自然言語処理 83－7
（1991. 5. 16）

文章生成における推敲の役割

乾健太郎, 徳永健伸, 田中穂積
東京工業大学 工学部
{inui,take,tanaka}@cs.titech.ac.jp

[概要]
文章生成には語彙選択や語順などさまざまな要素に関する選択が必要であるが, これらの選択は相互依存の関係にあるため, その実現が難しい. 本稿では, この問題へのアプローチとして, 一度表層化した文章を繰り返し改良し, 最終的に質の高い文章を生成するモデルを提案する. 生成過程全体を推敲過程としてとらえることによって, 生成に必要な種々の選択を相互に依存する形で実現できることについて議論し, モデルの概要と一部の実現について述べる.

The Role of Revision in Text Generation

INUI Kentaro, TOKUNAGA Takenobu and TANAKA Hozumi
Department of Computer Science, Tokyo Institute of Technology
(2–12–1 Ôokayama Meguro-Ku Tokyo 152 Japan)

Abstract

In the process of text generation, many kinds of choices should be decided, and these choices are dependent on each other. The versatile architecture which can handle these interdependencies of choices has been the main issue of the text generation research. We propose an architecture in which the text generation process is considered as a revising process. We show our architecture fulfills the above requirement and also describe its partial implementation.

1 はじめに

文章を生成するには, 語や語順などさまざまな要素に関する選択が必要である. これらの選択は, 文章中で述べる話題を選択・構成する what-to-say レベルと what-to-say の内容を表層化する how-to-say レベルに分けて考えることができる. 2つのレベルの選択は相互に依存するため, 両者の緊密な関係を実現するアーキテクチャの必要性が指摘されている [4]. たとえば, 1文の中にどれだけの話題を含めるかという問題は, 話題間の意味的なつながりから制約を受けると同時に (what-to-say の制約), それを表層化したときに適切な長さの文になるかという制約も受ける (how-to-say の制約). また, how-to-say レベルのみについて考えても, 種々の選択が相互に依存し, それらをどの順序で選択すればよいかが必ずしも明らかではない. たとえば, 後置詞句の語順の選択は, 各後置詞句の長さに依存するため, 語を選択しなければ適切に決めることができない. 語を選択する場合に照応表現を使うことも考えられるが, 照応表現の選択は, 先行詞と照応詞の距離などに依存するため, 適切な照応表現を選択するためには語順の情報が必要となる. このように, 生成に必要な種々の選択の間には相互依存関係がある.

これまでの文章生成の研究では, あらかじめ決められた順序に沿って逐次的に種々の選択をするものが主流であった. これらの枠組では, 選択の順序は what-to-say に関するものから how-to-say に関するものへ並んでいるのが普通である. これを図示すると図 1のようになる.

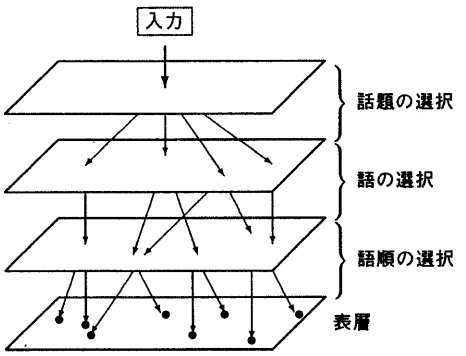

図 1　文章生成の多層モデル

図 1では, 例として, 話題の選択, 語の選択, 語順の選択を挙げているが, もちろんこれ以外にも多くの選択が考えられる. このモデルを文章生成の多層モデルと呼ぼう. このモデルでは, 最上位層で入力を受けとり, 各層の間で選択をおこなう. 図中では, 選択を矢印で表現している. 各層の点はそれまでの選択を蓄積した状態を表現しており, これは次の選択に対する制約となる. そして, 最終的に黒丸で表した表層の文章が生成される. 上層から下層へ向かう矢印のパスは, 文章を生成する際のプランニング (選択の集合) に相当し, これらのいずれかのパスを通って文章が 1 つ生成される.

