

# Shift-Reduce 法に基づく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とその評価

橋本 優希<sup>1</sup> 大野 誠寛<sup>1</sup> 松原 茂樹<sup>2</sup>

<sup>1</sup> 東京電機大学 大学院未来科学研究科 <sup>2</sup> 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、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手法を提案する。また、提案手法と人間の漸進的係り受け解析結果を比較し考察する。

## 1 はじめに

同時通訳 [1] や字幕生成 [2], 音声対話システム [3] 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、各研究が定めた構文情報を出力する漸進的係り受け解析手法 [4, 5, 6, 7] が数多く提案されている。

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 1) 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。本手法では、Masked Language Model (MLM) を用いて、Shift-Reduce 法における各時刻での操作選択を行うが、本稿では、BERT[8], RoBERTa[9], DeBERTa[10] の各モデルを比較評価する。また、提案手法と人間の漸進的係り受け解析結果を比較し考察する。

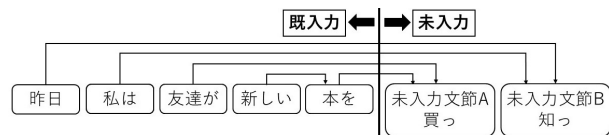


図 1: 本手法が出力する係り受け構造

## 2 漸進的係り受け解析の出力構造

漸進的係り受け解析では、文節が入力されるごとに解析を実行し、その時々どのような構造を出力するかが問題となる。これまでに、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 [4] や、未入力文節との構文的関係を明示した係り受け構造 [6] などが提案されている。

本研究では、さらに豊かな情報を後段のシステムに提供するため、与えられる既入力文節列から図 1 の係り受け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) を予測することを試みる。図 1 では、文節「友達が」と「本を」は同一の未入力文節 A に、文節「昨日」と「私は」は未入力文節 A とは異なる未入力文節 B に係るとともに、未入力文節 A が「買った」を、未入力文節 B が「知った」をそれぞれ主辞とする文節であることを示している。なお、本研究における文節の主辞の定義は、内元ら [11] による定義を一部変更したものとした。詳細は付録を参照されたい。

## 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、 $n$  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、その未入力文節の主辞トークンを明示した係り受け構造 (図

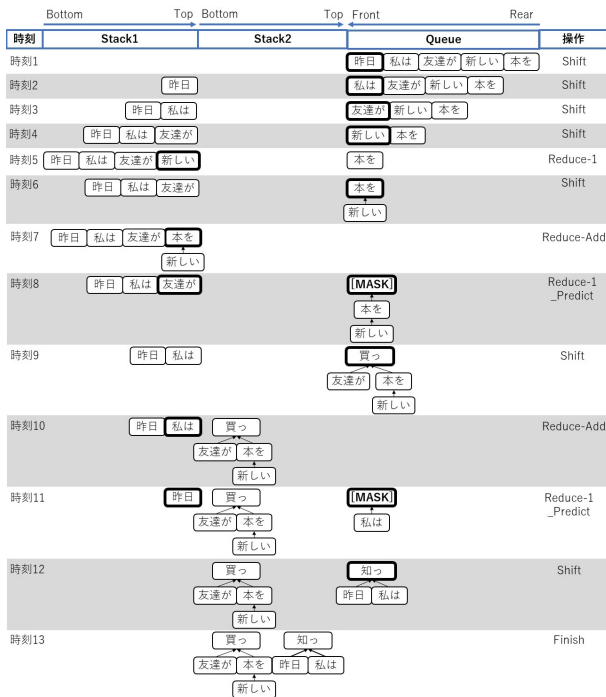


図 2: 本手法におけるアルゴリズム

1) を同定する。また、拡張した Shift-Reduce 法の各時刻では、係り受け判定と未入力トークン予測を同時実行する MLM 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 [12] を拡張することにより、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を実現する。その概要としては、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。詳細は付録を参照されたい。

具体的な動作例を図 2 に示す。なお、図 2 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、係り受け判定と [MASK] 予測の同時実行の結果、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

