

制約依存文法に基づいた日本語解析支援システム

渡辺日出雄, 丸山宏

日本アイ・ビー・エム株式会社
東京基礎研究所

文の認識を整合ラベリング問題として定式化した文法である制約依存文法に基づいた日本語解析システムについて報告する。自然言語解析の主要な問題である曖昧さの扱いに関しては、組合せ的爆発の問題を回避するために同じ構造を共有するなどの手法が試みられているが、長距離の依存関係をうまく扱えないなどの問題がある。そこで我々は日本語の解析を整合ラベリング問題と見なすことにより全ての解釈の可能性を保持したデータ構造を作ることが出来、それに対して制約を当てはめていくことにより解釈を絞ることが可能であることを示す。さらに、この手法を基にしたユーザーと対話的に日本語の解析を行うシステムについて述べる。

A system for analyzing Japanese sentences interactively based on Constraint Dependency Grammar

Hideo WATANABE, Hiroshi MARUYAMA

IBM Research, Tokyo Research Laboratory, IBM Japan, Ltd.
5-19, Sanban-cho, Chiyoda-ku, Tokyo 102, Japan.

We describe a system for analyzing Japanese sentences interactively based on Constraint-Dependency Grammar which treats parsing sentences, as a consistent labeling problem. So far, the way to treat ambiguous sentences has been one of the biggest problems in natural language processing: Some methods have been developed to avoid combinatorial explosion, but these methods seem to have difficulty handling long distance dependencies. Therefore we have applied techniques for solving consistent labeling problems in sentence analysis, because its network data structure can compactly represent all possible modifier/modifiee ambiguities, and we can minimize the ambiguities by applying constraints to this network. We also describe a system using a constraint based parser, in which a user can add constraints interactively.

0.はじめに

今日まで自然言語処理が抱えてきた多くの問題点の内の一つとして、自然言語が本来持っている曖昧性のため解析に要する計算量が膨大になってしまふということを挙げる事が出来る。そこで、多くの自然言語処理システムにおいては、この曖昧さをいかに効率よく扱える枠組みを提供できるかが鍵となっている。この曖昧さを構文木のような明示的(外延的)構造により保持していたのではかなり複雑で長い文あるいは文脈を扱おうとする際には、その持つ曖昧さの組合せの爆発の問題を防ぐことが困難であり、計算量の観点から見ても効率が良いとは言えない。そこで我々は、この問題を予め全ての可能性を内包的に表現したデータ構造に対して制約を当てはめていくことで次第に解を絞りこんでゆくという方法により解決した。これは、日本語の解析の一つの整合ラベリング問題 [Montanari74]であると見做す事により、制約を制約伝播アルゴリズム [Mack77] [Mohr86]により高速に伝播させ、解空間を効率的に絞るというものである。この方法が今までの句構造文法に代表される解析木のよな外延的データ構造を用いる手法と異なるのは、

- 1) それらがある種の規則(文法)に従う構造を順次作っていき、すなわち、規則が正の例を与えているのに対し、規則(制約)を満たさないものを削っていく、すなわち、規則が負の例を与えているという点と、
- 2) 最終的に解析結果が必要となるまではその内包的表現のまま処理を続ける事ができる、すなわち、解析木の様な明示的に文を表すような構造は最後まで作る必要はない、という点である。

これは特に文脈処理の様文という単位がそれほど必要とならない場合に有効であると思われる。

最近この様な制約指向型の自然言語処理を目指す研究が [杉村88] [橋田88a] [橋田88b]らによっても行なわれている。これらによって述べられている思想と共通すると思われるが、我々は自然言語解析とは、全ての解釈の可能性を表現したデータ構造に対して構文的、意味的、文脈的などの各種のレベルの知識を適用して解釈を絞りこむことであると見做している。よって、全ての解釈の可能性を保持しているデータ構造として解析木を用いることはすでに述べたように組合せの爆発の問題等から適当ではなく、内包的に表現されたものでなければならぬ。この手法によれば、十分な制約が与えられれば最終的に正しい解釈が得られることは容易に推測され、もし得られないとすればそ

