

名刺文意味解析システムにおける 推論制御機構

* 石川 孝

丸一 威雄 沼岡 千里 所 真理雄

* べんてる(株)中央研究所

慶応義塾大学理工学部電気工学科

文書の意味の認識においては自然言語におけるあいまい性のある文法規則、意味規則による推論が必要となることから、解釈として最も適した解を効率よく求めるための推論制御機構が大きな研究課題となっている。文書の構成要素(単語・句・文)の意味を推論する問題は、各構成要素の最適な意味を選択することであるから、分類型の探索問題(単に分類問題と呼ぶ)として一般化できる。そこで本論文ではまず分類問題において最適推論を行う推論モデルを提案し、その推論モデルに基づく推論システムの実現法について考察する。つぎに具体例として、名刺の文字列を氏名・所属などの意味項目に分類する名刺文意味解析システムにおける推論制御機構の実装および評価結果について報告する。

Inference Control in Semantic Analysis System for Visiting-card Understanding

Takashi Ishikawa*, Takeo Maruichi**, Chisato Numaoka** and Mario Tokoro**

* Central Research Laboratory, PENTEL Co., Ltd.
Yoshicho 4-1-8, Soka 340, JAPAN

** Department of Electrical Engineering, KEIO University

In document understanding we need inference with ambiguous syntactic/semantic rules of natural language. In this circumstance it would be still important to construct inference control mechanism to find most suitable solution efficiently. We propose here a inference model of classification problem to find most suitable solution. This paper first describes the inference model and inference system based on this model. To explain this model, in second half gives implementation of inference control in semantic analysis system for visiting-card understanding that classifies meaning of phrase of visiting card text into name, post and other items. Finally we report results of recognition experiment with this system.

1. はじめに

文書データの入力において、画像処理および光学的文字認識技術はデータ入力の自動化、効率化のための有効な1手段であって、文書入力の自動化をめざした多くの研究開発がなされている。近年、伝票や原稿用紙等の一定な形式の文書を自動入力する装置が製品化されており、さらに知識工学を応用して一般の文書（形式が一定でない文書）を自動入力するための文書自動認識に関する研究開発がなされている[1、2など]。しかし、文書自動認識システムは1文字レベルでの文字認識、行・段組などの文書構造の認識および文字列の集合としての文書の意味の認識など文書構造の多層に渡る認識を統合する必要があることから、いまだ解決が困難な問題となっている。特に、文書の意味の認識においては自然言語におけるあいまい性のある文法規則、意味規則による推論が必要となることから、解釈として最も適した解を効率よく求めるための推論制御機構が大きな研究課題となっていると考えられる。

文書の意味の認識では、文書の大局的な特徴あるいは局所的な特徴から既知の推論規則を適用して文書の構成要素（単語・句・文）の意味を推論する。この場合の問題は各構成要素の最適な意味を選択することであるから、分類型の探索問題（単に分類問題と呼ぶ）として一般化できる。そこで本論文ではまず分類問題において最適推論を行う推論モデルを提案し、その推論モデルに基づく推論システムの実現法について考察する。つぎに具体例として、名刺の文字列を氏名・所属などの意味項目に分類する名刺文意味解析システムにおける推論制御機構の実装および評価結果について報告する。認識対象として名刺を選択した理由は以下のとおりである。

- (1) 記載されている意味項目が氏名・所属などとほぼ一定しているが、その様式、レイアウトが多様であり、最適推論の研究対象として適当である
- (2) サンプルが入手しやすい
- (3) 住所録、名簿データの作成などにおいて実用的な価値がある

2. 名刺文意味解析について

分類問題において最適推論を行う推論モデルを考察するための準備として、具体例としての名刺文意味解析システム（以下本システムと略す）の

概要と分類問題としての句の意味の推定について説明する。

2. 1. システム構成

本システムは、図2.1に示すような名刺自動入力システムの3番目のサブシステムである。名刺自動入力システムは文字列領域の抽出[3]、文字列の認識および意味項目の解析の3つのサブシステムで構成される。本システムのブロック図を図2.2に示す。

