読みにくい日本語文に対する係り受け解析・語順整序・読点挿入の同時実行とその評価

荒木 駿介 [1]　大野 誠寛 [1]　松原 茂樹 [2]
[1] 東京電機大学 大学院未来科学研究科　[2] 名古屋大学 情報基盤センター
23fmi04@ms.dendai.ac.jp　ohno@mail.dendai.ac.jp
matsubara.shigeki.z8@f.mail.nagoya-u.ac.jp

概要

日本語において，語順や読点の使用法は比較的自由に書き手に任されるが，実際には選好が存在しているため，意味は伝わるものの読みにくい文が作成されることがある．本稿では，推敲支援のための要素技術として，Shift-Reduce アルゴリズムを拡張した，日本語文に対する係り受け解析・語順整序・読点挿入の同時実行手法を提案する．提案手法では，従来手法にビームサーチを組み込み，深層学習モデルとして RoBERTa を使用することにより，精度向上を試みる．また，逐次実行手法との比較を行い，同時実行することの有効性を検証する．

1　はじめに

日本語は語順が比較的自由であるが，実際には選好が存在しているため，文法的には間違っていないものの読みにくい語順を持った文が無意識に作成される．また読点についても同様に，その有無や位置によっては文が読みにくくなる．このような読みにくい文は，語順を整え，適切に読点を挿入し直せば読みやすくなる．

語順整序や読点挿入に関する研究は，推敲支援や文生成などに応用でき，いくつか行われている（例えば，[1, 2, 3, 4]）．その中でも係り受け解析・語順整序・読点挿入を同時実行する手法として，宮地ら [3] の Shift-Reduce アルゴリズムを拡張した手法がある．我々の先行研究 [4] では，宮地らの手法 [3] の操作選択において BERT[5] を用い，かつ語順入替アルゴリズムを変更した手法を提案し、大幅な精度の向上を達成している．しかし，まだ精度の改善の余地は残っている．また，いずれの研究 [3, 4] も，係り受け解析・語順整序・読点挿入の同時実行手法と逐次実行手法とを比較評価しておらず，逐次実行に対する同時実行の有効性は確認されていない．

そこで本稿では，従来手法 [4] に対して新たな改良を加え，係り受け解析・語順整序・読点挿入を高精度に同時実行することを試みる．具体的には，ビームサーチを導入するとともに，大規模言語モデルとして RoBERTa[6] を用いる．さらに，係り受け解析，語順整序，読点挿入の順での逐次実行手法との比較評価を実施し，これらの処理を同時実行することの有効性を検証する．

2　日本語の語順・読点・係り受け

吉田ら [7] は，係り受けと語順の相互関連性と，当時の係り受け解析器では読みにくい文に対して精度が低下することを考慮して，係り受け解析と語順整序を同時実行する手法を提案している．また，逐次実行手法との比較を行い、同時実行が語順整序において有効であることを示した．さらに，先行研究 [3, 4] では，係り受けと語順に加えて読点の関連性も考慮し，係り受け解析・語順整序・読点挿入を同時実行する手法を提案している．しかし，両研究 [3, 4] では、これら 3 タスクを逐次実行する手法との比較は行われていない．

一方，深層学習を用いた係り受け解析器の登場により，係り受け解析性能は向上しており，読みにくい語順の文に対する性能も改善している可能性がある．そのため，逐次実行手法よりも同時実行手法が有効であるとは言い切れない可能性があるが，その検証は行われていない．

そこで、本稿では，係り受け解析・語順整序・読点挿入を同時実行する手法を提案するとともに，提案手法のアルゴリズムに基づき，係り受け解析のみ，語順整序のみ，読点挿入のみを実行する手法をそれぞれ作成し，係り受け解析，語順整序，読点挿入の順で逐次実行する手法との比較を行う．

