

# モンテカルロ木探索を用いた早口言葉生成

川端 風貴<sup>†</sup>      加藤 芳秀<sup>‡</sup>      松原 茂樹<sup>†</sup>

<sup>†</sup> 名古屋大学大学院情報科学研究科

<sup>‡</sup> 名古屋大学情報連携統括本部

kawabata@db.ss.is.nagoya-u.ac.jp

## 1 はじめに

近年、文生成の分野において、様々な種類の文を生成する試みがなされている。熊谷らはモンテカルロ木探索を用いた、統語構造と単語のつながりを評価する文生成手法を提案している [2]。この手法では、文らしさを評価する評価尺度と文生成のための文脈自由文法が与えられると、その評価尺度の値が大きくなるような文を生成することができる。文生成の目的によっては、生成される文が文らしいだけでなく、その他の性質を持たせたい場合があるが、このとき文をどのように評価すればよいかは明らかではない。そこで本稿では、モンテカルロ木探索を用いた文生成において、早口言葉という発声のしにくさと文らしさの2つの特徴を併せ持った文の生成を例に、独立に定義された複数の評価尺度を使って文生成を行う手法を提案する。

## 2 モンテカルロ木探索を用いた文生成

本節では、熊谷ら [2] が提案したモンテカルロ木探索に基づく文生成について、その概要を述べる。まず、モンテカルロ木探索について説明する。

### 2.1 モンテカルロ木探索

モンテカルロ木探索 (以下, MCTS) は囲碁や将棋において、勝率の高い次の一手を導出する場合などに有効なアルゴリズムである [1]。MCTS はゲーム木探索の一種であり、現在の状況をルートノードとして、以下の4つの処理を繰り返し、子ノードの勝率を計算し、勝率の高い子ノードを導出することで最適な次の一手を決める。図1は各処理を図示したものである。

### 1. 選択

ルートノードを起点として、子ノードをたどり、拡張可能なノードを選択する。この選択は以下のUCB1値と呼ばれる尺度に基づきなされる。

$$v_i + C \sqrt{\frac{\log N}{n_i}} \quad (1)$$

$i$  はノードを識別するための番号,  $v_i$  はノード  $i$  の勝率,  $C$  は調整係数,  $N$  は全試行回数,  $n_i$  はそのノード  $i$  を訪れた回数を表す。UCB1値の第1項により、勝率の高いノードが優先され、第2項により探索回数の少ないノードが優先的に選択されることになる。これにより、勝率の高さと探索回数の少なさを考慮したバランスの取れた探索がなされる。

### 2. 拡張

選択したノードに対して未実行のアクション (ゲームの一手) を適用し、選択したノードに子ノードを1つ追加する。

### 3. シミュレーション

拡張により追加された子ノードを起点として、ランダムにアクションを適用し終端ノードを求める。

### 4. 逆伝播

シミュレーションにより得られた終端ノードの状況 (勝敗) に応じて、拡張により追加されたノード及びその祖先のノードの勝率を更新する。

### 2.2 MCTS の文生成への応用

MCTS では、途中状態の評価は一切せず、最終状態の評価のみを行っている。これを利用して、熊谷らは、文脈自由文法 (以下, CFG) による文の導出を木探索と見立てて、MCTS に基づき文生成する手法を提案している。Algorithm 1 はその文生成のアルゴリズム

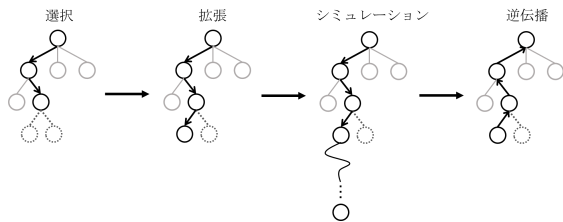


図 1: モンテカルロ木探索の 4 つの処理

である。生成される文候補について勝敗を決めることはできないため、この手法では、それまでに生成された文候補の平均スコアを超えた時に、それを勝ちとみなし、MCTS を実行する。

### 3 提案手法

前節で述べた熊谷らの MCTS を用いた文生成では、文の構造の正しさ、及び単語のつながりを評価している。しかし、文生成においてその他のファクターについて評価する方法は明らかではない。本稿では早口言葉生成を扱うが、早口言葉生成においては、少なくとも発声のしにくさ (= 早口言葉らしさ) と文らしさの 2 つの観点から評価しなければならない。しかし、複数の観点を総合的に評価する一般的な方法は明らかではなく、その方法を確立する必要がある。そこで、本節ではそれぞれ独立に定義された複数の評価尺度を用いて、MCTS に基づき文生成を行う一般的な方法を提案する。

