

Transformers の BERT は共通テスト『国語』を 係り受け解析する夢を見るか

安岡孝一*

はじめに

筆者が班長を務める京都大学人文科学研究所共同研究班「古典中国語のコーパスの研究」(班員: Christian Wittern、守岡知彦、池田巧、山崎直樹、二階堂善弘、鈴木慎吾、師茂樹、白須裕之、藤田一乗)では、古典中国語(漢文)の文法解析に精力を傾注しており、その道具立ての一つとして、Universal Dependencies^[1]の古典中国語への適用^[2]を研究してきた。この過程において、われわれは、古典中国語係り受け解析ツール UD-Kanbun^[3]を製作した^[4]。また、書き下し文の解析にも対応すべく、近代日本語係り受け解析ツール UniDic2UD^[5]も製作した^[6]。

これら UD-Kanbun と UniDic2UD は、いずれも、形態素解析エンジンに MeCab 0.996^[7]を、係り受け解析エンジンに UDPipe 1.2.0^[8]を採用している。言語処理としては枯れた技術で、安定はしているものの、悪く言えば古臭い。せめて、係り受け解析エンジンの方だけでも、カレル大学で開発中の UDPipe 2 に入れ換えられないだろうか。そう筆者は期待していたのだが、どういうわけか UDPipe 2 は正式リリースに至らないまま、UDPipe 3 への開発移行がアナウンス^[9]されてしまった。UDPipe 2 の係り受け解析アルゴリズムは Biaffine^[10]で、UDPipe 1.2.0 の arc-planar^[11]に較べて精度が高くなるはずだったが、リリースに至らなかった以上、われわれとしては使いようがない。

*京都大学人文科学研究所附属東アジア人文情報学研究センター

^[1]Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, Daniel Zeman: Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection, Proceedings of the 12th Language Resources and Evaluation Conference (May 2020), pp.4034-4043.

^[2]Koichi Yasuoka: Universal Dependencies Treebank of the Four Books in Classical Chinese, DADH2019: 10th International Conference of Digital Archives and Digital Humanities (December 2019), pp.20-28.

^[3]<https://github.com/KoichiYasuoka/UD-Kanbun>

^[4]安岡孝一: 四書を学んだ MeCab + UDPipe はセンター試験の漢文を読めるのか, 東洋学へのコンピュータ利用, 第 30 回研究セミナー (2019 年 3 月 8 日), pp.3-110.

^[5]<https://github.com/KoichiYasuoka/UniDic2UD>

^[6]安岡孝一: 形態素解析部の付け替えによる近代日本語(旧字旧仮名)の係り受け解析, 情報処理学会研究報告, Vol.2020-CH-124 (2020 年 9 月), No.3, pp.1-8.

^[7]Taku Kudo, Kaoru Yamamoto, Yuji Matsumoto: Applying Conditional Random Fields to Japanese Morphological Analysis, Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (July 2004), pp.230-237.

^[8]Milan Straka and Jana Straková: Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe, Proceedings of the CoNLL 2017 Shared Task (August 2017), pp.88-99.

^[9]<http://ufal.mff.cuni.cz/udpipe>

^[10]Timothy Dozat, Christopher D. Manning: Deep Biaffine Attention for Neural Dependency Parsing, 5th International Conference on Learning Representations, Monday (April 24, 2017) Afternoon, C25.

^[11]Carlos Gómez-Rodríguez, Joakim Nivre: A Transition-Based Parser for 2-Planar Dependency Structures, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (July 2010), pp.1492-1501.

UDPipe に代わる係り受け解析エンジンを探していたところ、ワルシャワ工科大学の Scwad プロジェクト^[12]から、係り受け解析エンジン COMBO-pytorch 0.0.1^[13]が発表された。旧版の COMBO^[14]を、AllenNLP 1.2.0^[15]の Biaffine 実装で再構成したものであり、さらに、Transformers 3.4.0^[16]の BERT^[17]モデル群^[18]に接続可能となっていた。おりしも、北京理工大学から GuwenBERT^[19]という古典中国語 RoBERTa^[20]モデルが発表されており、Transformers にもアップロード^{[21][22]}されていた。よし、役者は揃った。

CLARIN-PL へと移管された COMBO-pytorch 1.0.1^[23]に対し、われわれは、AllenNLP 2.x および Transformers 4.x への拡張をおこなった上で、UD-Kanbun の形態素解析モジュールと接続し、GuwenCOMBO と名づけてリリース^[24]した。また、同様の拡張を UniDic2UD に対してもおこない、UniDic-COMBO と名づけてリリース^[25]した。その上で、大学入学共通テスト『国語』の令和3年度本試験(2021年1月16日実施)第4問および第1～3問の問題文を、手作業で Universal Dependencies 化し、GuwenCOMBO と UniDic-COMBO の評価をおこなった。以下に詳細を述べる。

^[12]<http://zil.ipipan.waw.pl/Scwad>

^[13]<https://github.com/ipipan/combo>

^[14]Piotr Rybak, Alina Wróblewska: Semi-Supervised Neural System for Tagging, Parsing and Lematization, Proceedings of CoNLL 2018 Shared Task (October 2018), pp.45-54.

^[15]<https://docs.allennlp.org/v1.2.0/>

^[16]<https://huggingface.co/transformers/v3.4.0/>

^[17]Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Proceedings of the 2019 Conference of North American Chapter of Association for Computational Linguistics: Human Language Technologies, Vol.1 (June 2019), pp.4171-4186.

^[18]<https://huggingface.co/models>

^[19]閻覃, 遲澤聞: 基于繼續訓練的古漢語語言模型, 第19届中国計算語言學大會“古聯杯”古籍文獻命名實體識別(2020年10月31日).

^[20]Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov: RoBERTa: A Robustly Optimized BERT Pretraining Approach, arXiv: 1907.11692 (July 26, 2019).

^[21]<https://huggingface.co/ethanyt/guwenbert-base>

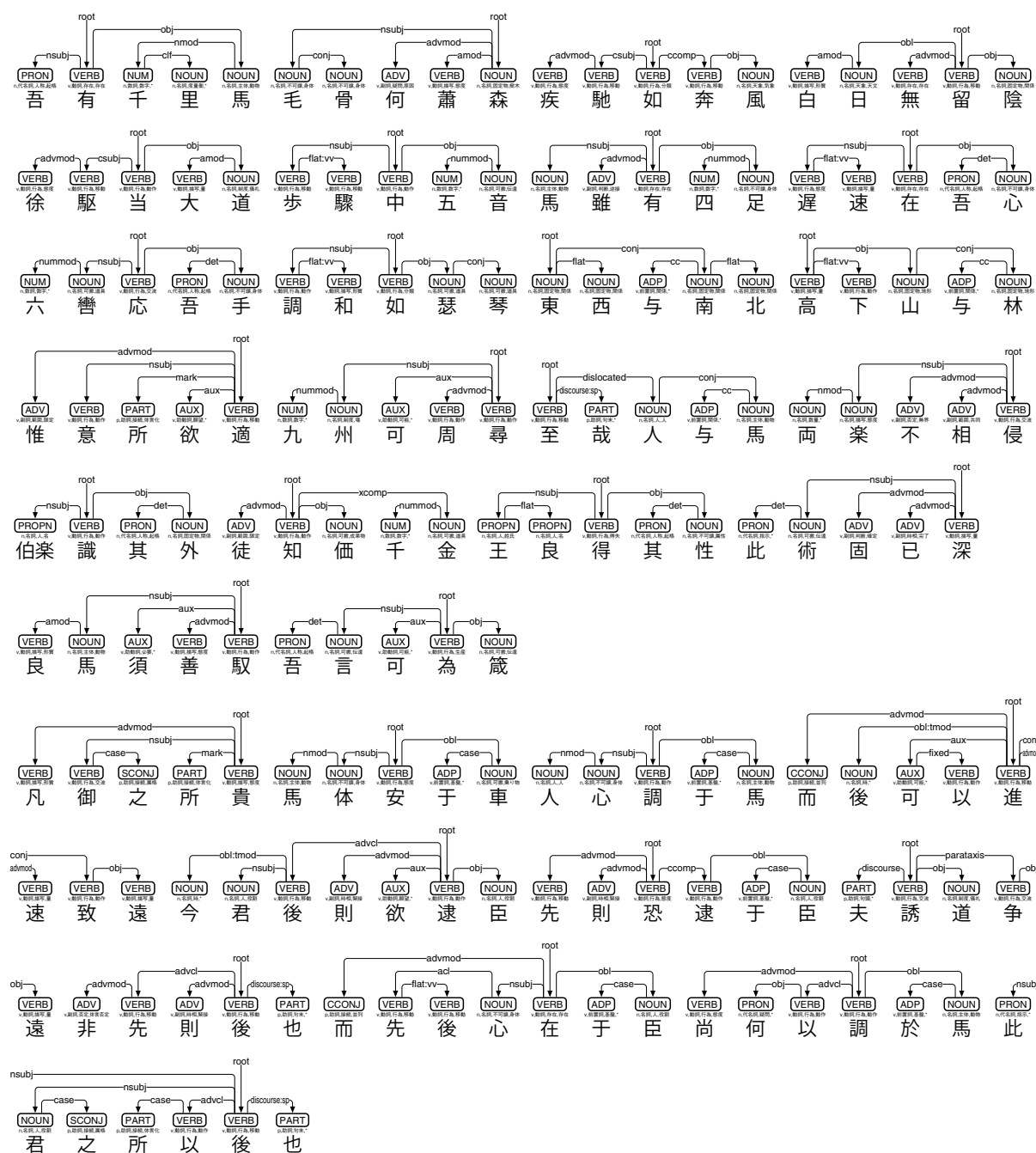
^[22]<https://huggingface.co/ethanyt/guwenbert-large>

^[23]<https://gitlab.clarin-pl.eu/syntactic-tools/combo>

^[24]<https://github.com/KoichiYasuoka/GuwenCOMBO>

^[25]<https://github.com/KoichiYasuoka/UniDic-COMBO>

共通テスト『国語』第4問「漢文」



GuwenCOMBO を製作するにあたり、われわれが作成中^[26]の『孟子』『論語』『禮記』『十八史略』(南宋を除く) Universal Dependencies データ 52018 文 250905 語を、係り受けの教師データとして用いた。また、BERT モデルに

- Ⓑ ethanyt/guwenbert-base
- Ⓕ ethanyt/guwenbert-large

のいずれかを用いた場合と、用いない場合とを、比較できるようにした。ただし、㊦㊧は「1文字=1単語」の簡化字モデルなので、UD-Kanbunの形態素解析モジュールに簡化

^[26]<https://corpus.kanji.zinbun.kyoto-u.ac.jp/gitlab/Kanbun/ud-kanbun>

字への変換処理^[27]を組み込み、BERT モデルの入口で簡化字変換をおこなった上、出口で戻している。訓練(係り受けモデル構築)に要した時間は、100epochs・1GPU (NVIDIA GeForce RTX 2080) で、各モデル 1～2 時間程度だった。なお、構築した係り受けモデルでは、UPOS と FEATS の解析も同時におこなっている。

このようにして製作した GuwenCOMBO 0.9.6 を、UD-Kanbun 2.7.4 および Stanza 1.2^[28] (lzh モデル) と共に評価した。共通テスト『国語』第4問は、【問題文Ⅰ】と【問題文Ⅱ】に分かれており、それぞれ別々に評価をおこなった。評価指標は、LAS (Labeled Attachment Score) / MLAS (Morphology-aware Labeled Attachment Score) / BLEX (Bi-LEXical dependency score) の3つの指標^[29]である。表1に結果を示す。

表 1: GuwenCOMBO を第4問で評価 (LAS / MLAS / BLEX)

		【問題文Ⅰ】	【問題文Ⅱ】
GuwenCOMBO	㊸	71.56 / 68.09 / 68.09	83.33 / 78.18 / 81.82
	㊹	77.98 / 74.87 / 74.87	83.33 / 76.36 / 80.00
	無	60.55 / 57.59 / 56.54	83.33 / 76.36 / 80.00
UD-Kanbun		72.48 / 66.67 / 68.78	78.79 / 73.39 / 77.06
Stanza		57.41 / 49.47 / 54.74	65.15 / 56.36 / 60.00

【問題文Ⅰ】においても【問題文Ⅱ】においても、BERT を用いた GuwenCOMBO の精度が高い。特に【問題文Ⅰ】の詩(欧陽脩『欧陽文忠公集』による)では、BERT 無しに較べ、㊹の結果がグンと改善されている。これは、GuwenBERT が17億字の漢文から作られており、その中に、この詩が含まれている(もちろん【問題文Ⅱ】『韓非子』も含まれている)からではないか、というのが筆者の推測である。

GuwenCOMBO の係り受け解析機能を、UD-Kundoku 1.7.4^[30]の自動訓読機能と接続し、【問題文Ⅰ】と【問題文Ⅱ】の自動訓読に挑戦してみた。「1文字=1単語」とみなして BLEU^[31]および RIBES^[32]で評価した結果を、表2に示す。なお、BLEU は、NLTK 3.5^[33]の method3 でスムージング (NIST geometric sequence smoothing) をおこなっている。

結果として、GuwenCOMBO の㊹が比較的高い評価値を示しており、自動訓読においても BERT モデルの効果が認められる。ただし、【問題文Ⅰ】の BLEU に限っては、少

^[27]たとえば「乾肝」は「干」「肝」に変換するが、「乾侯」は変換せず「乾」「侯」とする。

^[28]Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, Christopher D. Manning: Stanza: A Python Natural Language Processing Toolkit for Many Human Languages, The 58th Annual Meeting of the Association for Computational Linguistics: Proceedings of the System Demonstration (July 2020), pp.101-108.

^[29]Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov: CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, Proceedings of the CoNLL 2018 Shared Task (October 2018), pp.1-21.

^[30]安岡孝一: 漢文の依存文法解析にもとづく自動訓読システム, 日本漢字学会第3回研究大会予稿集 (2020年11月), pp.60-73.

^[31]Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu: BLEU: a Method for Automatic Evaluation of Machine Translation, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (July 2002), pp.311-318.

