

統合パーサによる統合的自然言語理解システムの実現

劉 学 敏 西 田 豊 明 堂 下 修 司
京 都 大 学 工 学 部 情 報 工 学 教 室

本稿では、我々が作成した統合パーサによる自然言語理解システムの構築について報告する。統合パーサは、整合性維持エンジンCMEと確からしさ維持エンジンPMEの二つのサブエンジンを統合して構成され、自然言語処理の過程で作られた信念の集合の整合性を維持し、処理を最も確からしい方向へ誘導する。統合パーサによって構成された自然言語理解システムは、解析の過程では、縦型探索と横型探索との間の中間的な探索を効率的に行うことができる。与えられた文に対してこのシステムは、各解析レベルの部分結果を統合して利用することによって、曖昧性をできるだけ早く解消する。また、本質的に解消できないような曖昧性を含んだ文に対して、このシステムは文脈情報を利用して、最も確からしい解釈を見つけることができる。

Implementation of a Natural Language Understanding System Based on an Integrated Parsing Engine

Xuemin LIU, Toyooki NISHIDA and Shuji DOSHITA
Department of Information Science, Kyoto University
Sakyo-ku, Kyoto 606, Japan

This paper describes a natural language understanding system based on an integrated parsing engine IPE which is constructed by combining two subengines: a consistency maintenance engine CME and a plausibility maintenance engine PME. The CME and the PME work in an interactive manner so as to keep the belief set from becoming inconsistent and guide the process to the most plausible direction. Using the IPE as a subsystem, the natural language system can do a search in a manner which takes both the advantages of the breadth-first method and depth-first method and always works in the most plausible environment during the processing. We show in this paper how the system has been constructed and how it works in analyzing a given sentence and finds the most plausible interpretation by taking into account the linguistic and non-linguistic informations from various analyzing levels.

1. はじめに

自然言語には非常に多種多様な曖昧性が含まれる。例えば、単語解析時の語境界の曖昧性や同音異義などによる曖昧性、構文解析時の同じ品詞列が異なる構文構造に対応する曖昧性、意味解析時の深層の格関係の曖昧性などがある。これらの曖昧性は単にそれが発生した解析レベルの情報だけで解決できないことが多い。例えば、文の統語構造は統語情報だけを用いて一意的に決定できず、意味構造も統語や文脈に関する情報なしには決定できない。このため、従来の方法、即ち単語解析、構文解析、意味解析という逐次的な処理方法では、解析の初期段階で処理が進められないほど爆発的に多くの曖昧性が出現する。

このような問題を解決するには、統合的な解析方法、即ち、単語解析、構文解析、意味解析の三つのレベルの解析を並行して処理を行うという方法によって、曖昧性の早期解消や無駄な処理の回避、最も確からしい解釈を見つける処理を行うことが望ましい。しかし、統合的な処理をどのように効率的に実現するかが長年の課題であった。

本稿では、統合的な処理を実現するための新しいメカニズムを提案する。我々は論理的な制約を管理する推論エンジンCMEと、確からしさにに関する情報を管理する推論エンジンPMEを統合して、統合的な処理のための推論エンジン(統合パーサ)を作成した^[4]。この統合パーサは次のような機能を持っている。

- (1) 曖昧性に応じた仮定の生成
- (2) 仮定の異なる集合における推論
- (3) 処理過程の確からしい方向への誘導
- (4) 矛盾を検出したときの仮定の組合せの切り換え
- (5) 新しい情報に応じた確からしさの修正

この統合パーサは、自然言語処理の各解析レベルの情報(言語的と非言語的)を一様な言明(信念)または仮定として取り扱い、CMEとPMEによって論理的な側面と確からしさの側面から管理する。自然言語理解システムは、処理の過程では、統合パーサが管理する情報を統合することによって、曖昧性を早期に解決し、処理を最も確からしい方向に向ける。これによって、従来の逐次的な処理方法のような曖昧性による無駄な処理を大幅に減少することができ、しかも唯一的な解釈が導出できないような曖昧な文に対しても、システムは知識や文脈情報を利用して、最適な解釈を見つけることができる。

本稿では、我々の基本的な考え方と実現手法を示す。以下の第2節は統合パーサの構成について説明し、第3節は統合パーサを利用した自然言語処理システムの構成を示す。また、このシステムの振舞いを示すために、第4節では一つの実行例をあげる。

2. 統合パーサの構成

統合パーサは二つのサブエンジンCMEとPMEを統合して構成されている。以下では、各々のエンジンについて説明する。

2.1 整合性維持エンジンCME

整合性維持エンジンCME(Consistency Maintenance Engine)は、自然言語処理の過程で作られた言明の集合の論理的な整合性を維持し、処理の過程が矛盾に陥らないことを保証する。CMEは与えられた言明や仮定の間の依存関係を記憶し、可能な推論の方向を保持する。そして、推論サブシステムからの質問に対する応答、仮定の変更に応じた作業用記憶の変更、矛盾が検出されたときの仮定の組合せの切り換えなどを行う。

2.1.1 CMEとATMS

整合性維持エンジンCME(Consistency Maintenance Engine)は、de KleerのATMS^[11](Assumption-based Truth Maintenance System)の技法をベースにして構成した。