#### 3.1 複数の観点からの生成文の評価

複数の観点から文を評価する場合、それらを総合的に評価する尺度が定義できるかどうかは自明ではない。総合的な評価尺度を定義することに比べれば、各観点ごとに評価する尺度を独立に定義することは容易である。提案手法では、熊谷らの手法を拡張して、複数の観点から文を評価するが、複数の観点を総合的に評価する尺度を求めるのではなく、それぞれの評価尺度で評価し、勝率によって複数の観点を総合する。すなわち、提案手法では、各ノードの勝率を以下の式で表される各評価尺度での勝率の相乗平均とする。これにより、MCTS に基づく文生成において複数の評価尺度を反映した探索がなされ、いずれの評価尺度において

---

#### Algorithm 1 MCTS を用いた文生成

---

```

function GENERATESENTENCE
  //S は開始記号
  root ← S
  while root has nonterminal symbols do
    root ← search(root)
  end while
  return root.sentence
end function

function SEARCH(root)
  for 1, ..., iterate do
    visitedNodesHistory ← [root]
    expandedNode ← select(visitedNodesHistory)
    newNode ← expand(expandedNode)
    visitedNodesHistory.add(newNode)
    win ← simulate(newNode)
    backPropagate(win, visitedNodesHistory)
  end for
  return highestWinRatioNodeIn(root.children)
end function

function SELECT(visitedNodesHistory)
  root ← firstNodeOf(visitedNodesHistory)
  while not exists untried rules to root do
    root ← highestUCB1NodeIn(root.children)
    visitedNodesHistory.add(root)
  end while
  return root
end function

function EXPAND(expandedNode)
  //未適用のルールをランダムに返す
  untriedRule ← untriedRuleTo(expandedNode)
  //ルールを適用して生成される新しいノードを返す
  newNode ← untriedRule.apply(expandedNode)
  add a newNode to expandedNode's children
  return newNode
end function

function SIMULATE(newNode)
  while newNode has nonterminal symbols do
    untriedRule ← untriedRuleTo(newNode)
    newNode ← untriedRule.apply(newNode)
  end while
  //平均スコアの更新
  updateAverageScore(newNode)
  score ← evaluate(newNode)
  if score > averageScore then
    return true
  else
    return false
  end if
end function

function BACKPROPAGETE(win, visitedNodesHistory)
  incrementVisitedCount(visitedNodesHistory)
  if win then
    incrementWinCount(visitedNodesHistory)
  end if
end function

```

---

も高い値をもつ文の生成が期待できる。

$$v_i = \sqrt[n]{\prod_j^n \text{評価尺度 } j \text{ での勝率}} \quad (2)$$

### 3.2 生成された文の評価尺度

提案手法では早口言葉を評価するにあたり、発声のしにくさについての評価尺度と文らしさについての評価尺度を用いる。以下では、2つの評価尺度を順に説明する。

#### 3.2.1 発声のしにくさの評価

提案手法では、発声のしにくさを定義するために、奏らの早口言葉に関する調査（1964, 鈴木ら [4] の引用による）を参考にした。奏らは、どのようなモーラの連続が発声をしにくいのかという条件を以下のようにまとめている。

1. 音節主音として母音音素/i/あるいは/u/を持つ音節の連続：/hihihifito/「ひしひしと」
2. 音節主音として母音音素/a/を持つ音素の連続：/atatakai/「暖かい」
3. 子音音素の次に半母音音素/j/が位置している音節の連続：/sjuzjucu/「手術」
4. 摩擦音・歯擦音系列の子音音素を音節副音としてもつ音節の連続：/shicizi/「7時」
5. 両唇音の子音音素を音節副音としてもつ音節の連続：/mimamoru/「見守る」
6. 歯茎の系列子音音素を音節副音としてもつ音節の連続：/banananado/「バナナなど」
7. 「ラ行音」+「ラ行音」
8. 「サ行音」+「シャ・シュ・シヨ」
9. 「シャ・シュ・シヨ」+「サ行音」

提案手法では、これを参考に発声のしにくさを評価する尺度を定義する。具体的な導出方法を以下に示す。

1. 与えられた文をひらがなに変換する。次に、モーラを基本単位とした 2-gram を取り出す。

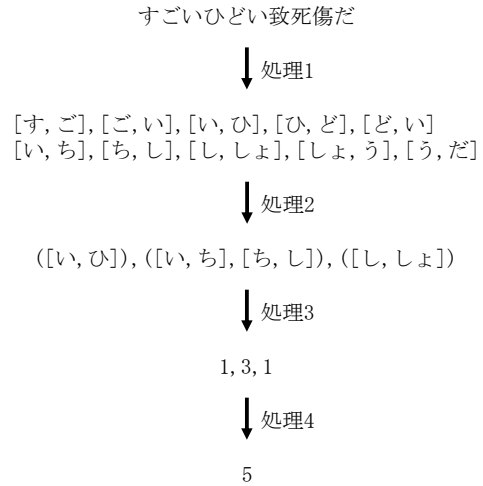


図 2: 発声のしにくさのスコア導出の流れ

2. 各 2-gram に対して、上の条件を満たすかどうかを判定し、満たさないものは取り除く。同じ条件を満たしている 2-gram が連続している場合、それらを 1 組としてまとめる。連続していない場合は、単一の 2-gram を 1 組とする。
3. 各組のスコアを求める。各組のスコアは組中のモーラの  $n$ -gram の総数とする。
4. 各組のスコアを足し合わせ、文全体の発声のしにくさを表すスコアとする。