^[32]平尾努, 磯崎秀樹, 須藤克仁, Duh Kevin, 塚田元, 永田昌明: 語順の相関に基づく機械翻訳の自動評価法, 自然言語処理, 第21巻, 第3号 (2014年6月), pp.421-444.

^[33]<https://www.nltk.org/api/nltk.translate.html>

表 2: 第 4 問に対する自動訓読の評価 (BLEU / RIBES)

	【問題文 I】	【問題文 II】
GuwenCOMBO ③	0.353714 / 0.830090	0.403931 / 0.901300
GuwenCOMBO ④	0.348924 / 0.846239	0.442873 / 0.909021
無	0.348709 / 0.807612	0.427628 / 0.905584
UD-Kundoku	0.367928 / 0.839227	0.406699 / 0.898083

しばらく UD-Kundoku が高い。そこで、【問題文 I】の各句に対する自動訓読結果を、GuwenCOMBO④と UD-Kundoku で比較してみることにした。表 3 に結果を示す。どうやら、GuwenCOMBO④と UD-Kundoku とで、それぞれに得手不得手があるらしく、たとえば「調和如瑟琴」は UD-Kundoku の方がいいが、「至哉人与馬」は GuwenCOMBO④の方がいい。これらの「いいとこどり」をおこなうアルゴリズムを開発したいところだが、なかなか難しいそうである。

表 3: 【問題文 I】 に対する自動訓読の比較 (BLEU / RIBES)

GuwenCOMBO ①		UD-Kundoku		書き下し文
吾は千里馬有り	0.278900 / 0.948548	吾は千里馬有り	0.278900 / 0.948548	吾に千里の馬有り
毛の骨は何ぞ森を蕭なり	0.098647 / 0.802096	毛の骨は何ぞ蕭なりて森	0.165904 / 0.859389	毛骨何ぞ蕭森たる
疾びて馳けるは風に奔るが如し	0.042465 / 0.695750	疾馳は風に奔るが如し	0.056775 / 0.749276	疾く馳すれば奔風の如く
白日は陰に留めること無し	0.113908 / 0.871331	白日は陰に留めること無し	0.113908 / 0.871331	白日に陰を留むる無し
徐して駆るは大道を当る	0.086061 / 0.813666	徐して大道に駆当す	0.167359 / 0.699183	徐に駆くれば大道に当たり
驟中に歩いて五音	0.128278 / 0.428888	步驟は五音を中	0.483545 / 0.935093	步驟は五音に中たる
馬は四足有りと雖も	0.750624 / 0.970984	馬は四足有りと雖も	0.750624 / 0.970984	馬に四足有りと雖も
遅速は吾の心に在り	0.596949 / 0.970984	遅きは速めて吾の心に在り	0.252119 / 0.871331	遅速は吾が心に在り
六轡は吾の手を応ず	0.330316 / 0.903602	六轡は吾の手を応ず	0.330316 / 0.903602	六轡は吾が手に応じ
和すを調べるは瑟瑟の如し	0.388273 / 0.878905	調和は瑟瑟の如し	0.408665 / 0.931571	調和すること瑟瑟の如し
東西南とも北	0.204124 / 0.903602	東西は南北をとも	0.165158 / 0.840896	東西と南北と
下の山林ともを高ず	0.149237 / 0.602401	高下して山林とも	0.165158 / 0.355656	山と林とを高下す
ただ適さんと欲する所を意ふ	0.361123 / 0.723584	ただ意ふは適さんと欲する所	0.377964 / 0.898242	惟だ意の適かんと欲する所にして
九州は尋を周きべし	0.149237 / 0.843362	九州は尋を周きべし	0.149237 / 0.843362	九州周く尋ぬべし
至るか人馬とも	0.154479 / 0.935093	か人馬ともに至る	0.128278 / 0.428888	至れるかな人と馬と
両の楽はたがいに侵さず	0.158512 / 0.821097	両の楽はたがいに侵さず	0.158512 / 0.821097	両楽相侵さず
伯楽は其の外を識る	0.894839 / 0.988950	伯楽は其の外を識る	0.894839 / 0.988950	伯楽は其の外を識るも
徒に価に千金を知る	0.194376 / 0.918466	徒に価の千金を知る	0.357388 / 0.949644	徒だ価の千金なるを知る
王良は其の性を得る	0.772551 / 0.960255	王良は其の性を得る	0.772551 / 0.960255	王良は其の性を得たり
此の術は固く已に深す	0.185751 / 0.880112	此の術は固く已に深す	0.185751 / 0.880112	此術固り已に深し
良馬は善く馭するべし	0.277762 / 0.840896	良馬は善く馭するべし	0.277762 / 0.840896	良馬は善馭を須つ
吾の言は箴を為すべし	0.302138 / 0.914691	吾の言は箴を為すべし	0.302138 / 0.914691	吾が言箴と為すべし
全体	0.348924 / 0.846239	全体	0.367928 / 0.839227	

舞姫/雪國/荒野より-Benchmarks

UniDic-COMBOを製作するにあたり、われわれは、Transformersの日本語BERTモデル^[34]をどれにするか迷ったあげく、以下の6つのBERTモデルを用いた場合を、それぞれ予備実験として比較することにした。

- ⒱ cl-tohoku/bert-base-japanese-whole-word-masking
- Ⓒ cl-tohoku/bert-base-japanese-char
- Ⓓ bandainamco-mirai/distilbert-base-japanese
- Ⓔ Cinnamon/electra-small-japanese-generator
- Ⓐ ALINEAR/albert-japanese-v2
- Ⓤ microsoft/unihanlm-base

係り受けの教師データとしては、Universal Dependencies 2.7^[35]のUD_Japanese-GSD^[36]・UD_Japanese-Modern^[37]に加え、われわれが作成^[38]した『大學』書き下しデータ^[39]を用いた。ただし、見出し語形(LEMMA)はUniDic準拠となるようにし、それにとまってデータを取捨選択したため、最終的には7177文146444語となった。さらに、訓練パラメータを入れ換えることで見出し語形(LEMMA)を係り受けにおいて優先するモデルと、元のまま表層形(FORM)を係り受けにおいて優先するモデルを、両方とも作成した。つまり、BERT無しのモデルも含め、全部で14種類のモデルを構築した。訓練(係り受けモデル構築)に要した時間は、100epochs・1GPU(NVIDIA GeForce RTX 2080)で、各モデル1時間程度だった。なお、構築した係り受けモデルでは、UPOSの解析も同時におこなっているが、FEATSは使用していない。

このようにして構築した14種類のモデルを、UniDic2UD 2.7.4・spaCy-SynCha 0.8.1^[40]・spaCy-ChaPAS 0.7.0^[41]と共に、『舞姫』^[42]冒頭部・『雪國』^[43]冒頭部・『荒野より』^[44]抜粋で評価した。なお、形態素解析には、近代文語UniDic^[45]・旧仮名口語UniDic^[46]・近世口語UniDic^[47]を用いている。表4・5・6に結果を示す。表4においては、見出し語形優先の⒱が、圧倒的な精度を示している。表5においては、表層形優先の⒱が、比較

^[34]<https://huggingface.co/models?filter=ja>

^[35]<http://hdl.handle.net/11234/1-3424>

^[36]松田寛, 若狭絢, 山下華代, 大村舞, 浅原正幸: UD Japanese GSD の再整備と固有表現情報付与, 言語処理学会第26回年次大会発表論文集(2020年3月), pp.133-136.

^[37]Mai Omura, Yuta Takahashi, Masayuki Asahara: Universal Dependency for Modern Japanese, Proceedings of the 7th Conference of Japanese Association for Digital Humanities (September 2017), pp.34-36.

^[38]安岡孝一: 漢日英 Universal Dependencies 平行コーパスとその差異, 人文科学とコンピュータシンポジウム「じんもんこん2019」論文集(2019年12月), pp.43-50.

^[39]<https://corpus.kanji.zinbun.kyoto-u.ac.jp/gitlab/Kanbun/ud-ja-kanbun>

^[40]<https://github.com/KoichiYasuoka/spaCy-SynCha>

^[41]<https://github.com/KoichiYasuoka/spaCy-ChaPAS>

^[42]鷗外森林太郎: 舞姫, 國民之友, 第6巻, 第69號(1890年1月)附録, pp.45-61.

^[43]川端康成: 雪國, 東京: 創元社(1937年6月).

^[44]三島由紀夫: 荒野より, 群像, 第21巻, 第10號(1966年10月), pp.6-17.

^[45]小木曾智信, 小町守, 松本裕治: 歴史的日本語資料を対象とした形態素解析, 自然言語処理, Vol.20, No.5(2013年10月), pp.727-748.

^[46]小木曾智信: 旧仮名遣いの口語文を対象とした形態素解析辞書, 人文科学とコンピュータ「じんもんこん2012」論文集(2012年11月), pp.25-32.

^[47]小木曾智信, 市村太郎, 鴻野知曉: 近世口語資料の形態素解析の試み, 第4回コーパス日本語学ワークショップ予稿集(2013年9月), pp.145-150.

的精度が高いものの、それでも UniDic2UD に負けている。表 6 においては、表層形優先の㊦が MLAS / BLEX が高く、見出し語形優先の㊣が LAS が高い。

これらの結果を勘案し、UniDic-COMBO の BERT モデルには

㊦ cl-tohoku/bert-base-japanese-whole-word-masking

を採用した。また、見出し語形優先と表層形優先では、どちらが有利か判然としなかったことから、両方のモデルを使い分けられるようにした。

表 4: 各 BERT モデルを『舞姫』冒頭部で評価 (LAS / MLAS / BLEX)

		近代文語 UniDic	旧仮名口語 UniDic	近世口語 UniDic
見出し語形優先	㊦	84.91 / 77.78 / 85.19	83.02 / 77.78 / 85.19	75.93 / 67.86 / 71.43
	㊣	81.13 / 70.37 / 77.78	79.25 / 70.37 / 77.78	74.07 / 64.29 / 67.86
	㊤	83.02 / 74.07 / 81.48	81.13 / 74.07 / 81.48	74.07 / 67.86 / 67.86
	㊥	81.13 / 71.70 / 83.02	79.25 / 71.70 / 83.02	72.22 / 69.09 / 69.09
	㊡	83.02 / 70.37 / 77.78	83.02 / 74.07 / 81.48	72.22 / 60.71 / 64.29
	㊢	75.47 / 65.45 / 72.73	73.58 / 65.45 / 72.73	64.81 / 56.14 / 56.14
	無	79.25 / 62.96 / 74.07	75.47 / 62.96 / 70.37	70.37 / 57.14 / 60.71
表層形優先	㊦	81.13 / 62.96 / 77.78	79.25 / 62.96 / 77.78	72.22 / 53.57 / 64.29
	㊣	79.25 / 62.96 / 74.07	79.25 / 62.96 / 74.07	68.52 / 53.57 / 57.14
	㊤	81.13 / 66.67 / 77.78	79.25 / 66.67 / 77.78	72.22 / 60.71 / 64.29
	㊥	81.13 / 62.96 / 77.78	79.25 / 62.96 / 77.78	74.07 / 57.14 / 67.86
	㊡	77.36 / 62.96 / 70.37	75.47 / 62.96 / 70.37	70.37 / 60.71 / 60.71
	㊢	77.36 / 69.09 / 76.36	75.47 / 69.09 / 76.36	68.52 / 59.65 / 63.16
	無	79.25 / 62.96 / 74.07	75.47 / 62.96 / 70.37	70.37 / 57.14 / 60.71
UniDic2UD		81.13 / 70.37 / 77.78	79.25 / 70.37 / 77.78	72.22 / 60.71 / 64.29
spaCy-SynCha		83.02 / 66.67 / 70.37	81.13 / 66.67 / 70.37	72.22 / 57.14 / 57.14
spaCy-ChaPAS		79.25 / 59.26 / 62.96	77.36 / 59.26 / 62.96	70.37 / 53.57 / 53.57

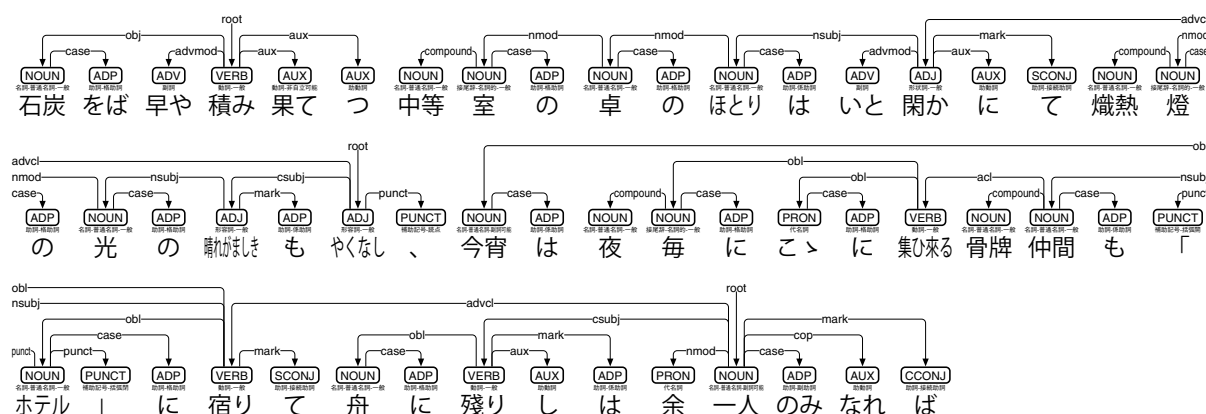


表 5: 各 BERT モデルを『雪國』冒頭部で評価 (LAS / MLAS / BLEX)

