

ニューラルコンピューティングの「いろは歌」作成への応用

吉池紀子[†] 北端美紀^{††} 武藤佳恭^{†††}

本論文では、ニューラルコンピューティングの組合せ最適化手法の応用例として、現代語を組み合わせた現代風「いろは歌」の作成方法を紹介する。ここでは、現代風「いろは歌」作成問題を2種類の組合せ最適化問題としてとらえてニューラルコンピューティング手法により解いた。1つ目の制約条件はすべての仮名を重複なく用いるような文節の組を選ぶ問題である。2つ目の制約条件は日本語の係り受け制約に基づいて語順を決める問題である。シミュレーションでは、ツリーサーチによる探索手法とCPU時間による性能の評価を行った。

A Neural Computing Approach for Composing *Iroha-Uta*

NORIKO YOSHIIKE,[†] MIKI KITABATA^{††} and YOSHIYASU TAKEFUJI^{†††}

We present a neural computing approach for composing a new version of *Iroha-Uta* using modern Japanese words and grammar. A new *Iroha-Uta* is composed by satisfying the following two restrictions. One of restrictions is how to chose words that satisfy the rule of *Iroha-Uta* and the other is how to order these words for making sentence based on Japanese grammar. In our simulation, the performance of the proposed algorithm is evaluated in terms of the CPU time comparing with the tree search method.

1. はじめに

現代風「いろは歌」作成問題とは、文学者の間で俳句や川柳とともに知られている作詞問題の1つで、現代の46個の仮名文字すべてを重複なく用いて歌を作るという言葉のパズル的な要素を含んでいる。最もよく知られている「いろは歌」は平安時代に作られたといわれる次のものである。

いろはにほへと ちりぬるを
わかたれそ つねならむ
うゐのおくやま けふこえて
あさきゆめみし ゑひもせす

この歌は47個の旧仮名文字をすべて使いながら、七五調の覚えやすい形式をとり、美しい情景描写と諸行無常の意味を込めた歌であるため、多くの人から慕わ

れている。

本論文では、これまで人の手によって作られていた「いろは歌」を自動的に作成するという新しい試みを行う。ここでは、問題を2つの組合せ最適化問題としてとらえ、ニューラルコンピューティングを用いて解いた。1つ目は、与えられた文節の組からどのように選ぶと「いろは歌」の条件にあてはまるかという文節選択の問題である。2つ目の問題は、選ばれた文節を並べ替えて日本語を作成する文生成の問題である。1つ目の文節選択問題は、どのように現代語を選ぶとすべての仮名を重複なく用いることができるかという言葉のパズル的な性質を持っている。この問題は、正方形を複数つないだ形を長方形内に重ならないように敷き詰めるポリオミノパズルと類似している。ポリオミノパズルの解の探索法に対しては多くの研究がなされているが、ニューラルネットワークを利用した並列アルゴリズムが有効であることが知られている。Akiyamaら¹⁾やKajiuraら²⁾は局所解から脱出するためにノイズを用いた確率的なニューラルネットワークモデルを提案している。Takefujiら³⁾はHopfieldが最適化手法として提案するニューラルネットワークモデル⁴⁾を拡張し、局所解脱出のためにエネルギー関数にヒルクライム項を導入して大規模問題への適用に成功している。ここでは、Takefujiらの手法を用いて、数百

[†] 慶應義塾大学大学院政策・メディア研究科
Graduate School of Media and Governance, Keio University

^{††} NTT生活環境研究所生活情報流通研究部ホームコミュニケーション研究グループ

NTT Home Communication Research Group, Human Communication Laboratory, Lifestyle and Environmental Technology Laboratories

