

TM-1312

制約と分散協調に基づく自然言語理解システム
—異種分散協調問題解決系Heliosの応用として—

津田 宏、相場 亮

October, 1994

© Copyright 1994-10-6 ICOT, JAPAN ALL RIGHTS RESERVED

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03)3456-3191 ~ 5

Institute for New Generation Computer Technology

制約と分散協調に基づく自然言語理解システム

- 異種分散協調問題解決系 Helios の応用として -

Natural Language Understanding by Cooperative Agents in Helios.

津田 宏

相場 亮

Hiroshi TSUDA Akira AIBA

(財) 新世代コンピュータ技術開発機構第二研究部 (ICOT)

{tsuda, aiba}@icot.or.jp

概要

異種分散協調問題解決系 Helios の自然言語処理への応用として、複数のエージェントがローカルな交渉を通じて行なうシステムを考える。ここでは、自然言語理解における制約の領域の異種性と、処理の異種性に着目する。まず、前者の例として制約ベース文法への応用を考える。制約ベース文法は、素性構造と単一化制約により統一的に各種の言語現象を説明する。ここでは、文理解は制約解消、単一化、時制論理、包摂関係(概念の上下関係)など異種の制約からなる制約解消もしくは制約緩和のプロセスと見なすことができる。次に、後者の例としてガーデンパス文の理解や、構文/意味制約のインタラクションといった、人間の認知的な理解を説明するものとして、様々な処理を行なうエージェントの交渉によるシステムを提案する。

1 はじめに

ICOT で開発が進められている異種分散協調問題解決システム Helios[10, 12] はマルチエージェントシステム構築のためのツールであり、既存の様々な問題解決器に皮をかぶせてエージェント化し、それらを環境の中で互いに通信させることができる。本稿では、Helios を自然言語理解へ応用する。まず自然言語理解における2種類の異種性を考える。一つは、制約の種類

の異種性であり、もう一つは処理の異種性である。前者の例としては制約ベース文法が考えられる。制約ベースの自然言語理解は、単一化、時制論理、包摂関係(概念の上下関係)など異種の制約からなる制約解消もしくは制約緩和のプロセスと見なすことができる。こうしたシステムを見通しよく効率的に構築するには、各モジュールはなるべく既存の問題処理器を用い、それらになるべく自由に情報交換を行なえる枠組が望ましい。本稿では JPSG(日本語句構造文法)の、ランスやアスペクトの処理における制約の異種性を取り上げる。

後者の異種性は、ガーデンパス文や、構文・意味のインタラクション、非文の理解などの認知的な現象に見ることができる。ガーデンパス文は、文の理解において人間もバックトラックを行なう例として取り上げられる。この処理は、異種の処理を行なうエージェント間の交渉と見なせる。

2 自然言語処理における異種性

自然言語処理における異種性として、もっともナイーブに考えられるのは、形態素解析、構文解析、意味解析といった、処理の種類による伝統的なモジュール分類である。これらのモジュールを、そのまま線形につなげた自然言語処理システムが数多く作られた。線形のシステムは確かに効率の面では優れている。しかし、線形につなげたシステムでは、複数のモジュールにまたがるような情報の処理において、generate and test 型の単調な処理になりがちである。ところが、実際の人間の自然言語理解の過程は、処理が固定された情報の流れによるもので

はなく、多様な情報の流れに基づくものと考えられる。また、制約ベース文法のように、文法は宣言的な制約のみを記述し処理の方向性を既定しないような文法理論も提唱されている。

ただし、自然言語における「異種性」とは、このような処理の異種性だけではない。制約ベース文法は、自然言語における構文/意味などの情報を全て制約の形式で宣言的に定義する文法理論であり、自然言語処理の立場から見ても近年非常に重要になってきている。それによれば、制約という観点から見ると、自然言語処理は単一化、時制論理、包摂関係(概念の上下関係)など異種の制約からなる制約解消もしくは制約緩和のプロセスと見なすことができる。例えば、[6]において、Shieber は自然言語の次のような異種の制約を指摘している。

分野	事例	制約の種類
構文	efficient encoding coordination	term equation, term inequations, subsumption
意味	ellipsis	higher-order term eq.
語用	collaborative dialog	temporal constraint
	pronoun reference,	statistical constraint