		近代文語 UniDic	旧仮名口語 UniDic	近世口語 UniDic
見出し語形優先	㊦	83.19 / 78.43 / 74.51	87.50 / 82.35 / 78.43	78.57 / 73.08 / 69.23
	㊣	83.19 / 78.43 / 74.51	87.50 / 82.35 / 78.43	78.57 / 73.08 / 69.23
	㊤	81.42 / 74.51 / 70.59	85.71 / 78.43 / 74.51	83.93 / 74.51 / 70.59
	㊥	76.11 / 66.67 / 66.67	80.36 / 70.59 / 70.59	78.57 / 62.75 / 66.67
	㊦	74.34 / 68.00 / 64.00	78.57 / 72.00 / 68.00	69.64 / 66.67 / 58.82
	㊧	81.42 / 74.51 / 70.59	85.71 / 78.43 / 74.51	76.79 / 68.00 / 64.00
	無	81.42 / 74.51 / 70.59	85.71 / 78.43 / 74.51	83.93 / 74.51 / 70.59
表層形優先	㊦	83.19 / 78.43 / 74.51	87.50 / 82.35 / 78.43	87.50 / 78.43 / 74.51
	㊣	83.19 / 78.43 / 74.51	85.71 / 78.43 / 74.51	85.71 / 74.51 / 70.59
	㊤	74.34 / 69.23 / 65.38	78.57 / 73.08 / 69.23	78.57 / 73.08 / 65.38
	㊥	79.65 / 74.51 / 70.59	85.71 / 78.43 / 74.51	85.71 / 78.43 / 70.59
	㊦	81.42 / 74.51 / 70.59	85.71 / 78.43 / 74.51	85.71 / 78.43 / 70.59
	㊧	81.42 / 74.51 / 70.59	85.71 / 78.43 / 74.51	85.71 / 74.51 / 70.59
	無	74.34 / 68.00 / 64.00	78.57 / 72.00 / 68.00	78.57 / 72.00 / 64.00
UniDic2UD		84.96 / 81.63 / 77.55	89.29 / 85.71 / 81.63	89.29 / 85.71 / 77.55
spaCy-SynCha		83.19 / 77.55 / 73.47	87.50 / 81.63 / 77.55	85.71 / 83.33 / 75.00
spaCy-ChaPAS		83.19 / 77.55 / 73.47	87.50 / 81.63 / 77.55	85.71 / 77.55 / 69.39

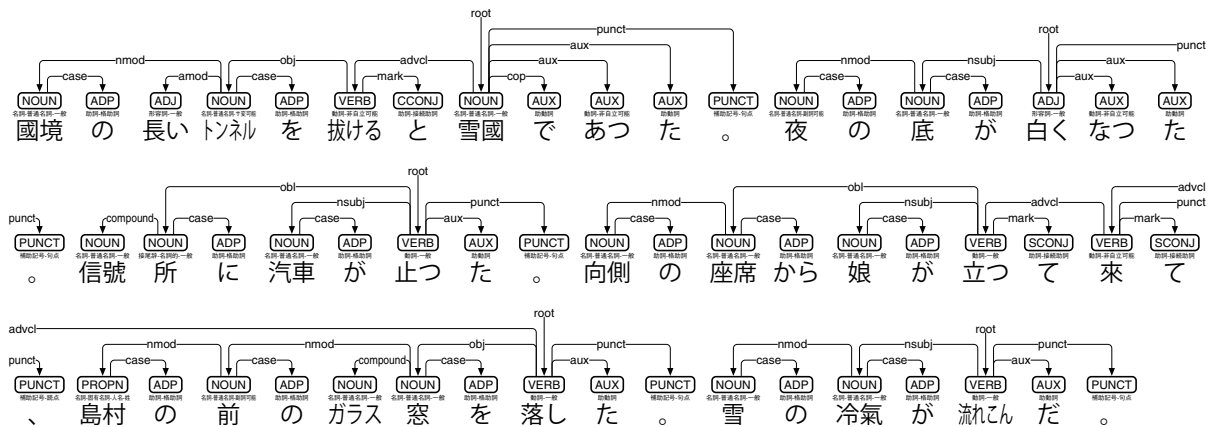
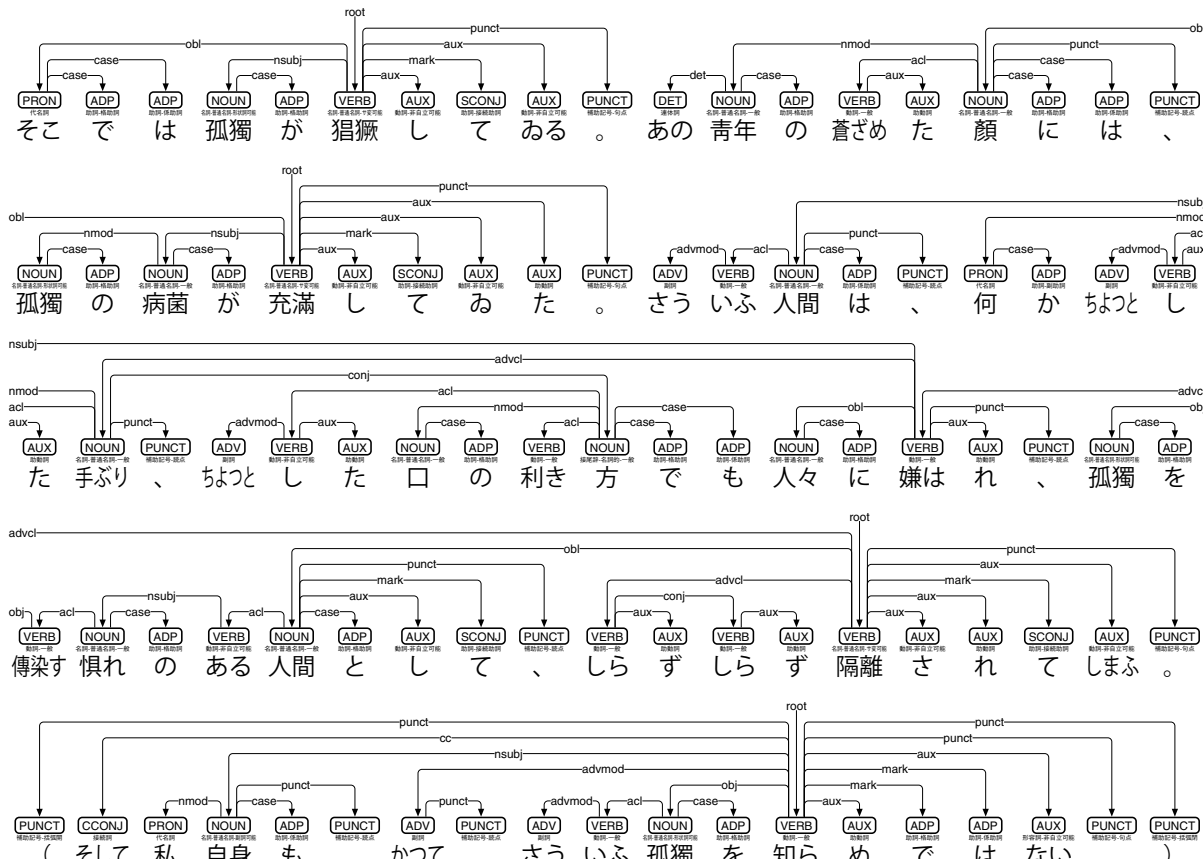
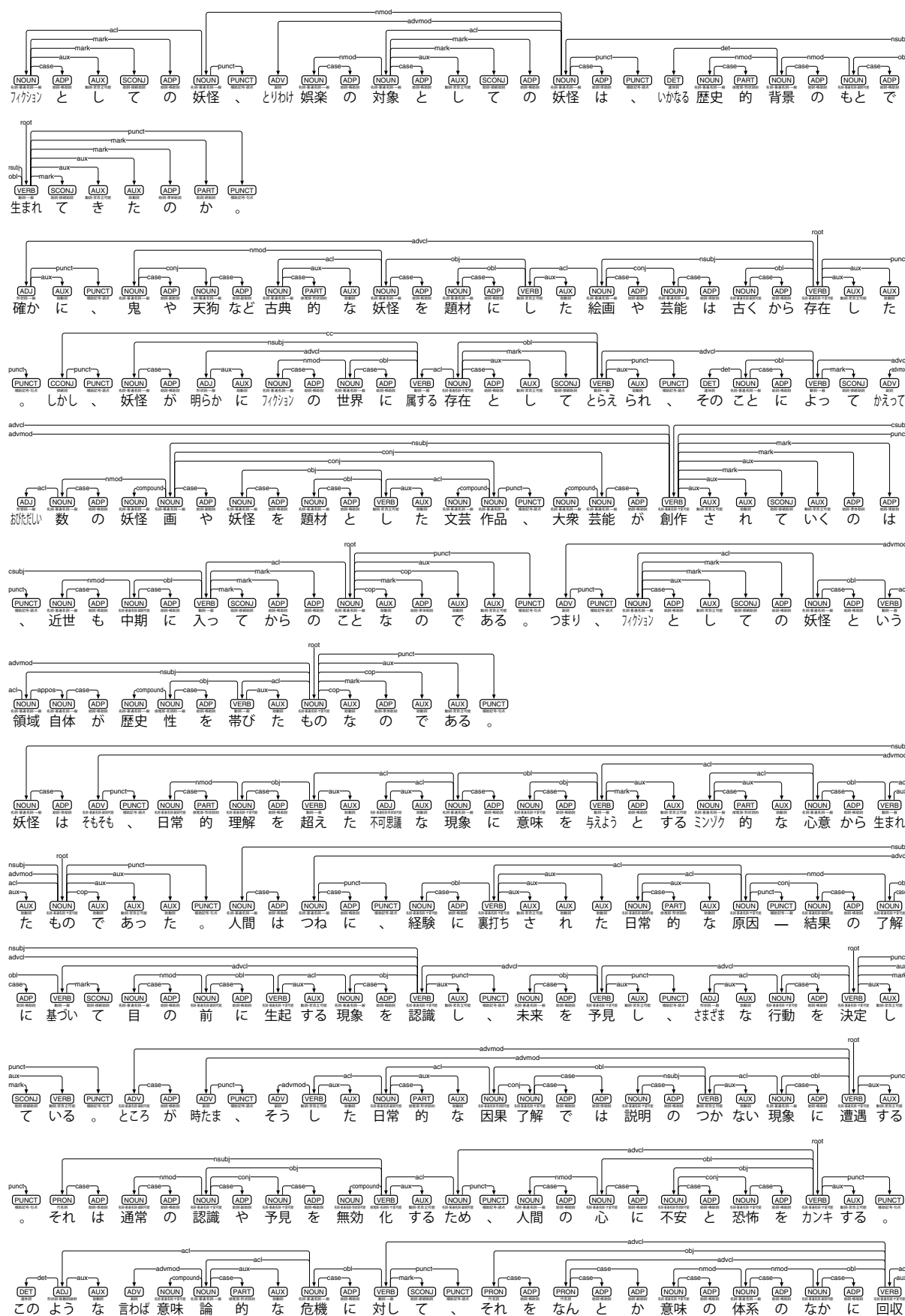


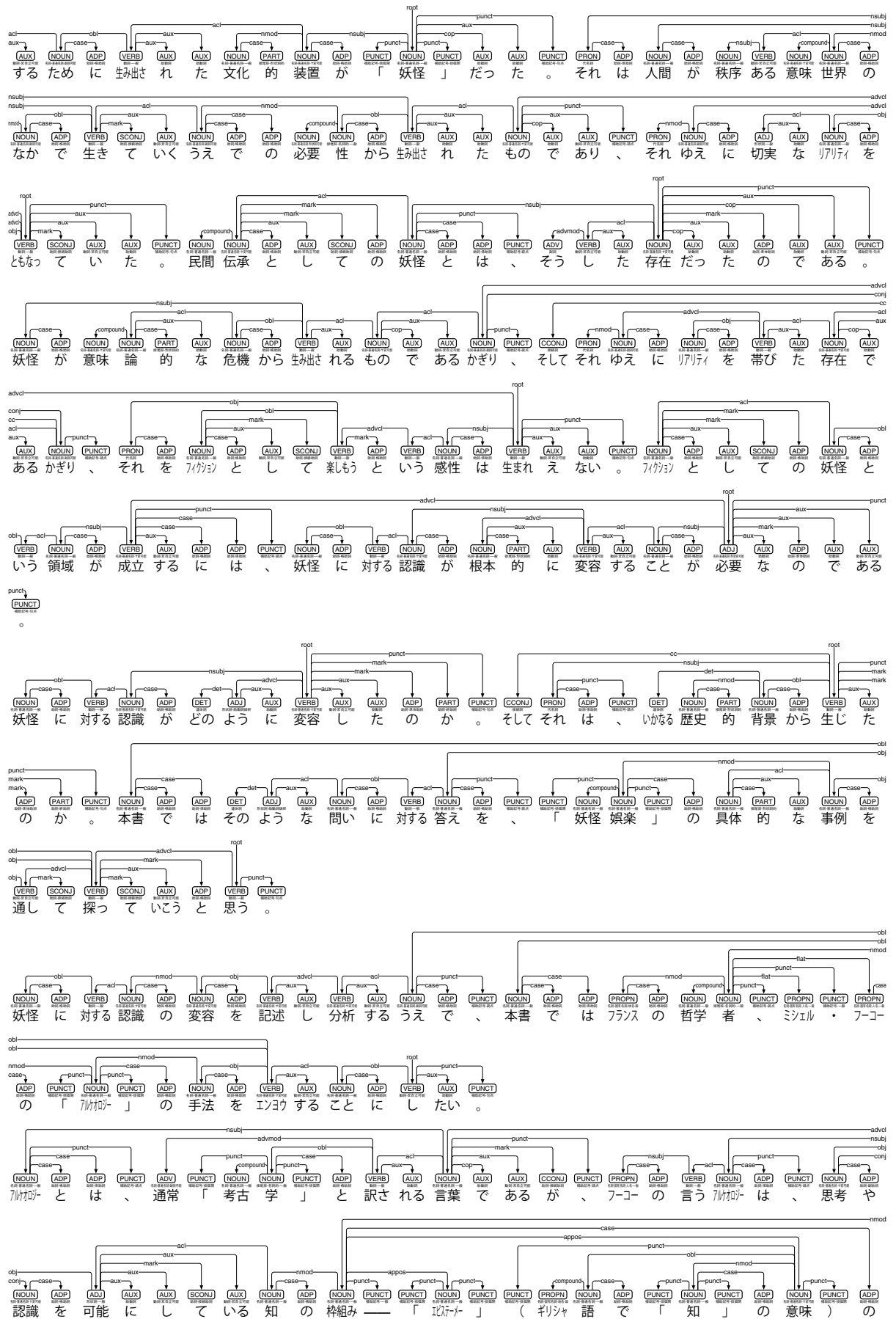
表 6: 各 BERT モデルを『荒野より』抜粋で評価 (LAS / MLAS / BLEX)

