自然言語処理 63－3
（１９８７．９．２５）

拡張ＣＦＧを用いた日本語構文解析

隅田　英一郎[1)]
丸山　直子[2)]

[1)]日本アイ・ビー・エム株式会社　東京基礎研究所
[2)]東京女子大学　文理学部・日本文学科

様々な自然言語処理において構文解析プログラムは、必須の要素である。本稿では、広範な言語現象に対応する日本語構文解析プログラムの設計と実現について報告する。構文解析プログラムにおける分野依存性を分析し、機械翻訳用にシステムを作成すれば、分野依存性を解消できることを論じる。この議論を例証するために、曖昧な解釈に優先度与える方法と法情報の処理について詳述する。本システムは拡張ＣＦＧで記述されている。

A Japanese syntactic analysis using an augmented CFG

Eiichiro SUMITA[1)]
Naoko MARUYAMA[2)]

[1)]Tokyo Research Laboratory, IBM Japan Ltd.
5-19, Sanbancho, Chyoda-ku, Tokyo 102, Japan
[2)]Department of Japanese Literature, Tokyo Woman's Christian University
2-6-1, zenpukuji, suginami-ku, Tokyo 167, Japan

Syntactic analysis is necessary for any kind of natural language application. This paper presents the design and implementation of a broad - coverage Japanese syntactic analyzer. The syntactic analyzer is written in an augmented CFG. After analyzing the problems of domain-dependency in syntactic analysis, we argue that by choosing machine translation as the application area, we can a build a domain-independent syntactic analyzer. In order to make the above clear we discuss the preference attachment for amabiguous interpretations, the handling of modal information.

1．はじめに

構文解析は、機械翻訳・質問応答・情報検索など種々の自然言語処理システムの基幹となる技術である。構文解析プログラムの品質がシステム全体の能力を決定するからである。

近年、多くの構文解析プログラムが実用化されている[1)2)3)4)]。しかしながら、従来の多くの構文解析プログラムは、処理内容・対象分野に依存した部分と一般的部分の分離が不十分であった。システムに必要な知識には、言語自体の一般的知識と処理や対象分野向きの特殊な知識とがある。複数システムの開発を考えると、一般的な部分と特殊な部分を明確に分離し、広い言語現象を扱えて、複数の後続する処理で共通に使える構文解析プログラムを作成する意義は大きい。

我々の目的は、広い言語現象に対応した日本語解析システムを実現することである。本稿では「機械翻訳用に日本語解析システムを作成すれば、この目的を達成でき、他のシステムにも利用できる。逆に他の処理用に作っても、機械翻訳への応用は困難である」ことを論ずる。以下、２節で、設計方針について、３節で曖昧さの取扱いについて、４節で法情報の取扱いについて、５節で出力構造について述べる。

2．設計の基本方針

２．１　処理・分野依存性について以下の５つの側面に分けて、従来の問題点と本システムの方針について議論する。

（１）受理する構文の制限　対象分野に出現しない構文を無視することがよく行なわれる。例えば、文献抄録には、命令文や疑問文は出にくいので無視するなど。しかし、機械翻訳システムでは、本来入力を制限できず、従って最も広く言語現象に対処する必要があるので、２．３節に示すように、本システムでは基本的に構文を制限しない文法をベースにし、入力サンプルも偏向しないようにしている。

（２）システムの意味的拘束　構文的曖昧さの解消のために後続する処理にとって意味があるかないか検査することは標準的な方法の一つである。例えば、ＤＢの質問応答では検索可能か検査することである。「最も安い部品の工場」という名詞句は、納入業者に関するＤＢなら「（最も安い部品）の工場」と解釈し、不動産に関するＤＢなら「最も安い（部品の工場）」と解釈するなど。本システムではこのようなことは、２．２節で示すように、構文解析プログラムの範囲外としている。すなわち、全ての可能な解釈を出力し、最終的な選択は後続の処理に委ねている。

（３）分野固有の構文パターン　構文的曖昧さの解消のために特定の分野固有の構文パターンを利用することがある。例えば動詞（Ｖ）の係受けを考えよう。「～V_1場合～V_2～V_3」というパターンでは、一般にはV_1の係先はV_2とV_3の２つの可能性があるが、マニュアルにおいてはV_1はV_3に係るとするなど。このようなことは実際的機械翻訳では特に重要なことである。しかし本システムでは、一般的な文法現象の処理を明確に分離できると考えて、上の（２）と同じように、構文解析プログラムの範囲外としている。

