

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

Natural Language Generation using Monte Carlo Tree Search Considering Syntactic Correctness and N-gram Language Model

熊谷 香織 *1
Kaori Kumagai

持橋 大地 *2
Daichi Mochihashi

小林 一郎 *1
Ichiro Kobayashi

麻生 英樹 *3
Hideki Asoh

ムハンマド アッタミミ *4
Muhammad Attamimi

中村友昭 *4
Tomoaki Nakamura

長井隆行 *4
Takayuki Nagai

*1お茶の水女子大学
Ochanomizu University

*2統計数理研究所
The Institute of Statistical Mathematics

*3産業総合技術研究所
National Institute of Advanced Industrial Science And Technology

*4電気通信大学
The University of Electro-Communications

In the field of natural language processing, the representation of non-verbal information using verbal information has been studied in detail. In particular, the methods employing deep neural network framework have been actively studied. However, many of those studies employ the method of n-gram based sentence generation with an encoder-decoder framework, and it is still difficult to generate sentences with correct syntactic structure. With this background, we attempt to generate sentences with correct syntactic tree structure employing Context-free Grammar as syntactic rules and Monte Carlo tree search to select syntactic rules properly. The results of our sentence generation experiments demonstrate the feasibility of the proposed method.

1. はじめに

近年、視覚などの非言語情報から言語を生成する研究などが盛んに研究されている [Xu 15, Regneri 13, Yu 13]. それらの多くが n グラム言語モデル [Ushiku 11] やニューラル言語モデル [Xu 15] などの単語の系列を予測するモデルに基づくものが大勢を占めており、単語の依存関係など統語情報を考慮したものは数少ない。そのためそれらのモデルによって生成される文は人間にとって不自然なものとなる可能性がある。一方でテンプレートや単純な文法を用いた文生成手法 [Yu 13] も存在するが、状況に応じた柔軟な表現を伴う文生成はできない。

これらのことを踏まえて、本研究では、生成文における統語構造と語彙の選択の双方が人間にとって自然であり、状況に応じて柔軟な表現を伴った文を生成する手法を、モンテカルロ木探索を用いたシミュレーションの枠組みを用いることにより提案する。

2. モンテカルロ木探索を用いた文生成

2.1 モンテカルロ木探索 (MCTS)

MCTS (Monte Carlo Tree Search) は、コンピュータ囲碁におけるゲーム AI の手法として注目されたが、ゲーム固有の知識を必要としないため様々なテーマに応用できる。ゲームの途中の状況を評価関数などを元に評価するのではなく、ゲームが終わるまで何度もシミュレーションを行い、その勝敗と探索回数を元に評価する。つまり、途中の段階を評価することが難しいタスクに適している。

MCTS のもうひとつの特徴として、Multi-Armed Bandit 問題に対処する指標として、Auer ら [Auer 02] によって提案された UCB (Upper Confidence Bounds) 1 値がある。UCB1

値スロットマシンを選択する指標として、従来の報酬の平均の代わりに用いられている。UCB1 値は、勝率の項、および探索が不十分なノードに対して選択の可能性を考慮した項から構成される (式 1)。

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

v_i はそのノードの勝率、 C は調整係数、 N は全試行回数、 n はそのノードを選択した回数を示す。UCB1 値における第 1 項が「知識の適用 (exploitation)」を、第 2 項が「探査 (exploration)」を考慮している。それによりバランスをとった探索が実行される。

2.2 MCTS の処理手順

本研究では、文生成の手法として統語規則に文脈自由文法 (以下、CFG) を使用し、MCTS による適切な構造を持つ構文木の形成に基づき文を生成する。

以下に、処理の流れを示す。

- step0. (初期設定): ルートノードに文の開始記号 S が適用される。
- step1. (選択): ルートノードから適用可能な文法規則を UCB1 値に基づいて選択する。
- step2. (拡張): 新たなノードを生成する。
- step3. (シミュレーション): 生成されたノードから文法規則をランダムに適用し終端記号の文字列を生成する。
- step4. (逆伝搬): 生成された文の文としての評価スコア (第 3 章にて詳述) と、他候補ノードの平均スコアを比較し、勝敗を決め、迎ってきた全てのノードに返し、勝率を更新する。
- step5. (ルートノードの更新): step1 から step4 を規定回数繰り返した後、ルートノードの子ノードの内、探索回数が最大のノードを次のルートノードとして更新し、step1 へ戻る。

