自然言語処理 63－8
（1987. 9. 25）

翻訳システムTITLE-1の概要

新井 幸宏　　安居院 猛　　中嶋 正之
東京工業大学 工学部 像情報工学研究施設

筆者らは中学校の教科書程度の英文を和訳する能力を持つシステムＴＩＴＬＥ-１(Tokyo Institute of Technology Language Engine-1)を構築中である．本システムは、ＵＮＩＸシステム上でのプロセスの連結によって実現されており、前処理部、翻訳部、後処理部の三つの部分から構成されている．前処理部では、入力された英文の構文は扱わずに、単語レベルでの処理を行う．処理の内容は、英語の単数複数、動詞の定形、形容詞の級などの不規則な変化を規則化し、さらに、翻訳部分で利用しやすい形式にすることである．ここで変換の仕方が一通りでないときは、各結果を列記したものを翻訳部へ送るようにしている．翻訳部では、文脈自由文法に意味属性を持たせた規則により、トップダウン的に構文解析と意味の生成を行う．ここでは、前処理部から与えられる多義性と構文解析によって生じる多義性により複数の結果が得られることもある．この段階では、日本語の非活用部分、活用型、客体表現語尾、主体表現語尾などが、分離された形式で出力される．後処理部では、これらの情報をもとに日本語文の生成を行う．

本報告では、ＴＩＴＬＥ－１の構成の概要と英文翻訳実験の結果を示す．

The Outline of Translator TITLE-1

Yukihiro ARAI, Takeshi AGUI and Masayuki NAKAJIMA

Tokyo Institute of Technology,
Imaging Science and Engineering Laboratory
Nagatsuta-cho 4259 Midori-ku Yokohama 227 Japan

We are constructing an English-Japanese translation system TITLE-1(Tokyo Institute of Technology Language Engine-1), which can deal with english sentences in junior highschool text books. This system is realized as a piped connection of processes on the UNIX operating system, and composed of three parts, a preprocessor, a translator and a Japanese generator. The preprocessor regurates the irregular conjugations of inputted english, such as irregular plural forms of nouns, irregular conjugations of verbs and adjectives. If the results have some ambiguity the preprocessor outputs all possible results in a parallel formats and send them to the translator. The translator has a set of context free grammar with meaning atributes. It analyses the inputted information and chages it into a meaning expression. This part also generats multipul results if there is any ambiguity. The Japanese generator composes Japanese sentences using the information from the translator.

This report presents the outline of TITLE-1 and some results of translating experiments.

1．はじめに

一般に、自然言語の機械翻訳では、分野の限定をしないと良好な訳文を得ることが困難である．限定分野として、科学技術書などがよく扱われるが[1]、教科書文の翻訳も重要な一分野である[2]．

筆者らは、中学校の教科書程度の英文を和訳する能力をもつ英日翻訳システムＴＩＴＬＥ－１（Tokyo Institute of Technology Language Engine-1）を構築中である．本システムは、ＵＮＩＸシステム上でのプロセスの連結によって構成されており、図１に示すように三つの部分から構成されている．第一番目の前処理部では、入力された英文の各単語に関する処理が行われる．処理の内容は、名詞の複数形、動詞の過去・過去分詞形、形容詞の比較級・最上級などの不規則変化を規則変化に戻すことと、形態素に分離された形式の出力を翻訳部に送ることである．ここで出力される形式は、構文的には原文と変わりはないが、不規則変化を含んでいないので、システマティック・イングリッシュ文（ＳＥ文）と呼ぶことにする．第二番目の翻訳部では、意味属性付き文脈自由文法を用いて、ＳＥ文を、システマティック・ジャパニーズ文（ＳＪ文）に変換する．ＳＪ文では、名詞、動詞の語幹、形容詞の語幹などの非活用部分と、助動詞、動詞の語尾、形容詞の語尾などの活用部分は分離されており、また、活用語尾も客体語尾[3]と主体語尾[3]などに分けられている．また、ここで特徴的なのは、名詞に対しても動詞や形容詞に準拠した活用型を導入して、統一的に扱おうとしていることである．第３番目の後処理部では、翻訳部から出力される上記の各種の情報を用いて、日本語文が生成される．