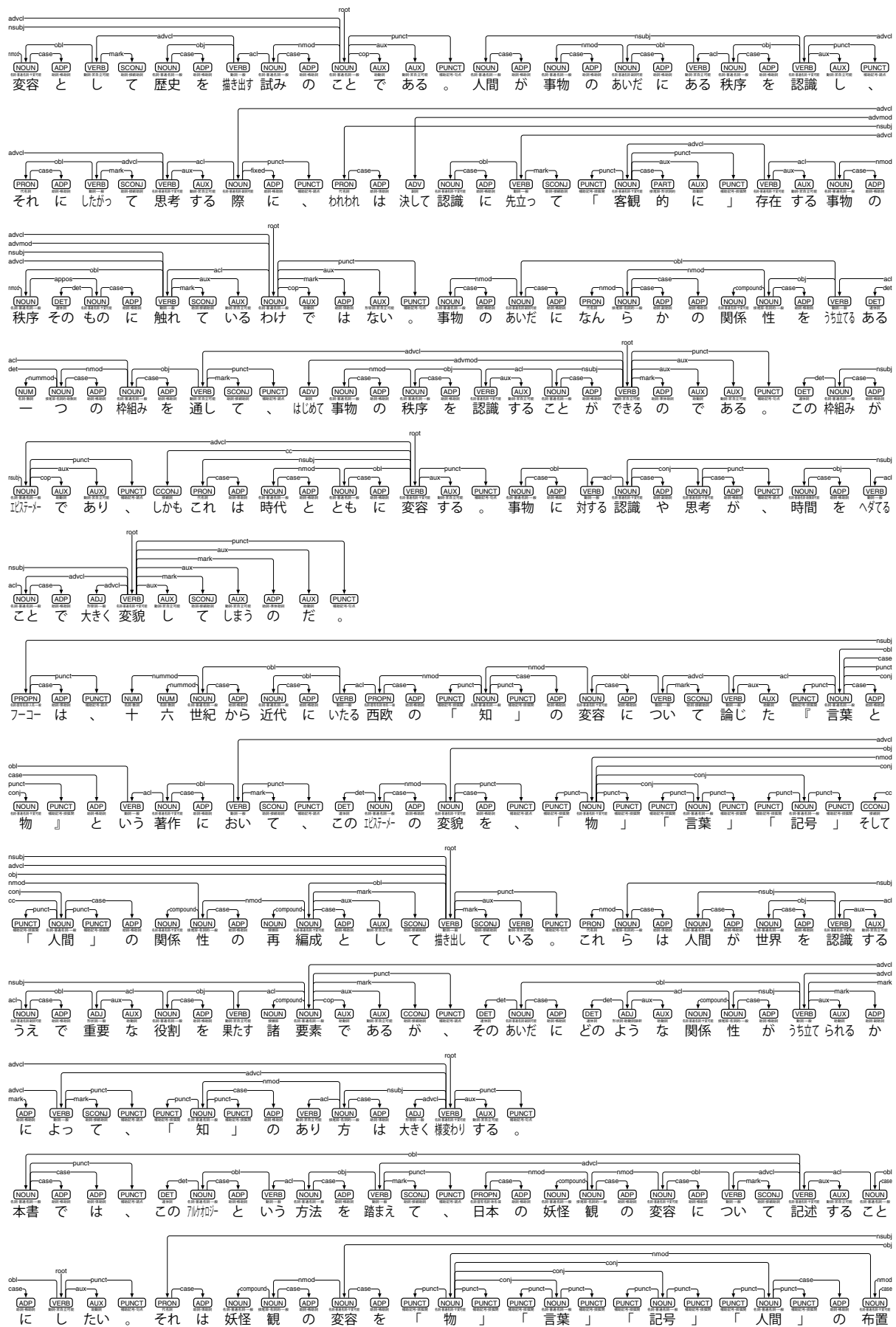
		近代文語 UniDic	旧仮名口語 UniDic	近世口語 UniDic
見出し語形優先	㊦	78.53 / 59.46 / 59.46	77.49 / 59.46 / 59.46	76.04 / 59.46 / 59.46
	㊢	80.63 / 62.34 / 59.74	77.49 / 60.53 / 57.89	77.08 / 59.74 / 57.14
	㊣	76.44 / 56.00 / 56.00	75.39 / 58.67 / 58.67	72.92 / 53.33 / 53.33
	㊥	74.35 / 53.33 / 53.33	72.25 / 53.33 / 53.33	71.88 / 50.67 / 50.67
	㊤	76.44 / 56.00 / 56.00	75.39 / 58.67 / 58.67	72.92 / 53.33 / 53.33
	㊧	74.35 / 53.33 / 53.33	72.25 / 53.33 / 53.33	70.83 / 50.67 / 50.67
	無	77.49 / 55.26 / 57.89	74.35 / 56.00 / 56.00	69.79 / 50.00 / 52.63
表層形優先	㊦	79.58 / 63.16 / 63.16	77.49 / 63.16 / 63.16	76.04 / 60.53 / 60.53
	㊢	73.30 / 50.67 / 53.33	71.20 / 50.67 / 53.33	69.79 / 48.00 / 50.67
	㊣	77.49 / 58.67 / 58.67	74.35 / 58.67 / 58.67	73.96 / 56.00 / 56.00
	㊥	75.39 / 53.33 / 56.00	72.25 / 50.67 / 53.33	73.96 / 53.33 / 56.00
	㊤	71.20 / 47.37 / 50.00	70.16 / 50.00 / 52.63	67.71 / 44.74 / 47.37
	㊧	78.53 / 54.05 / 56.76	76.44 / 54.05 / 56.76	75.00 / 51.35 / 54.05
	無	74.35 / 51.28 / 48.72	73.30 / 53.85 / 51.28	70.83 / 48.72 / 46.15
UniDic2UD		76.44 / 61.54 / 53.85	75.39 / 61.54 / 53.85	71.88 / 58.97 / 51.28
spaCy-SynCha		68.06 / 35.14 / 45.95	63.87 / 32.88 / 43.84	63.54 / 29.73 / 40.54
spaCy-ChaPAS		68.06 / 35.14 / 45.95	64.92 / 35.14 / 45.95	64.58 / 32.43 / 43.24

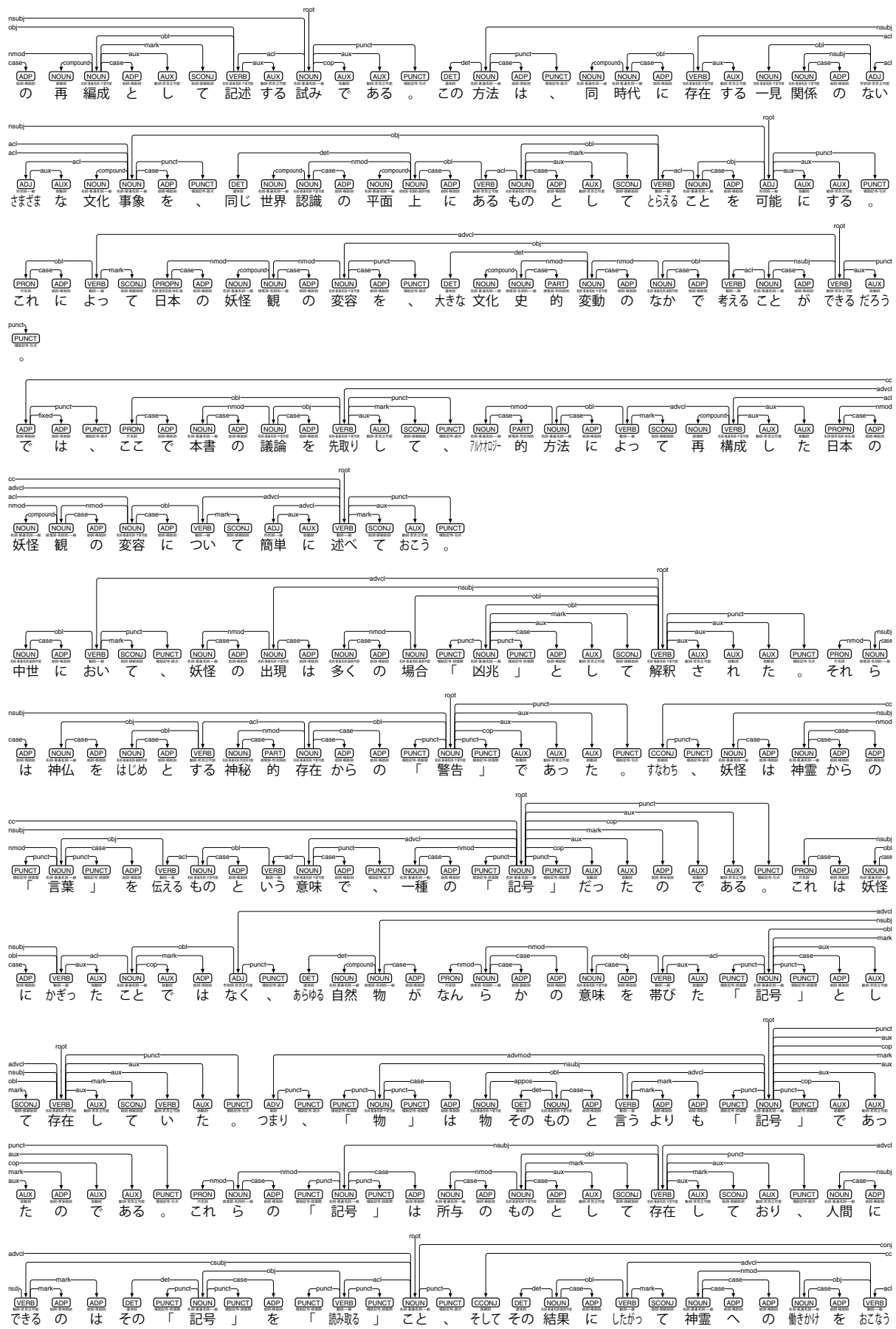


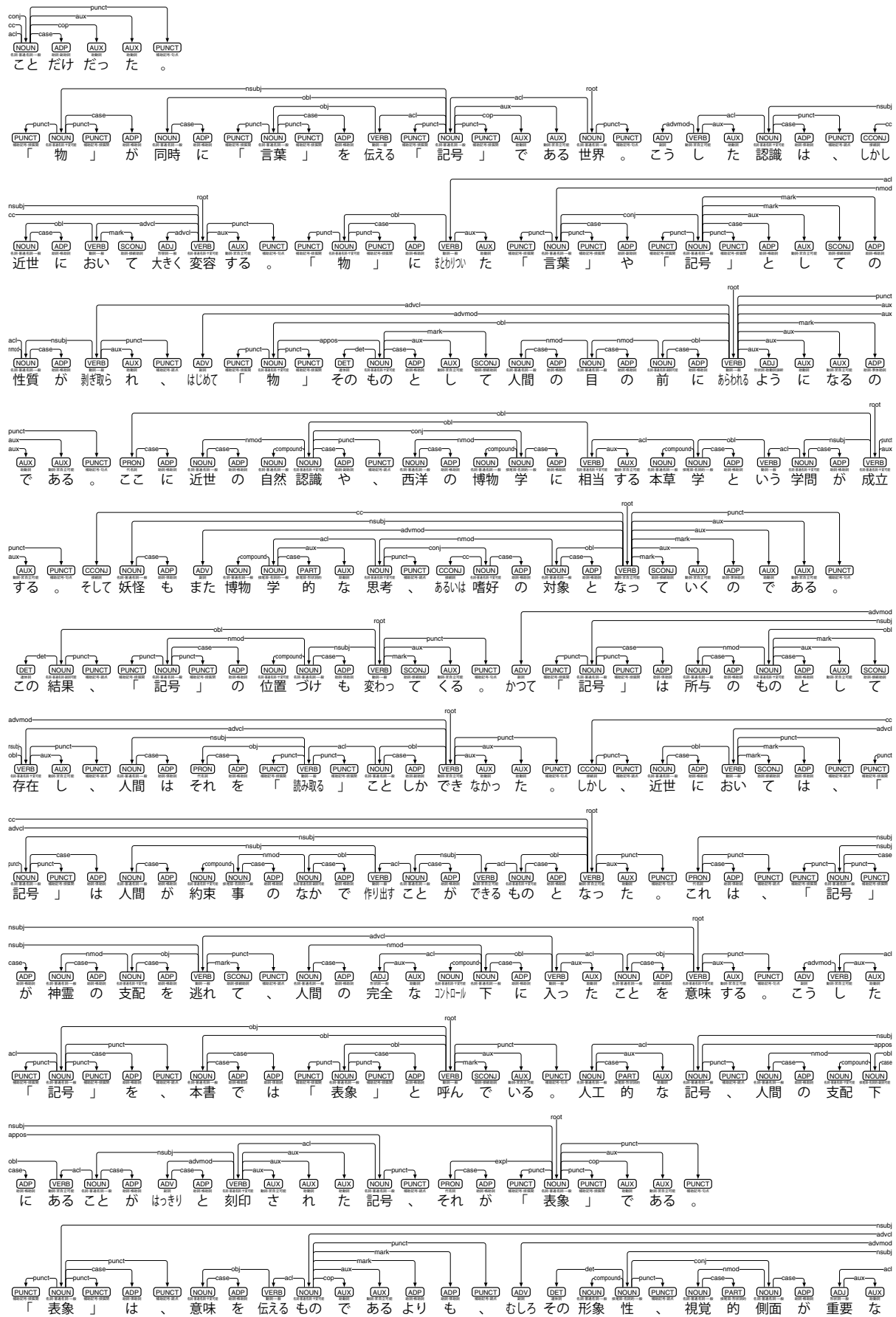
共通テスト『国語』第1問 [現代文]