MCTS が文生成に適していると考えられる理由は 2 つある。1 つは、MCTS が最終的な状態に対する評価に基づく探索を行う点である。これは文生成において、正しい文になるかどうかを構文木形成の途中の段階で評価することは難しいことに対応する。もう一つは、シミュレーションに基づく生成となるた

連絡先: 熊谷香織, お茶の水女子大学大学院人間文化創成科学研究科理学専攻情報科学コース小林研究室, 〒112-8610 東京都文京区大塚 2-1-1, kumagai.kaori@is.ocha.ac.jp

め、様々な文法や単語を選択する機会が生まれ、柔軟な表現の伴った文の生成が期待できる。

3. 文の評価

2.3 節の step4 における「文としての評価スコア」をどのように決めるのかという問題が MCTS を用いた文生成において大きな課題となる。囲碁においてゲーム終了時点での勝敗は明らかであるが、文生成においては生成された文が正文か非文かは明確な二値の状態としては与えられない。人間が自然と思える正しい文かどうかを数値的スコアによって計算機が分かるように評価する必要がある。

本研究では、文の正誤の判断に 2 つの視点による評価値を導入した。1 つは構文としての構造の正しさに対する視点、もう一つは単語の繋がりやすさに対する視点である。以下、それぞれについて説明する。

3.1 構造の正しさに対する評価

構造の正しさの評価するため、構文木の部分木を素性としたロジスティック回帰による識別を行った。岡野原ら [Okanoohara 07] は、セミマルコフクラスモデルによって単語の系列を意味的なクラスに分割し、そのクラスの n グラムを素性とし、文を正文と非文とに識別している。

本研究では、部分木を素性として採用し、正文・非文を判別する識別器をロジスティック回帰により学習した。作成した識別器により、生成された構文木が適切な構造から成り立っているかを構文木を構成する部分木の組み合わせより判断し、文が統語的に正しいか誤っているかを識別する。

学習データはコーパス中の構文木を正例とし、コーパスから得られた CFG を用いてランダムシミュレーションにより得られた構文木を疑似負例とした。この際、部分木抽出手法として freqt^{*1} を使用した。また、素性とする部分木は、構造のみに着目するために終端記号を除いたものとした。

3.2 単語の繋がりやすさに対する評価

単語の繋がりやすさの評価するため、Lau [Lau 15] らによって文の容認度を測る指標として適当だとされたスコア (Acceptability) を用いた。言語モデルによって得られる対数尤度から、unigram 対数尤度を引き、その値を文長で除算した値である。以下にその式を示す (式 2)。

$$\text{Acceptability} = \frac{\log P_{\text{model}}(S) - \log P_{\text{uni}}(S)}{|S|} \quad (2)$$

今回は言語モデルとして、直前の単語の種類数を重視するスムージング方法である、Kneser-Ney スムージング [Kneser 95] による 3 グラムを使用した。また、Acceptability を以下、AP と呼ぶ。

4. 状況に応じた柔軟な文生成のための条件の与え方の考察

前章までは、文生成における「いかに自然な文を作るか」について述べたが、「いかに状況に応じて柔軟に単語を選択し、柔軟な表現を伴った文を生成するか」について考える。本研究では、表現したい「状況」の情報が外部から得られたと仮定し、その情報をどのように文生成過程に組み込むかを考える。

