

1980年代の自然言語生成 — 1 —

Natural Language Generation in 1980s — 1 —

徳永 健伸* 乾 健太郎*
Takenobu Tokunaga Kentaro Inui

* 東京工業大学工学部情報工学科
Dept. of Computer Science, Faculty of Eng., Tokyo Institute of Technology.

1990年9月21日 受理

Keywords: natural language generation, text generation, text planning, cohesion, what to say, how to say, grammar and generation, discourse structure, hearer's model.

1. はじめに

自然言語生成と人工知能の関係は、エキスパートシステムの説明文生成にその接点を見出すことができる。エキスパートシステムは自然言語生成を必要とする数少ない分野の一つであった。人工知能研究の1分野である自然言語処理研究に目を移してみると、自然言語生成は機械翻訳の研究の一部として行われていたものの、生成自身が興味を中心となることは少なかった。自然言語理解の研究が自然言語処理研究の分野で中心的な位置を占めてきたのに対し、自然言語生成はどちらかという軽視されてきた感がある。しかし、自然言語生成が人間の高度な知的活動と関連しており、これが人工知能の研究と深く関わることはよく知られている。このような立場から、1980年代に入って自然言語生成の分野でも興味深い研究成果が多く提出されるにつれ、自然言語生成の研究もこの10年間で大きな発展を遂げてきた。そこで今回から3回に分けて1980年代に行われた自然言語生成に関する研究を概観し、1990年代に向けての課題と展望を考える。

自然言語生成に関する解説はこれまでもいくつかあるが^{(1)~(4)}、本稿では、各生成システムの具体的なイメージを読者につかんでもらうために具体例をできるだけ取り入れるように努めた。表題も含めて「文章生成」や「文生成」ではなく、「自然言語生成」という語を使うのは、発話生成なども含めて自然言語の生成に関わる話題をできるだけ網羅したかったからである。また、本文中の「聞き手」、「話し手」といった語は特に断らない限り、それぞれ「読み手」、「書き手」も含

んだ意味で使っている。

1-1 1970年代の自然言語生成

本節では、1970年代に行われた自然言語生成に関する研究を佐藤の分類⁽⁵⁾に浴って簡単に振り返る。佐藤は1970年代の自然言語生成に関する研究を以下のよう分類している。

- (1) ランダムな生成
- (2) 雛型による生成
- (3) 意味構造からの生成
- (4) 機械翻訳における生成
- (5) 物語の生成

この分類はおおむね歴史的な流れに沿った分類となっている。

(1) ランダムな生成では、文脈自由文法を用い、開始記号から書換え規則を使ってすべての記号が終端記号になるまで書換えを行う。DCG⁽⁶⁾を生成用に動かすことをイメージすればよいであろう。何の制約も加えない文脈自由文法を用いると統語的に正しくても意味的には正しくない文が生成される可能性がある。文脈自由文法に意味的な制約を加え、できるだけ意味のある文だけを生成することを目指した研究もある。いずれにしてもこの範疇の研究では、入力として特定の意味を与え、その意味を表現する文の生成を目指しているわけではなく、計算機によって文を自動的に生成すること自身が興味の対象である。

(2) 雛型による生成では、あらかじめ部分的に穴のあいた文を用意しておき、必要に応じてその穴に単語を埋めて完全な文を生成する。このようにあらかじめ用意した(不完全な)文を雛型(template)とか canned

text という。各雛型にはその雛型を使うとどのようなゴールが達成されるかが記述されており、どの雛型を使うかは文脈におけるゴールと雛型に記述されたゴールを照合することによって決める。この方法の例として、プログラミング言語のコンパイラが出力するエラーメッセージを思い浮かべればよいだろう。コンパイラには各エラーに対してどのようなメッセージを出力すればよいかあらかじめ雛型として埋め込まれており、実際にそのエラーが起こった時点で、対応するエラーメッセージを出力する。この場合の雛型は最初から完全な文であることもある。雛型を使った生成は初期のエキスパートシステム⁽⁷⁾の出力部や、多くの対話システム⁽⁸⁾⁻⁽¹⁰⁾で使用されてきた。一般に、雛型を用いて生成した文の品質は高いが、あらかじめ用意した雛型以外の文は生成できないという限界がある。したがって、生成すべき文があらかじめ予想できる場合は有効だが、いろいろな文を柔軟に生成する必要がある場合には適していない。