^{†††} 慶應義塾大学環境情報学部
Faculty of Environmental Information, Keio University

の語句の中から十数個の語句を選ぶという規模の大きい組合せ問題を解く方法を紹介する。2つ目の文生成問題は、文節の並べ替えを行って日本語を生成する問題である。自然な日本語で見られる係り受け制約に基づいて語順を決めるこの問題では、すでに用いる語句が選ばれているため語句間の関係を定める係り受け制約に従うペアを求めることが必要である。この問題に対しても複数の制約条件を同時に扱うことのできる Takefuji らの提案するニューラルコンピューティング手法⁵⁾を適用することが有効である。ニューラルコンピューティング手法がこの種の組合せ最適化問題に有効であることが多数報告されている^{6),7)}。

2. 文節選択のニューラルネットによるモデル化

この章では、1つ目の現代語を選択する問題についてニューラルネットワークを用いた手法の詳細を示す。ポリオミノパズルに対する Takefuji らのニューラル表現と同様にサブニューロンの付随したニューロンモデルを用いる。ここでは、文節の単位で区切られた現代語を「いろは歌」の制約条件にあてはまるように選択する問題を考える。

2.1 制約条件

1段階目の文節選択の問題のための制約条件は以下の2つである。

- 同じ文字を重複して使用しない。
- “あ”から“ん”まで46文字すべての平仮名を使う。

2.2 ニューラル表現

データとして与えられた文節に対し、その文節を使用するか使用しないかを表すニューロン V_i , ($i = 1, \dots, n$) (ここでの n は文節数) を用意する。

文節 i を使用することを $V_i = 1$, 使用しないことを $V_i = 0$ とすると、選択された文節の組は0か1を要素に持つニューロン配列の状態の1つとして表される。ニューロンの発火の様子を図1に示す。図中の配列において、白い四角はニューロンの値が0であることを示し、黒い四角は値が1であることを示している。選ばれた文節に対し、それぞれの仮名文字が重複するかどうかを調べるため、ニューロンの発火にともなって発火するサブニューロンを用意する。このサブニューロンは2次元配列 V'_{ij} , ($i = 1, \dots, n$), ($j = 1, \dots, 46$) として表すことができ、文節 i に仮名 j が使われるか否かを表す。つまり、 V'_{ij} は、文節 i が選択されていて ($V_i = 1$)、かつ文節 i 仮名 j が含まれているとき $V'_{ij} = 1$, それ以外るとき $V'_{ij} = 0$ となる。ここで、 l は文節 i に含まれている仮名 j の個数を示してい

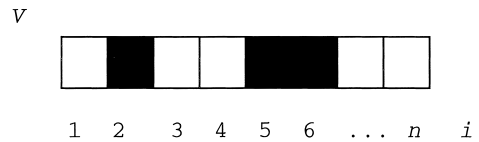


図1 ニューロン発火の様子

Fig.1 Neural representation for the problem.

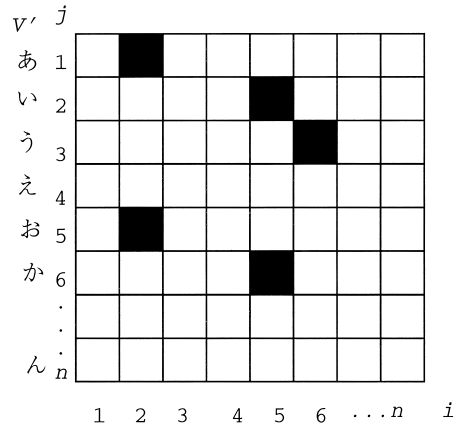


図2 サブニューロン発火の様子

Fig.2 A state of subneuron matrix.

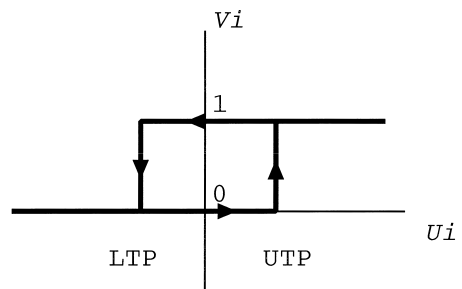


図3 ヒステリシス-マッカロックピッツ関数

Fig.3 Hysteresis McCulloch-Pitts input/output function.

