

文脈に応じたいろいろなバリエーションの文章の生成

李航 長尾真
京都大学工学部

本研究では、文脈に応じたいろいろなバリエーションの文章を生成するメカニズムを提案する。話題等文章の生成過程で参照される大域的な情報（記号データ）と話者の心理状態を表わす情報（数値データ）によって、生成を行う文脈を定義する。具体的には、文章の話題などを決め、知識ベースから生成する意味内容を抽出し、更に、その意味内容についている想起のしやすさを表わす重みを与えることによって生成の文脈を実現する。設定された文脈のもとで、ヒューリスティックスによって動的に生成のためのルールの適用順序を決め、意味表現にルールを適用する。その結果、いろいろなバリエーションの文章が出る。このメカニズムをインプリメントし、いろいろなバリエーションの文章の生成に成功した。

GENERATING VARIATION IN TEXT BASED ON DISCOURSE MODELS

Hang Li Makoto Nagao
Department of Electrical Engineering, Kyoto University
Sakyo-ku, Kyoto 606, Japan

A natural language processing mechanism for generating variation in text based on discourse models will be introduced in this paper. Discourse is defined by symbolic and analogue data. The former is frequently referred to during generation and expressed by global variables. The latter represents the writer's mental state and is expressed by the weights of frames and rules. When generating text, a discourse model is determined dynamically with the applied sequence of rules for generation changing correspondingly. As a result, variation of text is acquired. The mechanism has been implemented in a system named KUNGFU, and many variations of text have been generated.

1 はじめに

知識ベースの問い合わせシステム等の自然言語インタフェースにおいては、内部表現（意味表現）で表された意味内容を自然言語の形で出力する文章生成の機能が必要である。自然な文章を生成するためには、この文章生成のメカニズムを研究することが不可欠である。人間は、同じ意味内容を、いろいろなバリエーションの文章で表現することができる。本研究では、人間のもつこのような能力をどのようにすれば計算機上で実現できるかという問題に焦点を当てる。

文章生成の研究の中では、McKeownの研究[2]が有名である。McKeownは、文脈構造（Discourse Structure）¹の概念を導入し、それを使って文章の生成を行なっている。McKeownの文脈構造は、文章が文と同じように構造をもっていると仮定し、記号の組合せ的なデータによって文章の構造を定義するものである。しかし、そのような文脈構造をどんなに数多く定義しても、普通人間の書く文章のバリエーションを全部表現しきれない。それを解決するため、本研究では、文章の構造を静的に定義してから生成を行なうのではなく、人間の文章生成の計算過程に近いモデルを実現することによっていろいろなバリエーションの文章を生成する。

一方、計算機による自然言語生成の研究の中で、処理の流れの制御の研究が重要な位置を占めている。生成過程における文法規則の適用順序によって、処理の効率が変化するし、生成される内容も場合によって変わる。今までシステムック文法（Systemic Grammar）[5]による生成等のメカニズムが提案されたが、もっと優れたメカニズムの研究が必要である。

本研究では、計算機による文章生成の新しいメカニズムを提案する。まず、文章のバリエーションは生成を行なう際の文脈によって決まると仮定する。文脈を記号データと数値データによって定義し、人間の文章生成処理に近いメカニズムで生成を行なう。具体的には、文章生成の処理をヒューリスティックスによって制御し、いろいろなバリエーションの文章を生成する。

本研究のねらいは、新しい文章生成メカニズムの研究を通じて、自然言語処理における文脈の定義のしかたを探り、記号処理とコネクショニズム・モデルの融合を試み、ヒューリスティックスによる生成処理の制御を実現することである。

¹本当は文章の構造と呼ぶべきである。

2 新しい文章生成メカニズムについて

2.1 文脈と文章のバリエーション

2.1.1 文脈

本研究では、文章生成時の言語的、非言語的環境のことを文脈と呼ぶ。文章生成システムでは、文脈の初期状態から、文を一つ生成し、同時に新しい文脈を作り出す。新しい文脈のもとで次の文の生成を行なう。つまり、文章生成を文脈の初期状態を順次変化させる関数と考える。文章生成の過程では、次の二種類の情報で文脈を定義する。

- 話者の話題、焦点、相互知識など、文脈の中で動的に参照される情報。以下では、文脈情報と呼ぶ。文脈情報は、大域変数によって表現する。
- 話者の心理状態を表わす情報。以下では、状況情報と呼ぶ。状況情報は意味表現と生成ルール²に付く重みによって表現する。