れは本質的に入力文(或は文脈)の中に正しい解釈を導く情報が足りない場合と見做す事が出来る。¹

本論文では、日本語の解析を整合ラベリング問題として定式化した試みについて述べる。1章においては計算量の観点から曖昧さの表現法について論じ、2章において整合ラベリング問題について解説をし、3章においては制約に基づく解析のための制約依存文法とそれの整合ラベリング問題としての定式化について簡単に説明し、4章でそれに基づいた日本語の解析方法について述べ、最後に5章においてこの解析方法を取り入れたことにより対話的に解析を行う事が出来るシステムについて述べる。

1. 曖昧さの表現法

今日までの句構造文法に代表される構文解析手法は曖昧さの扱い方に関しては、計算量の観点から考えるとかなりの問題を抱えている事がわかる。たとえば、英語の文における前置詞句の並びや日本語の文における「～の」の並びなどにおいては、単純に計算するとその部分の構成要素の長さの指数個よりも多い Catalan 数 [Knuth75] 個の曖昧さが生じる。 [Church75] この Catalan 数とは、1, 1, 2, 5, 14, 42, 132, 469, 1430, ... と続く数列であり、

$$Cat(n) = \frac{2n C_n}{(n+1)}$$

によって得られる。これは、 $n > 6$ において e^n よりも大きくなる。よって、構文木のような構造のまま全ての解釈の可能性を保持することが計算量の観点から困難であることが容易に推測される。

今日までの自然言語の解析手法においては曖昧さの取り扱い方に関して膨大な計算空間の必要性を回避するため解釈の可能性をバックして保持する手法が [Barton85] [Tomita87] [Rich87] [Kaplan73]らによって試みられてきた。これらはいずれもある部分列 α に複数の解釈が存在していた場合にそれを一つのノードで代表させるというものである。これらの問題点の一つは、部分列の間での解釈の依存関係を表現できないことである。たとえば、部分列 α に $\{a_1, a_2, a_3\}$ 、部分列 β に $\{b_1, b_2\}$ という解釈がある場合 $\{a_1, b_1\}$ という組合せが許されないということは表現できない。すなわち、チャートに代表される複数の解釈をバックした構造は、それを個々の解釈を表す構造に展開するこ

1 これは、[橋田88a][橋田88b]で述べられている部分情報問題に他ならない。

となしに曖昧性の解消を行うことが容易ではない。この様な長距離の依存関係を表現するという事は、構文解析のみならず意味・文脈解析においても重要な問題であると思われる。

以上のような考察から我々は、自然言語を解釈するプロセス及びそれが用いるデータ構造としては、次の要請を満たすものでなければならないと考えた。

1. 組合せの爆発を起こさない。
2. 長距離の依存関係を表現できる。
3. あらゆる種類の知識(制約)をそのデータ構造に対して適用可能である。

そこで我々は、整合ラベリング問題(Consistent Labeling Problem: CLP) [Montanari74]の枠組みを用いて文の解析を行うこととした。それは、データ構造としては、文節を一つのノードと見做し、それらに間に係受け関係のリンクが張られているネットワーク構造を用い、それぞれのリンク上に係受けに関する制約が記述されていて、その制約を制約伝播アルゴリズム [Mack77] [Mohr86]により高速に伝播させ解釈を絞ってゆくというものである。この解析モデルによれば、前に述べた3つの要請は、

1. あらかじめ全ての解釈の可能性を表現しているので組合せの爆発は起こらない、
 2. 任意の二つの文節の間で制約が記述できるので長距離の依存関係を表現できる、
 3. 制約としては構文的知識だけでなく意味的・文脈的知識も使うことができる、
- ことから全て満たされていることが判る。