る。仮名の種類は {あ:1, い:2, ..., ん:46} とそれぞれ番号づけられている。図1に対応するサブニューロンの発火の例を図2に示す。図中の配列のうち白い四角はニューロンの値が0であることを示し、黒い四角は値が1以上であることを示している。

2.3 動作式

ニューロン V_i は式 (1), 図3で表されるヒステリシス-マッカロックピッツニューロンを用いる。

$$V_i = \begin{cases} 1 & \text{if } U_i > UTP \\ 0 & \text{else if } U_i < LTP \\ V_i & \text{otherwise} \end{cases} \quad (1)$$

ここでは、 $UTP = 1$, $LTP = -1$ とした。ヒステ

リシス-マッカロックピッツニューロンはマッカロックピッツニューロンモデル⁸⁾を拡張したものであり、擬似的に非同期にニューロンを発火させる働きを持つため、大規模な組合せ問題では有効である⁵⁾。ニューロン V_i の状態値 U_i は式 (2) で表される一次のオイラー法で変化させるものとする。

$$U(t+1) = dU(t+1) + U(t) \quad (2)$$

文節選択の制約条件を満たすようにニューロンの状態を変化させるニューラルネットワークの動作式は次式ようになる。

$$\frac{dU_i}{dt} = -A \times \sum_{b=1}^{m_i} \left(\sum_{a=1}^n V'_{ak_{ib}} - 1 \right) \times \frac{1}{m_i} + B \times \text{hill} \left(\sum_{b=1}^{m_i} \sum_{a=1}^n V'_{ak_{ib}} \right) \times m_i \quad (3)$$

ここで動作式の係数 A, B はステップ数 t によって変化するパラメータ, m_i は文節 i の文字数, k_{ic} は文節 i の c 文字目の仮名の種類, n は文節群に含まれる文節の数である。

動作式の第 1 項目は、各仮名文字を重複せずに使用するという第 1 の制約条件を満たすように働く項である。第 2 項目は、“あ” から “ん” までのすべての平仮名を使用するという第 2 の制約条件を満たすよう働く項である。第 2 項目の $\text{hill}(x)$ は、局所解 から抜出す働きをする式 (4) で与えられる関数である。

$$\text{hill}(x) = \begin{cases} 1 & \text{if } x < m_i \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

式 (3) では、第 1 項は仮名文字の数 m_i の逆数を掛け、第 2 項は仮名文字の数を掛けることで、文字数の多い文節を優先して選ぶように重みづけがされている。

3. 文生成のニューラルネットによるモデル化

この章では、2 つ目の問題である選択された文節を並べ替える文生成問題に対して、ニューラルネットワークを用いた手法の詳細を示す。ここでは、日本語の係り受け制約に従った文節の並べ替えを行うことにより、自然な日本語文生成を目的とする。係り受け制約とは、文節間の関係に見られる制約のことである^{9),10)}。文生成の手法として、句構造文法が用いられることがあるが、英語に比べて日本語を句構造文法で表現すると複雑になるという欠点がある¹¹⁾。ここでは、文節どうしの二項関係に着目し、文節ごとに分散化された制約条

件を扱うことでニューラルネットワークで解を求めることを可能にした。なお、係り受けを制約に基づいて決定する問題は、ツリーサーチなど他の手法を用いても解くことが可能であるが、文節選択の段階との統一を考慮してここではニューラルコンピューティング手法を紹介する。

3.1 制約条件

係り受け制約とは、自然な日本語の文節どうしの関係に見られる制約条件である。係り受けにともなう制約条件には以下のものがある。

1. 接続可能性：2 文節単位でみたときに文法的に接続可能であるものと係り受け対応する。
2. 非交差性：文節間の係り受けは互いに交差しない。
3. 係り先専有性：最後の文節を除いたすべての文節は自分より後方の文節いずれか 1 つに係る。
4. 単近接続性：「距離的に近い」文節間ほど係り受けが成立しやすい。

