

1980年代の自然言語生成[†]

Natural Language Generation in 1980s

徳永健伸, 乾健太郎
東京工業大学 工学部 情報工学科

TOKUNAGA TAKENOBU AND INUI KENTARO
Department of Computer Science, Tokyo Institute of Technology
{take,inui}@cs.titech.ac.jp

目次

1	はじめに	1
1.1	1970年代の自然言語生成	1
1.2	1980年代の自然言語生成	2
2	自然言語生成は自然言語理解と何が違うか?	3
3	自然言語生成のアーキテクチャ	4
3.1	KAMP	6
3.2	PAULINE	7
4	文から文章へ	8
4.1	結束性	9
4.2	what-to-say レベルの結束性を保つために	9
4.2.1	スクリプト的談話構造モデルとプランニング	10
4.2.2	抽象的談話構造モデルとプランニング	13
4.2.3	話し手の意図と聞き手のモデル	15
4.3	how-to-say レベルの結束性を保つために	16
4.3.1	参照表現	16
4.3.2	接続表現	18
5	生成と文法	19
5.1	MUMBLE	20
5.2	TELEGRAM	22
5.3	PHRED	24
6	その他の話題	25
6.1	語用論的制約にもとづく文章生成	25
6.2	推敲機能の導入	26
6.3	漸次的生成	28
7	おわりに	30

[†]本稿は人工知能学会誌, Vol. 6, No. 3-5 に連載された同名の解説記事をまとめて加筆訂正したものである.

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いだすことができる。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ひとつであった。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おこなわれていたものの、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ところ軽視されてきた感がある。関連領域である心理言語学でもこのような傾向がみられ、Leveltによれば人間の言語理解や言語獲得の研究に比べ、言語生成の研究は継子扱いであったという [93]。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かかわることはよく知られている。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、本稿では1980年代におこな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 [18, 23, 106, 115]、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採り入れるように努めた。表題もふくめて「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年代におこなわれた自然言語生成に関する研究を佐藤の分類 [14] に沿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に分類している。

- (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、開始記号から書き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き換えをおこなう。DCG [126] 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は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が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 [49] の出力部や、多くの対話システム [46, 146, 149] 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測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 [131] 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。たとえば、語の選択に関する代表的な研究として、概念依存関係 [133] から文を生成する Goldman の BABEL [60] 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、4章で詳しく述べる。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、生成をおこなう。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 [8, 25, 38]。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たとえば、文献 [23], [35] などを参照されたい。

(1) から (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4つをあげている [98]。

- 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式
- 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おこなわれた。このうち、生成と文法の関係については5章で述べる。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 [79]。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の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、語用論にもとづく生成、生成における推敲の役割、漸次的生成の3つを取り上げる。

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2つの要因をあげている [106].

- 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、MYCIN¹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 [49]。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とら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話し手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ひと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、生成におけるひとつの課題として、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ではなく、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ステムによる問題解決過程に似ている [125]。さらに、どの選択がよりすぐれているかは、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えないが、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 [6, 22] や推敲支援システム [19, 37] の研究の一部としていくつかの研究がおこなわれている。

自然言語生成の研究を困難にしているもうひとつの要因として、研究者が共通に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、誰がみ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。最近では、生成の出発点は話し手の意図であるべきだという点で多くの研究者の合意は得られているようであるが [54]、話し手の意図とは何か?、それはどのように記述されるのか? など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 [109]。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「何をもち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。

¹MYCIN の説明文生成の歴史については [49] の 17 章、説明生成の詳細については同 18 章に詳しい。

- 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、出力として何を得るかにに関して、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図 1 (a) に示す [24, 31].

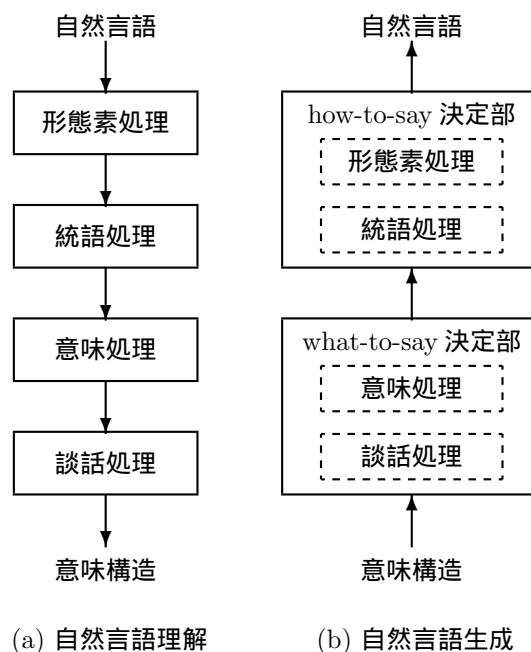


図 1 処理のアーキテクチャ

理解と生成では、情報の流れが逆であることを考えれば、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り分け方ではなく、図 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おこない、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。この時に、指示詞の導入や省略などもおこなう。図 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、そもそも切り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 2 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。たとえば、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 [109]、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 [55]²。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 [115]。Lvelt は what-to-say の決定をおこなう処理装置として概念活性器 (conceptualizer), how-to-say の決定をおこなう処理装置として表層化器 (formulator) をそれぞれ仮定した生成のアーキテクチャを提案している [93]。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 に関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、両者をバランスよ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 [77]。1980年代に提案された代表的なシステムもほとんどがこの古典的なアーキテクチャを踏襲している [80, 97, 107, 112]。

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 [43, 55, 68, 69, 72]。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。たとえば、語の選択は多くの場合、how-to-say 決定部によっておこなわれるが、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 [55]。

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方がよりよ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2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。この時、情報提示の順序は(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。この因果関係と行為を表現するのに“kill”と“shot”という語を使うとすると、最終的な出力を決定する残りの要素として、たとえば、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。動詞の順序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。たとえば、以下の4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。

²Danlos 自身は [55] 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。

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 [42, 43]. 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 の明確な区別は避けている. 図 2 に KAMP のアーキテクチャを示す.

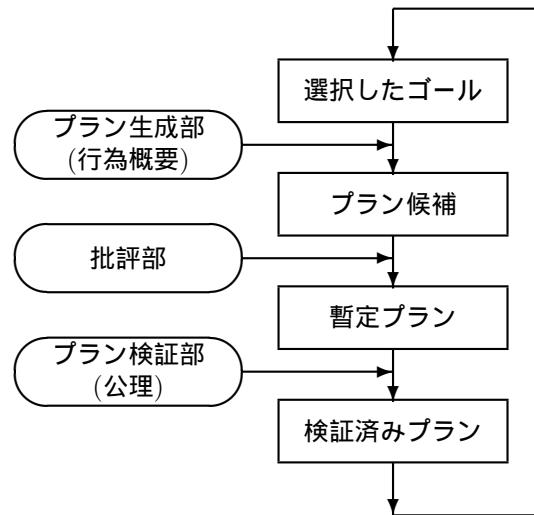


図 2 KAMP のアーキテクチャ

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おこない、最終的に発話を生成する. たとえば、話し手が「ポンプが台座からとりはずされた状態」をゴールとして持つとき、次の命題を入力として与える.

$\text{True}(\neg\text{Attached}(\text{ポンプ}, \text{台座}))$

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. たとえば、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「工具箱の中のレンチでポンプをはずして下さい」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「レンチでポンプをはずして下さい」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 [122] を使い、行為を可能世界間の遷移関数としてとらえている. KAMP のプランは、手続きネットワークという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もとづいておこな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、1つの発話で複数のゴールを達成することもできる。これらは 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 [68, 69, 107, 115].

- 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を KAMP に導入している [41, 44, 45]. 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、両者がコーチンの形で互いに情報を交換する方式をとっている [68, 69, 72, 74]. 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、前節で述べた 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) 心理学的妥当性がある

(3) の心理学的妥当性は、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もとづくもので、これについては6章でもう1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もとづいて、文法知識を考慮しながら詳細な部分を決める。たとえば、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無い新たな話題を加えたり、効果的な語句を用いたりする。

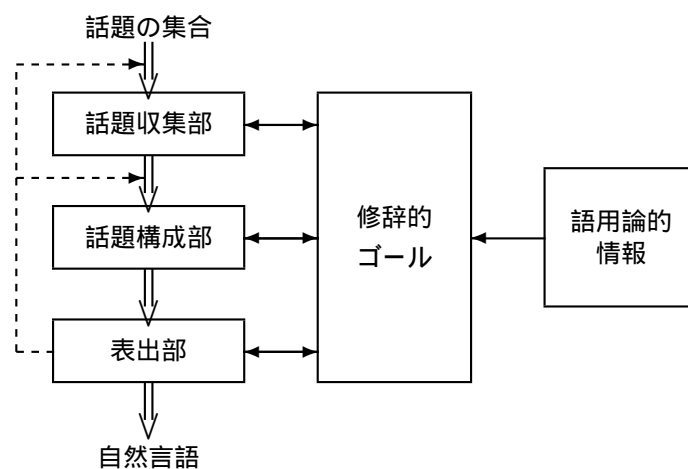


図3 PAULINE のアーキテクチャ

PAULINE のアーキテクチャを図3に示す。PAULINE の生成過程は、話題収集、話題構成、表出の3つの過程から構成され、2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(図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おこない、大まかなプランニングが終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(図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おこない、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辞的ゴールを各モジュールが参照するが(図3の双方向矢印)、これについては6章で具体的に述べる。

4 文から文章へ

1章で述べたように、1970年代の生成に関する研究は1文を生成することが主な目的であった。複数の文からなる文章の生成は1980年代における生成研究の中心的なテーマであったといえよう。質の高い文章を

生成するためには、結束性、あいまい性、冗長性など、文章の質に関するさまざまな要素を考慮する必要がある [27]。これらの要素の中でも特に多くの研究者によって取り上げられたのが結束性 (cohesion) である。文章の結束性は、文章が聞き手にとって理解しやすいことの必要条件であると考えられている [27]。本章では、結束性の高い文章を生成するために必要な考え方とその実現について述べる。

4.1 結束性

文章の結束性 (cohesion) は、文の集合が 1 つの文章として成り立つために必要な文や語の間のつながりを指し、Halliday らの言語学的研究で導入された概念である [62]。文、節、語などの文章の構成要素間のつながりには、因果関係のような意味的つながりや、照応関係のような表層的つながりがある。

Halliday らは、英語において結束性を保つために用いられる表現をまとめ、参照 (reference)、代用 (substitution)、省略 (ellipsis)、語彙結束 (lexical cohesion)、論理的接続 (logical connectives) の 5 種類に分類している。これらの表現は大きく 2 つのタイプに分けることができる。1 つは、既出のものが後出の部分によって受け継がれていく場合で、参照、代用、省略、語彙結束がこれに当たる。もう 1 つは、前後の内容が意味的に許容される何らかの関係によって結びつけられる場合で、接続詞や接続詞的な副詞によって表現される。論理的接続がこのタイプに当たる。cohesion に似た用語に coherence という用語があるが、Hobbs [65, 66] によると、coherence は節と節の間の意味的関係の妥当性を指す概念で、Halliday らの分類のうち論理的接続に近い概念である。

自然言語生成の立場から結束性を考えると、次の 2 つのレベルに分けることができる [9]。

what-to-say レベル: 文章中の表現には陽に現れない意味的なつながり。

how-to-say レベル: 文章中の表現に陽に現れる表層的なつながり。

複数の命題から 1 つの文章を構成する場合、話題を時間的順序で提示したり、原因を提示した後で結果を述べるなど、聞き手が意味的なつながりを容易に連想できるような順序に話題を並べる必要がある。たとえば、次の例文 (a) では 3 つの文の間に意味的な関係が読みとれるが、例文 (b) では文と文の間に意味的な関係が読み取れず、全体として 1 つの文章を構成しているとは言いがたい。

(a) 雨がやんだ。風もやんだ。日がさしてきた。

(b) 一生懸命勉強した。風がやんだ。ポチが死んだ。

what-to-say レベルでは、話題と話題の時間関係や因果関係などを考慮して、複数の話題をどのような論旨展開のもとに 1 列に並べるかが問題になる。

what-to-say レベルで結束性があっても、話題間の意味的なつながりを正確に相手に伝えるためには、さまざまな言語表現による努力が必要である。how-to-say レベルでは、論旨の展開や同一対象の指示を聞き手にわかりやすく示すためにどのような言語表現を用いるかが問題となる。たとえば、論旨展開を明確にするために「したがって」や「ところが」などの接続詞を用いたり、既出の登場人物を人称代名詞などで指示したりするのがこれに当たる。

Halliday の結束性の分類のうち参照、代用、省略、語彙結束はいずれも how-to-say レベルの結束性を向上させるための言語手段である。論理的接続は what-to-say レベル、how-to-say レベル両方の結束性に関係する。以下では、what-to-say レベルと how-to-say レベルそれぞれについて結束性を向上させるための主な手法を紹介する。

4.2 what-to-say レベルの結束性を保つために

what-to-say を決めるプランニングは、図 4 に示すように話し手の意図から文章の談話構造 (discourse structure) を構成する過程としてとらえることができる。

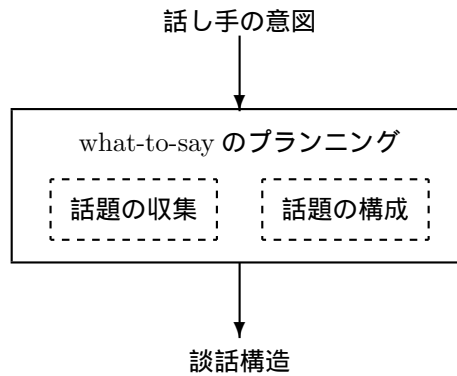


図4 what-to-say のプランニング

話し手の意図は、「ある事について聞き手を説得する」、「ある命題の正しさを聞き手に信じさせる」など、話し手が文章を通じて達成しようとする目的である。談話構造は Grosz らによると次の3種類の構造で表せる [61].