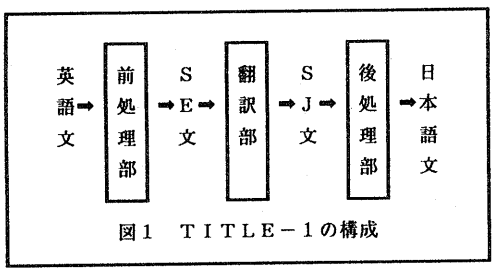

図１　ＴＩＴＬＥ－１の構成

本システムでは、各部分の情報のやりとりはＳＥ文、ＳＪ文などの中間形式だけによって行われており、共通の　ファイルを参照することなどは無いようになっている．また、ＳＥ文とＳＪ文は、人間が見ても意味のわかるテキスト形式になっている．これにより、各部分の開発および変更、修正を全く独立的に行うことができる．今回の報告では、各部分の概略的な内容を述べると共に中学校教科書の翻訳実験を行った結果について述べる．

2．前処理部

一般的には，形態素解析を行うときに，入力された単語を辞書引きして，その単語が見つかれば，そのまま出力し，見つからなければ単語を分解する規則などを用いて単語を接頭語，語幹，語尾などに分解して再び辞書引きする方法が取られることが多い．しかし，その方法を用いると形態素解析段階で全ての単語を辞書引きしなければならない．また，辞書引きの効率を良くするために，構文解析や意味解析に必要な情報もそのときにまとめて引き出すことにもなり，各部分の相互依存性が高くなってしまう．

本システムでは，前処理部，翻訳部，後処理部の独立性を高めるために，前処理部では，規則変化語や不規則変化語だけを規則的な記号の列に変換する方式をとった．本方式では，入力された単語を不規則変化語表で調べ該当するものがあれば原形と記号の列に変換し，次に規則変化規則に照らして該当するものは，同様に原形と記号の列に変換する．また，いずれにも該当しないものは，そのまま出力する．このとき実質的に辞書引きされるのは，不規則変化語の表だけであり，規則変化語の処理は単に語尾の数文字を調べるだけで行うことができる．両者の違いを図２に示す．

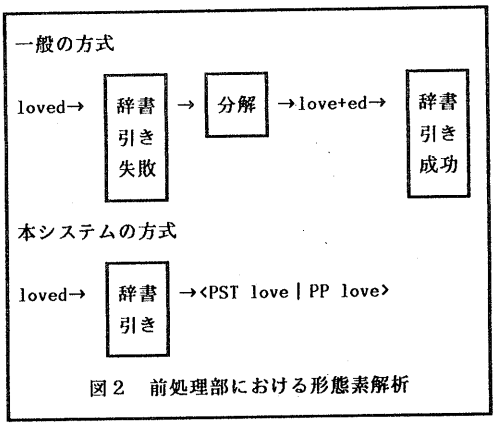

図２　前処理部における形態素解析

図２の<PST love | PP love>は，lovedがloveの過去または過去分詞であることを示している．前処理部の出力結果の例を図３に示す．

入力単語	出力単語と出力記号
birds	bird PL
children	child PL
walked	<PST walk | PP walk>
took	PST take
left	<left | PST leave | PP leave>
leaves	<leaf PL | S leave>
better	<ER good | ER well>

図３　前処理部の出力の例

出力中に含まれる記号の一覧を図４に示す．

記号	意味	用法	用例
PL	複数	後置	book PL
S	三単現	前置	S go
PST	過去型	前置	PST go
PP	過去分詞型	前置	PP go
ING	ING型	前置	ING go
ER	ER型	前置	ER big, ER speak
EST	EST型	前置	EST big

図４　前処理部の出力記号の一覧

出力記号のうちで複数を表すPLは、翻訳部が複合名詞を扱うときに便利なように後置型にしてあり、その他は記号が述部の先頭に現れるように前置型にしてある．

入力された英文の各単語に上述の変換を施したものをＳＥ文と呼ぶ．図５に入力文に対応するＳＥ文の例を示す．

```
(1)The man had some books under his arm.
  the man <PST have | PP have> some <S book |
  book PL> under <his | he's> arm.

(2)Jack and Betty wash their faces and hands.
  jack and betty wash they's <S face | face PL>
  and <S hand | hand PL>.
```