3 異種制約の分散協調モデル

前節で述べたように、自然言語処理では制約の種類と処理の2種類の異種性を取り扱うことのできる枠組が求められている。その候補の一つとして、マルチエージェントシステム(MAS)が考えられる。MASは、ある程度独立した処理単位をエージェントとし、それらが通信により協調することで、問題を解いていくシステムである。

現在 ICOT では、MAS を実現するツールとして異種分散協調問題解決系 Helios を構築中である。Helios では、異種の制約解消系(実)は、それに皮をかぶせることでエージェントとすることができる。エージェントは、その中で解決できない部分問題を見つけると、それが所属する環境に質問する。環境は別

のエージェントを選択し、それに質問を転送する働きをする。また、エージェントを含んだ環境は、それに皮をかぶせることで(複合)エージェントとすることもできる。

ここで、各エージェントは次のような性質を持つと仮定する。逆に言うと、Helios のユーザは、エージェントがこのような動きをするように、問題解決器に多少の手を入れたり、皮プログラムを記述する必要がある。

- 与えられた問題に対して処理を行ない結果を返す
- 自分で解決できない部分問題を認識し環境に問い合わせる

以下、主に自然言語の制約の領域の異種性の例として制約ベース文法の JPSG の処理を、処理の異種性の統合の例としてガーデンパスのような複数の制約処理のインタラクションの処理を示す。

3.1 制約ベース文法の処理

3.1.1 制約ベース文法

HPSG (Head-driven PSG) や JPSG (Japanese PSG) をはじめとする制約ベース文法 (constraint-based grammar) [7] は、素性構造をノードに持つ句構造により定式化された文法である。ただし、句構造の部分はあくまでも骨格であり、ほとんどの文法情報は、句構造における素性構造間のローカルな制約という形で宣言的に規定される。制約ベース文法は、かつて単一化文法とも呼ばれていたように、制約として主として素性構造の間の単一化を用いて記述される。

ここでは、JPSG を取り上げる。JPSG は自由な語順などの特徴を持つ日本語向けに調整された制約ベース文法理論である。主辞 (Head) が右端に来る二分木の句構造規則を仮定し、文法情報は制約の形で記述される。筆者は、cu-Prolog により JPSG のパーザを作成し、素性構造の単一化制約を扱う CLP により自然言語の曖昧性解消が自然に実現できることを示した [8]。しかし、その後 JPSG の定式化が進むにつれ、特に意味論の制約として時間関係のような単一化以外の制約も必要となってきた [2]。

3.1.2 JPSG のテンス / アスペクトの取り扱い

JPSG [2] では、(1) のように、同じ動詞句が時間に関して何通りかの読みがあること、また動詞の種類によっては幾つかの読みは存在しないことを制約を用いて定式化している

- (1) a. 健は今一着物を着ている [基本視野]
- b. 健は今朝からずっとあの着物を着ている [結果視野]
- c. 健はあの着物を 3 年前に着ている [経験視野]

基本視野は動作の行為最中、結果視野は動作が完了した結果が継続している状態、経験視野は動作が終了元の状態に復帰した読みをそれぞれ表している。

これらの読みは、動詞の原辞書に与えられる時点素性 tmp と、それが実際の辞書において展開されて得られる視野素性 view とから説明される。原辞書では、tmp は時点の 3 つ組 < 動作開始時点, 完了時点, 現状復帰時点 > をとり、その値のバリエーションにより動詞を 8 つに分類している。view 素性は時点の 2 つ組 < 視野開始時点, 視野終了時点 > を取り、その値は tmp 素性との間の選言的な時制制約により規定されている。時点素性の値と制約との組合せにより、実際の辞書では高々 3 通りの読みが生じる。

