Transformersと国語研長単位による日本語係り受け解析モデルの製作

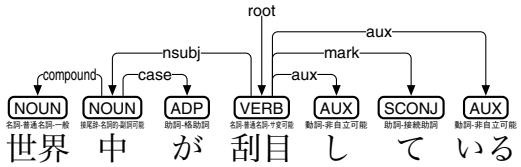
安岡孝一 (京都大学人文科学研究所附属東アジア人文情報学研究センター)

1 はじめに

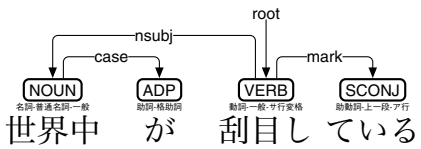
2021 年 11 月リリースの Universal Dependencies (UD) 2.9 において，国語研長単位コーパス [1] が追加された．UD 2.8.1 以前の日本語係り受けコーパスは，いずれも国語研短単位を採用していた [2, 3] が，UD 2.9 以降は，国語研短単位 UD と国語研長単位 UD の 2 本建て (図 1) となったわけだ．まずはめでたい．

これまで，日本語 UD に基づく係り受け解析ツールは，筆者が製作したものも含め，いずれも国語研短単位を採用してきた [4, 5]．ただ，国語研短単位は，少なくとも筆者に取っては扱いにくく，ツール開発に難渋してきた．たとえば図 1(a) では,「刮目」それ自体は名詞であるにもかかわらず，UD 品詞 (表 1) を VERB としなければならない．直後の「し」は動詞だが AUX とする．スル動詞をバラバラに扱う国語研短単位は，様々な実装上のトリックを必要とするのである．その点，国語研長単位は,「刮目し」を 1 語とみなす (図 1(b)) 点で，まだ扱いやすい．ただし，UDPipe [6] を用いた国語研長単位係り受け解析器は，解析精度の点で難がある [1] らしく，現時点の筆者としては Transformers [7] に頼るしかない．

本稿では，8 つの日本語 BERT/RoBERTa モデル (表 2) を下敷きに，国語研長単位による日本語係り受け解析モデルを製作し，その解析性能を比較評価することにした．なお，RoBERTa モデル①〜④は，データ活用社会創生プラットフォーム mdx「お試しプロジェクト」により作成したものである．



(a) 国語研短単位 UD



(b) 国語研長単位 UD

図 1: 日本語 UD における単語長

2 Universal Dependencies の概要

UD は，書写言語における品詞・形態素属性・依存構造 (係り受け関係) を，言語に関わらず記述する手法である．句構造を考慮せずに係り受け関係を記述することで，言語横断性を高めており，全ての文法構造を単語間のリンクで記述するのが特徴である．

依存構造解析それ自体は，Tesnière の構造的統語論 [8] に源を発し，Мельчук の有向グラフ記述 [9] によって，一応の完成を見た手法である．その最大の特長は，いわゆる動詞中心主義によって言語横断的な記述が可能だという点にあり，Мельчук 依存文法をコンピュータ向けに洗練した UD においても，言語に関わらない記述，という特長が前面に押し出されている．UD における文法構造記述は，句構造を考慮せず，全てを単語間のリンクとして表現する．これは，Мельчук の有向グラフ記述が，単語間のリンクという形態を取っていたからであり，そういう割り切りの結果として，言語横断的な文法構造記述を可能としているのである．

UD 依存構造コーパスの交換用フォーマットとして，CoNLL-U と呼ばれるタブ区切りテキスト (文字コードは UTF-8) が規定されている．CoNLL-U の各

表 1: UD 品詞タグ (UPOS)

Open class words	Closed class words	Other
ADJ 形容詞	ADP 側置詞	PUNCT 句読点
ADV 副詞	AUX 助動詞	SYM 記号
INTJ 感嘆詞	CCONJ 並列接続詞	X その他
NOUN 名詞	DET 限定詞	
PROPN 固有名詞	NUM 数詞	
VERB 動詞	PART 接辞	
	PRON 代名詞	
	SCONJ 従属接続詞	

表 2: 本稿で使用する日本語 BERT/RoBERTa モデルとその諸元

① `https://huggingface.co/KoichiYasuoka/roberta-base-japanese-aozora`
青空文庫 3 億字 (元データ 2.37 億字＋異体字増量分 0.64 億字) を Japanese-LUW-Tokenizer でトークナイズ．語彙 250315 トークン，入出力幅 512 トークン，入出力ベクトル 768 次元，深さ 12 層，アテンションヘッド 12 個，中間ベクトル 3072 次元．NVIDIA A100-SXM4-40GB での作成時間 57 時間 8 分.