例として「dog」が「run」しているという状況を説明する。

「dog」の品詞は名詞である。今、名詞の生成規則を、 $NN \rightarrow \text{dog}$, $NNS \rightarrow \text{dogs}$ に制限する。ここで、人が言語で何かを説明しようとする時に、対象を説明する単語の選択は様々である。このことから、本研究において、入力として与えられると仮定した単語の意味に近い単語も文生成に選択される可能性があるとして、分散意味表現において、意味の近い単語も文生成に用いる。今回は学習した、word2vec によりスコアが 0.5 以上のものを似ているとして、「cat, puppy, cats, rabbits, foxes, hares, horses, animals, wolves」を名詞として語彙情報に追加した。また「run」の品詞は動詞であり、動詞の生成規則を、 $VB \rightarrow \text{run}$, $VBD \rightarrow \text{ran}$, $VBG \rightarrow \text{running}$, $VBP \rightarrow \text{run}$, $VCN \rightarrow \text{run}$, $VBZ \rightarrow \text{runs}$ に制限する。動詞の場合も名詞と同様に、分散意味表現により、run, ran, running, runs と似ているとされる動詞も生成規則に加えた。似ているとされた動詞は「passes」のみだった。

また、MCTS による文生成のシミュレーションにおいて、生成文中に名詞として「dog, dogs, cat, puppy, cats, rabbits, foxes, hares, horses, animals, wolves」が含まれていない場合、また動詞として「run, ran, running, runs, passes」が含まれていない場合はその時点で負けとした。

以上のような条件を与えた範囲の中で MCTS による探索を行うことで、状況に応じた使いたい単語を使いつつ適切な構造と語順から成る自然な文の生成を行なう。

また、使いたい単語の周辺の単語 (例えば名詞や動詞を修飾する形容詞や副詞など) は、対象となるドメインにおいて適切に補われるべきである。今回はそれらの情報は与えないため、表現したい状況を説明しつつも、多様な文が生成できることを想定している。

5. 自然な文の生成実験

5.1 実験設定

まず、状況に応じた条件を考えず、より自然な文を生成することのみを考えた実験を行った。

コーパスは、Brown コーパス^{*2} を使用し、その中で文長が 3~7 で、コンマなどの記号を含まない、4,661 文を CFG を構築するための対象コーパスとした。このようにコーパスを制限した理由として、生成内容として与える制約単語数から文生成に必要な文法は、短く単純な文に出現する簡単な文法のみで十分と考えたからである。コーパスを Stanford parser^{*3} で構文解析し、解析結果をもとに CFG を作成した。CFG のサイズは、文法数 7,220 個で終端記号は 5,867 個であった。

次に構造的正しさの評価するための識別器の学習データを作成した。正例として、コーパスの 4,661 文を Stanford parser で構文解析した構文木を使用した。負例は、岡野原ら [Okanoohara 07] の手法を参考に、コーパスより作成した CFG に基づいてランダムに生成した 46,610 個の構文木とした。負例数を正例の 10 倍とした理由としては、作成した CFG から生成され得る構文木の中で負例の方が圧倒的に多く存在することを考慮したためである。

また、MCTS のシミュレーション回数は、全勝ち数が、その時点での構文木から考えられる候補ノード数の 5 倍になるまでとした。このようにシミュレーション回数を動的な値に決めた理由は、シミュレーション回数を 10000 回などに統一 [Kumagai 16] してしまうと、勝敗の差が十分に付かないまま次のノードを選ばなければならないことがあった。また逆に、早い段階で勝敗の差が十分に付く場合もあり、無駄な探索をし

*1 <http://chasen.org/taku/software/freqt/>

*2 <http://clu.uni.no/icame/browneks.html>

*3 <http://nlp.stanford.edu/software/lex-parser.shtml>

てしまうこともあった。このことから、正確に、かつ無駄な探索を省くために、このようにシミュレーション回数を設定した。

5.2 統語構造に対する識別器の精度評価

作成したロジスティック回帰による識別器が正文と非文(疑似負例)を正確に識別できているかを確認するため10分割交差検定を行った。その結果98%の精度を確認した。ここで作成した識別器のことを、以下「構造評価識別器」と称し、それにより得られる確率をSP(Syntactic Probability)とする。

5.3 構造的正しさの評価

まずは構造的な正しさのみに対し評価をする文生成実験を行った。2.1節のstep4における文としての評価スコアに、構造評価識別器により得られる確率を用いた。MCTSにより得られた生成文例を表1に示す。

