自然言語処理 29－3
（1982. 1. 22）

日英機械翻訳システムATLAS／U

内田裕士，小部正人，西野文人，増山顕成，松井くにお
（富士通研究所）

1．はじめに

情報化社会の発展に伴って，あらゆる分野において国際的交流が激しくなり，交換されるべき文書の量は技術資料，通信文，契約書類など膨大なものとなっていると推定される．この膨大な量の文書は，世界共通語がない以上翻訳されなければならないが，これを人手で行うことは自ずから限界があり，機械翻訳が所望される所以である．

翻訳方式は，過去幾種類か提案されてきたが，実用性という観点からは次の2方式が注目される．

1）普遍的意味表現経由方式（ピボット方式）

2）言語間対応変換方式（トランスファ方式）

図1に示すように，ピボット方式は以下の手順をとる．

・入力言語の文を普遍的な意味表現へ変換する．

・普遍的な意味表現から出力言語の文へ変換する．

これに対し，トランスファ方式は以下の手順をとる．

・入力言語の文を中間レベルの表現形式へ変換する．

・入力言語の中間レベル表現形式から出力言語の中間レベル表現形式へ変換する．

・中間レベルの表現形式から出力言語の文へ変換する．

言語には，同一の事柄を表現する際に，幾種類もの表現の仕方がある．ピボット方式は，同一の事柄を表現した多様な文から，ある一つの普遍的な意味表現を抽出するものであり，

・多数の言語間の翻訳

・翻訳以外の自然言語理解システム

に応用できる．また意訳などを行うのも有利である．しかし，ピボット方式は，あらゆる言語の文を同じ意味表現形式でとらえようとするため，先ず，普遍的な意味表現形式を定めなければならないが，現在のところ，普遍的な意味表現形式はどのようなものであるか解明されていない．

他方，トランスファ方式では，中間レベル表現は対象となる言語に依存したものであるので研究対象をしぼりやすく，実用化のメドが立てやすい．このためピボット方式に比べてよく研究されている．しかし，どの程度普遍的な意味をとらえて，それを中間レベル表現に組み込めるかによって翻訳の質やトランスファのためのコストが変わってくる．例えば，ほとんど意味をとらえていないところに中間レベル表現を設定する手法，すなわち，入力言語の文型パターンを出力言語の文型パターンにそのまま移す手法では，一般的な文章を翻訳する際には手に負えないほど多くの文型パターンが出現する恐れがある．また，中間表現が，その言語の構文的特徴を強く残した場合は，英語，独語，仏語，など語族の近い（構文や単語の意味が近い）言語間の翻訳の場合には，語彙変換と，それに伴う若干の統語構造の変換を行うだけで，十分な場合が多く，有効と思われるが，日本語，英語，などのように語族が遠い場合には，かなり大きな構造変形を行わなければならないことになるので，あまり有効とは思えない．

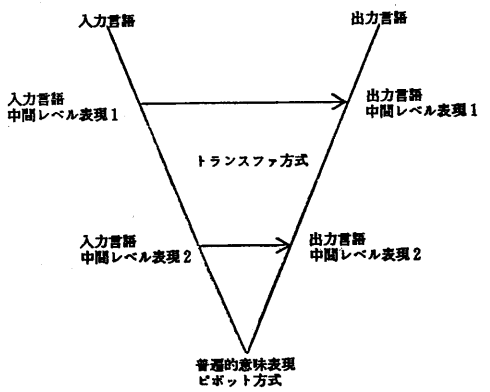

図1　ピボット方式とトランスファ方式

ＡＴＬＡＳ／Ｕにおいては，概念構造と呼ばれる中間レベル表現を介して翻訳を行っている．以下に，この概念構造はどういったものであるのか，また，中間レベル表現に概念構造を採用することの利点，及び，ＡＴＬＡＳ／Ｕにおける処理の概要を述べる．