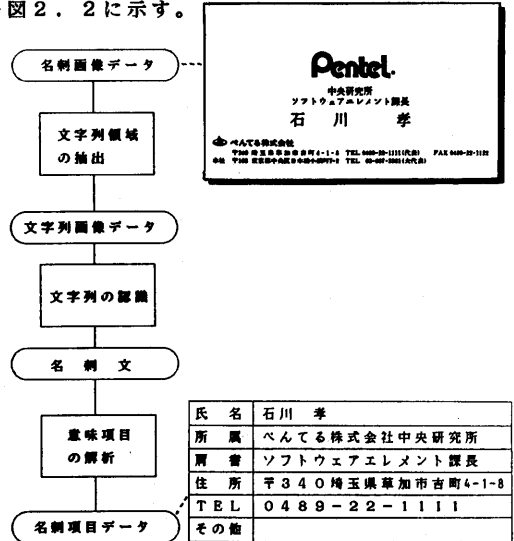


図2.1 名刺自動入力システムのフロー

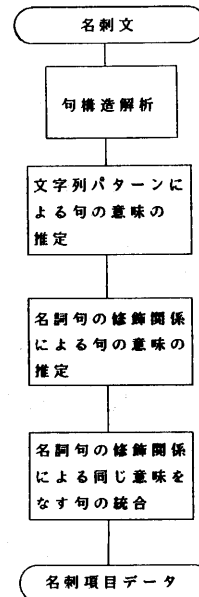


図2.2 名刺文意味解析システムのブロック図

本システムは名刺文を名刺項目データに変換する。ここで、名刺文というのは名刺画像から文字列領域の抽出、文字列の認識によって得られた文字列コードデータ（いわゆるテキストデータ）を意味する。また、名刺項目データとは氏名、所属などの意味項目に対応する文字列コードデータであって、いわば名簿データの1レコード分に相当するデータである。

本システムはまず入力された名刺文を句構造解析で句の区切りとなる改行、空白などの文字コードによって句に分解する。つぎにそれぞれの句を意味項目に特徴的な部分文字列パターンと照合し、最長一致する文字列パターンによって句の意味を推定する。この方法によって、<肩書>、<住所>、<電話番号>などの項目はかなりの部分について意味を推定できる（記号<>は意味項目を表すものとする）。

つぎに、<氏名>や一部の<社名>については固有名詞であることによって意味を推定できない場合があるので（本システムでは固有名詞の辞書を持たない）、後に述べる名詞句の修飾関係によって句の意味の推定を行う。例えば、<氏名>の場合、文字列パターンからはその意味が不明であっても、<肩書>との間の図2.3のような修飾関係を推論規則として、<肩書>が既知ならばその前後の句が<氏名>であることを推定できる。

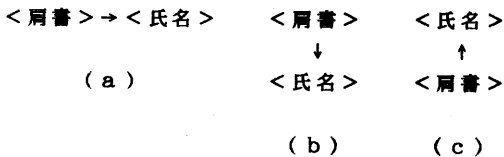


図 2.3 肩書による氏名項目の推定

図2.3(a)は名刺文において、同じ行の左側に<肩書>、その右側に<氏名>があり、<肩書>が<氏名>を修飾していることを表している。以下、本論文では修飾関係を表す横向きの矢印は同じ行において矢印をはさむ句の間に矢印の向きの修飾関係があることを表すものとする。また、(b)はより上の行の<肩書>が<氏名>を修飾し、(c)はより下の行の<肩書>が<氏名>を修飾していることを表している（上向き、下向きの矢印の意味も行が違えば横向きの矢印の意味に準ずる）。

最後に、名詞句の修飾関係によって同じ意味をなす句を統合する。例えば、<部署名>→<住所>

>という修飾関係によって、前の句の<部署名>は<住所>の一部をなし全体として<住所>を表すものとしてこれらの句を統合する。

2.2. 名刺における名詞句の修飾関係

ある2つの名詞句において一方が他方の属性を表すとき、これらの句の間には修飾関係があると定義する（この定義は言語学的に厳密な定義ではなく、ここでの定義とする）。例えば図2.1の名刺において、句「ソフトウェアエレメント課長」は句「石川」と「孝」の<肩書>という属性であり、前の句が後ろの2つの句を修飾している。また、句「石川」は句「孝」の<姓>という属性であり修飾関係にある。