各項目を以下に詳しく説明する。

1. 接続可能性は係り受け対応が文法規則に従うようにするためのものである。ここでは規則を簡単にするため、次の 3 通りの係り受け対応のみを考え、用言の活用、助詞助動詞の用法は考慮しないことにする。

- 連用詞句 - 動詞句
- 連体詞句 - 名詞句
- 名詞句 - 動詞句

2. 非交差性は文の構造的な制約であり、一般的に交差を含む文は意味があいまいになるため用いられることが少ない。交差を含まない文と含む文の例を図 4、図 5 にあげる。図 5 の「私は太郎君が見た、キャッチボールをしていた。」という文は主語述語の関係が曖昧になって伝わってしまうため、一般的には用いられない。

3. 係り受け先専有性とは、係り先の文節と係り元の文節が一对多の関係であることを示している。つまり、係り受け専有性の満たされた文は最後の文節を根

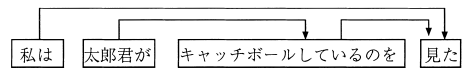


図 4 交差を含まない文の例

Fig. 4 An example of dependency structure including no intersections.

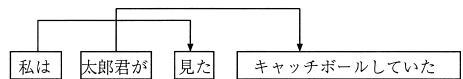


図 5 交差を含む文の例

Fig. 5 An example of dependency structure including intersections.

ここでの局所解とは、ニューロンの状態が解ではないデッドロックに陥った状態を指す。

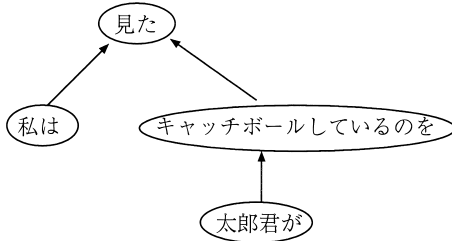


図 6 ツリー表記の例
Fig. 6 Tree representation.

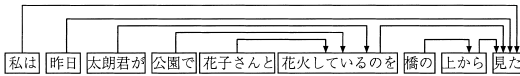


図 7 係り受けの文節間の距離が長い文例
Fig. 7 An example of dependency structure with long distance.

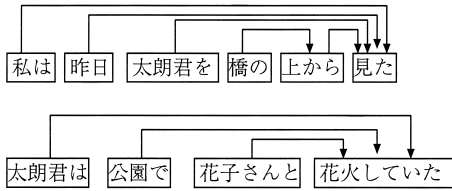


図 8 係り受けの文節間の距離が短い文例
Fig. 8 An example of dependency structure with short distance.

とするツリー構造で書き表すことができる。「私は太郎君がキャッチボールしているのを見た。」という文をツリーで表したものを図 6 に示す。

4. 卑近接続性は自然に生成された文においては係り受けの文節間の距離が離れた複雑な文は作られにくいという性質である。係り受けの文節間の距離が遠い文、近い文の例を図 7、図 8 にあげる。図 7 の例文は、図 8 のように 2 つの文に分割することによって、係り受け関係にある文節間の距離を短くすることができる。係り受けの文節間の距離は近い方が構造が分かりやすく、意味をとらえやすい。

これら 4 つの制約を考慮して、自然な日本語に近い文の生成を行うことが目的である。ここでは、1. 接続可能性、3. 係り先専有性、4. 卑近接続性の制約条件を満たす係り受けのペアをニューラルネットワークによって求め、2. 非交差性を満たす語順を求めるため後処理を行う。

3.2 ニューラル表現

ニューラルネットワークで係り受けのペアを決めるために、ニューロン X_{ij} , ($i, j = 1, \dots, l$) (l は選ばれた文節の数) を用意し、文節 i から文節 j へ係り

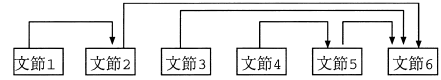


図 9 ニューロンの発火の様子
Fig. 9 Neural representation.