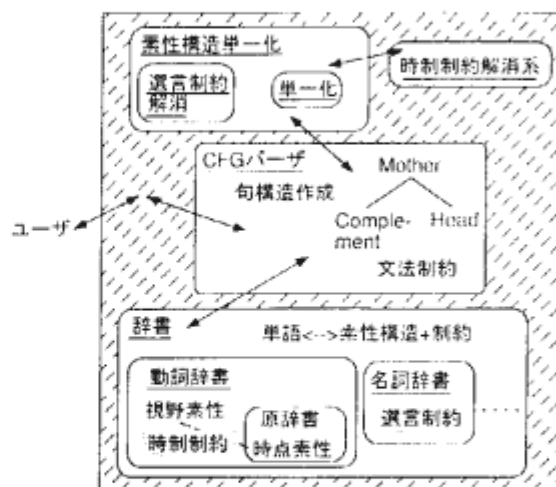


図 1: Helios 版 JPSG パーザの構成図

3.1.3 JPSG パーザ

さて、制約ベース文法の記述 / 実装システムを、このような文法の制約の拡張に対しても応用が利く方法で考えたい。まず、一つの実装言語で、それ自身を拡張していくというアプローチも考えられる。例えば、cu Prolog が Prolog に組合せ制約を付け加えたように、さらに時制制約の制約ソルバーを付け加えていくというアプローチである。しかし、これは文法の拡張の度に実装言語にも手を入れていかなければならない。また、新たに付け加わった制約領域の解決器が既存の場合、それと組み合わせることができれば現実的であろう。

また、[4] に見られるように、特に制約ベース文法の処理においては、句構造の作成部分の処理と、言語制約の処理のコントロールとが重要になってくる。このコントロールとは、句構造作成から制約解消に処理を渡すタイミングであり、ローカルな木構造ができるたびに渡すとか、木が全てでき上がってから制約を一気に処理する等である。制約の種類が増えると、さらにコントロールは多様なものが必要になる。このような、制約の処理順序のコントロールにも柔軟性のあるシステムでなければならない。

以上のような要件を満たすアプローチとして、分散協調モデルをとらえる。JPSG のパーザにおいては、エージェントとして、backbone である CFG パーザ、(選言的)素性構造単一化、時制制約処理系を考え、CFG パーザが随時制約処理系を呼び出して文の解析を行なう (図 1)。ユーザからの入力は、CFG パーザに送られ、句構造作成の段階に応じて、辞書、制約処理部分に部分問題が投げられる。制約の処理は、木のノードである素性構造の単一化および、そこで関連のある制約の制約解消により行なわれる。ここで、制約解消とは、制約をそれと等価な標準形に変換することである。標準形が存在しない場合は矛盾である。また、変数への値代入は変数に制約が最も強くかかった場合に他ならない。

各モジュールを流れるデータ構造は、

(素性構造, 素性構造の中の変数に関する制約)

のペアである。筆者は、[8] において constrained-PST を提唱し、選言的素性構造を (PST + ユーザ定義 Prolog 述語による

制約)として扱ったが、これはその制約部分の拡張となっている。これらの単一化は、(constrained-PSTと同じく)対応する素性構造の単一化およびそれにより変更を受けた素性を持つ制約の制約解消により行なわれる。Heliosでは異種の制約は別々のエージェントに送られ制約解消される。ユーザが最終的に受けとる答も、このペアにより表現される。曖昧性がなければ答の制約は空であるが、答えに曖昧性がある場合には制約が残る場合もある。

JPSGパーザにおいて、制約は動詞辞書の中の時制制約、曖昧語の辞書エントリに記述されている選言的な制約、句構造に付随する文法制約、といったように様々な場所で記述されている。これらが、解析処理の中で適当なタイミングで処理されることになる。自然言語解析のテーマの一つに曖昧性解消がある。この枠組では、種々の曖昧性は制約という形で統一的に扱われ、また曖昧性の解消も制約解消により自然に実現される。

なお、Heliosにおいては、これらのエージェントは必ずしも同一マシンにある必要はない。分散という面を強調するのであれば、特に辞書などは、ネットワーク上で公開されている既存の大規模辞書データベースを使うことも考えられる。

3.2 エージェントの協調に基づく自然言語処理

3.2.1 制約処理のインタラクション

ここでは、複数のエージェントがローカルな交渉を通じて協調することの自然言語処理への応用を考える。特に、認知的なものを含む複数の制約処理により説明される言語現象への応用について検討する。

まず、ガーデンパス文を取り上げる。これは、例えば

- (2) a. The cotton clothing is made of grows in Mississippi.
b. Have all the eggs broken {? or !}

のように[3]、多くの人が理解する際にバックトラックを必要とする文を指す。(2a)では、まずcotton clothingを構成素として読むが、growsで続かないため、前に戻ってclothing以下が関係代名詞である読みを見出すのである。ガーデンパスは、人間が文を理解する時に、全ての可能性を尽くすのではなく、部分的に可能性のありそうな読みの処理をしていく例証として用いられる。

同様に複数の処理モジュール間のインタラクションとして説明されるものは、数多い。例えば、[3]では、(3)のようなWII文の容認性の違いを、

- 構文制約: 動詞の直後の名詞句がIOになりやすい
- 意味制約: IOになりやすい順として
higher_animate > animate > inanimate

の二種類の制約の重要度の関係により説明している。

- (3) a. Which dragon did the knight give the boy?
b. *Which boy did the knight give the dragon?