このような名詞句の修飾関係を考えるとき何を主体とし何を属性と考えるかには任意性があるので、主体と属性の役割を明示するために以下では2つの名詞句の間の修飾関係を属性から主体へ向かう矢印で図示する。

後で述べる5章の実験で用いた名刺サンプルのグループI(103件)を調べた結果、<氏名>について図2.4の修飾関係があることが分かった。（企業の方の名刺が多いので<所属>は<社名>と<部署名>に分けた。）

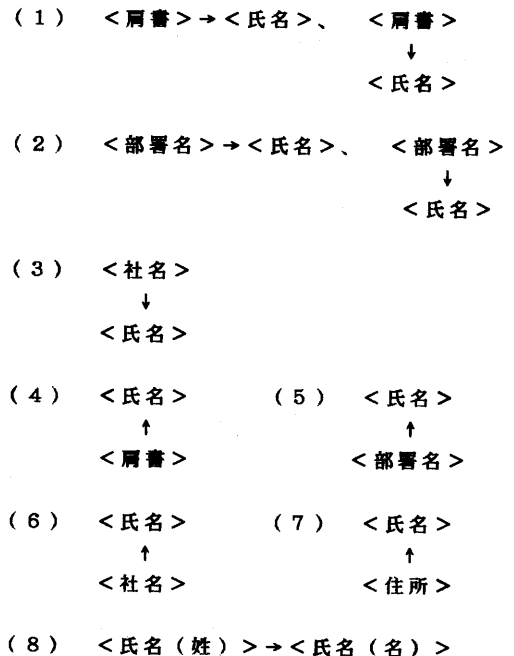


図 2.4 <氏名>についての修飾関係

2. 3. 修飾関係の競合

名詞句の修飾関係によって句の意味を推定する場合に問題になることは、修飾関係の競合が存在することである。競合の1つのパターンは図2. 4の(1)と(4)のように修飾関係にある句の前後関係が逆転する場合である(他に、(2)と(5)、(3)と(6))。このような修飾関係の前後関係の競合は意味の推定において複数の候補を生じる原因となる。例えば図2. 5に示すように2番目の句の意味が<肩書>で、句A、Bがともに意味不明である場合、どちらを<氏名>と推定することが適当であるかは、他に解の選択に関する知識がなければ一意に決定することができない。

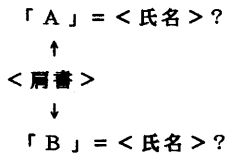


図2. 5 修飾関係の前後関係が競合する例

修飾関係の競合のもう1つのパターンは、1つの未知の句について複数の矛盾する修飾関係が存在する場合である。例えば、<社名>についての<社名> という修飾関係は <氏名> ↑ <住所> という修飾関係と競合する。この結果として、図2. 6のように2つの句があり一方の句の意味が<住所>であることが既知、他方の句の意味が未知であるとき、未知の句の意味を<社名>とするか<氏名>とするかを選択する必要が生じる。

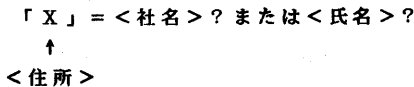


図2. 6 句の意味が矛盾する競合の例

これら2つの競合パターンが複合した例として図2. 7のような場合がある。この場合にも、句A、Bとも<氏名>である可能性があり、さらにBは<社名>である可能性があるために候補を選

択することが必要となる。

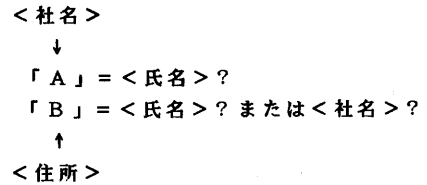


図2. 7 競合の複合した例

3. 分類問題に適した推論モデル

句の意味の推定を分類問題としてとらえて、分類問題における規則の競合解決について考察する。

3. 1. 規則の競合の集合論的考察

2種類の分類項目A、Bがあり、対象が分類項目Aであることを推定する規則を R_A 、Bであることを推定する規則を R_B 、さらに R_A を満たすものの集合を $D(R_A)$ 、 R_B を満たすものの集合を $D(R_B)$ とする(図3. 1)。ここで規則 R_A と R_B が競合するのは、 $D(R_A) \cap D(R_B) \neq \emptyset$ であるときである($D(R_A) \cap D(R_B) = \emptyset$ なら R_A と R_B は両立し、競合しない)。

