

Shift-Reduce 法に基づく 未入力トークン予測と漸進的係り受け解析の同時実行

橋本 優希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
23fmi24@ms.dendai.ac.jp ohno@mail.dendai.ac.jp
matsubara.shigeki.z8@f.mail.nagoya-u.ac.jp

概要

同時通訳や字幕生成などの音声言語システムに対して、音声入力の途中で随時、構文情報を提供することを目的に、文節が入力されるごとに係り受け構造を同定し出力するという漸進的係り受け解析手法が提案されている。

本稿では、さらに豊かな情報を後段のシステムに提供するため、未入力トークン予測と漸進的係り受け解析を同時実行する手法を提案する。講演文を用いた実験の結果、本手法の有効性を確認した。

1 はじめに

同時通訳 [1] や字幕生成 [2] などの音声言語システムでは、入力と同時的に処理することが求められる。このようなシステムにおいて構文的情報を利用するには、音声入力の途中で随時、構文構造を提供できる必要がある。

このような要請に答えるため、文節が入力されるごとに解析を実行し、係り先が入力されていない文節に対して、その係り先は未入力であることを明示した係り受け構造 (図 1) [3] や未入力文節との構文的関係を明示した係り受け構造 (図 2) [4, 5] を漸進的に解析する手法が提案されている。

本稿では、より豊かな構文情報を後段のシステムに提供するため、未入力文節との構文的関係とともに、その未入力文節の先頭トークンを明示した係り受け構造 (図 3) を漸進的に解析する手法を提案する。本手法では、Shift-Reduce 法を拡張することにより、未入力文節の先頭トークンを予測するタスクと、未入力文節との構文的関係を明示した係り受け構造を解析するタスクを同時実行する。両タスクの間には密接な関係があると内省でき、同時実行することにより両タスクの精度向上に好影響を与えるも

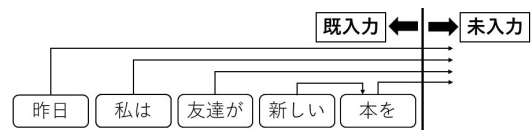


図 1 従来手法 [3] が出力する係り受け構造

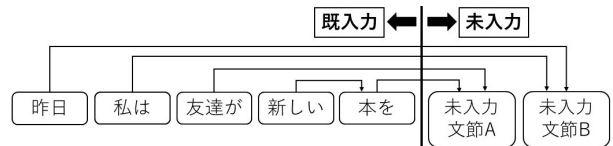


図 2 従来手法 [4, 5] が出力する係り受け構造

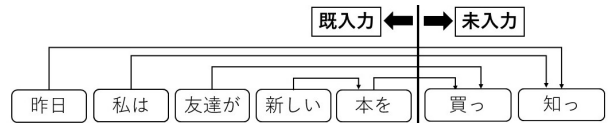


図 3 本手法が出力する係り受け構造

のと期待できる。

また、従来手法 [4, 5] では、従来手法 [3] の出力構造 (図 1) を入力として、未入力文節との構文的関係を明示した係り受け構造 (図 2) を同定しており、従来手法 [3] のエラー伝搬が問題となっていた。本手法では、与えられる既入力文節列から一挙に図 3 の係り受け構造を同定することができる。

2 漸進的係り受け解析の出力構造

従来手法 [3] は、文節が入力されるごとに解析を実行し、係り先が入力されていない文節に対して、その係り先は未入力であることを明示した係り受け構造を出力する。図 1 は、文「昨日私は友達が新しい本を買ったのを知った」の「本を」までが入力された段階で従来手法 [3] が出力する構造を示しており、「昨日」「私は」、「友達が」、「本を」の係り先が未だ入力されていないことを示している。

従来手法 [4, 5] では、従来手法 [3] による出力構造 (図 1) を入力として、係り先が未入力である文節が 2 つ以上存在したとき、それらの係り先が同一で

あるか否かを決定することにより、図2の係り受け構造を同定している。図2では、文節「友達が」と「本を」は同一の未入力文節に、文節「昨日」と「私は」は「友達が」や「本を」とは異なる文節に係ることを示している。

上記の従来研究に対して、本研究では、未入力文節との構文的関係とともに、その未入力文節の先頭トークンを明示した係り受け構造(図3)の同定を試みる。図3では、文節「友達が」と「本を」は同一の「買っ」から始まる未入力文節に、文節「昨日」と「私は」は「友達が」や「本を」とは異なる「知っ」から始まる未入力文節に係ることを示している。