ATMSと同じように、CMEの管理対象の基本となるものは節点で表す。節点は大別して、仮定節点と非仮定節点の二種類がある。仮定節点は自分自身に依存する節点である。このような仮定は純仮定と呼ばれる。仮定節点以外の全ての節点は、最終的には仮定節点に依存する。仮定の組合せは環境と呼ばれ、ある環境において成り立つ全ての節点はこの環境のCONTEXTと呼ばれる。節点には、この節点が成り立つ全ての環境をLABELというデータ構造で記録する。また、節点の間の依存関係は理由付けの形で記録する。CMEは節点に与えられた理由付けの集合を用いて、その節点の成り立つ環境、即ちその節点のLABELを動的に更新する。また、ある環境を与えると、CMEは節点のLABELによって、その環境に対応するCONTEXTを決定することができる。

ATMSと異なると、我々のCMEは節点の集合を管理するだけではなく、仮定の組合せからなる環境の管理も考える。また、異なった環境の確からしさを比較するために、我々のCMEでは、全ての仮定節点に対して、その確からしさを表す重みを付加する。これらの点について以下で述べる。

2.1.2 推論環境の管理

ATMSは本来横形探索を指向した推論管理システムである。予め全ての仮定を生成し、ある拘束条件を満たす全ての解(各々は仮定のある組合せに対応する)を見つける場合、この方法は効率的である。しかし、自然言語理解の場合、曖昧性が多いため、横形探索は効率が悪い。一般的に、仮定の数を n とすると、可能な環境の数は 2^n 乗となる。従って、仮定の数が増えると、

可能な環境の数は急激に増大する。計算速度や記憶容量上の制約のため、これらの環境の全てを考えるのは不可能である。また、仮定間の排他性のため、仮定の全ての組合せを考える必要はない。ここで、次の例を見よう。

例1 あめをかう (S1)

この文には、単語の同音異義の曖昧性が二ヶ所ある。単語「あめ」は「雨」、「飴」のどちらでも可能であるし、また、単語「かう」は「買う」と「飼う」二つの可能性がある。ここで、単語「あめ」をword-1とし、単語「かう」をword-3とする。これらの曖昧性は次のように表す。

word-1 → noun-1, noun-2 (1)

word-3 → verb-1, verb-2 (2)

ここで、節点noun-1, noun-2はそれぞれ名詞「雨」、「飴」を表し、verb-1, verb-2はそれぞれ動詞「買う」、「飼う」を表す。また、本稿では、次の記号

$X_1, \dots, X_m \rightarrow Y_1, \dots, Y_n$

を利用して、 $X_1, \dots, X_n$ から排他的に $Y_1, \dots, Y_n$ を導出することを表す。

システムは上の(1)の曖昧性に対して、仮定節点A1(word-1を名詞「雨」とする仮定)とA2(word-1を名詞「飴」とする仮定)を作り、(2)の曖昧性に対して、仮定節点B1(word-3を動詞「買う」とする仮定)とB2(word-3を動詞「飼う」とする仮定)を作るとする。

この四つの仮定からなる可能な環境は16個があるが、そのうち、初めから考える必要のないものが多い。例えば、単語word-1の曖昧性に応じて作られた仮定A1とA2は互いに排他的であるため、仮定の組合せ $\{A1, A2\}$ を含む環境は初めから矛盾する。また、単語word-3の曖昧性に応じて作られた仮定B1とB2も同様である。従って、仮定の組合せ $\{A1, A2\}$ 或は $\{B1, B2\}$ を部分集合として含んでいる環境における処理は無駄である。

この問題を解決するために、我々のCMEでは、仮定の排他的という特性を利用して、次に述べる環境の木という技法を用いて、無意味な環境をはじめから作り出さず、推論可能な環境だけを取り出す。

まず、我々は仮定の組合せを表す環境節点というような特殊な節点を導入する。このような節点を利用して、環境の木は次のように構成する。仮定を全く含まない環境 $\{\}$ を環境の木のルート節点とする。処理はこの環境から始まる。処理の過程である環境Eから曖昧性が生じ、排他的にk個の仮定 $A_1, \dots, A_k$ が作られると、CMEは新たに作られた仮定の各々をもとの環境Eに加えて、k個の新しい推論環境 $E_1, \dots, E_k$ を生成する。ここで、 $E_i = E \cup \{A_i\}$ である。これらの新しい環境をもとの環境の子孫環境と呼ぶ。また、環境の間の依存関係を記録するために、次のような理由付けを用いる。

$E, A_i \rightarrow E_i \quad (i=1, \dots, k)$

こうすると、全ての推論可能な環境は一つの木になる。例えば、前述の例1の処理の過程で生成された環境の木は図1のようになる。更に、我々は環境節点に対して、状態を付加する。矛盾しない環境にはその状態をINにし、矛盾した環境にはその状態をOUTにする。ある環境がOUTになったら、CMEはその全ての子孫環境もOUTにする。また、環境の状態の変化に応じて、指定する操作を起動することができる。

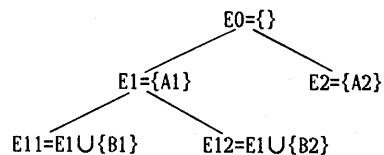


図1. 例1の環境の木

環境の木の任意の節点はシステムの一つ可能な処理方向を指示する。但し、枝節点はその子孫節点の部分集合であるため、考慮する必要がない。CMEはシステムの処理の一つの環境に制限する。この環境を現在の推論環境と呼ぶ。処理の過程の任意の時点において、システムは現在の環境の節点しか見えない。環境の木によって、CMEはシステムの全ての可能な処理方向を保持する一方、考える必要のない環境を初めから排除する。従って、ATMSの場合に比べて、無駄な処理がかなり減少する。

