

# 日本語講演文に対する漸進的係り受け解析と文節主辞トークンの入力予測 Incremental Dependency Parsing and Input Prediction of Head Tokens for Japanese Lecture Sentences

橋本 優希<sup>†,a)</sup> 大野 誠寛<sup>†,b)</sup> 松原 茂樹<sup>‡</sup>  
Yuki Hashimoto Tomohiro Ohno Shigeki Matsubara

## 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] 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、その未入力文節の主辞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と、未入力文節の主辞トークンを予測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のと期待できる。

## 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」の係り先が未入力であることを示している。これにより、既入力文節内の「新しい本を」が構文的まとまりを構成することを後段のシステムは把握できる。

従来手法 [4] 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。これにより、図 1 よりも詳細な構文的まとまり (「友達が新しい本を」から「未入力文節 A」までが構文的にまとまっており、それが「昨日私は」から「未入力文節 B」までの構文的まとまりに埋め込まれていること) を後段のシステムは捉えることができる。従来手法 [4] では、従来手法 [3] の出力を入力としており、従来手法 [3] のエラー伝搬が問題となっていた。

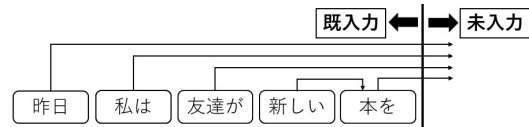


図 1 従来手法 [3] が出力する係り受け構造

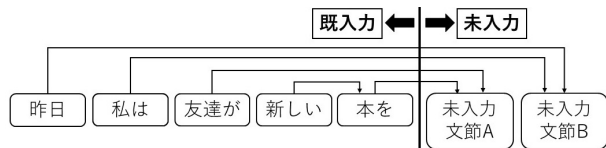


図 2 従来手法 [4] が出力する係り受け構造



図 3 本手法が出力する係り受け構造

上記の従来研究に対して、本研究では、与えられる既入力文節列から一挙に図 2 の構造を同定するだけでなく、未入力文節の主辞トークン (複数にサブワード分割される場合はその先頭トークン) を予測し、図 3 の構造を同定することを試みる。図 3 では、文節「友達が」と「本を」は同一の「買った」を主辞とする未入力文節に、文節「昨日」と「私は」は「友達が」や「本を」とは異なる「知っ」を主辞とする未入力文節に係ることを示している。これにより、各構文的まとまりを構成する未入力文節に関する予測情報を後段のシステムは活用できるようになる。

## 3 漸進的係り受け解析と

### 未入力文節主辞トークン予測の同時実行

本手法では、文節  $n$  個からなる文  $b_1 \dots b_n$  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、その未入力文節の主辞トークンを明示した構造 (図 3) を同定する。拡張した Shift-Reduce 法の各時刻では、BERT[5] による係り受け判定と未入力文節主辞トークン予測の結果に基づいて操作する。

### 3.1 拡張した Shift-Reduce アルゴリズム

本アルゴリズムでは、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、漸進的係り受け解析と未入力文節主辞トークン予測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 が空であること、あるいは、空でなければ

<sup>†</sup> 東京電機大学大学院未来科学研究科, Graduate School of Science and Technology for Future Life, Tokyo Denki University.

<sup>‡</sup> 名古屋大学情報基盤センター, Information Technology Center, Nagoya University.