3 未入力トークン予測と漸進的係り受け解析の同時実行

本手法では、 n 個の文節からなる文 $b_1 \dots b_n$ に対して、文節 $b_t (1 \leq t < n)$ が入力されるたびに、既入力文節列 $B_t = b_1 \dots b_t$ を拡張した Shift-Reduce 法に与え、未入力文節との構文的関係とともに、その未入力文節の先頭トークンを明示した係り受け構造(図3)を同定する。また、拡張した Shift-Reduce 法の各時刻では、BERT[6]を用いた未入力トークン予測と係り受け判定の結果に基づいた操作を行う。

3.1 拡張した Shift-Reduce アルゴリズム

本研究では、日本語係り受け解析を行う Shift-Reduce アルゴリズム [7] を拡張することにより、未入力トークン予測と漸進的係り受け解析の同時実行を実現する。具体的には、1つの Queue と2つの Stack を用意するとともに、5つの操作 (Shift, Shift_Predict, Reduce-n, Reduce-n_Predict, Reduce-Add) を用意し、その操作選択を繰り返す。

図4に、未入力トークン予測と漸進的係り受け解析を同時実行する様子を示す。本手法では、次の手順で既入力文節列を先頭から順に処理する。

1. 既入力文節列をその語順のまま Queue に格納し、Stack1 と Stack2 を共に空とする。
2. Stack1 の各文節が Queue Front の文節に係るか否かを後述する3.2節のBERTを用いて判定し、その結果に基づき、5つの操作 (Shift, Shift_Predict, Reduce-n, Reduce-n_Predict, Reduce-Add) のいずれかを選択・実行する。
3. 2を繰り返し、Stack1 と Queue が空の状態になれば終了する。

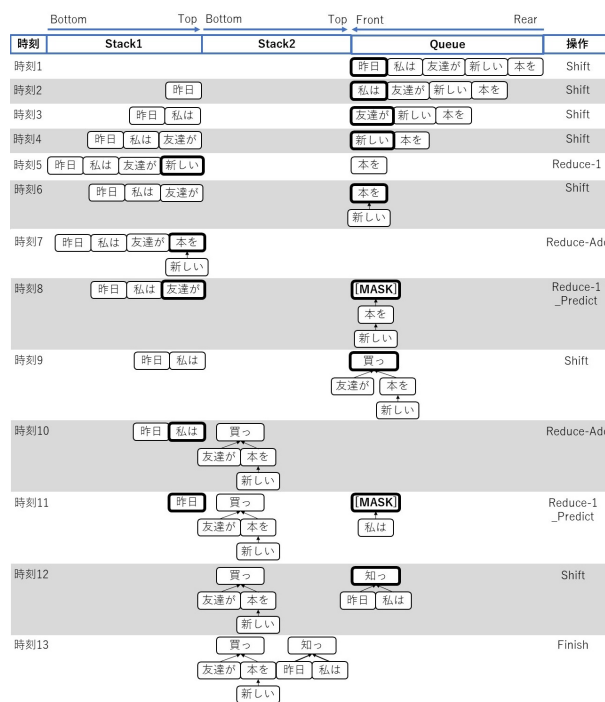


図4 本手法におけるアルゴリズム

なお、Queue や Stack に格納されるものは、各文節をノードとする部分係り受け木であるため、正確には、手順2の「Stack1 の各文節」や「Queue Front の文節」は、それぞれ「Stack1 の各部分係り受け木のルート文節」と「Queue Front の部分係り受け木のルート文節」と表記すべきであるが、以下では同様に略記する。

各操作の概要を以下に示す(詳細は付録を参照)。

- **Reduce-n:** Reduce を n 個同時に行う操作。
- **Reduce-Add:** Queue Front に未入力文節 [MASK] を追加して Reduce する操作。
- **Reduce-n_Predict:** 未入力トークン予測と Reduce-n を同時に行う操作。
- **Shift:** Queue Front の文節を Stack1 または Stack2 に移す操作。
- **Shift_Predict:** 未入力トークン予測と Shift を同時に行う操作。

具体的な動作例を図4に示す。なお、図4では各時刻で最適な操作が選択されるとし、操作対象を太枠で示している。時刻2では、Stack1 Topにある「昨日」が「私は」に係らないため、Shiftの操作が行われる。時刻5では、Stack1 Topにある「新しい」が「本を」に係るが、Topの次にある「友達が」は係らないため、係る個数は1個であり、Reduce-1の操作が行われる。時刻7では、Stack1が空でなく、Queue