次は、構文制約、意味制約、処理の嗜好とが入った例文である。

- (4) a. 背が高い女性を振った男子学生
b. 詰め襟の女性を振った男子学生

下線の関係詞の掛かり先は、処理的に直後の名詞に掛かるのがコストが低い(前者)。ただし、それが意味的におかしい場合(後者)には、コストが高い別の読みが行なわれる。

3.2.2 交渉に基づく処理モデル

前述のガーデンパス文などの処理について、従来よく行なわれてきた、形態素-構文解析-意味解析という直列につなげた枠組では、単純に行なうと前の方の処理の決定が次のモジュールで矛盾となれば、処理が失敗した場合に最初からやり直すことになる。また、前節のJPSGパーザのように、制約という形でバックするのは適当に曖昧性が解消されて全ての解の可能性は出るものの、ガーデンパス文の認知的なモデルとはならない。

ガーデンパス文理解は、構文解析の中の構文木作成を担当するエージェントと、補語(subcat)に関する制約処理を行なうエージェントとの相互作用とみなすことができる。ただし、多くの人がバックトラックするということは、構成要素作成の部分は全解を一度に得るのではなく、ある処理の嗜好性により順次解を出していくものと考えられる。前述の(3)や(4)の理解も同様に説明できる。

そこで、複数制約のインタラクションのある文の理解モデルとしては

- 各種制約に重要度があり、なるべく重要度の大きな制約を満たす読みが優先される
- 処理にコストがあり、なるべくコストの少ないような読みが優先される

を満足するものが直観的に必要であると思われる。

さて、MASにおいては複数のエージェントの間のインタラクションは、分散制約緩和[9]や交渉といった研究が行なわれている。特に関連が深い研究として[5]は、MASにおける(1:1の)エージェント間の交渉を、ゲーム理論の観点からいくつかの領域に分類している。各エージェントは、ゴールと効用(utility)と呼ばれる関数を持ち、自分の効用をできる限り大きくするように交渉を行なう。交渉の目的はindividual rational(効用が非負)で、パレート最適(一方の効用を上げようとする他方の効用が小さくなる)、またその中でも両者の効用の積を最大にするような状態に行くことである。

ガーデンパス文等の理解モデルとしては、[5]の領域の分類のうちWOD(Worth Oriented Domain)が適当である。WODはエージェントがゴールを緩和することができ、その緩和の程度を価値(worth)関数により定めている。WODにおける個々のエージェントの効用は以下の式で与えられる。

$$U_i = W_i(f) - c_i(f)$$

ここで、 U_i は状態 f におけるエージェント i の効用、 $W_i(f)$ は、状態 f におけるエージェント i の価値関数の値、 $c_i(f)$ はエージェント i が状態 f に至る処理のコストである。

$W_i(f)$ を満たされている制約の重要度の和、 $c_i(f)$ を文の処理と読みかえればガーデンパス文の理解モデルになる。価値関数として満足している制約の重要度の和、処理のコストとしては、木の深さ、省略語の補いの処理、slashの導入といった要素が考えられる。また、実際の交渉プロトコルや制約の緩和処理に関しては、分散制約緩和[9]が関連が深い。

こうした処理を行なうエージェントの要件としては、

- エージェントの持つ制約には重要度(価値)がついている
- エージェントは処理のコストを計算することができる
- 交渉は1:1で行なわれ、plan - anti_plan - accept というプランと効用の投げ合いによるプロトコルによる。



図 2: 複数の制約の interaction

である (図 2)。

例として、構文処理エージェントと意味処理エージェントとの交渉により、[3]における(3)の扱いを記述してみよう。boyがIOの読みをG1、dragonがIOの読みをG2とし、交渉は構文処理エージェントがプランを提示することから始まるとする。(3a)の場合には、 $\text{plan}(\{G1 > G2\}) - \text{accept}$ と、構文的に重要度の高いプランが意味的にも有利(重要度の高い制約を満たしている)であるため良い文ということでき合意が成立する。(3b)の場合には、 $\text{plan}(\{G2 > G1\}) - \text{anti-plan}(\{G1 > G2\}) - \text{anti-plan}(G2) - \text{accept}$ と、構文的に有利な読みが意味的には有利でなかった。意味が取れない文であるが、やむを得ず構文的に有利な読みを優先する。