a) 23fmi24@ms.dendai.ac.jp

b) ohno@mail.dendai.ac.jp

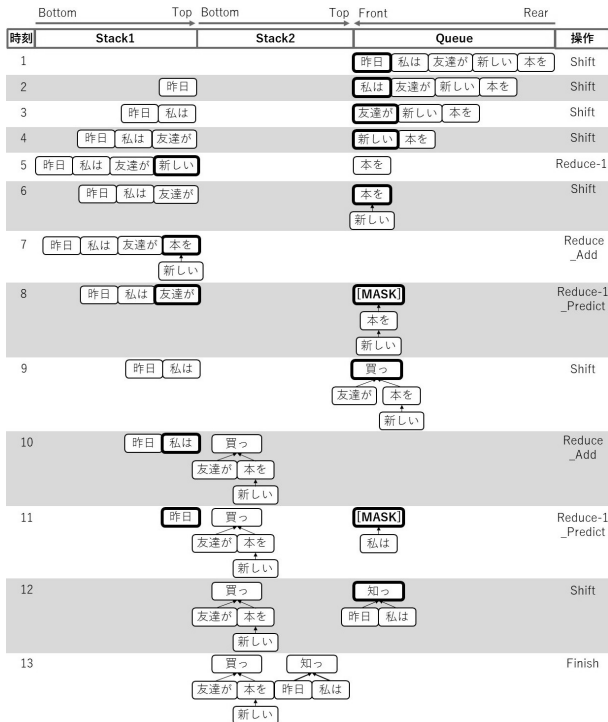


図 4 本手法におけるアルゴリズム

Stack1 の各文節が Queue Front の文節<sup>\*1</sup>に係るか否かを 3.2 節の BERT を用いて判定した結果に基づき、5 つの操作 (Shift, Shift\_Predict, Reduce-n, Reduce-n\_Predict, Reduce-Add) のいずれかを選択・実行し、Queue と 2 つの Stack を更新する。

3. 2. を繰り返し、Stack1 と Queue が空になれば終了。

各操作の概要を以下に示す。

**Reduce-n:** Stack1 の Top から連続する n 個の文節が Queue Front に係ると判定された場合に選択。Queue Front への Reduce を n 回繰り返す操作。

**Reduce-Add:** Stack1 が空でなく、Queue が空の場合に選択。Queue Front に予測対象の未入力文節主辞トークン [MASK] を追加し、それに対する Reduce を 1 回行う操作。

**Reduce-n\_Predict:** Reduce-n の選択条件を満たし、Queue Front が [MASK] の場合に選択。Queue Front の [MASK] を予測結果で埋め、それに対する Reduce-n を行う操作。

**Shift:** Stack1 が空で Queue が空でない場合、または、Stack1 Top が Queue Front に係らないと判定された場合に選択。Queue Front が既入力文節であれば Stack1 に、未入力文節 (の主辞トークン) の予測結果 ([MASK] を予測結果で埋めたもの) であれば Stack2 に移す操作。

**Shift\_Predict:** Shift の選択条件を満たし、QueueFront が [MASK] の場合に選択。Queue Front の [MASK] を予測結果で埋め、それを Stack2 に移す。

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、「昨日」が「私は」に係らないため、Shift が実行される。時刻 5 では、Stack1 の各文節のうち「新しい」のみが「本を」に係るため、Reduce-1 の操作が行われる。時刻 7 では、Stack1 が空でなく、Queue が空な

<sup>\*1</sup> なお正確には、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が、本論文では、そのルート文節を指して「文節」として略記する。

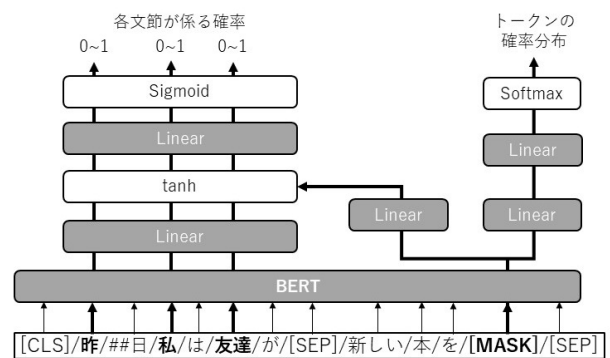


図 5 本手法における BERT モデル (図 4 の時刻 8)

ので、Reduce-Add が実行され、Queue への [MASK] の追加と、それに対する「本を」の Reduce が行われる。時刻 8 では、[MASK] 予測と係り受け判定の同時実行の結果、Stack1 の各文節のうち「友達が」のみが Queue Front の [MASK] の予測結果「買った」に係るため、Reduce-1\_Predict が実行され、[MASK] を予測結果「買った」で埋め、それに対する「友達が」の Reduce が行われる。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 を用いた係り受け判定と未入力文節主辞トークン予測