（４）深い知識　一般に構文解析プログラムで利用する知識には、形態素情報、名詞や動詞の意味分類、結合価・深層格フレーム、スクリプトなどがある。容易に獲得できない「深い」知識をつかえば、処理・分野に依存することになる。

機械翻訳のように大量の語彙が必要な分野で深い知識を使うのは困難である。本システムでは３節，４節で更に論じるように、形態素情報、名詞や動詞の意味分類、結合価フレームなどの容易に大量に集められる「浅い」知識だけを使って、一般性を保持しようとしている。

（５）出力する情報　論理式やＣＤ表現等は、ある特定の約束ごとの下での理解の表現であり、処理に強く依存する。これだけでは機械翻訳に必要な表層の情報を欠いていて役に立たない。５節に示すように、本システムの出力の依存構造は、機械翻訳に直接利用できるし、より深い情報への有効な出発点としても機能する。

要約すると、機械翻訳では１）構文を制限できない、２）システム固有の意味制約がない、３）分野固有の構文パターンの使用は分離できる、４）深い知識が使えない、５）機械翻訳用の出力である依存構造は他のシステムへの自然な入力になる。従って、機械翻訳用に構文解析プログラムを作成すれば、広い言語現象に対応した一般的構文解析プログラムになり、他の処理システムへの応用が可能である。しかし、逆に他の処理用に作っても、機械翻訳への応用は困難である。

２．２　構文解析プログラムの位置付け

図１に全体の構成を示す。これから分るように、構文解析プログラムは形態素解析された入力文を受取り、可能な全ての解釈を優先度のついた依存構造の集合として後続する処理システムにわたす。各処理システムは、システム固有の意味的制約、分野固有の構文パターン、深い知識を使って、更に曖昧さを解消する。

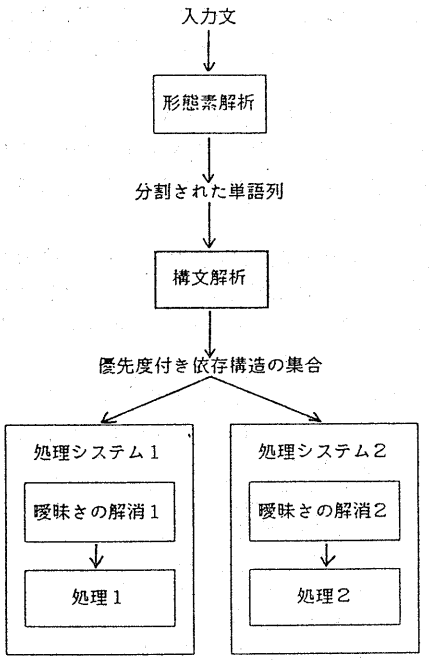

図１　構文解析の位置付け

２．３　基礎文法

言語学でいう文法には２種類ある。言語現象の記述を目的とする個別文法、言語現象を説明する原理を求める普遍文法とである。

日本語に対して、両文法とも多くの研究があるが、計算機上で稼働する構文解析プログラムを作るという目的には十分でない。個別文法は普通、微細な現象を説明していたり、そうでなくても十分形式化されていないことが多い。普遍文法は本来あらゆる文法現象を説明することを目的としているのに、却って説明できる文法現象が限定されているのが現状である。

現実的解決策としては、比較的形式化された個別文法を、基礎文法に選定し、これを現実の偏向していない入力文で実験しながら、拡張していく方法がある。この方法で開発された英文解析プログラムシステムにＰＥＧ[5]がある。ＰＥＧの一般性は英文校正システムＣＲＩＴＩＱＵＥ[6]、英日機械翻訳システムＳＨＡＬＴ[7]の入力部として使われて確かめられている。

我々の採用した基礎文法は、水谷文法[8]である。水谷文法は、言語学者によってＣＦＧで書かれた日本語文法の一つであり、構文的情報のみを用い、できるだけ多くの言語現象を取り扱うことを目的にしている。その結果、弱い文法になっていて、非文を受理したり、非常に多くの曖昧な解釈を出力する。