このモデルは各選択のモジュールを独立に設計できるためシステムの構築が容易であるという利点があるが, 前述したように最適な選択の順序を決めることが難しいし, 選択相互の影響を動的に考慮することができないという問題がある.

多層モデルのこの問題に対する解決策として, 次に行なう選択を動的に決定する手法がある. Appelt [8] や Hovy [10] は, how-to-say レベルの選択過程で必要に応じて what-to-say レベルの選択を行なうことにより, 両者の相互依存性に対処している. また, Hovy は, how-to-say の各選択についてモジュールを用意し, モジュールの適用順序を動的に変えることによって選択の順序に柔軟性を持たせる手法を提案している [10]. これは, 図 1で言えば, ある層の決定をおこなった時点で動的に次の層を用意するようなモデルであると言える.

しかしながら, このような手法では, 一旦行なった選択については変更しないため, 将来の影響を十分に予測した上で個々の選択をおこなう必要がある. Appelt, Hovy の手法では, 統語的要因を考慮しながら what-to-say を選択するため what-to-say の選択に複雑なメカニズムを必要とし, what-to-say 選択モジュールを呼び出すタイミングの管理も困難である. 文章生成では, 論旨展開や照応表現などの文脈的な問題も考慮しなければならず, メカニズムはさらに複雑になる.

このような問題を考慮すると, 一旦表層化した文章をすぐに最終結果として出力するのではなく, その文章を評価して, 必要があれば修正するという手法が考えられる. Mann らの Penman [13], Gabriel の *Yh* [9], 柴田らのシステム [3] などはこの手法を部分的に採り入れている.

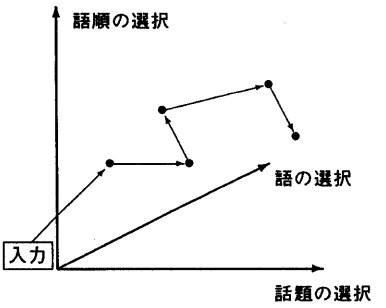

図 2　文章生成の多次元モデル

本稿では, この考え方を一歩進めて, 表層化した文章に対し評価と修正を繰り返すことにより, 最終的に質の高い文章を生成する手法を提案する. 一度書いた文章をさらに改良することを一般に推敲と呼ぶが, われわれの手法では生成過程全体を推敲過程としてとらえ

る．図1と対比させてわれわれの手法を図示すると図2のようになる．図1と同様に選択の次元として，話題の選択，語の選択，語順の選択の3つをあげたのでこの図は3次元空間であるが，一般には多次元空間となる．このモデルを文章生成の多次元モデルと呼ぶ．このモデルでは，まず文章を表層化し，そこから種々の選択を変更しながら文章の質を向上させる．空間に点在する個々の黒丸は種々の選択の結果として生成される文章を表している．最初の表層化は入力から最初の黒丸へのパスに相当する．その後，どの選択を変更するかを動的に決め，選択の変更によって修正された文章を生成する．この操作は，図2において変更する選択の次元に沿って別の黒丸へ動くことに対応する．このように，推敲は，選択の多次元空間に点在する黒丸(文章)の間を動き回る過程としてとらえることができる．多層モデルとの重要な違いは，すべての選択が暫定的なものであり，いつでも変更することができる点にある．このような手法を用いることによって生成に必要な種々の選択を相互に依存する形で実現できる．次節では，われわれの手法の概要について述べる．

2 全体の枠組

2.1 推敲にもとづく生成のアーキテクチャ

われわれが文章を書くとき，書くべき内容を一旦文字列として目に見える形にした後，それを推敲して徐々に満足のいく文章に仕上げていくことが多い．推敲の過程では，一旦作った文章(以後，草稿と呼ぶ)を読み返して不適切な部分を検出し，それを修正した後再び読み返すという作業を繰り返す．われわれの手法は，人間が行なう推敲を草稿の部分的な修正の繰り返しと見なすモデルにもとづいている．