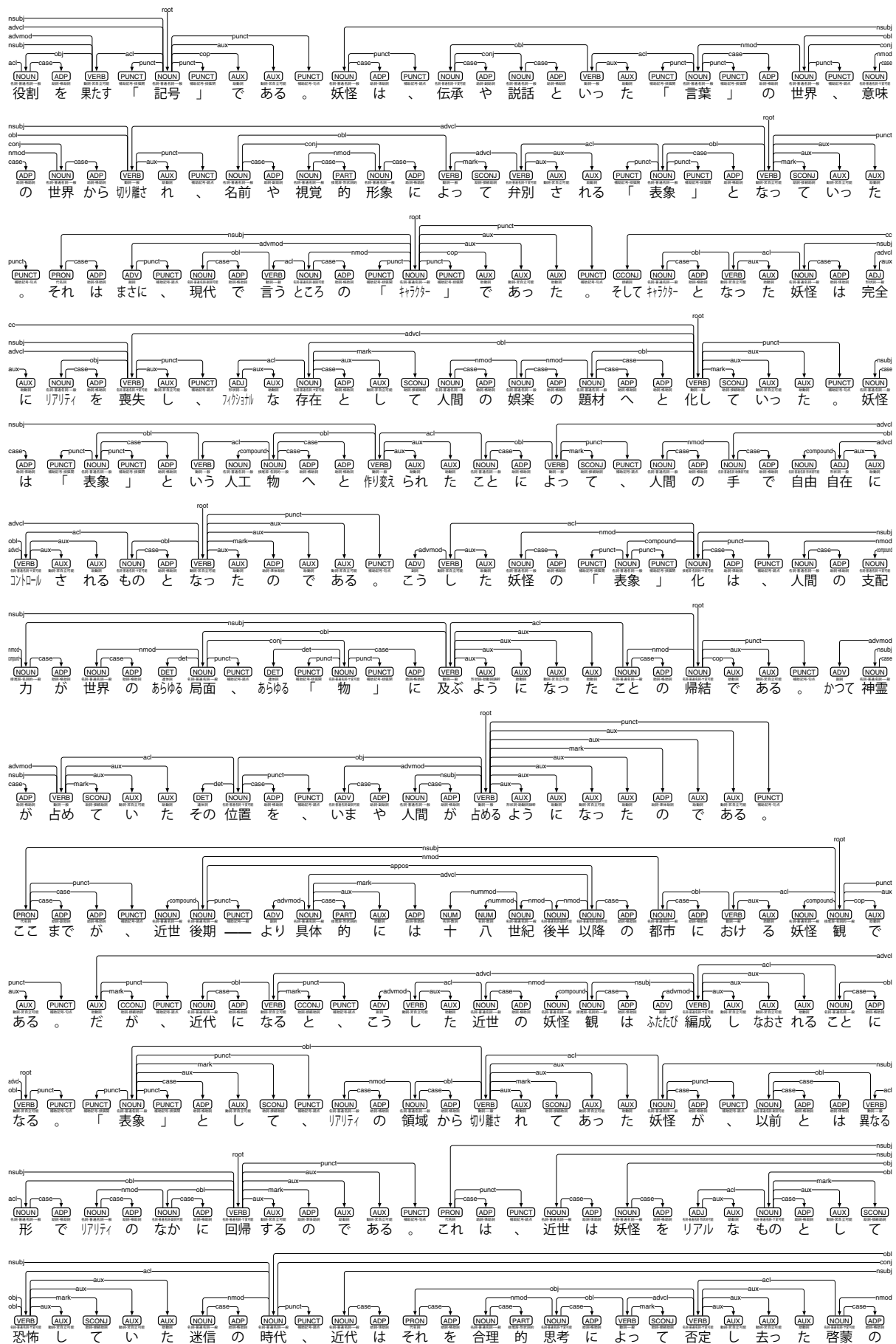


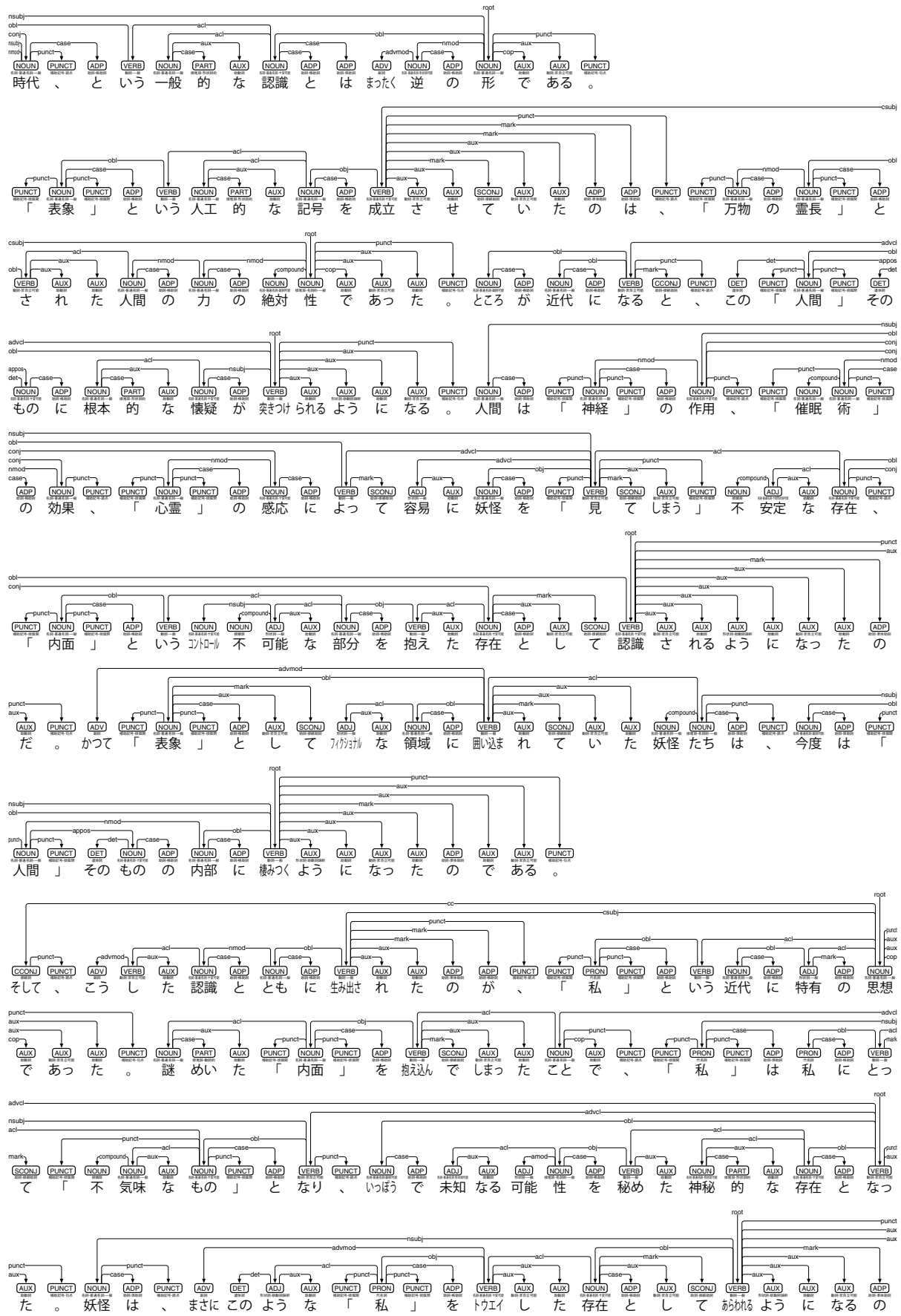


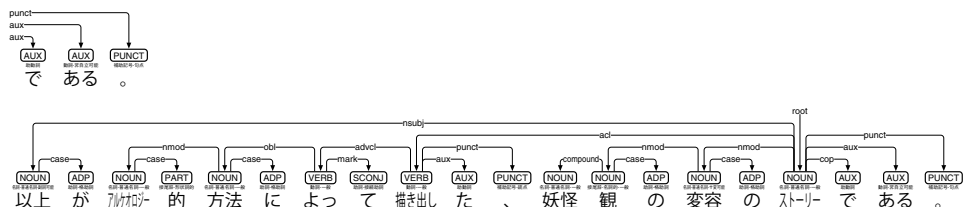












共通テスト『国語』第1問の問題文は、18の段落から構成されているが、これをひとまとめにして、全体を UniDic-COMBO 1.1.1 の評価に用いた。形態素解析には、現代書き言葉 UniDic^[48]・現代話し言葉 UniDic^[49]を用いた。UniDic2UD・spaCy-SynCha・spaCy-ChaPAS・GiNZA 4.0.5^[50]・Stanza (ja モデル) と共に、結果を表7に示す。

表 7: UniDic-COMBO を第1問で評価 (LAS / MLAS / BLEX)

	現代書き言葉 UniDic	現代話し言葉 UniDic
UniDic-COMBO ^{BERT} 見出し語形優先	78.07 / 61.57 / 68.09	77.95 / 61.41 / 67.93
	75.89 / 57.43 / 63.18	75.78 / 56.96 / 62.83
UniDic-COMBO ^{BERT} 表層形優先	79.73 / 63.94 / 70.23	79.56 / 63.66 / 69.95
	75.56 / 56.96 / 62.36	75.49 / 56.84 / 62.13
UniDic2UD	79.30 / 62.67 / 67.02	79.23 / 62.63 / 66.86
spaCy-SynCha	63.90 / 39.46 / 45.94	63.79 / 39.18 / 45.53
spaCy-ChaPAS	60.26 / 40.14 / 45.38	60.15 / 39.91 / 45.03
GiNZA	67.05 / 44.19 / 38.58	
Stanza	72.19 / 48.51 / 45.30	

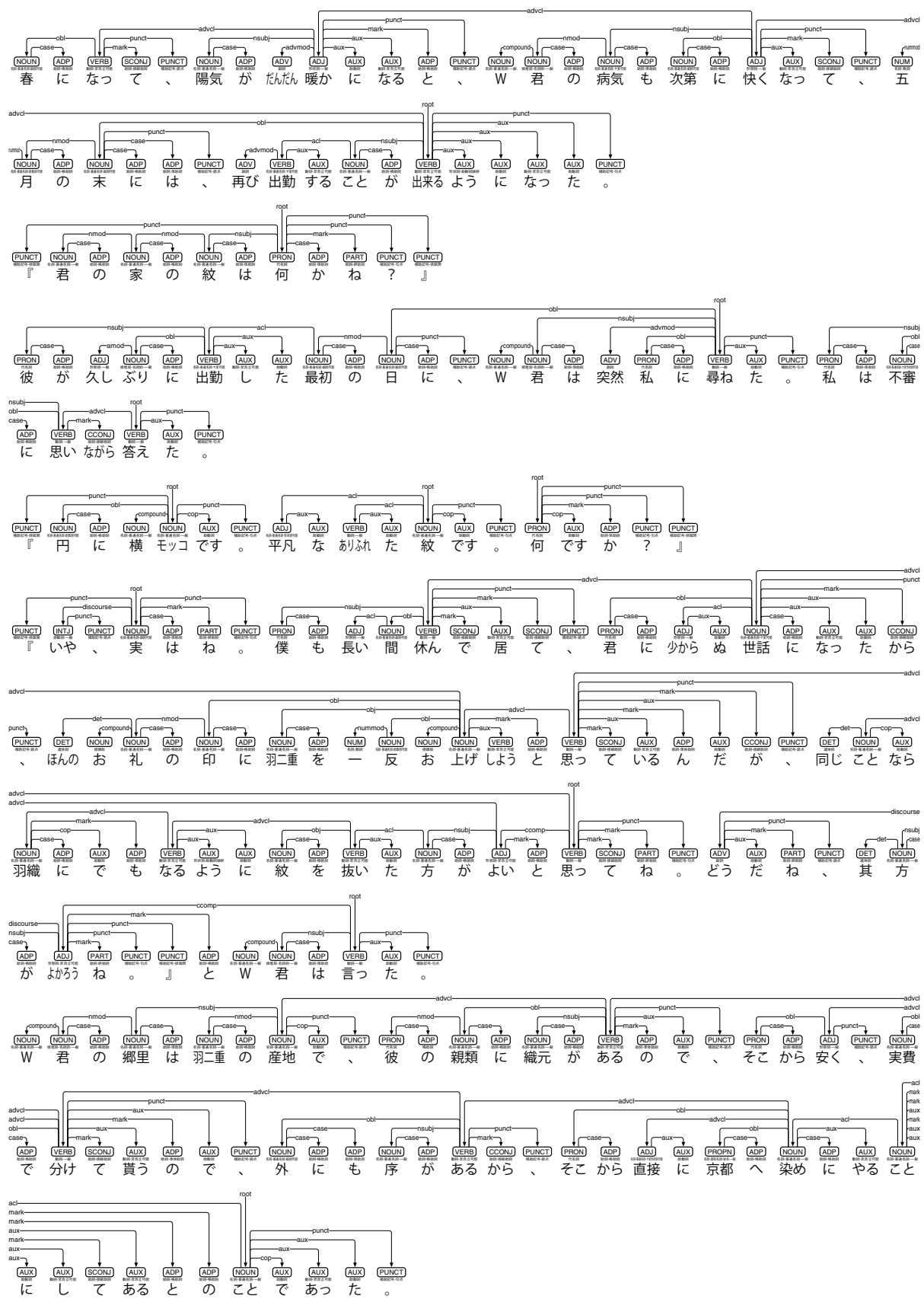
UniDic-COMBO において、BERT モデルを用いた方が精度が高くなっている。中でも、表層形優先の BERT モデルが、最も精度が高い。日本語の現代文においては、BERT モデルが十分な効果を発揮するということだろう。