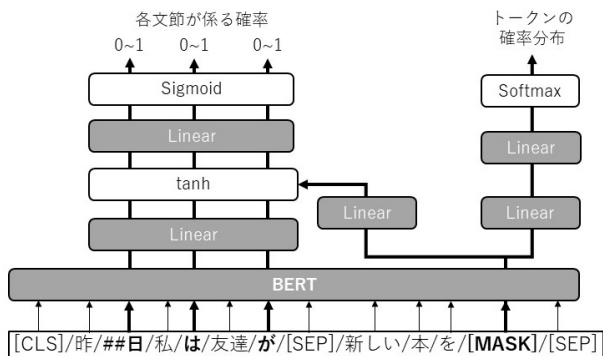


図5 本手法におけるBERTモデル(図4の時刻8)

が空であるので、Reduce-Addの操作が行われる。その結果、Queueに未入力文節[MASK]が追加され、Stack1 Topの「本を」がReduceされる。時刻8では、Stack1 Topにある「友達が」のみが未入力文節に係るため、Reduce-1.Predictの操作が行われる。この操作では、未入力文節[MASK]を予測し、その結果の「買っ」で[MASK]を埋めている。最後に時刻13でStack1とQueueが空となり、処理が終了する。

3.2 BERTを用いた未入力トークン予測と係り受け判定

本手法では、3.1節で述べた各時刻における操作選択を行うための係り受け判定と、Reduce-Addで追加される未入力トークン[MASK]の予測をBERT[6]を用いて同時実行する。

図5に、本手法のBERTモデルの概要(図4の時刻8における計算例)を示す。BERTへの入力、Stack1、Stack2、Queueの各々において、格納された全ての部分係り受け木を入力語順の文節列に直し、それらを[CLS]、Stack1の文節列、Stack2の文節列¹⁾、[SEP]、Queueの文節列、[SEP]の順に結合し、BERTに入力する。ただし、各文節列にはサブワード分割を施す。

未入力トークンの予測では、BERTの出力のうち、Queue Frontの[MASK]の出力のみを取り出し、2層のLinear層とSoftmaxを介してトークンの確率分布を得て、確率スコアが最も高いトークンを[MASK]の予測結果とする。

係り受け判定では、BERTの出力のうち、Stack1に格納されている各文節の末尾のトークンの出力をそれぞれ取り出す。それらの各出力と、Queue Frontの文節の先頭トークン(図5では[MASK]²⁾の出

1) ただし、予測された未入力トークンを除く
2) 例えば、図4の時刻5では「本」となる

力をLinear層を通してそれぞれ足し合わせ、Linear層1層とSigmoidを介して、各々の係り受け確率を得る。この確率が0.5以上であれば、Stack1の当該文節がQueue Frontの文節に係ると判定する。

学習時には、未入力トークン予測と係り受け判定を同時学習させる。学習時のパラメータ更新に用いるLossを式(1)に示す。

$$Loss = \lambda BCELoss + (1 - \lambda) CE Loss (0 \leq \lambda \leq 1) \quad (1)$$

ここで、BCELossは係り受け判定のLossであり、CELossは未入力トークン予測のLossである。

4 評価実験

本手法の有効性を確認するために、日本語講演データを用いて評価実験を行った。

4.1 実験概要

実験データとして、同時通訳データベース[8]に収録されている日本語講演音声の書き起こしデータ(形態素情報、文節境界情報、節境界情報、係り受け情報付)を使用した。学習には、上記のアルゴリズムで正解の係り受け構造を再現できる文(1,825文、21,553文節)を使用した。実験は全16講演を用いた交差検定により実施した。すなわち、1講演をテストデータ、残りの15講演を学習データとする実験を16回繰り返した。ただし評価では、従来研究[3,5]と同じ14講演(1,714文、20,707文節)を使用した。

比較のために、以下を用意した。

[λ = 1]: 本手法の式(1)においてλ = 1として、未入力トークン予測を行わず漸進的係り受け解析だけを行う手法。

従来手法[3,5]: 従来手法[3]の後に従来手法[5]を適用し、漸進的係り受け解析だけを行う手法。

人間: 後藤らの研究[9]と同様の方法で評価用データに対して作業員1名が未入力文節予測と漸進的係り受け解析を行い、図3の構造を推測した結果。

漸進的係り受け解析の評価では、従来研究[5]と同様に、係り先が既入力の文節と係り先が未入力の文節に分けて、係り先が正解と一致するかに関して、再現率と適合率をそれぞれ算出した。

未入力トークン予測の評価には、再現率、適合率を用いた。再現率は、正解の係り受け構造において係り先が未入力である文節のうち、正しくトークンを予測できた文節の割合を示す。適合率は、予測さ