アーキテクチャを図3に示す．生成過程は，評価，プランニング，文法適用という一連の処理の繰り返しから構成される．この一連の処理を行なう過程を推敲サイクルと呼ぶ．各処理を扱うモジュールをそれぞれ評価部，プランナ，文法適用部と呼ぶ．各モジュールの処理対象は，草稿に相当する草稿記述と呼ぶデータ構造である．生成過程では，1つの推敲サイクルが終了するたびにこの草稿記述が少しずつ変更される．草稿記述は，すべての語選択が終了した状態の統語構造とその統語構造が表現する意味を記述した意味構造からなる．意味構造は各推敲サイクルが終了した時点でのwhat-to-sayの決定結果を表し，統語構造はhow-to-sayの決定結果を表す．草稿記述では，統語構造と意味構造の対応する部分どうしがリンクによって結ばれている．この対応関係を参照することにより，各モジュールは意味のレベルまで立ち入った処理を行なうことができる．システムのデータベースには生成する文章の話題に関連する事実関係が記述してあり，what-to-sayの選択ではデータベースから必要な情報を選択して意味構造を構成する．入力は必ず文章に含めなければならない話題の集合で，意味構造の初期情報を与える．

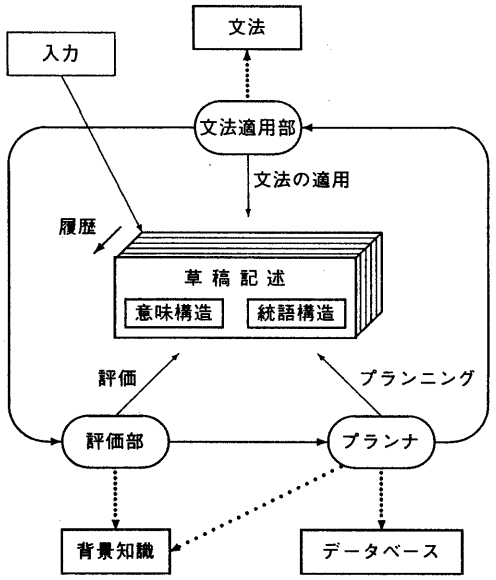

図3 推敲にもとづく生成のアーキテクチャ

評価部は草稿記述を評価し，問題点を検出する．問題点は問題箇所と理由の組としてプランナに渡される．評価項目については次節で述べる．文章の評価では「読み手に正しく情報が伝わるか」，「読み手が理解しやすいか」などが基本的な基準になるため，評価部は読み手に関する背景知識を参照しながら評価をおこなう．背景知識には，読み手があらかじめ持っていると書き手が信じている知識，文章の途中まで読み進めた時点で読み手が持つ信念の状態などがある．

プランナは，評価部から受けとった問題点の中から1つを選び，その問題点を修正する手段を決定する．修正手段は大きく分けて2種類ある．1つは，草稿記述中の情報が不足しているという問題点に対し，データベースを参照して必要な情報を意味構造に付加するという修正である．また，情報が冗長な場合には意味構造から情報を部分的に削除する．もう1つは，意味構造を修正せずに，文型，語順，語などに関する選択を変更する修正である．プランナは，過去に生成した草稿記述の履歴を管理し，推敲過程がループに陥るのを避ける役割も持つ．

文法適用部は，プランナの決定にしたがって，文法を適用しながら統語構造を部分的に書き換える．プランナが意味構造を修正した場合は，その修正箇所に対応する統語構造の一部を書き換える．また，プランナが同じ意味構造から異なる統語構造を生成するよう要求し

た場合は，たとえば別の語を用いることによって，統語構造を部分的に書き換える．

われわれの手法では，個々の修正は比較的単純であっても，さまざまな種類の修正の順序を動的に制御し，繰り返すことによって，結果的に複雑な選択を行なうのと同様の効果を得ることができる．われわれの主な関心は比較的単純な修正の繰り返しによって草稿の質を少しずつ向上させる過程自体にあるため，どこで推敲サイクルを止めるかという問題はあまり重要ではない．われわれの手法は，推敲に十分な時間をかければそれだけ質の高い文章が得られる可能性が高くなるという点で人間が行なう推敲作業と共通点があるし，満足する評価基準のしきい値を状況に応じて変えることもできる．ここでは，推敲に要する時間の上限や評価基準のしきい値を自由に設定できるパラメタとして扱う．