受けが行われることを $X_{ij} = 1$ 、行われなかったことを $X_{ij} = 0$ と表す。図 9 に文節の係り受けに対応するニューロンの発火の様子を表す。

3.3 動作式

ニューロン X_{ij} は式 (5) で表されるヒステリシス-マッカロックピッツニューロンモデルで、ニューロン X_{ij} の内部状態 W_{ij} は一次のオイラー法で変化させるものとする。

$$X_{ij} = \begin{cases} 1 & \text{if } W_{ij} \geq 0 \\ 0 & \text{else if } W_{ij} \leq -1 \\ X_{ij} & \text{otherwise} \end{cases} \quad (5)$$

係り受けペアを作る際に制約条件 1. 3. 4. を考慮して、ニューロン i, j の内部状態 W_{ij} の更新式は以下のように定義する。

$$\begin{aligned} \frac{dW_{ij}}{dt} = & -A \times \left(f_1 \left(\sum_{a=0}^{n-1} X_{ai} + \sum_{a=0}^{n-1} X_{ia} \right) \right. \\ & \left. + f_1 \left(\sum_{a=0}^{n-1} X_{aj} + \sum_{a=0}^{n-1} X_{ja} \right) \right) \\ & - B \times f_2 \left(\sum_{a=0}^{n-1} X_{ia} \right) - C \times \sum_{a=0}^{n-1} X_{aj} \\ & - D \times (1 - e_{h_i h_j}) \times X_{ij} \end{aligned} \quad (6)$$

動作式の第 1 項目は係り受けのペアが必ず 1 つ存在するように働く項であり、関数 $f_1(x)$ は式 (7) によって定義される。

$$f_1(x) = \begin{cases} -1 & \text{if } x < 1 \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

第 2 項目は 3. 係り先専有性を満たすためのもので

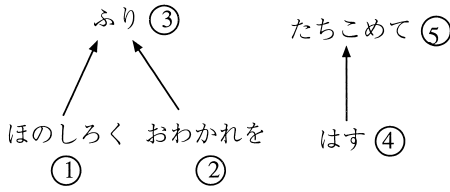


図 10 交差を防ぐ語順決定の例
Fig. 10 Method for avoiding intersections.

あり，関数 $f_2(x)$ は式 (8) によって定義される．

$$f_2(x) = \begin{cases} 1 & \text{if } x > 1 \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

第 3 項目は 4. 卑近接続性に対応するコスト関数であり，1 つの文節へ係り受けが集中しないようにすることで離れた文節どうしがペアになることを防いでいる．

第 4 項目は 1. 接続可能性を満たすための項である． $e_{h_i h_j}$ は文節 i に含まれる単語 h_i と文節 j に含まれる単語 h_j が係り受け可能かどうかを示しており，係り受け可能なときに 1，可能でないときに 0 とする．係り受け可能な文節間の関係は次の 3 種類の文節関係とする．

- 連用詞句 → 動詞句
- 連体詞句 → 名詞句
- 名詞句 → 動詞句

3.4 語順決定

ニューラルネットワークで係り受けのペアを求めた後，2. 非交差性を満たすような語順決定処理を行う．すでに 3. 係り先専有性が満たされているため，文節間の係り受け関係をツリーを使って表すことができる．このツリーに対し葉に近い順に番号をふることによって交差を防ぐ（図 10）．

4. シミュレーション結果

文節選択の処理と文生成のシミュレーション結果を示す．文節選択のシミュレーションでは，ツリーサーチによる厳密解法と本手法の実行時間による比較を行う．次に与えるデータ数を変えたときの収束率の変化を示す．さらに特定の 964 語の文節群を与えたときの実行結果を示す．文生成のシミュレーションでは，選ばれた文節を本手法によって並べ替えた結果と係り受け構造を示す．

4.1 文節選択のシミュレーション

すべての文節選択のシミュレーションを通して，ニューラルネットワークの動作式のパラメータ A, B の初期値は $A = 1, B = 1$ とし，50 ステップごとに $A = A + 1, B = B / 5$ という演算を行った．本手法を

表 1 ツリーサーチ手法との CPU 時間における比較
Table 1 Comparisons with tree search method.