この欠点を解消するために、我々は拡張ＣＦＧのfeature systemを利用している。本システムでは２種類のfeatureを用いている。非終端記号を細分類するものと意味処理のためのものとである。前者を使って例えば、無題述態句、アル述態句、主抜き述態句を句（＋無題）句（＋アル）句（＋主抜き）などと表現する。後者を使って、名詞意味素性、結合価フレームなど辞書的なものと、埋め込文や法情報などの解析結果とを表現する。

２．４　記述言語

次の理由で拡張ＣＦＧを採用した。

（１）基礎文法が、ＣＦＧで記述されている。

（2）手続き的文法と異なり、明晰性が高く、大規模な文法の開発に向いている。

（3）拡張ＣＦＧで書かれた文法を効率的に実行する計算機構がよく研究されている[9)10)11)]。

（4）後述の例からも分るように、日本語の語順の任意性や詳細な意味解釈規則も容易に記述できる[5)]。

我々は拡張ＣＦＧの一つＰＬＮＬＰ[11)]を使っている。ＰＬＮＬＰはＰＥＧの記述言語である。文法規則は、Bottom-up parallelに実行される。文法規則の例を図２にしめす。

```
TAIRG KAKUHJ
  -> KAKUYOUSO(=TAIRG,CASE=CASE(KAKUHJ),
          TOPIC=TOPIC(KAKUHJ),KOOU=KOOU(KAKUHJ),
          PSMODS=PSMODS...KAKUHJ)
YOURG -> JUSSO(=YOURG,HINSISEI='YOU')
KAKUYOUSO(-OAI,-NO,SF,CASE)
  JUSSO(-NO,CASES0,CASE(KAKUYOUSO).ISIN.CASES0,
      CASE(KAKUYOUSO).NOTIN.CASES,
      <@CASE(KAKUYOUSO).EQ.'DIV'|
      @CASE(KAKUYOUSO).ISIN.SF(KAKUYOUSO)>)
  -> JUSSO(-NAI,CASES=LISTIFY<CASE(KAKUYOUSO)>...CASES,
        PRMODS=KAKUYOUSO...PRMODS,N=N(KAKUYOUSO)+N+4
        <TOPIC(KAKUYOUSO),+OAI>)
```

図２　文法規則の例

３．曖昧さの取扱い

自然言語処理では、曖昧さの解消は重要な問題である。本システムのような一般的な構文解析プログラムでは深い知識を使うことはできないが、浅い知識のみを使って出来るだけ曖昧さの数を抑え、かつ残った解釈に対して優先度を付与して、文脈が指示しないかぎり最も自然な解釈が選択できるようにすることを目指す。

格要素と述語の係受けは日本語の曖昧さの重要な原因の一つである。このための標準的方法では、格文法の考え方に従って述語と格要素の意味的整合性を調べる。

しかしながら格文法には、深層格の種類、判断基準等に安定したものがなく[12)]、解析に必要とされる知識を大量に矛盾なく収集することが非常に困難である。

我々は、格文法の代りに結合価文法[13)]を採用した。結合価は表層格と名詞の構文的・意味的素性のパターンで、深層格を用いないところに特長がある（図３）。現状では約７０００語に対して人手で結合価パターンを作成したが、主観的な作業ではないので、辞書の拡張時には機械的に作成することを検討している。また我々の名詞の意味素性は図４に示すように単純であるからこの付与作業も機械化を検討している。

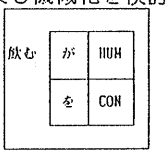

図３　結合価パターン

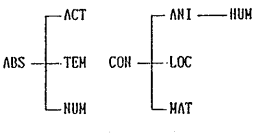

図４　名詞意味素性

結合価パターンを用い、次の手続きで意味的整合性を検査し、同時にその結合の優先度Ｎを計算する。

１）構文木の全ての節点のＮを０にする。

２）ＮはBottom-upに全ての子節点のＮの和を親節点に渡して伝搬する。構文木の根のＮがその構文木の優先度になる。

３）述語と格要素の結合を処理する時に述語の期待されるスロットの条件を満たすか否か検査される。条件を満足する場合は、図５に示された点が述語のＮに加点される。

実際には２）の所で、格助詞が明示されていない時の推定や態による結合価パターンの変換などの複雑な処理が同時に行なわれる。

	Sm＝Sp	Sm¬＝Sp	Sm＝Unknown or Sp＝Unknown*
Cm＝Cp	＋4	＋1	＋2
Cm＝Unknown**	＋2	＋0	＋1

Sm ： 格要素の意味素性　　Cm ： 格要素の表層格　　*)　辞書に指定されていない
Sp ： 述語の意味素性制限　Cp ： 述語の表層格制限　　**)　表層格が明示されていない