(3) 意味構造からの生成では、意味ネットワーク、フレーム、論理式などの何らかの意味表現⁽¹¹⁾と文法や辞書などの統語的な情報を用いて文を生成する。一般には、意味表現を文法に沿って再帰的に分解することによって文を生成する。この分解操作は各部分表現が語彙項目に対応するくらい十分細くなるまで繰り返す。意味表現から文を生成する場合には、どのような統語構造や語彙を用いるかを決めなければならない。例えば、語の選択に関する代表的な研究として、概念依存関係⁽¹²⁾から文を生成する Goldman の BABEL⁽¹³⁾がある。また、複数の文からなる文章を生成する場合には、文章の結束性を保つための工夫も必要となる。文章の結束性については、この連載の第2回で詳しく述べる。意味表現からの生成では、高品質の出力を得ることは雛型による生成に比べ難しくなるが、より柔軟な生成が可能となる。

(4) 機械翻訳における生成では、原言語の解析結果を目標言語の構造へ変換し、生成を行う。意味表現からの生成と同様に、どのような統語構造や語を選択するかが大きな問題となるが、原言語と目標言語の間の変換規則でこの問題を解決することが多い⁽¹⁴⁾⁻⁽¹⁶⁾。なお、本稿では機械翻訳を特に意識した自然言語生成についてはこれ以上述べない。機械翻訳における生成に興味のある読者は、例えば、文献(2)(17)などを参照されたい。

(1)~(4)の範疇の研究の主な目的が1文を生成することであったのに対し、(5) 物語の生成では、複数の文からなる文章の生成を目的としている。文章を生成する

ためには、文章の中にどのような内容を含めるか、どのような順番で文を並べるかといった問題を解決しなければならない。すなわち、これらを決定するプランニングが重要になる。文章のプランニングは1980年代になって盛んに研究されるようになった問題である。

以上、佐藤の分類に沿って1970年代の自然言語生成に関する研究を概観した。これらのうち、(1)ランダムな生成、(2)雛型による生成は、現在ではあまり自然言語生成とは考えられていない。

1-2 1980年代の自然言語生成

Mannらは1982年に自然言語生成に必要な要素として、以下の四つをあげている⁽¹⁸⁾。

- ・理解しやすく言語学的に妥当性のある文法
- ・生成に必要な種々の情報を表現できる知識表現形成
- ・聞き手のモデル
- ・談話構造のモデル

実際、1980年代には、これらの項目について多くの研究が行われた。このうち、生成と文法の関係については5章(連載第3回)で述べる。また、聞き手のモデル、談話構造のモデルについては、1980年代の中心的研究テーマであった文章の結束性の問題にからめて4章(連載第2回)で解説する。自然言語生成を特に意識した知識表現形式に関する研究は残念ながら少なく、本稿では特に触れないが、一例をあげるなら、Jacobsが自然言語生成に必要な言語知識、世界知識を統一的に記述するための記述言語 Ace を提案している⁽¹⁹⁾。また、自然言語生成のアーキテクチャはいかにあるべきかという問題も多く研究者の関心事であった。3章では、自然言語理解の研究と比較しながら、この問題について述べる。6章(連載第3回)では、1990年代の自然言語生成研究につながると筆者らが考えている新しい研究の流れとして、語用論に基づく生成、生成における推敲の役割、漸次的生成の三つを取り上げる。

連載の構成

- | | |
|-----|------------------------|
| 第1回 | 1. はじめに |
| | 2. 自然言語生成は自然言語理解と何が違うか |
| | 3. 自然言語生成のアーキテクチャ |
| 第2回 | 4. 文から文章へ |
| 第3回 | 5. 生成と文法 |
| | 6. その他の話題 |
| | 7. 1990年代の自然言語生成に向けて |

2. 自然言語生成は自然言語理解と何が違うか

具体的な研究の解説に入る前に、本章では、まず自然言語生成の特徴を自然言語理解と比較しながら述べる。

自然言語生成の研究が自然言語理解の研究に比べ、大きく立ち後れていた理由として McDonald は次の二つの要因をあげている⁽³⁾。

- 研究者の絶対数が少なかった。
 - 自然言語の生成は理解よりも困難な問題である。
- そのほかに、
- 自然言語生成は応用プログラムの付属物としか認識されていなかった。
 - 応用プログラムの付属物程度ならアドホックな手法で十分であった。

などの要因も考えられる。事実、初期の自然言語生成の研究は、MYCIN*¹に代表されるようなエキスパートシステムの推論過程の説明文生成などが中心であり、このような目的には、雛型を使った生成で十分であった⁽⁷⁾。自然言語理解にも困難な問題が山積していることは周知の事実であり、理解より生成が困難であるという McDonald の主張は正しいとはいえない。しかしながら、自然言語の生成には、単に理解の逆の過程としてだけでは捉えられない特徴があることも確かである。