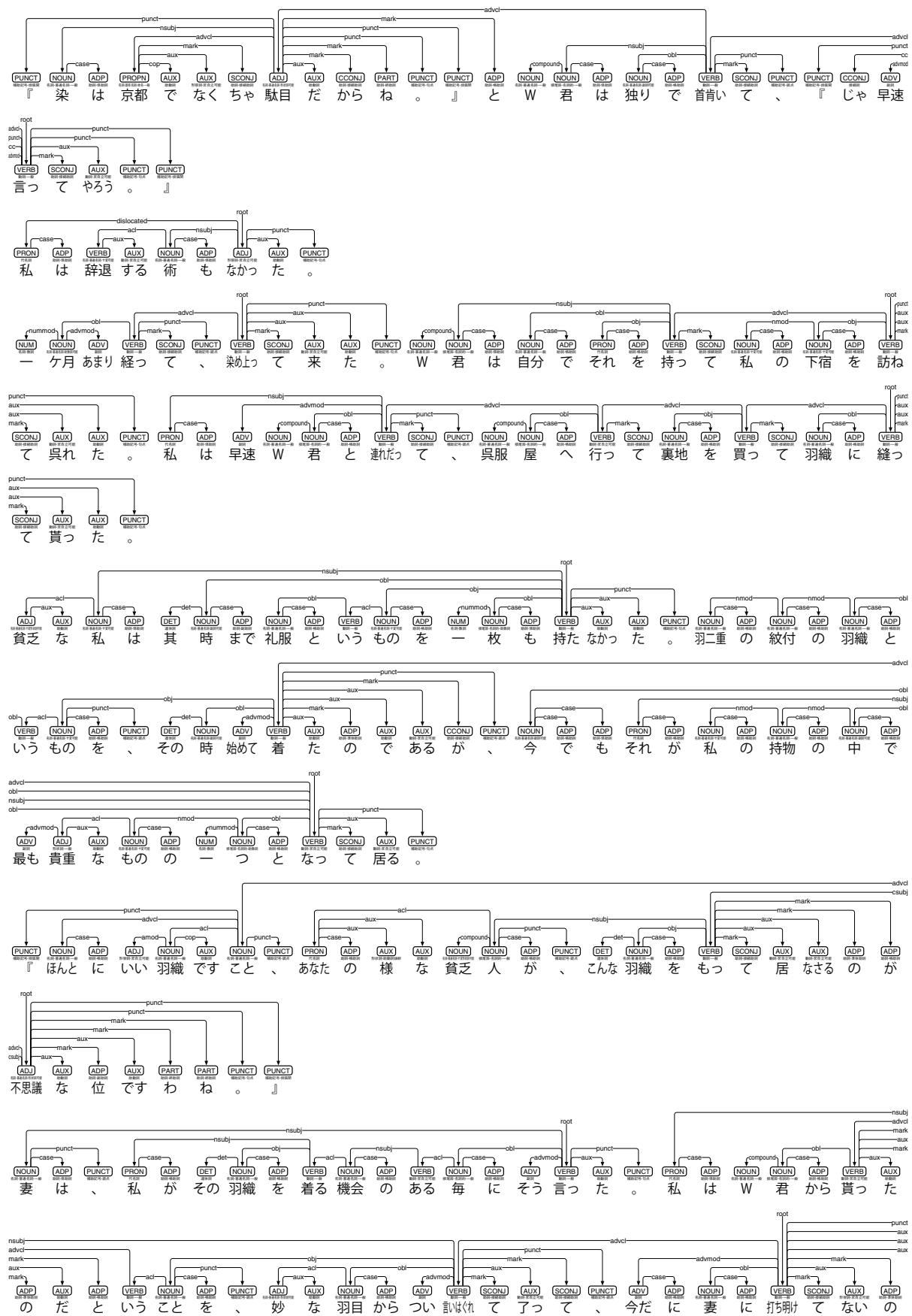
^[48]伝康晴, 小木曾智信, 小椋秀樹, 山田篤, 峯松信明, 内元清貴, 小磯花絵: コーパス日本語学のための言語資源: 形態素解析用電子化辞書の開発とその応用, 日本語科学, 第22号 (2007年10月), pp.101-123.

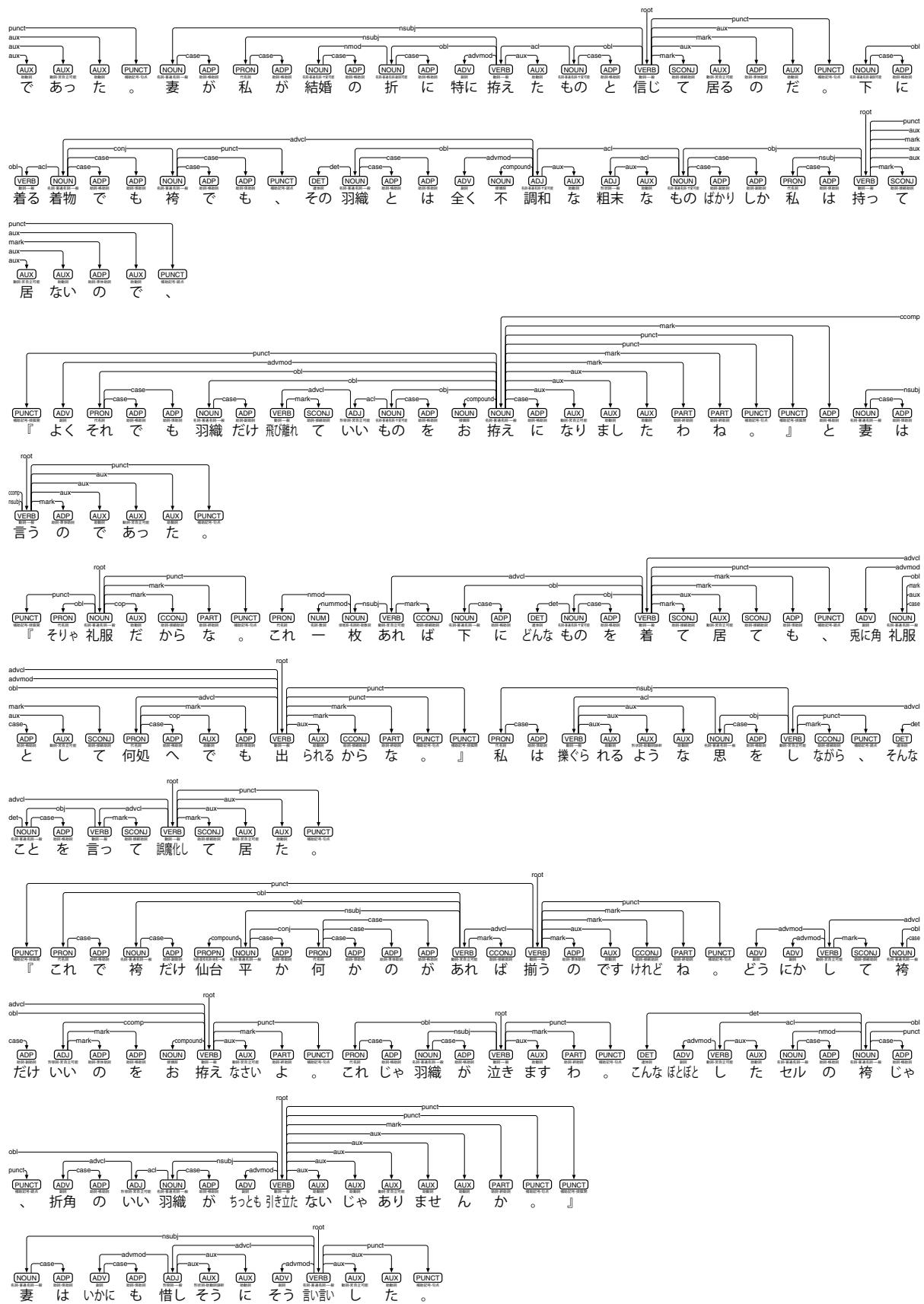
^[49]岡照晃: 言語研究のための電子化辞書, コーパスと辞書, 東京: 朝倉書店 (2019年3月), pp.1-28.

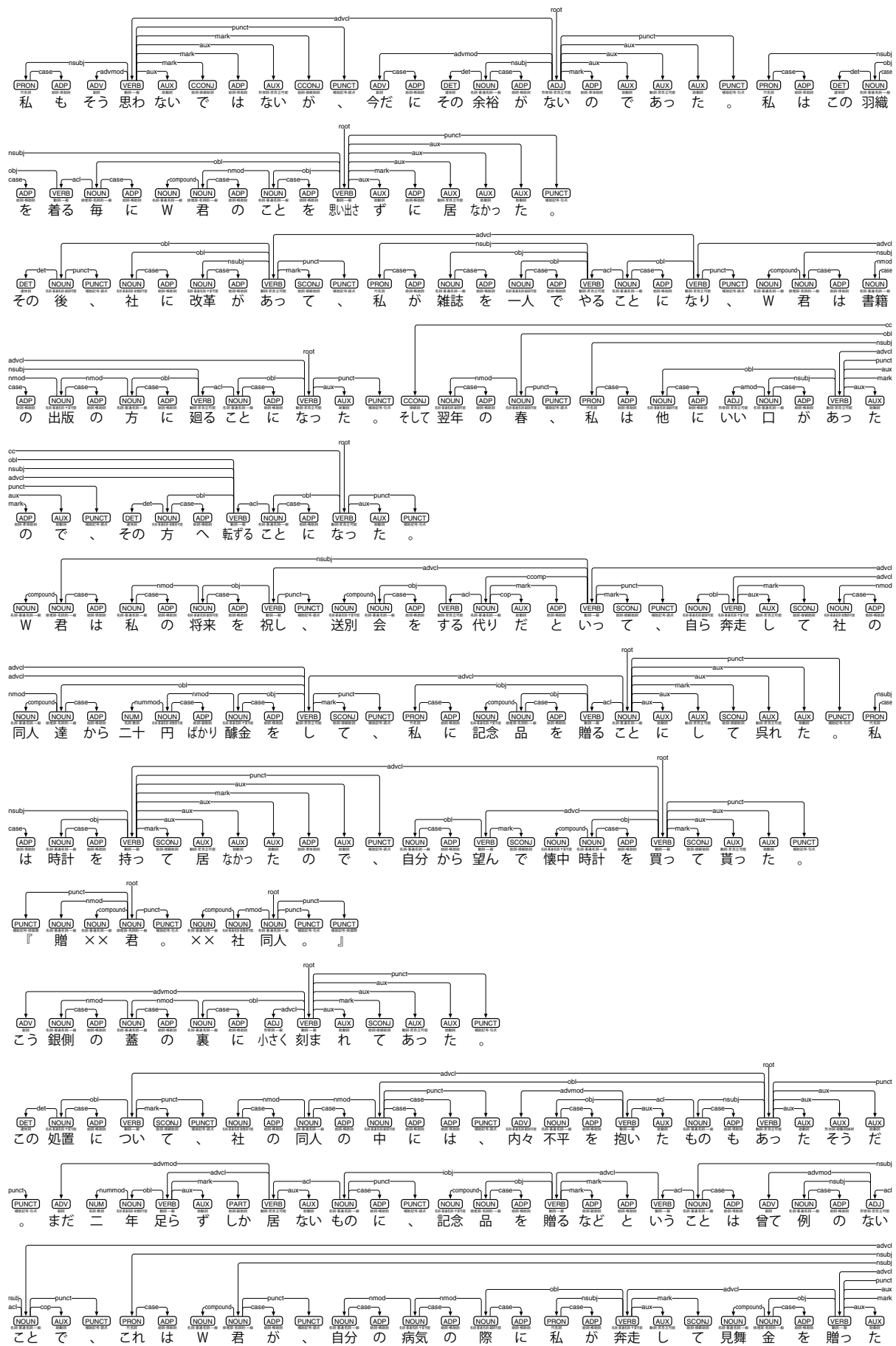
^[50]松田寛: GiNZA - Universal Dependencies による実用的日本語解析, 自然言語処理, Vol.27, No.3 (2020年9月), pp.695-701.