② `https://huggingface.co/KoichiYasuoka/roberta-base-japanese-aozora-char`
青空文庫 3 億字 (元データ 2.37 億字＋異体字増量分 0.64 億字) を単文字トークナイズ．語彙 9415 トークン，入出力幅 512 トークン，入出力ベクトル 768 次元，深さ 12 層，アテンションヘッド 12 個，中間ベクトル 3072 次元．NVIDIA A100-SXM4-40GB での作成時間 19 時間 11 分.

③ `https://huggingface.co/KoichiYasuoka/roberta-large-japanese-aozora`
青空文庫 3 億字 (元データ 2.37 億字＋異体字増量分 0.64 億字) を Japanese-LUW-Tokenizer でトークナイズ．語彙 250315 トークン，入出力幅 512 トークン，入出力ベクトル 1024 次元，深さ 24 層，アテンションヘッド 16 個，中間ベクトル 4096 次元．NVIDIA A100-SXM4-40GB での作成時間 124 時間 11 分.

④ `https://huggingface.co/KoichiYasuoka/roberta-large-japanese-aozora-char`
青空文庫 3 億字 (元データ 2.37 億字＋異体字増量分 0.64 億字) を単文字トークナイズ．語彙 9415 トークン，入出力幅 512 トークン，入出力ベクトル 1024 次元，深さ 24 層，アテンションヘッド 16 個，中間ベクトル 4096 次元．NVIDIA A100-SXM4-40GB での作成時間 56 時間 57 分.

Ⓐ `https://huggingface.co/cl-tohoku/bert-base-japanese-v2`
日本語 Wikipedia 13 億字を BertJapaneseTokenizer (fugashi・unidic-lite・WordPiece) でトークナイズ．語彙 32768 トークン，入出力幅 512 トークン，入出力ベクトル 768 次元，深さ 12 層，アテンションヘッド 12 個，中間ベクトル 3072 次元．v3-8 TPU での作成時間 5 日 (らしい).

Ⓑ `https://huggingface.co/cl-tohoku/bert-base-japanese-char-v2`
日本語 Wikipedia 13 億字を単文字トークナイズ．語彙 6144 トークン，入出力幅 512 トークン，入出力ベクトル 768 次元，深さ 12 層，アテンションヘッド 12 個，中間ベクトル 3072 次元．v3-8 TPU での作成時間 5 日 (らしい).

Ⓒ `https://huggingface.co/cl-tohoku/bert-large-japanese`
日本語 Wikipedia 13 億字を BertJapaneseTokenizer (fugashi・unidic-lite・WordPiece) でトークナイズ．語彙 32768 トークン，入出力幅 512 トークン，入出力ベクトル 1024 次元，深さ 24 層，アテンションヘッド 16 個，中間ベクトル 4096 次元．v3-8 TPU での作成時間 14 日 (らしい).

Ⓓ `https://huggingface.co/cl-tohoku/bert-large-japanese-char`
日本語 Wikipedia 13 億字を単文字トークナイズ．語彙 6144 トークン，入出力幅 512 トークン，入出力ベクトル 1024 次元，深さ 24 層，アテンションヘッド 16 個，中間ベクトル 4096 次元．v3-8 TPU での作成時間 14 日 (らしい).

表 3: UD 係り受けタグ (DEPREL)

	Nominals	**Clauses**	**Modifier Words**	**Function Words**
Core arguments	nsubj 主語 obj 目的語 iobj 間接目的語	csubj 節主語 ccomp 節目的語 xcomp 節補語		
Non-core dependents	obl 斜格補語 vocative 呼称語 expl 形式語 dislocated 外置語	advcl 連用修飾節	advmod 連用修飾語 discourse 談話要素	aux 動詞補助成分 cop 繋辞 mark 標識
Nominal dependents	nmod 体言による連体修飾語 appos 同格 nummod 数量による修飾語	acl 連体修飾節	amod 用言による連体修飾語	det 決定語 clf 類別語 case 格表示

Coordination	**MWE**	**Loose**	**Special**	**Other**
conj 接続 cc 接続語	fixed 固着 flat 並列 compound 複合	list 細目 parataxis 隣接表現	orphan 親なし goeswith 泣き別れ reparandum 言い損じ	punct 句読点 root 親 dep 未定義

ID	FORM	LEMMA	UPOS	XPOS	FEATS	HEAD	DEPREL	DEPS	MISC
1	世界	世界	NOUN	名詞-普通名詞-一般	_	2	compound	_	SpaceAfter=No
2	中	中	NOUN	接尾辞-名詞的-副詞可能	_	4	nsubj	_	SpaceAfter=No
3	が	が	ADP	助詞-格助詞	_	2	case	_	SpaceAfter=No
4	刮目	刮目	VERB	名詞-普通名詞-サ変可能	_	0	root	_	SpaceAfter=No
5	し	為る	AUX	動詞-非自立可能	_	4	aux	_	SpaceAfter=No
6	て	て	SCONJ	助詞-接続助詞	_	4	mark	_	SpaceAfter=No
7	いる	居る	AUX	動詞-非自立可能	_	4	aux	_	SpaceAfter=No