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文節主辞トークン [MASK] の予測を BERT[5] 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<sup>\*2</sup>、[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。

未入力文節主辞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、確率が最大のトークンを [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主辞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 の文節の主辞トークン (図 5 では [MASK]<sup>\*3</sup>) の出力を 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 以上となった Stack1 の各文節が Queue Front の文節に係ると判定する。

学習時には、係り受け判定と未入力文節主辞トークン予測を同時学習させる。学習時のパラメータ更新に用いる Loss を式 (1) に示す。

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

ここで、 $BCELoss$  は係り受け判定の Loss であり、 $CELoss$  は未入力文節主辞トークン予測の 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。

<sup>\*2</sup> 未入力文節主辞トークン予測が失敗した場合の影響を避けるため、[MASK] を埋めたトークン (予測されたトークン) は除いている。

<sup>\*3</sup> [MASK] となるとは限らず、例えば、図 4 時刻 5 では「本」となる。

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	82.70	81.72	82.21	73.58	74.26	73.92
従来手法	75.12	73.34	74.22	69.09	71.74	70.39
[ $\lambda = 1$ ]	82.59	81.62	82.10	73.39	74.03	73.70
人間	87.77	88.28	88.02	79.74	77.29	78.49

表 2 未入力文節主辞トークン予測精度 (%)

手法	再現率	適合率	F 値
本手法 (Top 1)	6.11	6.16	6.13
本手法 (Top 3)	14.72	14.85	14.78
本手法 (Top 5)	20.63	20.82	20.72
[ $\lambda = 1$ ]	0.00	0.00	0.00
人間	11.26	10.91	11.08

#### 4.1 実験概要

実験データとして、同時通訳データベース [7] 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。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, 4] と同じ 14 講演 (1,714 文, 20,707 文節) を使用し、残り 2 講演は開発データとした。また、各学習では、3.1 節のアルゴリズムで正解の係り受け構造を再現できる文のみを使用した。

比較のために、以下を用意した。

**従来手法:** 従来手法 [3] の後に従来手法 [4] を順次実行することにより、漸進的係り受け解析のみ行う手法。

**比較手法 [ $\lambda = 1$ ]:** 本手法の式 (1) において  $\lambda = 1$  として、未入力文節主辞トークン予測を強制的に学習させず漸進的係り受け解析のみ学習した手法。

**人間:** 後藤らの研究 [8] と同様の方法で評価用データに対して人間（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 [4] と同様に、係り先が既入力であるか未入力であるかに分けて、再現率と適合率をそれぞれ算出した。

未入力文節主辞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係り元文節のうち、係り先文節の主辞トークンを正しく予測できた係り元文節の割合を示す。適合率は、予測さ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係り元文節のうち、係り先文節の主辞トークンを正しく予測できた係り元文節の割合を示す。正しく予測できたか否かは、本手法の出力の Top K ( $K=1, 3, 5$ ) の中に正解が含まれているか否かで判定した。

モデルの実装は PyTorch<sup>\*4</sup> 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<sup>\*5</sup> 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  $5e-6$ 、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 8、Loss の比率  $\lambda = 0.91$  とした。同様に、比較手