自然言語理解の目的は文章や発話からそれらが表す命題や発話者の意図を抽出することである。一方、生成の目的は命題や意図から文章や発話を作り出すことである。自然言語理解における重要な課題の一つが入力あいまい性をいかにして解消するかという問題であるのに対し、生成における一つの課題として、あいまい性のない、むしろ冗長な入力情報から必要な情報だけをいかに取捨選択するかという問題がある。

自然言語理解におけるあいまい性の解消は多くの候補から正しいものを選択する過程であるが、十分な文脈が与えられれば正しい選択が可能であることが多い。一方、自然言語生成では、ほとんどの場合、完全な正解というものはなく、よりもっともらしいものを選ぶことしかできない。もっともらしさの基準は一般に複数存在し、それらが互いに競合することも多い。自然言語生成では、この競合の解消が重要となる。この点では、自然言語の生成過程はプロダクションシ

テムによる問題解決過程に似ている⁽²⁰⁾。さらに、どの選択がより優れているかは、最終的な文章、発話が生成されても一般には明らかでない。この判断は人間の評価に依らざるを得ないが、現状では、生成システムの出力をどのように評価するかに関する合意も評価の方法論も存在しない。出力された文章の評価に関しては、機械翻訳⁽²¹⁾⁽²²⁾や推敲支援システム⁽²³⁾⁽²⁴⁾の研究の一部としていくつかの研究が行われている。

自然言語生成の研究を困難にしているもう一つの要因として、研究者が生成の出発点として合意できる対象を見い出せていないということがある。これは自然言語理解の研究者が、誰が見ても厳然と存在する文字列を出発点として研究できるのとは大きな違いである。生成の出発点を命題とするのか、発話者の意図とするのか、といった入力のレベルの問題に加え、たとえ同じレベルでも入力をどのように記述するかという点に関して研究者間の合意がないのが現状である。このように出発点があいまいで、その記述形式もはっきりしないために、研究を横並びで比較することが困難となっている⁽²⁵⁾。自然言語生成の研究におけるこのような困難さは、自然言語理解の研究の目標が漠然としていることの裏返しの結果でもある。すなわち、古くて新しい疑問「何をもって計算機で自然言語を理解できたというのだろうか」が生成研究の困難さの根源であるといえよう。情報の流れという意味では、理解と生成ではまったくの逆であるから、これは当然の帰結である。

以上、自然言語生成の特徴、困難さを自然言語理解と比較しながら述べた。まとめると、次のようになる。

- 自然言語理解の重要な課題であるあいまい性の解消は正解を目指す選択過程である。一方、自然言語生成の選択過程では複数の評価基準の競合の解消が重要となる。しかしながら、その評価基準を定めることは困難である。
- 自然言語理解では、出力として何を求めるかに関して、研究者間の合意があるとはいえないが、入力については文字列という共通な土台がある。それに対し、自然言語生成では入力に関する明確な共通の土台がないため、研究を横並びで比較することが難しい。

3. 自然言語生成のアーキテクチャ

自然言語理解の一般的なアーキテクチャを Fig. 1 (a)に示す⁽²⁶⁾⁽²⁷⁾。

理解と生成では情報の流れが逆であることを考えれ

* 1 MYCIN の説明文生成の歴史については文献 (7) の 17 章、説明文生成の詳細については同 18 章に詳しい。

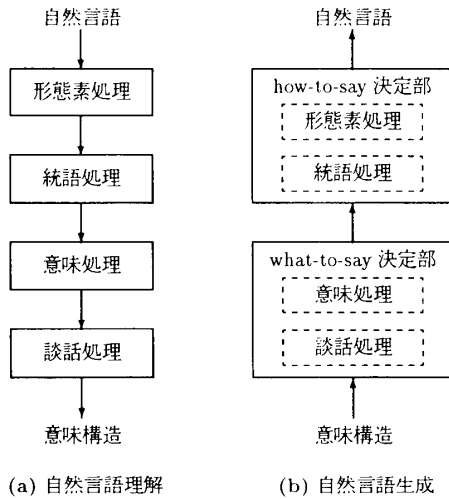


Fig. 1 処理のアーキテクチャ

ば、自然言語生成のアーキテクチャは逆方向、つまり、談話処理から始まり形態素処理に終わるとも考えられる。しかしながら、これまでの生成研究では、理解のアーキテクチャのような処理の切分け方ではなく、Fig 1 (b)のような形をとることが多い。これは2章でも述べたように自然言語の生成が、自分の持っている情報の中から出力すべき情報を選択するという特徴を持っているからである。