2.1.3 CMEによる整合性の維持

CMEは処理システムに対して次の機能を提供する。

- (a) 曖昧性に応じて仮定を作る。
- (b) 新しい推論環境を生成し、環境の木を拡張する。
- (c) 矛盾した環境を削除する。
- (d) 現在の環境で成り立つ信念の集合を決定する。

処理システムはこれらの機能を利用して、CMEに仮定を作らせたり、新しい推論環境を生成させたりする。システムは処理の過程で、既存した信念 $X_1, \dots, X_n$ から新しい結果 $Y_1, \dots, Y_m$ を導出したとき、CMEに対して、次のような理由付けを与える。

$X_1, \dots, X_k \rightarrow Y_1, \dots, Y_m$

すると、CMEは次のように動作する。

(1) $m=1$ の場合

これは実際に論理的な推論であるため、CMEはその理由付けを記録すればよい。

(2) $m>1$ の場合

CMEはその右辺の諸々の可能性に対して、 m 個の仮定 $A_1, \dots, A_m$ を作る。それぞれは「節点 $X_1, \dots, X_k$ から導出可能な結論は Y_i である」と仮定する。CMEは作られた諸仮定を利用して、新しい環境 $E_1, \dots, E_m$ を生成し、

もとの理由付けを次のような m 個の理由付け

$$X_1, \dots, X_k, A_j \rightarrow Y_j$$

に変換する。ここで、 $E_i = E \cup \{A_i\}$ 、 E は現在の推論環境である。次に、CMEは新しい環境を生成したことをシステムに報告する。すると、システムは次節で述べる確からしさ管理エンジンPMEを利用して、生成された新しい環境のうち最も確からしいものを一つ選択し、処理を続ける。

解析の過程で矛盾が検出されると、処理システムは現在の推論環境が矛盾したことをCMEに通知する。すると、CMEは節点の間の依存関係を利用して、矛盾の原因となった仮定(または仮定の組合せ)を探して、環境の木から矛盾した環境を削除する。

2.2. 確からしさ管理エンジンPME

確からしさ維持エンジンPME(Plausibility Maintenance Engine)は仮定に与えられた確からしさと環境の木を利用して、一つの信念ネットワークを構成するこれを利用して、環境の確からしさを計算し、最も確からしい環境を選択する。また、新しい情報が得られたとき、仮定の確からしさを更新する。

2.2.1 信念ネットワークの構成

信念ネットワークを構成するため、我々はまず仮定節点に対して、確からしさを導入する。仮定を生成したとき、処理システムは今までの処理結果と与えられた知識を利用して、各々の仮定の確からしさを決定する。PMEは仮定の確からしさを利用して、環境の確からしさを決定し、最も確からしい環境を選択する。例えば、前述の例1において、もし仮定A1に高い確からしさが与えられれば、システムは単語「あめ」を「雨」と見なして処理を続け、逆に、仮定A2に高い確からしさが与えられれば、システムは単語「あめ」を「飴」と見なして処理を続ける。

仮定の確からしさは仮定を生成した時点に知識と処理結果によって決定されたものである。これは不変なものではない。処理の進行に伴って、矛盾が検出されたり、新しい結果が得られたりすると、確からしさを修正する必要がある。このため、我々はPearlらの信念ネットワーク^[2]の修正手法を参照し、環境の木を利用して、信念ネットワークを構成し、これを利用して、環境の確からしさを修正する。信念ネットワークを次のように構成する。

環境の木の節点を一つの確率変数とみなし、これに対して値域 $\{0, 1\}$ を与える。環境が矛盾した場合、その値を0にし、環境に含まれる仮定の全てが解析によって事実となることがわかったとき、その値を1にする。それ以外の場合、その節点の値をNILにする。仮定の確からしさをその仮定の成り立つ事前確率と見なし、環境

E の確信度 $BEL(E)$ を、その環境の値が1である確率として定義する。

ある環境 E で仮定 $A_1, \dots, A_n$ が生成され、これによって子孫環境 $E_1, \dots, E_n$ を生成したとしよう。(ここで、 $E_i = E \cup \{A_i\}$ 、 $i=1, \dots, n$ である)。この場合、新しく生成された各子孫環境 E_i の親環境 E に関する条件付き確率を次のように定義する。

$$P(E_i=0 | E=0)=1, \quad (1)$$

$$P(E_i=1 | E=0)=0, \quad (2)$$

$$P(E_i=0 | E=1)=1-p_i, \quad (3)$$

$$P(E_i=1 | E=1)=p_i. \quad (4)$$

ここで、 p_i は仮定 A_i の確からしさである。上の4つの条件付き確率のうち、(4)だけを記録すればよい。

2.2.2 処理方向の誘導

PMEによる処理方向の誘導は環境の木の葉節点のうち最も確からしいものを選択することによって実現する。推論環境を選択する必要性は次の二つの場合に生じる。

- (1) 現在の推論環境から子孫環境が生成された場合。
- (2) 現在の推論環境が矛盾した場合。

上の(1)の場合に対して、PMEは現在の環境の子孫環境のうちその条件付き確率の最も大きいものを選択すればよい。

一方、(2)の場合の環境の選択は環境の木の全ての葉節点に対して行う。PMEは環境の木のルート節点から出発して、確からしい方向に向かって探索する。ある枝節点 E に到着すると、 E の子孫節点の確信度(BEL 関数値)を比較することによって、次の探索方向を決定する。例えば、 E の子孫環境が $E_1, E_2, \dots, E_n$ である場合、探索は BEL 関数値の最大の子孫環境 E_i で指示する方向に進む。このようにして、環境の木の葉節点に到着するまで探索を続ける。最後に到着する葉節点で表す環境を次の推論環境として選出する。