例として「すごいひどい致死傷だ」という文に対する発声のしにくさに関するスコアの算出について述べる（図 2 参照）。処理 1 では、ひらがな表記にした対象文のモーラ単位での 2-gram を導出している。処理 2 では、導出した 2-gram の中から、上述の条件を 1 つも満たしていないものを取り除いて、連続しているものを 1 つの組としてまとめる。[い, ひ], [い, ち], [ち, し] が条件 1 を満たし, [し, しょ] が条件 8 を満たしている。また, [い, ち], [ち, し] は連続していて同じ条件を満たしているものなので, 1 つの組となる。処理 3 では、各組について  $n$ -gram の総数を求める。組 ([い, ひ]) は 2-gram が 1 つ含まれるのでスコア 1, 組 ([い, ち], [ち, し]) は 2-gram が 2 つ, 3-gram が 1 つ含まれるのでスコア 3, 組 ([し, しょ]) は 2-gram が 1 つ含まれるのでスコア 1 となる。最後に、処理 4 で各組のスコアを足し合わせて文全体のスコアは 5 となる。

表 1: 生成例

| 使用した評価尺度          | 生成された文                          | 文らしさ   | 発声のしにくさ |
|-------------------|---------------------------------|--------|---------|
| 文らしさ              | この少数民族が語る。                      | 0.0889 | 9       |
|                   | かなってるのか同じなのかによる。                | 0.0885 | 0       |
| 発声のしにくさ           | あからさまに引き続き優れて過ぎるからや。            | 0.0837 | 44      |
|                   | 率いるざらざらにはああいうコンチネンタルミクロネシアが続いた。 | 0.0840 | 35      |
| 文らしさ +<br>発声のしにくさ | 切実にしきりと引きつけて過ぎるからだ。             | 0.0887 | 31      |
|                   | また改めてからはああなっている。                | 0.0976 | 21      |

### 3.2.2 文らしさの評価

提案手法では、文としてのもっともらしさを評価するために、 $n$ -gram モデルを使用する。形態素列  $w_1, \dots, w_n$  が与えられたとき、評価式は以下の式に表されるように、形態素 2-gram と形態素 3-gram の生起確率の和とした。

$$\begin{aligned} \text{Score} &= 2\text{-gram 生起確率} + 3\text{-gram 生起確率} \\ &= \prod_{i=1}^{n-1} P(w_{i+1}|w_i) + \prod_{i=1}^{n-2} P(w_{i+2}|w_i w_{i+1}) \end{aligned}$$

## 4 生成された早口言葉の例

本節では、提案手法を使って実際に生成された早口言葉の例を挙げる。CFG の作成と  $n$ -gram モデルには、京都大学テキストコーパス [3] の係り受け解析と形態素解析の施された 8,529 文を使用した。MCTS のシミュレーション回数は 30,000 回に設定した。生成例を表 1 に示す。文らしさのみを評価尺度として使った場合と、発声のしにくさのみを評価尺度として使った場合と、両方の評価尺度を使った場合の 3 つの場合について生成を行った。各評価尺度を単独で用いた場合、使用した尺度は高くなるが、使用していない尺度は低くなった。一方、両方の尺度を用い、勝率の相乗平均を用いた提案手法では、両尺度が高い値となっている。このことから、提案手法は、複数の評価尺度を考慮した文生成ができることが確認できた。

## 5 おわりに

本稿では、MCTS を用いた文生成を、独立に定義された複数の評価尺度を用いて行う方法を提案した。生成例では、発声のしにくさと文らしさの 2 つの評価尺度を用いて、早口言葉の生成を行った。提案手法により、実際に 2 つの評価尺度において高い値を持つ文が

生成されていることを確認できた。今後は、より早口言葉らしさを表せるような評価尺度を考案し、生成された文を人手で評価する実験を実施したい。

## 参考文献

- [1] Cameron Browne, Edward Powley, Daniel Whitehouse, Simon Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A Survey of Monte Carlo Tree Search Methods. *IEEE Transactions on Computational Intelligence and AI in Games*, Vol. 4, No. 1, pp. 1–43, 2012.
- [2] 熊谷香織, 持橋大地, 小林一郎, 麻生英樹, ムハンマドアッタミミ, 中村友昭, 長井隆行. モンテカルロ木探索を用いた構造的正しさと言語モデルを考慮した文生成. 第 30 回人工知能学会全国大会, 2016.
- [3] 黒橋禎夫, 長尾真. 京都大学テキストコーパス・プロジェクト. 言語処理学会第 3 回年次大会論文集, pp. 115–118, 1997.
- [4] 鈴木誠史, 白杵秀範, 島村徹也. 日本語早口言葉の構造と性質. 放送教育開発センター研究紀要, Vol. 12, pp. 131–149, 1995.