what-to-say 決定部では、システムが持つ情報の中から発話時の状況、聞き手のモデルなどを参考に出力すべき情報内容を選択し、話題の展開の概略を決める。一方、how-to-say 決定部では、主に統語情報、語彙情報を参照しながら統語構造や語の選択を行い、what-to-say 決定部が決定した情報内容を1次元の文字列として出力する。このときに指示詞の導入や省略なども行う。Fig. 1 (a)の自然言語理解のアーキテクチャと対応づけるなら、what-to-say 決定部が談話処理と意味処理に、how-to-say 決定部が統語処理と形態素処理にほぼ対応するであろう。しかしながら、そもそも切分け方の観点が異なるのでこの対応づけは厳密なものではない。これらの二つの機能を何と呼ぶかは研究者によって異なる場合がある。例えば、McDonald は生成の参照モデルとして、①話者の状況の認識、②状況から発話への写像、③発話、の3階層からなるモデルを提案しているが⁽²⁵⁾、このモデルでは①が what-to-say 決定部に、②と③が how-to-say 決定部に対応する。Danlos はそれぞれ、概念的決定 (conceptual decisions) と言語的決定 (linguistic decisions) と呼

んでいる^{*2(28)}。また、McKeown は What-to-say の決定を深層生成 (deep generation)、how-to-say の決定を表層生成 (surface generation) と呼んでいる⁽⁴⁾、このほか、what-to-say 決定部は戦略部 (strategic component)、あるいは文章プランナ (text planner)、how-to-say 決定部は表出部 (realization component)、言語部 (linguistic component) と呼ばれることもある。

このアーキテクチャに照らして歴史的な経緯を述べるなら、1970年代は雛型などを用いた非常に初歩的な how-to-say にする研究が中心であったといえる。つまり、エキスパートシステムなどの応用プログラムの出力からどのように自然言語を生成するかという問題が中心であり、応用プログラムの内容から出力すべき情報を選択したり、その順序を考慮したりすることにはほとんど注意が向けられていなかった。本格的な what-to-say の研究に関心が向けられたのは1980年代に入ってからである。しかし、両者をバランス良く融合した研究は少なく、ほとんどの研究はいずれかの側面に重点をおいたものであった⁽³⁰⁾⁻⁽³³⁾。

しかし、how-to-say と what-to-say の関係はもっと緊密であるべきだという主張もある⁽²⁸⁾⁽³⁴⁾⁻⁽³⁷⁾。what-to-say/how-to-say アーキテクチャでは、情報の流れが what-to-say 決定部から how-to-say 決定部への1方向に限られている。すなわち、what-to-say 決定部が生成する内容を決定し、how-to-say 決定部がそれをうまく文章化できなかつたら、バックトラックをして what-to-say 決定部からやり直すか、できる範囲で文章を生成するしかない。例えば、語の選択は多くの場合、how-to-say 決定部によって行われるが、語の選択に依存して、表現できる情報内容や情報の順序、すなわち、what-to-say が制約を受けることも十分考えられる。これについて、Danlos は次のような例をあげている⁽²⁸⁾。

今、「John が Mary の頭を銃で撃った」という行為と、その結果として「Mary が死んでいる」という状態を自然言語で出力したいとする。Danlos はこのような因果関係に関する表現では、原因と結果の組ではなく、因果関係とその原因となった行為の組を表現するほうがより良い表現になることを主張している。そこで、まず、①「撃つ」と「死んでいる」の因果関係と②「撃つ」行為という二つの要素の提示順序を what-to-say レベルで決定し、それから how-to-say のレベルで統語構造や語を選択するものとしよう。このとき、情報の提示順序は①と②のどちらを先に出力するかで2通りが考えられる。この①と②を表現するのにそれぞれ “kill” と “shot” という語を使うとすると、最終的な

* 2 Danlos 自身は文献 (28) の中で、概念的決定も言語的決定も how-to-say の決定であるとしているが、後述の例からすると概念的決定は what-to-say の決定に対応すると考えることができる。

出力を決定する残りの要素として、例えば、1文にするか2文に分けるか(2通り)、それぞれの動詞について態の選択(受動化した場合、動作主は明示するかどうかを含めて3通り)が考えられる。動詞の順を含めて考えると合計36通りの出力が考えられるが、このうち適切な出力は15通りしかない。例えば、以下の四つの出力のうち、(d)は不適切である。したがって、動作主を伴う受動文と動作主を伴わない受動文からなる文の並びでは、結果が行為に先行するか、または、行為が先行する場合は動作主を伴わなければならないことがわかる。

- (a) Mary was killed by John. She was shot.
- (b) Mary was killed. She was shot by John.
- (c) Mary was shot by John. She was killed.
- (d) *Mary was shot. She was killed by John.