意図構造: 文章中の各話題がどのような効果を意図しているか。また、文章全体が何を目的としているか。

修辞構造: 各話題が他の話題とどのような意味的關係にあるか。どのような順序で話題を述べているか。

注意構造: 談話の各時点でどの対象物、属性、イベントに注意の焦点が当たっているか。

これら3つの構造のうち、修辞構造がプランニングの対象としてもっとも強く意識されてきた。談話構造の構成を修辞構造の構成と同義に扱っている研究も多い。プランニングの対象として意図構造や注意構造をどの程度扱っているかについては研究者によって隔たりがあるが、本章では、what-to-say 決定過程で得られる中間生成物を一括して談話構造と呼ぶ。

談話構造の構成には次のような2種類の処理が必要になる。

話題の収集: どの話題を聞き手に提示するか。

話題の構成: どのような順序で話題を提示するか。

話題の収集は、話し手が聞き手に伝えたい内容や聞き手が関心を持っている話題などを収集する処理である。話題の構成は、「一般的説明 → 具体例」、「理由 → 結論」など、聞き手が理解しやすい順序に話題を並べる処理である。以下では、談話構造を構成するためのこれら2つの処理を指してプランニングと呼ぶ。

what-to-say レベルの結束性をテーマとする研究には、何らかの談話構造のモデルを用意し、そのモデルにもとづいてプランニングをおこなうものが多い。自然言語生成の分野でこれまで用いられてきた談話構造のモデルは、記述の方法という観点からスクリプト的談話構造モデルと抽象的談話構造モデルとに大きく分けることができる [68]。本節では、それぞれの談話構造モデルについて簡単に説明し、それをを用いたプランニングの手法を紹介する。また、聞き手が何に興味を持っているか、あるいは何を知っているかに依存して話すべき話題やその構成が決まることを重視する立場から、聞き手のモデルを扱った研究も多い。聞き手のモデルを考慮した生成については、話し手の意図のプランニングとからめて後述する。

4.2.1 スクリプト的談話構造モデルとプランニング

スクリプト的談話構造モデルは、文章の談話構造をある程度具体的なシナリオとしてあらかじめ定義したものである。われわれが実際に書いた文章を分析してみると、文章の目的や分野に応じてある程度固定化した談話構造を用いていることがわかる。たとえば、新聞記事では、事件の概略を述べた後で具体的内容を

述べ、個々の情報については時間的順序に沿って述べるという構成を用いることが多い [28]。また、ある概念を聞き手に説明する場合には、まずその上位概念を示し、次に下位概念や属性などを列挙するのが一般的な戦略である [110, 111]。このような分析にもとづいて、実際の文章の談話構造から一般化できる部分を抽出したものがスクリプト的談話構造モデルである。

スクリプト的談話構造モデルを用いたシステムには、McKeown の TEXT [110, 111]、Kukich の Ana [90, 91] などがある。これらのシステムでは、目的や話題に応じてあらかじめスクリプトを用意し、そのスクリプトに沿って話題の収集と構成をおこなう。その他にも、Schank のスクリプト [134, 136] を構造化した MOP [136] を用いて話題の構成をおこなう石崎 [78]、内田ら [28] の研究などがある。

TEXT はデータベースに対する質問応答システムの自然言語インタフェースである。TEXT は、スキーマと呼ばれる答え方のシナリオをあらかじめ持っており、複数あるスキーマの中から質問の種類に応じて 1 つ選択し、選択したスキーマに沿って文章を生成する。つまり、TEXT ではスキーマを選択することによって話題の収集と構成を同時におこなう。スキーマはスクリプト的談話構造モデルの典型的な例である。TEXT のプランニングについては後述する。

Ana は株価の数値データを入力として一日の株式の動きを報告するシステムである。Ana では、人間が実際に書いたレポートの分析から得られた文章の談話構造がプロダクション・ルールの中に埋め込まれており、そのルールによって話題の収集、構成をおこなう。たとえば、初めに株価全体の動きを概説し、次に平均株価の動きを具体的な数字を示して述べるなどの構成が定義されている。

石崎や [78] や内田ら [28] の研究では、話題の時間関係、因果関係、階層関係にもとづいて話題構成をおこなっている。石崎は、MOP [136] を用いて話題間の時間順序を推論し、話題を時間の流れに沿って提示するシステムを開発している。また、内田らは、新聞記事の話題構成を分析することによって、話題の時間関係や因果関係を記述したシーン木から文章中での話題の順序関係を定める規則を抽出している。内田らの分析によれば、同じ新聞記事でも英語の新聞と日本語の新聞では話題の提示順序が異なる。これは一種の対象領域に依存する知識の利用といえよう。

スクリプト的談話構造モデルを用いると、われわれが日常用いる文章構成を容易にシステムに与えることができる。しかし、システムの適用分野や文章の目的、話題に応じてそれぞれ別々のスクリプトを用意する必要があるため、談話構造モデルの記述の負担が大きい、システムが適用分野に依存しやすいなどの欠点もある。この問題に対し、McKeown らは、TEXT で用いたスクリプト的談話構造モデルから分野に依存しないより一般的な談話構造モデルを抽出し、それを用いてシステムの柔軟性を高める手法を提案している [114]。

ここで、スクリプト的談話構造モデルを用いるプランニングの例として McKeown の TEXT を見てみよう。

TEXT は、軍用乗物や武器に関するデータベースに対する質問応答システムの自然言語インタフェースで、データベース中の概念の説明、概念と概念の比較などについて英語で答えることができる。TEXT には、生成する文章の談話構造の標準パターンがあらかじめいくつか用意されている。この標準パターンはスキーマと呼ばれ、回答文は入力された質問に対応するスキーマに沿って生成される。たとえば、図 5 のような質問に対し、TEXT は概念の説明をおこなうためのスキーマとして〈同定〉スキーマを選択する。

質問 (入力): What is a ship?

回答 (出力): (1)A ship is a water-going vehicle that travels on the surface. (2)Its surface-going capabilities are provided by the DB attributes DISPLACEMENT and DRAFT. ...

図 5 TEXT における質問応答の例

図 6 に〈同定〉スキーマの一部を示す。図 6 はスキーマを ATN(Augmented Transition Network [151]) の形式で表現したものである。

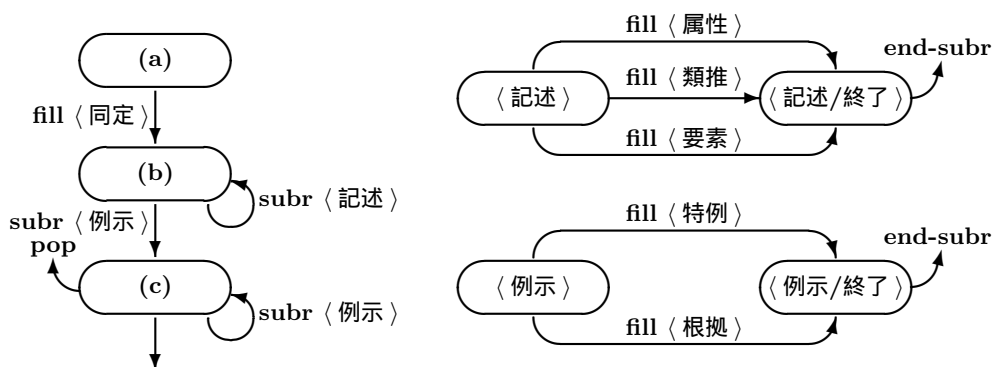


図6 〈同定〉スキーマ

それぞれの弧のラベルはシステムがとるべき行動を規定している。たとえば、“fill 〈同定〉”は修辭的述語〈同定〉を満足せよという命令で、この命令により、質問された概念をその上位概念を使って説明することになる。“subr 〈記述〉”は〈記述〉というラベルのついたサブネットワークへ制御を移せという命令であり、制御はサブネットワークをたどった後、“end-subr”によって元の状態にもどる。

システムは、まず、質問の中心である概念 ship に関係する概念や属性を収集し、データベースの部分集合を作る。次に、図6のノード(a)を出発し、ネットワークをたどりながら修辭的述語の指示にしたがって生成すべき話題をデータベースの部分集合から取り出し、順に文章として表出する。この例の場合、システムはまず修辭的述語〈同定〉にしたがって回答(1)を生成する(図5)。このとき、システムの状態はノード(b)である。ノード(b)からはサブネットワーク〈記述〉または〈例示〉に遷移できるので、次にたどることのできる弧は、“fill 〈属性〉”、“fill 〈類推〉”、“fill 〈要素〉”、“fill 〈特例〉”、“fill 〈根拠〉”のいずれかである。ただし、データベース中に対応する情報が存在しない弧は考慮されない。今、この理由で“fill 〈特例〉”が候補から落されたとしよう。システムは、次に起こす行動として残りの4通りの修辭的述語の中から1つを選択する。それぞれの修辭的述語が選択された場合の生成例を示すと次のようになる。

〈類推〉: The ship carries guided projectiles.

(概念 ship から連想される事柄について述べる)

〈要素〉: There are 5 types of ships in the database: aircraft carriers, frigates, ocean escorts, cruisers, and destroyers.

(ship の下位概念を列挙する)

〈属性〉: The ship has DB attributes MUXIMUM SPEED, PROPULSION, FUEL, DIMENSIONS, SPEED DEPENDENT RANGE and OFFICIAL NAME.

(ship の属性を列挙する)

〈根拠〉: Its surface-going capabilities are provided by the DB attributes DISPLACEMENT and DRAFT.

(ship が水に浮くことのできる根拠を説明する)

このように、スキーマは、話題の収集や構成を一意に定めるわけではなく、ある点まで生成が進んだ時点で、次に述べる話題の選択肢を列挙する役割を果たす。TEXT では、次に続ける話題の選択基準として、Sidner の焦点モデル [139, 140] を用いている。Sidner は、文章で現在話題の中心となっている概念を焦点 (focus) と呼び、焦点の移動に関するいくつかの制約が自然言語理解に有効であることを主張している。McKeown は、Sidner が示した焦点移動に関する制約を生成用に拡張し、次のような4つのパターンを定義している。焦点移動のパターンには相対的な優先度が与えられており、(1) から (4) の順に優先度が低くなる。

- (1) 前文で新たに導入された項目について話す (そこに焦点を移す)
- (2) 同一の対象について話し続ける (焦点を移さない)
- (3) 以前に述べた話題に焦点をもどす
- (4) 現在の焦点と関連する対象について話す

TEXT は、この焦点移動パターンにしたがうように話題を展開し、結束性のある自然言語生成を実現している。この例では、〈類推〉、〈要素〉、〈属性〉を選ぶと焦点は ship のままであり、〈根拠〉を選ぶと焦点が ship から surface-going capabilities に移動する。後者は前文で導入された項目に相当するので、システムは、焦点移動パターンの優先度にもとづいて〈根拠〉を選択し、図 5 の回答 (2) を生成する。

このように、TEXT では、スクリプト的談話構造モデルとしてスキーマを定義し、スキーマに焦点の移動に関する制約を適用することによって、what-to-say レベルの結束性を保ったプランニングを実現している。

4.2.2 抽象的談話構造モデルとプランニング

文章を構成する節、文、段落などの各ブロックは理由と結論、説明と具体例などの意味的な関係でつながっている。これらの意味的な関係の種類は無限にあるわけではなく、いくつかの決まった関係しか用いられない。抽象的談話構造モデルは、ブロック間の意味的な関係を表現するプリミティブの集合として定義される。抽象的談話構造モデルを用いたプランニングでは、このプリミティブから文章の談話構造を再帰的に構成する。

代表的な抽象的談話構造モデルに Mann らが提案した RST (Rhetorical Structure Theory: 修辞構造理論) [95, 100] がある。RST は、英語を母語とする人に結束性があると認められた文章を分析することによって、隣接するブロック間の意味的な関係を〈根拠〉、〈継続〉、〈詳細化〉など約 20 種類に分類し、形式化したものである。個々の関係は RST スキーマと呼ばれ、それぞれ図 7 のような 2 つのブロック間の 2 項関係として定義される。

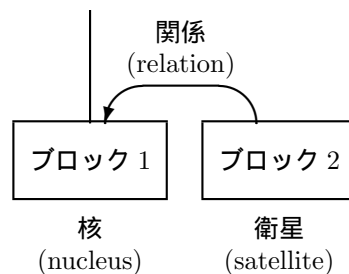


図 7 RST スキーマの一般形

2 つのブロックはそれぞれ核 (nucleus)、衛星 (satellite) と呼ばれ、たとえば、〈根拠〉スキーマであれば、核の内容に対する根拠として衛星が核を支持するという関係を表す。核と衛星はさらにそれぞれ別の RST スキーマで表現することができ、文章の談話構造は木構造として表現される。たとえば、次の段落の談話構造を分析すると図 8 のような木になる³。

³各節の先頭に付加した番号 [n] はブロック番号を表す。

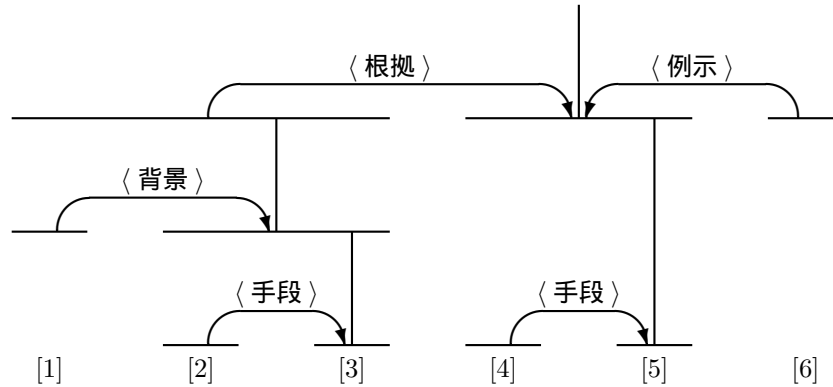


図8 修辞構造の分析例

[1]RST スキーマは文章の目的や話題に依存しない基本的な意味的關係に対応しているため、[2]RST スキーマを構造的に組み合わせることによって[3]多様な談話構造が表せる。したがって、[4]プランニングにRST を用いると、[5]適用分野に対して比較的柔軟なシステムを作ることができる。[6]RST を用いたプランニングに関しては Hovy [67, 76] や Moore [120, 121] の研究などがある。

これらの研究の主な関心は談話構造を構成する過程にあり、完成した談話構造から表層の文章を生成する過程についてはあまり議論されていない。談話構造の表層化については、談話構造の木を左から右へたどり、1つの葉に対して1つの文を生成し、葉と葉の意味的關係に対応する特定の接続語を挿入するという手法をとる場合が多い。また、これらの手法では、what-to-say を決めてから how-to-say を決めるため、3章で述べたような what-to-say と how-to-say の相互依存の問題が残る。現状では、以下の具体例で見られるようにRST スキーマをトップダウンに展開して談話構造を構成するため、how-to-say を考慮しながらのプランニングは難しいといえる。完成した談話構造からの表層化に関する研究としては、Scott らが文の埋め込みなどに関するいくつかのヒューリスティックを用いて読みやすい文章を生成する手法を提案している [135]。