2.2 処理の例

例にもとづいて処理の流れを具体的に説明する．

システムにはいくつかの中心的話題の並びを入力として与える．中心的話題はイベントまたは状態を格フレームで表現したものである．システムは，この他に中心的話題に関連する事実関係を格フレームの集合としてデータベースの中に保持している．システムの仕事は中心的話題を適切な文章にすることである．このとき，必要に応じて中心的話題を補足する情報をデータベースから取り出し，読み手が理解しやすい文章に修正していく．

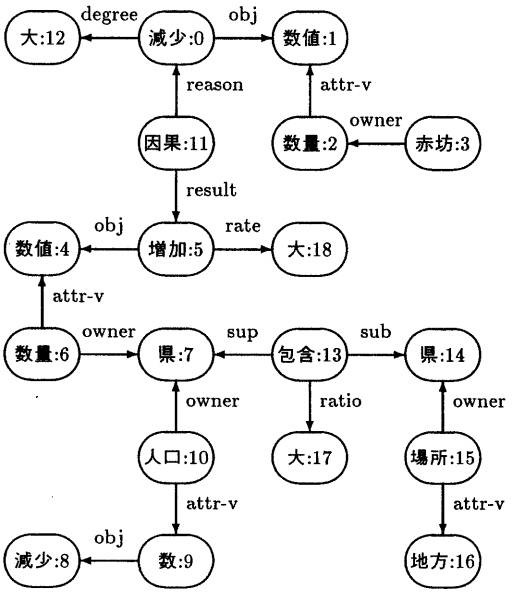

図4　データベースの内容の1部

いま，データベースの一部に図4のような事実関係が記述されており，

（ 増加:5 ，減少:0 ，包含:13 ）

という中心的話題の並びがシステムへの入力であるとする．

最初のサイクルでは推敲過程の初期化をおこなう．初期化は2つの単純な処理からなっている．まず，入力された中心的話題の並びからプランナが図5のような意味構造を生成する．この意味構造は，各中心的話題に対応するデータベース中のノードから外に向かうアークをたどってできる部分ネットワークである．したがって，初期化では意味構造が入力に対して一意に決まる．

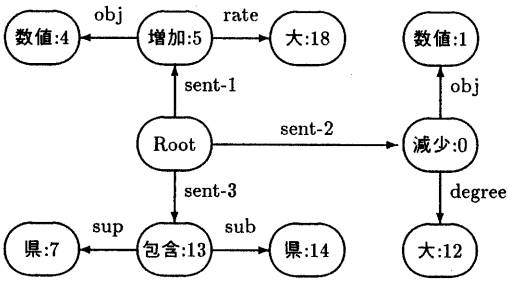

図5　初期化された草稿記述の意味構造

次に，文法適用部が意味構造から統語構造を生成する．初期化の段階では，デフォルトルールにしたがって文型，語順，語などを選択し，(a1) のような単文の並びを生成する．

(a1) 数が急速に増加した．数が極端に減少した．県の大部分が県である．

初期化が終了すると，以後評価から始まる推敲サイクルを繰り返す．まず，評価部が草稿記述を評価する．(a1) については次のような問題点が検出される．

(p1) 第1文の「数」の指示対象が不明．

(p2) 第2文の「数」の指示対象が不明．

(p3) 第3文の「県」の指示対象が不明．

(p4) 第1文の内容と第2文の内容の意味的関係が不明．

⋮

次に，プランナがこれらの問題点の中から1つを選択し，その問題点を解消する手段を決定する．問題点の選択は，問題の種類や問題箇所の相対的位置から決まる優先度にもとづいて行なう．この例では (p1) を選択し，プランナは「数」に対応するノード 数値:4 に対する新たな情報として 数量:6 と 県:7 を意味構造に付加する．最後に，文法適用部が意味構造の修正に対応する統語構造の修正を行ない，(a2) を生成する．文法適用が終ると次の推敲サイクルに移る．(a2) 以後の草稿の修正例と各サイクルで取り上げられた問題点を以下に示す．下線は問題箇所を，( ) 内は問題の種類を表す．