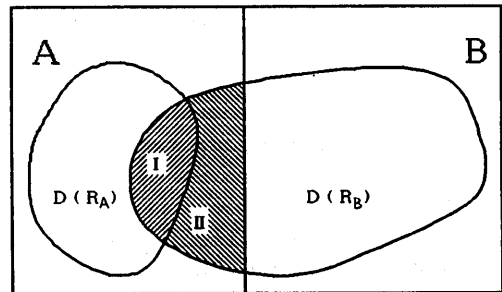


図3. 1 規則の競合のモデル

図3. 1の例において R_A と R_B は競合しており、領域I、IIが誤認識となる部分である。領域IIは R_A の推定能力が不足していることに起因する第1種の誤りであるから一旦無視するとして、規則が競合する領域Iについて考える。ここで規則が競合する本質的な原因は規則 R_B がAをBと誤って推定する場合をもつためであり(第2種の誤り)、規則 R_A には問題はない。一般的に、全ての推論規

則の推定に誤りがなく規則の間に競合がないことが望ましい訳であるが、規則の誤りをなくそうとすると規則の推定能力が小さくなってしまい、推定不能 (reject) のケースが増えてしまう。そこで、 R_B の推定能力を落とさずに競合をなくす方法を考えると、 $D(R_B)$ を $\overline{D(R_A)}$ に制限すれば領域 I は正しく A と認識されることになる (上線は補集合を表す)。すなわち、規則 R_A を R_B より優先して適用すれば正しい解を得ることができることになる。また、一旦無視していた領域 II については図 3. 2 のように A であることを推定する新たな規則 $R_{A'}$ を追加すれば解決でき、このとき正しい $D(R_B)$ は

$$D(R_B) \cap D(R_A) \cap \overline{D(R_{A'})}$$

となり、最適推論のための規則の優先順位は $R_{A'} > R_A > R_B$ または $R_A > R_{A'} > R_B$ となる (R_A と $R_{A'}$ は両立するので適用順序は任意でよい)。

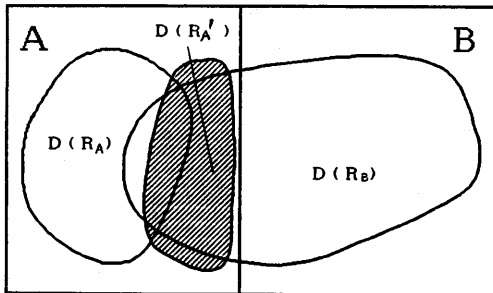


図 3. 2 規則の追加による競合解決

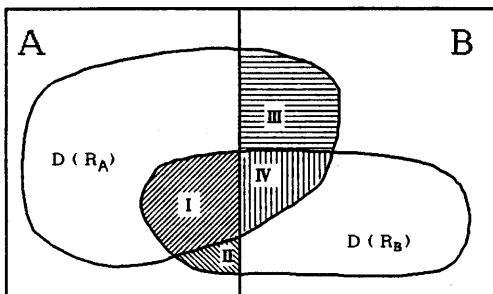


図 3. 3 両方の規則に誤りがある場合

図 3. 1 の例では一方の規則に第 2 種の誤りがある場合を考えたが、図 3. 3 のように両方の規則に第 2 種の誤りがある場合には以下のようにすればよい。図 3. 3 の領域 I、II は図 3. 1 の場合と同様にして解決でき、領域 III、IV についても B であることを推定する新たな規則 $R_{B'}$ 、 $R_{B''}$ を追加すればよい (図 3. 4)。以上の考察では