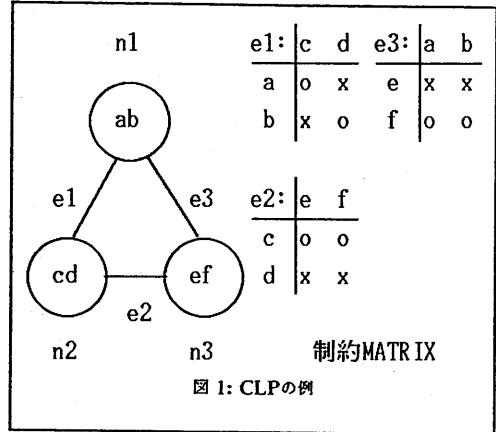
本章では、この整合ラベリング問題及び制約伝播アルゴリズムについて簡単に説明する。

2. 整合ラベリング問題

整合ラベリング問題²³とは、あるネットワークがあり、それぞれのノードは取りうるラベルの集合を持ち、それぞれのリンクは両端のノードのラベルの許される組合せを制約(関係マトリックス)として持つ場合に、その制約を満たすそれぞれのノードのラベルを決定する問題である。

- 2 整合ラベリング問題は、1972年に [Waltz75]により線面の認識問題を解くために提唱されて以来、 [Mack77]などにより高速な制約伝播アルゴリズムが提案され、今日までアルゴリズムの改良が続けられている。
- 3 制約により関係付けられた複数の変数のそれらの制約を満たすような値を求めるという制約充足問題 (Constraint Satisfaction Problem)の一種とも考えられる。

一つのCLPは、
 ノードの集合: $N = \{i, j, \dots\}$, $|N| = n$,
 ラベルの集合: $A = \{b, c, \dots\}$, $|A| = a$,
 リンクの集合: $E = \{(i, j) | (i, j) \text{は } N \times N \text{の要素}\}$,
 $|E| = e$,
 単項制約: $R1 = \{(i, b) | (i, b) \text{は } N \times A \text{の要素}\}$,
 二項制約: $R2 = \{(i, b, j, c) | (i, b, j, c) \text{は } N \times A \times N \times A \text{の要素}\}$
 で表現される。例を図1に示す。



このCLPのConsistency (制約を満足している状態)のレベルには以下の3種類がある。

- Node Consistent(NC) ...全てのノードに於いてそのノードの取りうるラベルに関する制約(単項制約)を満たしている状態。
- Arc Consistent(AC) ...全てのリンクに於てそのリンクの制約(二項制約)を満たしている状態。
- Path Consistent(PC) ...全てのパス(path)に於てそのパス内に解とならないラベルは存在しない状態。

ACよりもPCの方がConsistencyのレベルは高い。

制約伝播アルゴリズムとは、基本的にはそれぞれのリンクについて二項制約を満たさないラベルを取り除いて行くことによって、制約を伝播させるものである。このアルゴリズムの計算時間は、ACが $O(ea^2)$ [Mohr86], PCが $O(n^3a^3)$ [Han88]であることが示されている。

ただ、解を求めるということを考えた場合、本質的にこの様な制約ネットワークを満足するラベル

の集合すなわち解を数え上げることは NP完全問題であるから、制約伝播アルゴリズムを用いたからと言って助かるわけではない。しかし、このアルゴリズムを用いれば、その制約を満足しないことが判るラベルを取り除くこと(Filtering)が比較的高速に出来るので、十分な制約を与えることができれば最終的な解の数え上げの手間をかなり軽減することが可能となる。我々はこの性質を文の解析に用いたわけである。 [丸山88b][丸山88c]

また、ACまたはPCに於ての解の存在性が問題となるが、実際、ACまたはPCであって解が存在しないこともある。しかし、これはかなり特殊な問題の場合であり、我々の用いたモデルに於てはその様な状態にはならないのではないかと推測している。さらに、ネットワークがACであってPCでない状態は問題の性質にもよるがそれほど多くないと思われる。特に、我々のモデルの様に制約を一度だけ適用するのではなく、制約を次々に適用していった解を絞っていく場合には、計算コストのかかるPCの状態にまでする必要はなくACの状態ですら十分である。そこで我々はConsistencyのレベルとしては ACを用いている。