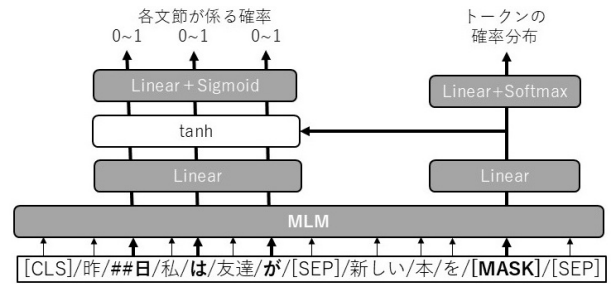


図 3: 本手法における MLM モデル (図 2 の時刻 8)

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 MLM 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 MLM を用いて同時実行する。図 3 に、本手法の MLM モデルの概要 (図 2 の時刻 8 における計算例) を示す。MLM への入力は、[CLS], Stack1 の文節列, Stack2 の文節列<sup>1)</sup>, [SEP], Queue の文節列, [SEP] の順に結合したものとする。

未入力文節主辞トークンの予測では、MLM 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、MLM の出力のうち、Stack1 内の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 3 では [MASK]) 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。

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

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。

1) 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。

## 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 [13] 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 講演のうち、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 回繰り返す交差検定により実施した。ただし学習には、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。また、評価では、従来研究 [4, 6] 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。

$[\lambda = 1]$ : 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を行わず漸進的係り受け解析だけを行う手法。

人間: 評価用データに対して作業者 1 名が漸進的係り受け解析と未入力文節予測を行い、図 1 の構造を推測した結果 [7]。

漸進的係り受け解析の評価では、従来研究 [6] 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、既入力文節と係り受け関係を持つ未入力文節のうち、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、予測された係り受け構造における未入力文節のうち、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 PyTorch<sup>2)</sup> を使い、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。MLM には、BERT[8]、RoBERTa[9]、DeBERTa[10] を用いて比較した。学習アルゴリズムは AdamW を使い、パラメータの更新はミニバッチ学習により行った。開発データを用いてハイパーパラメータの調整を行い、付録 A3 の表 6 のように設定した。評価では、乱数シードを変更し、各モデルをそれぞれ 5 つ用意し、上述した各評価指標の平均値を測定した。

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

表 1: 係り受け解析結果

手法	係り先が既入力			係り先が未入力		
	R	P	F	R	P	F
本手法 (BERT)	82.22	81.73	81.97	72.58	73.60	73.09
本手法 (RoBERTa)	84.25	83.64	83.94	73.74	74.99	74.36
本手法 (DeBERTa)	84.19	83.78	83.99	73.84	74.67	74.25
$[\lambda = 1]$ (BERT)	82.01	81.43	81.72	72.40	73.56	72.97
$[\lambda = 1]$ (RoBERTa)	83.63	84.10	83.86	74.53	73.61	74.06
$[\lambda = 1]$ (DeBERTa)	84.52	83.09	83.80	72.78	75.72	74.22
人間	87.77	88.27	88.02	78.30	77.30	77.80

表 2: 未入力トークン予測結果

手法	未入力トークン予測		
	R	P	F
本手法 (BERT)	4.01	4.40	4.20
本手法 (RoBERTa)	5.45	6.12	5.77
本手法 (DeBERTa)	4.11	4.53	4.31
$[\lambda = 1]$ (3 モデル共通)	0.00	0.00	0.00
人間	10.15	11.31	10.70