最後に、RST を用いたプランニングの例として Hovy の研究 [71] を紹介する。

Mann らによって提案された初期のRST では、RST スキーマを自然言語で定義していた。Hovy は、RST スキーマの操作性を高めるために Cohen らの発話行為演算子 (speech act operator)[51] を用いている。例として〈目的〉スキーマの定義を図9に示す。

目的

核に関する制約:

1. (BMB S H (ACTION ?act-1))
2. (BMB S H (ACTOR ?act-1 ?agt-1))

衛星に関する制約:

1. (BMB S H (STATE ?state-1))
2. (BMB S H (GOAL ?agt-1 ?state-1))
3. (BMB S H (RESULT ?act-1 ?act-2))
4. (BMB S H (OBJ ?act-2 ?state-1))

意図した効果:

1. (BMB S H (BEL ?agt-1 (RESULT ?act-1 ?state-1)))
2. (BMB S H (PURPOSE ?act-1 ?state-1))

図9 目的スキーマの例

各 RST スキーマは、「核に対する制約」、「衛星に対する制約」、「意図された効果」からなる。「意図された効果」は、そのスキーマを用いることによって達成される状態 (意図) を定義している。S, H はそれぞれ話し手, 聞き手を表し, 発話行為演算子 BMB は S と H がある命題について相互に信じていることを表す。「?」で始まる項は変数を表し, 行為や状態のインスタンスが入る。

Hovy のアルゴリズムは, プログラムの改良をおこなうエキスパートシステムの自然言語インタフェースに組み込まれており, Mann らの開発した Penman [97] の上で実現されている。システムは, エキスパートシステムから出力すべき情報の集合と話し手の意図を入力として受けとり, その意図を達成するような談話構造を構成し, 文章を出力する。

談話構造の構成は RST スキーマをトップダウンに展開することによっておこなう。たとえば, 話し手が次のような意図を持っていたとしよう。

(BMB S H (PURPOSE 〈行為 1〉 〈状態 1〉))

システムは, この意図を用いて RST スキーマを検索し, 意図とマッチする「効果」を持つ RST スキーマを 1 つ選択する。ここでは図 9 に示した〈目的〉スキーマを選択したとしよう。〈目的〉スキーマを用いるためには核と衛星に対する制約が全て満たされなければならない。そこで, システムは制約を満たすように談話構造を展開する。具体的には, 制約を直接満足させるような情報が入力中に見つかれば, それを文として出力する。そのような情報が得られない場合は, その制約を新たな意図として, その意図を達成させる別の RST スキーマを探し, それを用いて談話構造を展開する。談話構造の展開は全ての制約が満たされるか, 入力情報を使い果たした時点で終了する。

一般に, ある意図を達成させる RST スキーマは複数あるため, このアルゴリズムでは談話構造が一意に定まらない。さらに, どの談話構造が適当なのかに対する基準も明らかではない。Hovy [76] らの後の研究では, RST と McCoy らが提案した焦点木 [103] を組み合わせることによってこの問題の解決を試みている。すなわち, 談話の焦点が焦点木上をうまく移動できるように, どの RST スキーマによって核や衛星を展開すべきかを決定するのである。Hovy の生成アルゴリズム [76] は, McKeown の TEXT においてスキーマを RST スキーマに置き換え, Sidner の焦点モデルを焦点木に置き換えたものと考えることができる。すなわち, TEXT のスキーマよりも抽象的な RST スキーマを用いることによって話題の展開の柔軟性を, 焦点木を用いることによって焦点の移動の柔軟性を実現している。

また, Hovy は, 複数の段落からなる大きな文章のプランニングを RST のみでおこなうのは困難であることを指摘し, その解決策として, 文章全体を構成するレベルでスクリプト的談話構造モデルを用い, 各段落の詳細なプランニングで抽象的談話構造モデルを用いる方法を示唆している [75]。

4.2.3 話し手の意図と聞き手のモデル

対話型のシステムでは, 聞き手の持っている興味に応じて話題を選択したり, 聞き手が既に知っていることについては説明を加えないなどの配慮が必要である。話し手は各時点での聞き手の信念や関心の焦点について推測しなければならない。「話し手が推測する聞き手の心の状態」を記述したものを聞き手のモデルという。これまでに多くの生成システムが聞き手のモデルを採り入れている。

Wilensky らの UC [147] は, UNIX のコマンド体系について説明するシステムであるが, 聞き手のモデルを利用し, 聞き手の知識の専門度に応じて説明内容を変更する機能を持っている。垣内らのシステム ASSIST [4, 34] は, 聞き手との対話を介して聞き手のモデルを動的に変化させ, 聞き手の知っている概念とそうでない概念とを比較しながら, どこまで詳しく説明するかを決定する。また, Paris [123, 124], McCoy [101, 102] は, 聞き手が何に興味を持っているか, どの程度詳しい知識を持っているかに応じてスクリプトを動的に選択する研究をおこなっている。これらの研究では, 聞き手のモデルを談話の進行にしたがって動的に変化させ, 談話の各時点で聞き手の状態に適した文章の生成を実現している。

対話では、話し手の発話を聞き手が理解できなかつたり、信じなかつたりした場合、話し手の発話について聞き手が聞き返すことも起こる。聞き手からの「聞き返し」に対しシステムが適切に答えられるためには、システムの発話の各部分がどのような効果をねらったものなのか、すなわち意図構造（本節の冒頭参照）を明示する談話構造が必要である。スクリプティック談話構造モデルを用いて構成した談話構造は、「聞き手に X を知らせる」、「聞き手に X を説明する」といったトップレベルの意図と修辞構造との関係しか明示しない。このため、スクリプティック談話構造モデルは複雑な相互作用が起こる対話には適さないことが Moore らによって指摘されている [120, 121]。

一方、RST に代表される抽象的談話構造モデルを用いると、それぞれの RST スキーマが達成する効果（意図）を明示するため（図 8 参照）、発話の各部分と意図の関係が明らかになる。RST スキーマは、文章の各ブロック間の意味的な関係を表現するものであるが、4.2.2項で見たように上位の意図から下位の意図への展開規則として捉えることもできる。この意味で RST スキーマは Allen らが意図のプランニングに用いたプランスキーマ [40, 94] と共通する側面を持っている。Moore らは、プランスキーマと RST スキーマを融合させることによって、話し手の意図の構造と修辞構造の密接な関係を保持しながら対話のプランニングをおこなう手法を提案している [120, 121]。Moore らの手法では聞き手からの聞き返しが起こるような複雑な対話も扱えることが示されている。田淵らも、伝達目的と呼ばれる話し手の意図を論旨展開の戦略によって下位の伝達目的に展開し、そこから RST を用いて修辞構造を生成するシステムを開発している [32]。論旨展開の戦略は Moore らのプランスキーマにほぼ対応する。田淵らのシステムは、伝達の目的や論旨展開の戦略に依存して、同じ伝達内容から異なる修辞構造を持つ文章を生成することができる。

このように、結束性を考慮した文章や対話のプランニングに関する研究では、表層に現れるブロック間の単なる意味的な関係から話し手の意図の構造と修辞構造の関係へと関心が移ってきている。すなわち、「何をどのような順序で言うか」だけでなく「どうしてそれを言うのか」を考えることが重要視されてきている。話し手の意図と聞き手のモデルは自然言語生成に不可欠なものとして定着したといえよう。意図や行為のプランニングからの生成を扱った研究には、上で紹介したものの他に Mellish [117], Dale [52], Cawsey [50] などがある。

4.3 how-to-say レベルの結束性を保つために

結束性を考慮した文章生成の研究は、4.1節で述べたような what-to-say レベルのものが多く、how-to-say レベルの結束性に関する研究は比較的少ない [124]。文章に how-to-say レベルの結束性を与える主な言語表現には参照表現と接続表現がある [9, 12, 27, 62, 78, 91]。参照表現は代名詞や指示詞など特定の対象を指示するための言語表現であり、接続表現は接続詞などブロックとブロックとの意味的な関係を示す言語表現である。これらの表現やその評価基準はいずれも言語に大きく依存するため、特定の言語に限定した研究が多い。たとえば、英語では既出の対象に対して代名詞化が頻繁に起こるが、日本語では代名詞化の頻度は比較的低い。これまでに開発された自然言語生成システムのほとんどがこれらの表現手法の一部を備えているが、それを第一義的に扱った研究は少ない。また、言語学的研究が生成システムに適用できるほど十分に形式化されていないために、これらの表現を生成するための規則にはアドホックなものも多い。本節では、参照表現と接続表現の生成がどのように実現されているかについて述べる。

4.3.1 参照表現

参照 (reference) には大きく分けて談話内照応 (endophora) と談話外照応 (exophora) がある [62, 13]。談話内照応は文章中に既に登場した対象を指し、談話外照応は、談話がおこなわれている場面や談話の参加者があらかじめ持っている知識など談話の環境にある対象を指す表現である。談話内照応には、既に出てきた対象を指す前方照応 (anaphora) と、今後登場する予定の対象を指す後方照応 (cataphora) があるが、生成の分野では後方照応はほとんど扱われていない。

前方照応には次のような種類の言語表現がある [13].

代名詞:

I have a daughter. She is five years old.

指示代名詞:

John will probably win the match. That will please Mary.

語の繰り返し:

きのうわたしは学校の近くのそば屋で一人の男にあった。男は突然自分は私と同じ大学で同級だったものだと言い出した。

上位語:

A 社では新しい制ガン剤の開発に成功したと発表した。同社の発表によれば、新薬の原料はなんとタバコであるという。

省略:

山本さんは給料をそのまま奥さんに渡す。鈴木さんは ϕ そのまま母親に渡す。

TEXT(4.2.1項参照)では、参照表現の生成にも焦点情報を用いている。たとえば、図5で示した出力例では、回答(2)で代名詞“its”が用いられている。これは、前文で焦点が当たっていた対象(この例では“ship”)を代名詞で指示することができるという規則によるものである [110].

Grosz らは、談話において修辞構造と注意構造(4.2節冒頭参照)が参照表現に影響することを指摘し、その関係の形式化を試みている [61]. たとえば、現在焦点が当たっている対象は代名詞化されやすいが、この代名詞化は修辞構造におけるある程度大きなブロックの境界を越えては起こらない。Dale は、Grosz らの理論を拡張し、談話構造を用いた参照表現の生成を実現している [52, 53].

Gabriel のシステム *Yh* [59] では、名詞句を生成する際、(1) 同一の対象を指す名詞句が文章の前方にあるか、(2) 類似性の高い対象を指す名詞句が前方にあるかを調べ、それぞれ次のような処理をおこなう。(1) の場合、2つの名詞句が比較的近くにあれば、それぞれの名詞句を含む文を1つの文に統合することを試みる。2つの名詞句が離れている場合には、前置詞句などの修飾句を付加することによって名詞句がどの対象を指しているかを明確にする。(2) の場合、2つの対象の相違点を際立たせるような修飾句を名詞句に付加して、あいまい性を取り除く。

上位語による参照表現を扱った例に Kukich [90, 91], 石崎 [78] の研究などがある。次の例は Kukich の Ana の生成例であるが、斜体の名詞句はいずれも同じ対象を指示している。

The Dow Jones average of 30 industrials gained 19.13 points in the afternoon and finished with a 16.28 point gain at 1092.35. *The transportation measure* tacked on sizable gains, as *the utility index* edged higher.

日本語では、「～は」で提示される主題⁴が2文にわたって継続するとき、2番目の文で主題の省略が起こりやすいことが言語学の立場から指摘されている。これにもとづいて、主題の省略表現を扱うシステムも提案されている [9, 27, 78].

談話外照応を扱ったものには、スクリプト的談話構造モデルと聞き手のモデルを用いることによって物理的対象を適切に指示する名詞句の生成をおこなう Paris [123, 124] らの研究がある。Appelt の KAMP (3.1節参照) も談話参加者の個別知識、共有知識を用いて適切な談話外照応の生成を実現している。

Reiter は、聞き手の知識に応じて、照応表現に必要な情報の量だけでなく情報の質が変わる現象を扱っている [129]. たとえば、「鯨が出るので遊泳は危ない」ということを聞き手に伝えたい場合、鯨についての知識を持っている聞き手には発話 (a) が、鯨を知らない聞き手に対しては発話 (b) が適している。

⁴ここで述べる日本語の主題に関しては [7, 15, 17] を参照。

(a) この辺は鮫が出る。

(b) この辺は危険な魚が出る。

仮に、鮫を知っている聞き手に (b) を発話したとする。聞き手は、「もし、話し手が鮫を指して言ったとすれば、「鮫」という表現を直接使うはずだ。わざわざ「危険な魚」と言ったのは、出るのが鮫でなく、話し手も知らない魚だからだろう」という誤った推論をしてしまう。Reiter は、個々の聞き手が持つ知識の違いを形式的に記述し、聞き手に誤った推論をさせない照応表現の生成アルゴリズムを示している。

4.3.2 接続表現

文章の中では、節と節、文と文、あるいは段落と段落など、ブロック間の意味的關係を聞き手に明確に提示するためにさまざまな接続表現が用いられる。接続詞などを用いてブロックとブロックの意味的關係を明示することは聞き手の理解の助けになるが、逆に接続表現の多用は文章の冗長化を招くおそれもある。また、接続表現の選択は、複数の話題をいくつの文に分けて述べるかというような問題とも直接的な關係にあり、文章のスタイルなども考慮しなければならない。

TEXT では、各修辭的述語に対し用いるべき接続表現を定義している。たとえば、修辭的述語〈特例〉が選択されると、システムは“for example”という接続表現とともに文を生成する。RST を用いる Penman でも同様の手法としており、各 RST スキーマの中に核と衛星をつなぐ接続表現の候補をあらかじめ定義し、その情報にもとづいて適当な接続表現を生成する。このように、広い意味で宣言的に記述された談話構造モデルを用いるシステムでは、談話構造からの情報をもとに接続表現の選択をおこなう。一方、談話構造が規則中に埋め込まれている Ana では、隣接する話題の時間に関する属性などを比較することによって、適切な接続表現の生成を実現している。

ブロック間の意味的關係には接続表現を用いて明示する必要のないものもある。たとえば、〈詳細化〉という關係を持つ次の 2 つの命題を表層化すること考えよう。

(a) My watch is a SEIKO.

(b) My watch is new.