図５　入力文とＳＥ文

実際、前処理部は、入力英文に対して図５に示されるようなＳＥ文を出力する．

３．翻訳部

３．１．並記内容の選択

翻訳部では、前処理部から受け取ったＳＥ文に対してトップダウン的に構文解析を行うが、受け取ったＳＥ文に多義並記部分、すなわち＜＞で囲まれた部分があれば、並記内容を順次選択しながら複数の文を作成して、その

```
ＳＥ文
he <PST have | PP have> some <S book | book PL>.

選択１　he PST have some S book.
選択２　he PST have some book PL.
選択３　he PP have some S book.
選択４　he PP have some book PL.
```

図６　並記内容の選択の例

それぞれに対して構文解析を行う．並記内容の順次選択の例を図６に示す．

この例の場合、選択２だけが正しい構文を表しており、その他は誤っているので、次の構文解析段階で切り捨てられる．

３．２．　構文規則の記法

本システムでの構文規則の記法は、通常の文法書などの文法の説明の形式に類似した形式をとっており、システムの動作を理解していなくても規則の意味は理解できるようになっている．最も簡単な規則の例を図７に示す．

```
Sentence
  good morning./          #ohayougozaimasu.;
_ good afternoon./        #konnnichiha.;
_ good evening./          #konnbannha.;
```

図７　構文規則の例１

図７の規則は、Sentenceという非終端記号の定義と各定義の意味を表している．すなわち、good morning., good afternoon.およびgood evening.は、いずれもSentenceであり、それぞれの意味ないしは訳語は、後ろに書かれたローマ字によって与えられる．ローマ字は、後処理部で仮名に変換される．また、# は、分かち書きをするときのスペースの位置を表す記号である．分かち書きをするかどうかは、後処理部で決める．実際に、それぞれの入力に対して、訳語を出力するためには、START という予約語の定義の中に、Sentenceを含めなければならない．その記法を図８に示す．

```
START
  Sentence/&;
```

図８　STARTの定義

図８の記法は、図７の記法と全く同じ形式のものである．Sentenceは、図７で定義されているので、自動的に非終端記号と見なされる．& 記号は、Sentenceの意味をそのまま用いることを表している．このように、任意の文字列を非終端記号として用いることができるが、終端記号と非終端記号が混同されることのないように、非終端記号の頭文字には大文字を用いることにしている．図

```
Verb
  have Drink/           Drink=wo #no MG;
_ have Meal/            Meal=wo #tabe IC;
_ have Time/            Time=wo #sugo SG;
```

図９　構文規則の例２

８の規則の後に別の内容を追記すれば、START に対して、Sentence以外のものも同時に定義することができる。図９の例では、Verbの定義に終端記号と非終端記号の両方が含まれている。

haveの日本語訳は、目的語の種類によって変化させる必要があるので、このように目的語の部分に非終端記号を用い、目的語のグループ化を行う。例えばDrinkの定義は図１０の様に行うことができる。

```
Drink
  coffee/          #ko-hi-;
_ juice/           #ju-su;
_ milk/            #miruku;
```

図１０　Drinkの定義

図９の意味属性部の非終端記号には、それぞれの意味が代入されることになる。woの前の =記号は、非終端記号と助詞をつなぐための記号で、非終端記号に空文字列が代入されると助詞が消えるようになっている。また、最後にあるMG,IC,SGなどは動詞の活用型を表しそれぞれ「ま」行五段活用、一段活用、「さ」行五段活用を表している。

図１１はVerbを用いて、Sentenceを定義した例である。

```
Sentence
  SUB_A Verb./          SUB_A=ha,Verb Dt;
_ SUB_B S Verb./        SUB_B=ha,Verb Dt;
```