(a) 国語研短単位 UD

1	世界中	世界中	NOUN	名詞-普通名詞-一般	_	3	nsubj	_	SpaceAfter=No
2	が	が	ADP	助詞-格助詞	_	1	case	_	SpaceAfter=No
3	刮目し	刮目する	VERB	動詞-一般-サ行変格	_	0	root	_	SpaceAfter=No
4	ている	て居る	SCONJ	助動詞-上一段-ア行	_	3	mark	_	SpaceAfter=No

(b) 国語研長単位 UD

図 2: 「世界中が刮目している」の CoNLL-U データ

行は各単語に対応しており，以下に示す 10 個のタブ区切りフィールドで構成される．

1. ID: 単語ごとに付与されたインデックスで，文ごとに 1 から始まる整数．縮約語に対しては，単語の範囲を示すのも可．
2. FORM: 語，または，句読記号．
3. LEMMA: 基底形，語幹．
4. UPOS: UD で規定された言語普遍な品詞タグ (表 1)．
5. XPOS: 言語固有の品詞タグ．
6. FEATS: UD で規定された言語普遍な形態素属性のリスト．言語固有の拡張も可．
7. HEAD: 当該の単語の係り受け元 ID．係り受け元が無い場合は 0 とする．
8. DEPREL: UD で規定された言語普遍な係り受けタグ (表 3)．HEAD が 0 の場合は root とする．言語固有の拡張も可．
9. DEPS: 複数の係り受け元を持つ場合，全ての HEAD:DEPREL ペア．
10. MISC: その他のアノテーション．

ID･FORM･LEMMA は，単語そのものに関するフィールドである．UPOS・XPOS・FEATS は，単語の品詞と形態素属性に関するフィールドである．HEAD・DEPREL・DEPS は，単語の係り受けに関するフィールドである．

UD における係り受け関係は，単語間の有向グラフを HEAD と DEPREL で記述する．HEAD は，その単語に入る有向枝のリンク元 ID を示しており，DEPREL は，その有向枝における係り受けタグである．ただし，HEAD が 0 の場合，その枝に入るリンク元は存在しない．リンクの本数は単語の個数に等しく，各リンクのリンク先は，全て互いに異なっている．すなわち，各単語から出るリンクは複数有り得るが，各単語に入るリンクは 1 つだけである．なお，リンクはループしない．

UD の係り受けリンクは，Мельчук 依存文法の後裔であり，いわゆる動詞中心主義である．動詞をリンク元として，主語や目的語へとリンクする．修飾関係においては，被修飾語から修飾語へとリンクする．ただし，側置詞 (前置詞や後置詞) を体言の修飾語だとみなす点 [10] が，Мельчук とは異なっている．また，コピュラ文においては動詞中心主義を採らず，補語をリンク元として，主語や繋辞へとリンクする．

なお，UD は単語長を規定しておらず，各言語ごとに，自由に単語長を決めることができる．日本語に対しては，国語研短単位・国語研長単位のいずれを用いても (図 1)，矛盾なく CoNLL-U を記述可能である (図 2)．

3 係り受け解析モデルの製作

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) は，多言語係り受け解析エンジン esupar [a)] を開発中である．esupar は，Transformers 上の言語モデルを用いて，系列ラ