この例は、統語構造などの how-to-say のレベルの決定も考慮しないと what-to-say レベルでの情報の提示順序が、適切に決定できないことを示している。以下では、what-to-say の決定と how-to-say の決定が相互に依存するアーキテクチャの例として、Appelt の KAMP と Hovy の PAULINE を紹介する。

3・1 KAMP

Appelt の KAMP は、話者のゴールを満たすための行為として発話をとらえ、発話生成をその行為のプランニングとして考えている⁽³⁴⁾⁽³⁸⁾。このプランニング過程において、ある行為がどれくらいゴールを満たすかは、その文脈で使用可能な言語的選択に依存する。したがって、ゴールを発話に具体化するまでの各過程において、ゴールを満足するための手段として使用可能な言語的手段も常に考慮しなければならない。このような考え方から KAMP では、what-to-say と how-to-say の明確な区別を避けている。Fig. 2 に KAMP のア

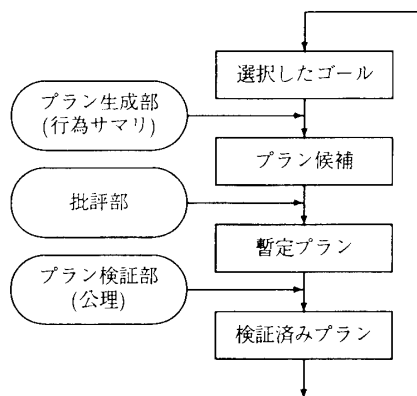


Fig. 2 KAMP のアーキテクチャ

ーキテクチャを示す。

KAMP の主な関心は、対象物の参照表現を生成するためのプランニングを公理化することである。KAMP への入力の話し手のゴールを表す命題の形式をしている。KAMP はゴールを満たすためにプランニングを行い、最終的に発話を生成する。例えば、話し手が「ポンプが台座から取り外された状態」をゴールとして持つとき、次の命題を入力として与える。

True (\neg Attached (ポンプ, 台座))

プランニングでは、話し手と聞き手の共有知識も参照する。例えば、このゴールを実現するためにレンチを使ってポンプを取り外すことを聞き手に依頼する際に、KAMP は聞き手がレンチのありかを知らないと思ったら「道具箱の中のレンチでポンプを外して下さい」という発話を生成するし、聞き手がレンチのありかを知っていると思ったら「レンチでポンプを外して下さい」という発話を生成する。KAMP はプランニングのための論理体系として Moore の知識と行為の論理⁽³⁹⁾を使い、行為を可能世界間の遷移関数として捉えている。KAMP のプランは、手続きネットワークと呼ばれるデータ構造で表現されている。手続きネットワークは、行為の時間的な順序関係を表す軸と、行為とその部分行為の階層関係を表す軸を持つ 2 次元のデータ構造である。ネットワークの各ノードにはそれぞれ可能世界が割り当てられており、その時点で成り立つ事象を表している。

KAMP は初期ゴールを与えられると、まず、そのゴールだけを含む手続きネットワークを生成し、以下の手続きを繰り返す。

- (1) ネットワーク中のノードで可能世界が割り付けられていないものに可能世界を割り付ける。
- (2) 各ノードのゴールを可能世界に照らして評価する。
- (3) 満足されていないゴールを選択する。
- (4) プラン生成部が行為サマリを参照してプランを選択し、ネットワークに追加する。
- (5) 批評部は(4)で選択されたプランとネットワーク中の他のプランとの関係を調べ、不都合があればプランの変更を要求する。
- (6) プラン検証部は(5)までで提案されたプランが本当にゴールを満たすかどうかを証明する。
- (7) (6)の証明が成功すれば、満たされていないゴールについて(1)からの処理を繰り返し、(6)の証明に失敗すればバックトラックを起こす。

このように KAMP は与えられたゴールをトップダウンに展開し、最終的に発話を生成する。暫定プランを

選択し、それを後で検証するという形式をとっているのは、すべてを厳密な証明に基づいて行うと探索空間が広がり、効率が悪いためである。行為サマリはゴールとそれを満たすための行為のリストであり、プラン生成部は行為サマリを参照することによって大まかな行為を選択する。ここでのプラン選択は局所的なものであり、選択したプランが大域的に妥当かどうかは批評部が評価する。