2.2.3 確からしさの修正

推論の過程において、矛盾が生じたとき、システムはPMEを呼び出し、確からしさの修正を行う。確からしさの修正は、実際には各節点の観察結果(現在の環境が矛盾する)に対する事後確率を計算する。この場合、PMEはまず矛盾した環境に値0を与え、次に、その値を利用して、各環境の確信度を再計算する。

以下では、便宜上、我々は E で事象 $E=1$ を表し、 $\neg E$ で事象 $E=0$ を表す。ある環境 E_j が矛盾した場合、その親環境 E の事後確率はベイズの定理によって次のように計算できる。

$$P(E | \neg E_j) = \frac{P(\neg E_j | E) \cdot P(E)}{P(\neg E_j)}$$

$$= \frac{(1-P(E_i | E)) \cdot P(E)}{(1-P(E_j))} \quad (a)$$

ここで、 $P(E)$ と $P(E_j)$ 各々は環境 E と E_j の確信度である。また、同じ曖昧性に応じて作られた仮定同士は互いに排他的であるため、もし E_i と E_j は兄弟であるならば、次の式(b)が成立する。

$$P(E_i | E_j) = 0 \quad (b)$$

これを式(a)に代入して、次の式(c)が得られる。

$$P(E_i | \neg E_j) = \frac{P(E_i)}{1-P(E_j)} \quad (c)$$

PMEは上の式を利用して、各環境の事後確率を計算し、その計算結果によって、確からしさを修正する。例えば、 E が E_i の親環境であるとする、環境 E_j が矛盾した場合、 E_i の E に関する条件付き確率を次のように変更する。

$$P'(E_i | E) = \frac{P(E_i | \neg E_j)}{P(E_i | \neg E_j)} \quad (d)$$

ここで、PMEの動きを前節の例1を用いて示す。

「あめ」からなる単語word-1に曖昧性が生じ、システムは節点noun-1に対応する仮定A1に確からしさ2/3を与え、noun-2に対応する仮定A2に確からしさ1/3を与え、PMEは推論を仮定A1が含まれる環境E1に誘導する。

その後、この環境で単語「かう」が検出され、これを認識しようとしたとき、その曖昧性が生じた。これに対して、システムは仮定B1、B2を生成し、それぞれに確からしさとして同様に2/3と1/3を与えるとする。すると、PMEは処理過程を仮定B1を含む環境E11に誘導する。この場合の環境の木からなる信念ネットワークを図2に示す。

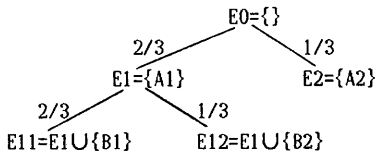


図2. 例1の信念ネットワーク

環境E11において、システムは文(S1)の意味概念を導出しようとする。このとき、システムは「雨を買う」ことは意味上で矛盾することがわかる。従って、いまの環境から矛盾が生じたので、それを削除しなければならない。すると、システムはCMEを呼び出して、環境E11を削除し、次いで、PMEは各環境の確からしさを修正する。

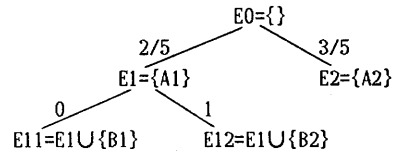


図3. 修正後の信念ネットワーク

修正された信念ネットワークを図3に示す。修正した後、環境E2の確信度が環境E12の確信度より高くなるので、PMEはE2を次の環境として選択し、システムは単語「あめ」を「飴」を見なして処理を続ける。

この例で示したように、我々のシステムは状態空間(環境の木)を探索する場合に採用した方法は横型探索方法でもなく、縦型探索方法でもない。この二つの方法の間の中間的な探索方法である。例から分かるように、環境E11が矛盾した場合、システムはこの環境の兄弟環境E12を選択したのではなく、現時点で最も確からしい環境E2を選択したのである。しかし、縦型探索の場合、システムは環境E12を選択するのが普通であろう。この例では、もし環境E12が選択されたら、この環境の処理がまた無駄になることが明かである。一方、PMEによって処理の方向を誘導する場合、このような無駄を避けることができる。

2.3 CMEとPMEの統合

CMEとPMEは互いに無関係なものではない。この二つのスプエンジンは互いに作用し、協同して信念の集合を管理する。

既に述べたように、環境節点には状態があり、その状態の変化によって指定する操作を起動することができる。実際には環境に付加された操作は、対応した確からしさの修正命令である。即ち、PMEを起動する操作である。従って、ある環境が矛盾したら、CMEはその状態をOUTにすると同時に、PMEの処理が始まる。

また、確からしさを修正する過程で、ある環境の確信度が0になったら、PMEはこの環境を矛盾環境と見なして、CMEを呼び出し、この環境を削除する。例えば、ある環境の全ての子孫環境が矛盾した場合、この環境の確信度は0になる。これはこの環境における全ての可能な処理が失敗したことを意味する。従って、これは実際に矛盾した環境である。しかし、このような環境自身は直接矛盾する節点をサポートしないから、処理システムだけでは検出できない。一方、CMEとPMEを結合すると、このような環境も削除できる。

3. 統合パーサによる自然言語理解システム

統合パーサを使った統合的自然言語理解システムの

構造を図4に示す。システムへの入力は一ローマ字列で表す仮名文字列からなる日本語の文であり、出力は入力文の意味表現のリストである。システムの知識ベースには言語的な知識(辞書、構文規則など)と対象世界に関する知識(単語の意味概念や意味概念の間の関連など)のほか、文脈情報も格納される。