2 つの規則について考えたが、3 つ以上の場合についても 2 つずつの組を考えれば状況は同じである。

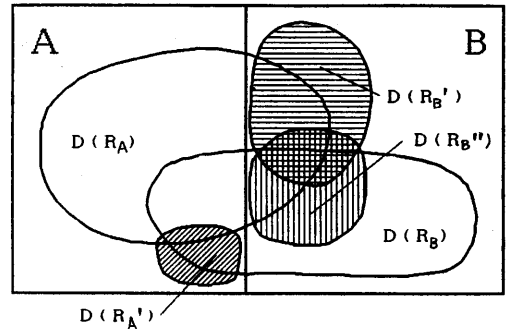


図 3. 4 両方の規則に誤りがある場合の解決法

このように分類問題においては、分類項目を推定する規則に優先順位を付けることによって最適推論を行うことができるので、この方法は「排他的分類における例外の除去の原理」として一般化できる。すなわち、対象が必ず 1 つの分類項目に属する排他的分類においては、分類項目を推定する規則に誤りがあるとき、誤りとなる部分集合を例外として区別する規則を設けることによって正しい分類項目の推定ができる。

また、規則の競合解決に関する考察において、正しい認識結果を得るための規則の優先順位は個々の規則の推定能力によって決まるものであるから、規則の推定能力が対象集合に依存して決まることを考えると、ある対象集合を限定すれば規則の優先順位は常に一定であると考えられる。

3. 2. 優先順位による規則の競合解決

分類問題において複数の解候補から 1 つを選択する場合に、分類項目の選択基準に優先順位を付けることが通常行われている。同様に推論規則によって分類問題を解くシステムで最適解を得る方法として、1) メタ知識による推論制御、2) 競合解決戦略に基づく規則の選択などがある。前者はメタ知識を使って最適解を得る規則を選択し、後者は競合解決戦略に基づいて規則を選択する。これらの方法は本質的には競合する規則に優先順位を付けることで最適解を得るものであるが、優先順位の付け方が異なっているために一長一短がある。前者はきめ細かい制御ができるもののメタ知識が一般に複雑になりシステム構築が難しい。また後者はシステム化は比較的簡単であるが、確実に最適解を得ることができにくい。

そこで規則に優先順位を付ける新しい方法として、競合する規則の間のみ優先順位を付ける推論モデルを提案する。この推論モデルは、規則の優先順位を個々の規則の間の関係として定義するものである。この方法では、優先順位が局所的な関係として定義できるので最適推論する規則のシステム化がしやすく、また、競合しない規則は並列的に適用できるために将来の並列処理化において実行を効率化できる可能性があるという利点がある。

3. 3. 推論制御機構

つぎに推論規則の優先順位による推論の制御をどのように実現するかについて検討する。今、2つの規則 R_a と R_b の間に優先順位 $R_a > R_b$ が存在する場合を考える(図3.4の場合)。この優先順位のもとの意味は、 R_b による推定において正しい $D(R_b)$ が $D(R_a) \cap D(R_b)$ でなければならないことを意味するので、 R_a が成り立たないときに R_b が成り立つときの推定が正しいことを優先順位は意味していると解釈できる。この解釈によって規則の優先順位を図3.5のように手続き的に表現できる。この手続きは、プロダクション・システムでは「 R_a が成り立つならば A であるという命題が成立し、 R_a が成り立たないときに R_b が成り立つならば B であるという命題が成り立つ」と解釈し実行する。

```

IF      Raが成り立つ
THEN   Aである
ELSE
    IF      Rbが成り立つ
    THEN   Bである

```

図3.5 規則の優先順位の手続き的表現

この手続き的な制御構造は構造化プログラミング手法の分岐構造に他ならない。そこで我々はこのように構造化された規則をもつプロダクション・システムを構造化プロダクション・システム(Structured Production System)と呼ぶことにする。構造化プロダクション・システムでは規則の優先順位を知識ベースに明示的に表現でき、しかもメタ規則の導入などのような特別な制御構造を必要とせずに、最適な解を得る推論システムを実現することができる。この推論システムでは競合する規則にあらかじめ優先順位を付けるので、あ