表1: 構造的正しさの評価をした場合の生成文例

生成文	SP
all mass nudged no teacher	0.999
this principle observed all super-condamine	0.999
all kay sank all round	0.999
some camping departs	0.994
those rim made these amount	0.999

上記の生成文を見るとSVOやSVの構造をした構造的に正しい文が生成できていることが分かる。スコアをみるとおよそ0.99であり、構造評価識別器により明らかに正例と判断される構文木が生成できていることも分かった。

一方で、単語の繋がりについては何も評価していないため、「some camping」「those rim made」など、不自然な単語の繋がりが見られた。

5.4 単語の繋がりに対する評価の追加

構造の評価に加え、単語の繋がりについての評価をした文生成実験を行った。

2つの評価値を適切に探索に影響させるためにUCB値を工夫した。2.3節のstep4において、SPのみについての勝敗(SP-Resultとする)とAPのみについての勝敗(AP-Resultとする)を求める。全体の勝敗(All-Result)を表2の通り決める。

表2: 2つの評価値の元での勝敗(0または1)の決め方

SP-Result	AP-Result	All-Result
0	0	0
0	1	0
1	0	0.5
1	1	1

単語の繋がりスコアが悪くても構造的に正しければある程度評価できる構文木だと判断し、SPのみ勝ちのときは0.5という値を返す。

また、様々な文長を生成できることを確認するため、文長の制約を導入した。設定した文長よりも短い文が生成された時は、その時点で負けとした。

また、Acceptabilityの効果を確認するため、APの代わりにKneyser-Neyスムージングによる3グラムのパープレキシティの値(以下、PP)を単語の繋がり goodness を評価する値として実験を行った。ただしパープレキシティは小さいほど良いので、他候補ノードの平均値と比較して小さい時に勝ちとした。この時の生成文は表3の通りである。

実験結果より、単語の繋がり評価を追加したときも構造的に正しい文が生成できた。Acceptabilityによる評価をすることで、頻度が低いと考えられる単語が選択されているこ

表3: nグラムによる評価を追加した時の生成文例

文長	生成文	SP	AP
5	those memorial neglected neither contraction-extension	0.999	85.46
6	all marketing half-straightened neither contraction-extension understandingly	0.999	91.79

文長	生成文	SP	PP
5	no theirs defied no improvement	0.999	1008.63
6	no one said his own work	0.999	156.85

とが確認でき、単語の頻度に影響されない生成ができた。一方で、パープレキシティを用いたスコアによる評価をすると、頻度が高いと思われる「no」などの単語を複数回選択しており、APを採用したときよりも単語の頻度に影響されてしまったことがわかった。

6. 状況に応じた文の生成実験

6.1 実験設定

状況に応じた使いたい単語が得られたと仮定して文を生成することを考える。「dog」が「run」している状況を説明する課題を例として実験設定を説明する。

文生成に使用した単語資源とコーパスの概要を図1に示す。

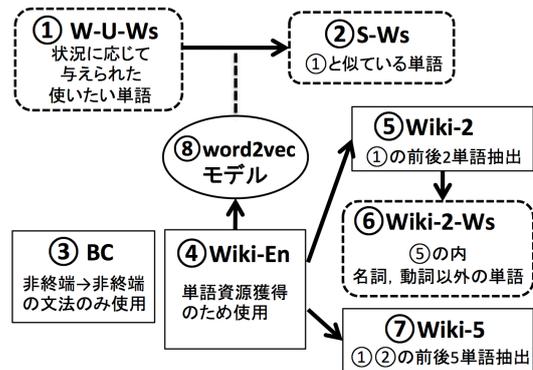


図1: 単語とコーパス

「dog, dogs」「run, ran, running, runs」を Want to Use Words(以下、①W-U-Ws)とし、分散意味表現により、それらの単語と似ているとされた単語「cat, puppy, cats, rabbits, foxes, hares, horses, animals, wolves」「passes」を Similar Words(以下、②S-Ws)とする。