3.制約依存文法(Constraint Dependency Grammar)

整合ラベリング問題として日本語の解析を定式化するため [丸山88a]により制約依存文法 (Constraint Dependency Grammar: CDG)が提唱された。この章では、それを簡単に解説する。

アルファベット Σ 上の長さ n の文 S

$$S = w_1 w_2 \dots w_n \quad (w_i \in \Sigma)$$

があった時、文 S に於て各単語 w_i は k 個の異なる役割(role) r_1, r_2, \dots, r_k を果たすことができるものとする。

	w_1	w_2	...	w_n
r_1 -role	[]	[]		[]
r_2 -role	[]	[]		[]
.				
.				
r_k -role	[]			[]

図 2: 制約依存文法概念

各roleの取りうる値はラベル l と被修飾

句(dependent) d の直積 $\langle l, d \rangle$ となる (l はその修飾関係を表すシンボル、 d は被修飾語の位置を表す整数、ただし 0 は d が意味がないことを表す)。よって、一つの解釈とはそれぞれのroleに値を割当てたものである。

制約依存文法理論におけるある文法 G は、文 S が与えられた時に可能な割当て A の集合を規定する。ここで割当て A とは role からその値への関数である。 G は次の4つ組みで表される。

$$G = \langle \Sigma, R, L, C \rangle$$

ここで、

Σ は有限のアルファベット、

R はrole名の有限集合、

L は各ラベルがとりうる値の有限集合、

C は割当て A が満たすべき制約の有限集合、

である。

制約 C は次のような形式をしている。

$$P(r_1, r_2, \dots, r_p) \quad (\text{ここで } r_i \text{ は role})$$

role名の集合 R の要素数 $|R|$ を文法 G のdegree、制約 C における各項の変数の数 p の最大値を文法 G のarityと呼ぶ。

この制約依存文法を用いれば、最終的な解析結果は各語が他の語に対してどのような修飾関係を持つかを表すネットワークとなる。

3.1.整合ラベリング問題としての定式化

次に、この制約依存文法を整合ラベリング問題としてどのように定式化するかということが問題となる。制約依存文法は、各単語に割当てられる役割(role)をスロットと見做せば、各単語に幾つかのスロットを持ったフレームが付随していて、そのスロット間に制約があると考えることが出来る。よって、これを整合ラベリング問題として見ると、値に曖昧さがあるスロットがCLPのノードに対応し、それらの間の制約がCLPの制約と対応すると考えることが出来る。ただし、CLPでは制約は二項関係で表現されるから arity=2までの制約依存文法を整合ラベリング問題として定式化出来ることになる。

degree = k , arity = 2の制約依存文法は次のように整合ラベリング問題として定式化される。

ノードの集合: $N = \{role(i, r) \mid 1 \leq i \leq n, r \in R\}$

ラベルの集合: $A = \{\langle l, d \rangle \mid l \in L, 0 \leq d \leq n\}$

リンクの集合: $E = \{(x, y) \mid x \neq y, x, y \in N\}$

単項制約: $R_1 = \{\langle x, \langle l, d \rangle \rangle \mid x \in N, p_1(\langle x, \langle l, d \rangle \rangle) \text{ を満たす} \}$

二項制約:

$$R2 = \{ \langle \langle x1, \langle l1, d1 \rangle \rangle, \langle x2, \langle l2, d2 \rangle \rangle \rangle \mid x1, x2 \in N, p2(\langle x1, \langle l1, d1 \rangle \rangle, \langle x2, \langle l2, d2 \rangle \rangle) \text{ を満たす} \}$$

ACの計算時間は $O(ea^2)$ であるが、文法Gはdegreeがkだから、ノードの数は最大 $n \times k$ (nは入力系列を構成する単語の数)であるから、ラベルの数aは $|L| \times n$ 、リンクの数eは nkC_2 であり、結局 $O(nkC_2 \times (|L| \times n)^2)$ となる。