\*4 <https://pytorch.org/>

\*5 <https://github.com/cl-tohoku/bert-japanese>

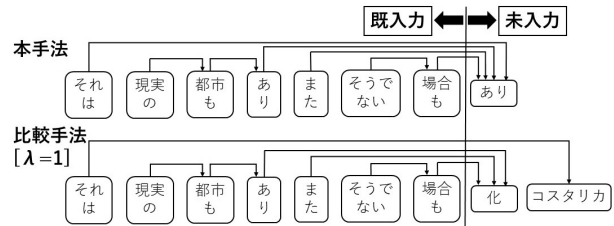


図 6 本手法の成功例

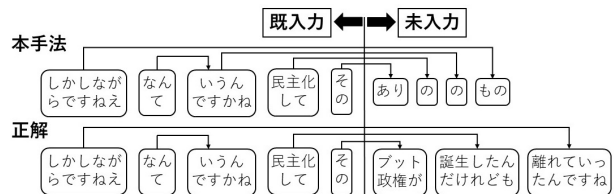


図 7 本手法の失敗例

法 [ $\lambda = 1$ ] ではエポック 8 とした。

#### 4.2 実験結果

表 1 と表 2 に実験結果を示す。本手法と従来手法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。従来手法が 2 つの手法 [3, 4] を逐次的に適用し、図 2 の構造を解析するのに対し、本手法は Shift-Reduce 法を拡張し BERT を使って、入力文節列から一度に図 2 の構造を解析しており、その有効性を確認できる。

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、未入力文節主辞トークンの予測では、比較手法 [ $\lambda = 1$ ] がほぼ予測できない<sup>\*6</sup> のと比べて、本手法は、Top 1 で 6% 程度、Top 5 で 20% 程度、予測できており、より豊かな情報を提供できる可能性を確認した。

最後に、本手法と人間を比較すると、係り受け解析精度、未入力文節主辞トークン予測精度の両方で本手法が大きく下回った。本手法には改善の余地があるといえる。

#### 4.3 本手法の成功例と失敗例

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、係り先の未入力文節主辞トークン「あり」を予測でき、係り受け解析にも成功している。

図 7 に、本手法の失敗例を示す。本手法は、係り先がない文節「いうんですかね」を未入力文節に係ると判定し、係り受け解析に失敗している。3.1 節のアルゴリズムは、係り受けにおいて、係り先が 1 つ必ず存在し、後方修飾することを前提としている。係り先がない文節や、前方修飾への対応は今後の課題である。

### 5 考察

#### 5.1 未入力文節主辞トークンの正確な予測が漸進的係り受け解析に与える効果の検証

本手法では、漸進的係り受け解析と未入力文節主辞トークン予測を同時実行することにより、漸進的係り受け解析の精度向上を試みた。しかし、未入力文節主辞トークン予測の精度が十分でなく、漸進的係り受け解析の精度向上が見られなかった。そのため本節では、未入力文節主辞トークンをすべて正確に予測できた場合に、

\*6 比較手法 [ $\lambda = 1$ ] は、未入力文節主辞トークン予測を学習していないが、その出力は行うため、偶然に正解することがわずかにある。

表 3 追加実験結果 (係り受け解析精度・%)

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
82.21	81.85	82.03	75.76	75.12	75.44

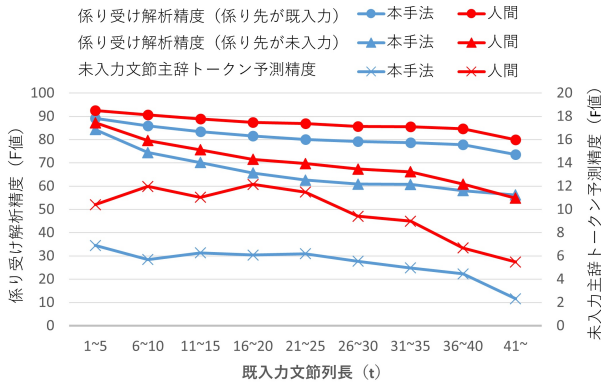


図 8 既入力文節列ごとの解析結果

漸進的係り受け解析の精度が向上するのかを検証するため、追加実験を行う。

追加実験では、拡張した Shift-Reduce アルゴリズムにおける操作の選択において Reduce\_Add の操作が選択されたときに、[MASK] ではなく、係り先の文節主辞トークンの正解トークンを Queue に追加することで、未入力文節主辞トークンの正確な予測を模倣した手法を用意し、4 節と同様の評価実験を行った。なお、この手法では、[MASK] を予測することなく、正解のトークンを使うため、トークン予測の学習をする必要がないことから、比較手法と同様に式 (1) で  $\lambda = 1$  として学習をした。