文脈情報を D 、状況情報を T 、生成内容に關与する知識ベースを E とすると、文脈を次のように定義することができる。

$$S = D + T \quad (1)$$

$$S_i = U_i(S_{i-1}, E) \quad i = 0, 1, \dots, n \quad (2)$$

ただし、 U はある文脈から新しい文脈を作り出すための関数である。

2.1.2 文脈に応じた文章のバリエーション

自然言語文章のバリエーションとは、同じ意味内容を違う文章で表現することである。本研究は、文章を形成する文と文の間、或いは文中の文字列の順序の違いから生じるバリエーションだけを問題にする。

文章のバリエーションに影響する要因としては、以下のものが考えられる。

意味内容間の関係 例えば、因果関係にある意味内容を言語表現にする時に、因果関係を表す言語表現で表現しなければならない。

文章生成の過程での意味内容の想起順序

生成ルールによって定義される制約 例えば、時間順に事象を説明する制約によって生成を行なうと、時間順に事象を説明する文章が得られる。

生成ルールの適用 例えば、代名詞化する生成ルールが適用されるか否かによって、代名詞化が行われたり、行われなかったりする。

²生成ルールは、普通の書換えルールではなく、文章生成のためのルールである。

文脈モデルの違い 例えば、話題、焦点などの設定値の違いによって、文章にバリエーションが出る。

以上の考察によると、静的な記述としての意味表現と生成ルール以外に、生成の過程で動的に変わる文脈の要素が文章のバリエーションに影響することがわかる。しかも、その文脈要素には二種類のものがある。一つは、話題、焦点のような文章中の言語表現間の関係を表わすものであり、もう一つは、意味表現の想起しやすさ、及び適用する生成ルールの想起しやすさを表わすものである。この二種類の文脈要素が前に定義した文脈情報と状況情報に他ならない。

2.2 処理の立場から

文章の生成は、人工知能の領域の問題である。文章生成では、意味内容を表す意味表現も、文章生成のためのルールも制約[3]と見なすことができる。処理の立場から考えると、理想的な文章生成システムの持つべき特徴は以下のようなものになる。

- 意味内容そのものを組合せ的なデータで表現する。本研究では、それを意味表現と呼び、具体的にはフレームで表す。生成の処理を意味表現の変換としてとらえる。生成の過程では、意味表現と文脈情報を特定し、意味表現に生成ルールを次々と適用する。同時に文脈情報を書き換えていく。
- 生成ルールは記述の部分性をもつ。生成ルールは文章全体の構造を規定するのではなく、部分的な構造(例えば、文と文の関係)を規定する。
- 生成ルールは宣言的に定義され、処理の流れと独立である。処理はヒューリスティックスによって制御される。つまり、意味表現にも、生成ルールにも重みが付いており、生成の過程では、重みの計算によって生成ルールの適用順序を決める。

2.3 新しい文章生成のメカニズム

2.3.1 入力

文章生成システムへの入力は、意味表現と文脈情報と状況情報である。具体的には、大域変数と重みのついているフレームが入力される。大域変数は文脈情報を表す。フレームそのものは意味内容を表し、各フレームのスロットについている重みは状況情報を表す。

意味表現、文脈情報、状況情報の設定によって、ある特定の文脈を設定することになる。設定された文脈のもとで、システムが自動的に文章の生成を行なう。

重み付きフレームは、人間の知識ベースに相当する各ノードとリンクに重みの付いている意味ネットワークから与えられる。フレームの各スロットに入る値が、意味ネットワークのノードに入る値で、各スロットにつく重みが、同じネットワークの活性化関数によって計算される。ある文脈のもとで、ネットワークの一部のノードに重みの変化(状況情報)が与えられる。ネットワークは活性化関数に基づいて活性化し、収束後、意味内容間の重みで表わすその文脈に於ける「想起関係」が定まる。スロットにつく重みは、意味内容間の「想起関係」を表わす。

2.3.2 生成ルール

人間の書く文章は、その全体の「流れ」に統語的、語法的な矛盾がなく、いわば結束性(Cohesion)をもっている。本文章生成メカニズムでは、生成する文章に結束性を与えることを中心に行なわれる。文章の結束性を生成ルールによって定義し、ルールの適用によって実現する。しかも、生成ルールは、段落を越えない結束性を規定することだけをを対象にする。段落を越える文章の生成には、各段落の構成を「プランニング」してから生成を行なう必要があると考えられる。