この場合、核となる命題 (a) の中に衛星である命題 (b) を埋め込んで、“My watch is a new SEIKO.” という文を作ることができる。Scott らは、RST で表現された談話構造からどの話題とどの話題を取り出して 1 つの文を構成するかについて議論し、埋め込み文や等位接続などを生成するヒューリスティックを提案している [135]。

日本語における接続表現の生成には高橋 [10, 27] らの研究がある。高橋らは、小学校低学年の国語の教科書の分析から得られた接続表現生成の方略を汎用文章生成システム上に実現している。接続表現生成モジュールへの入力、文フレームと呼ばれる単文ごとの意味構造が順序づけられて並べられたもので、各文フレームには前の文フレームとの連接關係が明示されている。モジュールは規則にしたがって接続表現を生成する。

高橋らは、接続詞の多用がかえって読み手に冗長な印象を与えることも指摘している。たとえば、次の例では、(b) に比べ (a) は冗長である。また、(c) は句点結合の例で、接続關係を明示的に表現していないにもかかわらず、読み手は容易に接続關係を推測することができる。

(a) 空が暗くなった。そして、風が吹いてきた。そして、雨が降ってきた。

(b) 空が暗くなって、風が吹いてきた。そして、雨が降ってきた。

(c) 雨がやんだ。日がさしてきた。

この例のように、接続詞などを用いて意味のつながりを表層で明示することと、冗長性を減らすこととは、トレードオフの関係になる場合がある。最適な接続表現は、説明文、物語文など文の種類や、丁寧、普通、口語調など文のスタイルにも大きく依存する。高橋らは、これらの点を考慮して、たとえば、接続関係(継起)に対する生成規則を次のように定義している。

1. 接続詞の種類は文の種類、スタイルでおおまかに限定される
2. 直前の生成文で句点結合が用いられていなければ句点結合、用いられていれば接続詞を生成する
3. 主題が継続していて主題が省略されている場合には「そして」、「て」を、主題が変化している場合には「すると」、「と」を、時間の変化が特に文フレームで示されている場合は「それから」を用いる。

内田らは、接続表現の要不要が聞き手の知識に依存することについて議論している [28]。内田らによると、いくつかの話題を 1 列に並べるとき、それらの間の意味的關係を話し手と聞き手が既に共有知識として持っているならば、接続表現は一般に不要になる。また、阿部らは、接続関係の種類と接続詞の必要性との相関関係について調べ、逆接や順接の関係を表す接続詞の必要性が他より高いことを実験的に示している [1]。

5 生成と文法

1.2節で述べたように、Mann らは、理解しやすく言語学的に妥当性のある文法を生成システムの持つべき要素としてあげている [98]。文法は、自然言語理解の研究では統語処理をおこなう上で重要な役割を担っている。直観的には、自然言語生成においても自然言語理解で使用されている文法を逆方向に使えるとよさそうである。実際、論理文法 [39] の分野を中心に、解析器と生成器で文法を共有しようという研究がおこなわれている [30, 64, 92, 132, 137, 138, 143]。しかし、これらの研究における生成器への入力には論理形式があるいはそれに近いもので、生成器にはすべての情報が細部まで決定されて渡されるのが普通である。したがって、このような研究の成果を自然言語生成にそのまま利用しようとすると、必然的に 3章で説明したような what-to-say 決定部と how-to-say 決定部をはっきり切り分けたアーキテクチャを採用することになる。現在の生成研究の関心のひとつが what-to-say 決定部と how-to-say 決定部の間でどのようにして緊密に情報を交換するかという点であることを考えると、これらの研究の成果を採り入れるためには何らかの工夫が必要である。

これまでの自然言語生成における文法の扱い方を大きく分類すると次のようになる。

- (1) 文法を手続きとしてプログラムの中に直接埋め込む
- (2) 文法を宣言的な規則として用意する
- (3) 文法を語彙の中を含めた句語彙 (phrasal lexicon) を用いる

(1) の代表的な例として、McDonald の MUMBLE [105, 107] や Appelt の初期の KAMP [43, 42] がある。文法を手続きとしてプログラム中に直接埋め込むと、実行効率はよくなるが、修正や拡張が困難になるという欠点がある。

(2) の例としては Kay の FUG [84, 85] を使用した McKeown の TEXT [110, 111, 112]、Appelt の TELEGRAM [41, 44, 45]、Halliday のシステミック文法 [63] を使用した Mann らの Penman [96, 97]、Patten の SLANG [125]、田淵らのシステム [32] などがある。自然言語理解では、文法の枠組として句構造文法の枠組を使用することが多いが、自然言語生成では、FUG やシステミック文法が多く使われている。その主な理由として、次の 2 点が考えられる。

- 句構造文法の対象が基本的に単文に限られているのに対し、4章で述べたように自然言語生成の主な興味が文を越えた文章にあること

- FUG やシステミック文法の枠組では生成で重要となる選択操作が扱いやすいこと

FUG は、統語的な情報とともに、話題や焦点などの文を越えた情報を属性として容易に文法中に取り込める特徴を持ち、属性の単一化によって適切な表現を選択することができる [113]。ただし、FUG の単一化は一般に計算コストが高いため実現には注意しなければならない。

システミック文法は文法的素性を節点に持つ一種の弁別ネットワークの集合によって文法を表現する。各々のネットワークをシステムという。生成は、

- 選択器という特別な手続きが環境と対話しながらシステムをたどり、適切な文法的素性を選択する
- 選択した素性の集合から文を生成する

という 2 つの段階を経ておこなう。システミック文法に関しては文献 [148] 6 章に詳しい。

(3) の例としては Kukich の Ana [90, 91], Jacobs の PHRED [80, 81], Hovy の PAULINE [68, 70, 72, 73] などがある。句語彙は初期の生成システムで使われていた雛型をより抽象化したものと考えられる。句語彙の背景には、人間も定型句を組み合わせる言語を使っている側面があるという観察がある。Kukich は、人間がよく使う言い回しを句語彙としてシステムに与えることと流暢な自然言語の生成との関連性を指摘し、句語彙の有用性を主張している [91]。句語彙をどのように組み合わせるかはシステムによって方法が異なり、Kukich のように規則を使うものもあれば、Jacobs や Hovy のように手続きを使うものもある。

以下では、3 つのシステムを例としてとりあげ、文法の扱い方について説明する。

5.1 MUMBLE

McDonald の MUMBLE はプランナによって生成された表出仕様 (realization specification) から表層構造 (surface structure) を経て自然言語を生成する。表層構造は Chomsky の変形文法 [128] でいう表層構造に近いもので、形態的には統語木の形をしている。表層構造の葉以外のノードは非終端記号であり、葉ノードは単語、表出仕様、スロットのいずれかである。スロットは、表出仕様を付加することのできる特殊なノードである。MUMBLE の生成過程では、以下の 3 つのモジュールが互いに他のモジュールを呼び出しながら表層構造を成長させていく。

- 付加モジュール
- 表出モジュール
- 句構造実行モジュール

実行制御にはデータ駆動方式を採用しており、次にどのモジュールを呼び出すかは表層構造の状態によって決まる。

付加モジュールは、表層構造のスロットに表出仕様を付加する。たとえば、1 文を生成し始める状態では、[sent] というスロットの下に 1 文として生成する表出仕様を付加することになる。

表出モジュールは、あらかじめ用意された表出仕様と表層構造の対応関係を参照して、文脈上適切な表層構造を選択する。こうして得られる表層構造は句のレベルから語のレベルまでさまざまである。得られた表層構造の葉には表出仕様とともに文法的な手続きが付加される場合もある。これらの手続きは知識表現言語のデモンのように適当なタイミングで起動され、種々の変形操作を実行する。

句構造実行モジュールは表層構造をトップダウンに左から右へ探索する。探索中、非終端記号に出会うとそのまま再帰的に探索を続けるが、葉に到達すると、

- 葉が語の場合は、形態素処理をおこないその語を出力する。

- 葉がスロットの場合は, 付加モジュールを呼び出して, 適切な表出仕様をスロットに付加する.
- 葉が表出仕様の場合は, 表出モジュールを呼び出して, 表層構造を成長させる. この時に付加手続きが起動される場合もある.

```
(r-spec1
  (r-spec2  color-of (door-3 red))
  (r-spec3  part-of (door-3 house-1))
  (r-spec4  condense-on-property (r-spec2 r-spec5 red))
  (r-spec5  color-of (gate-4 red))
  (r-spec6  part-of (gate-4 fence-2)))
```

図 10 表出仕様の例

McDonald [105] の例を用いて MUMBLE の生成過程の概略を説明しよう. 図 10 に表出仕様の例を示す. r-spec1 は以下の段落の第 2 文 (斜体) に相当する表出仕様である.

This is a picture of a white house with a fence around it. *The door of the house is red, and so is the gate of the fence.* There is a mailbox across the street in front of the fence, and ...

ここで, condense-on-property という関係はプランナが付加したもので, r-spec2 と r-spec5 は性質 red を共有した形で表層化せよという意味である. このように, 表出仕様は, color-of や part-of といった対象間の関係の他に, どのような修辞構造を用いるかという言語的な情報も含んでいる. 表出モジュールはこの言語的な情報を手がかりに対応する表層構造を選択する.

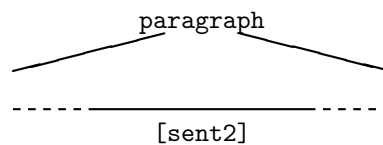


図 11 表層構造 (1)

今, 第 1 文を生成した時点で図 11 のような表層構造ができていたとしよう. 句構造実行モジュールはこの表層構造を探索し, スロット [sent2] を見つけ, 付加モジュールを呼び出す. 付加モジュールは r-spec1 をこのスロットに付加し, 制御を句構造実行モジュールに戻す. 句構造実行モジュールは同じ場所に表出仕様 r-spec1 を見つけ, 表出モジュールを呼び出す. 表出モジュールは r-spec4 に注目し, たとえば, 図 12 のような節の等位構造を選択し, 句構造実行モジュールを呼び出す.

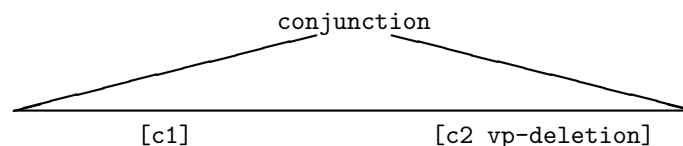


図 12 表層構造 (2)

句構造実行モジュールは付加モジュールを呼び出し, スロット [c1], [c2 vp-deletion] にそれぞれ, r-spec2, r-spec5 を付加し, 図 13 のような表層構造を作り出す. 同様にして, 3 つのモジュールが互いに呼び合いながら最終的にすべての葉が語として表出された時に 2 番目の文の生成が終了する.

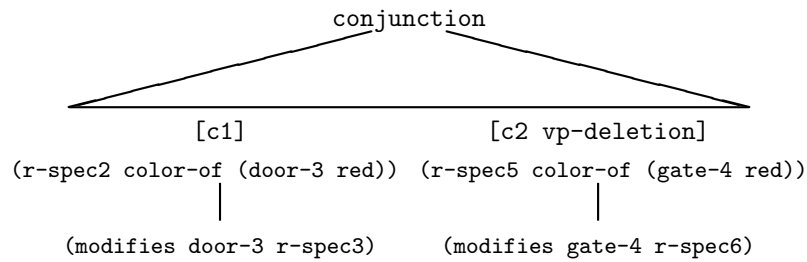


図 13 表層構造 (3)

MUMBLE では、このように文法的な知識が表出モジュールという手続きの形で実現されており、関係節の関係詞、等位接続のコンマ、接続詞などは変形操作によって挿入される。

5.2 TELEGRAM

3章で紹介した Appelt の KAMP の初期の版には、

- 文法知識がプランナに分散して表現されているために変更/修正が困難
- プランナを効率的に制御することができない

という欠点があった。Appelt はこれらの点を改善するために TELEGRAM という文法の枠組を提案している [41, 44, 45]。

TELEGRAM は FUG [84, 85] の次のような欠点を改善した拡張となっている。

- FUG の機能記述 (FD: Functional Description) は、意味のレベルから表層の文を直接生成できるほど十分な能力を持っているわけではなく、素性を詳細に決定してからでないと生成を開始できない。
- 単一化だけで生成をおこなうと効率が悪い。

TELEGRAM では FD 中にプランナのための情報を素性として記述できる。単一化器は文法的な素性については通常の単一化をおこなうが、プランナのための素性を見つけると制御をプランナに移す。プランナはその素性情報を手がかりにその部分を文法的素性に詳細化し、再び単一化器に制御を戻す。このように単一化器とプランナの相互作用によって最終的な FD を作り出し、その FD 中の語順を表す PATTERN 素性をもとに語句を 1 次元に並べ、文を生成する。

KAMP のプランニングは以下のように 4 階層になっているが、TELEGRAM はこのうち、(2) と (3) を受け持っている。

- (1) 発話内行為 (情報伝達/約束/要求/感謝)
- (2) 表層発話行為 (大まかな統語構造の決定)
- (3) 概念活性行為 (対象の指示表現の決定)
- (4) 発話行為 (表層化)

Appelt の例 [44, 45] を使って、TELEGRAM の動きの概略を説明しよう。

今、ある人 (AGT1) がある人 (AGT2) にねじ回し S1 が道具箱 TB1 の中にあることを教えようとしている場面で、これをどのように言語化するかを考えよう。KAMP はレベル (1) のプランニングによって次のようなプランをたてる。

Do(AGT1, Inform(AGT2, Location(S1)=TB1))

次に、レベル (2) のプランニングによって、このプランから宣言文に対応する FD1 を選択する。これを文章 FD という。TELEGRAM は文法 (FD の集合) を参照しながら、この文章 FD を詳細化する。

$$\left[\begin{array}{l} \text{CAT} = \text{S} \\ \text{SUBJ} = [\dots] \\ \text{VERB} = [\dots] \\ \text{COMP} = \left[\begin{array}{l} \text{CAT} = \text{PP} \\ \text{PREP} = [\text{LEX} = \text{IN}] \\ \text{POBJ} = \left[\begin{array}{l} \text{CAT} = \text{NP} \\ \text{REFERENT} = \text{TB1} \end{array} \right] \end{array} \right] \end{array} \right] \quad (\text{FD1})$$

