AKIMASUKA?
 アタリ シツメン
 メイショ メイ カ ワンセン ナ 7L
 ヨウ
 カイメン シゴウト
 メイショ メイ
 ヲ オシエテクダサイ
 ナスガ? カクニンシテ クダサイ
 A ナイ ソウチス
 B イイ チカイマス
 C モウ ナマス
 ? A
 コゴ
 1 ケン シカ アリマセン
 メイショ-メイ ワンセン
 カイメンシゴウト 8:30
 ツキニ トウシマスカ?
 A ニホゴ ナ シツメン ヲ シタイ
 B ケイシキナンゴ ナ シツメン ヲ シタイ
 C シュウリョウ シタイ
 ? C

b. 未定義語処理例

\$ NAGASAKIKARA.
 --- NAGASAKI ---> NEW WORD ? (Y/N)
 *N
 INPUT CORRECT NAME, PLEASE
 *OUSAKA
 --- OUSAKA ---> OSAKA ? (Y/N)
 *Y