意味内容を表わす意味表現がニュートラルな表現で、それらを結束性のある文章にするためには、文脈情報を取入れないといけない。例えば、意味表現を表層化する前に、文の話題を意味表現に設定することによって、文の表層化の時の態(Voice)を決める情報を与えなければならない。

意味表現を言語表現になおすためには、生成ルールと呼ばれるルール³を適用する。生成ルールは、表層化する生成ルールを除いて、ニュートラルな意味表現を現在の文脈情報も含めた中間表現に変換するルールである。表層化する生成ルールは、文章生成の最後に中間表現を言語表現に変換するルールである。表層化する生成ルールの適用は、中間表現をDCGルールに従ってトップ・ダウン的に文にすることである。

生成システムでは、意味表現(フレーム)を一つずつ文にする処理が行なわれる。各々の意味表現からの生成では、意味表現(フレーム)に生成ルールの適用によってその表層化の時の木構造を少しずつ決め、最後に、表層化する生成ルールの適用によってそれを一次元の文字列にする。

2.3.3 処理の流れ

本文章生成メカニズムでは、文を基本単位とし、ボトム・アップ的にそれらを文章に組み立てる。また、各々の文をトップ・ダウン的に生成する。本研究で言っている文章のバリエーションは、実は文を違う順序で生成することによって生じる。

³以下では、ルールと言えば、生成ルールのことを指す。

生成システムでは、生成ルールとそれを解釈するインタープリタとが独立している。

生成ルールの適用は大域的な文脈情報の書換えを伴う。つまり、生成する文を少しずつ確定しながら、新しい文脈をつくることに相当する。

インタープリタが、入力フレームの中の想起しやすいフレームから順に生成を行なう。生成ルールの適用に成功と失敗があるため、出力される文章の文の順序が必ずしもフレームの想起の順と同じとは限らない。

各々のフレームに生成ルールを適用する時に、フレームの重みから生成ルールの重みを計算し、その重みで、生成ルールの適用順序を判断する。計算された各生成ルールの重みはルールの成功しやすい順序を表す。

従って、フレームの想起しやすさの情報（状況情報）、生成ルールのヒューリスティクス、生成ルールの記述する制約によって、文章のバリエーションが現われる。

2.4 検証

2.4.1 言語学の立場から

Halliday 等いわゆる機能主義的な立場を取る言語学者達は、言語をコミュニケーションの目的を満たすための機能としてとらえている。Halliday は、自然言語の文章が四つのメタ機能（Meta Function）からなると分析している[4]。その四つのメタ機能とは、Experimental Function、Logical Function、Interpersonal Function、Textual Functionである。Experimental Function は話者の経験からなる一般知識と対応する。Logical Function は言語表現間の制約（文法）と対応する。Textual Function は前後の言語表現の整合性を保つための文脈モデルと対応する。Interpersonal Function は具体的なコミュニケーションに於ける話者と聴者の間の相互作用モデル（Model of Speaker-Listener Interaction）と対応する。Halliday の分析から、文章は四種類の表現に基づいて生成できることが分かる。この四種類の表現は、本研究の提案する文章生成メカニズムの中の意味表現、生成ルール、文脈情報、状況情報に対応する。

2.4.2 コネクショニズム・モデルの立場から

本文章生成メカニズムは記号処理とコネクショニズム・モデルを結合している。そのメカニズムでは、まずコネクショニズム・モデルによって意味表現から生成ルールへの連想を行い、次にその連想された生成ルールの重みをヒューリスティクスとして使い、記号処理によって生成を行っている。

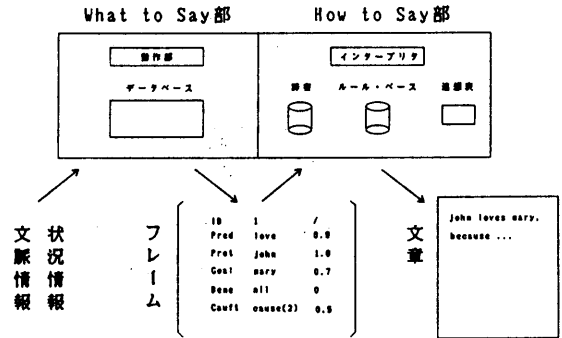


図1: KUNGFU の構成

2.4.3 システミック文法との比較

本文章生成メカニズムは、システミック文法（Systemic Grammar）[4]より、以下の点で優れている。これらは特に文章の生成では大きな利点となる。

- 言語とそれ以外の機能の相互作用モデルではなく、意味内容と生成器（Generator）の相互作用モデルで生成を行なうことによって、生成過程での計算量を減らしている。
- 制約を生成ルールによって統一的に記述し、処理の流れと分離させている。
- 生成ルールの適用順序を動的に変えることによって、処理の効率を上げている。