2．概念構造

概念構造は，文の表わしている意味を概念と概念間の関係を明示することによって表現したものであり，いわゆる文法的な情報は一切入ってこない[1)]．この概念構造においては，概念間の関係は普遍的であるようにしている．従って，概念構造が普遍的であるのかどうかは概念構造を構成する概念が普遍的であるかどうかに依存してくる．一般に，普遍的な概念のセットを定義することは非常に困難であるので，概念はある程度言語に依存したものとなり，概念構造も言語に依存する．

このように，中間レベル表現を，意味を深くとらえたものに設定すれば，ピボット方式の利点をある程度享受するため，他の言語の翻訳システムや，自然言語理解システムへの応用が容易となるし，意訳も可能となり質の高い翻訳が可能になる．また，入力言語の中間レベル表現と出力言語の中間レベル表現との構造的，概念的差異は小さいためトランスファのコストが低く抑えられるという利点もある．この場合は，概念の変換のみが行われる．

概念構造においては，概念をノードで表わし，概念間の関係をアークで表わすので，概念はネットワークになる．

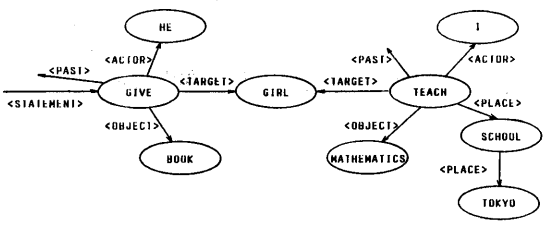

図２　概念構造の例

概念構造において概念をどのレベルに置くかは大きな問題である．我々は概念をＳｃｈａｎｋのようにプリミティブな要素に分解することをせず，単語の持っている概念（あるいはその一部）をそのまま採用した．これは，プリミティブな概念のみで意味を表現すれば，非常に限られたプリミティブ記号のみでよいのでモデルとしてはすっきりしたものとなる反面，そのプリミティブだけですべての意味が表現できるのかどうかといった不安や，（特に文を合成する場合）意味構造から複合概念の抽出の困難さといったものが実用化におけるネックとなる可能性が高いと思われたからである．

「彼は私が東京の学校で数学を教えた少女に本をあげた．」という入力文に対する概念構造の例を図２に示す．

3．ＡＴＬＡＳ／Ｕの概要

ＡＴＬＡＳ／Ｕは文節間の係り受けに基づいた日本語解析を行い，概念構造を作り上げ，そこから英語の構文パターンをもとにして英語文を生成する．

ＡＴＬＡＳ／Ｕの処理概要を図３に示す．

日本語解析は単語（形態素）レベル，構文（文法）レベル，意味レベルの解析の３つからなる．まず，単語レベルの解析が行われ，次に構文レベルの解析が行われる．意味レベルの解析は構文レベルの解析を補う形で行われる．

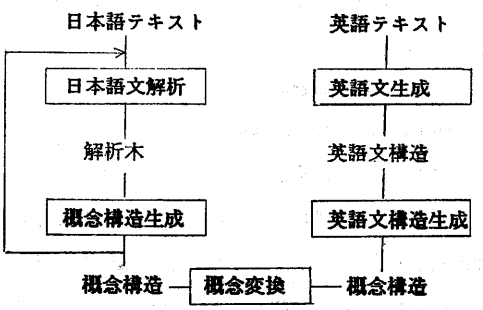

図３　ＡＴＬＡＳ／Ｕの処理概要

単語レベル解析ではベタ書きの漢字カナ混じり文からの単語抽出が行われる．構文レベル解析は文法規則に基づいて係り受け解析を行う．単語の意味のあいまいさや係り受けのあいまいさは意味レベル解析で解消される．解析結果は文節間の修飾関係を表わした解析木である．

概念構造生成では解析木を入力し，各単語の意味（意味は概念記号として定義されている）を解釈して，概念構造を生成する．生成された概念構造は，その妥当性が現実世界との対応において検証される．妥当でないと判断された場合は，日本語解析に対しフィードバックがかけられる．

概念変換では日本語に固有な概念による意味表現を英語に適した概念による意味表現に変換する部分である．

英語文構造生成では，この概念構造の概念記号を対応する英語の句あるいは単語への変換を行い，さらに，英語の構文情報（主語や目的格等）を付加する．この際，概念記号に対して複数個の英語の句や単語が対応することがあるが，その場合，概念構造と一番なじみ易くかつ言い回しからみても正しい句や単語が選択される．このようにして得られたものが英語文構造である．