(a2) 県の数が急速に増加した. 数が極端に減少した. 県の大部分が県である.
(指示対象があいまい)

⋮ (途中省略)

(a3) 人口が減少した県の数が急速に増加した. 赤ちゃんの数が極端に減少した. 人口減少県の大部分が地方の県である.
(話題間の意味的関係が不明)

(a4) 赤ちゃんの数が極端に減少したために人口が減少した県の数が急激に増加した. 人口減少県の大部分が地方の県である.
(係り先があいまい)

(a5) 赤ちゃんの数が極端に減少したために, 人口が減少した県の数が急激に増加した. 人口減少県の大部分が地方の県である.
(主題化が不適切)

(a6) 赤ちゃんの数が極端に減少したために, 人口が減少した県の数が急激に増加した. 人口減少県の大部分は地方の県である.
(修飾の入れ子が深過ぎる)

(a7) 赤ちゃんの数が極端に減少したために, 人口が減少した県が急激に増加した. 人口減少県の大部分は地方の県である.

2.3 利点

われわれの手法の利点について考察する.

第1に, 推敲サイクルを繰り返し行なうため, 生成に必要な選択の順序を決める必要がない. 推敲という枠組では1度行なった選択を後から修正することができ, また, 各サイクルにおいてどの選択を修正するかは各時点の評価とプランニングによって動的に決まる. したがって, 種々の選択を相互に依存する形で実現できる. たとえば, 2.2 項の例では次のような順序で選択が修正されている.

(a1) → 話題 (情報) の収集, 照応表現の選択 → (a2)
→ 話題の収集 → (a3)
→ 接続表現の選択, 1 文の話題の収集 → (a4)
→ 主題化 → (a5)
→ 読点付加の選択 → (a6)
→ 語選択 → (a7)

(a3) から (a4) への修正では 2 つの文を統合しているが, これは 1 文にどれだけの話題を含めるかという選択の変更である. 1 節で述べたように, 1 文の話題収集は話題間の意味的なつながりの良さに依存するだけでなく, 表層化したときに適切な長さの文になるかなどの how-to-say の制約にも依存する. 文の長さは, たとえば, 参照表現にどれだけの情報の提示が必要で, どの語を用いるかなどの選択に依存する. われわれの手法では, 参照表現の選択を含む種々の選択を暫定的に行なっているため, このような相互依存関係を実現しやすい.

第 2 に, プランナが比較的単純なメカニズムで修正手段を選択することができる. 1 回のサイクルでは評価部が生成した複数の問題点のうち 1 つだけを考慮して修正するため, 一般に次のサイクルには直前のサイクルで考慮されなかった問題点の多くがそのまま持ち越される. 1 回のサイクルで行なった修正が新たな問題点を生み出すこともある. 2.2 節の例では, (a3) から (a4) への修正で話題間の意味的関係を明示するために 2 つの文を統合したが, この修正によって, 修飾句「減少したために」の係り先のあいまい性という新たな問題を引き起こしている. 修正手段の選択は新たな問題を引き起こさないように修正の副作用を考慮しながら行なうことが望ましいが, われわれの手法では推敲サイクルを繰り返すため, 新たに生じる問題を以後のサイクルに持ち越し, その中で解消していくことができる. したがって, プランナは取り上げた問題点の解消に要する局所的な修正戦略を考えるだけでよい.

第 3 に, 推敲サイクル中のどの操作も草稿記述という同一の記述形式を対象とするため, 種々の選択を行なうための知識を統一した形式で記述できる.

