

キーワード主体の意味解釈とタスクの階層構造に基づく質問応答モデル

浅見 徹 谷戸 文廣 樽松 明

(K D D 研 究 所)

§ 1. まえがき

筆者らは、国際電話の受付、およびその接続時における会話をモデルに質問応答システムの作成を行っている。自然言語処理の分野では拡張LINGOL⁽¹⁾やATN⁽²⁾に代表される統語規則に重点を置いた構文解析法の発達により、英語や日本語の構造が次第に明らかになりつつあるが、本論文で扱う会話のような談話理解レベルでは、単一文の理解における統語規則のような明確な指針がなく、統語・意味・文脈を統一した方法論については、未解決の問題が数多く残されている。

本論文では国際電話交換業務に特徴的に現われるキーワードを用い、キーワード間の意味的照応関係と簡単な文脈処理の導入により、適切な応答を作成するシステムの検討を試みている。このシステムでは、少数のキーワードが入力文の意味を決定する上で大きな役割を果たしていることを示し、パズ過程ですべての単語を平等に扱うのではなく、タスクに依存した単語を中心に構文解析や意味解析の方向を決定する方法の利点を明記している。対象言語は英語である。

§ 2. 対象タスクの談話構造

電話交換業務の内容は多岐に渡り、図1にその概要を示した。ここで、業務内容の大半を占める各コールの受付・接続時における対話内容には、次に示す著しい言語上の特徴がある。(1) キーワード主体の情報交換である(例えば、ステーション・コール、エリア・コード、電話番号

- (1) 各コールの受付・接続(ステーション、パーソナル、コレクト、クレジット・カード)
- (2) 案内(電話番号、接続時間、相手国時刻、通話時間、通話料金など)
- (3) 通話の取り消し
- (4) 通話の再接続(途中瞬断時、エコー過多時等)
- (5) 故障診断(相手電話器の)
- (6) 他の番号による再接続
- (7) hang-up時の呼び返し

図1. 国際電話交換業務の概要

Op1: Operator. May I help you?

Ito1: I want to place a station call to Tokyo, Japan. The area code is 03. and the number is 147-6402.

Op2: Area Code 03, and the number 147-6402?

Ito2: That's right.

Op3: May I have your number?

Ito3: 1-683-2241.

Op4: 1-683-2241?

Ito4: Right. Could you please let me know the time and charges after the call?

Op5: Certainly, sir. Hold the line, please. I'll put you through.

Ito5: Thank you.

図2. ステーションコール受付例

(“国際電話の英会話”より⁽³⁾)
 など)。(2) 交換手への入力には口語であるため省略が多い。(3) 人名地名等の固有名詞を除くと、用語数は少なく、数百語程度で十分カバーできる。(4) 交換手主導型の対話の占める割合が多い。(5) 交換手は入力データの確認のため、問い返しができる。

図2に代表的ステーションコールの受付例をKDD発行“国際電話の英会話”⁽³⁾から転記した。ここで、下線部はキーワード、省略文はOp2、Ito3、Op4、Ito4に、問い返し文はOp2、Op4に現われている。また、Op3およびOp5は交換手主導型の対話部分である。この例題から明らかかなように、談話内の文は省略のため文として不完全なものが多く、文を理解するためには省略語の推定、指示詞の解釈が重要な要因となる。また、倒置法も頻繁に使用されるため、書き言葉に比較して、統語上の自由度は非常に大きくなる。この傾向は日本語において特に顕著であり、例えば図2のIto4に対応するものとして、“通話が終わったら知らせてください、分数と料金。”といった入力も許される。この例では、日本語の動詞が文末に位置するという書き言葉における一般的規則も破られている。従って、統語規則に重点を置いた解析では、省略や倒置文をも含めて処理しようとする、規則数・あいまいさの増加を生じ、ディバッグが困難になる。

省略語に関しては久野⁽⁴⁾の、指示詞と省略語に関しては長尾⁽⁵⁾の検討があるが、単語の意味的かかり受けに基礎を置いた方が、省略や指示詞を一般的に処理できることが予想され、横尾の文脈処理モデル⁽⁶⁾などが提案されてきた。また、Groszはfocus spaceの導入により指示詞や省略語の処理を行い⁽⁷⁾、これはReichmanによってContext spaceとして拡張が試みられている。⁽⁸⁾