[a)] https://pypi.org/project/esupar

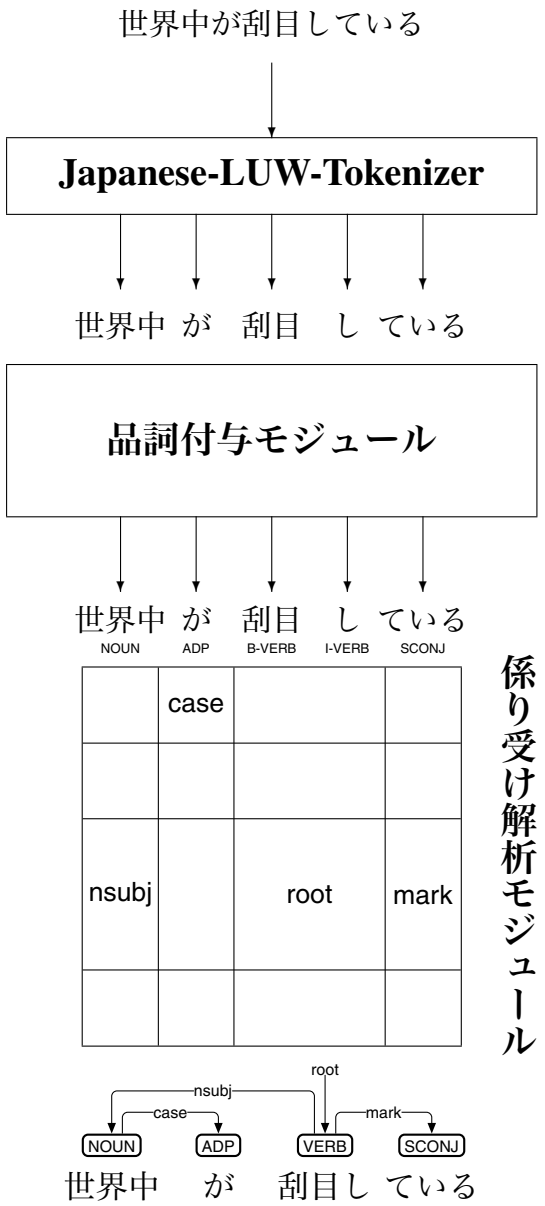


図 3: Japanese-LUW-Tokenizer を用いた場合の国語研長単位係り受け解析

ベリングによる品詞付与と，隣接行列型アルゴリズム [11] による係り受け解析をおこなう．表 2 の各モデルに対し，国語研長単位コーパス UD_Japanese-GSDLUW による学習をおこなった場合を，それぞれ考えてみよう．

モデル①・③では，Japanese-LUW-Tokenizer [b)]が日本語長単位でトークナイズをおこなう．ただし，Japanese-LUW-Tokenizer は完璧ではなく，たとえば「世界中が刮目している」という文字列に対しては，やや短めに「世界中」「が」「刮目」「し」「ている」とトークナイズする (図 3)．これを受けて品詞付与モジュールは,「世界中」に NOUN を,「が」に ADP を,「刮目し」に VERB を,「ている」に SCONJ を付与する．ただし,「刮目し」は 2 つのトークンにまたがっていることから，実際の系列ラベリングでは,「刮目」に B-VERB を,「し」に I-VERB を付与し，その後「刮目し」に組み上げる．これら 4 つの単語に対し，係り受け解析モジュールが

$$\begin{bmatrix} - & \text{case} & - & - \\ - & - & - & - \\ \text{nsubj} & - & \text{root} & \text{mark} \\ - & - & - & - \end{bmatrix}$$

という隣接行列を生成し，依存構造グラフを得る．

モデル②・④は，単文字トークナイズをおこなう．品詞付与モジュールでは，たとえば「世界中」「が」「刮目し」「ている」に対しては，単文字の単語「が」に ADP を付与できるものの，他の単語は先頭トークンに B-を，後続トークンに I-を，それぞれ UD 品詞に追加する形となる (図 4)．これらの単語を組み上げた後，係り受け解析モジュールが

$$\begin{bmatrix} - & \text{case} & - & - \\ - & - & - & - \\ \text{nsubj} & - & \text{root} & \text{mark} \\ - & - & - & - \end{bmatrix}$$

という隣接行列を生成し，依存構造グラフを得る．なお，モデルⒷ・Ⓓも同様だが，語彙に「刮」を含んでいないため,「刮」は未知語 [UNK] 扱いとなる．

モデルⒶ・Ⓒは，BertJapaneseTokenizer が国語研短単位 unidic-lite [c)]でトークナイズをおこなう．ただし，モデルⒶ・Ⓒは，語彙に「刮目」を含んでいないため，たとえば「世界中が刮目している」という文字列に対しては,「世界」「中」「が」[UNK]「し」「て」「いる」とトークナイズする (図 5)．これを受けて品詞付与モジュールは，pytokenizations [d)]の助けを借りつつ,「世界」に B-NOUN を,「中」に I-NOUN を,「が」に ADP を，[UNK] に B-VERB を,「し」に I-VERB を,「て」に B-SCONJ を,「いる」に I-SCONJ を付与する．これら 4 つの単語を組み上げた後，係り受け解析モジュールが

$$\begin{bmatrix} - & \text{case} & - & - \\ - & - & - & - \\ \text{nsubj} & - & \text{root} & \text{mark} \\ - & - & - & - \end{bmatrix}$$