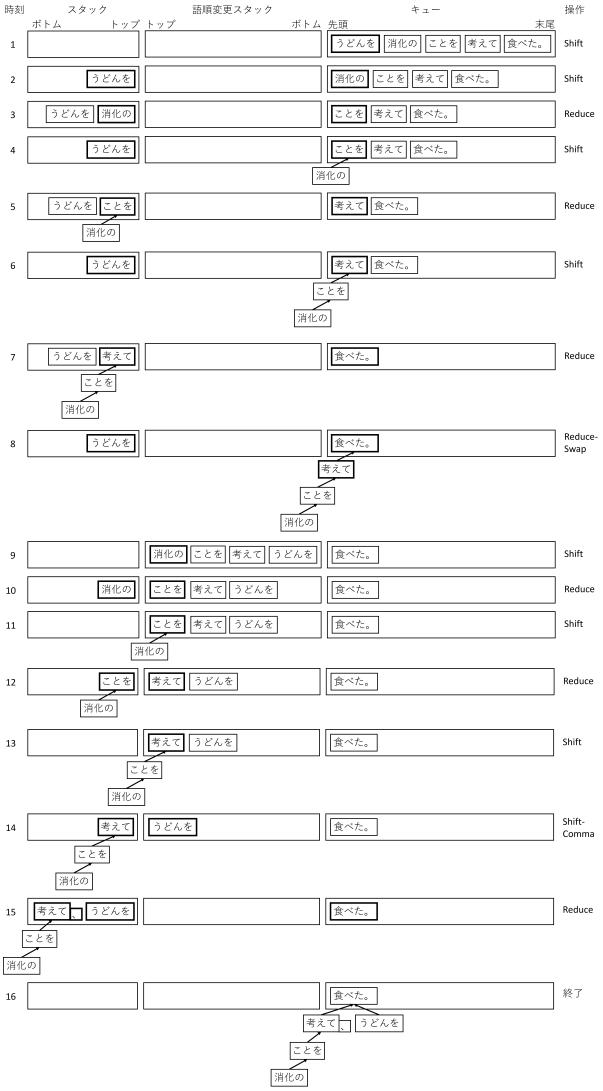

図 1　提案手法の動作例

3　提案手法

提案手法では，Shift-Reduce アルゴリズムを拡張し，係り受け解析・語順整序・読点挿入を同時実行する．なお，提案手法の入力は，意味は伝わるものの読みにくい語順を持った文の文節列を想定する．

3.1　提案手法のアルゴリズム

提案手法では，Shift-Reduce アルゴリズムにおいて，Shift と Reduce の他に，読点挿入のための Shift-Comma と Reduce-Comma，語順変更のための Reduce-Swap という 3 つの操作を追加するとともに，キューやスタックの他に，語順変更のためのスタック（以下，語順変更スタック）を新たに用意するという拡張を行い，係り受け解析・語順整序・読点挿入の同時実行を実現している．その概要は、語順変更スタックが空であればスタック Top の文節とキュー Front の文節を，語順変更スタックが空でなければスタック Top の文節と語順変更スタック Top の文節を，それぞれ操作対象として，これらの間に対する操作（Shift, Shift-Comma, Reduce,Reduce-Comma, Reduce-Swap のいずれか）を選択することを繰り返すというものである．以下では，操作対象の 2 文節のうち，スタック Top の文節を前方文節，もう一方の文節を後方文節と呼び，各操作を説明する．

Shift：前方文節が後方文節に係らないことを決定し，後方文節をスタックに移す．

Shift-Comma：Shift の操作に加えて，前方文節と後方文節の間に読点挿入することを決定する．

Reduce：前方文節が後方文節に係ることを決定し，前方文節をスタックから削除した上で，後方文節の子ノードとして追加し係り受け木を構成する．

Reduce-Comma：Reduce の操作に加えて，前方文節と後方文節の間に読点挿入することを決定する．

Reduce-Swap：前方文節が後方文節に係ることと，前方文節と「既に後方文節に係ると決定されている文節のうち，最も先頭に位置する文節（以下，Swap 候補文節）」の語順入替を決定し，前方文節，Swap 候補文節の順に語順変更スタックにプッシュする．なお，前方文節や Swap 候補文節に子ノードがある場合は，その語順を保つ形で，全ての子孫を語順変更スタックにプッシュする．また，語順変更に伴い読点位置を再度判定するため，語順変更スタックにプッシュされる文節に読点が付与されている場合は削除する．

