5C−5　コーパス解析に基づく事例ベースパーザ構築

島津秀雄　高島洋典
日本電気(株) C&C情報研究所

はじめに

自然言語理解の研究の歴史は長いが、その目標はかなたにあって、とかなか実用化しない。一方、近年グラフィックユーザインタフェース技術に基づくメニューシステムやハイパーテキストが普及してきている。従って、それらのインタフェースでは提供できないような機能を提供しない限り自然言語による対話インタフェースの前途は暗い。自然言語による対話の利点とは、発話者が、発話者の好みの言い方で、対象分野・目的に特有な表現で発話者主導に節約的に発話できることである。そういう発話は当然ながら非文法的表現になる。自然言語インタフェースを提供するならこういう種類の発話を扱えなくてはならない。

非文法的表現を扱うには、フレーズベース(Phrase-based)のアプローチ[1]は有望である。この手法では、対象領域で使われる言語パターンとその意味表現を対にして格納しておき、入力文に対してマッチするパターンを見つけそれに対応する意味表現を入力文の意味表現として生成する。我々は、実際に収集したコーパスの解析に基づいたフレーズベースシステムの新しい構築手法Corpus-based Parsing(以下CBPと略す)を提案する。またそれにもとづいたパターン・意味表現対と概念世界を簡単に定義できるような知識獲得機構をつくっている。システムへの要求としてはデータベースやエキスパートシステムへの自然言語質問応答システム程度を簡単に作ることである。

CBPでは、まず対象世界の実際のコーパスを収集する。パターン定義者は、コーパス中の例文の意味を1つ1つ定義していくが、その時、特定の文に対する定義だと、10文の定義をしても10種類の文しか解釈出来ない。それを1文の定義で10文、100文とマッチするように間口を広げたい。そこで、定義者が最初の定義を変形・修飾して汎化することでマッチングの間口を広げる。CBPではそのためのオペレータを提供している。

2　Corpus-based Parsing システム

収集したコーパスに基づいて定義者がパターン・意味表現対を定義していく一般的な定義の手順は、以下の流れで表わされる(図1参照)。

1. 定義モードでは、定義者はコーパスから新しい例文パターンを取り出す。それが既存のパターン・意味表現対定義で解釈出来れば終り。そうでなければ2へ。
2. 定義者は与えられた例文の意味を構築する。解釈のために足りない概念ボキャブラリや呼び名、関係等の定義を必要に応じて追加・修正・精密化する。このとき既存の概念世界のボキャブラリーを検索する必要がある。
3. 定義者は、例文を汎化する。汎化されたパターン・意味表現対と概念表現はCBPインタプリターによって解釈され、パターン・意味表現対事例ベース及び概念世界に漸増的に格納されていく。

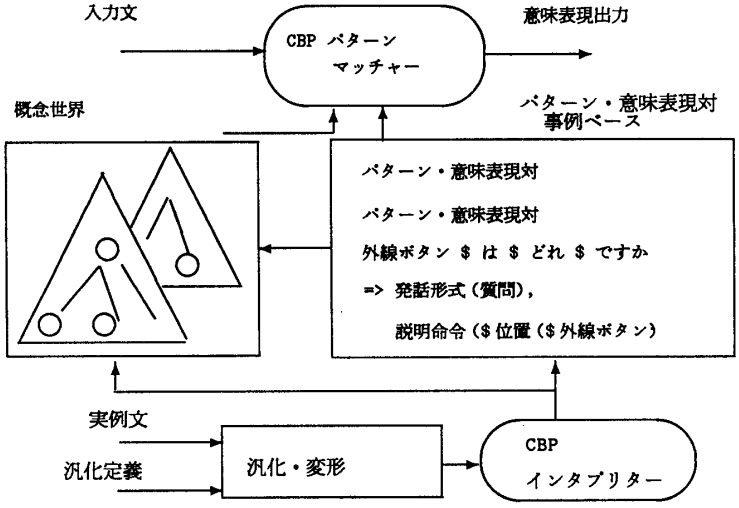

図1: CBP全体構成

実行モードでは、入力文が入ってくると、CBPパターン・マッチャーが入力文を左から右に1語ずつ取り出し、パターン・意味表現対事例ベース中の定義と比較していく。

3　パターン・意味表現対定義

CBPによるパターン・意味表現対定義の例を示す。ここでは、ある電話機の使い方に対する問い合わせのコーパス例を使って説明する。まず、定義者が「外線ボタンは、上面パネルにあります」を受けとって例1から例3のように定義したとする。

例1:外線ボタンは、上面パネルにある。
　→ $場所に属する($外線ボタン、$本体上面)
例2: 汎化(外線ボタン、$操作ボタン)は、上面パネルにある。
　→ $場所に属する(外線ボタン、$本体上面)
例3: 汎化(外線ボタン、$電話機構成部品)は、汎化(上面パネル、$電話機表面)にある。
　→ $場所に属する(外線ボタン、上面パネル)

Corpus-based Parser
Hideo SHIMAZU & Yosuke TAKASHIMA
NEC Corp.