表 1 係り受け解析精度 (%)

手法	係り先が既入力			係り先が未入力		
	再現率	適合率	F 値	再現率	適合率	F 値
本手法	83.11	82.12	82.62	73.49	74.35	74.02
[$\lambda = 1$]	83.08	82.05	82.56	73.61	74.39	73.99
従来手法	75.12	73.34	74.22	69.09	71.74	70.39
人間	87.77	88.28	88.02	79.74	77.29	78.49

表 2 未入力トークン予測精度 (%)

手法	再現率	適合率	F 値
本手法	3.41	3.44	3.42
[$\lambda = 1$]	0.00	0.00	0.00
人間	12.50	12.17	12.36

れた係り受け構造において係り先が未入力と判定された文節のうち、正しくトークンを予測できた文節の割合を示す。

モデルの実装は PyTorch³⁾ を用い、BERT の事前学習モデルは東北大学が公開しているモデル⁴⁾ を用いた。Linear 層の次元数はいずれも 768 とし、それぞれの入力を 0.1 の確率でドロップアウトさせた。学習アルゴリズムは AdamW を用い、パラメータの更新はミニバッチ学習 (学習率 $2e-6$, バッチサイズ 32) により行った。開発データを用いてハイパーパラメータの調整を行い、本手法ではエポック数 6, Loss の比率 $\lambda = 0.98$ とした。同様に、比較手法 [$\lambda = 1$] ではエポック 7 とした。

4.2 実験結果

表 1 と表 2 に実験結果を示す。まず、本手法と従来手法 [3, 5] の係り受け解析精度 (F 値) を比較すると、本手法が大幅に上回った。Shift-Reduce 法を拡張し BERT を使って、一度に漸進的係り受け解析を行う本手法の有効性を確認した。

次に、本手法と比較手法 [$\lambda = 1$] を比べると、係り受け解析精度 (F 値) において、本手法がわずかに上回っているものの、両者の解析性能は大きな違いはなかった。一方、本手法はわずかではあるものの、未入力トークンの予測に成功しており、比較手法 [$\lambda = 1$]⁵⁾ や従来手法 [3, 5] が未入力トークンを予測できないのとは比べ、より豊かな情報を提供できる可能性を確認した。

最後に、本手法と人間と比較すると、係り受け解析精度、未入力トークン予測精度のいずれも、本手

3) <https://pytorch.org/>

4) <https://github.com/cl-tohoku/bert-japanese>

5) 比較手法は偶然にわずかに予測が正解する可能性がある。

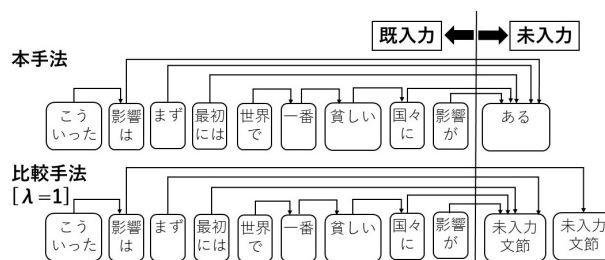


図 6 本手法の成功例

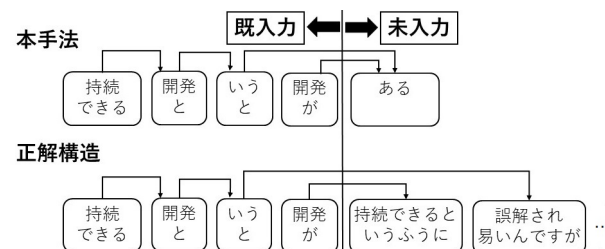


図 7 本手法の失敗例

法が大きく下回った。本手法には改善の余地が残されているといえる。

4.3 考察

図 6 に、本手法が成功し、比較手法 [$\lambda = 1$] が失敗した例を示す。本手法は、係り先のトークンを予測することができたため、この係り受け解析に成功したものと推察できる。

一方、図 7 に、本手法が失敗した例を示す。正解では「いうと」の係り先文節が「誤解」から始まり、「開発が」の係り先文節は「持続」から始まるが、本手法では、「開発が」の係り先文節の先頭トークンを「ある」と誤って予測し、さらにその「ある」に係ると誤って判定している。このように、未入力トークンの予測に失敗し、漸進的係り受け解析に失敗している例が多くみられた。