KAMPでは、統語的な情報も含めてすべてをプランとして表現しており、最終的に得られるプランとして単語列の出力が得られる。また、プランの階層を調べることにより、行為の包含関係を認識し、一つの発話で複数のゴールを達成することもできる。これはKAMPの大きな特徴である。

KAMPの欠点として以下のような点が指摘されている⁽⁴⁾⁽³²⁾⁽³⁵⁾⁽³⁶⁾。

- ・統語情報（いわゆる文法）がプラン全体に分散して表現されているために変更が困難である。
- ・すべてのプランが確定するまで出力ができない。
- ・プランニングのすべてのサブタスクを同一の枠組みで扱う必要がある。
- ・モジュールごとの設計ができない。

実際、初期のKAMPの文法は非常に限られたものであった。後にAppeltはこの点を改善するためにTELEGRAMと呼ばれる文法の枠組みをKAMPに導入している⁽⁴⁰⁾⁻⁽⁴²⁾。TELEGRAMの導入によって、KAMPのwhat-to-sayとhow-to-sayの完全融合型アーキテクチャは崩れ、プランナと文法が相互に情報を交換しながら生成が進むアーキテクチャに変更された。これは、次に述べるHovyのPAULINEと同様のアーキテクチャである。なお、TELEGRAMについては5章（連載第3回）で説明する。

3・2 PAULINE

HovyのPAULINEは、モジュールとしてはwhat-to-sayの決定とhow-to-sayの決定を明確に分けているが、両者がコルーチンの形で互いに情報を交換する方式をとっている⁽³⁵⁾⁻⁽³⁷⁾⁽⁴³⁾。Hovyによると、この方式には次のような利点がある。いずれも、3・1節で述べたKAMPの欠点を改善していることに注意してほしい。

- (1) what-to-say決定部とhow-to-say決定部を適切な枠組みで別々に開発できる。
- (2) 文法知識を考慮しながらwhat-to-sayを決定できる。
- (3) 心理的妥当性がある。

(3)の心理的妥当性は、人間が発話する際、what-to-sayを完全に決定しないうちに話し始めるという観察に基づくもので、これについては6章（連載第3回）でもう一度触れる。

Hovyは、生成に必要な種々の決定を広くプランニングと呼び、プランニングには性質の異なる次の2種類があることを指摘している。

トップダウンプランニング：

話し手のゴールをトップダウンに展開しながら話題を収集し、文章全体の大枠を決める。

ボトムアッププランニング：

トップダウンなプランニングだけでは、発話の状況に応じた柔軟なプランニングができない。ボトムアッププランニングでは、話題に関してシステムが実際に持っている知識に基づいて、文法知識を考慮しながら詳細な部分を決める。例えば、話し手の主張を聞き手に納得させるために、システムがあらかじめ持っている戦略にはない新たな話題を加えたり、効果的な語句を用いたりする。

PAULINEのアーキテクチャをFig.3に示す。PAULINEの生成過程は、話題収集、話題構成、表出の三つの過程から構成され、二つのモジュール（プランナと表出部）が相互に呼び合いながらこれらの過程を進める（Fig.3の2重矢印）。入力としては、フレームベースの表現形式で表した話題の集合と語用論的な情報を与える。プランナは、与えられた話題の集合から話題収集と話題構成を主としてトップダウンに行い、大まかなプランニングが終わると表出部を呼び出す。表出部は、文法知識に照らして話題を表層化するが、複数の選択肢やサブゴールの競合が発生すると、再びプランナを呼び出す（Fig.3の点線矢印）。プランナは、表出部に呼び出されると、選択や競合解消をボトムアップに行い、結果を表出部に返す。このように、PAULINEでは、プランナにボトムアッププランニングの能力を持たせることによって、what-to-sayの決定

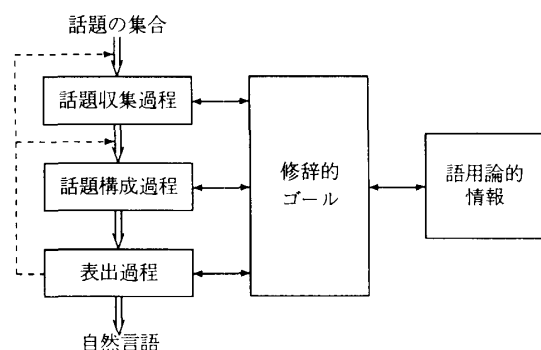


Fig.3 PAULINEのアーキテクチャ

と how-to-say の決定の相互依存を実現している。

語用論的情報は、発話状況に関するさまざまな情報であり、生成に必要な種々の決定を制御する。実際には、語用論的情報から変換した修辭的ゴールを各モジュールが参照するが (Fig. 3 の双方向矢印)、これについては 6 章 (連載第 3 回) で具体的に述べる。