3 評価項目と修正手段

2 節で述べた評価部, プランナ, 文法適用部という 3 つのモジュールのうち, 文法適用部のインプリメントはほぼ終っている. 現在, プランナを実現するために, プランニングのためのルール作りを進めており, 評価部についてはまだ人間が代行している段階である. 本節では, 現在考慮している評価項目を挙げ, プランニングの例として, 係り受けのあいまい性を解消するためのプランニング・ルールについて検討する.

3.1 評価項目

文章の評価についてはこれまでに言語学や日本語学の分野でいくつかの研究がある [1, 6]. また, 推敲支援の研究では, 計算機で扱える可能性のある具体的な評価項目が提案されている [2, 7]. 推敲支援の研究における文章の評価は, 多くの場合, 内容の正確さと内容のわかりやすさという指標にもとづいて行なわれる. 計算機で文章を評価することを考えて, これまでに提案されている評価項目の例を処理のレベルごとに列挙すると次のようになる.

統語レベル タイプミス, 文法的誤り, 文や名詞句の複雑さ, 文体 (敬語表現) のゆれなど

意味レベル 用語のゆれ, 参照や係り受けのあいまい性, 表現の冗長性など

談話レベル 話題構成の良さ, 内容の正しさ, 情報の過不足など

これらの評価項目は, 統語レベルから談話レベルに移るにしたがって計算機での扱いが難しくなり, 談話レベルの評価を計算機で行なう研究はまだ報告されていない.

自然言語生成における推敲の研究では, 柴田らが文の読みやすさに関する評価項目として, 文字数, 漢字比率, 句の数, 用言の数, 修飾の入れ子の深さを取り上げている [3]. これらはいずれも統語レベルの評価項目であり, 十分とは言えない.

われわれは, 現在, 次の評価項目に絞って研究を進めている.

- 係り受けのあいまい性
- 指示・照応のあいまい性
- 主題の結束性
- 文・名詞句の長さ
- 修飾の入れ子の深さ

3.2 係り受けのあいまい性の修正

われわれの手法では, 文章の評価によって検出された問題点を動機として修正を行なうため, 評価の質が文章の質を大きく左右する. 誤った評価のもとでは質の高い文章の生成は実現できない. このため, 評価は可能な限りの厳密さが要求される. これに対し, プランニングは2節で述べたように局所的な修正手段を求めるだけでよい. ただし, 1つの問題点に対して選べる修正手段の種類が少な過ぎると, 図2に示した多次元空間を自由に動き回ることができない. プランナは, 問題点の種類と適当な適用条件に対して1つ以上の修正手段を割り当てるヒューリスティックなルールの集合として実現することができる. ここでは, 例として, 係り受けのあいまい性を解消するためのプランニング・ルールについて述べる.

係り受けのあいまい性には次のような3通りのケースがある.

(1) 1つの文節が複数の用言に係り得る場合

太郎は昨日デパートで ($_1$買った) ばかりの財布を ($_2$なくしてしまった).

(2) 1つの文節が複数の体言に係り得る場合

太郎はきれいな($_1$ワンピース) を着た ($_2$女性) と食事をした.

(3) 1つの文節が用言と体言の両方に係り得る場合

黒い髪の($_1$美しい)($_2$少女)

このうち (1) の場合についていくつかの修正手段を示す. ここでは, 係り先の候補になっている用言の数を2つに限定し, 文頭に近い方の用言を (用言1), もう一方を (用言2) と呼ぶ.

(1-1) (係り句) が (用言1) に係る場合.

(r1) (係り句) と (用言1) の間に (用言1) に係る別の (係り句′) がある場合, (係り句) を (係り句′) と (用言1) の間に置く.

太郎はデパートで昨日買ったばかりの財布をなくしてしまった.

(r2) (用言1) と (用言2) の間に (係り句) と同じ格を持つ (係り句′) を挿入する.

太郎は昨日デパートで買ったばかりの財布を今朝なくしてしまった.

(r3) (用言1) を別の文として独立させる.

太郎は財布をなくしてしまった. その財布は昨日デパートで買ったばかりのものだ.

(r4) (係り句) の直前の (係り句′) が (用言1) に係らない場合, (係り句′) と (係り句) の間に読点を打つ.