図１１　構文規則の例３

この例では、SUB_B が三人称単数の主語を表し、SUB_A がその他の主語を表している。S は、前述したように動詞の三単現の型を表すものである。また、意味属性部の最後にあるDtは、断定の語尾を表している。この部分は後処理部によって動詞の活用型に対応した形に変換される。

```
Sentence
  do SUB_A Verb?/       SUB_A=ha, Verb Gm.;
_ S do SUB_B Verb?/     SUB_B=ha, Verb Gm.;
```

図１２　構文規則の例４

```
Sentence
  SUB_A do not Verb./   SUB_A=ha, Verb Ht Dt.;
_ SUB_B S do not Verb./ SUB_B=ha, Verb Ht Dt.:
```

図１３　構文規則の例５

同様にして、疑問文や否定文も定義できる。図１２は疑問文、図１３は否定文の定義である。

図１２中の記号Gmは、疑問の語尾を表し、後処理部によってDtと同様の取り扱いを受ける。DtやGmは、客観的な事象を表すのではなく、それに対する話者の態度を表すものなので、主体語尾と呼ぶ。主体語尾には、このほかに命令を表すMr、勧誘を表すKu、依頼を表すIrなど各種のものを用意している。これに対して図１３のHtは、動詞の意味の否定という客観的な事象を表しているので客体語尾という。客体語尾は、通常、主体語尾または他の客体語尾または単語が後に続かなければならない。文は必ず話者の態度を伴って終了しなければならないからである。実際の日本語では主体語尾が空文字列の場合もあるが、本システムでは、翻訳部の出力は必ず主体語尾で終了するようにしてある。また、客体語尾は、最終的な日本語文に成るときに活用語として表現される場合が多い。実際、図１３のHtは、「～しない」という形を表し、この形は、形容詞と同じ活用をする。客体語尾には、このほかに、受け身を表すUm,過去を表すKk など10数種が用意されている。

同様な手法により、英語の各種の助動詞表現なども日本語の客体語尾と主体語尾の組み合わせに変換する。

英語の形容詞に関しても同様であるが、英語の形容詞に対応する日本語は、形容詞、形容動詞、動詞の連体形、動詞＋過去の助動詞、名詞＋助詞など実に多様である。

形容詞の翻訳に関する規則の例を図１４に示す。

```
Adjective
  red/              #aka IK;
_ healthy/          #kenkoutcki NK;
_ dead/             #shi NG Kk;
_ japanese/         #nihon M;
```

図１４　構文規則の例６

図１４では、red,healthy,dead,japanese というような英語の形容詞が、日本語では「赤い」、「健康的な」、「死んだ」、「日本の」などと各種の品詞に変換される。図中のIKは形容詞、NKは形容動詞、NGは「な」行五段活用、M は名詞をそれぞれ表している。

以上に述べたような記法により、英語における各種の表現が、日本語の非活用部分、客体語尾および主体語尾の列に変換されて後処理部へ渡される。

３．３．翻訳部の動作機構

本システムでは、前述のような構文解析およびＳＪ文合成の規則を通常のエディタを用いて作成し、システム起動時に、本体プログラム上の内部形式に変換する方式

をとっている．本体プログラムは、ＵＮＩＸ／Ｃ言語によって作成したものである．内部形式では、各終端記号、非終端記号およびＳＪ文用の単語と記号は全て２進木リストに格納され、各規則はリストの要素を指すポインタの列に置き換えられている．

翻訳部には構文解析を行うためにスタックが一個用意されており、その第一要素には、予約語である START をさすポインタが代入されている．ＳＥ文が入力されるとＳＥ文自身にも２進木探索が施されて、ＳＥ文に相当するポインタの列が生成される．続いて、各構文解析規則によって START記号が展開され、展開過程がスタックに積まれてゆく．展開過程は、常にＳＥ文と比較されており展開の結果がＳＥ文になる見込みが無くなった時点で直ちにバックトラックするようになっている．比較の内容は終端記号のマッチングは勿論であるが、そのほかに文の長さや、マッチングされる以前の単語の語順など幅広い点検を行っている．これによって左回帰則による暴走等も防ぐことができ、誤った探索経路に入り込んでも早い段階でバックトラックすることができる．