FD1 が選択された時点で、制御は単一化器に移り、単一化器は FD1 と文法 FD を単一化しようとする。ここで、前置詞句 (PP) の目的語の素性 POBJ の値である名詞句 FD2 について考えよう。

$$\left[\begin{array}{l} \text{CAT} = \text{NP} \\ \text{REFERENT} = \text{TB1} \end{array} \right] \quad (\text{FD2})$$

単一化器は文法中から FD3 を見つけ、FD2 と単一化しようとする。

$$\left[\begin{array}{l} \text{CAT} = \text{NP} \\ \text{PATTERN} = (\text{DET HEAD QUAL}) \\ \text{DET} = [\dots] \\ \text{HEAD} = [\text{CAT} = \text{N}] \\ \text{QUAL} = [\dots] \end{array} \right] \quad (\text{FD3})$$

FD2 中の素性 REFERENT はプランナのための素性で、TELEGRAM の単一化器はこれを認識して、プランナを呼び出す。プランナは対話参加者 (この場合は AGT1 と AGT2) の共有知識を参照して、概念活性レベルのプランニングをおこなう。今、共有知識の中に AGT2 は工具箱が机の下にあることは知らないが、机の位置は知っているという知識があったとすると、プランナは次のような FD4 を作り出す。

$$\left[\begin{array}{l} \text{CAT} = \text{NP} \\ \text{DESC} = (\text{Toolbox}(\text{TB1}), \text{Under}(\text{TB1}, \text{TABLE1})) \end{array} \right] \quad (\text{FD4})$$

DESC もプランナのための素性なので再びプランナが起動され、今度はレベル (2) のプランニングによって次のように詳細化する。

$$\left[\begin{array}{l} \text{CAT} = \text{NP} \\ \text{DET} = \left[\begin{array}{l} \text{NBR} = \text{SG} \\ \text{SUBCAT} = \text{DEF} \end{array} \right] \\ \text{HEAD} = [\text{LEX} = \text{TOOLBOX}] \\ \text{QUAL} = \left[\begin{array}{l} \text{CAT} = \text{PP} \\ \text{PREP} = [\text{LEX} = \text{UNDER}] \\ \text{POBJ} = \left[\begin{array}{l} \text{CAT} = \text{NP} \\ \text{REFERENT} = \text{TABLE1} \end{array} \right] \end{array} \right] \end{array} \right] \quad (\text{FD5})$$

この FD5 は文法中の FD3 と単一化できる。以下、同様にして単一化器とプランナが協調して最終的な FD を作り出す。FD5 の部分 FD からは、たとえば、“the toolbox under the table” のような表現が生成される。

このように TELEGRAM では、FD にプランナのための注釈を付加し、適宜プランナを呼び出すことによって、意味レベルの素性を含む FD からの生成を可能にしている。また、プランナと単一化器の協調によって無駄な単一化を防ぎ、効率を上げている。

5.3 PHRED

Jacobs の PHRED は UNIX のコンサルテーションシステム UC [147] の生成部であり、句語彙を用いて生成をおこなう。句語彙は表層文の難型であるパタンとそのパタンに対応する概念表現の組 (PC ペア) となる。この句語彙は同じ UC の解析部である PHRAN と共有している。すなわち、UC では、共通の知識を用いて生成と解析をおこなっている。図 14 に PC ペアの例を示す。PC ペアはパタン、概念記述、制約記述の 3 つの要素から構成されている。

```
パタン: (agent) "remove" (obj) ["from" (container)]
概念: (状態変化
      (対象 ?X)
      (状態 場所)
      (初期状態 (inside-of 対象 ?Y))
      (最終状態 (not (inside-of 対象 ?Y))))
制約: 時制=(tense (position 2))
      X=(position 3)
      Y=(position 5)
```

図 14 PC ペアの例

図 14 において概念記述の “?” で始まる記号は変数を表し、これらの変数は制約記述によってパタンの構成素に対応付けられる。たとえば、“X=(position 3)” により、概念記述の変数 X とパタンの 3 番目の要素 (obj) が対応していることがわかる。制約記述にはその他にも種々の制約が記述できる。図 14 の最初の制約記述は、この PC ペアによって規定される文の時制がパタンの 2 番目の構成素、すなわち主動詞の時制と一致しなければならないことを表している。パタンは構成素のデフォルトの順序を規定するが、変形パタンと呼ばれる特殊なパタンを持つ PC ペアによって構成素の順序を入れ換えることもできる。変形パタンは変形文法 [128] の変形規則に似ており、これによって受動化などのパタンを生成する。

生成は次の 3 つの手続きの繰り返しによっておこなう。

- 選択 (fetching)
- 制限 (restriction)
- 解釈 (interpretation)

選択手続きは、入力された概念表現の属性にマッチする PC ペアを選択する。たとえば、入力として図 15 の概念表現が与えられると図 14 の PC ペアが選択される。図 15 の表現は最終的に “the file1 to be removed from the current directory1” のような出力を生成する。

```
品詞=不定詞句
態=受動
概念=(状態変化
      (対象 file1)
      (状態 場所)
      (初期状態 (inside-of 対象 directory1))
      (最終状態 (not (inside-of 対象 directory1))))
```

図 15 入力の例

制限手続きは、選択手続きが選択した PC ペアの変数を入力との単一化によって具体化し、パタンの一部を展開する、また、必要なら変形パタンを用いてパタンを変形する。たとえば、図 14 と図 15 を単一化すると、変数 X は file1 に、変数 Y は directory1 にそれぞれ具体化される。

解釈手続きでは PC ペア中の制約の検査をおこなう。パタンのすべての構成素が語のレベルになっていればそれを出力し、パタンがまだ展開可能な場合は再び選択-制限-解釈のサイクルを繰り返す。制約が満たされていない場合はバックトラックを起こす。

PC ペアは句構造規則に対応するレベルのものもあれば、定型句、あるいは語彙項目の場合もある。PHRED の特徴はこのように文法と語彙をすべて同じレベルで記述している点にある。

6 その他の話題

1980 年代に入って本格化した自然言語生成は、これまで見てきたように what-to-say のプランニングと文章の結束性を中心課題として発展してきた。4章では結束性を課題とした研究を中心に解説したが、他にも注目すべき研究の流れがいくつか生まれている。本章では、これらの中から特に 3 つの研究を取り上げ解説する。

6.1 語用論的制約にもとづく文章生成

3章で述べたように従来多くの生成システムでは what-to-say と how-to-say が独立に扱われていた。Hovy はこれら 2 つを結ぶものとして談話状況の語用論的制約を取り上げ、語用論的制約を用いた生成のアーキテクチャを提案している (3.2節参照)。語用論的制約を生成に組み込む動機は、「人間は、何かを話すとき、たとえ同じ内容であっても話す相手によって、また同じ相手であっても会話の状況によって、文のスタイルを変えたり、異なる語句を用いたりする」という観察にある。Hovy はこのような現象を語用論的制約によるものと考え、この観察と同じような振舞いをする文章生成システム PAULINE を開発している [68, 70, 72, 73]。

PAULINE では次のような談話状況の語用論的特徴を設定している。

談話の雰囲気

話す時間が十分にあるか、騒がしい場所かどうかなど場面の設定

話し手、聞き手の特徴

話題に対する談話参加者の知識が専門的かどうか、談話参加者の興味は大きいのか、感情の状態はどうかなどの設定

話し手と聞き手の関係

両者が面識のない間柄であるかどうか、社会的立場の関係、話し手が聞き手を好きかどうかなどの設定

聞き手に対するゴール

談話によって聞き手の知識や意見、感情をどのように変えたいのかというゴールの設定

話し手と聞き手の関係に対するゴール

両者の関係をどのように変化させたいかというゴールの設定

話し手、聞き手が各話題に対して持つ共感度

各話題に対して、談話参加者が持っている共感度

これらの語用論的特徴は、あらかじめ決められたいくつかの値の中から選択され、PAULINE の入力となる (図 3 の語用論的情報)。語用論的情報の導入は、既知/未知や焦点の扱いに限られていた従来の聞き手のモ

デル(4.2節参照)からその他のさまざまな特徴を加味した談話参加者のモデルへの拡張を意味する。ただし、PAULINE で用いる談話参加者のモデルは静的なものであることに注意したい。

PAULINE は語用論的情報に依存して異なる文章を生成することができる。上に示した語用論的情報はプランナがおこなう種々の決定の基準としては一般的すぎるため、PAULINE ではプランナが直接参照できる語用論的制約として修辭的ゴールを設けている(図3参照)。修辭的ゴールは、語用論的情報から活性化規則によって作られるゆるい言語的な制約の集合である。これは、話題収集の戦略、全体の文章の長さ、スタイル、詳細を述べるかどうかなどを指定する。「ゆるい」制約というのは、必ず満足されなければならない制約ではなく、他の制約と競合が起こったとき競合解消によって無視される場合もあるということである。

修辭的ゴールがどのように用いられているかを例で見てみよう [68, 72]。PAULINE には、MOP [136] と概念依存表現 [133] で表現した話題の集合、および語用論的情報を入力として与えるが、その他に、どの話題を中心に述べるかを示す中心的话题の列も与える。PAULINE は、中心的话题を並べて文章にする際、中心的话题に関連する補足的話題をおりませながら文章化していく。この時、どの話題を補足的話題として選択するかを決定する基準として修辭的ゴールと話題収集戦略を用いる。話題収集戦略は3種類あり、修辭的ゴールにもとづいて選択される。たとえば、「話し手の意見を聞き手に納得させるような話題を収集する」という戦略は、話題中の登場人物が持つゴールに対する話し手と聞き手の共感性が対立している場合に選択される。

- (a) 4月の初めに多数の学生が小屋を Beinecke Plaza に建設した。
- (b) 企業に南アフリカでの商業活動を放棄させるよう Yale 大学に対して要求するため、4月の初めに多数の学生が—Winnie Mandela City と名づけた—小屋を Beinecke Plaza に建設した。

(a) の内容が中心的话题である。いま、話し手は「企業に南アフリカでの商業活動を放棄させるよう Yale 大学に対して要求する」という小屋建設の目的に対して共感しており、聞き手は「大学の秩序を維持する」という目的に対し共感しているとしよう。この場合、当然、(a) の話題を述べるだけでは聞き手に学生の行動を納得させることができない。そこで、PAULINE は、話題収集戦略にしたがい、学生の行動を支持するための補足的話題として小屋建設の目的を選択する。(b) は中心的话题に補足的話題を付け加えた文章である。ただし、修辭的ゴールが短い文章の生成を指示している場合には、補足的話題の収集はおこなわず、(a) を最終的な出力とする。

修辭的ゴールは話題の収集以外にもさまざまな処理の決定基準になっている。たとえば、PAULINE は、同じ話題を表層化する場合にも修辭的ゴールに依存してさまざまな表現を生成することができる。上の例で話し手が大学側に共感しているという語用論的情報を与えると、PAULINE は (c) の文章を生成する。

- (c) 4月の初め、少数の学生が Yale 大学との衝突に巻き込まれた。衝突の原因は、南アフリカで商業活動をしている企業に大学が投資したことにあった。学生は、要求を押し通すため、Beinecke Plaza を接收し、Winnie Mandela City と名づけた小屋を建設した。

(c) では、「衝突」や「接收」などの表現を用いており、話し手が学生の行動に批判的であることをうかがわせている。

このように、PAULINE では語用論的情報を考慮したプランニングを実現しているが、PAULINE で用いている語用論的特徴が文章生成にとって本質的なものであるという心理学的確証は得られていない。Hovy は、語用論的特徴を洗練し、説得力のあるものにするを今後の課題にあげている。

6.2 推敲機能の導入

自然言語生成の研究には書く行為を意識した研究と話す行為を意識した研究があるが、書く行為と話す行為にはいくつかの異なる特徴がある。発話による対話の場合、話し手は聞き手の態度に応じて話題を変更

したり追加したりすることができる。また、声の調子、表情、みぶりなどによって話し手の意図が伝わりやすくなる場合も多い。一方、書き手が文字によって意図や情報を読み手に一方的に伝える場合、上に述べたような対話の特徴が失われるため、より質の高い文章が要求される。

それでは人間はどのようにして質の高い文章を書いているのだろうか。質の高い文章を書くためには多くの複雑な要求や制約の考慮が必要だが、すべての要求や制約を一度に考えるのは難しい。そこで多くの場合、一度書いた文章を推敲する。推敲の過程を設けることによって、複雑な要求や制約を少しずつ考慮することができるからである。文章を書く作業の大部分は推敲 (revision)⁵に当てられているという心理学的観察も報告されている [152]。このような心理学的研究を背景に、推敲過程を採り入れた生成のモデルがいくつか提案されている [3, 16, 59, 97, 99, 144, 150]。

Mann [99] らの KDS では、生成過程の途中に推敲モジュールが組み込まれている。KDS は、まず話題収集と話題構成を決定し、単文に相当する中間表現の集合を生成する。次に、得られた中間表現に対し結合規則と評価規則を適用することによって推敲をおこなう。結合規則には、たとえば次のようなものがある。

- *Whenever C then X + Whenever C then Y → Whenever C then X and Y*
- *Whenever X then Y + Whenever Y then Z → Whenever X then Z*

評価規則には、

- 文と文を結合すると加点
- 文を時間的順序に並べると加点
- *if-then-else* など特定の統語構造を使うと加点
- 同じ結合規則を繰り返し使うと減点

などがある。システムは、文章の評価値がもっとも大きく増加する結合規則を用いて新しい中間表現を得る。この操作を繰り返すと、評価値が (局所的) 最大の中間表現を得ることができる。最後に、表出モジュールが中間表現を表層文章に変換し生成過程を完了する。ただし、KDS では評価規則が文脈を考慮していないため、*how-to-say* レベルの結束性 (4章参照) を扱えない。また、最終的な表層文を評価・推敲するわけではないので、句や節の長さ、照応表現の適切さなども考慮することができない。

Gabriel の *Yh* [59] も推敲をおこなうモジュールを持っているが、*Yh* では推敲は 1 度しかおこなわない。また、このシステムも語彙選択をおこなう前に統語構造を決定し、その統語構造に対して推敲処理を適用するため、KDS と同様、表層レベルの問題の考慮が難しい。

これらのシステムに対し、Mann の Penman [97] では、*what-to-say* 決定部、*how-to-say* 決定部、評価部が互いに相互作用する独立したモジュールとして導入されている。図 16 に Penman のアーキテクチャを示す。