提案手法の動作例を図 1 に示す．図 1 では，入力文「うどんを消化のことを考えて食べた。」に対して，係り受け解析と語順整序，読点挿入を同時的に施し，読みやすい文「消化のことを考えて、うどんを食べた。」に整形する過程が示されている（最適な操作選択ができるとして）．まず入力文の文節列[1)]がキューに格納され，時刻 1 で Shift により，キューの先頭「うどんを」がスタックにプッシュされる．次に時刻 2 で Shift により，「うどんを」が「ことを」に係らないことが決定され，「消化の」がスタックにプッシュされる．時刻 3 で Reduce により，「消化の」がスタックから削除され，「消化の」が「ことを」に係るとする係り受け木が構成される．時刻 8 で Reduce-Swap により，「うどんを」が「食べた。」に係ることと，「うどんを」と「考えて」の語順入替

1)　入力文にある読点はすべて事前に削除される．

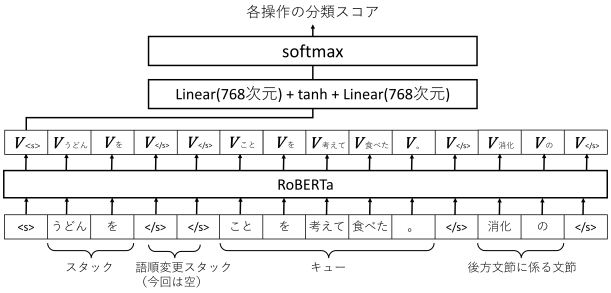

図 2　提案手法の操作選択モデル（図 1 の時刻 4）

が決定され，「うどんを」，「考えて」，「ことを」，「消化の」がこの順で語順変更スタックにプッシュされる．時刻 14 で Shift-Comma により，「考えて」と「うどんを」の間に読点が付与されるとともに，「うどんを」がスタックにプッシュされる．最後に時刻 15 で「考えて、」と「うどんを」が「選んだ。」の子ノードとして Reduce され，終了する．

3.2　RoBERTa を用いた操作選択モデル

図 1 では，説明のため，各時刻で最適な操作が選択されているが，実際には機械的に選択する必要がある．提案手法の操作選択モデル（図 1 の時刻 4 での計算例）を図 2 に示す．スタック・語順変更スタック・キューに格納された各文節列，および後方文節に直接係る文節列を結合し，先頭に <s> を，各々の後に </s> を付与したうえで，サブワード分割を施したものを各時刻での計算状況とみなし，RoBERTa に入力する．RoBERTa の出力のうち，<s> に対応する出力のみを取り出し，2 層の Linear 層と softmax を介し分類スコアを得て，分類スコアが高い操作をビームサーチで複数選択する．

本モデルを学習するには，各時刻での計算状況と，それに対する適切な操作の組が大量に必要となる．このような学習データを構築するには，読みにくい文と，それを読みやすく整形した文に対して係り受け構造を付与したデータが大量に必要となるが，そのようなデータはない．本研究では，係り受け構造が付与された読みやすいと想定される文を収録したコーパスを元に，疑似的な読みにくい文を機械的に生成し，それを提案アルゴリズムで元の読みやすい文に戻す過程を自動生成し，計算状況と適切な操作の組を大量に構築した．

4　評価実験

提案手法の有効性を確認するために，擬似的に作成した読みにくい文を用いた実験を行った．

4.1　実験概要

テストデータと開発データには，京大テキストコーパス [8] を元に人手を介して疑似的に作成された読みにくい語順の文 1,000 文（従来研究 [3, 4] と同一）をそれぞれ用いた．学習データには，京大テキストコーパス [8] の 32,506[2] 文に対して，3.1 節の手順を適用して得られる計算状況と適切な操作の組 1,467,980 件を用いた．学習データとテストデータの間で，元となった新聞記事文に重複はない．