3 KUNGFU

3.1 KUNGFU の構成

本研究の提案する文章生成のメカニズムを KUNGFU（Kyoto UNiversity text Generation For Universal languages system）と名付けたシステムで実現した。

KUNGFU は、生成する意味内容を決める部分（What to Say 部）と意味内容を言語表現に変換する部分（How to Say 部）からなる（図1を参照）。What to Say 部は、文章生成のための意味内容を表現する知識ベース（Knowledge Base）からなる。What to Say 部における処理は、その動作部の制御によって行われる。How to Say 部は、生成ルールのルール・ベース、辞書、連想表からなる。How to Say 部における処理は、そのインタープリタによって行われる。

システム全体への入力は、What to Say 部に対して行われる。その入力は、生成する文章の話題と意味ネットワークのノードの重みの変化分である。What to Say 部の出力

は、各スロットに重みのついているフレーム⁴である。How to Say 部の入力は、What to Say 部で得られた重み付きフレームである。How to Say 部の出力は、生成される文章である。

なお、What to Say 部は、本研究の提案する文章生成メカニズムの生成内容を決める前処理の部分を実現している。How to Say 部は、文章生成メカニズムそのものを実現している。

3.2 What to Say 部

What to Say 部は、ネットワーク型の「知識ベース」である。そのネットワーク⁵は、次の特徴をもつ。

- ノードには、動詞概念のノードと名詞概念のノードの、二種類がある。
- 各ノードに入る値は、文章を生成するための深層概念を表わす。
- 各ノードには重みがついている。ノードの重みは生成の過程での想起しやすさを表わす。
- リンクは動詞概念のノードから名詞概念のノード、或は他の動詞概念のノードへはられる。
- 各リンクには重みがついている。その重みは隣接する二つのノードの関係の強さを表わす。

What to Say 部への最初の入力は、生成する文章の話題である。What to Say 部の動作部は、その話題に関するネットワークを知識ベースから引出し、ワーキング・メモリにロードする。次に What to Say 部の意味ネットワークの一部、或は全部のノードに対し、重みの変化分を与える。具体的には、ユーザのマウス・クリックによる各ノードの重みの調節（増加、或は減少）によって実現する。ユーザによる重みの調整は次の約束に従う。

- 生成する文章の話題になる概念の重みを上げる。
- 生成する文章の中で強調したい概念の重みを上げる。
- 聴者が知らないと思われる概念の重みを上げる。

したがって、この重みの入力は、システムの置かれるある特定の状況を模擬する状況情報になる。

ノードの重みの変化が意味ネットワークの不安定をもたらし、一定の時間経過後、ネットワークは再び収束状態に向かう。KUNGFU のネットワークは次の活性化関数に基づいて活性化する。

⁴以下では、重み付きフレームと呼ぶ。

⁵以下では、単に意味ネットワークと呼ぶ。

$\vec{X}(t)$ は各ノードの時間 t に於ける重みとすると、時間 $t+1$ の時、

$$\vec{X}(t+1) = \vec{A} \times \vec{X}(t) + \vec{C} \quad (3)$$

但し、 \vec{A} と \vec{C} は、それぞれ係数行列と \vec{X} の初期状態である。この活性化関数は線形的であるため、次の特徴を持つ。

- より活性化するノードの周りのノードが活性化しやすい。
- 多くのリンクを持っているノードが活性化しやすい。
- 重みの大きいリンクでつながったノード同士がお互いに活性化しやすい。

What to Say 部の動作部は、次に、収束したネットワークから文章を生成するための意味表現を抽出する。その意味表現は重み付きフレームによって表わされる。重みつきフレームの抽出は動詞概念のノードから行なう。具体的には、ノードの値が対応するフレームのスロットに入れられ、ノードの重みが対応するフレームのスロットに付けられる。