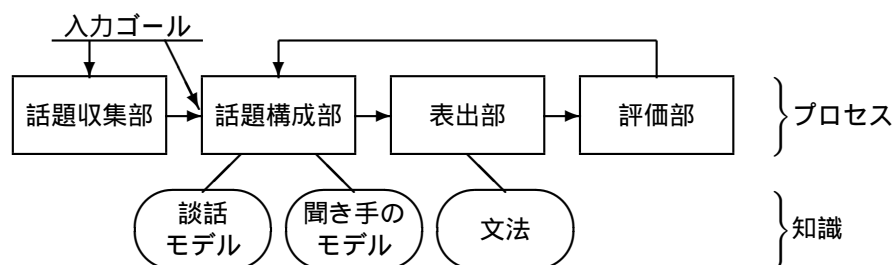


図 16 Penman におけるデータの流れ

⁵revision は一般に「校正」と訳されるが、revision をテーマとする研究のほとんどは文章を練り直すという意味で revision という語を用いているため、本稿では「推敲」に訳を統一する。

Penman は、適用領域に柔軟に適應できるシステムを目指して開発されたもので、Nigel [96, 97] と呼ばれるシステム文法を備えた汎用生成システムである。図中の矢印はデータの流れを示している。システムは、与えられたゴールにもとづいて話題収集、話題構成、表出処理をおこない、生成した文章を評価部に評価させる。評価部は、受けとった文章の評価値を求め、不満足な点を検出する。評価部の出した評価にもとづいて、再び話題の構成から修正していくという枠組になっている。また、Vaughan [144] らは、生成された文章を実際に統語解析や意味解析して評価するモデルを提案している。

推敲過程の導入については、心理学的背景の他に以下のような利点が指摘されている。

第 1 に、推敲過程の導入によって文章生成という複雑な処理をいくつかの小さな処理に分けることができる。質の高い文章の生成には多くの複雑な要求や制約の考慮が必要である。推敲過程では、一度生成された文章が満たしていない要求や制約を文章の修正によって少しずつ満足させていくことができる。したがって、一度に考慮するのは部分的な要求や制約でよい [152]。

第 2 に、文章が満たすべき制約の中には、表層化した後の方がその制約が満たされているかどうかを検出しやすいものがある。たとえば、係り受けのあいまい性を持つ文は避けなければならないが、ある文型を選択したとき係り受けのあいまい性が生じるかどうかは表層化した後でないとわからない。一度表層化した文章を推敲すれば、この種の制約について満たされていないものを検出し、効率的に修正できる [144]。

第 3 に、生成過程ではさまざまなレベルの決定が必要であるが、推敲過程の中でそれらの間の相互依存関係を実現できることが指摘されている。たとえば、乾らは日本語の語順の決定と照応表現の決定を取り上げ、それらの間の相互依存関係の実現と推敲との関係を議論している。推敲過程では、すでに語順や照応表現を含むすべての決定が暫定的におこなわれているため、語順を修正する場合には照応表現の情報を考慮することができ、逆に照応表現を修正する場合には語順の情報を使うことができる。したがって、推敲過程では個々の決定を他の決定に依存する形で修正することができるといえる [3]。

推敲機能を実際に計算機上で実現した例はほとんど報告されていない。文章の評価の難しさが主な理由として考えられるが、生成の中で推敲が果たす役割の重要性は次第に注目を集めてきている。推敲機能を持った生成システムの実現はこれからの課題である。

6.3 漸次的生成

6.2節では、情報が書き手から読み手へと一方向に流れる場合の文章生成に焦点を当て、推敲を取り上げた。本節では、発話の生成に焦点を当て、発話生成を対象とする漸次的生成 (incremental generation) について述べる。

発話における how-to-say 決定過程は、統語的な要素を集めて組み立てる過程と統語要素を音にして発音する過程の 2 つの過程に分けられる。ここでは、前者を統語木決定過程、後者を発声過程と呼ぶことにする。伝統的な発話文生成のモデルは、what-to-say 決定過程、統語木決定過程、発声過程を順におこなうものであった。what-to-say 決定過程で 1 文に対応する意味構造を生成し、統語木決定過程で 1 文に対応する統語木を生成し、発声過程で 1 文を発声するモデルである [87]。ところが、実際の発話にはこのモデルでは説明できない現象がしばしば観察される [47, 87, 141]。

人間がおこなう実際の発話では、しばしば、次の例のように (a) 付け足し、(b) 言い直しなどが起こる。

(a) John ... and Mary went to a party.

(b) John comes ... uh ... likes to come to the party.

“...” は発話されたときに時間的な空白があったことを表している。(a) では、話し手が、初め “John went to ...” と発話するつもりで “John” まで発話したところ、途中で Mary のことも思い出し、少し言葉にまつた後で “and Mary ...” と付け足している。

Kempen らは、これらの現象を説明するモデルとして、漸次的生成のモデルを提唱している。漸次的生成では、概念が発生する概念想起過程と前述の統語木決定過程、発声過程の3つの過程を想定している。漸次的生成の特徴は、図17のように各過程が同時に進行することである。概念想起過程で想起された概念は、次の概念の想起を待たず、直ちに統語木決定過程に渡され、統語要素に変換される。変換された統語要素は発声過程によって発話されるが、この間も概念想起過程と統語木決定過程の処理は継続されている。基本的に漸次的生成では想起された概念から順に次々と発話されていくが、一般には、概念想起された順に発話しても統語的に正しい文は得られない。図17では、〈JOHN〉、〈MARY〉、〈LOVE〉の順に概念が想起されているため、統語木決定過程において、動詞“loves”が発話されるまで“Mary”は待たなければならない。

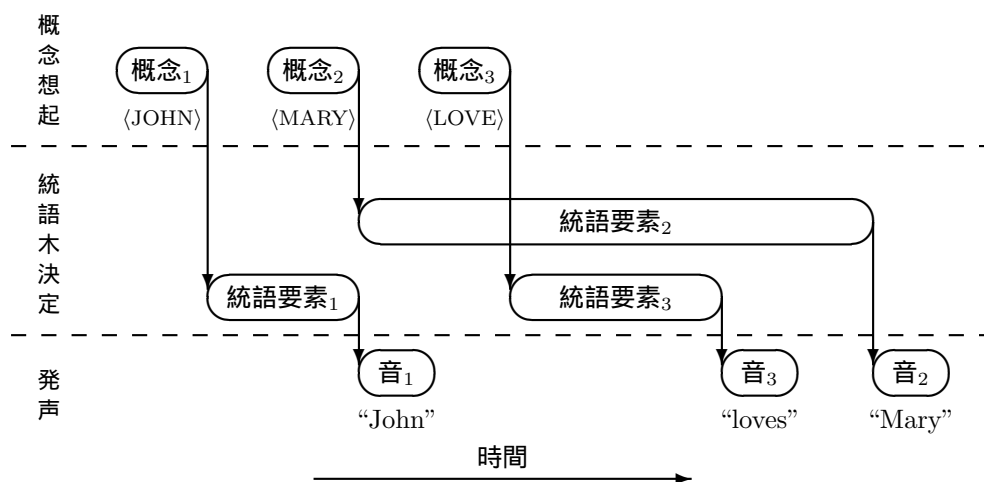


図17 漸次的生成の過程

漸次的生成は、人間が文を発話するとき、文の形をすべて決定しないうちに発話（発声）を開始し、発話の途中で次に何を言うべきか考えているという心理学的仮説に立った生成モデルである [93]。図17に示したような各過程の同時性を仮定することにより、言い直しなどの発話に現れるいくつかの現象を説明することができる。このように、Kempen らはおもに心理学的関心から漸次的生成を提唱しているが、漸次的生成の考え方を生成システムとして実現する試みもいくつかある。

漸次的生成の統語木決定過程では、想起された概念を表層語に変換し、少しずつ統語木を組み立てていかなければならない。Kempen らは、統語木を構築する操作を上方展開 (upward expansion)、下方展開 (downward expansion)、挿入 (insertion) の3種類に分け、いずれの操作も可能な文法を提案している。上方展開は図18の(a)から(b)を作る操作で、統語木が上に向かって成長する。下方展開は図の(b)から(c)を作る操作で、統語木のノードを展開する。挿入は図の(c)から(d)を作る操作で、1つのノードに別の統語構造を挿入する。

Kempen らは、これらの操作を扱える文法として、IPG (Incremental Procedural Grammar) [87]、IG (Incremental Grammar) [86]、SG (Segment Grammar) [142] を提案している。IPG は心理学的背景から離れて漸次的生成の手続き的実現を試みたものである。IG は単一化操作を組み込むことによってIPGを洗練したものといえる。また、SGはLFG [48]の形式化の一部を取り込んでいる。この他にも、FUG (Functional Unification Grammar, 5章 参照)、TAG (Tree Adjoining Grammar) [82, 83, 108]、など統語木の漸次的な成長を可能にする文法形式があるが、Kempen によると、いずれの文法形式も彼の示した3種類の操作すべてを扱えるわけではない。漸次的生成を生成システムとして実現した例に、Reithinger [57, 130]、北野 [88, 89] らの研究がある。

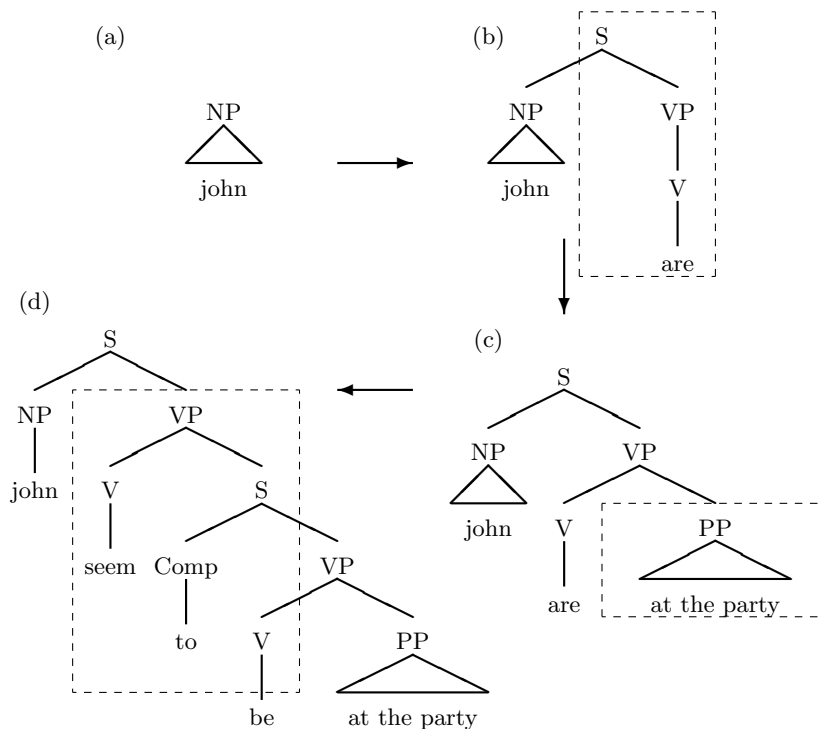


図 18 漸次的生成

7 おわりに

本稿では、1980年代の自然言語生成に関する代表的な研究について解説した。一言で言えば、1980年代は、それ以前の1文の生成から複数の文の生成に興味が多ってきたといえるだろう。また、生成の出発点も出力文章の意味構造から遡って話し手の意図へと多ってきた。これは、自然言語生成の研究が1980年代になってやっと本格化したということの意味している。計算機による自動的な文の生成自身が興味の対象であったランダムな生成やエキスパートシステムの説明文を中心とした雑型による生成の時代から見ると、自然言語生成の研究はこの10年間で大きく進歩してきた。この理由のひとつとして、自然言語理解の研究の関心が形態素や統語といった浅い解析から意味や談話といったより深い解析に移ってきたことがあげられる。自然言語生成は深い理解の結果を出発点として処理を始めなければならないからである。今後、自然言語生成の研究と自然言語理解の研究はお互いの研究成果をフィードバックしながら研究を進めていくことが重要だろう。

自然言語生成の研究が大きく発展したもうひとつの理由として、自然言語生成の応用範囲が広がってきたことが考えられる。たとえば、より人間に近い対話システム [21, 2] や要約文の生成システム [36, 29] などが具体的な応用として考えられる。より人間に近い対話システムを作るためには深い言語理解のみならず、適切な応答を生成できることが必要である。また、質の高い要約を作るには、重要な情報が含まれているだけでは不十分で、それが文章としてまとまりをなしていなければならない。この他に、自然言語生成の研究成果を活かせる応用分野として機械翻訳がある。現在のほとんどの機械翻訳システムは、1文単位で翻訳をおこなうが、将来、段落単位、あるいはもっと大きな文のまとまりで、結束性の高い、高品質な訳文を生成するシステムが登場することは十分考えられる。

自然言語生成の研究の関心にも幅がひろがり、より人間に近い生成のモデルを目指す認知科学的なアプローチをとる研究や、言語学の成果をできるだけ生かそうとするアプローチも生まれてきた。McDonaldのMUMBLEやKempenらの漸次的生成のアプローチは人間の認知過程を強く意識した研究である。Levelt

は心理言語学の立場から人間の発話過程のモデル化において重要な要因として漸次性と並列性を挙げている [93]. これらの要因を満足するための情報処理機構としてコネクショニストモデルを採用し, 自然言語を生成しようとする研究もいくつか始まっている [56, 88, 89, 145].

また, 当初は言語的知識をプログラムの中に直接埋め込む生成システムが多かった. 最近の傾向としては, 言語学の成果である文法理論を採り入れて, 言語的知識をできるだけ宣言的に記述する生成システムが増えている.

最後に, 今後に残された課題をいくつかあげて本稿の結びとしたい.

第 1 に, 生成された文章をどのように評価するかについてはこれまであまり論じられてこなかった. 2章でも述べたように, 人間の書いた文章を評価することが難しいのと同様, 計算機によって生成された文章の評価も困難である. しかしながら, 生成システムを横並びで比較するためには, 文章の評価項目, および評価基準が必要である. 個々の文の評価については, 機械翻訳システムの訳文の評価に関する研究, あるいは実験心理学の手法が参考になると思われる. また, 評価自体をテーマとした研究もいくつか出てきている [11, 20, 33]. この他に, 文章を解析し推敲に役立つ情報を書き手に提供することを目的とした文章推敲支援システムの研究において, 文章の評価法が模索されている [5, 37]. これらの研究も参考になるだろう.