5.1節で使用したBrown Corpus(以下、③BC)のみでは、「dog」や「run」が出現している文が少なく、単語の資源が不十分であった。そのため、単語の資源を増やすのに、Wikipedia English Corpus^{*4}(以下、④Wiki-En)を使用した。また、④Wiki-Enから①W-U-Wsの前後2単語を抽出する(以下、⑤Wiki-2)。⑤Wiki-2に含まれる単語は①W-U-Wsの周辺に出現する単語である。⑤Wiki-2をStanford Log-linear Part-Of-Speech Tagger^{*5}にかけ、品詞付けを行い、事前に与えられる単語の品詞である、名詞と動詞以外の単語(以下、⑥Wiki-2-Ws)を抽出した。この時、品詞付けの際の誤りを考慮し、2回以上出現した単語のみをWiki-2-Wsに加え、生成される文中に現れる単語の候補として採用した。

また、単語nグラムモデルに基づく評価指標が上手く機能する様に、ある程度の単語の系列から得られる単語資源を辞書に持たせるため、④Wiki-Enから、①W-U-Wsと②S-Wsの前後5単語を抽出した(以下、⑦Wiki-5)。

*4 <http://dumps.wikimedia.org/enwiki/>

*5 <http://nlp.stanford.edu/software/tagger.html>

①W-U-Ws から②S-Ws を求めるための word2vec のモデル(⑧word2vec) は Gensim ライブラリ^{*6} の word2vec モデルを使用し, ④Wiki-En をコーパスとして学習した.

次に MCTS の探索範囲について述べる. 非終端記号 → 非終端記号の生成規則については, 5.1 節で使用した文法と同様のもの(③BC から作成)を使用した. 単語情報となる, 前終端記号 → 終端記号の生成規則に関しては①W-U-Ws, ②S-Ws, ⑥Wiki-2-Ws を用いた.

MCTS における制約について説明する. 文長の制約に関しては 5.4 節と同様である. シミュレーションで生成された文が①W-U-Ws もしくは②S-Ws を含まない場合負けとした. また, 語彙選択において, ⑦Wiki-5 の中で, ある単語の次に出現したことがある単語から選択するという制約を設けた. この制約を設けることは可能性の低い手を探索の候補から除外することを意味し, より正確な探索を進めることを目指した.

また, 文の Acceptability を計算するために用いる n グラム辞書は, ⑦Wiki-5 から作成した.

6.2 実験結果

生成文例は表 4 の通りである

表 4: 状況に応じた条件を与えた時の生成文例

内容	文長	生成文	SP	AP
dog	4	either dog runs his cat	0.999	39.95
run	5	every dog runs her cat	0.999	37.13
	5	every dog eats his bread	0.999	37.58
dog	5	another dog eats his bread	0.999	42.73
eat	6	neither dog eats its own bread	0.999	42.71
bread	6	all dog eats its original bread	0.999	41.33
	5	girls tennis played the rugby	0.998	59.65
boy	5	volleyball boys played both rugby	0.998	72.03
play	6	girls tennis was played senior football	0.996	71.96
basketball	6	girls tennis played played and los	0.996	66.55
内容	文長	生成文	SP	PP
dog	4	this cat is run	0.999	76.87
run	5	some dog runs his cat	0.999	350.72
	5	every dog eats his bread	0.999	310.92
dog	5	no dog eats its flour	0.999	383.59
eat	6	every dog eats its first flour	0.999	380.06
bread	6	every dog eats its original bread	0.999	358.97
	5	boys soccer played the tennis	0.999	317.28
boy	5	girls tennis played an football	0.999	448.10
play	6	le boy plays her own tennis	0.999	549.45
basketball	6	boys tennis was played to all the	0.996	114.92

表のように, SP の値はおよそ 0.99 に収束し, 構造的に正しい文が生成された. AP と PP を採用した時を比較すると, PP を採用した時より AP を採用した時の方が多様な単語を選択できている. 具体例としては「dog, eat, bread」を指定したとき, PP のときは形容詞として「every」ばかりを選択してしまっているが, AP のときは「another, neither, all」なども選択できている. ただし, 大きな違いは見られない. また, 今回の生成では「both rugby」や「all dog」など単語の繋がりが不自然なものが見られた. これは, 語彙数が多すぎることによって MCTS の探索範囲が大きくなり, 探索の正確さが不十分になってしまったためと考えられる. 生成の精度をあげるために, コーパス中に, ある程度出現した単語の繋がりのみ生成文に現れるようにするなどして, 候補の語彙を絞り込む必要がある.