展開結果がＳＥ文に相当するポインタ列に一致するとその時点で、スタックに積まれている展開過程に従ってＳＪ文の合成が行われる．ＳＪ文の合成も完全にポインタの形で行われ、合成が完了して出力される段階でテキスト形式に変換される．

展開結果がＳＥ文と一致しても解析過程は続行される．これによって複数の解釈がありえる場合には、それぞれに対するＳＪ文が合成されることになる．

さらに、前述したようにＳＥ文自身に多義性があれば、構文解析が再実行されるので、さらに多くのＳＪ文が合成されることもある．

中学校程度の文法および辞書をテキスト形式から内部形式に変換するのに、１～２分の時間を要するが、現時点では、文法の修正と翻訳実験の繰り返しがほとんどであるので、内部形式をそのままファイルに記録するようなことは行っていない．将来、文法規則が確定的なものになれば、内部形式による保存が有効になるであろう．

３．４．疑問詞に関する記法

以上、基本的な文法規則の記法と、翻訳部の動作機構について述べたが、このままでは、英文の基本的要素である疑問詞を有効に扱うことができない．疑問詞を用いた疑問文では質問すべき内容が文中の本来の位置から抜けて疑問詞の形で左端へ置かれると考えることができる．本システムでは、抜け落ちる可能性のある単語に空文字列を定義しておき、ＳＪ文合成段階で変則的な代入を行うことによって疑似的に解決している．その記法の例を図１５に示す．

この例では、Q という非終端記号が疑問詞、すなわち what、who などを表している．また、<Q Obj>は、それ以降の部分かにObjという非終端記号があれば，そこにQの意味を代入せよということを表している．これによって、例えば、What do you eat?という文が「アナタハ、ナニヲタベマスカ．」というような文に変換される．

```
Sentence
  Q do SUB_A Verb?/  SUB_A=ha <Q Obj>Verb Gm.
Q
  what/                #nani;
_ who/                 #dare;
_ which Obj/           #dono Obj;
```

図１５　疑問詞に関する記法

３．５．翻訳規則の実際

本システムで実際に用いられている規則は、辞書および文法を含めて数千行に及んでいるので本文中で紹介することできないが、その概要について述べる．

動詞に関しては、図９と同様の形式で約６００個の動詞句を持っている．但し実際の規則では動詞に付随して各種の修飾語に関する規則も並記されている．例えば場所に関する修飾語または副詞を考えると日本語に変換したときに用いられる助詞が場合によって異なるのでそれぞれの動詞に対して訳し方の規則が設定されている．この例を図１６に示す．

```
Verb
  eat Obj Place/    Place=de Obj=wo #tabe IC;
_ put Obj Place/    Place=ni Obj=wo #o KG;
_ walk Place/       Place=wo #aru KG;
```

図１６　場所の訳し方の例

名詞に関しては、人、動物、物、などの区別の他に、have,take,playといった特殊な動詞に関して動詞の訳し方が異なるようなものについては、それぞれのグループを表す非終端記号を設けて、正しい訳文が生成されるようにしている．実際、人、動物、物を分けているのも、old 等の日本語訳が生物と無生物で異なったり、数え方が人と動物で異なったりすることが大きな理由になっている．本システムでは、約３００個の名詞を持っている．

形容詞は、前述したように様々な品詞の日本語に訳される．また修飾する相手の単語によって訳し方の異なるものも多い．本システムでは約１００個の形容詞を持っている．

副詞に関しては、形容詞に類似した扱いをしているが、意味や用法が多彩なため、現時点では十分な翻訳を行うには、まだ不満がある．

４．後処理部

翻訳部から出力されるＳＪ文は、非活用部分、活用型、客体語尾、主体語尾から成り立っている．ＳＪ文の例を図１７に示す．

```
(1) watashi ha shounen M Kk Dt.
(2) anata ha sono hon wo yo MG Se Um Kk Gm.
```