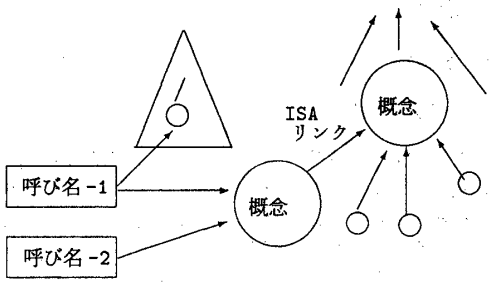

図 2: 概念表現

$で始まるものは、特定の概念の実体を表す。「外線ボタン」や「短縮ボタン」は「$操作ボタン」を継承し、「$操作ボタン」は「$スイッチ」を継承し、「$スイッチ」は「$電話機構成部品」を継承している。また、すべての操作ボタンは本体の上面に位置しているとする。例1のままでパターン・意味表現対事例ベースに格納すると、実行時に全く同一の文が入ってくれば良いが、そうでないと決してマッチしないので効率が悪い。例2中の、汎化 (a, b) とは、aを汎化してbに置き換えよという命令である。ここでは「外線ボタン」が「$操作ボタンを継承する何でも」という表現に置き換わる。例2の右辺の定義はは例1のそれと同じだが、CBP インタプリタは例2の右辺の「外線ボタン」を自動的に「$操作ボタンを継承する何でも」の形に置換する。例3のように汎化されると更に一般的になって「何かが、電話機表面のどこかにある」という表現とマッチするようになる。従って「受話器は、左側面にある」のような文もこれで解釈可能になる。

概念表現側には階層木が沢山存在する。概念表現世界を漸増的に定義していくと、対象領域のオブジェクトや操作の階層、ユーザの問い合わせ内容の階層等、自然に多重階層木表現になる。概念階層を作ると言う仕事も、実際にコーパスから見つけてこないとどういう階層があるのかわからないものである。したがって、例文の定義と概念の定義・階層化定義・精密化は、コーパスの例文を見ながら、いずれも漸増的に行なっていくことになる。

4 CBP による知識獲得

4.1 概念、概念階層の定義

概念側のモデルの構成要素は、諸概念を表すノードとそれらを関係づけるリンク、ノードに対する参照名前となる呼び名、からなる (図2参照)。パターン・意味表現対の汎化に関係するリンクは ISA である。呼び名はノードにつけられた名前である。呼び名は1つの単語でも単語列でもよい。同一の呼び名が異なる階層中の異なるノードに付けられることも、異なる呼び名が同一のノードに付けられることも可能なのでパターン定義者が概念ノードを指示する時には使いやすいが、非常に曖昧なのでCBP インタプリタが曖昧性を解消しなくてはならない。

4.2 パターン・意味表現対定義手段

パターン・意味表現対定義の一般的形式は以下の通りである。

形式1:

.... パターン、... 操作 (パターン..) .. → 理由つけ、全体意味表現、関係制約.

形式2:

.... パターン、... 操作 (パターン..) .. → 理由つけ、部分意味表現、関係制約 |
.... パターン、... 操作 (パターン..) .. → 理由つけ、部分意味表現、関係制約 |
.... パターン、... 操作 (パターン..) .. → 理由つけ、部分意味表現、関係制約 |
→ 理由つけ、全体意味表現、関係制約.

コーパスから取り出された実例文は、任意個のサブパターンに分割され、その内の任意個のサブパターンが、操作の対象となる。CBP で提供する操作オペレータの種類としては以下のものがある。

汎化 (パターン、汎化表現): パターンを汎化表現に置き換える。
同列 (パターン、兄弟表現) : パターンの参照するものとその兄弟表現が表すものの共通の親に置き換える。
省略 (パターン): 引数のパターンを省略してもよい。
補完 (補完表現): 元のパターンに、補完表現を付加する。
順不同 (パターンまたは操作表現、...): 任意順序で出現可能

形式2のように、1つの例パターンが複数の独立なパターン・意味表現対に分割できるときもある。定義者は縦棒表現を使って、元の例パターンを分割し分割された部分パターン毎にその部分意味表現等を定義する。それらの最後に、分割されたパターンが順番にならんだときに作られる元の例パターン全体としての全体意味表現を定義する。理由つけとは、ある特殊な解釈をするときに、「こういう状況だから以下のような解釈とする」という表現で理由を書くことである。CB インタプリタは、この表現を「もし、こういう状況なら」という適用条件に変換する。

関係制約は定義者が行なった汎化表現間の関係を制約として明示したい時に使われる。例えば、前出例3のパターン・意味表現対定義だけだと CBP インタプリタは「X, {isa(X, $電話機構成部品)}, は, Y, {isa(Y, $電話機表面)} ...」と展開するだけだが、意味的にはXとYの間には「YはXの位置である」という関係が満たされていないと意味表現としては正しくない。そこで、定義者が制約として関係オペレータを使って「関係 ($位置 (外線ボタン、上面パネル))」を定義して加えてやると、CBP インタプリタは「$位置 (X, Y)」を右辺に加えることをする。

5 まとめ

領域を限定した自然言語インタフェースを構築する時にフレーズベースの手法は有効である。Corpus-based Parsing は、フレーズベースシステム構築のための知識獲得モデルである。実例の解析結果を直接使っていることから事例ベース推論の一種であると位置付けることも出来る。

参考文献

[1] Arens, Y., "CLUSTERS: An Approach to Contextual Language Understanding", Rep. UCB/CSD 86/293, Ph.D. Thesis, 1986.