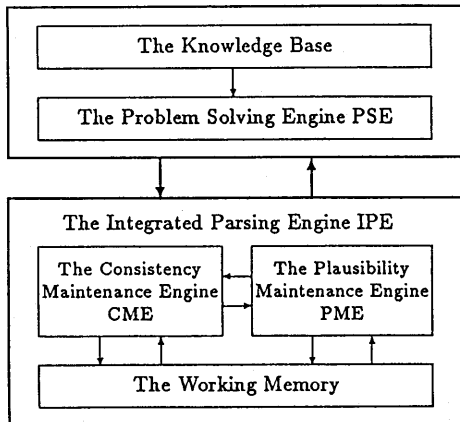


図4. 統合パーサを利用した自然言語処理システムの構造

3.1 基本的な処理の流れ

システムは問題解決エンジンPSEの制御下で動作する。システムに一つの入力文を与えると、PSEはまず環境の木を初期化し、次いで、自然言語処理の各レベルの解析を並行して行なう。処理の過程で作られた言明や仮定はすべてCMEによって管理し、また、仮定に関する確からしさの知識はPMEに与える。CMEはシステムの処理過程が矛盾に陥らないことを保証する一方、PMEは確からしさの知識を利用して、システムの処理を確からしい方向に誘導する。

システムの処理の基本的な流れは次のようになる。

[step1] 前処理

最初の推論環境 $E_0 = \{\}$ を作って、これを環境の木のルートにする。 E_0 には何の仮定も含まれないから、それに対して、事前確率を1にする。環境 E_0 を現在の推論環境として、環境変数 E に記録する。(以下は E で現在の推論環境を表す。)

[step2]

環境 E において、次の事象が発生するまで、解析の処理を繰り返し行う。

(A)入力文に対する意味解釈が生成された。

(B)曖昧性が生じた。これに対して新しい環境が生成した。

(C)矛盾が検出され、現在の推論環境には矛盾した仮定あるいはその組合せが含まれていることがわかった。

[step3]

[step2]の終了条件をチェックし、下のように次の処理を決定する。

条件A)によって[step2]を終了した場合、PSEは生成された意味表現リストを出力し、それを文脈情報として記録した後、処理を終了する。

条件B)によって[step2]を終了した場合、[step4]に飛んで、新しい環境を選択する。

条件C)によって[step2]を終了した場合、PSEはまずCMEを呼び出して、矛盾の解消を行う。即ち、環境の木から全ての矛盾した仮定或は仮定の組合せを含んでいる環境を削除する。次に、PSEはPMEによって、確からしさの修正を行う。

[step4]

PMEは環境の木から今の時点で最も確からしい環境を選択して、これを環境変数 E に設定し、制御をPSEに戻す。

ただし、選出できない場合、即ち、環境の木が空になった場合、PMEは環境変数 E をNILにする。すると、PSEは入力に誤ったことがわかり、エラーメッセージを出力し、処理を失敗として終了する。

[step5]

[step2]に戻って、処理を続ける。

3.2 曖昧性の解消

自然言語理解を行う場合、まず考えなければならないことはその曖昧性への対処方法である。曖昧性が生じると、それに対応して何かの仮定を作られなければならない。ここで、本研究でのシステムが曖昧性に対してどのように仮定を生成するか、また、生成された仮定がどのように役にたつかを説明する。仮定には大別すると、次の二種類がある。

(1) 直接信念をサポートする仮定

この種の仮定はシステムが何を信じるかを指示するための仮定である。このような仮定は直接中間結果をサポートする。これによって、曖昧性に応じて可能な排他的な推論結果はそれぞれ異なった環境で成立する。例えば、前述の例1の処理の過程で、単語「あめ」の曖昧性に応じて作られた仮定A1とA2を利用して、次の理由付けを与える。

word-1, A1 $\rightarrow$ noun-1

word-1, A2 $\rightarrow$ noun-2

こうすると、noun-1は仮定A1を含んだ環境だけに成立し、また、noun-2は仮定A2を含んだ環境だけに成立するようになる。従って、仮定A1を含んだ環境 E_1 においては、noun-2が成立しないから、noun-1をword-1の”論理的な結論”と見なすことができる。

(2) 処理の過程を制御するための仮定

この種の仮定を説明するために、まず次の例を考え

よう。

例2 小さい小学校の門

これに対する品詞列は次のようになる。

形容詞 名詞 助詞 名詞

いま、システムに次の構文規則を与えたとしよう。

G1: 連体修飾句 $\Rightarrow$ 名詞句+助詞「の」

G2: 名詞句 $\Rightarrow$ 名詞

G3: 名詞句 $\Rightarrow$ 形容詞+名詞句

G4: 名詞句 $\Rightarrow$ 連体修飾句+名詞句

ここで、もし名詞「小学校」が先に後ろの助詞「の」と結合すると、一つの連体修飾句が生成できる。また、これが先に前の形容詞と結合すると、新しい名詞句が生成できる。即ち、異なった構文規則を利用すると、その解析木の構造が異なる。このような曖昧性を解決するために、本システムでは解析過程を制御するための仮定を利用する。例えば、今の場合、名詞「小学校」に対して構文規則G1とG3が適用できるので、どちらを適用するかを制御するために、仮定@G1と仮定@G3を生成する。