データ数	ツリーサーチ		本手法	
	CPU 時間 (秒)	解	CPU 時間 (秒)	解
100	0.11	×	11.06	×
200	979.74	×	46.00	×
300	77808.90	×	102.38	×
400	260.17		56.62	
500	2491.38		339.87	×
600	12464.15		154.71	
700	988.14		676.84	×
800	233.65		173.19	
900	1057.40		649.30	
1000	2444.13		349.98	

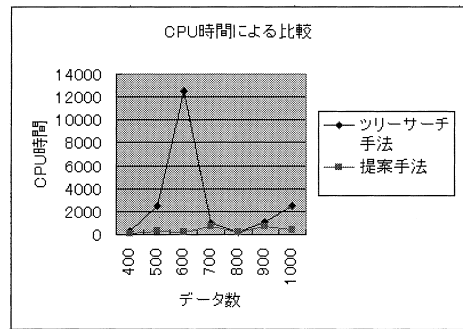


図 11 CPU 時間による比較

Fig. 11 Comparisons with tree search method in terms of CPU-time.

C 言語でプログラムし，Sun Ultra2 (Ultra SPARC 170 MHz × 2, SunOS 5.5.1) 上でシミュレーションした．それぞれの実験の結果を以下に示す．

4.1.1 CPU 時間における比較

深さ優先のツリーサーチ手法と本手法により文節選択を行った際のデータ数による CPU 時間の変化を表 1 に示す．それぞれの手法により，解を得られた場合，得られなかった場合 × を記した．本手法では乱数のシードを固定した場合の実行結果を示している．1000 回の繰返し計算で収束しなかった場合，解を得られなかったと判定した．このうち，厳密解法であるツリーサーチによって解が示された 400 以上のデータ群に対して CPU 時間を図 11 に示す．

本手法を用いた場合，データ数 500 やデータ数 700 のように解を得られない場合もあるが，CPU 時間はほぼ一定していることが分かる．これに対し，ツリーサーチは必ず解を得るが CPU 時間は与えられたデータの順番に依存するため，データ数 600 のように解を得るまでに多くの時間がかかることがある．与えるデータの中に解が存在しているかどうかは多くの場合未知であるため，データ数 100 からデータ数 300 の場

表 2 データ数による収束率の変化

Table 2 Changes of convergence rate with number of data.

データ数	収束率 (%)
100	0
200	0
300	0
400	49
500	30
600	67
700	93
800	96
900	94
1000	93

合のように解が存在しない場合はツリーサーチによって解を求めると多くの CPU 時間を消費してしまう可能性がある。このため、「いろは歌」の文節選択問題に対してはこのような厳密解法は適していないことが分かる。

4.1.2 収束率

データ数を増やしたときの収束率の変化を表 2 に示す。データ数 700 以上の条件で、収束率が 90% を超えている。CPU 時間による比較実験 (表 1) において、本手法で解が得られない場合があったが、十分なデータ数を用意すれば一定時間内で解を得られる確率が高いことが分かる。

4.1.3 文節選択結果

次に、稲垣の本¹²⁾の中から任意に文節を選び、合計 964 語の文節群を用意して 1 段階目の文節選択のシミュレーションに用いた。この文節群の属性の分布、文節長の分布は表 3、表 4 のとおりである。データとして与える文節群が文字数 1 や 2 のような短い文節のみで構成される場合、「いろは歌」の解を得ることは容易である。しかしながら、短い文節のみから文を生成すると歌の意味をとえらにくくなる。逆に、文節群が文字数 5 以上であるような長い文節のみで構成される場合、解を得ることが困難になる。このため、表 4 のような長い文節と短い文節が適度に含まれる文節群が望ましい。今回のシミュレーションでは初期値を変えて 1000 回のシミュレーションを行い、繰返し数が 1000 回を超えても「いろは歌」の制約条件を満たす解の求まらないものに関しては収束しなかったものとした。この結果、収束率は 93.3% であった。以下にシミュレーションの結果得られた文節群の例を 2 つ示す。この文節群は、制約条件である「いろは歌」の 2 つのルールに従ったものである。

表 3 文節群の属性の分布

Table 3 Distribution of attributes of data.