第 2 に, what-to-say と how-to-say の相互作用をどのようなアーキテクチャの中で実現するかという問題がある. what-to-say のプランニングに関する研究が進むにつれ, what-to-say と how-to-say の決定過程の関係についても広く研究されるようになった. 最近では, 両者の情報交換を緊密にすべきであるという考え方が定着してきているが, その実現法については多くの問題を残している. また, 文章生成のプランニングについては, その多くは what-to-say に関するものであり, how-to-say に関する研究は比較的少ない. 今後, これまでに得られた what-to-say のプランニングに関する知見をふまえた上で, 文章の結束性を考慮した how-to-say のプランニングに関する研究も進めなければならない.

第 3 に, 適用領域に対する依存性の問題も多くの研究者によって指摘されている. より高度な生成をおこなうためには談話参加者のモデル, 背景知識, 言語知識などを洗練する必要があるが, これらはいずれも領域に依存する部分を含んでいる. したがって, 記述や参照の方法に何らかの工夫をほどこさなければ, 知識が領域に依存し過ぎてシステムが特定の領域に特化するか, あるいは, 知識が一般的過ぎて質の高い生成が難しくなる. 領域依存性の問題は知識表現と深く関わっており, 生成に用いることを意識した知識表現の研究が急がれる.

第 4 に, 生成器への入力は何かという問題についてはさらに検討する必要がある. 2章で述べたように, 最近では話し手の意図が自然言語生成の出発点であるべきだという大方の合意はあるが, 具体的に意図がどのようなものであるか, またそれをどのように表現したらよいのかについては明らかでない. 一方, 生成器への入力を仮定することすら疑問視する考え方もある. McNeill は, 心理言語学の立場から発話の自発的な生成能力を重視し, 思考と言語 (記号) はそもそも不可分なものであるとしている. 発話の入力として意図を生成器に与えるアプローチ, すなわち思考過程と切り離れた形で発話の入力を仮定するのでは真の発話とはいえないというのが McNeill の主張である [116]. McNeill は, Fodor [58] や Pylyshyn [127] 流の思考と切り離された記号操作を批判し, より人間に近い発話のモデルを作るためにはコネクショニストアプローチの方がまだ見込みがあるとしている. 計算機科学の立場からはすぐにこのような考え方を実現できるとは思えないが, 人間の発話のモデルを追求する研究者にとっては傾聴に値する意見である.

第 5 に, 入力に関連するもうひとつの問題として, 解釈 (interpretation), あるいは概念化 (conceptualization) と呼ばれる問題がある. 情報の伝達手段として自然言語生成を考える場合, システムには伝達すべき情報が入力として与えられる. この情報は格フレームや概念依存表現 [133] によって表現されることが多い. この場合, 情報内容は入力の時点ですでに特定の概念を用いて表現されており, その概念に対応する語を用いて言語化していることになる [104]. これに対し, Hovy は, 伝達すべき情報をどのような概念でとらえるかがまず重要であると指摘している [68, 70]. 6.1節で見たように, Hovy の PAULINE は談話状況の語用論的特徴に依存して同一の事象を異なる概念化を経由して言語化することができる. Hovy はこれを話題

の解釈と呼んでいる。しかしながら、入力概念の組み合わせで表現されている以上、すでに何らかの解釈を含んでいることになる。一方、概念で表現されていない入力を扱った例は少なく、Kukich の Ana (4章参照) [90, 91] はその中の 1 つである。Ana の機能は、株式の数値データを入力として株価の変動状況を報告することである。たとえば、ある数値の変動を上昇と見るか、無変化と見るかは解釈に依存する。また、このような見方 (perspective) が言語表現に与える影響を扱った例に Meteer [118, 119] の研究がある。これらの研究でも、見方の問題を含めて解釈の過程を説明する理論は明らかにされていない。解釈を生成過程に取り込むことは、今後の研究課題である。

最後に、自然言語生成の研究者が共通に使えるツールの整備も重要な課題である。自然言語理解の分野では、少なくとも形態素処理、統語処理に関しては多くのツールが利用できる状況にある。2章で述べたように生成においては出発点が不明確なので、多くの研究者が共通に使えるツールをすぐに用意することは難しいかもしれないが、可能な部分からツールの整備をおこなう努力は必要であろう。たとえば、素性の集合から形態素合成をおこなう ICOT の LTB 生成部 [26] は、形態素合成より手前の問題に関心のある研究者にとっては有益なツールである。

自然言語生成は自然言語理解に比べ若く、これからの発展が期待される研究分野である。本稿が自然言語生成の研究の発展に少しでも寄与できれば幸いである。

謝辞

草稿に対する有益なコメントを頂きました ICOT の池田光生氏、横浜国大の田村直良氏、ソニーコンピュータサイエンス研究所の長尾確氏、東工大の奥村学氏に感謝します。また、文献を提供してくださいました東大の Nigel Ward 氏に感謝します。

参考文献

- [1] 阿部純一, 金子康朗, 李光五, 邑本俊亮, 佐山公一, 伊藤俊一. 人間の言語理解における文章の文脈構築過程のモデル化. 言語情報処理の高度化研究報告 7, 言語情報処理の高度化の諸問題, 737–751, 1989.
- [2] 安西祐一郎, 神岡太郎. 自然言語理解の構造—理解メカニズム: 対話と文脈. 情報処理学会学会誌, 30(10):1150–1160, 1989.
- [3] 乾健太郎, 徳永健伸, 田中穂積. 文章生成における推敲の役割. 情報処理学会 自然言語処理研究会, NL83-7, 1991.
- [4] 垣内隆志, 榎本英治, 上原邦昭, 豊田順一. ユーザモデルを利用した説明文生成プランニング. 人工知能学会誌, 4(2):185–195, 1989.
- [5] 牛島和夫. 日本語文章推敲支援ツール『推敲』. *bit*, 23(1):4–14, 1991.
- [6] 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, NL74-3, 1989.
- [7] 久野すすむ. 談話の文法. 大修館書店, 1978.
- [8] 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, NL40-6, 1986.
- [9] 高橋晃, 桃内佳雄, 宮本衛市. 汎用文章生成システムによる日本語主題表現生成生成方略の実現. 情報処理学会 自然言語処理研究会, NL56-2, 1986.

- [10] 高橋晃, 桃内佳雄, 宮本衛市. 文章生成における接続詞の生成方略について. 情報処理学会 自然言語処理研究会, NL62-3, 1987.
- [11] 高橋善文, 牛島和夫. 計算機マニュアルのわかりやすさの定量的評価方法. 情報処理学会論文誌, 32(4):460-469, 1991.
- [12] 国立国語研究所. 談話の研究と教育 I. 日本語教育指導参考書第 11 巻, 大蔵省印刷局, 1983.
- [13] 国立国語研究所. 日本語の指示詞. 日本語教育指導参考書第 8 巻, 大蔵省印刷局, 1981.
- [14] 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, 1981. 電子技術総合研究所調査報告第 205 号.
- [15] 柴谷方良. 日本語の分析. 大修館書店, 1978.
- [16] 柴田昇吾, 藤田稔, 柵木孝一. 読みやすさを考慮した文生成. 情報処理学会 第 41 回全国大会, pp. 3:177-3:178, 1990.
- [17] 三上章. 象は鼻が長い. くろしお出版, 1960.
- [18] 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.
- [19] 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, 31(6):792-800, 1990.
- [20] 石崎俊, 井佐原均. 日本語文の複雑さの定性的・定量的特徴抽出. 情報処理学会 自然言語処理研究会, NL67-6, 1988.
- [21] 大澤一郎, 米澤明憲. オブジェクト指向方式による対話理解システム. コンピュータソフトウェア, 2(1):11-28, 1985.
- [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, NL47-11, 1985.
- [23] 辻井潤一. 機械翻訳における文章の生成. 人工知能学会誌, 4(6):645-651, 1989.
- [24] 辻井潤一. 自然言語理解の歴史と現状. 情報処理, 30(10):1142-1149, 1989.
- [25] 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.
- [26] 池田光生, 幡野浩司, 福島秀顕, 重永信一. 汎用日本語処理系 (LTB) 文生成部の生成方式. 日本ソフトウェア科学会第 5 回大会論文集, pp. 29-32, 1988.
- [27] 桃内佳雄, 高橋晃. 生成文章の結束性の良さを考慮した文章生成に関する研究. 言語情報処理の高度化研究報告 7, 言語情報処理の高度化の諸問題, 359-378, 1989.
- [28] 内田ユリ子, 石崎俊, 井佐原均. 実際的な知識に基づく文脈表現構造からの英語テキスト生成. 電子情報通信学会論文誌, J72-D-II(9):1472-1483, 1989.
- [29] 田村直良. 要約過程の形式化と実現について. 人工知能学会学会誌, 4(2):192-206, 1989.
- [30] 田中穂積, 片桐恭弘, 内田裕士, 辻井潤一, 橋田浩一. 自然言語の解析と生成 - 解析用と生成用の辞書・文法は統合可能か -. コンピュータソフトウェア, 5(4):83-96, 1988.

- [31] 田中穂積. 自然言語解析の基礎. 産業図書, 1989.
- [32] 田淵篤, 辻井潤一, 長尾真. 文脈を考慮したテキスト生成システム. 情報処理学会 自然言語処理研究会, NL65-6, 1988.
- [33] 箱守聡, 杉江昇, 大西昇. 日本語を対象とした文評価システムに関する研究. 情報処理学会 自然言語処理研究会, NL65-7, 1988.
- [34] 槇本英治, 垣内隆志, 上原邦昭, 豊田順一. 対話型システムにおける文脈情報を利用した文章生成について. 情報処理学会 自然言語処理研究会, NL59-8, 1986.
- [35] 野村浩郷, 田中穂積 (編). 機械翻訳 *bit* 別冊. 共立出版, 1988.
- [36] 北研二, 小松英二, 安原宏. 要約支援システム COGITO. 情報処理学会 自然言語処理研究会, NL58-7, 1986.
- [37] 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.
- [38] 鈴木雅実, 橋本和夫, 野垣内出, 榊博史. 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, NL53-6, 1986.
- [39] H. Abramson and V. Dahl. *Logic Grammars*. Springer-Verlag, 1989.
- [40] J. F. Allen and C. R. Perrault. Analyzing intention in utterances. *Artificial Intelligence*, 15(3):143–178, 1980.
- [41] D. E. Appelt. *Planning English Sentences*. Cambridge University Press, 1985.
- [42] D. E. Appelt. Planning natural-language referring expressions. In D. D. McDonald and L. Bolc, (eds), *Natural Language Generation Systems*, chapter 3, pp. 69–97, Springer-Verlag, 1988.
- [43] D. E. Appelt. Planning natural-language referring expressions. In *the Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 108–112, 1982.
- [44] D. E. Appelt. TELEGRAM: A grammar formalism for language planning. In *the Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 595–599, 1983.
- [45] D. E. Appelt. TELEGRAM: A grammar formalism for language planning. In *the Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 74–78, 1983.
- [46] D. G. Bobrow, R. M. Kaplan, M. Kay, D. A. Norman, H. Thompson, and T. Winograd. GUS, a frame-driven dialog system. *Artificial Intelligence*, 8(2):155–173, 1977.
- [47] K. Bock. Exploring levels of processing in sentence production. In G. Kempen, (ed), *Natural Language Generation*, chapter 22, pp. 351–363, Martinus Nijhoff, 1987.
- [48] J. Bresnan, (ed). *The Mental Representation of Grammatical Relations*. The MIT Press, 1982.
- [49] B. G. Buchanan and E. H. Shortliffe. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley, 1984.

- [50] A. Cawsey. Generating explanatory. In R. Dale, C. Mellish, and M. Zock, (eds), *Current Research in Natural Language Generation*, chapter 4, pp. 75–102, Academic Press, 1990.
- [51] P. R. Cohen and H. J. Levesque. Speech acts and rationality. In *the Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 49–59, 1985.
- [52] R. Dale. Generating recipes: an overview of Epicure. In R. Dale, C. Mellish, and M Zock, (eds), *Current Research in Natural Language Generation*, chapter 9, pp. 229–256, Academic Press, 1990.
- [53] R. Dale. The generation of subsequent referring expressions in structured discourse. In M. Zock and G. Sabah, (eds), *Advances in Natural Language Generation*, chapter vol. 2, 4, pp. 58–75, Ablex Publishing Corporation, 1988.
- [54] R. Dale, C. Mellish, and M. Zock. Introduction. In R. Dale, C. Mellish, and M Zock, (eds), *Current Research in Natural Language Generation*, chapter 1, pp. 1–15, Academic Press, 1990.
- [55] L. Danlos. Conceptual and linguistic decisions in generation. In *the Proceedings of the International Conference on Computational Linguistics*, pp. 501–504, 1984.
- [56] G. S. Dell. A spreading activation theory of retrieval in sentence production. *Psychological Review*, 93(3), 1986.
- [57] W. Finkler and G. Neumann. POPEL-HOW: A distributed parallel model for incremental natural language production with feedback. In *the Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1518–1523, 1989.
- [58] J. A. Fodor. *The Modularity of Mind: An Essay on Faculty Psychology*. MIT Press, 1983. 邦訳: 伊藤笏康, 信原幸弘訳, 精神のモジュール形式, 産業図書, 1985.
- [59] R. P. Gabriel. Deliberate writing. In D. D. McDonald and L. Bolc, (eds), *Natural Language Generation Systems*, chapter 1, pp. 1–46, Springer-Verlag, 1988.
- [60] N. Goldman. Conceptual generation. In R. Schank, (ed), *Conceptual Information Processing*, North-Holland, 1975.
- [61] B. J. Grosz and C. L. Sidner. Attention, intention, and the structure of discourse. *Computational Linguistics*, 12(3):175–204, 1986.
- [62] H. A. K. Halliday and R. Hassan. *Cohesion in English*. Longman, 1976.
- [63] M. A. K. Halliday. *An Introduction to Functional Grammar*. Edward Arnold, 1985.
- [64] K. Hasida and S. Isizaki. Dependency propagation: A unified theory of sentence comprehension and generation. In *the Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 664–670, 1987.
- [65] J. R. Hobbs. Coherence and coreference. *Cognitive Science*, 3(1):67–90, 1979.
- [66] J. R. Hobbs. *On the Coherence and Structure of Discourse*. Technical Report CSLI-85-37, Center for the Study of Language and Information, 1985.