図１７　ＳＪ文の例

(1) の例では、「watashi ha shounen」 は 「ワタシハショウネン」を表し、「M」 は、そこまでの部分を名詞として扱うことを示しており、「Kk」は、過去を表す客体語尾で、「Dt」は、断定を表す主体語尾である．

(2)の例では、「MG Se Um Kk Gm」がそれぞれ「ま」行五段活用、使役、受け身、過去、疑問を表している．

本システムでは、活用型の種類として、名詞、形容詞、形容動詞、各行の五段活用、一段活用、「さ」行変格活用、「か」行変格活用、「ゆく」活用、「ある」活用などが用意してある．普通は、名詞は活用しないと考えられているがここでは、助詞や、助動詞を、名詞の活用語尾と考えることによって、動詞や形容詞と同様の取り扱いをしている．また、「ゆく」と「ある」は慣習的に特殊な用いられ方をするので独立した活用型として扱っている．

活用型の次に続く客体語尾は、話者の態度と直接関係の無い客観的な状況を表すものである．客体語尾には、過去、否定、受け身、使役などの客体世界の関係を表す語が含まれる．

活用型と客体語尾が結び付くと新たな活用型が作り出される．例えば、図１８に示すように、各活用語に否定の語尾が結び付くと形容詞が生まれる．

```
hon M Ht    --->  hondehana IK
ka KG Ht    --->  kakana IK
ARU Ht      --->  na IK
```

図１８　活用型と客体語尾の結合

図中の ARUは、「ある」活用をあらわし「IK」は、形容詞を表している．すなわち、「本」の否定型は「本ではない」、「書く」の否定型は「書かない」、「ある」の否定型は「ない」ということである．

後処理部では、活用型に語尾を結合させて新たな活用型を作り、それに再び語尾を結合するという操作を繰り返す．

最後に主体語尾が結合されると、もはや活用型は作られずに、完成した日本語文が出力される．主体語尾は、話者の態度を表す語尾で、命令、断定、推量、勧誘などを表す語尾が含まれる．主体語尾は直接活用型に結合することもできる．図１９に各種のＳＪ文が日本語文に変換される過程を示す．

```
(1)hon M Dt      --> honda     --> ホンダ
(2)ka KG Kk Dt --> kaita     --> カイタ
(3)ka SG Kk Dt --> kashita --> カシタ
(4)ka TG Kk Dt --> katta     --> カッタ
(5)Ka MG Kk Dt --> kanda     --> カンダ
(6)ka GG Kk Dt --> kaida     --> カイダ
(7)ka KG Dt      --> kaku      --> カク
(8)ka KG Se Dt --> kakasu   --> カカス
(9)ka KG Se Um Dt -->kakasareru --> カカサレル
```

図１９　ＳＪ文から日本語文への変換

５．翻訳実験の結果

以上に述べたシステムを用いて中学校程度の英文を翻訳した例を図２０に示す．

```
(1)Jack and Bill are my friends.
   ジャックトビルハ、ワタシノトモダチダ．
(2)I have been sick in bed.
   ワタシハ、イママデビョウキデネテイタ．
(3)We had much snow last year.
   キョネンハ、タイリョウノユキガフッタ．
(4)Let's play base ball.
   ヤキュウヲシヨウ．
(5)Don't be angry with me.
   ワタシニタイシテオコルナ．
```

図２０　翻訳結果の例

現段階で、筆者らは中学校教科書の英文を次々に入力し正しい結果が得られないときには構文規則を修正するという作業を続行している．現段階で、翻訳に要する時間は文の複雑さに応じて、数秒から数十秒である．

６．おわりに

本報告では、ＵＮＩＸシステムのプロセスの連結を利用した翻訳システムについて述べた．本システムの能力は非常に限られたものであるが、これを基本にして今後の発展の方向を探って行きたいと考えている．

７．参考文献

[1] 野上ほか「英日機械翻訳システムにおける英文の解析方法について」，情処研報87-NL-61-4.

[2] 工藤「機械翻訳システムによるＣＡＩ」，情報処理学会第３２回(昭和６１年前期)全国大会3M-4,pp1219-1220.

[3] 芳賀綏「新訂日本文法教室」教育出版、1982.