という隣接行列を生成し，依存構造グラフを得る．

[b)] https://github.com/KoichiYasuoka/Japanese-LUW-Tokenizer

[c)] https://pypi.org/project/unidic-lite

[d)] https://github.com/explosion/tokenizations

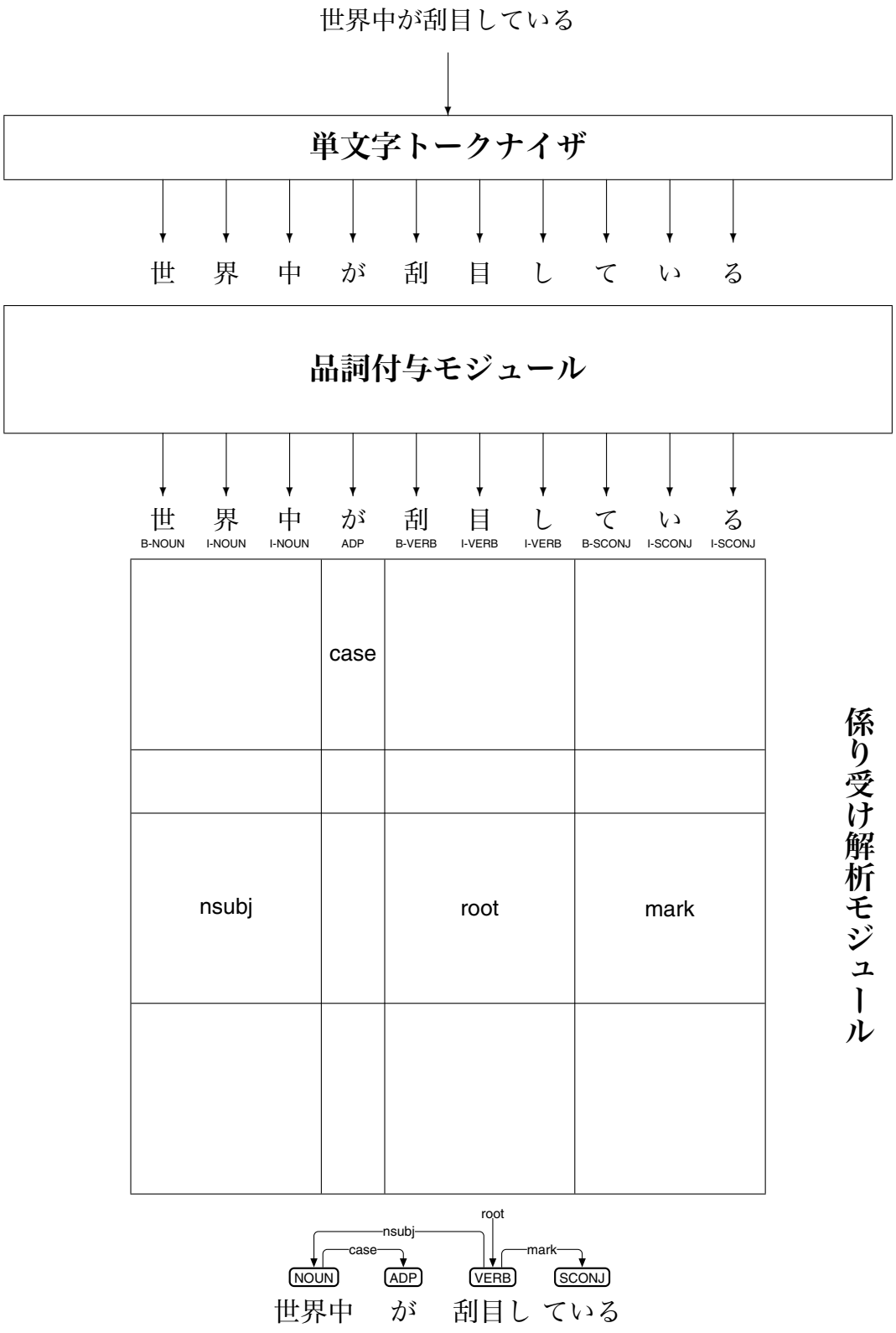


図 4: 単文字トークナイザを用いた場合の国語研長単位係り受け解析

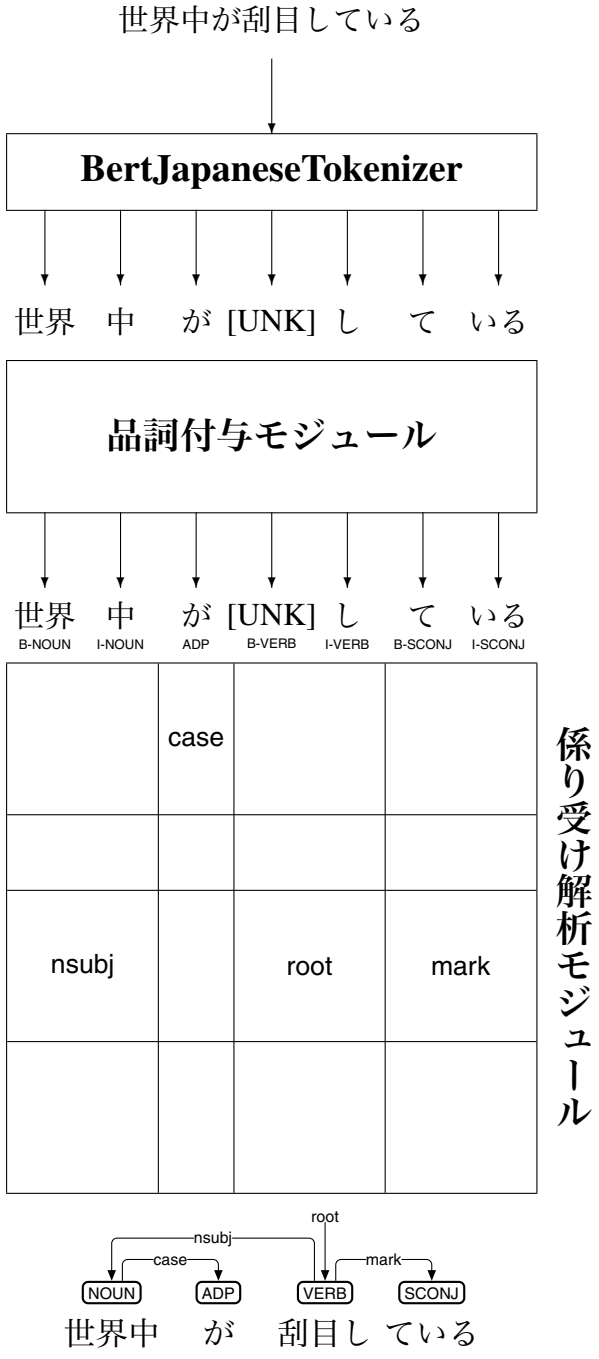


図 5: モデルⒶ・Ⓒによる国語研長単位係り受け解析

4 評価と考察

esupar 1.0.3 と表 2 の各モデルを用いて，国語研長単位係り受け解析モデルを製作した．訓練データには UD 2.9 の `ja_gsdluw-ud-train.conllu` を，評価データには `ja_gsdluw-ud-dev.conllu` を，テストデータには `ja_gsdluw-ud-test.conllu` を，それぞれ用いた．図 6 のスクリプトを用いて訓練 (係り受け解析モデル構築) に要した時間は，1 GPU (NVIDIA GeForce RTX 2080) で，各モデル 1〜3 時間程度だった．参考として，UDPipe 1.2.0 によるモデル U も製作した．モデル構築時の評価結果を表 4 に示す．評価指標は，UPOS / LAS / MLAS [e] を用いた．

表 4: モデル構築時の評価 (UPOS / LAS / MLAS)

	評価 (evaluation)	テスト (predict)
①	93.40 / 87.62 / 77.40	93.07 / 86.69 / 76.10
②	95.80 / 91.86 / 80.92	95.53 / 90.55 / 79.11
③	92.58 / 86.72 / 75.09	92.18 / 85.63 / 74.02
④	94.93 / 91.63 / 79.23	94.42 / 89.45 / 76.38