属性	度数
名詞句	505
動詞句	300
連体詞句	97
連用詞句	62

表 4 文節群の文節長の分布

Table 4 Distribution of length of data.

文字数	度数
1	3
2	142
3	395
4	258
5	122
6	33
7	9
8	2

結果 1:

ひぬけようとあさほのしろく
ふりやねへなにもたちこめて
いえみせゆきはす
らんつむおわかれをそまる

結果 2:

そんなぬすまれたほうへ
つきひとのけむりにわかあめ
おをみせよふゆはて
もえるいちやさくらしろねこ

4.2 文生成のシミュレーション

2 段階目の文生成では、1 段階目で選ばれた文節の属性 ({0:名詞句,1:動詞句,2:連体詞句,3:連用詞句} のうちいずれか) を入力データとする。ニューラルネットワークの動作式の係数を $A = 1, B = 1, C = \max(0, 0.5 - 0.15 \times (t/50)), D = 1$ とした。C をステップ数とともに減少させた。

シミュレーション結果を以下に示す。それぞれ、文節選択における結果 1、結果 2 を並べ替えたものである。また、結果 1、結果 2 に対する係り受けの関係を図 12、図 13 に示す。

結果 1 (並べ替え後):

ほのしろくおわかれをふり
はすたちこめて
やねへなにもぬけようと
いえつむ
ゆきらんみせ
あさひそまる

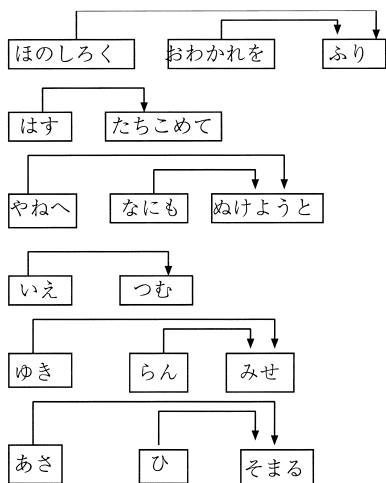


図 12 結果 1 の係り受け関係
Fig. 12 Dependency structure of Result 1.

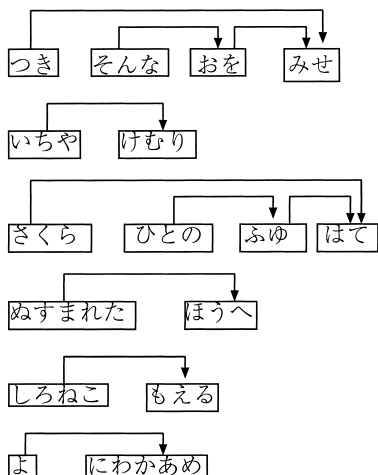


図 13 結果 2 の係り受け関係
Fig. 13 Dependency structure of Result 2.

結果 2 (並べ替え後):

つき	そんな	おを	みせ
いちや	けむり		
さくら	ひとの	ふゆ	はて
ぬすまれた	ほうへ		
しろねこ	もえる		
よ	にわかあめ		

結果 1, 結果 2 の係り受け関係から, 係る文節対受けの文節の数の比率が 1 対 1, あるいは 2 対 1 となり, 近い文節どうして係り受けが起きていることが分かる. 本手法によって作成された文では「いえ → つむ」などの意味の通りにくい係り受けが起こっている. そのため文全体の意味が分からなくなっているものが

出現している. 今後の課題として, 今回用いた係り受けの制約に加えて文節間の係り受け関係の整合度を評価をすることが必要である. 自然言語の構文解析の手法として係り受けの整合度の評価を行う研究が近年多く行われている^{13)~15)}. これらの手法を文生成に応用することが可能であり, 係り受け結合の起こりやすさを事例から学習することで, 生成される文をより自然な文に近づける効果が期待できる.