太郎はさびしそうに去っていく花子を見つめていた.
→ 太郎は, さびしそうに去っていく花子を見つめていた.

(1-2) (係り句) が (用言2) に係る場合.

(r5) (係り句) を (用言1) と (用言2) の間に置く.

太郎はデパートで買ったばかりの財布を昨日なくしてしまった.

(r6) (係り句) の直後に読点を打つ.

太郎は昨日, デパートで買ったばかりの財布をなくしてしまった.

(r7) (用言1) を別の文として独立させる.

(r8) (用言1) が名詞に係る場合, 可能ならその名詞といっしょにして1つの名詞を作る.

赤ちゃんの数が減少したために人口が減少した県の数が増加した.
→ 赤ちゃんの数が減少したために人口減少県の数が増加した.

これらルールは, 今後実験によって有効性を確かめなければならない. また, ルールの適用によっていたずらに問題点の数が増えないように, ルールの適用条件の精密化も必要である. たとえば, ルール (r3) とルール (r7) は, (用言 1) が連体修飾句を構成するか, 連用修飾句を構成するかでさらに場合分けが必要だが, (用言 1) が連体修飾句を構成する場合, 次のような問題を考慮する必要がある.

連体修飾句は機能の点から 2 種類に分類できる. 1 つは (b1) や (b2) のように係り先の名詞の指示対象を限定する制限修飾句で, もう 1 つは (b3) のように係り先の名詞の指示対象を補足的に説明する非制限修飾句である. さらに, 制限修飾句は, 指示対象が (b1) のように特定のインスタンスであるか, (b2) のように不定のインスタンスであるかによって性質が異なる. これらのうち, 不定インスタンスを指示する名詞の制限修飾句 (b2) に対しては, ルール (r3), (r7) を適用できない. また, 定インスタンスの制限修飾句 (b1) についても適用不可能な場合があり, 必ず適用できるのは非制限修飾句 (b3) の場合に限られる.

(b1) 太郎は昨日花子にもらった本を 1 日で読み上げた.

(b2) 太郎はいつも山が見える家に住みたいと思っている.

(b3) 太郎はさびしそうに去っていく花子を見つめていた.

名詞が定インスタンスを指示する場合, 連体修飾句が制限修飾句であるか非制限修飾句であるかは名詞の性質や文脈に依存するため, データベースにあらかじめその情報を記述しておくことはできない. ルール (r3), (r7) の適用条件をもっとも厳しくすると, たとえば「連体修飾句を外しても残りの名詞句が指示対象を十分に同定できる」となる.

4 実現

現在, 上述したモデルのインプリメントを進めているが, 評価部とプランナについては多くの問題を残しており, 今後の研究課題である. 本節では, 統語構造に対して種々の修正を可能にする文法とその適用について述べる.

システムの文法には phrasal lexicon を用いている. これによって, 慣用的表現や自然な言い回しの扱いが容易になる [12]. また, phrasal lexicon は文法と辞書を融合しているため, 文や句の構造の選択と語選択を同じレベルで扱うことができ, 統語構造に種々のレベルの修正をほどこす我々の手法には有効である. phrasal lexicon にはいくつかの実現法があるが, ここでは, Jacobs の生成システム Phred[11] と同様, PC ペア (Pattern-Concept-pair) の集合として記述する方法をとっている. Phred の PC ペアは部分的な意味ネットワークと部分的な統語構造との対応を表す. Phred は, PC ペアを参照することによって意味ネットワークから部分的な統語構造を作り, それを組み合わせることによって文を生成する.