一般に、構文解析レベルで意味・文脈処理をも実行するのは、統語処理のみの場合に比較して処理量が増大するため好まれないが、これは入力文が文字列のような明確な構文

上の単位を持っている場合に言えることであり、音声入力の場合のように入力にあいまいさがあるときは、文脈上・意味上・構文上の制約により候補となる単語を初期の段階で絞ってから構文解析を実行する方が、全処理量を軽減できる。

そこで、将来の意味処理に基づくシステムへの移行の前段階として、電話交換に特徴的に現われるキーワードの意味上の照応と、部分的構文木に基づくシステムの検討を試みることにした。この談話理解システムでは以下に示す部分否定と再パーズの問題の解決を試みている。

2.1 部分否定の問題

質問応答システムとユーザ間の会話の自由度を上げ、ユーザからの複数の質問あるいは情報の提供を受付けられるようにし、自然な会話であるように指向した場合、次に示す部分否定の問題を解決しなければならぬ。図2のIto1に対する交換手の問い返しOp2の代りに、図3の応答をした場合、Ito-1のような肯定入力であれば、次の質問へ進むことができるが、Ito-2～5の否定入力に対しては、既得情報と入力情報との参照を行う必要がある。

Ito-3は部分否定の例であり、否定項目は番号だけである。従って、opの次の問い返しは、“The number is 747-6402?”が適切であり、都市名等を再度問い返すのは煩わしい。これは、Ito-4の例に示したように、部分否定の後に新情報が

```
Op: The station call to Tokyo,
      Japan, the area code 03, and
      the number 147-6402?
Ito-1: That's right.
Ito-2: No.
Ito-3: No, the number is 747-6402.
Ito-4: No, the number is 747-6402,
      and my number 1-683-2241.
Ito-5: The number is 747-6402.
```

図3. 問い返しへの部分否定

付加される場合に、問い返しの情報を必要最小限に抑えるためにも必要である。部分否定は、“No”のような否定の副詞を伴わずに行うことも可能である。Ito-5はその例であり、問い返し文中の番号情報が訂正されている。もちろん、その後には発信者の番号等の新情報が付加されることもある。

2. 2 再パーズの必要性

自然言語は、そのあいまいさのために、話者と受話者間の会話の同期がとれなくなる場合がある。図4のop1では局番を42422とみなしているが、“The number”の示す電話番号は、(1) エリア・コードと局番付き、(2) エリア・コードが無く、局番付き、(3) エリア・コードも局番もないものという計3種の電話番号に対して使用可能であるため、Ito2はエリア・コード=42422と解釈してくれたものと誤解している。Itonが入力された時点で、会話の同期を回復するため、システムは以下の処理をする必要がある。

- 1) 42422を受信者のエリア・コードと仮定する。
- 2) 既存の42422-0311とエリア・コードの組が、電話番号の所定の条件を満たしているかをチェックする。
- 3) 2)は失敗したため、次の仮説を立てる。<1> 1)が誤りである。
<2> 42422-0311の解析時の仮定(2)が誤りである。
- 4) 仮説<2>の実証のため、解析仮定(1)により、42422-0311を再パーズする。その結果、area-code=42422, number=0311を得る。
- 5) 入力42422は再パーズ後のエリア・コード部と一致するため、<2>が<1>に優先され、opnの応答が作成される。

Ito1: The number is 42422-0311.
 Op1: The number is 42422-0311?
 Ito2: Yes, and my number is

 Opn-1: What's the area code in Japan?
 Iton: 42422 is the area code!
 Opn: Aha! The area code is 42422
 and the number is 0311?