表 3 に追加実験結果 (係り受け解析精度) を示す。係り先が未入力文節である場合の係り受け解析精度において、追加実験結果が本手法の結果 (表 1) を上回った。未入力されていない係り先の主辞トークンを正確に予測できれば、漸進的係り受け解析の精度が向上することを確認した。一方、正解の未入力文節主辞トークンを用いても追加実験結果は、人間の精度 (表 1) には及んでおらず、その原因究明や本手法の改善は今後の課題である。

## 5.2 本手法と人間の比較

### 5.2.1 既入力文節列長に基づく比較

本手法と人間の漸進的係り受け解析精度と未入力文節主辞トークン予測精度について、既入力文節列 (3 節の  $B_i$ ) の長さに基づいた比較を行う。図 8 に、各々の F 値を既入力文節列の長さごとにプロットしたグラフを示す。

漸進的係り受け解析では、本手法と人間の両方とも、既入力文節列が長くなるほど F 値が低下する傾向にあった。既入力文節の数が増えるほど、利用可能な文脈情報が増えるが、その係り受け構造の複雑さも増すと言える。特に独話の講演文を対象としていることもあり、既入力文節列が長くなるほど、係り受け構造が複雑化すると言え、タスクの難化がより強く影響したと考えられる。

一方、未入力文節主辞トークン予測では、既入力文節列が長くなるほど本手法の F 値は低下していたが、人間の F 値は既入力文節列が短い時よりも長い時の方が高くなるがあった。人間が本手法に比べて、長い文脈をより高度に活用して予測できることが示唆された。

表 4 品詞別未入力文節主辞トークン予測精度 (再現率・%)

品詞	本手法		人間	
名詞	10.50	(2,325/22,141)	9.75	(2,158/22,141)
動詞	3.27	(828/25,344)	12.69	(3,217/25,344)
形容詞	0.54	(6/1,114)	10.95	(122/1,114)
副詞	0.21	(3/1,451)	11.99	(174/1,451)
形容動詞	0.09	(1/1,073)	6.90	(74/1,073)
連体詞	0.00	(0/69)	10.14	(7/69)
接続詞	0.00	(0/60)	0.00	(0/60)
感動詞	0.00	(0/26)	80.77	(21/26)
その他	0.00	(0/21)	4.76	(1/21)

### 5.2.2 品詞に基づく比較

本手法と人間の未入力文節主辞トークン予測精度について、予測対象 (正解データ上) の品詞に基づいた比較を行う。表 4 に、本手法と人間の未入力文節主辞トークン予測の再現率を品詞ごとに算出した値を示す。

本手法は、名詞が最も高い値 10.50%、動詞が次に高い値 3.27% となり、他の品詞の予測再現率は 1% に満たなかった。名詞と動詞は、他の品詞と比べて、主辞に選ばれやすく、学習データ量が多く存在するため、高い予測再現率を達成できたと考えられる。

一方、人間は、接続詞を除いて、どの品詞も予測再現率が 3% を上回っていた。特に、動詞、副詞、形容詞、連体詞、感動詞は 10% を上回る予測再現率を達成していた。人間は、その頻度に依存することなく、多彩な語を予測できていることが分かる。

## 6 おわりに

本論文では、漸進的係り受け解析と未入力文節主辞トークン予測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、科研費 JP19K12127, JP24K15076 の助成を受けたものです。

### 参考文献

- [1] 笠ら, “英日同時翻訳のための依存構造に基づく訳文生成手法,” 信学論, Vol. J92-D, No. 6, pp. 921–933, 2009.
- [2] 村田ら, “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,” 信学論, Vol. J92-D, No. 9, pp. 1621–1631, 2009.
- [3] 大野ら, “文節間の依存・非依存を同定する漸進的係り受け解析,” 信学論, Vol. J92-D, No. 4, pp. 709–718, 2015.
- [4] 橋本ら, “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.
- [5] J. Devlin et al., “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” Proc. ACL2018, pp. 4171–4186, 2019.
- [6] 颯々野, “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.
- [7] S. Matsubara et al., “Bilingual Spoken Monologue Corpus for Simultaneous Machine Interpretation Research,” Proc. LREC2002, pp. 153–159, 2002.
- [8] 後藤ら, “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.