るデータベースの状態において条件を満たす全ての規則を適用してデータベースを更新するというサイクルを実行することで最適推論ができる。

4. 名刺文意味解析システムの実際

この章では優先順位付きの推論モデルに基づく推論システムの構成について名刺文意味解析システムを例に説明する。

我々が採用した推論システムの基本構造はプロダクション・システムである。プロダクション・システムは、データベース/規則/制御の各要素が分離しているのでAIシステムの透明な記述を構成するのに適している。また、句の意味の推定という問題においては終状態を明示的に与えることができないので、文字列パターンから分かる句の意味と句の位置関係を初期状態として、修飾関係によって句の意味を推定する規則を順次適用して不明な句の意味を推定してゆく前向き推論方式が適していると考えられる。

推論システムを構成するデータベースは句の位置関係、字句的な意味などの句に関する事実を記述するものであって、修飾関係による句の意味の推定の基礎情報となる。2番目の要素である規則は句の意味を推定するための推論規則であって、句の修飾関係による推論規則、字句的な特徴による推論規則などがある。最後の要素である制御は規則を順次適用して、解釈として最も適した句の意味を選択するための手続きに関するものである。

4. 1. データベース

本システムのデータベースは基本的に<対象>、<属性>、<値>の3組である命題によって対象を記述している。具体的には、本システムの入力データである名刺文は、例えば図4.1のようにデータベースに表現される。

```

(#:001  is-a    名刺文)
(#:001  文字列
"PENTEL
  中央研究所
  ソフトウェアエレメント課長
  石川 孝
  . . . . .")
. . . . .

```

図4.1 データベース表現の例

図4.1で、'#:001'は対象を識別するオブジェクトidを表すシンボルであり、'is-a'はクラス名を表す属性名である。図4.1の表現はオブジェクトidの記述が冗長なので、図4.2のようなフレーム的表現に置き換えられる(内部的にはこのような表現をとっている)。

```
(#:001 (is-a 名刺文)
      (文字列 "PENTE L . . .")
      . . . . .)
```

図4.2 フレーム的データベース表現

図4.1(または図4.2)のデータベースから句構造解析および文字列パターンによる句の意味の推定によって図4.3のようなデータベースが生成される。

```
(#:001 (is-a 名刺文) . . . . .)
(#:002 (is-a 行))
(#:003 (is-a 句)
      (part-of #:002)
      (まえの句 nil)
      (文字列
       "PENTE L")
      (意味 nil)
      . . . . .)
. . . . .
```

図4.3 句レベルでのデータベースの例

図4.3で、'part-of'は所属関係を表す属性名であって、(X part-of Y)は「XがYに属している」ことを表す。属性名'意味'に対応する値は句の意味を表しており、その値が'nil'であるのは文字列パターンによっては句の意味が推定できなかったことを表している。修飾関係による句の意味の推定は図4.3の段階のデータベースを始状態とする。

4.2. ルール

つぎに、ルールは図4.4のBNFで定義する構文によって表現する。構文要素の意味は以下のとおりである。

- (1) <ルールid>はルールを識別するためのシンボルまたは文字列とする。
- (2) 超記号 { } * は0個以上の繰り返しを表す。
- (3) 超記号 { } + は1個以上の繰り返しを表す。
- (4) 組み込み述語 or は or に続く条件式のいずれかが成り立つときに成り立つ。
- (5) 組み込み述語 not は not に続く条件式が成り立たないときに成り立つ。
- (6) 組み込み述語 eval は別にLispで定義した述語の評価を意味する。
- (7) 条件式中の<変数>は<変数>が最初に現れた条件式が成り立つとき、照合するデータベースの命題の対応する値が代入される。1度値が代入された<変数>はその値と等価である。
- (8) 組み込み関数 remove はそれに続く<命題>をデータベースから削除する。
- (9) 組み込み関数 eval は別にLispで定義した関数を評価し、その結果をデータベースに追加する。

```
<ルール> ::= (<ルールid>
             <IF部> <THEN部>
             [ <ELSE部> ])
<ルールid> ::= <シンボル> | <文字列>
<IF部> ::= (IF { <条件式> } *)
<条件式> ::= (<対象>
             { (<属性> <値> ) } *)
             | (<対象> <属性> <値> )
             | (or { <条件式> } +)
             | (not <条件式> )
             | (eval <Lisp述語> )
             | nil
<対象> ::= <シンボル> | <変数>
<属性> ::= <シンボル> | <変数>
<値> ::= <シンボル> | <変数>
           | (or { <値> } *)
           | (not <値> )
<変数> ::= <$ で始まるシンボル>
<THEN部> ::= <THEN { <命題> } * >
<命題> ::= (<対象> <属性> <値> )
           | (remove <命題> )
           | (eval <Lisp関数> )
<ELSE部> ::= (ELSE <ルール> ) |
             (ELSE { <命題> } *) | nil
```