7. おわりに

CFG を適用規則とする MCTS による正文となる構文木の探索を行うことによる文生成手法を提案した. 構文構造の正し

さと単語の繋がりの良さに対する 2 つの視点における評価を行いながら探索を行うことにより, より自然な文の生成を目指した. また, MCTS によるシミュレーションに基づく生成のため, 様々な文法や単語を選択する可能性を持たせることができ, 状況に応じた柔軟な表現の伴った文の生成を目指した.

実験より, 構造評価識別器により構造的に正しい文が生成できた. パープレキシティを使用した時より, Acceptability による評価をした時の方がコーパス中での単語の頻度に影響されず, 多様な単語を選択できた. 状況に応じた使いたい単語を指定した時, その状況に即した構造的に正しい文を生成することができ, 非文が生成されることはなかった.

現段階では文長が短いとでも単純な文を生成しているが, 文長を 10 などと指定した時に構造的な正しさの判断が難しくなる可能性がある. 大局的な視点で構文木を評価する識別器が必要になるだろう. また, 探索手法についても改善する必要がある. 現段階では, ルートノードが更新されるたびにこれまで探索してきた情報を解放してしまっているため, 無駄な探索が多い. また, 現在はシミュレーションごとに勝敗を返しているが, 強化学習の枠組みを取り入れ, 方策を考えることにより, より正確な探索が行えるようになるだろう.

参考文献

- [Xu 15] K.Xu, J.Ba, R.Kiros, K.Cho, A.Courville, R.Salakhutdinov, R.Zemel, Y.Bengio, Show, Attend and Tell: A Neural Image Caption Generation with Visual, arXiv:1502.03044 [cs.LG], 2015.
- [Regneri 13] M.Regneri, M.Rohrbach, D. Wetzell, S. Thater, B. Schiele, and M. Pinkal, Grounding Action Descriptions in Videos, Transactions of the Association for Computational Linguistics (TACL), 2013.
- [Yu 13] Haonan Yu and Jeffrey Mark Siskind, Grounded Language Learning from Video Described with Sentences, Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 53–63, Sofia, Bulgaria, August 4-9 2013.
- [Ushiku 11] Yoshitaka Ushiku, Tatsuya Harada, and Yasuo Kuniyoshi. A Understanding Images with Natural Sentences. the 19th Annual ACM International Conference on Multimedia (ACMMM 2011), pp.679-682, 2011.
- [Kneser 95] R. Kneser and H.Ney. Improved backing-off for m-gram language modeling. In Proceedings of ICASSP, Vol. 1, pp.181-184, 1995.
- [Auer 02] P.Auer, N.Cesa-Bianchi, and P.Fischer, Finite-time analysis of the multi-armed bandit problem, Machine Learning, 47:235-256, 2002.
- [Okanohara 07] D.Okanohara, and J.Tsujii, A discriminative language model with pseudo-negative samples, In Proceedings ACL, 73-80, 2007.
- [Kumagai 16] 熊谷香織, 持橋大地, 小林一郎, 麻生英樹, M.Attamimi, 中村友昭, 長井隆行, モンテカルロ木探索を用いた統語構造を考慮した文生成, 第 22 回言語処理学会年次大会, 2016.
- [Lau 15] J.H.Lau, A.Clark, and S.Lappin, Unsupervised Prediction of Acceptability Judgements, In Proceedings of the 53rd ACL, 2015.
- [MCTS] Monte Carlo Tree Search (MCTS) research hub, <http://mcts.ai/>

*6 <https://radimrehurek.com/gensim/index.html>