英語文生成では，英語の文構造を英語の文法に乗っ取って語形変化などを行いつつ英語文を生成し，テキストを出力する．この場合，ひとつの概念構造が２つ以上の文に置き変わる場合もある．

4．日本語解析

自然言語の解析では，文法解析とともに意味処理を本格的に行う必要がある．とくにＡＴＬＡＳ／Ｕにおいては，中間表現を概念構造のような深い意味レベルに設定したので，入力文からその中間表現を作り出すためには，深い意味解析が必要になり，いわゆる知識の利用が不可欠となる．ＡＴＬＡＳ／Ｕの日本語解析は，このような知識の利用を前提とした解析を行っている．

日本語解析は図４に示すような処理から構成される．

4．1．単語抽出

ここでは分かち書きされていない漢字カナ混じり文の日本語テキストから単語を抽出し，文を単語単位に分割する．[2)] 単語は，単語のもつ隣接情報および単語間接続表を使っての単語接続検定，および単語の長さとその出現頻度に基づく最尤評価を用いて，テキストの左から順に選択していく．単語抽出のアルゴリズムはバックトラック付きの縦型探索を用いている．

単語抽出の結果は単語のリストとして表現される．多義語は，それが表わしている概念，または機能毎に１つの単語として登録されているので，そのような単語はリストに候補群がつながれることになる．

4．2．構文解析

文法規則をもとに文節の認定（文節合成）や文節間の修飾関係を明らかにし，入力から生成可能な解析木の１つを生成する．文法規則にマッチした文要素間の関係は意味規則を用いて検査される．これにより，それぞれの単語のもつ概念，または機能が決定される．解析木が出来上がった時点では，単語の候補のうちどれが使用されたかが決定されているので，多義語などのあいまいさは解消されている．

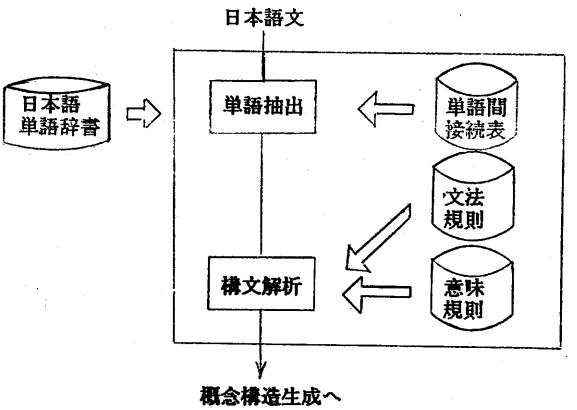

図４　日本語解析

4．3．文法規則

文法規則は図5の形式をしている．文法規則には次のような3つのタイプがある．

1）合成

文法カテゴリ1なる文要素と文法カテゴリ2なる文要素から，文法カテゴリ3なる文要素ができることを示す文節合成規則である．

2）修飾

文法カテゴリ1なる文要素と文法カテゴリ2なる文要素が修飾関係にあり，一方が他方を修飾することによって，修飾された文要素が文法カテゴリ3になることを示す．いわゆる文節間の係り受けを表わす規則である．一般には，左の文要素が右の文要素を修飾するが，連体修飾の場合は逆になることもある．連体修飾や連語などのように，助詞など修飾関係を明示するものが存在せず，文節間に係り受けが生じる場合は，＜relation＞でその修飾関係を補うことができるようになっている．例えば，連体修飾では，名詞から連体形に修飾がおきるが，修飾関係を明らかにする助詞は出現しない．そこで，＜actor＞，＜object＞，＜target＞，等の可能性のある関係を指定しておき，後の意味レベル解析で選択する．