Ⓐ	96.70 / 93.16 / 84.33	96.38 / 91.77 / 81.95
Ⓑ	96.34 / 93.13 / 83.97	95.36 / 90.94 / 80.39
Ⓒ	**97.40 / 93.73 / 85.66**	**97.50 / 92.59 / 84.23**
Ⓓ	97.00 / 93.37 / 84.69	97.04 / 92.00 / 82.95
U	92.28 / 82.89 / 68.90	92.01 / 80.90 / 67.59

表 5: 共通テスト『国語』第 1 問ベンチマーク

	【文章 I】	【文章 II】
①	88.49 / 73.72 / 57.72	93.78 / 81.93 / 66.20
②	91.91 / 82.88 / 63.21	96.47 / 88.39 / 71.85
③	87.30 / 71.72 / 54.67	91.89 / 82.10 / 62.50
④	91.05 / 79.08 / 60.99	95.46 / 85.96 / 69.42
Ⓐ	93.34 / 84.06 / 69.05	97.04 / 89.10 / 72.54
Ⓑ	91.41 / 81.73 / 62.71	96.38 / **89.58 / 74.48**
Ⓒ	**94.47 / 84.94 / 69.50**	**97.40** / 88.60 / 73.79
Ⓓ	92.74 / 83.61 / 65.54	96.90 / 88.39 / 71.72
U	84.71 / 67.53 / 46.90	86.79 / 68.75 / 53.78

さらに，大学入学共通テストの令和 4 年度本試験『国語』(2022 年 1 月 15 日実施) 第 1 問の問題文を手作業で UD 化 (図 7) し，これを用いて，同様に UPOS / LAS / MLAS で評価をおこなった．評価結果を表 5 に示す．

全体の傾向としては，UPOS (品詞付与と単語組み上げ) の精度が，LAS (係り受けタグとリンク元) の精度を左右しており，結果として MLAS (UPOS と FEATS と LAS の総合性能，ただし esupar は FEATS を使用していない) に影響を与えている．当たり前と