What to Say 部は、フレームの重みの平均があるしきい値以下のフレームをカットし、残りのフレームを次の How to Say 部に渡す。

以上の操作と別に、What to Say 部の動作部は、文章生成に必要な文脈情報の初期化を行なう。

3.3 How to Say 部

3.3.1 How to Say 部の構成

How to Say 部は、生成ルールのルール・ベース、辞書と連想表からなる。

生成ルールは入力する意味表現（重みつきフレーム）を中間表現に、中間表現を中間表現に、また中間表現を出力文に変換するための変換ルールである。中間表現とは意味表現に文脈情報を付け加えた表現（やはり、重みつきフレームで表現する）である。生成ルールは次のように PROLOG の述語によって統一的に記述される。

$$\text{generate_rule}(IN_1, \dots, IN_n, OUT_1, \dots, OUT_m) \quad (4)$$

IN_i 、 OUT_j は、それぞれ生成ルールの適用前と後の表現である。また、生成ルールのデフォルトの適用順序を表わすデフォルトの重みも一緒に定義される。

辞書は、意味表現をそれと対応する言語表現に変換するのに使われる。

連想表は、意味表現から生成ルールの重みを計算する時に使われる。

3.3.2 生成ルールの重み

How to Say 部での処理はすべてインタープリタによって制御される。生成の過程では、インタープリタが一定のアルゴリズムに従って、各生成ルールの重みを計算し、その重みで生成ルールの適用順序を判断する。

生成ルールの重みは、以下のように計算される。

まず、生成ルールのデフォルトの重み $DWOR(i)$ が、そのデフォルト大域重み $DWOR_{global}(i)$ とデフォルト局所重み $DWOR_{local}(i)$ によって定義される。

$$\begin{aligned} DWOR(i) &= DWOR_{global}(i) + DWOR_{local}(i) \\ 0 < DWOR_{local}(i) < 1 \\ DWOR_{global}(i) &\text{は整数} \end{aligned} \quad (5)$$

一方、生成ルールの重み $WOR(i)$ がやはりその大域重み $WOR_{global}(i)$ と局所重み $WOR_{local}(i)$ によって定義される。

$$WOR(i) = WOR_{global}(i) + WOR_{local}(i) \quad (6)$$

更に、

$$WOR_{global}(i) = DWOR_{global}(i) \quad (7)$$

$$\begin{aligned} WOR_{local}(i) &= V(DWOR_{local}(i)) \\ 0 < WOR_{local}(i) < 1 \end{aligned} \quad (8)$$

生成ルールの重みの大きさ(絶対的な大きさ)は生成ルールの適用順序を表す。その中で、選択的な(Alternative)関係⁶にある生成ルールを、それらのデフォルト大域重みを同じ大きさの自然数にすることによって、一つの「グループ」にまとめることができ、同じグループに入る生成ルールの重みは、二つの連続する自然数にはさまれる。

V は、連想表を使ってデフォルトの局所重みから局所重みを計算する関数で、次の式で定義される。

$$V(DWOR_{local}(i)) = DWOR_{local}(i)F(x(i)) \quad (9)$$

$$x(i) = \sum_{j=1}^m (t(i,j)WOF(j)) \quad (10)$$

WOF (Weight of Frame) はフレームの各スロットの重みである。 $t(i,j)$ は連想表の要素である。また、

$$F(x) = \begin{cases} 0 & x < 0 \\ x & 0 < x < 1 \\ 1 & x > 1 \end{cases} \quad (11)$$

⁶複数のルールの中で、一つのルールが適用に成功したら、他のルールの適用が必要でなくなることを指す。

i,j	Pred	Prot	Goal	Bene	Cause	Conces
set_topic1	0	1	0	0	0	0
set_topic2	0	0	1	0	0	0
set_topic3	0	0	0	1	0	0
set_cause1	0	0	0	0	1	0
set_conces1	0	0	0	0	0	1

図 2: 連想表の例

連想表は、入力の意味表現(フレーム)と生成ルールの間の「想起関係」を表わす 0 と 1 で埋められた表である(図 2 を参照)。連想表の横列は各フレームのスロットに対応し、縦列は各生成ルールに対応する。連想表では、位置 (i,j) の値が 1 の時、 j 番目のスロットの重みが大きければ大きいほど i 番目のルールが想起され易いことを意味し、位置 (i,j) の値が 0 の時、 j 番目のスロットと i 番目の生成ルールの上に想起関係がないことを意味する。