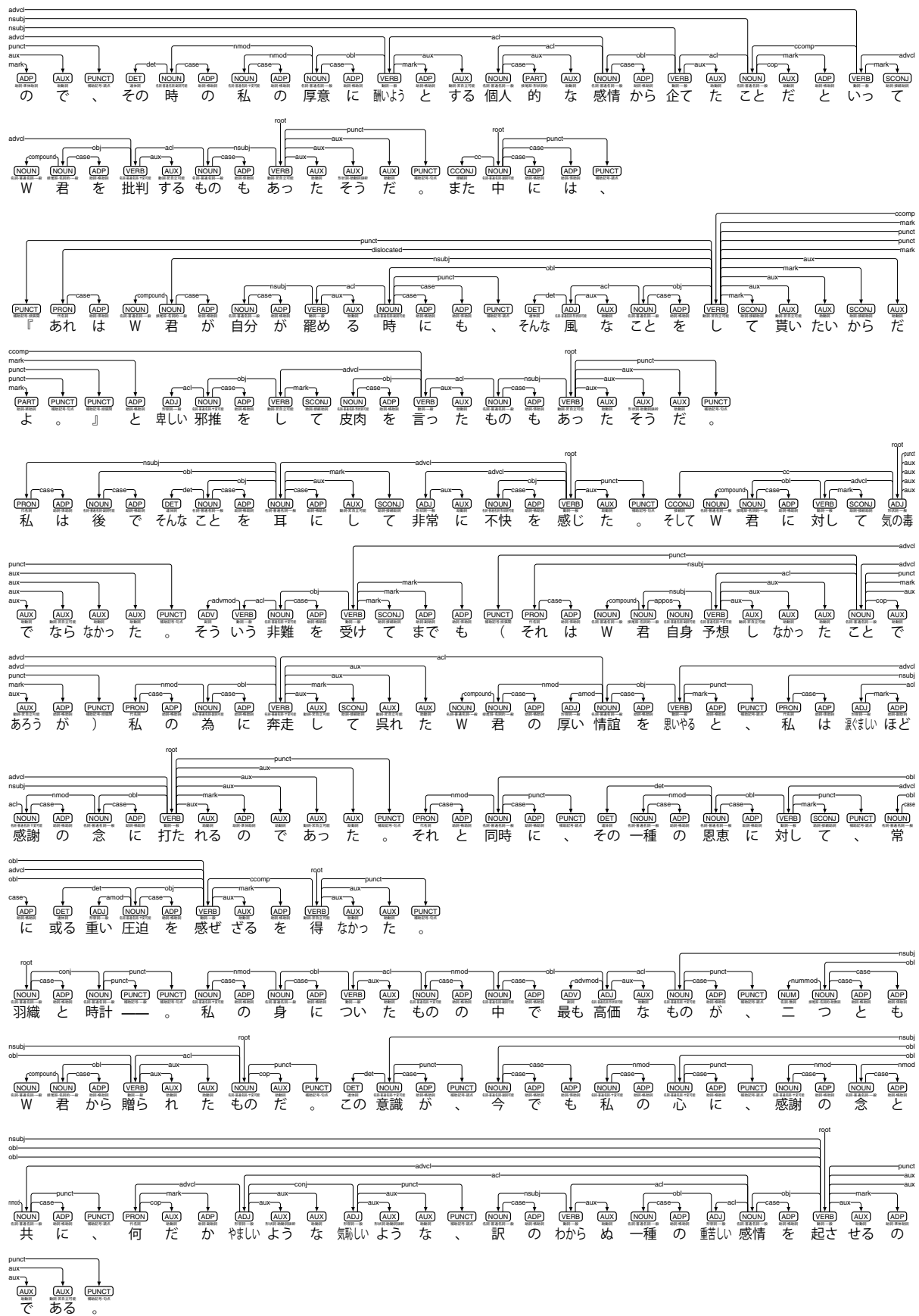
共通テスト『国語』第2問 [近代文]