比較手法として以下を用意した．

従来手法 [4]：提案手法と同様の手法であるが，ビームサーチを利用せず，大規模言語モデルとして BERT[5] を用いた手法．

逐次手法：提案手法の Shift-Reduce 拡張アルゴリズムを元に，係り受け解析・語順整序・読点挿入をこの順序で逐次的に実行する手法．

モデルは PyTorch を用いて実装し，RoBERTa の事前学習モデルには，早稲田大学が公開しているモデル[3] を用いた．学習アルゴリズムは Adam を用い，パラメータの更新はミニバッチ学習 (学習率 1e-6，バッチサイズ 16) により行った．損失関数には Cross Entropy Loss を使用した．学習は 10 エポック行い，開発データに対する語順整序の文単位正解率が最良となるモデルを 1 つ選択して評価に使用した．なお，本実験では，提案手法の同時実行モデル，逐次手法における係り受け解析モデル，語順整序モデル，読点挿入モデルを学習しており，それぞれ選択されたエポック数は，6，5，6，5 であった。

評価では，係り受け解析・語順整序・読点挿入の精度をそれぞれ測定した．係り受け解析では，係り受け正解率（文末文節以外の全文節のうち，正解と係り先が一致している文節の割合）と文単位正解率（正解の係り受け構造と完全一致している文の割合）を測定した．語順整序では，2 文節単位正解率（文末文節以外の文節を 2 つずつ取り上げたとき，それらの前後関係が正解と一致している割合）と文単位正解率（正解の語順と完全一致している文の割合）を測定した．読点挿入では，語順が正解と完全一致している文のみを対象に，読点位置に関する再現率，適合率，F 値を測定した．

2）　構文的制約 [9] から 1 通りの語順しか考えられない文や，構文的制約を満たさない文は事前に削除した．
3）　https://huggingface.co/nlp-waseda/roberta-large-japanese-seq512-with-auto-jumanpp

表 1　係り受け解析の正解率

手法	係り受け単位	文単位
従来	90.68%	59.00%
手法 [4]	(6,880/7,587)	(590/1,000)
逐次	93.86%	68.70%
手法	(7,121/7,587)	(687/1,000)
提案	93.41%	66.20%
手法	(7,087/7,587)	(662/1,000)

表 2　語順整序の正解率

手法	2 文節単位	文単位
従来	86.10%	52.40%
手法 [4]	(27,344/31,760)	(524/1,000)
逐次	89.94%	59.20%
手法	(28,564/31,760)	(592/1,000)
提案	90.17%	61.60%
手法	(28,638/31,760)	(616/1,000)

表 3　読点挿入の再現率・適合率・F 値

手法	再現率	適合率	F 値
従来	72.31%	76.69%	74.43
手法 [4]	(329/455)	(329/429)	
逐次	78.01%	85.63%	81.64
手法	(447/573)	(447/522)	
提案	72.56%	77.66%	75.02
手法	(431/594)	(431/555)	

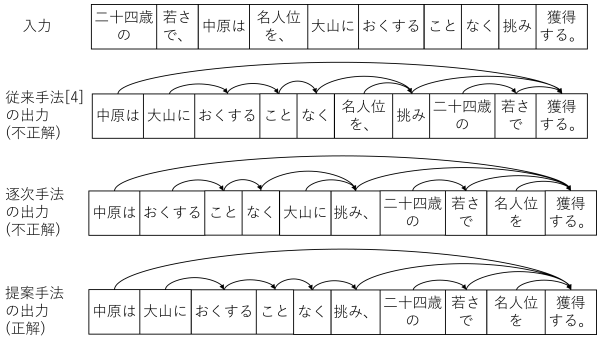

図 3　提案手法の成功例

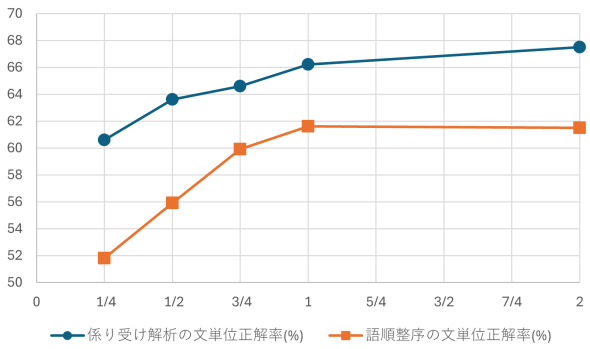

図 4　学習データの量と各文単位正解率の関係

4.2　実験結果

実験結果を表 1 から表 3 に示す．まず，提案手法はすべての指標で従来手法 [4] を上回っており，ビームサーチと RoBERTa を導入したことの有効性が確認された．次に，提案手法は逐次手法と比較して，係り受け解析の正解率で下回ったものの，語順整序の正解率で上回っており，語順整序に対する同時実行の有効性が確認された．