連想表を使う計算(式 9、10、11)によって、同じグループ内の生成ルールの適用順序は入力によってグループ内で動的に変動することになる。

生成ルールを適用する時には、大域重みの大きさの順で、各グループを適用していく。同じグループ内の生成ルールの適用順序は、グループ内でのルールの局所重みの大きさによって決まる。グループ内で一番大きい局所重みをもつルールから適用し、適用が成功すれば、次に大域重みの大きいグループに移る。もし適用が失敗したら、グループ内の次に局所重みの大きい生成ルールの適用を行なう。同じグループ内の全てのルールの適用が失敗したら、やはり次に大域重みの大きいグループに行く。

3.3.3 インタープリタ

生成の過程では、インタープリタが重みの一番大きいフレームから生成を行う。各フレームを呼び出す度に、連想表を使う計算によって、そのフレームに適用する生成ルールの重みを計算する。計算された重みに従って、フレームに生成ルールを適用する。また、あるフレームへの生成ルールの適用が失敗した場合(いまだ生成された文章と辻つまが合わない場合)、そのフレームを一旦元に戻し、次のフレームを想起し、生成をおこなう。

インタープリタのアルゴリズムを以下に示す。

1. ワーキング・メモリにあるフレームから一番重みの大きいフレームの一つを選ぶ。フレームが全部なくなったら、生成を終える。
2. 生成ルールをルール・ベースからロードする。連想表等を使う計算によって、選ばれたフレームに適する生

成ルールの重みを計算する。生成ルールの重みの計算の結果、ルールがいくつかのグループに分けられる。

3. まだ適用されていない大域重みの一番大きい生成ルールのグループを「カレント・グループ」とする。

(a) カレント・グループの生成ルールの大域重みが0の時、表層化する生成ルールの大域重みが0に定義されているため、表層化する生成ルールを適用する。それは必ず成功し、文（関係節を含む時もある）を一つ生成する。1へ戻る。

(b) カレント・グループの生成ルールが $rule(Input, Output)$ の形を取る場合、フレームにカレント・グループ内の生成ルールを局所重みの大きい順に適用していく。適用が成功したら、3へ戻る。カレント・グループ内の生成ルールの適用が全部失敗したら、3へ戻る。

(c) カレント・グループの生成ルールが $rule(Input1, Input2, Output)$ 、 $rule(Input1, Input2, Output1, Output2)$ の形を取る場合、ワーキング・メモリに残ったフレームの中から一番重みの大きいフレームをとり、それを第二フレームと呼ぶ。次へ、もし、残りのフレームがなくなったら、3へ戻る。

i. 連想表などから、第二フレームに適する生成ルールの重みを計算する。

ii. 一番大きい重みのグループから、順に第二フレームに生成ルールを適用し、カレント・グループまで適用する。この時、もう一回本アルゴリズムを使う。

それらの適用が成功したら、3へもどる。失敗したら、3cへもどる。

4. 1へ戻る。

3.4 生成ルール

生成ルールは、生成する文章に結束性を持たせるために定義する。KUNGFUでは、以下のような生成ルールが定義されている。

- 話題を決める生成ルール
- 埋め込み文を生成する生成ルール
- 接続詞をつける生成ルール
 - 時間順関係
 - 並列関係

- 前進関係
- 因果関係
- 選択関係
- 逆接関係

- 代名詞化する生成ルール
- 表層化する生成ルール

一つの目的には、普通いくつかの生成ルールを定義する必要がある。KUNGFUでは、同じ目的のためのルールを同じ大きさの重みをつけることによって同じグループにまとめている。以下では、それらのいくつかについて説明する。

3.4.1 話題を決める生成ルール

自然言語の文章を話者の注意状態に着目し、解析することができる。つまり、話者の注意がある一定の時点に於て特定の物事に向けられていて、話者がこの注意状態に従って文章を組み立てていく。注目状態によって、文章の結束性が守られると考えられる。

本研究では、話者の注意する項目を話題（Topic）と呼ぶことにする⁷。文章の生成では、まず、話題が指定され、話題にしたがって文が一つずつ生成される。同時に、話題に関する文脈情報（大域述語）も書き換えられる。ある命題の中のある名詞句が話題と指定されると、その名詞句は命題が表層化するときに文の主語になる。これらの操作は、話題を決めるルールによって定義され、その適用によって実現される。その時、次の三つの大域変数を参照する必要がある。