[e] 通常は LAS (Labeled Attachment Score) / MLAS (Morphology-aware Labeled Attachment Score) / BLEX (Bi-LEXical dependency score) の 3 つの評価指標を用いる [12] が，esupar は LEMMA を使用していないため BLEX を外し，代わりに UPOS の F 値を用いた．

```
#! /bin/sh
pip3 install esupar==1.0.3 fugashi unidic-lite pytokenizations
U=UD_Japanese-GSDLUW
test -d $U || git clone -b r2.9 https://github.com/UniversalDependencies/$U
python3 -m esupar.train KoichiYasuoka/roberta-base-japanese-aozora KoichiYasuoka/roberta-base-japanese-luw-upos $U
python3 -m esupar.train KoichiYasuoka/roberta-base-japanese-aozora-char KoichiYasuoka/roberta-base-japanese-char-luw-upos $U
python3 -m esupar.train KoichiYasuoka/roberta-large-japanese-aozora KoichiYasuoka/roberta-large-japanese-luw-upos $U
python3 -m esupar.train KoichiYasuoka/roberta-large-japanese-aozora-char KoichiYasuoka/roberta-large-japanese-char-luw-upos $U
python3 -m esupar.train cl-tohoku/bert-base-japanese-v2 tmp-tohoku-luw-upos/bert-base-japanese-v2 $U
python3 -m esupar.train cl-tohoku/bert-base-japanese-char-v2 tmp-tohoku-luw-upos/bert-base-japanese-char-v2 $U
python3 -m esupar.train cl-tohoku/bert-large-japanese tmp-tohoku-luw-upos/bert-large-japanese $U
python3 -m esupar.train cl-tohoku/bert-large-japanese-char tmp-tohoku-luw-upos/bert-large-japanese-char $U
test -f udpipe-1.2.0-bin.zip || curl -LO https://github.com/ufal/udpipe/releases/download/v1.2.0/udpipe-1.2.0-bin.zip
test -d udpipe-1.2.0-bin || unzip udpipe-1.2.0-bin.zip
udpipe-1.2.0-bin/bin-linux64/udpipe --train ja-gsdluw.udpipe $U/ja_gsdluw-ud-train.conllu
```

図 6: 国語研長単位係り受け解析モデル構築用シェルスクリプト

いえば当たり前で，品詞付与モジュールが解析にしくじったら，そこから後は，もう，どうにもならない，ということだ．また，品詞付与モジュールの解析精度は，トークナイザに依存しているように見える．単文字トークナイザと BertJapaneseTokenizer に対しては，品詞付与モジュールはまずまず良い精度を出すが，Japanese-LUW-Tokenizer とは明らかに相性が悪い．一方で，係り受け解析モジュールの精度は，base・large の違いよりも，基となったテキストの量 (モデル①〜④は 3 億字，モデルⒶ〜Ⓓは 13 億字) に依るようだが，これは何らかの追試を必要とするだろう．

5 おわりに

表 2 の日本語 BERT/RoBERTa モデルを下敷きに，国語研長単位 UD_Japanese-GSDLUW による日本語係り受け解析モデルを，Transformers を用いた多言語係り受け解析エンジン esupar と共に製作した．結果としては，モデルⒸ cl-tohoku/bert-large-japanese が，最も解析性能が高くなった．ただし，UD_Japanese-GSDLUW には，アノテーション上のバグが残っており[f]，最終的な結論はバグフィクスを待つべきだ．

ちなみに，筆者の問題意識の 1 つは，近代日本語の係り受け解析 [5, 13] にある．本稿における比較評価も，今後，近代日本語の係り受け解析モデルを製作する際に，どのような戦略でおこなうべきか確認する，という側面が強い．少なくとも本稿の結果からは，トークナイザの選択が非常に重要だ，という示唆が得られている．「単語間に区切りのない書写言語」におけるトークナイザの選択を，もう少し突き詰めてみる必要があるだろう．

参考文献

[1] Mai Omura, Aya Wakasa, Masayuki Asahara: Word Delimitation Issues in UD Japanese, Proceedings of the 5th Workshop on Universal Dependencies (December 2021), pp.142-150.

[2] 浅原正幸, 金山博, 宮尾祐介, 田中貴秋, 大村舞, 村脇有吾, 松本裕治: Universal Dependencies 日本語コーパス, 自然言語処理, Vol.26, No.1 (2019 年 3 月), pp.3-36.

[3] 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.

[4] 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.

[5] 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『人文科学とコンピュータ』, No.3 (2020 年 9 月 5 日), pp.1-8.

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

[7] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, Alexander M. Rush: Transformers: State-of-the-Art Natural Language Processing, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (October 2020): System Demonstrations, pp.38-45.

[8] Lucien Tesnière: Éléments de Syntaxe Structurale, Paris: C. Klincksieck (1959).