@G1: 名詞「小学校」に対してG1を適用する。

@G3: 名詞「小学校」に対してG3を適用する。

これらの仮定を利用して、現在の推論環境Eから新しい子孫環境E1=E $\cup$ {@G1}とE2=E $\cup$ {@G3}が生成されたとする。異なった環境において、システムは名詞「小学校」に対して異なった構文規則を適用する。例えば、環境E1では、名詞「小学校」は助詞「の」と結合して連体修飾句を生成できるが、形容詞「小さい」と結合することはできない。

3.3 確からしさの設定

既に述べたように、PMEは仮定に与えられた確からしさによって処理の方向を誘導する。一方、環境の木によってシステムは全ての可能な探索方向を保持するので、確からしさの設定に誤りがあっても、処理が失敗するわけではない。但し、この場合、本当に曖昧な文に対して、確からしい解釈を見つけることは保証できない。確からしさをどのように設定するかは本研究の目的ではないが、システムを構築するためには、これを考えなければならない。ここで、本システムの確からしさの設定基準を簡単に述べる。

(1) 単語の語境界の曖昧性

この場合、一番長い単語を高い確からしさを与える。

(2) 同音異義の曖昧性

辞書で各々の可能性の確からしさを指定する。

(3) 意味表現の曖昧性

対象世界に関する知識や文脈情報などを利用して設定する。これを実現するため、我々後述のような意味概念の関連を表すネットワークを用意する。

(4) 構文構造の曖昧性

格関係などの構文的な知識を利用して設定する。

3.4 仮定の切り換えによる解釈の再構成

本システムは解析過程で作られた信念の全ては最終的に仮定或はその組合せに依存するため、仮定の切り換えによって信念の集合を容易に変更することができる。従って、ある解釈が拒否された場合、システムは新しい環境において解釈を再構成することは簡単である。これを示すために、次の例を見よう。

例3 台風ではしが折れた。 (S2)

だから木の枝は短い。 (S3)

ここで、単に(S2)だけを考えると、単語「はし」の同音多義の曖昧性のため、次のように

「台風で橋が折れた」 (I1)

「台風で端が折れた」 (I2)

二通りの解釈が可能である。ここで、システムはいま文(S2)の解釈として(I1)を出力したとする。しかし、文(S3)を(S2)の後続文とすると、解釈(I1)が文(S3)の理由にならないため、これを拒否すべきである。しかし、これは文(S2)に対する処理は全部無駄になったことではない。実際には、(S2)に関する結果の主な部分は新しい解釈を見つけるためにまた役に立つ。つまり、システムはこのような場合、文(S2)に対して再び初めから処理する必要はない。文(S2)の解釈結果を利用して、これを再解釈することができる。この実現方法として、我々に変数節点という特殊な節点を利用する。変数節点とはその“値”即ちその節点で表す信念が環境に依存するような節点である。環境が変わると、変数節点で表す信念も変わる。このような節点によって、もし解釈に対応する節点に変数節点に依存するならば、この解釈自身も環境によって変化する。従って、システムは推論環境の切り換えによってその前に作った解釈節点を新しい環境で再解釈することができる。

例えば、例3で単語「はし」からなる品詞をnoun-2とし、その概念節点はconc-2(橋)とconc-3(端)としよう。この曖昧性に応じてシステムは仮定@conc-2と@conc-3を生成する。次に、節点noun-2の概念を表す変数節点conc-xを作る。また、CMEに対して、次のような理由付けを与える。

noun-2, @conc-2 $\rightarrow$ conc-2

noun-2, @conc-3 $\rightarrow$ conc-3

conc-2 $\rightarrow$ conc-x

conc-3 $\rightarrow$ conc-x

こうすると、節点conc-xは仮定@conc-2を含んでいる環境でconc-2を表し、また、仮定@conc-3を含んでいる環境でconc-3を表すようになる。文(S2)の解析過程で作られた諸信念の依存関係を図5に示す。ここで、目標節点、即ち解釈に対応する節点はmean-3である。システ

ムはまず仮定@conc-2を含む環境で解釈(11)を見つけた。その後、文(S3)の処理によって解釈(11)が矛盾することがわかったので、システムは再び(S2)の処理を行う。このとき、システムはCMEによって現在の環境を削除し、PMEによって@conc-3を含む環境を選択する。図5から分かるように、新しい環境で目標節点mean-3がまた成立するので、システムはその依存関係を利用して、新しい解釈(12)を出力する。

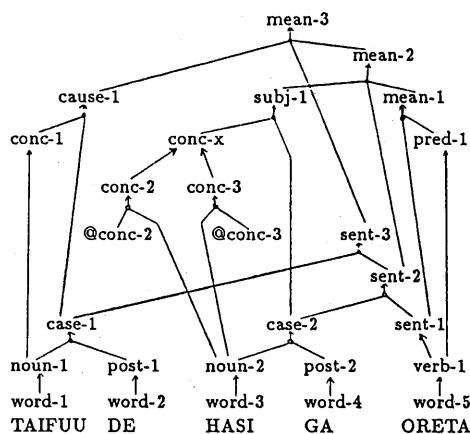


図5. 例3の処理における信念の依存関係

4. 実行例

この節で、一つの例を用いて、我々のシステムが解析における曖昧性への対処、文脈情報の利用、処理方向の誘導及び矛盾の解消などの具体的な動きを示す。

システムに対して次のような発話を与えられたとしよう。

例4. (S4) KA GI WO KA SI TE KU DA SA I
「かぎをかしてください」

発話の場所はある教室の事務室であるとし、発話者は教室の先生で、聞き手は事務室の係員とする。いま、対象世界には、「かぎ」は三つがあって、それぞれ図書室のかぎ(key1)、コピー室のかぎ(key2)、または会議室のかぎ(key3)であるとしよう。この文には「KAGI」がどのかぎを指すか言明していないため、意味概念の曖昧性も生じる。従って、この場合(S4)に対して唯一的な解釈を見つけないことができない。