図4.4 ルールの構文

ルールはつぎのように解釈して実行する。

- (10) < I F 部 > を評価してすべての < 条件式 > が成り立てば、< T H E N 部 > を評価する。このとき、新しい < 命題 > のみが追加される。もし、< I F 部 > がなりたないときは、< E L S E 部 > があれば < E L S E 部 > を評価する。
- (11) < E L S E 部 > の < ルール > は独立した < ルール > として評価する (前の < I F 部 > の < 変数 > はクリアされる)。
- (12) < ルール > の定義は再帰的であって < E L S E 部 > のネスティングは任意である。
- (13) < ルール > は記述された順に逐次的に評価され、< ルール > の適用結果は次のサイクルのデータベースにおいて更新される (サイクルの途中ではデータベースは更新しない)。データベースが変化しなくなればプログラムは停止する。

図 4. 5 にルールの 1 例を示す。

```
("氏名の推定：後ろの行から推定、
他に文字数の少ない氏名がない場合")
(IF ($1 (is-a 句) (項目 nil)
      (後ろの句 $2)
      (part-of $3) (文字列 $4))
   ($5 (is-a 句) (項目 "<氏名>"
                 (文字列 $6))
      (eval (length-less $4 $6))
      ($2 (is-a 句) (part-of (not $3))
           (項目 (or "<肩書>" "<部署名>"
                    "<社名>" "<住所>")))))
(THEN (remove ($1 項目 nil)
              ($1 項目 "<氏名>")))
(ELSE
 ("社名の推定：後ろの行の住所、
 まえの行の肩書から推定")
 (IF (not ($X (is-a 句) (項目 "<社名>"))
      ;他に社名がなければ
      ($1 (is-a 句) (項目 nil)
            (後ろの句 $2) (まえの句 $3))
      ($2 (is-a 句) (項目 "<住所>"
                          (part-of (not $3))))
      ($3 (is-a 句) (項目 "<肩書>"
                          (part-of (not $3))))))
      (THEN (remove ($1 項目 nil)
                    ($1 項目 "<社名>")))))
```

図 4. 5 ルールの例

図 4. 5 のルールは、「項目が nil の句について他に文字数の少ない < 氏名 > がなく、後ろの行の句が < 肩書 >、< 部署名 >、< 社名 >、< 住所 > のいずれかならば、その句を < 氏名 > と推定し、そうでないとき、項目が nil の句について他に < 社名 > がなく、後ろの行の句が < 住所 > で前の行の句が < 肩書 > ならば、その句を < 社名 > と推定する」ことを表している。

4. 3. 制御および実装

本システムにおける推論システムは、単純な前向き推論を実行する構造化プロダクション・システムとして K C L (Kyoto Common Lisp) によってインプリメントした。このプロダクション・システムは recognize and act サイクルによって前向きに推論を実行し、サイクルごとにすべてのルールを評価し (E L S E 部は I F 部が成り立たないときのみ評価する)、I F 部が成り立つすべてのルールを適用してデータベースを更新する。この際、手続的に定義された優先順位による競合解決以外の競合解決は行っていない。このアルゴリズムは推論の実行効率は悪いが、最適解の選択という目的に対しては十分である。

5. 評価

優先順位付きの推論モデルではサンプリングによって得た適当な数の学習サンプルに対して優先順位付きの規則を作成すれば、学習サンプル以外の対象に対してもある程度安定して認識できることになるが、この点について実験的検証を行った。