3）先読み：　1つ先の文要素により現在の文要素の文法カテゴリが変化する場合に使用する文法規則である．例えば，動詞連用形は，動詞本来の性質をもつ場合と，名詞として転用される場合とがある．この場合，名詞として働くのか動詞として働くのかはそれより前の文節の係り受けに影響を与える．このような場合，先読みができれば便利である．例えば，“きれいな散り際”のように連用形を名詞化する接尾語がある場合は，連用形が名詞として働き，“きれいな”は“散り”を修飾する．

```
<condition>+<grm. cat. 2> → <grm. cat. 3>
<grm. cat. 1>                <type>
                             <relation>
                             <action>
                             <priority>
```

図5　文法規則の形式

4．4．意味規則

構文解析に使用される意味規則は図6のような形式で意味関係表に格納されている．意味関係表は言語的知識や常識を概念と概念間の関係として記述したものであり，構文解析のみならず，概念構造生成でも使用される．

4．5．無意味文の解析

自然言語の文には，例えば“数学が走る”といった無意味な文が出てくることがある．このような場合，意味解析を行えば解釈できなくなってしまう．ATLAS／Uでは，意味解析のレベルを順次落としていけるようにして，意味的におかしいとされた関係も，それ以外に解釈のしようがないときは，文法規則による解析を優先することができるようになっている．

5．概念構造生成

日本語文を解析することによって得られた解析木より，各単語の意味を解釈して概念構造を生成する．単語の意味は概念記号として，辞書のエントリに示されている．生成された概念構造は意味検証規則と意味関係表の知識に基づいて検証される．ここで概念構造の表現する意味が現実世界との対応において妥当でないと判断された場合は，日本語解析に対してフィードバックをかけ，次の解析木を要求する．これは妥当と思われる概念構造がえられるまで続けられる．

概念1	概念2	関　係
HUMAN	TEACH	＜ACTOR＞
HUMAN	TEACH	＜TARGET＞
TOKYO	SCHOOL	＜PLACE＞
BOOK	GIVE	＜OBJECT＞
・	・	・
・	・	・
・	・	・

図6　意味規則

解析木からそのまま得られた概念構造は原言語に強く依存したものであるので，より普遍的な概念を用いた意味表現への変換が行われる．例えば，＂信号を赤に変化させた＂という文章と＂信号を赤に変えた＂が同じ概念構造になるように処理される．このような例は上例の使役動詞とそれに対応する動詞の間で行われる他に，可能動詞とそれに対応する動詞に＂ことができる＂を付けたものとの間などで行われる．

これによってある程度普遍的な概念構造が得られる．また日本語によくある省略語に対する補いもここでなされる．

6．概念変換

ある程度普遍的な概念構造を目標言語になじみやすい概念構造に変換する部分であり，英語文生成の一部と見なすこともできる．概念変換は概念変換規則もとにして行われる．

7．英語文構造生成

ＡＴＬＡＳ／Ｕでは，一種のセマンティックネットワークである概念構造から英語文を生成している．セマンティックネットワークから英語文を生成する方式として，Ｒ．Ｓｉｍｍｏｎｓらの再帰遷移ネットワークによる生成文法を用いたアルゴリズムがあるが[3)]，ＡＴＬＡＳ／Ｕでは翻訳の際の中間表現である概念構造から英語文を生成するため，語彙選択に重点をおき，構文パターンと言い回し規則を用いたアルゴリズムを考えた．

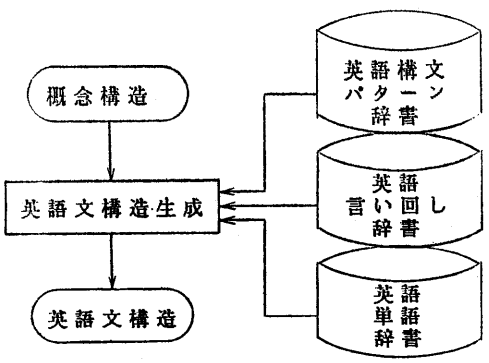

図7　英語文構造生成

英語文構造生成の処理の概要を図７に示す．図７では，入力された日本語文から作成した概念構造を，英語文構造生成によって英語文構造を生成する際に必要な辞書を示してある．