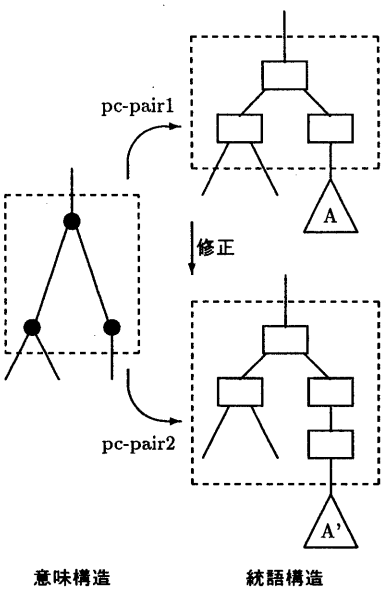

図 6 統語構造の修正

われわれは以下の 3 点において Phred の PC ペアを拡張した. 第 1 点は, Phred が 1 文の生成しか扱っていないのに対し, 文章の生成を扱えるようにした点である. われわれのシステムでは, 意味ネットワークと複数の文とのマッピングを phrasal lexicon で行なう. 第 2 点は, Phred の PC ペアが意味ネットワークから 1 次元の構成素の並びへのマッピングを表すのに対し, われわれの PC ペアが統語木へのマッピングを表す点である. Phred は 1 節で述べた多層モデルにもとづく生成システムであるため, 評価に必要な情報を出力する必要はなく, 意味表現から 1 次元の文字列を生成することを目的としている. これに対し, われわれのシステムは統語木を生成する. 統語木は, 語の意味や語順などの情報を含む依存構造として表現し, 評価に関するさまざまな情報を与える. 第 3 点は, 直前の推敲サイクルで使用した PC ペアの情報を保持する点である. われわれのシステムでは, 図 6に示すように, 直前の推敲サイクルで用いた PC ペアを別の PC ペアで置き換えることによって統語構造の修正を実現している. このとき, たとえば, 部分木 A が新しい PC ペアの部分木にそのまま接続できるかどうかを調べる必要がある. 部分木 A は, 過去の推敲サイクルで修正を受けている可能性がある

ため, できるだけ変更しないことが望ましい. 文法適用部は, 草稿記述の履歴と PC ペアの情報を参照し, 統語構造の修正範囲を最小限に抑える機能を持つ.

5 おわりに

本稿では, 生成過程において推敲の役割を重要視することの有効性について述べた. 1 度表層化した文章を繰り返し評価, 修正することによって, 種々の選択間の相互依存を実現することができ, 修正のプランニングも比較的単純なルールを用いて実現できる.

3 節でも触れたように, われわれの手法では評価の厳密性が非常に重要になる. Vaughan は, 推敲における評価を読み手の仕事として書き手から独立させなければならないと指摘し, 草稿を実際に解析して評価するモデルを提案している [15]. しかしながら, 現在の解析技術では, あいまい性の評価に限ってみても十分な精度で問題点を検出することは難しい [5]. 文章の機械的な評価方法は今後の主要な研究課題である.

現在のところ, システムへの入力は中心的話題の並びを仮定しているが, 本来なら, その中心的話題を文章にする動機となる書き手の意図があるはずである. 中心的話題を補足する話題をデータベースから取り出す際にも意図の果たす役割は重要である. 意図からの生成は最近盛んに研究されおり, その 1 つに RST を用いる Moore らの手法がある [14]. われわれは, Moore らが提案する意図からのトップダウンなプランニングとわれわれが提案する表層からのボトムアップな推敲とを融合する手法を検討している.

参考文献

[1] 岩淵悦太郎 (編). 悪文. 日本評論社, 1984.

[2] 牛島和夫. 日本語文章推敲支援ツール『推敲』. *bit*, 23(1):4–14, 1991.

[3] 柴田昇吾, 藤田稔, 栅木孝一. 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.

[4] 徳永健伸, 乾健太郎. 1980 年代の自然言語生成 − 1 −. 人工知能学会学会誌, 6(3):380–387, 1991.

[5] 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.

[6] 木下是雄. 理科系の作文技術. 中央公論社, 1981.

[7] 林良彦, 高木伸一郎. 日本文推敲支援機能の実現方式. 人工知能学会全国大会, pp. 399–402, 1989.

[8] 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.

[9] 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.

[10] E. H. Hovy. *Generating Natural Language under Pragmatic Constraints.* Lawrence Erlbaum Associates, 1988.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.