4 おわりに

自然言語処理で、複数モジュールが並列に動作するシステムには、音声認識システムである Hearsay (II) が有名である。これは、音声入力、構文解析などのモジュールが、黒板を通じて結合している、いわゆる黒板モデルによる実装である[1]。また Ensemble [11] は、構文主導モジュールと意味主導モジュールとが黒板を介してつながっているアーキテクチャで、非文の解析を行なうシステムである。

これらは、全モジュールが共通に参照することのできる黒板というデータベースを使用している。この欠点として、例えば Hearsay-II において全処理時間の半分以上が、計算以外のオーバーヘッド(黒板の読み書き、黒板での同期など)である[1]というように、並列システムと大域性とは一般にあまり相性が良くないことを指摘しておきたい。対して、本稿では、各エージェントが、通信に基づいたローカルな交渉を行なうことで、全体の処理を行なうアプローチを示した。交渉モデルを選ぶことで大域性を排除し、WODモデルなどに当てはめることができた。

本稿では、工学的な応用として制約ベース文法のパーザ、認知的な現象の説明としてガーデンパス文など制約のインタラクションによる例文を取り上げた。また、このような自然言語処理における工学的・認知的に多様な情報の流れを実現することのできる枠組として Helios の有効性を述べた。

現在、簡単なプロトタイプを作成中である。従来、個別に作っていかねばならなかった通信部分およびデータ(型)変換の部分、Helios の機構を利用することで比較的簡単に実装することができている。今後個体関数の形や交渉のプロトコル・方略について、実験していく予定である。なお、本稿で述べた交渉によるモデルは Helios による制限ではない。Helios においても黒板エージェントを作ること、黒板モデルを実装することも可能である。Helios をブリットフォームとして、黒板モデルにおける黒板への通信の集中、および交渉モデルにおける交渉の通信オーバーヘッドなどの比較を行なうことを考えている。

また、取り上げたい自然言語のその他のトピックとしては、非文の処理、および人や状況による容認度の違いがある。前者は構文処理エージェントと意味処理エージェントの間の交渉としてかなり説明がつくのではないかとと思われる。容認度の違いは、エージェントの値関数が個人個人、また状況において形が変わることに相当する。

謝辞

有益なコメントを頂いた、横田 正氏をはじめとする ICOT の Helios グループおよび HKB-TG の諸氏に感謝いたします。

参考文献

- [1] R. D. Fennell and V. R. Lesser. Parallelism in Artificial Intelligence Problem Solving: A Case Study of Hearsay II. *IEEE Transaction on Computers*, C-26(2):98-111, 1977.
- [2] T. Gunji. A Proto-Lexical Analysis of Temporal Properties of Japanese Verbs. In B.S.Park, editor, *Linguistics Studies on Natural Language*, pp. 197-217. Hanshin Publishing, December 1992.
- [3] M. P. Marcus. *A Theory of Syntactic Recognition for Natural Language*. MIT Press, Cambridge:Mass, 1980.
- [4] J. T. Maxwell and R. M. Kaplan. The Interface between Phrasal and Functional Constraint. Workshop of ECAI92, February 1992.
- [5] J. S. Rosenschein and G. Zlotkin. *Rules of Encounter*. MIT Press, 1994.
- [6] S. M. Shieber. Constraints and Natural-Language Analysis. Tutorial in ILPS, October 1991.
- [7] S. M. Shieber. *Constraint-Based Grammar Formalisms*. MIT Press, A Bradford Book, 1992.
- [8] H. Tsuda. cu-Prolog for Constraint-Based Natural Language Processing. *IEICE Transactions on Information and Systems*, E77-D(2):171-180, February 1994. (anonymous FTP from ftp://ftp.icot.or.jp).
- [9] M. Yokoo. Constraint Relaxation in Distributed Constraint Satisfaction Problem. In *Proc. of ICTAI*, pp. 56-67, 1993.
- [10] 横田, 相場. マルチエージェントによる異種問題解決系の構築. In *MACC93*, 1993.
- [11] 島津明, 小暮潔, 中野幹生. 分散協調型言語理解モデルの頑健な解析への応用. 統合とマルチメディアワークショップ, June 1994.
- [12] 和住誠一郎, 柳川信晃, 木島秀治. 異種分散問題解決系 Helios におけるメッセージ通信の実現. ソフトウェア科学会第11回大会, 1994.