5 おわりに

本論文では、Shift-Reduce 法に基づく未入力トークン予測と漸進的係り受け解析の同時実行手法を提案した。本手法が従来手法よりも高精度に漸進的係り受け解析を実行でき、精度は低いものの、既入力文節と係り受け関係にある未入力トークンを予測できることを確認した。今後は、より強力な言語モデルの導入と、そのモデルへの入力を再検討し、精度向上を図る予定である。

謝辞

本研究は、一部、科学研究費補助金基盤研究 (C) NO.19K12127 により実施した。

参考文献

- [1] 笠浩一郎, 松原茂樹, 稲垣康善. 英日同時翻訳のための依存構造に基づく訳文生成手法. 電子情報通信学会論文誌, Vol. J92-D, No. 6, pp. 921–933, 2009.
- [2] 村田匡輝, 大野誠寛, 松原茂樹. 読みやすい字幕生成のための講演テキストへの改行挿入. 電子情報通信学会論文誌, Vol. J92-D, No. 9, pp. 1621–1631, 2009.
- [3] 大野誠寛, 松原茂樹. 文節間の依存・非依存を同定する漸進的係り受け解析. 電子情報通信学会論文誌, Vol. J92-D, No. 4, pp. 709–718, 2015.
- [4] 相津徹也, 大野誠寛, 松原茂樹. 漸進的係り受け解析における未入力文節との構文的関係の同定. 情報処理学会第 82 回全国大会講演論文集, No. 2, pp. 441–442, 2020.
- [5] 橋本優希, 大野誠寛, 松原茂樹. 漸進的係り受け解析における BERT を用いた未入力文節との構文的関係の同定. 情報処理学会第 85 回全国大会講演論文集, No. 2, pp. 803–804, 2023.
- [6] 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**, pp. 4171–4186, 2019.
- [7] 颯々野学. 日本語係り受け解析の線形時間アルゴリズム. 自然言語処理, Vol. 14, No. 1, pp. 3–18, 2007.
- [8] Shigeki Matsubara, Akira Takagi and Nobuo Kawaguchi, and Yasuyoshi Inagaki. Bilingual Spoken Monologue Corpus for Simultaneous Machine Interpretation Research. In **Proceedings of the 3rd International Conference on Language Resources and Evaluation**, pp. 153–159, 2002.
- [9] 後藤亮, 大野誠寛, 松原茂樹. 人間の漸進的言語処理能力の分析. 情報処理学会第 82 回全国大会講演論文集, No. 1, pp. 457–458, 2020.

A 付録

本アルゴリズムにおける各操作が選択される条件とその操作内容の詳細は以下の通りである。

- **Reduce-n:** Stack1 の Top から連続する n 個の文節が Queue Front の文節に係ると判定された場合に選択される。本操作は、Stack1 の Top から n 個の文節を削除した上で、それらを Queue Front の文節の子ノードとして追加した部分係り受け木を構成する。なお、3.2 節の BERT において Stack1 の各文節が Queue Front の文節に係るか否かを一挙に判定するため、Reduce を n 回同時に行う本操作を設けた。
- **Reduce-Add:** Stack1 が空でなく、Queue が空の場合に選択される。本操作は、Queue Front の未入力文節 [MASK] を追加するとともに、Stack1 Top の文節を削除した上で、それを当該 [MASK] の子ノードとして追加した部分係り受け木を構成する。
- **Reduce-n Predict:** Reduce-n の選択条件を満たし、かつ、Queue Front の文節が未入力文節 [MASK] の場合に選択される。本操作は、Queue Front の [MASK] を予測し、その結果で埋めると同時に、Stack1 Top から n 個の文節を削除した上で、それらを当該予測結果の子ノードとして追加し係り受け木を構成する。なお、3.2 節の BERT では、Queue Front が [MASK] の場合、Stack1 の各文節が Queue Front の文節に係るか否かと、[MASK] のトークン予測を同時に行う。
- **Shift:** Stack1 が空で、Queue が空でない場合、Stack1 Top の文節が Queue Front の文節に係らないと判定された場合に選択される。なお、直前の操作が Reduce-n, Reduce-n.Predict の場合は、必然的に Stack1 Top の文節が Queue Front の文節に係らないことになるため、自動的に本操作が選択される。本操作は、Queue Front の文節が既入力文節であれば Stack1 に、未入力文節 ([MASK] を予測結果で埋めたもの) であれば Stack2 に移す。なお、Stack2 には、トークン予測された未入力文節をルートに持つ部分係り受け木のみが格納される。
- **Shift Predict:** Shift が選択される条件で、Queue Front の文節が未入力文節 [MASK] の場合に選択される。本操作は、Queue Front の [MASK] を予測し、その結果で埋め、Stack2 に移す。