`current_topic` 現時点の話題を保持する。

`topic_candidate_list` 現在の文に出てくる次の文の話題候補をリストで保持する。

`topic_stack` 前の話題をスタックで保持する。

大域変数は、最初 What to Say 部で初期化され、How to Say 部の生成の過程で常時に書き換えられる。

3.4.2 接続詞をつける生成ルール

入力フレームの表わす意味内容の間の関係がフレームの双方に記されている。例えば、因果関係にある二つのフレームの間には、入力の時点で因果関係を記すスロットに因果関係の情報（因因果か）と相手の ID が入っている。生成の過程では、この二つのフレームが違う順序で想起さ

⁷研究者によっては、話題を焦点（Focus）或は中心（Center）と呼ぶこともある。

1. ○ john is loved by everybody, because he is very kind. he loves dancing and swimming. he is a student at kyoto university which is very famous.
2. ○ john is a student at kyoto university, which is very famous. he is loved by everybody, because he is very kind. he loves swimming and dancing
3. ○ kyoto university is very famous. john is a student there. john loves swimming and dancing. he is loved by everybody, because he is very kind.
4. × kyoto university is very famous. john is very kind, so he is loved by everybody. he is a student at kyoto university. he loves swimming and dancing.

図 3: 同じ内容による出力のバリエーション

れることがある。フレームの生成される順序の違いによって、同じ因果関係であっても、表層化する時につけられる接続詞が違ふ。違ふ接続詞を付けるための生成ルールは、別々に定義されている。

3.4.3 代名詞化する生成ルール

How to Say 部に入力される意味表現は、ニュートラルなもので、言語学的な指示の情報をもっていない。生成の過程で、既出の名詞句を、後に生成される文章中で代名詞によって指示することができる。既出した名詞句のリストを大域変数によって保持し、必要な時に代名詞化のために使うようにしている。代名詞化の条件を満足すれば、代名詞化の生成ルールの適用によって代名詞化を行なう。例えば、固有名詞の代名詞化を代名詞化する生成ルールで表現する。実際の生成の過程では、動的に変わる重みによって、固有名詞を代名詞化するルールが適用されたり、されなかったりする。

3.4.4 表層化する生成ルール

表層化する生成ルールは、中間表現を DCG ルールで表す文の構文構造(木構造)に従って、表層化する生成ルールである。それは、中間表現にどの様な文脈情報が付けられても、吸収できるように定義されている。

4 実験及び考察

本研究では、KUNGFU をつかって文章の生成を行なった。同じ意味内容に対して、いろいろな表現で表現する自然な文章が得られた(図3を参照)。しかし、出力の中には、不自然な文章も含まれる。不自然な文章の生成は、生

成ルールの不足、生成ルールの適用順序の不適切(デフォルトの重みの与え方や重みの計算方法にまだ問題点があるため)、意味ネットワークにおける状況情報の不均衡な与え方等による。

本文章生成システムでは、初期値の与え方に意味付けしにくいことや、重みの計算と記号処理の間に「意味」を互いに結びつけにくいことが最も大きな難点である。

5 おわりに

本研究では、人間に近い文章生成の新しいメカニズムを提案し、それに基づいて文脈に応じたいろいろなバリエーションの文章の生成を行なった。

本研究を通じて、以下のことを明らかにした。

- 文脈を記号情報と数値情報によって定義し、より人間に近いメカニズムで文章を生成した。
- シンボルによる処理とコネクショニズム・モデルによる処理を結びつける一例を示した。
- 生成ルールの適用順序をヒューリスティックスによって規定し、新しい生成の制御方法を提案した。

本研究の提案するメカニズムは人間の情報処理(文章生成)のシミュレーションの性格が強く、実用するためには、今後一層の研究が必要である。

参考文献

- [1] 李航. 文脈に応じたいろいろなバリエーションの文章の生成. 修士論文. 京都大学工学部. 1990
- [2] Kathleen R. McKeown. Text generation. Cambridge University Press. 1985
- [3] 橋田浩一. 制約と言語. ディスコースと形式意味論ワークショップ論文集. ソフトウェア学会「論理と自然言語」研究会. 161-170. 1989
- [4] Christian M. Matthiessen. Notes on the organization of the environment of a text generation grammar. Nature language generation. 278-283. Nijof. 1986
- [5] Christian M. Matthiessen. Systemic Grammar in Computation: The Nigel Case. ISI/RS-83-121. February 1984