図4. 再パーズの一例

従って、入力文を内部形式に変換して記憶する場合、解釈にあいまいさのあるとされた解釈の枠組、棄却できていない他の解釈の枠組を付加して記憶しておき、後での再パーズに備える必要がある。ここでは、Ito2を解釈した時点で、この作業をしておかなければならない。

§3. 談話モデル

談話モデルは、ステーションコールの受付にタスクを限定し、このタスクを実行するのに必要最小限の数のキーワードを選び、キーワード間の意味的照応、フォーカス、知識ベースを用いて発話内容を理解することを目標にする。入力文の省略も許し、例えば図2のIto1の入力として“Station, Tokyo, Japan, area code 03, the number 147-6402”といった文も受付られるものとする。

このモデルでは単語の意味はタスクを与えた時に定まるとの立場から、タスクを図5で示した階層構造に分割する。⁽⁹⁾ここで各ノードは(サブ)タスクを、アーケはdisjoint elementを示す。ここでノードNからdeで分割されたサブタスクNiは互いに同時生起できないことを、eで分割されたサブタスクNjはNから分割した他のサブタスクと同時生起できることを示している。各ノードは、lexicon、procedure、parent、children slotを含む。lexiconは2つのfacetからなり、definition facetには、図6に示すように、そのノード以下にあるノードに共通かつ固有の意味に用いられる単語と

その意味記述を入れ、alllistには、そのノード以下にあるノードでdefinitionされている単語のリストを入れる。例えば、stationは、station-callノードで、personalはpersonal-callノードで、call, number等はcall-handlingノードでdefinitionされている。また、call-handlingのalllistにはpersonal, call, number等は含まれているが、personalはstation-callのalllistには含まれていない。また、procedure slotはそのノード以下にあるタスクに共通する手続きをproduction facetとして収容している。例えば、personal-callには、受信者と発信者の名前を決定する手続きを、credit-card-callにはクレジットカード番号決定手続きを、call-handlingには、発信者および受信者の電話番号を決定する手続きが埋め込まれている。従ってこの構造では、station-callからtaskまでのpath上にあるノードにstation-callを処理するために必要な単語および手続きが定義されているため、ステーションコール受付時にはstation-callノードから上にたどれば必要な手続きと単語の定義が分る。

また、parent slotにはこのノードの上位ノードがnpであり、その接続情報をfacet名deまたはeで記入してある。逆に、children slotには、このノード直下にあるdeノードおよびeノードのリストを取め、defaultノードはfacet名default下に記入してある。

図7に示したように、図5のタスク構造スペースという知識ベース以外に、KスペースとDスペースという知識ベースを含む。Kスペースはその時点までの会話で得られた情報をいわゆるフレーム形式で記

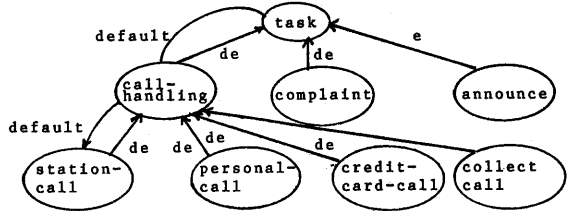


図5. タスク階層 (タスクスペース)

```
(ni (procedure (production (p1) (p2)...(pn)))
  (lexicon (definition ((wi-1 (semantics)..
                          (semantics)))
                      .....
                      ((wi-n (semantics)..
                          (semantics))))))
  (parent (de (np)))
  (children (de (nc-d1...nc-dn))
            (e (nc-e1...nc-en))
            (default (nd))))
pi = procedure, wi-j = word, wi-sj = word used in
subtasks, np = parent node, nc-dj = desjont child
node, nc-ej = non-disjont child node,
nd = default child node
```

図6. 各taskノードの情報構造

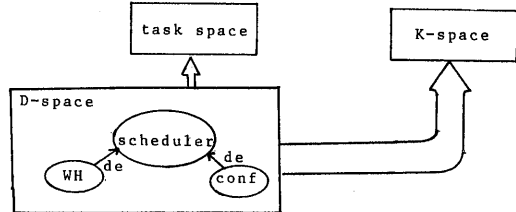


図7. 各スペース間の階層

```
(call1 (parent (e ((station-call) (c1 . cs1) (I1))))
  (sender (number ((747-6452) (c2 . cs2) (I2)))
  (receiver (area-code ((03) (c3 . cs3) (I3))))))
```