5. おわりに

本論文では, 新しい「いろは歌」を作成するためのニューラルコンピューティングを用いた手法を提案した. 本手法では一定時間内で制約を満たす文節の組を選ぶか選べないという判定をすることができ, 乱数の初期値によって異なる組合せを得ることができるため, 新しい「いろは歌」の候補を数多くを生成することが可能である. 選ばれた文節を並べ替えた際に自然な係り受け関係を生成することが今後の課題である.

参考文献

- 1) Akiyama, Y., Yamashuta, A., Kajiura, M. and Aiso, H.: Combinatorial Optimization with Gaussian Machines, *Proc. Int. Joint Conf. on Neural Networks*, I, pp.533-540 (1989).
- 2) Kajiura, M., Akiyama, Y. and Anzai, Y.: Solving Large-scale puzzles with Neural Networks, *Proc. IEEE Workshop on Tools for Artificial Intelligence*, pp.562-569 (1989).
- 3) Takefuji, Y. and Lee, K.C.: A Parallel Algorithm for Tiling Problems, *IEEE Trans. Neural Networks*, Vol.1, No.1, pp.143-145 (1990).
- 4) Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities, *Proc. Natl. Acad. Sci. USA*, 79, pp.2554-2558 (1982).
- 5) Takefuji, Y.: *Neural Network Parallel Computing*, Kluwer Academic Publishers (1992).
- 6) 武藤佳恭: ニューラルネットワーク, 産業図書 (1996).
- 7) 武藤佳恭: ニューラルコンピューティング, コロナ社 (1996).
- 8) McCulloch, W.S. and Pitts, W.: A logical calculus of the ideas imminent in nervous activity, *Bull. Math. Biophys.*, 5, pp.115-133 (1943).
- 9) 長尾 真(編): 自然言語処理, 岩波書店 (1996).
- 10) 清水浩行, 佐藤秀樹, 林 達也: ニューラルネットを利用した日本語係り受け解析—係り受け解析の数値問題としての取り扱いについて, 電子情報通信学会論文誌 (DII), Vol.J80-DII, No.9, pp.2457-2465 (1997).

- 11) 橋田浩一：自然言語の構文解析と文生成の統合，*情報処理*，Vol.33, No.7, pp.790-800 (1992).
- 12) 稲垣足穂：一千一秒物語，新潮社 (1969).
- 13) 江原暉将：係り受け整合度を計算するいくつかの統計的手法の比較，*情報処理学会研究会報告*，98-NL-126, pp.25-30 (1998).
- 14) 江原暉将：最大エントロピー法を用いた日本語係り受け整合度の計算，*言語処理学会第4回年次大会発表論文集*，pp.382-385 (1998).
- 15) Yamazaki, T. and Ido, D.: Mistake-driven learning with thesaurus for text categorization, *Proc. Natural Language Processing Pacific Rim Symposium 1997*, pp.369-374 (1997).

(平成 12 年 8 月 25 日受付)

(平成 12 年 10 月 19 日再受付)

(平成 12 年 12 月 5 日採録)



吉池 紀子 (学生会員)

昭和 50 年生。平成 11 年慶應義塾大学大学院政策・メディア研究科修士課程修了。現在、同研究科博士課程に在学中。自然言語処理や動画認識処理へのニューラルネットワーク

の応用に興味を持つ。



北端 美紀

昭和 51 年生。平成 11 年慶應義塾大学大学院政策・メディア研究科修士課程修了。同年 NTT 生活環境研入社。画像認識，マルチモーダルインタフェースの研究に従事。



武藤 佳恭

昭和 30 年生。昭和 58 年慶應義塾大学大学院博士課程電気工学専攻修了。工学博士。現在，ケースウエスタンリザーブ大学電気工学準教授，慶應義塾大学環境情報学部教授。ニューラルコンピューティング，ハイパースペクトラルコンピューティング等の研究に従事。情報処理学会 20 周年記念論文賞 (1980)，IEEE Trans. on Neural Networks 功労賞 (1992) 授賞。