- [67] E. H. Hovy. Generating language with a phrasal lexicon. In D. D. McDonald and L. Bolc, (eds), *Natural Language Generation Systems*, chapter 10, pp. 353–384, Springer-Verlag, 1988.
- [68] E. H. Hovy. *Generating Natural Language under Pragmatic Constraints*. Lawrence Erlbaum Associates, 1988.
- [69] E. H. Hovy. Integrating text planning and production in generation. In *the Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 848–851, 1985.
- [70] E. H. Hovy. Interpretation in generation. In *the Proceedings of the National Conference on Artificial Intelligence*, pp. 545–549, 1987.
- [71] E. H. Hovy. Planning coherent multisentential text. In *the Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 163–169, 1988.
- [72] E. H. Hovy. Pragmatics and natural language generation. *Artificial Intelligence*, 43(2):153–197, 1990.
- [73] E. H. Hovy. Some pragmatic decision criteria in generation. In G. Kempen, (ed), *Natural Language Generation*, chapter 1, pp. 3–17, Hijhoff, 1987.
- [74] E. H. Hovy. Two types of planning in language generation. In *the Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 179–186, 1988.
- [75] E. H. Hovy. Unresolved issues in paragraph planning. In R. Dale, C. Mellish, and M. Zock, (eds), *Current Research in Natural Language Generation*, chapter 2, pp. 17–45, Academic Press, 1990.
- [76] E. H. Hovy and K. F. McCoy. Focusing your RST: A step toward generating coherent multisentential text. In *the Proceedings of the Annual Conference of the Cognitive Science Society*, pp. 667–674, 1989.
- [77] E. H. Hovy, D. D. McDonald, and S. R. Young. Current issues in natural language generation: An overview of the AAAI workshop on text planning and realization. *AI MAGAZINE*, FALL:27–29, 1989.
- [78] S. Ishizaki. Generation Japanese text from conceptual representation. In D. D. McDonald and L. Bolc, (eds), *Natural Language Generation Systems*, chapter 7, pp. 256–279, Springer-Verlag, 1988.
- [79] P. S. Jacobs. KING: A knowledge-intensive natural language ganarator. In G. Kempen, (ed), *Natural Language Generation*, chapter 15, pp. 219–230, Martinus Nijhoff, 1987.
- [80] P. S. Jacobs. PHRED: A generator for natural language interfaces. *Computational Linguistics*, 11(4):219–242, 1985.
- [81] P. S. Jacobs. PHRED: A generator for natural language interfaces. In D. D. McDonald and L. Bolc, (eds), *Natural Language Generation Systems*, chapter 7, pp. 256–279, Springer-Verlag, 1988.
- [82] A. K. Joshi. Factoring recursion and dependencies: An aspect of Tree Adjoining Grammars (TAG) and a comparison of some formal properties of TAGs, GPSGs, PLGs, and LPGs. In *the Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 7–15, 1983.

- [83] A. K. Joshi. The relevance of Tree Adjoining Grammar to generation. In G. Kempen, (ed), *Natural Language Generation*, chapter 16, pp. 233–252, Martinus Nijhoff, 1987.
- [84] M. Kay. Functional Unification Grammar: A formalism for machine translation. In *the Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 75–78, 1984.
- [85] M. Kay. Parsing in Functional Unification Grammar. In D. R. Dowty, L. Karttunen, and A. Zwicky, (eds), *Natural Language Parsing*, pp. 251–278, Cambridge University Press, 1982.
- [86] G. Kempen. A framework for incremental syntactic tree formation. In *the Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 655–660, 1987.
- [87] G. Kempen and E. Hoenkamp. An Incremental Procedural Grammar for sentence generation. *Cognitive Science*, 11(2):201–258, 1987.
- [88] H. Kitano. Incremental sentence production with a parallel marker-passing algorithm. In *the Proceedings of the International Conference on Computational Linguistics*, pp. 217–222, 1990.
- [89] H. Kitano. Parallel incremental sentence production for a model of simultaneous interpretation. In R. Dale, C. Mellish, and M. Zock, (eds), *Current Research in Natural Language Generation*, chapter 12, pp. 321–351, Academic Press, 1990.
- [90] K. Kukich. The design of a knowledge-based report generator. In *the Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 145–150, 1983.
- [91] K. Kukich. Fluency in natural language reports. In D. D. McDonald and Leonard Bolc, (eds), *Natural Language Generation Systems*, chapter 8, pp. 280–311, Springer-Verlag, 1988.
- [92] J. Lancel, F. Rousselot, and N. Simonin. A grammar used for parsing and generation. In *the Proceedings of the International Conference on Computational Linguistics*, pp. 536–539, 1986.
- [93] W. J. M. Levelt. *Speaking: From Intention to Articulation*. MIT Press, 1989.
- [94] D. J. Litman and J. F. Allen. Discourse processing and commonsense plans. In *Intentions in Communication*, chapter 17, pp. 365–388, The MIT Press, 1990.
- [95] W. C. Mann. Discourse structures for text generation. In *the Proceedings of the International Conference on Computational Linguistics*, pp. 367–375, 1984.
- [96] W. C. Mann. An overview of the Nigel text generation grammar. In *the Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 79–84, 1983.
- [97] W. C. Mann. An overview of the Penman text generation system. In *the Proceedings of the National Conference on Artificial Intelligence*, pp. 261–265, 1983.
- [98] W. C. Mann, M. Bates, B. Grosz, D. McDonald, McKeown K. R., and W. Swartout. Text generation. *American Journal of Computational Linguistics*, 8(2):62–69, 1982.
- [99] W. C. Mann and J. A. Moore. Computer generation of multiparagraph English text. *American Journal of Computational Linguistics*, 7(1):17–29, 1981.

- [100] W. C. Mann and S. A. Thompson. Rhetorical Structure Theory: Description and construction of text structures. In G. Kempen, (ed), *Natural Language Generation*, chapter 7, pp. 85–96, Martinus Nijhoff, 1987.
- [101] K. F. McCoy. Conceptual effects on responses to misconceptions. In G. Kempen, (ed), *Natural Language Generation*, chapter 3, pp. 43–54, Martinus Nijhoff, 1987.
- [102] K. F. McCoy. Reasoning on a highlighted user model to respond to misconceptions. *Computational Linguistics*, 14(3):52–63, 1988.
- [103] K. F. McCoy and J. Cheng. Focus of attention: Constraining what can be said next. Presented at the 4th International Workshop on Text Generation, Los Angeles, 1988.
- [104] D. McDonald. On the place of words in the generation process. In C. L. Paris, W. R. Swartout, and W. C. Mann, (eds), *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, chapter 9, pp. 229–247, Kluwer Academic Publishers, 1991.
- [105] D. D. McDonald. Description directed control: Its implications for natural language generation. *Computers and Mathematics*, 9(1):519–537, 1983.
- [106] D. D. McDonald. Natural language generation. In S. Shapiro, (ed), *Encyclopedia of Artificial Intelligence*, pp. 642–654, Wiley-Interscience, 1987.
- [107] D. D. McDonald and J. D. Pustejovsky. Description-directed natural language generation. In *the Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 799–805, 1985.
- [108] D. D. McDonald and J. D. Pustejovsky. TAGs as a grammatical formalism for generation. In *the Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 94–103, 1985.
- [109] D. D. McDonald, M. M. Vaughan, and J. D. Pustejovsky. Factors contributing to efficiency in natural language generation. In G. Kempen, (ed), *Natural Language Generation*, chapter 12, pp. 159–182, Martinus Nijhoff, 1987.
- [110] K. R. McKeown. Discourse strategies for generating natural-language text. *Artificial Intelligence*, 27(1):1–41, 1985.
- [111] K. R. McKeown. *Text Generation*. Cambridge University Press, 1985.
- [112] K. R. McKeown. The TEXT system for natural language generation: An overview. In *the Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 113–120, 1982.
- [113] K. R. McKeown and M. Elhadad. A contrastive evaluation of functional unification grammar for surface language generation: A case study in choice of connectives. In C. L. Paris, W. R. Swartout, and W. C. Mann, (eds), *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, chapter 14, pp. 351–396, Kluwer Academic Publishers, 1991.
- [114] K. R. McKeown, M. Elhadad, Y. Fukumoto, J. Lim, C. Lombardi, J. Robin, and F. Smadja. Natural language generation in COMET. In R. Dale, C. Mellish, and M Zock, (eds), *Current Research in Natural Language Generation*, chapter 5, pp. 103–140, Academic Press, 1990.

- [115] K. R. McKeown and W. R. Swartout. Language generation and explanation. In M. Zock and G. Sabah, (eds), *Advances in Natural Language Generation*, chapter 1, pp. 1–51, Ablex Publishing Corporation, 1988.
- [116] D. McNeill. *Psycholinguistics: A New Approach*. Harper & Row, 1987. 邦訳: 鹿取廣人他訳, 心理言語学: 「ことばと心」への新しいアプローチ, サイエンス社, 1990.
- [117] C. Mellish and R. Evans. Natural language generation from plans. *Computational Linguistics*, 15(4):233–249, 1989.
- [118] M. W. Meteer. Defining a vocabulary. In *the Proceedings of the AAAI Workshop on Text Planning and Realization*, pp. 115–122, 1988.
- [119] M. W. Meteer. The implications of revisions for natural language generation. In C. L. Paris, W. R. Swartout, and W. C. Mann, (eds), *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, chapter 6, pp. 155–177, Kluwer Academic Publishers, 1991.
- [120] J. D. Moore and C. L. Paris. Planning text for advisory dialogues. In *the Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 203–211, 1989.
- [121] J. D. Moore and W. R. Swartout. A reactive approach to explanation. In *the Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1504–1510, 1989.
- [122] R. C. Moore. *Reasoning about Knowledge and Action*. Technical Report 191, SRI International Artificial Intelligence Center, 1980.
- [123] C. L. Paris. Tailoring object descriptions to a user’s level of expertise. *Computational Linguistics*, 14(3):64–78, 1988.
- [124] C. L. Paris and K. R. McKeown. Discourse strategies for descriptions of complex physical objects. In G. Kempen, (ed), *Natural Language Generation*, chapter 8, pp. 97–116, Martinus Nijhoff, 1987.
- [125] T. Patten. *Systemic text generation as problem solving*. Cambridge University Press, 1988.
- [126] F. C. N. Pereira and D. H. D. Warren. Definite Clause Grammars for language analysis – A survey of the formalism and a comparison with Augmented Transition Networks. *Artificial Intelligence*, 13(3):231–278, 1980.
- [127] Z. W. Pylyshyn. *Computation and Cognition: Toward a Foundation for Cognitive Science*. MIT Press, 1984. 邦訳: 佐伯胖, 信原幸弘訳, 認知科学の計算理論, 産業図書, 1988.
- [128] A. Radford. *Transformational Syntax: A student’s guide to Chomsky’s Extended Standard Theory*. Cambridge University Press, 1981.
- [129] E. Reiter. Generating discriptions that exploit a user’s domain knowledge. In R. Dale, C. Mellish, and M Zock, (eds), *Current Research in Natural Language Generation*, chapter 10, pp. 257–286, Academic Press, 1990.
- [130] N. Reithinger. POPEL — a parallel and incremental natural language generation system. In C. L. Paris, W. R. Swartout, and W. C. Mann, (eds), *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, chapter 7, pp. 179–199, Kluwer Academic Publishers, 1991.

- [131] G. A. Ringland and D. A. Duce. *Approaches to Knowledge Representation: An Introduction*. John Wiley & Sons Inc., 1988.
- [132] G. Russell, S. Warwick, and J. Carroll. Asymmetry in parsing and generating with Unification Grammars: Case studies from ELU. In *the Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 205–211, 1990.
- [133] R. C. Schank. *Conceptual Information Processing*. Volume 3 of *Fundamental Studies in Computer Science*, North-Holland, 1975.
- [134] R. C. Schank and C. K. Riesbeck. *Inside Computer Understanding*. Lawrence Erlbaum, 1981. 邦訳: 石崎他訳, 自然言語理解入門, 総研出版, 1986.
- [135] D. R. Scott and C. S. Souza. Getting the message across in rst-based text generation. In R. Dale, C. Mellish, and M Zock, (eds), *Current Research in Natural Language Generation*, chapter 3, pp. 47–74, Academic Press, 1990.
- [136] R. C. Shank. Language and memory. *Cognitive Science*, 4(3), 1980.
- [137] S. M. Shieber. A uniform architecture for parsing and generation. In *the Proceedings of the International Conference on Computational Linguistics*, pp. 614–619, 1988.
- [138] S. M. Shieber, G. van Noord, F. C. N. Pereira, and R. C. Moore. Semantic-head-driven generation. *Computational Linguistics*, 16(1):30–42, 1990.
- [139] C. L. Sidner. Focusing in the comprehension of definite anaphora. In M. Brady and R. C. Berwick, (eds), *Computational Models of Discourse*, pp. 267–330, MIT Press, 1983.
- [140] C. L. Sidner. *Towards a Computational Theory of Definite Anaphora Comprehension in English Discourse*. PhD thesis, MIT, 1979.
- [141] K. Smedt and G. Kempen. Incremental sentence production, self-correction and coordination. In G. Kempen, (ed), *Natural Language Generation*, chapter 23, pp. 365–376, Martinus Nijhoff, 1987.
- [142] K. D. Smedt and G. Kempen. Segment grammar: a formalism for incremental sentence generation. In C. L. Paris, W. R. Swartout, and W. C. Mann, (eds), *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, chapter 13, pp. 329–349, Kluwer Academic Publishers, 1991.
- [143] T. Strzalkowski and P. Peng. Automated inversion of logic grammars for generation. In *the Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 212–219, 1990.
- [144] M. M. Vaughan and D. D. McDonald. A model of revision in natural language generation. In *the Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 90–96, 1986.
- [145] N. Ward. *A Flexible, Parallel Model of Natural Language Generation*. PhD thesis, UCB, 1991.
- [146] J. Weizenbaum. ELIZA: A computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1), 1966.

- [147] R. Wilensky, Y. Arens, and D. Chin. Talking to UNIX in English: An overview of UC. *Communications of the ACM*, 27(6):574–593, 1984.
- [148] T. Winograd. *Language as a Cognitive Process*. Volume 1:Syntax, Addison-Wesley, 1983.
- [149] T. Winograd. *Understanding Natural Language*. Academic Press, 1972. PhD Thesis.
- [150] W. C. Wong and R. F. Simmons. A blackboard model of text production with revision. In *the Proceedings of the AAAI Workshop on Text Planning and Realization*, pp. 99–106, 1988.
- [151] W. A. Woods. Transition network grammars for natural language analysis. *Communications of the ACM*, 13(10):591–606, 1970.
- [152] M. Yazdani. Reviewing as a component of the text generation process. In G. Kempen, (ed), *Natural Language Generation*, chapter 13, pp. 183–190, Martinus Nijhoff, 1987.