図 3 に従来手法 [4] および逐次手法では不正解だったが，提案手法では正解した例を示す．従来手法 [4] では，「名人位を」が「挑み」に係ると誤って解析しており，それに伴って，これらの文節に関わる語順整序や読点挿入に失敗している．また，逐次手法では，「大山に」が「挑み」に係ると誤って解析しており，「大山に」と，その正しい係り先「おくする」の語順整序に失敗している．一方，提案手法は，RoBERTa によって高精度な分類スコアが得られ，ビームサーチによって操作候補を複数保持できたことにより，正しく解析できたと考えられる．

4.3　考察

本節では，提案手法において，学習データの量が係り受け解析と語順整序の精度に与える影響を検証する．4.1 節では，元となった新聞記事文 1 文から疑似的な読みにくい文を 1 文作成し，それを元に計算状況と適切な操作の組 1,467,980 件を学習データとして作成している．これを基準（1 倍）として，1/4 倍 (元となった新聞記事文の文数を 1/4 倍にする)，1/2 倍，3/4 倍，2 倍 (元となった新聞記事文 1 文から，疑似的な読みにくい文を 2 文作成する）の 4 つの学習データをそれぞれ作成し，各学習データで学習したモデルを用いて，4.1 節と同様の実験を実施した．その結果を図 4 に示す．

係り受け解析の文単位正解率は，データ量が増加するにしたがって単調増加しているものの，増加の幅が鈍化する結果となった．また，語順整序の文単位正解率は，データ量が 1/4 倍から 1 倍の間では単調増加しているものの，1 倍から 2 倍の間で低下に転じる結果となった．これらの結果から，学習データの件数を今回の実験以上に増加させたとしても，精度の大幅な向上は見込めないと考えられる．

5　おわりに

本稿では，Shift-Reduce アルゴリズムを拡張し，日本語文に対する係り受け解析・語順整序・読点挿入を同時実行する手法を提案した．読みにくい文に対する評価実験を実施した結果，逐次実行と比べて語順整序の正解率が上回っており，本手法の有効性を確認した．今後は，操作選択モデルの精緻化などにより，更なる精度向上を図りたい．

謝辞

本研究は JSPS 科研費 JP19K12127, JP24K15076 の助成を受けたものです.

参考文献

[1] 内元清貴, 村田真樹, 馬青, 関根聡, 井佐原均. コーパスからの語順の学習. 自然言語処理, Vol. 7, No. 4, pp. 163–180, 2000.

[2] 大野誠寛, 吉田和史, 加藤芳秀, 松原茂樹. 係り受け解析との同時実行に基づく日本語文の語順整序. 電子情報通信学会論文誌, Vol. J99-D, No. 2, pp. 201–213, 2016.

[3] 宮地航太, 大野誠寛, 松原茂樹. 係り受け解析との同時実行に基づく日本語文の語順整序と読点挿入. 言語処理学会第 26 回年次大会発表論文集, pp. 243–246, 2020.

[4] 荒木駿介, 大野誠寛, 松原茂樹. 処理途中での非文生成の回避を考慮した日本語文に対する係り受け解析・語順整序・読点挿入の同時実行. 言語処理学会第 30 回年次大会発表論文集, pp. 3182–3186, 2024.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)**, pp. 4171–4186, 2019.

[6] Yinhan Liu, Myle Ott, Nman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. **arXiv:1907.11692**, 2019.

[7] 吉田和史, 大野誠寛, 加藤芳秀, 松原茂樹. 係り受け解析を伴った日本語文の語順整序. 言語処理学会第 20 回年次大会発表論文集, pp. 701–704, 2014.

[8] 黒橋禎夫, 長尾眞. 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 回年次大会発表論文集, pp. 115–118, 1997.

[9] 内元清貴, 関根聡, 井佐原均. 最大エントロピー法に基づくモデルを用いた日本語係り受け解析. 情報処理学会論文誌, Vol. 40, No. 9, pp. 3397–3407, 1999.