図５　優先度の加点

この方法で、図６の例にあるように文脈が特に示唆しない限り自然な構文木に最も高い優先度を与えることができる。

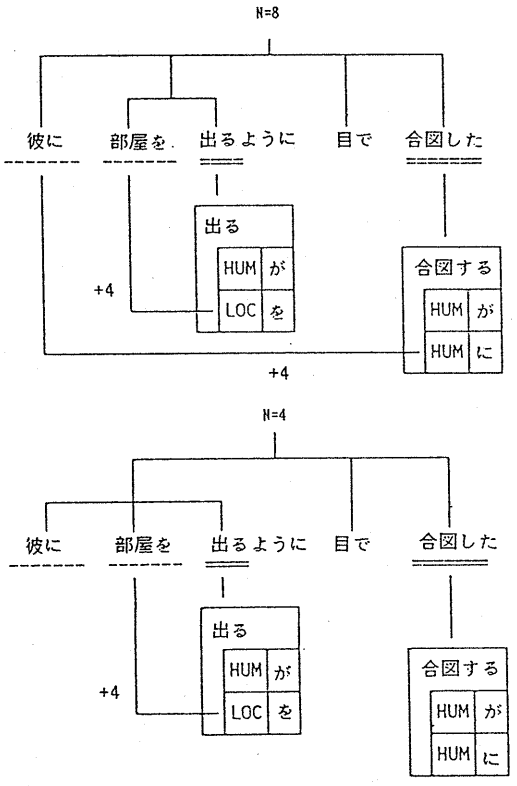

図６　優先度付与の例

この優先度の与え方では全ての曖昧さは扱えないので、他の評価基準との結合も含めて、改良の途中であるが、現状でも十分有用であることが分った。

図７に２２５文で実験した結果を示す。

適中率は曖昧な文に対して本方法が有効に働く、つまり最も自然な構文木に最高の優先度を与えた百分率である。この方法は述語が多いほど、従って、より長くより曖昧な文ほど有効なことが図から読み取れる。

減少率はこの方法が正しく働いた場合に最高点の構文木の全構文木に占める百分率である。複数の構文木が同じ優先度を得ることがあるが、この方法だけで曖昧さが半減できることが図から分る。

本稿の方法では、比喩、否定などの場合に高い優先度を与えることができないという限界がある。

例　この車はよくガソリンを食う
　　パソコンはビールを飲まない

４．法情報の取扱い

文の意味には主に助動詞が関係する法情報と呼ばれるものがある。例えば次の例文では「ている」は「進行」、「だろう」は「推量」の意味を担っている。

雨 が 降っ ている だろう

	文の数	曖昧な文の数	適中率	減少率
Total	225	97	41.2	48.4
1-pred*	116	17	17.6	50.0
2-pred*	88	60	38.3	46.8
3-pred*	18	17	64.7	45.7
4-pred*	3	3	100.0	58.3

*)　１文当りの述語の数
**) 文は「英語表現辞典」朝日出版社による

図７　優先度付与の効果

法情報は述語のfeatureとして表される。

法情報は主に助動詞とそれと結合する述語の組合せによって決定される。そこで各助動詞ごとに、法情報の解釈規則を対応させた。

ここでは法情報の一つアスペクトについて述べる[14]。

図８の「ている」の翻訳を考える。日本語では同じ表層のアスペクト「ている」であっても、英語では一つ一つ異なる表層のアスペクトになっている。何らかの意味、例えば「状態」、「進行」、「結果の残存」、「経験」を利用しなければ、翻訳が困難である[15]。

私は彼を知っている
I know him
彼は走っている
He is running
彼女は結婚している
She is married
彼は犯人を目撃している
He has seen the criminal