英語文構造生成では，概念構造中に示す概念記号から，英語の候補単語群を英語単語辞書から抽出する．この候補単語群から正しい言い回しになるような候補単語を英語言い回し規則を参照することによって調べ，かつ候補単語の持つ構文パターンが概念構造にマッチするものを選び，最適な単語をネットワークに沿って，順次決定していく．この過程で候補単語がなくなった場合は，バックトラックがかけられる．構文パターンには，単語の支配する構文およびその語順が定義されているので，マッチングがとれると，単語の構文的役割と語順が決定される．

7．1．アルゴリズム

構文パターンの適用と言い回し規則を利用し，単語とその構文的役割および語順を決定するアルゴリズムを次に示す．

1）＜ｓｔａｔｅｍｅｎｔ＞と明示されたアークからネットワークにはいる．
2）アークに対する候補単語群を英語単語辞書からもってくる．
3）アークに対する候補単語群から最尤候補語を選び出す．最尤候補がない場合は，現在訪れているノードの最尤候補を捨てて，５）に戻る．
4）アークの先のノードを訪れ，ノードに対応する単語で，アークの最尤候補と言い回しが正しくなる単語を英語単語辞書からもってくる．
5）ノードに対する候補単語群から最尤候補を選び出す．最尤候補がない場合はノードを去り，アークの最尤候補を捨てて，３）に戻る．
6）ノードの最尤候補の支配する構文パターンに従って，まだ埋められていない構文スロットを埋めるアークを見つけ出す．

－アークが見つかれば2）へ行く．

－アークが見つからない場合で，その構文スロットが必須である場合，5）に戻る．

－構文スロットを埋めるべきアークがなくなった場合で，必須の構文スロットがすでにすべて埋められている場合は，ノードを去り，6）に戻るか，終了する．

7．2．構文パターン

構文パターンは図8のような構文スロットの全順序集合である．この順序は語順を表わしている．図9に動詞ｍａｋｅに対する構文パターンの例を示す．

構文スロットには，そのスロットを適用すべき，および適用してはいけない環境（概念構造のパターン）が記述されている．従って，構文スロットは種々の環境に応じ適用されたりされなかったりする．

ここで実際の例に当てはめて，構文パターンの適用を行う．図１０には「彼は東京に家を建てた．」という入力文に対する概念構造を示す．また図１１には，概念記号に対応する単語の候補を示す．

まずMAKEからｍａｋｅを選ぶ．ｍａｋｅがとる構文パターンに従い，＜ａｃｔｏｒ＞とHEのｈｅはマッチングがとれる．次にｍａｋｅと＜ｏｂｊｅｃｔ＞を介してｈｏｕｓｅとの言い回し規則を調べるが，これが不成立のため，ｍａｋｅを選択せずに，ｂｕｉｌｄを選ぶ．

このようにして順次単語を選択し，その構文パターンを適用しながら，言い回し規則を調べて，最適語と語順を決定していく．

2	PTNID			構文パターンｉｄ
V	SYNPTN			構文パターン
	1	ELMNUM		構文要素の数
	*	2	CATNO	左カテゴリ番号
		1	SR	構文役割
		1	CMT	備考
		1	NCCNO	NCCの数
		*	1 NCC	Network Configuration Condition