4. 制約依存文法による日本語の解析

前章で述べた制約依存文法を整合ラベリング問題として定式化したモデルに基づいた日本語の解析は以下ようになる。

基本的な単位(CLPのノード)は単語の代りに文節を用い、roleとしては、“係受け”を用いる、そして、roleの値としては $\langle d \rangle$ ($\langle l, d \rangle$ で $|L| = 1$ と考える)を用いている。制約はそれぞれの文節の係先に関するものとなる。このそれぞれの文節に関する係先候補と文節間の係先候補に関する制約マトリックスが可能な解釈を内包的に保持したデータ構造であり、今後これに対して各種の制約をかけていくことによって曖昧さを取り除いていくわけである。

以下に具体的な例で解析の進み方を説明する。初期状態は、「文節は自分の後ろの文節にしか係らない」という単項制約がかかっている状態である。図3にそれぞれの文節の係先と制約マトリックスの例を示す。これで判るように、係先が取り除かれるのはある制約マトリックスにおいてその係先に関する行要素あるいは列要素全てがxとなった場合である。単項制約とは強制的にそれをxにすることに対応する。

たとえば、「連用修飾句は用言に、連帯修飾句は体言にしか係らない」という単項制約を用いれば、図4のようなになる。さらに、「係受けは交差しない」という二項制約をかけると、「太郎は」と「おいしい」の間の制約マトリックスは図5となるから、「太郎は」の係先として「きれいな」が削られ、図6のようなになる。ここで曖昧さは「太郎は」の係先だけとなったので、次に「文頭の「は」格は文末の用言に修飾する」という制約をかけて係受け関係が決定される。

文節	1	2	3	4	5
	太郎は	おいしい	きれいな	水を	飲んだ。
係先	[2,3,4,5]	[3,4,5]	[4,5]	[5]	[0]

『太郎は』<->『おいしい』の間の制約マトリックス

おいしい 太郎は				
おいしい	X	0	0	0
きれいな	X	0	0	0
水を	X	0	0	0
飲んだ	X	0	0	0

図3: 初期状態

文節	1	2	3	4	5
	太郎は	おいしい	きれいな	水を	飲んだ。
係先	[2,3,5]	[4]	[4]	[5]	[0]

図4: 連体句・連用句の制約の後の状態

『太郎は』<->『おいしい』の間の制約マトリックス

おいしい 太郎は	
おいしい	0
きれいな	X
飲んだ	0

図5: 非交差制約

文節	1	2	3	4	5
	太郎は	おいしい	きれいな	水を	飲んだ。
係先	[2,5]	[4]	[4]	[5]	[0]

図6: 非交差制約の後の状態

この方式の計算時間は、文法Gはdegreeが1だから、ノードの数はn個、ラベルの数aは、 $|L|$ が1であることから高々n個、またリンクの数eは nC_2 であり、結局 $O(nC_2 \times n^2)$ であるが、これは $O(n^4)$ となり、かなり良いアルゴリズムであると言えるだろう。

5.対話的日本語解析システム

前章で述べたようなかなり高速な制約伝播エンジンがあれば、ユーザーからの指示を制約として制約ネットワークに加え制約伝播を繰り返すことはそれほど重大な問題とはならない。そこで我々はユーザーと対話的に解析を進めることができる日本語解析ワークベンチ (Japanese Analysis WorkBench: JAWB)を試作した。

今まで制約と一口に言ってきたが、この制約には、

- 絶対制約(Absolute Constraint) ...どのような状況でも必ず成立する制約
- 選好制約(Preferable Constraint) ...ある特定の状況に於て成立する制約