謝 辞

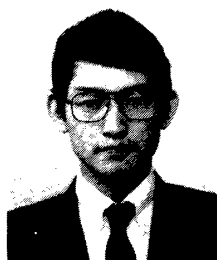
草稿に対する有益なコメントをいただきました査読者の方々、ICOT の池田光生氏、横浜国大の田村直良氏、日本 IBM の長尾確氏、東工大の奥村学氏に感謝します。また、本稿を執筆する機会を与えてくださいました東工大の田中穂積教授に感謝いたします。

◇ 参 考 文 献 ◇

- (1) 上原邦昭：文生成，人工知能学会 (編) 人工知能ハンドブック，pp.274-286，オーム社 (1990)。
- (2) 辻井潤一：機械翻訳における文章の生成，人工知能学会誌 Vol.4, No.6, pp.645-651 (1989)。
- (3) McDonald, D.D. : Natural language generation, In Shapiro, S. (ed.), *Encyclopedia of Artificial Intelligence*, pp.642-654, Wiley-Interscience (1987)。
- (4) McKeown, K. R. and Swartout, W. R. : Language generation and explanation, In Zock, M. and Sabah, G. (ed.) : *Advances in Natural Language Generation*, chapter 1, pp.1-51, Ablex Publishing Corporation, (1988)。
- (5) 佐藤泰介：文の生成，自然言語処理技術と言語理論，第 4 章，電子技術総合研究所調査報告第 205 号，電子技術総合研究所 (1981)。
- (6) Pereira, F. C. N. and Warren, D. H. D. : Definite Clause Grammars for language analysis—A survey of the formalism and a comparison with Augmented Transition Networks, *Artif. Intell.*, Vol.13, No.3, pp.231-278 (1981)。
- (7) Buchanan, B.G. and Shortliffe, E.H. : *Rule-Based Expert Systems : The MYCIN Experiments of the Stanford Heuristic Programming Project*, Addison-Wesley (1984)。
- (8) Bobrow, D.G., Kaplan, R.M., Kay, M., Norman, Thompon, D.A. and Winograd, T. : GUS, a frame-driven dialog system, *Artif. Intell.*, Vol.8, No.2, pp.155-173 (1977)。
- (9) Weizenbaum, J. : ELIZA : A computer program for the study of natural language communication between man and machine, *Communications of the ACM*, Vol.9, No.1 (1966)。
- (10) Winograd, T. : *Understanding Natural Language*, PhD Thesis, Academic Press (1972)。
- (11) Ringland, G. A. and Duce, D. A. : *Approaches to Knowledge Representation*, John Wiley & Sons Inc. (1988)。
- (12) Schank, R. C. : *Conceptual Information Processing*, Volume 3 of *Fundamental Studies in Computer Science*, North-Holland (1975)。
- (13) Goldman, N. : Conceptual generation, In Schank, R. (ed.), *Conceptual Information Processing*, North-Holland (1975)。
- (14) 熊野 明，天野真家：英日機械翻訳システムの訳文生成について，情報処理学会自然言語処理研究会，NL 40-6, (1986)。
- (15) 堤泰治郎，筧 義郎，原田雅弘，西嶋智恵子，堤 豊：英日機械翻訳システム SHALT における日本語生成，情報処理学会自然言語処理研究会，53-5 (1986)。
- (16) 鈴木雅美，橋本和夫，野垣内出，榎 博史：英日機械翻訳における補文の変換と生成，情報処理学会 自然言語処理研究会，NL 53-6, (1976)。
- (17) 野村浩郷，田中穂積 (編)：機械翻訳，*bit* 別冊，共立出版 (1988)。
- (18) Mann, W.C., Bates, M., Grosz, B., McDonald D., McKeown, K.R. and Swartout, W. : Text generation, *American Journal of Computational Linguistics*, Vol.8, No.2, pp.62-69 (1982)。
- (19) Jacobs, P.S. : KING : A knowledge-intensive natural language generator, In Kempen, G. (ed.), *Natural Language Generation*, chapter 15, pp.219-230, Martinus Nijhoff Publishers (1987)。
- (20) Patten, T. : *Systemic text generation as problem solving*, Cambridge University Press (1988)。
- (21) 吉村裕美子，平川秀樹，天野真家：自然な文章生成のための規範，情報処理学会 自然言語処理研究会，NL 74-3 (1989)。
- (22) 長尾 真，辻井潤一：Mu プロジェクトにおける日英翻訳結果の評価，情報処理学会 自然言語処理研究会，NL 47-11 (1985)。
- (23) 菅沼 明，倉田昌典，牛島和夫：日本語文章推敲支援ツール『推敲』における否定表現の抽出法，情報処理学会論文誌，Vol.31, No.6, pp.792-800 (1990)。
- (24) 林 良彦，菊井玄一郎：日本文推敲支援システムにおける書き換え支援機能，人工知能学会全国大会，pp.461-464 (1990)。
- (25) McDonald, D.D., Vaughan, M.M., Pustejovsky, J. D. : Factors contributing to efficiency in natural language generation, In Kempen, G. (ed.), *Natural Language Generation*, chapter 12, pp.159-182, Martinus Nijhoff Publishers (1987)。
- (26) 辻井潤一：自然言語理解の歴史と現状，情報処理，Vol.30, No.10, pp.1142-1149 (1989)。
- (27) 田中穂積：自然言語解析の基礎，彦業図書 (1989)。
- (28) Danlos, L. : Conceptual and linguistic decisions in generation, In *the Proc. of the Int'l Conf. on Computational Linguistics*, pp.501-504 (1984)。
- (29) Hovy, E.H., McDonald, D.D. and Young, S.R. : Current issues in natural language generation : An overview of the AAAI workshop on text planning and realization, *AI MAGAZINE*, FALL : 27-29 (1989)。
- (30) Jacobs, P.S. : KHRED : A generator for natural language interfaces, *Computational Linguistics*, Vol.11, No.4, pp.219-242 (1985)。
- (31) Mann, M.C. : An overview of the Penman text generation system, In *the Proc. of the National Conf. on Artif. Intell.*, pp.261-265 (1983)。
- (32) McDonald, D. and Pustejovsky, J.D. : Description-directed natural language generation, In *the Proc. of the Inter'l Joint Conf. on Artif. Intell.*, pp.799-805