しかし、このような曖昧な文に対して、もし先行文脈として、次のような発話

HO NN WO KO PI I SI TA I NO DE SU GA (S5)

「本をコピーしたいのですが」

があった場合、本システムは文(S4)に対して、「鍵」をkey2と解釈し、また、次のような発話

KA I GI WO HI RA KU NO DE SU GA (S6)

「会議を開くのですが」

があった場合、システムは「鍵」をkey3として解釈することができる。

ここで、先行文脈として(S5)があった場合のシステムの(S4)に対する解析の過程を示す。システムはまず環境の木を初期化し、ルート節点E0(E0={})を生成する。次いで、環境E0で入力した文字列から最初の単語を見いだす。文字列の先頭から二つの単語「か」と「かぎ」を発見し、単語の語境界の曖昧性が生じた。これに応じて、PSEはCMEによって次の仮定

@word-1: 最初の単語は「か」である

@word-2: 最初の単語は「かぎ」である

を生成する。そして、PSEは前述の確からしさの設定基準によって、生成された仮定に確からしさを与える。いまの場合、PSEは仮定@word-1に確からしさ1/3、仮定@word-2に確からしさ2/3を与える。すると、CMEはこの二つの新しい仮定を利用して、新しい推論環境E1(E1={@word-1}), E2(E2={@word-2})を生成し、環境の木を図6のように拡張する。

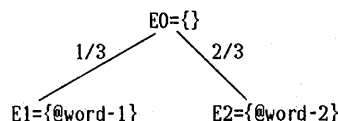


図6. 例4の処理の環境の木

(仮定@word-1, @word-2が生成された直後)

次に、PSEは仮定と節点の間の依存関係を表す理由付け

ch-1, @word-1 → word-1

ch-1, ch-2, cn-1, @word-2 → word-2

をCMEに与える。こうすると、環境E1において、節点word-1は(S4)の最初の単語と見なされ、環境E2において、word-2は(S4)の最初の単語と見なされる。ここで、ch-1, ch-2は文字KAとGIを表し、cn-1は文字節点ch-1とch-2が隣接することを表す。

次いで、PSEは確からしさ管理エンジンPMEを呼び出し、次の推論環境を選択する。PMEは各環境の確からしさを計算し、環境E2を選出し、制御をPSEにもどす。

処理は新しい環境E2に続ける。PSEは単語「かぎ」の認識を行って、節点noun-1を生成する。次に、noun-1の意味解析を行うとき、その意味概念の曖昧性が生じる。PSEは前の処理と同じように、まずCMEによって仮定

@key1: 名詞「かぎ」がkey1を指すと仮定する

@key2: 名詞「かぎ」がkey2を指すと仮定する、

@key3: 名詞「かぎ」がkey3を指すと仮定する、

を生成し、また、先行文脈情報を利用して、仮定に確からしさを与える。

我々が試作したシステムでは、文脈情報から仮定の

確からしさを決定するために、D. Waltzらのマクロ素性^[3]の概念を参照して、図7に示したような関連ネットワークを利用する。このネットワークの節点は意味概念或は事象を表し、節点の間の弧はその両端の意味概念或は事象に関連していることを意味する。

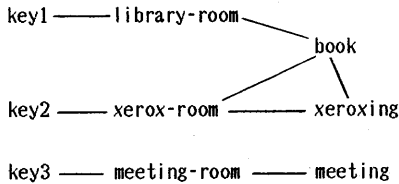


図7. 意味概念の関連ネットワーク

このような関連ネットワークを利用して、システムは先行文脈の意味概念(例えば、book, xeroxingなど)に関連の強いものに対して高い確からしさを与える。この例では、key2は最も確からしい。次はkey1である。key3は関連がない。従って、各仮定に与えられた確からしさは次のようになる。

@key1: 1/3
@key2: 1/2
@key3: 1/6

すると、CMEは環境の木を図8のように拡張する。また、環境E4($E4=\{\text{@word-2}, \text{@key2}\}$)が最も確からしい環境としてPMEによって選ばれる。その後の処理はこの環境で続行する。

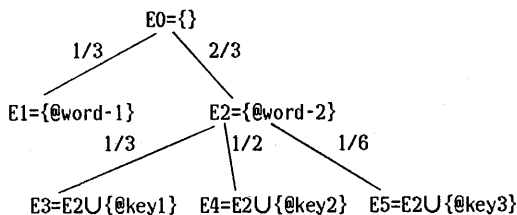


図8. 例4の処理の環境の木
(仮定@key1, @key2, @key3が生成された直後)

この環境において、システムはまず単語「を」を発見し、それに対する単語認識や構文解析などを行う。その後統単語として、「か」と「かしてください」の二つの可能性がある。前の語境界の曖昧性に対する対処と同じように、システムは仮定