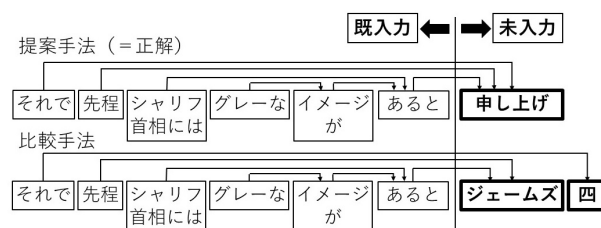


図 4: 本手法の成功例

### 4.2 実験結果

表 1 と表 2 に実験結果 (R: 再現率, P: 適合率, F: F 値) を示す。まず、本手法と比較手法  $[\lambda = 1]$  を比べると、係り受け解析精度において、同じモデル同士では、本手法が若干上回っており、未入力文節主辞トークン予測と同時実行する有効性が確認できる。また、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、より豊かな情報を提供できる可能性を確認した。図 4 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、未入力主辞トークン予測をできたため、この係り受け解析に成功したものと推察できる。

次に、本手法で採用した 3 つのモデル間で比較すると、係り受け解析と未入力トークン予測の両方で RoBERTa が最も精度が高かった。

最後に、人間と比較すると、係り受け解析、未入力トークン予測のいずれも、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。

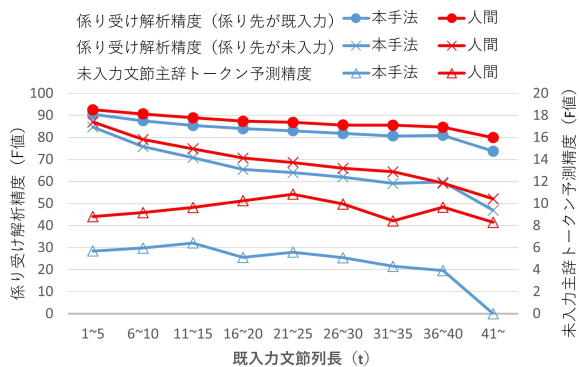


図 5: 入力文節列長ごとの解析結果

## 5 考察

本章では、本手法の MLM の 3 つのモデルのうち、最も性能が良かった本手法 (RoBERTa) (以下、本手法) の実験結果を用いて人間と比較し考察する。

### 5.1 既入力文節列長に基づく人間との比較

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  $B_t$ ) の入力文節長 ( $t$ ) に基づいた比較を行う。図 5 に、各々の F 値を入力文節列長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。既入力文節の数が多くなるほど、利用可能な文脈情報が増えるが、その係り受け構造が複雑化する。タスクの難化がより影響したといえる。

一方、未入力文節主辞トークン予測では、両者とも、入力長が一定の長さになるまで F 値は上がり続け、それよりも長くなると F 値は低下していった。ただし、本手法では  $11 \leq t \leq 15$  が予測の精度が最大になるのに対し、人間は  $21 \leq t \leq 25$  が最大になった。人間が本手法に比べて、より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。

### 5.2 品詞に基づく人間との比較

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 3 に、各々の再現率を品詞ごとに算出した値を示す。

本手法は、動詞の予測が 7.79% となり、人間の精度を上回った。一方、他の品詞の予測再現率は 1% に満たなかった。動詞は、他の品詞と比べて、学習データ量が大きく存在するため、高い予測再現率を

表 3: 品詞ごとの未入力トークン予測精度 (再現率)

品詞	本手法	人間
動詞	7.79 (1,638/21,025)	6.59 (1,385/21,025)
名詞	0.66 (60/9,139)	14.29 (1,306/9,139)
形容詞	0.31 (3/966)	6.63 (64/966)
副詞	0.00 (0/162)	3.09 (5/162)
連体詞	0.00 (0/113)	7.96 (9/113)
その他	0.00 (0/68)	16.18 (11/68)

表 4: 未入力トークン予測の正解有無ごとの係り受け精度

手法	トークン予測	係り先が既入力	係り先が未入力
	正解有無	F 値	F 値
本手法	有	85.04	75.97
	無	84.38	74.35
人間	有	88.24	81.26
	無	87.98	77.14

達成できたと考えられる。一方、人間は、どの品詞も予測再現率が 3% を上回っており、特に、名詞は 10% を上回る予測再現率を達成した。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。