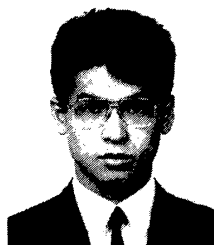
- (1985).
- (33) McKeown, K.R. : The TEXT system for natural language generation : An overview, In *the Proc. of the Annual Meeting of the Association for Computational Linguistics*, pp.113-120 (1982).
- (34) Appelt, D.E. : Planning natural-language referring expressions, In *the Proc. of the Annual Meeting of the Association for Computational Linguistics*, pp.108-112 (1982).
- (35) Hovy, E.H. : *Generating Natural Language under Pragmatic Constraints*, Lawrence Erlbaum Associates (1988).
- (36) Hovy, E.H. : Integrating text planning and production in generation, In *the Proc. of the Inte'l Joint Conf. on Artif. Intell.*, pp.848-851 (1985).
- (37) Hovy, E.H. : Pragmatics and natural language generation, *Artif. Intell.*, Vol.43, No.2, pp.153-197 (1990).
- (38) Appelt, D.E. : Planning natural-language referring expressions, In McDonald, D.D. and Bolc, L. (eds.), *Natural Language Generation System*, chapter 3, pp. 69-97, Springer-Verlag (1988).
- (39) Moore, R.C. : *Reasoning about Knowledge and Action*, Techsical Report 191, SRI, International Artificial Intelligence Center (1980).
- (40) Appelt, D.E. : *Planning English Sentences*, Cambridge University Press (1985).
- (41) Appelt D.E. : TELEGRAM : A grammar formalism for language planning, In *the Proc. of the Inte'l Joint Conf. on Artif. Intell.*, pp.595-599 (1983).
- (42) Appelt, D.E. : TELEGRAM : A grammar formalism for language planning, In *the Proc. of the Annual Meeting of the Association for Computational Linguistics*, pp.74-78 (1983).
- (43) Hovy, E.H. : Two types of planning in language generation, In *the Proc. of the Annual Meeting of the Association for Computational Linguistics*, pp.179-186 (1988).

 著 者 紹 介



徳永 健伸 (正会員)

1961年生まれ。1983年東京工業大学工学部情報工学科卒業。1985年同大学院理工学研究科修士課程修了。同年、(株)三菱総合研究所入社。1986年東京工業大学大学院博士課程入学。1987年より同大学工学部情報工学科助手。自然言語処理、知識表現に関する研究に従事。情報処理学会、日本ソフトウェア科学会、認知科学会、計量国語学会各会員。



乾 健太郎

1967年生まれ。1990年東京工業大学工学部情報工学科卒業。現在、同大学院修士課程在学中。自然言語生成に関する研究に従事。特に文章の推敲に関心を持つ。情報処理学会会員。