5. 1. 実験サンプル

まず認識対象として手持ちの名刺の中から日本語の横書き名刺を約 300 枚抽出し、ランダムにほぼ同数の 3 グループ (I、II、III) に分けた。ここで横書き名刺に限定した理由は、名刺は横書きと縦書きでレイアウトに若干の相違があるためである。また、この実験を行った時点では画像から文字列を認識するサブシステムが完成していなかったため、名刺の文字列をテキストエディタによって入力した。文字列の大きさおよび余白は無視した。

5. 2. 規則の作成

実験サンプルのグループ I (103 件) を学習サンプルとしてこのグループのすべてのサンプルを正しく認識できるように、文字列パターンと推論規則を試行錯誤して作成した。得られた文字列パターン数は 109、推論規則数は 13 であった。

5. 3. 認識実験

学習サンプルのグループ I に対して得た文字列パターンと推論規則を使ってグループ II、III (各

102件)を試験サンプルとして認識した結果と誤認識の原因を表5.1に示す。グループIIに対しては正解率が80/102、グループIIIに対しては75/102であり、推論規則に起因する誤認識率がそれぞれ5/102、7/102と少数(約6%)であることから、優先順位付きの最適推論によってかなりいい結果が得られることを確認した。この実験での誤認識の約2/3は文字列パターン不足によるものであり、文字列パターンの追加によって認識率を向上できると考えられる。

表5.1 認識実験の結果

サンプルグループ	正解数 試験数	誤認識の原因	
		文字列パターン	推論規則
II	80/102	17	5
III	75/102	20	7

注： 学習サンプルはグループI

5.4. 学習サンプル数の影響

さらに学習サンプルを増やした場合の認識率の変化を調べるために、試験サンプルI+IIを正しく認識できるように規則の追加・修正を行った。その結果、学習サンプルをIだけとした場合の試験サンプルIIIの認識率は75/102であったが、学習サンプルをI+IIにした場合には81/102に向上した(表5.2参照)。この結果において、学習サンプルを増やすことによって認識率が向上することが認められ、本論文で提案した推論モデルの妥当性を確認した。

表5.2 学習サンプル数の影響

学習サンプル	試験サンプルの認識率		文字列パターン数	推論規則数
	II	III		
I	80/102	75/102	109	13
I+II	102/102	81/102	128	18

この実験では規則をうまく作成すれば少なくとも学習サンプルのすべてを正しく認識できたが、これは学習サンプルが少ないためであると考えられ、学習サンプルをさらに増やした場合には必ずしも全数正解とはならないと考えられる。

6. まとめ

本論文の内容をまとめると以下ようになる。

- (1) 分類問題における規則の競合解決方法として優先順位付きの推論モデルを提案した。
- (2) 優先順位付きの推論モデルに基づく推論制御機構として構造化プロダクション・システムを作成し名刺文意味解析システムにおいて評価した。
- (3) 分類問題において優先順位付きの最適推論を行えば、かなりいい結果を得ることができる。

7. 今後の課題

本研究では文書自動入力システムのサブシステムとしての意味解析システムの実現について検討したが、今後より実用的な文書自動入力システムの実現をめざして以下のような課題に取り組んでいきたい。

- (1) より複雑な構造をもつ文書(例えばカタログなど)の意味解析システムの実現
- (2) 推論システムの効率的な構築手法の確立
- (3) 知識ベースの学習の効率化、自動化

8. 謝辞

本研究の実施にあたり、文書自動認識システムの構築に関して技術指導頂いた電子技術総合研究所画像処理研究室の山本和彦室長、山田博三技官、斉藤泰一技官ならびに松原仁技官に深く感謝致します。

参考文献

- [1] 木田博巳、岩城 修、荒川弘照、
「文書自動認識システムの構成法」
画像電子学会誌 Vol. 15 No. 2
(1986)
- [2] 桜井善紀、黄瀬浩一、村瀬宏一、田中直樹、
真田英彦、手塚慶一、
「名刺画像認識システムに関する考察」、
電子通信学会 PRU85-62
- [3] 山田博三、岡 昭宏、斉藤泰一、山本和彦、
「文書画像の構造理解における文字列の抽出について」
昭和63年電子情報通信学会春季全国大会、
SD-7-5