@word-4: 単語「を」の後統単語は「か」であると仮定する。

@word-5: 単語「を」の後統単語は「かしてください」であると仮定する。

を作って、各々に確からしさを

@word-4: 1/8

@word-5: 7/8

を与える。これを利用して、CMEは環境の木を図9のように拡張する。処理は環境E7に移って続行される。

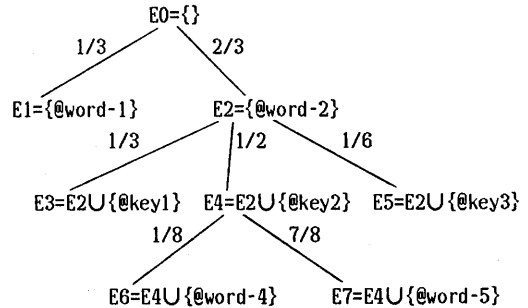


図9. 例4の処理の環境の木
(仮定@word-4, @word-5が生成された直後)

上で述べたような処理を(S4)の意味表現が得られるまで続ける。最後に、次のような意味解釈を出力する。

event = asking-for
actor = <発話者> (1)
object = key2

(S5)のような先行文脈がなくても、システムは対象世界においてコピー室がよく使われることを知っているならば、やはり最適な解が見つけれられる。例えば、予め次のような知識

「対象世界には、カギという言葉で図書室のかぎ(key1)を指す確率は0.3, コピー室のかぎ(key2)を指す確率は0.5, 会議室のかぎ(key3)を指す確率は0.2である。」

を与えるなら、システムは上の解析過程と全く同じように動作する。

上に示したように、CMEとPMEからなる統合パーサを用いた自然言語理解システムは、解析の過程で曖昧性に対する複数の可能性を全て探索することではなく、最も確からしい方向だけに向かって処理を進める。これによって、無駄な処理がかなり少なくなる。

一方、曖昧性に応じた全ての可能な方向はCMEによって保持されるので、矛盾が生じて、処理が崩れることはない。矛盾の解消や推論方向の切り換えは統合パーサによって容易に行われる。そして、新しい環境において、重複な処理は必要なく、CMEは自動的に前の処理結果を新しい環境に移る。例えば、上の例4において、もし出力した解釈が発話者によって拒否された場

合、システムは今の環境E7を矛盾した環境として削除し、確からしさを修正する。修正した後の環境の木は図10に示す。

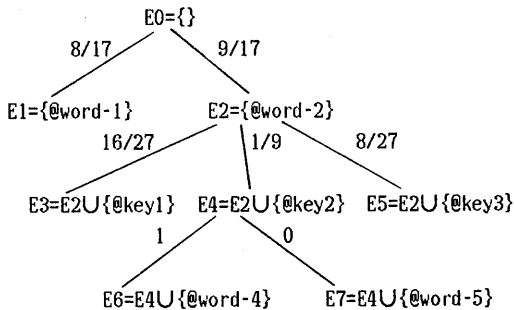


図10. 例4の処理の環境の木
(環境E7が矛盾したとき修正した直後)

次の環境として、PMEはE3を選択する。この環境において、システムは前に作られた仮定@word-4と@word-5を利用して、新しい子孫環境E8(=E3U{@word-4})とE9(=E3U{@word-5})を生成し、環境E9を次の推論環境として選択する。環境E9を生成するとともに、CMEが仮定@word-5に依存した解析結果も環境E9に移るので、矛盾が検出した前の処理結果の主な部分は環境E9で直ちに再利用できる。つまり、システムは単語「かぎ」の意味概念を"key1"に切り換えて、次のような新しい解釈

event = asking-for
actor = <発話者> (2)
object = key1

を出力する。

5. 終わりに

本稿では、統合的自然言語理解を効率的に実現するための新しいメカニズムを提案した。この方法では、統合的自然言語理解システムは統合パーサに基づいて構築される。統合パーサは整合性維持エンジンCMEと確からしさ管理エンジンPMEと二つのサブシステムからなる。

CMEはATMSの技法をベースにして、ATMSに対して推論環境の管理機能を付加することによって作成した。CM EはATMSの全ての利点を継承する一方、推論環境の管理機能によって、考慮する必要のない環境をはじめから排除して、無駄な処理を行わず、処理の効率の向上を図っている。一方、PMEは仮定に与えられた確からしさから一つの信念ネットワークを作り、確からしさの管理を行う。推論の方向を選択する必要が生じると、PMEは各々可能な方向の確からしさを計算し、処理の方向を誘導する。また、処理の進行に伴って、PMEは仮定の

確からしさを動的に修正する。

統合的自然言語理解システムは統合パーサを用いて、解析の過程において、各解析レベルの知識や解析の部分結果と情報を統合利用し、曖昧性をできるだけ早期に解決する一方、本当に曖昧な文に対しても、最適な解釈を出力することができる。

本研究で実現したシステムは統合パーサの効率を実証するために構成したものであるため、きわめて簡単な文法を利用した。今後の課題としては、さらに複雑な文を解析できるようにシステムを拡張することがあげられる。また、確からしさの設定も今後の課題の一つとなる。さらに、本研究で実現した統合パーサの基本的な考え方を人工知能処理の他の領域例えば音声認識、パターン認識などのような誤りを含んだ入力の場合に適用して、検討してみたい。

【参考文献】

- [1] Johan. de Kleer. An assumption-based TMS. Artificial Intelligence, 28:127-162, 1986.
- [2] Judea. Pearl. Distributed Revision of Composite Beliefs, Artificial Intelligence, 33:173-215, 1987.
- [3] D. Waltz and J. B. Pollack. Massively parallel parsing: a strongly interactive model of natural language interpretation. Cognitive Science, 9:51-74, 1985.
- [4] 劉, 西田, 堂下: ATMSの拡張とその統合的自然言語理解システムへの応用, 情報処理学会第36回全国大会, 4T-1, 1988.