図8　構文スロット

```
<CAUSE>       0           OPT       0
<CONDITION>   0           -OPT      0
<ACTOR>       SUBJECT     NEED      (-INTRO->,V(SUBJ))
<OBJECT>      DIR_OBJ     NEED      (-INTRO->,V(OBJ))
<TARGET>      INDIR_OBJ   -NEED     (-INTRO->,V(TRGET))
<PLACE>       ADVERBIAL   -NEED     (-INTRO->,V(PLACE))
<SUFFERER>    SUBJECT     NEED      SUBJ?
  BE          0           OPT       OBJ?
<TARGET>      0           OPT       TRGET?
<PLACE>       0           OPT       PLACE?
<TIME>        0           OPT       TIME?
              0           G(DO)     (<-STMT,INTRO->,SUBJ->)
<NOT>         0           OPT       (<-STMT,INTRO->,SUBJ->)
              0           G(BE)     (INTRO->,-SUBJ->)
<SUFFERER>    SUBJECT     -NEED     (SUBJ->,-V(SUBJ),-SUBJ?)
              0           G(DO)     (-INTRO->,NOT->,-CONT->,-ASPECT->)
<FUTURE>      0           OPT       -INTRO->
<NOT>         0           OPT       -INTRO->
  BE          SUBJECT     -NEED     (-SUBJ->,-<-OBJ,-<-FOR,-V(OBJ),-35)
              0           G(BE)     (-INTRO->,-SUBJ->,-<-OBJ,-<-FOR,-35)
<NOT ONLY>    0           OPT       0
<BUT ALSO>    0           OPT       0
              ACT_VERB    OWN       SUBJ->.35
              PASS_VERB   OWN       (-SUBJ->,-<-OBJ,-<-FOR,-35)
              INF_VERB    OWN       (-SUBJ->,-<-OBJ,<-FOR)
  BE          DIR_OBJ     OPT       (SUBJ-> ,-V(OBJ))
  BE          DIR_OBJ     OPT       <-OBJ,35
<TARGET>      INDIR_OBJ   OPT       -V(TRGET)
              0           *         0
              0           G(,)      <-COND
              0           G(?)      (<-STMT,INTRO->)
              0           G(.)      (<-STMT,-INTRO->)
```

図９　構文パターンの例

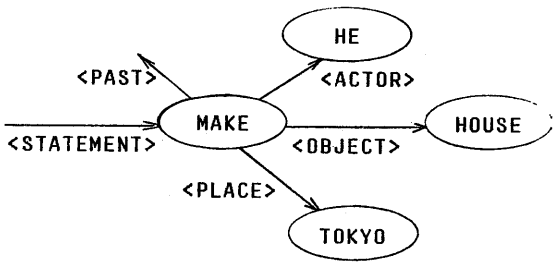

図１０　概念構造

```
<STATEMENT>   →   .
<OBJECT>      →   を
<PLACE>       →   ｉｎ，ａｔ
<ACTOR>       →   は
MAKE          →   ｍａｋｅ，ｂｕｉｌｄ
HE            →   ｈｅ
HOUSE         →   ｈｏｕｓｅ
TOKYO         →   Ｔｏｋｙｏ
```

図１１　概念記号と単語の対応

8．英語文生成

概念構造のノードやアークに対して選ばれた単語は，前述の構文パターンで指定された順にしたがって，文字列（単語列）に置き換えられる．その際，単語に与えられた意味（単複，人称等）や構文的役割（受身形，完了形等）に従ってそれ自身，及びそこから構文的に影響を受ける単語（主語に対する動詞等）の語形変化が行われる．また，文章の整形（例えば，同じ概念を表わす単語を2度出力しようとしたときに，それを代名詞で置き換えたり，2度めの単語を省略したりするなど）もこの段階で行われる．

9．おわりに

ＡＴＬＡＳ／Ｕは，現在ＦＡＣＯＭ　Ｍシリーズ上で動いており，計算機マニュアルを対象とした翻訳を行い，評価，改良中である．

ＡＴＬＡＳ／Ｕのように知識の利用を前提とした深い意味解析を行いながら自然言語解析をする際に問題となるのは，意味的な検証をどの範囲で行うかということである．ＡＴＬＡＳ／Ｕでは現在１つの文章単位にこのような検証を行っているが，実際には１つの文では論理的におかしくてもパラグラフ単位で見れば論理的に間違っていないと言う場合もあり，１つの文に限定する理論的根拠はない．現在，このような場合においてどの程度意味を無視しながら解析を行うかを検討中である．

参考文献

1) Uchida, H. and Sugiyama, K. : "A Machine Translation System from Japanese into English based on Conceptual Structure", COLING－80, pp. 455－462, Oct., 1980

2) 内田他："係り受けと格を用いた文章解析について"，情報処理学会第２０回全国大会，５Ｃ－３，１９７９

3) Simmons, R. S and Slocum, J. : "Generating English Discourse from Semantic Networks", Comm. ACM Vol. 15 No. 10 (1972)