の2種類ある。実際に実用的なシステムを構築しようとする場合に、100%完全な制約を与えるのは現状では無理である。なぜならば、最終的に一つの解釈に決定される過程に於て選好制約の果たす役割は大きい。選好制約には副作用があるので、現在のところどのような条件の元でどのような選好制約を当てはめるべきかはまだ良く判っていないからである。⁴そこで我々はユーザーを制約知識源としてシステムのなかに取り込んだ方が得策であると考えた。特に、母国語に関してはそれを話すことが出来る者にとって係先を指定するのは容易なことであり特に学習を必要としないという利点がある。将来的にはこのユーザーの代りに何らかのエキスパート・システムが絶対・選好制約を与えることができれば良いということになる。

さて、ユーザーを制約知識源として取り込んだわけであるから、ユーザーが制約を与えやすい環境すなわち良いユーザーインターフェースを提供しなければならない。そこで、1)ユーザーはアクション・バーの中の制約規則セットのメニューの中から任意の制約規則を選んで実行することができる。2)制約規則の中には必ずしも常に成立するとは限らない制約(選好制約)があるから、制約伝播の結果を見て誤っていることが判れば、元の状態へ戻することもできる。3)さらに、制約規則セットの中に用意されている制約でも係受けを一意に定めることが出来なかった場合は、マウスを用いてカーソルがある文節の上にある時その係先の候補となっている文節が反転して見えるのでそこから係先を指定できるようにした。このユーザーによる係先の指定は、その文節に関する単項制約として扱われる。

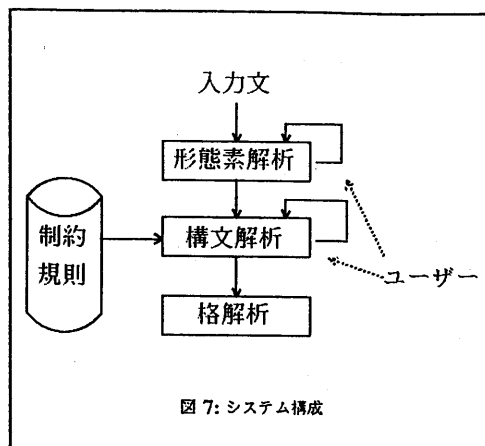


図 7: システム構成

図 7にシステム構成を示す。まず最初に入力文は形態素解析モジュールにより文節に区切られる。ここでもユーザーが単語及び文節の区切りに関して指定が可能である。形態素解析が終了すると単語及び文節が確定する。次に係受け解析を行う。ここでは、先に述べた様にユーザーとのインタラクションにより解析を進めていく。ここでは、それぞれの文節の係先が確定する。最後に、それぞれの用言に対して格要素を決定する処理を行う。

以下、係受け解析モジュールの動き方を例とともに説明する。図 8は係受け解析の画面のイメージであり、形態素解析を終えて係受けの解析を始めるところを示している。各文節の下に付いている数字は係先の候補数を表している。曖昧さ推定とあるのは単純に各文節ごとの係先候補数を掛け合わせたものであり、正確な曖昧さの数ではないが、曖昧さの数を推定する手助けとなる。この初期画面は「ある文節は自分の後ろの文節に修飾する」という「後方修飾」制約しかかかっていない状態であり、数十億の曖昧さがあることが判る。

そこで、次に「連体修飾句は体言に、連用修飾句は用言に修飾する」という「連体/連用修飾」制約をかけると、たとえば10個あった「厚生省は」の係先は用言である文節 4個に絞られる。次に、「係受けは交差しない」という「非交差」制約をかけると、たとえば、「さがすための」の係先は「ガイドラインづくりを」と「今年度から」であるが、「ガイドラインづくり」の係先が「始めた」となっているので、「さがすための」は「ガイドラインづくりを」に係ることになる。結果を図 9に示す。推定曖昧さはかなり減って

4 選好制約とは、[橋田88b]で言うところの部分情報問題である解析の処理に於て、完全な制約を与えることが出来ない時に、人間がある種の評価関数を用いて係先をとりあえず特定する時に用いられると考えることができる。