図８　「ている」の翻訳

我々はアスペクトの解釈規則を下のように定式化した。

動詞の意味分類
　－＞動詞の意味分類ｘアスペクト

「ている」のための規則を図９に示す。
一行目は動詞と「ている」が結合したときの全体の意味分類の変化を示している。２行目の「ｘ」は結合が不可能なことを示している。３行目以降は動詞の意味分類毎に決定されるアスペクトを示している。右辺の中の「＞」は読みの優先を示している。

```
                    ==> +STATE,+CONT
+STATE,+CONT        --> X
+STATE,-CONT        --> 状態
+PSYCH              --> 状態
+CH,-PSYCH,-PROG    --> 結果の残存 > 経験，反復
+CH,+PROG           --> 結果の残存 > 進行，経験，反復
+ACT,+CONT,-PSYCH   --> 進行 > 経験，反復
+ACT,-CONT          --> 経験 > 反復
```

図９　「ている」の解釈規則

動詞の分類を図１０に示す。この分類のための手続きは、既に定式化されており、７０００語に対して作業が済んでいる。

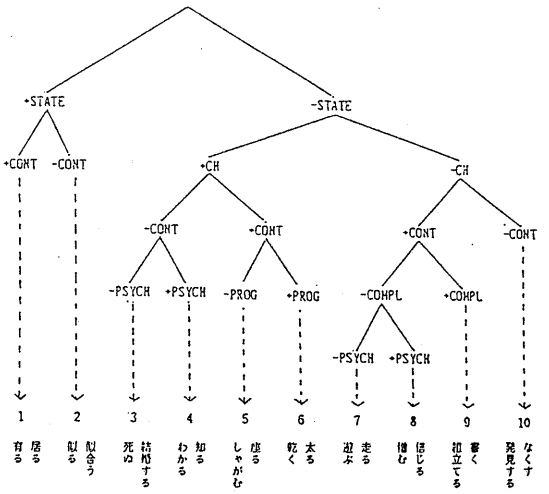

図１０　動詞の意味分類

この枠組は、また助動詞Ａと動詞Ｖの係受けの決定に役立つ。V_1～V_2ＡというパターンでV_1とＡの結合が不可能な場合、ＡのスコープをV_2に限定できる。

５．出力記述

我々の出力記述は依存構造である。
依存構造では単語とその依存関係のみを表示する。

節点となるのは内容語だけである。機能語等種々の情報は節点のFEATUREとして埋めこまれる。

句構造は、本来、規則の適用の記録に過ぎず、

自然言語の構造としては不自然であったり、非常に深い構造になって必要な情報のアクセスが非効率的になるなど、自然言語処理に不適当なことが多い[5]。

論理式やＣＤ表現等は、ある特定の約束ごとの下での理解の表現であり、処理に強く依存する。これだけでは機械翻訳に必要な表層の情報を欠いていて役に立たない[6]。

我々の採用した依存構造は、上記の短所を持たない上に、更に以下の長所を持つ。

（１）依存構造は語順の任意性を表現するのに適している。

（２）依存構造は、トランスファ方式の機械翻訳に向いている。日本語と英語のようにかなり構文の異なる言語対では、品詞変換とそれに伴う構造変換が頻繁に必要になる。

図１１の例文で(j)の文を品詞を変えずに単語を置き換えて得られるのは(*)のような非文であり、正しい翻訳を得るためには以下のように３つの単語の品詞を変換し、大きく構造を変える必要がある。句構造と異なり、依存構造ではこれを容易に実現できる（図１２）。

(j)　今日の暑さは格別だ

(*)　Heat of today is extreme.

(e)　It is extremely hot today.

図１１　品詞変換の例

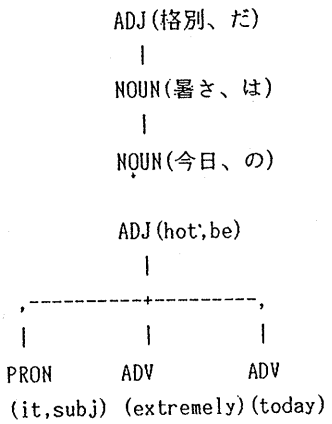

図１２　日英の依存構造

今日　->today　　(noun->adv)

暑さ　->hot　　(noun->adv)

格別　->extremely　(adj->adv)

（３）依存構造から、より抽象的な表現に変換できることは明らかだからここでは依存構造が直接役に立つことを示す。例えば、文脈処理においては、文脈中の部分構造が互いに似ていることを検出することが重要である。

図１３の例で動詞句「I come back to Japan」と名詞句「my arrival at Narita」が意味的に類似していることは、図１４の依存構造で、品詞を無視したときに、線で囲まれた部分が互いに構文的に類似していて、各要素が互いに意味的に類似していることから分る。

I came back to Japan this morning.