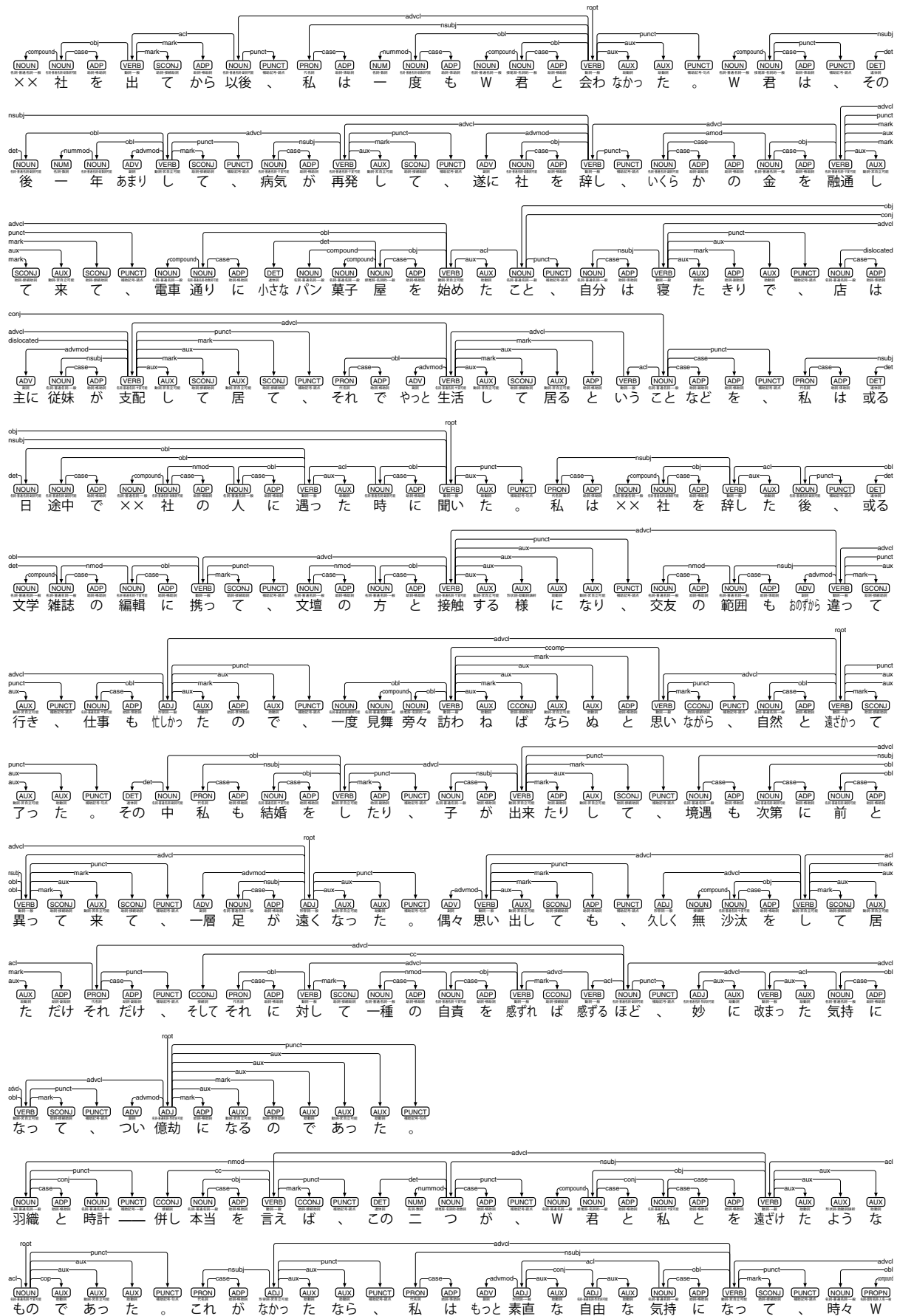


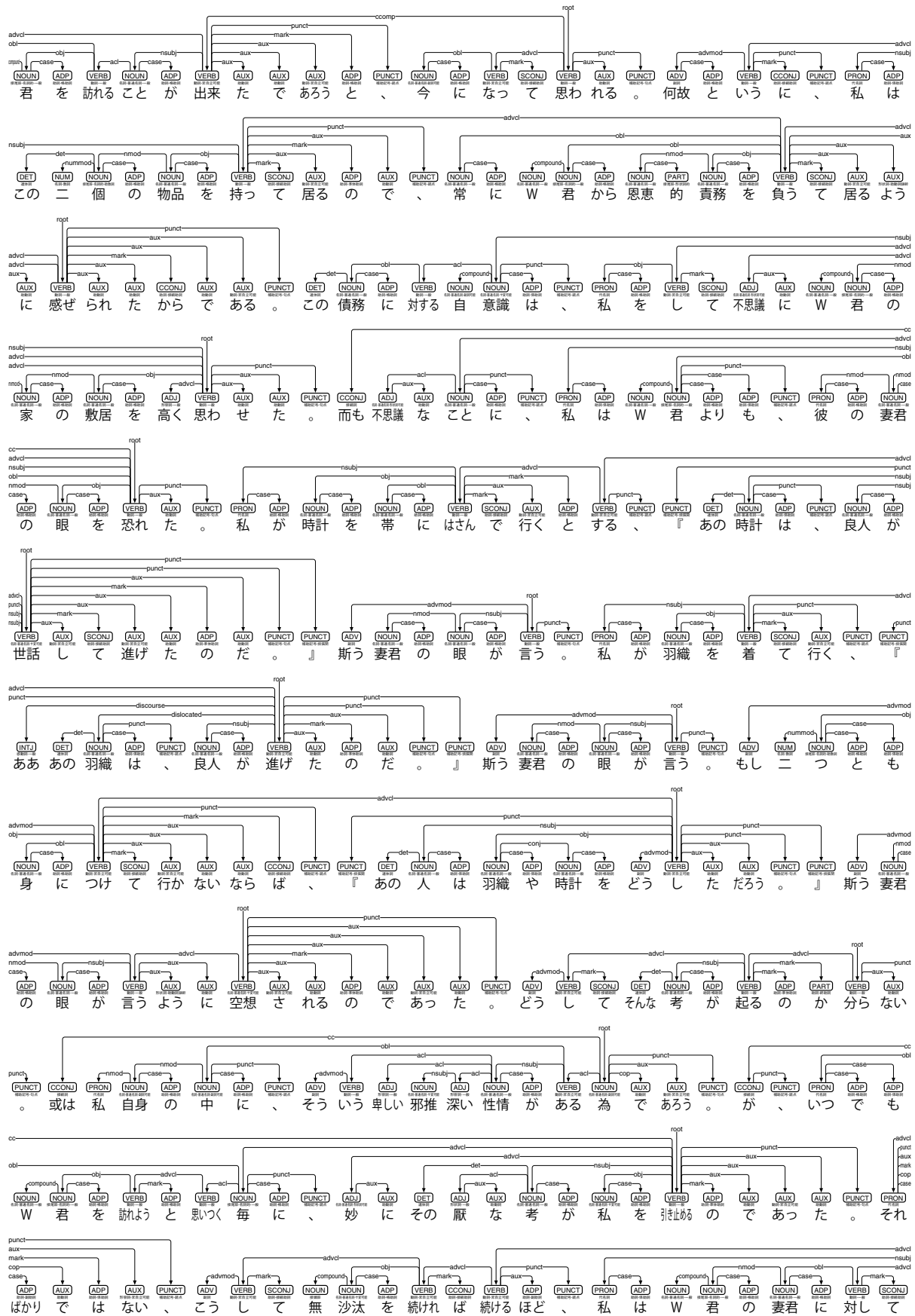


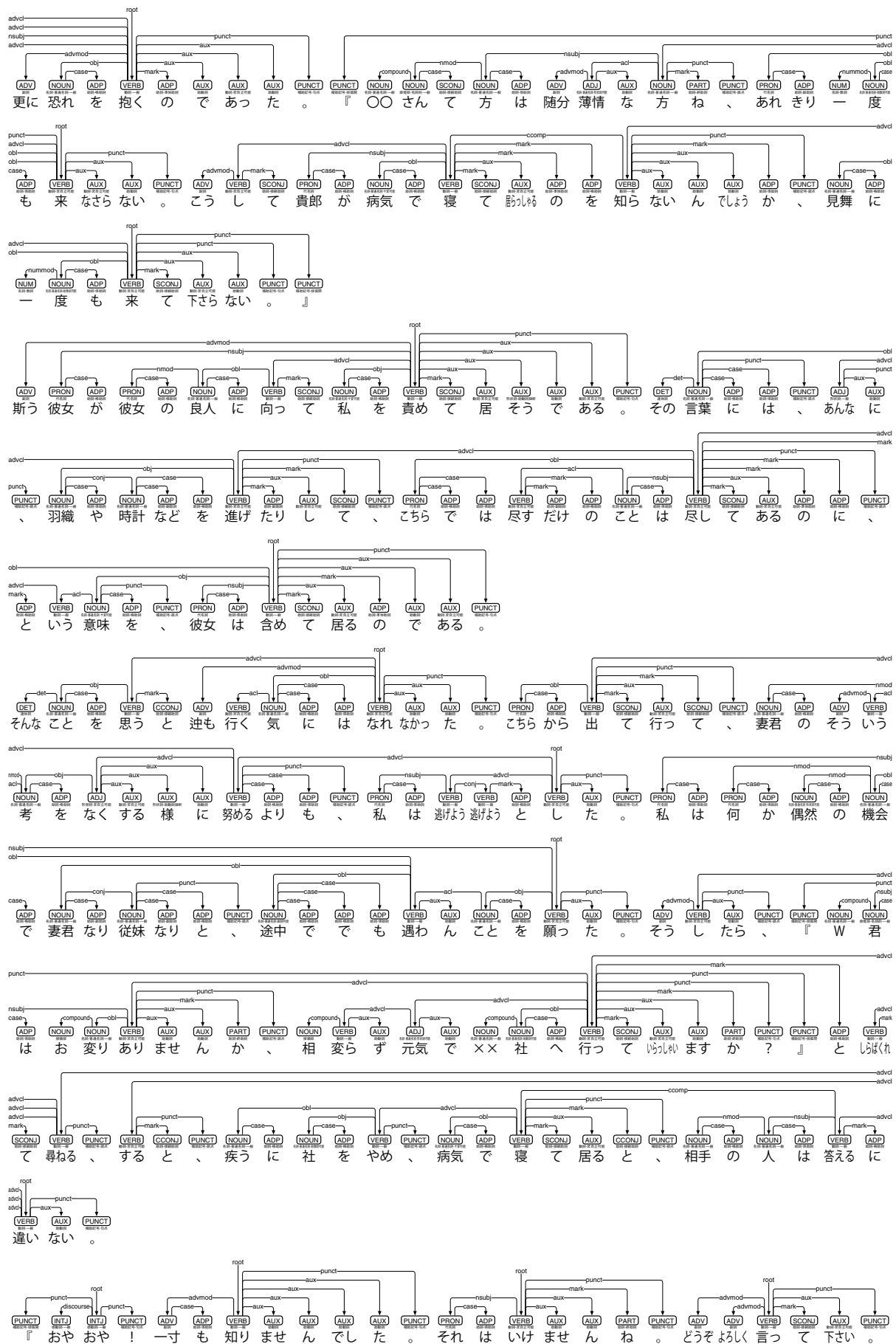


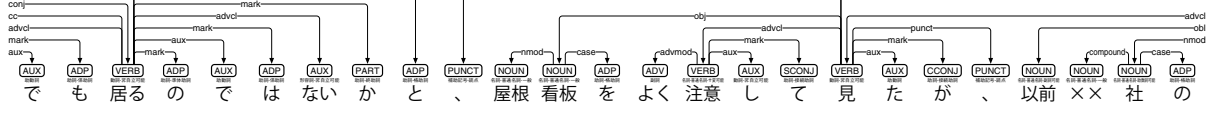
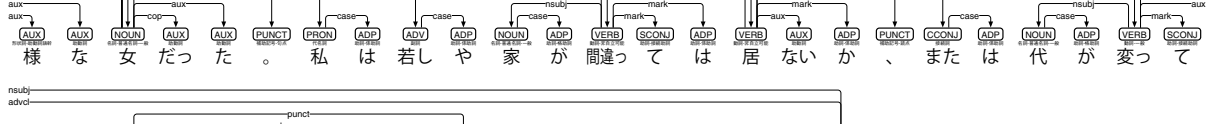
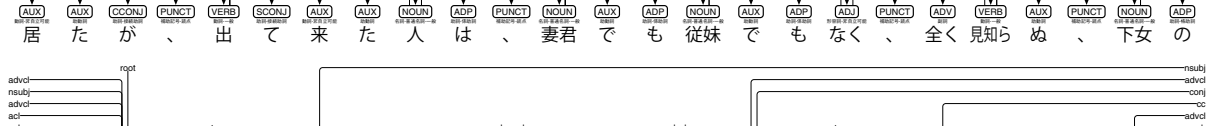
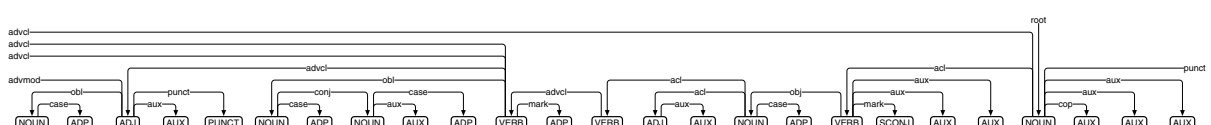
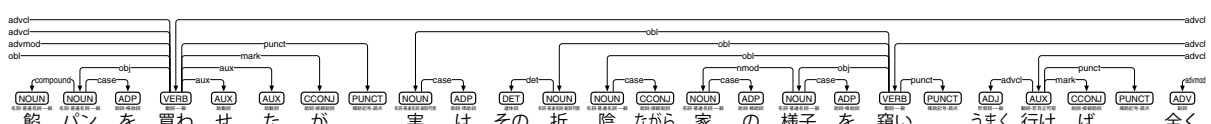
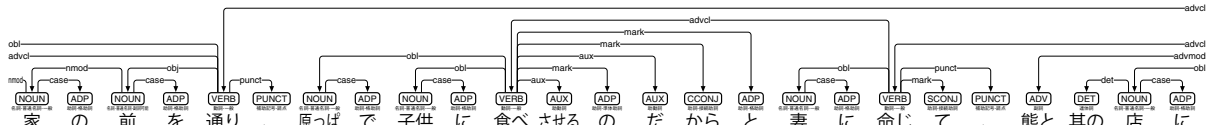
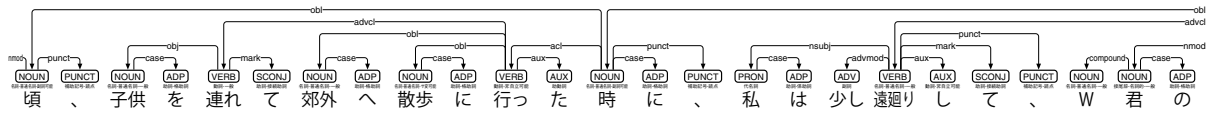
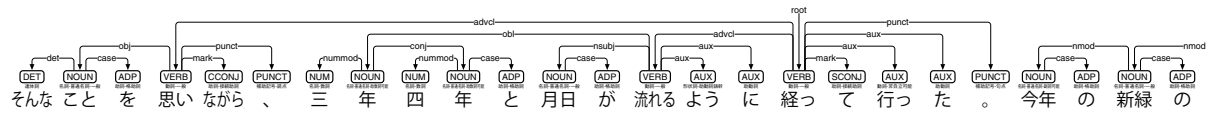
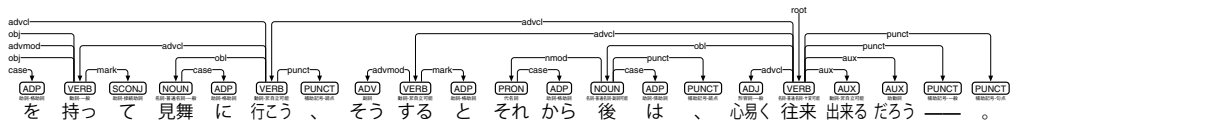
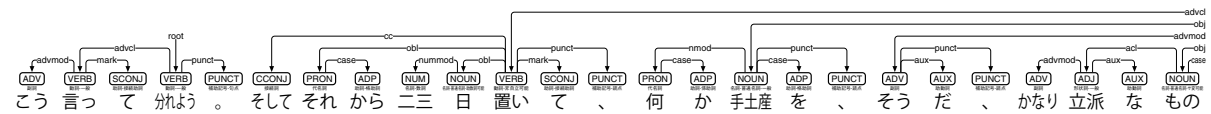
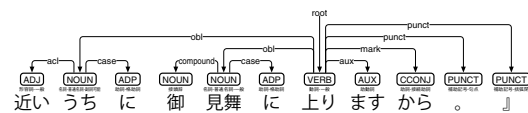


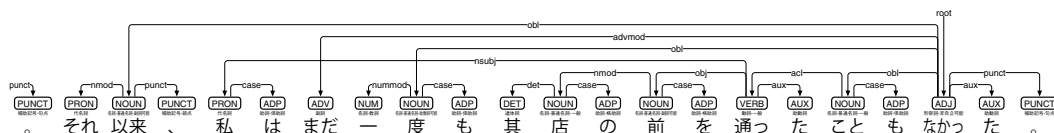












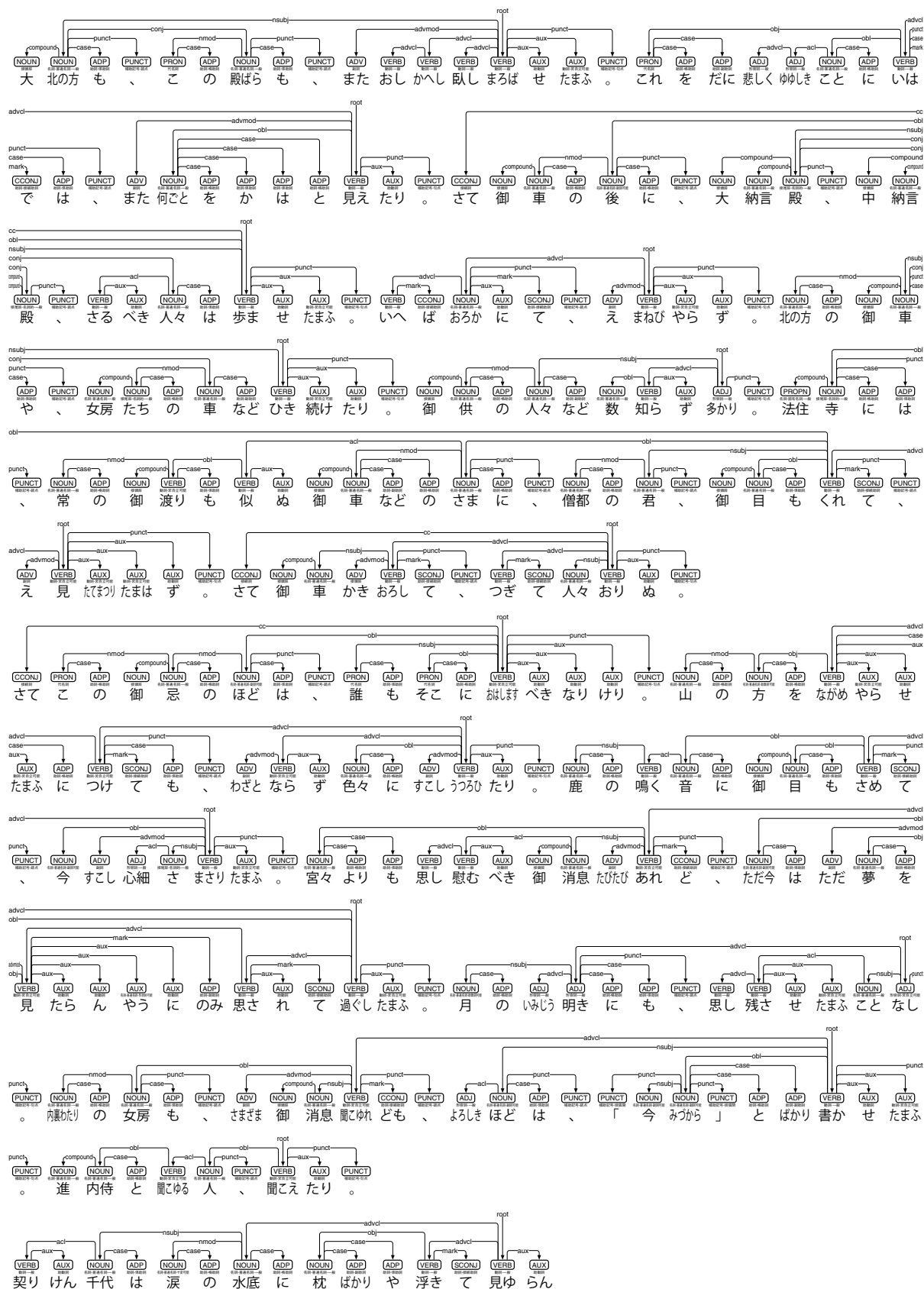
共通テスト『国語』第2問の問題文を用いて、UniDic-COMBOの評価をおこなった。形態素解析には、近代文語 UniDic・現代書き言葉 UniDic を用いた。UniDic2UD・spaCy-SynCha・spaCy-ChaPAS・GiNZA・Stanza (ja モデル) と共に、結果を表8に示す。

表 8: UniDic-COMBO を第2問で評価 (LAS / MLAS / BLEX)

	近代文語 UniDic	現代書き言葉 UniDic
UniDic-COMBO ^{BE} _{RT}	75.23 / 58.67 / 64.35	75.94 / 60.23 / 61.30
見出し語形優先	74.59 / 57.83 / 62.99	75.14 / 58.61 / 59.48
UniDic-COMBO ^{BE} _{RT}	76.07 / 60.01 / 65.49	75.86 / 60.12 / 61.19
表層形優先	72.58 / 55.54 / 60.67	72.97 / 56.48 / 56.87
UniDic2UD	78.48 / 63.70 / 68.84	78.55 / 61.49 / 65.03
spaCy-SynCha	66.96 / 47.35 / 52.55	65.70 / 44.62 / 49.59
spaCy-ChaPAS	59.86 / 42.79 / 47.19	57.83 / 39.78 / 43.25
GiNZA	65.59 / 44.00 / 41.73	
Stanza	64.65 / 46.31 / 42.47	

UniDic-COMBOの結果は、いずれも UniDic2UD に負けている。UniDic-COMBOの結果だけを見た場合には、見出し語形優先であっても、表層形優先であっても、BERTモデルの効果が出ている。だがしかし、日本語の近代文においては、十分にチューニングされた UniDic2UD (つまり UDPipe) に、UniDic-COMBO (現代日本語 BERT モデルによる COMBO-pytorch) は敵わない。

共通テスト『国語』第3問〔古文〕



共通テスト『国語』第3問の問題文を用いて、UniDic-COMBOの評価をおこなった。形態素解析には、中古和文 UniDic^[51]・中世文語 UniDic^[45]を用いた。UniDic2UD・spaCy-SynCha・spaCy-ChaPAS と共に、結果を表9に示す。

表 9: UniDic-COMBO を第3問で評価 (LAS / MLAS / BLEX)

	中古和文 UniDic	中世文語 UniDic
UniDic-COMBO BERT 見出し語形優先	70.07 / 55.53 / 58.91	60.85 / 46.04 / 47.55
UniDic-COMBO BERT 表層形優先	71.15 / 56.17 / 58.82	60.67 / 44.49 / 44.49
UniDic-COMBO BERT 表層形優先	67.74 / 52.71 / 55.70	58.69 / 42.18 / 44.44
UniDic-COMBO BERT 表層形優先	66.31 / 48.21 / 52.73	55.99 / 38.11 / 40.00
UniDic2UD	80.11 / 65.51 / 67.82	68.05 / 51.84 / 52.61
spaCy-SynCha	69.18 / 45.69 / 56.11	60.49 / 36.62 / 44.27
spaCy-ChaPAS	73.12 / 46.78 / 57.31	61.93 / 37.65 / 45.10

UniDic-COMBOの結果は、いずれも UniDic2UD に負けている。見出し語形優先の場合には、BERT モデルの効果もあまり出ていない。

^[51] 小木曾智信, 小椋秀樹, 田中牧郎, 近藤明日子, 伝康晴: 中古和文を対象とした形態素解析辞書の開発, 情報処理学会研究報告, Vol.2010-CH-85 (2010 年 2 月), No.4, pp.1-8.

おわりに

古典中国語係り受け解析システム GuwenCOMBO と、日本語係り受け解析システム UniDic-COMBO を、開発・公開した。共通テスト『国語』による評価の結果をまとめると

第4問 古典中国語 BERT モデルは、たとえ簡化字で作られていても、古典中国語の係り受け解析に効く

第1問 Wikipedia をもとにした現代日本語 BERT モデルは、現代日本語の係り受け解析に効く

第2問 Wikipedia をもとにした現代日本語 BERT モデルは、近代日本語の係り受け解析にあまり効かない

第3問 Wikipedia をもとにした現代日本語 BERT モデルは、中古和文の係り受け解析に全く効かない。

ということになるだろう。端的に言えば、適材適所ということだ。

ただ、近代日本語や中古和文の BERT モデルを作成するのは、正直なところ、われわれの手に余る。次善策として、UniDic 見出し語をもとにした NWJC-BERT^[52]の採用も考えたが、しかし、NWJC-BERT は有償配布^[53]である。UniDic-COMBO に組み込むわけにはいかないだろう。近代日本語や中古和文の係り受け解析に対しては、しばらくは UniDic2UD で我慢するしかない、ということである。

^[52]浅原正幸, 西内沙恵, 加藤祥: NWJC-BERT: 多義語に対するヒトと文脈化単語埋め込みの類似性判断の対照分析, 言語処理学会第 26 回年次大会発表論文集 (2020 年 3 月), pp.961-964.

^[53]<https://www.gsk.or.jp/catalog/gsk2020-e/>