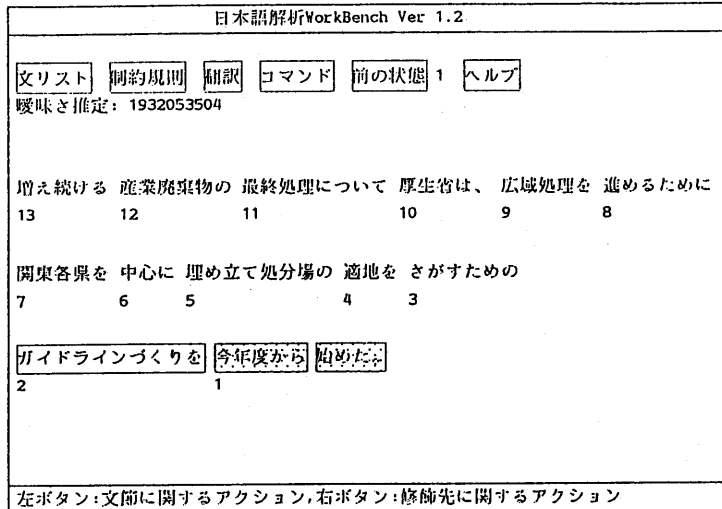


図 8: 初期画面

約5500くらいになる。ここで、次に「一つの用言

が同じ格要素を重複して取らない」という「同一格修飾」制約をかけると、「ガイドラインづくりを」が「始めた」に係っていることから、「適地を」の係先が「さがすための」になり、同様に「関東各県を」が「中心に」に、「広域処理を」が「進めるために」に係ることになる。次に「は格が一番最後の用言にかかりやすい」という制約をかけると、「厚生省は」が「始めた。」に係ることから、「非交差」制約により「最終処理について」は「始めた」に、「産業廃棄物の」は「最終処理について」に係ることになり、「増え続ける」は「中心に」と「ガイドラインづくりを」に係らないことになる。結果を図 9 に示す(*が付いているもの)。この段階で、推定曖昧さは24になる。

文節	係先
増え続ける	*産業廃棄物の *最終処理について 中心に ガイドラインづくりを
産業廃棄物の	*最終処理について 中心に ガイドラインづくりを
最終処理について	進めるために 中心に さがすための *始めた。
厚生省は、	進めるために 中心に さがすための *始めた。
広域処理を	*進めるために 中心に さがすための 始めた。
進めるために	*中心に *さがすための *始めた。
関東各県を	*中心に さがすための 始めた。
中心に	*さがすための *始めた。
埋め立て処分場建設の	*適地を *ガイドラインづくりを
適地を	*さがすための 始めた。
さがすための	*ガイドラインづくりを
ガイドラインづくりを	*始めた。
今年度から	*始めた。
始めた。	

図 9: 「連体/連用修飾」「非交差」制約後の係先と「同一格修飾」「は格」制約の後の係先 (*が付いているもの)

この様に制約をかけても曖昧さは全て解消されないでここからはそれぞれの文節について係先をユーザーが指定することになる。ここで「中心に」の係先を「さがすための」とすると、「埋め立て処分場建設の」の係先の候補が「適地を」と「ガイドラインづくりを」であるから、非交差制約より「適地を」に決定される。残りは、「増え続ける」と「進めるために」であるから、それぞれ「産業廃棄物の」と「始めた」を係先と指定すれば、終了となる。

現在この日本語解析ワークベンチは日英機械翻訳システム JETS[MaruN]の日本語解析モジュールとして、新聞記事を対象としたかなり大量の文章の解析のために使用される予定である。新聞記事にはかなり長い文が多く、現在の機械翻訳技術で全てを自動的に行うのは無理であろう。自動的に行おうとするあまり、何らかのヒューリス