I informed her of my arrival at Narita.

図１３　文脈の例

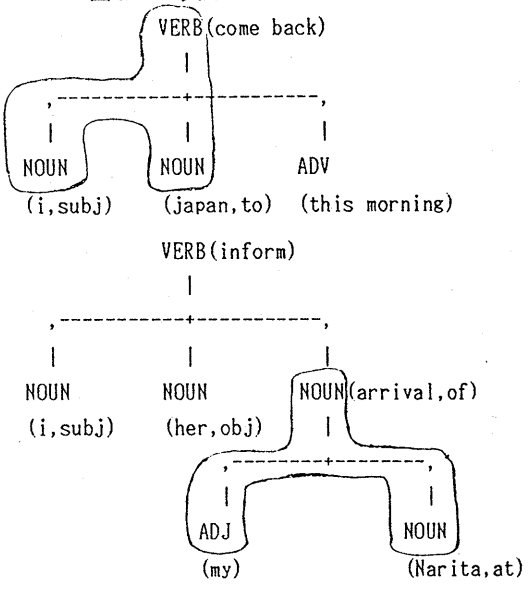

図１４　文脈の依存構造

おわりに

この研究は１９８５年５月に始まり、現在、規則数４１５である。この構文解析プログラムは、ＩＢＭの日英機械翻訳と自然言語インターフェースの研究[17]で使用されている。

今後の課題としては各規則の条件を厳しくし、曖昧な解釈の個数を減らすこと、優先度の規則を洗練することなどがある。

謝辞

この研究は、多くの人の協力に依っている。記して、感謝の意を表したい。

George HEIDORN,Norman HAAS(PLNLP漢字化), Martin MIKELSONS(LISP/VM漢字化),Karen JENSEN, 諸橋正幸,山内智恵子,鈴木恵美子,北村博,丸山宏, 堤豊,広瀬京子(有益な議論)。

参考文献

1)M. Nagao et.al., 1985,"The Japanese Government Project for Machine Translation",Computational Lingusitics vol.11, no.2-3

2)H.Nomura et.al.,1986,"Translation by Understanding:A Machine Translation System LUTE", Proc. of Coling 86

3)藤崎哲之助他, 1979, "データベース照会システム『ヤチマタ』と名詞句データ模型", 情報処理論文誌, vol.20 no.1

4)日高達, 1983, "格文法による日本語の構文解析", 情報処理学会ＮＬシンポジウム論文集

5)K. Jensen, 1986, "PEG 1986:A Broad-coverage Computational Syntax of English"(manuscript)

6)S.D.Richardson,1985."Enhansed Text Critiquing Using a Natural Language Parser."IBM Research Report RC 11332, Yorktown Heights, NY.

7)T. Tsutsumi,1986,"A Prototype English Japanese Translation System For Translating IBM Computer manuals", Proc. of Coling 86

8)水谷静夫, 1983, "国文法素描", 文法と意味Ｉ, 朝倉書店

9)M. Tomita,1987,"An Efficient Augmented-Context-Free Parsing Algorithm", Computational Linguistics vol. 13,no.1-2

10)松本裕二他, 1983, "ＢＵＰの高速化", 情報処理学会WGNL31-7

11)G.E. Heidorn,1972,"Natural Language Inputs to a Simulation Programming System."Technical Report NPS-55HD72101A. Naval Postgraduate School, Monterey, CA.

12)辻井潤一他, 1985, "格とその認定基準", 情報処理学会WGNL52-3

13)石綿敏雄, 1983, "結合価から見た日本文法", 文法と意味Ｉ,朝倉書店

14)賀来直子, 1985, "日本語のアスペクト", 情報処理学会第３１回全国大会予稿集

15)J. Tsujii, 1982, "The Transfer Phase in an English-Japanese Translation System",Proc. of Coling 82

16)J. Tsujii, 1986, "Future Directions of Machine Translation", Proc. of Coling 86

17)H. Maruyama, 1987, "A Discource Analysis Technique for Natural Language Interface System", Proc. of COMPSAC 87