ci,csi,Iiはそれぞれ入力解釈時の環境、未選択環境、入力文を示す。

図8. 既得情報の構造

述してある(図8)。ただし、情報フレームの値は、((得られた情報)(入力解釈環境・未選択解釈環境)(入力文))という3種のエレメントからなるリストとして得られる。また、Dスペースは全処理のスケジューラの役割を担っており、WH, conf, schedulerの3フレームからなる。各ノードの記述は、lexicon slotのない点を除いて図6のものと同形式である。schedulerフレームはスケジューラの役割を果し、WHフレーム・confフレームの起動、入力文のページング、部分否定・再スペースの処理等を行なう。WHフレームが駆動されるとschedulerフレームが

ら渡された質問項目を VH-question に変換し、ユーザに問うと同じに、ユーザからの入力を解釈する VHq フレームを作成する。また、Conf フレームが駆動されると scheduler から渡された確認項目を Yes-no-question に変換し、ユーザに確認を求めるとともに、ユーザからの入力を解釈する confirm フレームを作成する。

VHq および confirm フレームは、K スペース上に各質問が行なわれるたびに新しく作成され、次の質問時に消去されるため、一種の short-term メモリとして使用できる。

D スペースから、K スペースおよびタスク構造スペースへの arc は、D スペースが後者の情報を使用できることを意味する。

§ 4. 動作説明

このシステムでは、タスクを定めなければ入力文を解釈できない。タスク決定手続は scheduler により、次のように実行される。

- 1) ノード名 $n \leftarrow \text{task}$ とし、入力文から n の alllist にある単語のみからなるリストを作り lc とし、go to 2。
- 2) n に children ノードがなければ、 n が求めるタスクであり、そうでなければ、go to 3。
- 3) (null lc) ならば、 $n \leftarrow$ (n の default ノード) とし、go to 2、そうでなければ、go to 4。
- 4) $lc \leftarrow$ (lc から n の definition にある単語を除去) し、go to 5。
- 5) (null lc) ならば、 $n \leftarrow$ (n の default ノード) とし、go to 2、そうでなければ、go to 6。
- 6) $nl \leftarrow$ (n の children であり、その alllist が lc を含むノードのリスト) し、go to 7。
- 7) (null nl) であれば、 $n \leftarrow$ (n の parent ノード) とし、go to 7。

```
(confirm (parent (e (station-call)))
  (receiver (country (Japan))
    (city (tokyo))
    (area-code (03))
    (number (147-6402))))
```

図 9. 解釈フレーム (図 3 の Op のもの)

S → KP ... KP

KP → KPi

S は文、KPi は 1 つ以上のキーワードからなるフレーズ。

図 10. 構文規則

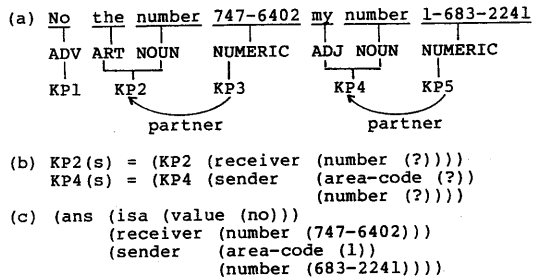


図 11. パーズ過程

8) $n \leftarrow (car nl)$, $nl \leftarrow (cdr nl)$ とし、go to 2。

例えば、図 2 の Op1 以前に入力文はないので、全て、default arc が選択され、タスクを station call とした上で、Itol の解釈フレームを定めている。Itol でキーワード station call が入った後は、arc は default 値でなく instance として固定される。ただし、現在のシステムではノードとして、task, call-handling, station-call しかタスク空間にないため、この処理はあまり意味を持たないが、ここでは、タスク空間のノードをトレースすることにより、フォーカスの変化に追従できることを示した。

図 3 を例にパーズ過程を説明する。Op を発した時点での解釈環境 Ci は図 9 で示したように与えられる。構文規則のトップレベルは、図 10 に示すように、文を 1 つ以上のキーワードフレーズ KP の集合としてとらえ、各 KP には順序関係はない。KPi は No などの副詞、名詞句、前置詞句に相当し、各 KPi を作成する構文規則は主に KPi の名詞の semantics 部に埋め込まれている。ここに Ito-4 が入力さ