ティックスにたよって誤った翻訳結果を出してしまうと、後編集に敵よせが来る。むしろ、人間を知識源としてシステムの中に取り込んで正解に近い訳を出した方が、後編集の手間まで考慮した場合に総合的なシステムとしての効率は良いのではないかと我々は考えている。もちろん、闇雲に全てをユーザーに任せるのではなく、あくまでもユーザーはシステムの補佐役として機能すべきである。この様な観点から、我々のシステムにおいては、ユーザーの介入があるのは、一番問題が多い係受け解析の部分に限定している。

7. おわりに

本論文により、我々は自然言語の持つ曖昧さから派生する解釈の組合せの爆発を防ぐために、解釈の可能性(曖昧さ)を内包的に保持したデータ構造に対して制約をかけることにより、次第に解釈を絞りこんで行く方法が有効であることを示した。そして、この解釈を絞りこんでゆく手法として、自然言語の解析を一つの整合ラベリング問題であると見做すことにより効率よく処理することが出来ることを示した。今後は、この手法を用いて意味・文脈解析までも含んだ自然言語解析のモデルを構築することが課題となる。

文献

- [Barton85] Barton, G. and Berwick, C.: Parsing with assertion sets and information monotonicity. Proc. of IJCAI'85 (1985).
- [Church75] Church, K. and Patil, R.: Coping with syntactic ambiguity or How to put the block in the box on the table. American Journal of Computational Linguistics, Vol.8, No.3-4 (1982).
- [Han88] Han, C. and Lee, C.: Comments on Mohr and Henderson's Path Consistency Algorithm. Artificial Intelligence 36 (1988) pp. 125-130.
- [橋田88a] 橋田浩一: 制約プログラミングにおけるヒューリスティックスの役割. 日本ソフトウェア学会第5回大会論文集 (September 1988).
- [橋田88b] 橋田浩一、杉村領一、田中裕一、向井国昭: DUALS-IIIにおける談話処理. 情報処理学会第37回全国大会講演論文集 (September 1988).
- [Kaplan73] Kaplan, R.M.: A general syntactic processor. in Kustin, R.(ed.): Natural Language Processing (1973).
- [Knuth75] Knuth, D.: The Art of Computer Programming, Vol. 1: Fundamental Algorithms. Addison-Wesley (1975).
- [Mack77] Mackworth, A.K.: Consistency in networks of relations. Artificial Intelligence 8 (1977) pp. 99-118.
- [丸山88a] 丸山宏: 制約依存文法理論 (Constraint Dependency Grammar Theory). TRL Research Report TR87-1021 (August 1988).
- [丸山88b] 丸山宏、渡辺日出雄: 制約伝播アルゴリズムを用いた対話的日本語解析システム. 日本ソフトウェア学会第5回大会論文集 (September 1988).
- [丸山88c] 丸山宏: 動的整合ラベリング問題とその自然言語への応用. 日本ソフトウェア学会 知識プログラミング研究会資料 (December 1988).
- [MaruN] Maruyama, N., Morohashi, M., Umeda, S., and Sumita, E.: A Japanese sentence analyzer. IBM Journal of Research and Development, Vol.32, No.2 (March 1988) pp. 238-250.
- [Mohr86] Mohr, R. and Henderson, T.C.: Arc and Path Consistency Revisited. Artificial Intelligence 28 (1986) pp. 223-233.
- [Montanari74] Montanari, U.: Networks of Constraints: Fundamental Properties and Applications to Picture Processing. Inform. Sci. 7 (1974) pp. 95-132.
- [Rich87] Rich, E. et. al.: Ambiguity procrastination. Proc. of AAAI'87 (1987).
- [杉村88] 杉村領一: 論理型文法における制約解析. 情報処理学会NLWG67-2 (July 1988).
- [Tomita87] Tomita, M.: An efficient augmented-context-free parsing algorithm. Computational Linguistics Vol.13 (1987)
- [Waltz75] Waltz, D.: Understanding Line Drawings of Scenes with Shadows. in Winston, P.H.(ed): The Psychology of Computer Vision (1975).

5 『始めた』に係るのが正しいという解釈もあるがこの文だけでは正確に決定できない。