### 5.3 未入力文節主辞トークン予測が漸進的係り受け解析に与える影響

本節では、未入力文節主辞トークン予測の正解が漸進的係り受け解析の精度に影響するかについて検証する。表 4 に、本手法と人間の各々において、未入力文節主辞トークン予測に成功したものが 1 つでもあったか否かで各入力文節列を分類し、それぞれの漸進的係り受け解析精度を算出した。

本手法と人間の両者とも、未入力トークン予測が正解しているものがある場合、係り受け解析精度の向上が見られた。特に、係り先が未入力の場合、本手法では 1.6 ポイント、人間では 4.1 ポイントも向上しており、漸進的係り受け解析と未入力文節主辞トークン予測の両タスクの関係性が確認された。

## 6 おわりに

本稿では、Shift-Reduce 法に基づく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。実験の結果、両タスクを同時実行する有効性や、より豊かな構文情報を提供できる可能性を確認した。今後は、MLM モデルの改良を再検討し、精度向上を図る予定である。

## 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] 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] 下岡和也, 徳久良子, 吉村貴克, 星野博之, 渡部生聖. 音声対話ロボットのための傾聴システムの開発. 自然言語処理, Vol. 24, No. 1, pp. 3–47, 2017.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] Hiroki Unno, Tomohiro Ohno, Koichiro Ito, and Shigeki Matsubara. Human Performance in Incremental Dependency Parsing: Dependency Structure Annotations and their Analyses. In **Proceedings of the 38th Pacific Asia Conference on Language, Information and Computation**. 10pages, 2024.
- [8] 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.
- [9] Yinhan Liu, Myle Ott, Naman 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.
- [10] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. DeBERTa: Decoding-enhanced bert with disentangled attention. In **Proceedings of the 9th International Conference on Learning Representations**, pp. 1–21, 2021.
- [11] 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.
- [12] 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.
- [13] Shigeki Matsubara, Akira Takagiand, 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.

## A 付録

### A.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。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 Stack1 の各文節が Queue Front の文節に係るか否かを 3.2 節の MLM 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を繰り返す、Stack1 と Queue が空になれば終了。

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、本稿では、そのルート文節を指して「文節」として略記している。

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、Queue

表 5: 使用した事前学習モデル

BERT	tohoku-nlp/ bert-base-japanese-char-whole-word-masking
RoBERTa	nlp-waseda/roberta-base-japanese
DeBERTa	ku-nlp/deberta-v2-base-japanese

表 6: ハイパーパラメータ

手法	$\lambda$	Learning_Rate	Epoch
本手法 (BERT)	0.96	5e-6	10
本手法 (RoBERTa)	0.96	5e-6	7
本手法 (DeBERTa)	0.87	5e-6	8
$[\lambda = 1]$ (BERT)	1	5e-6	9
$[\lambda = 1]$ (RoBERTa)	1	5e-6	13
$[\lambda = 1]$ (DeBERTa)	1	2e-6	9

Front 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。

### A.2 文節主辞トークンの決定方法

本手法において、文節の主辞の設定は以下のような手順で求めている。ここでの品詞分類は、ipadic の品詞体系に基づく。

1. 品詞が品詞大分類が特殊、助詞、助動詞、もしくは品詞細分類が名詞-接尾、動詞-接尾、形容詞-接尾 (以下、非主辞形態素) でない最も文末に近い形態素を主辞の候補にする。
2. 主辞の候補が、品詞細分類が名詞-非自立、動詞-非自立、形容詞-非自立、もしくは未知語 (以下、低優先形態素) でない場合、その形態素が主辞と確定する。
3. 主辞の候補が、低優先形態素の場合は、非主辞形態素や低優先形態素でない最も文末に近い形態素を探す。
4. 3 で非主辞形態素や低優先形態素ない形態素が存在したなら、3 で見つけた形態素を主辞とし、存在しないなら、1 で決まった主辞の候補を主辞と確定する。

### A.3 MLM の設定

本実験において使用した各 MLM の事前学習モデルは表 5 であり、各モデルのハイパーパラメータは表 6 の通り設定した。