れると入力文の中から図11(a)に示す単語が抽出される(数字列は、allistにないがschedulerが付加する)。My, the等の指示詞の処理は、人称代名詞の所有格の場合は、station-callタスク作成時に指示対象が定まるが、theに関しては、その句の前の句を逐次たどり、slotにsenderかreceiverを持つ句を捜し、最初に見つけた値を指示対象としている。この例では見つからないため、Ciのslot receiverを指示対象としている。従って、この段階の構文解析により、KP1には否定文であることを示す意味情報KP1(s)が、KP2にはreceiver slotのnumberに関するマーカであることを示す意味情報KP2(s)が、KP4にはsender slotの電話番号(エリアコードがあるか否か未定)に関するマーカであることを示すKP4(s)が入る。さて、KP3, KP5は数詞であるため、その意味処理のため文中またはCi中から意味マーカを選択する。ステーションコールの場合、数詞にはarea-code, number facetを持つものを選ぶ。従って、KP3, KP5の相手には、Ci, KP2, KP4が候補になる。その候補選択手順は、(1)候補を1つしか持たない数詞の相手は他の数詞の相手候補リストから除く。以下(2)~(3)の処理の後に(1)を実行する。(2)相手候補リストの中で、各数詞の前後最隣接のマーカをarea-code, numberに関して選び、候補を高々4つに限定する。(3)候補が1つに絞りきれない最後尾の数詞に関しては、その後のマーカを指しているときはそれを相手とし、後にマーカがなければ、種類を問わず前方最隣接のマーカを相手とする。この手続きで定められたマーカ情報を利用して数詞パーザを駆動する。KP2(s)の情報によりKP3はエ

リアコードの無い電話番号として処理され、数詞そのものの持っている意味上のあいまいさはパーザ過程へ影響を及ぼさない(図11(b))。以上から、Ito-4の意味構造として図11(c)が抽出され、Ciとの比較からreceiverのnumberが否定され、送信者電話番号が付加されたことが分る。2.2節の場合も図8の入力解釈環境と未選択解釈環境情報を利用すれば、同様にして処理可能である。

§ 5. まとめ

入力文のキーワード集合により、解釈環境を選んで語の意味を限定した後にキーワード間の意味的照応によって入力文の解釈を行う質問応答システムを検討した。解釈環境を定めることにより、複数の意味を持つ語の意味が限定され、単純な構文規則で入力文の理解が可能になることをステーションコールを例に述べてみた。

謝辞

日頃御指導を戴く当研究所の鍋治所長、寺村副所長、中井次長、並びに端末装置および第一特別研究室の各位に感謝致します。

参考文献

- (1) 田中穂積「計算機による自然言語の意味処理に関する研究」、電総研研究報告第797号、昭55；
- (2) Woods, W.A. "Transition network grammars for natural language analysis," CACM, Vol. 13 (1970), No. 10, pp. 571-608；
- (3) KDD広報室「国際電話の英会話」、昭55；
- (4) 久野「談話の文法」、大修館、1978
- (5) 長尾、辻井、田中「意味および文脈情報を用いた日本語文の解析—文脈を考慮した処理」、情報処理 Vol. 17, No. 1 (1976), pp. 19-28；
- (6) 横尾「日本語理解システムにおける文脈処理のモデム」、情報処理学会人工知能と対話技法研究会 19-2 (1981)；
- (7) Grosz, B.J., "Discourse knowledge," in D.E. Walker (Ed), Understanding Spoken Language, North-Holland, 1978, pp. 229-344.；
- (8) Reichman, R. "Conversational Coherency", Cognitive Science, Vol. 2, No. 3, 1978, pp. 283-327；
- (9) Hendrix, G.G. "Encoding Knowledge In Partitioned Networks," In N.V. Findler (Ed.), Associative Networks, Representation and Use of Knowledge by Computers, Academic Press, 1979, pp. 51-92.