[9] Igor A. Mel'čuk: Dependency Syntax: Theory and Practice, New York: State University of New York Press (1988).

[10] Joakim Nivre: Towards a Universal Grammar for Natural Language Processing, CICLing 2015: 16th International Conference on Intelligent Text Processing and Computational Linguistics (April 2015), pp.3-16.

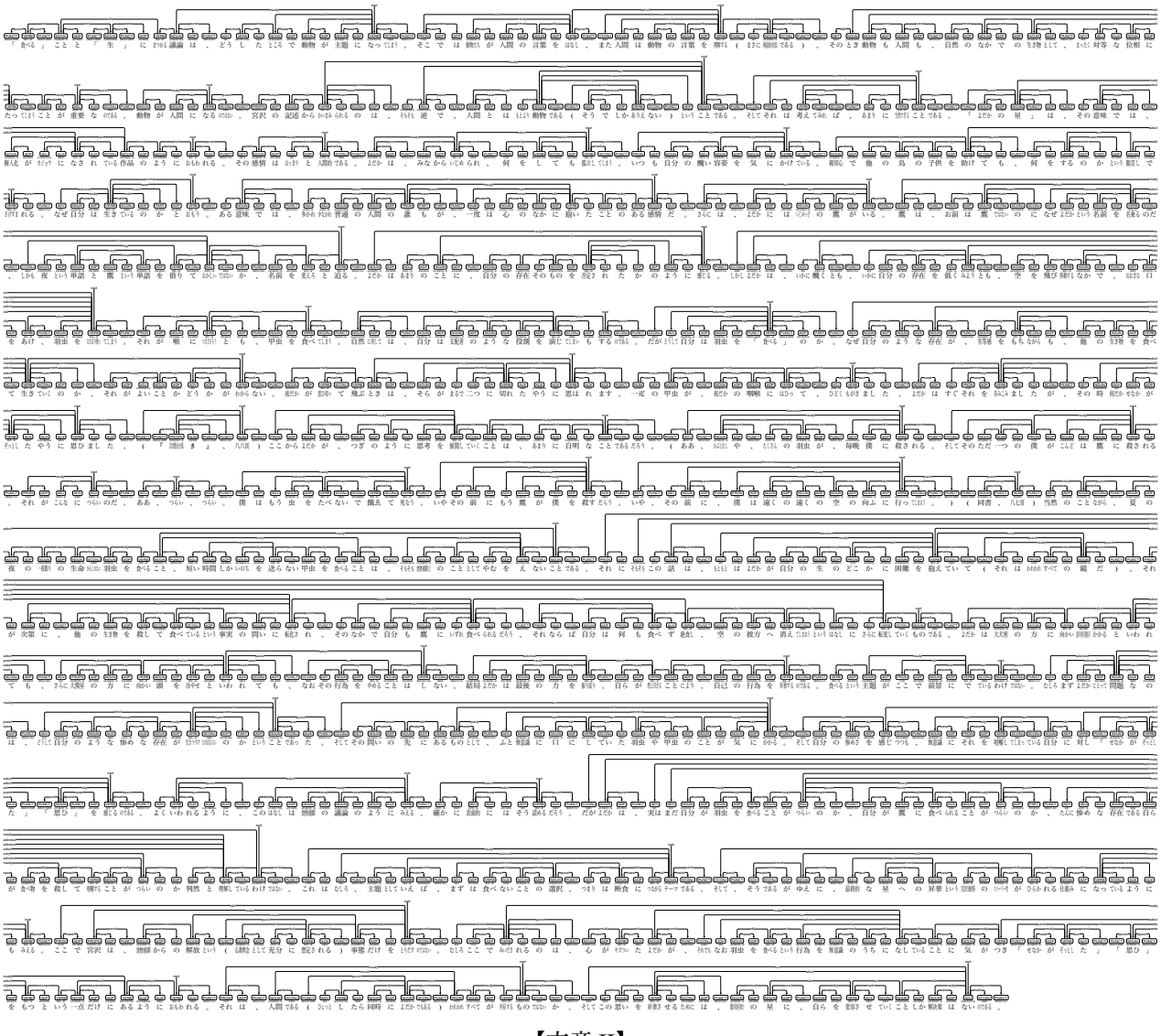
[11] Timothy Dozat, Christopher D. Manning: Deep Biaffine Attention for Neural Dependency Parsing, 5th International Conference on Learning Representations (April 2017), C25.

[12] 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.

[13] 安岡孝一: 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, 東洋学へのコンピュータ利用, 第 33 回研究セミナー (2021 年 3 月 5 日), pp.3-34.

[f] https://github.com/UniversalDependencies/UD_Japanese-GSDLUW/issues/1

【文章 I】



【文章 II】

図 7: 大学入学共通テスト『国語』(2022 年 1 月 15 日実施) 第 1 問の問題文を基にした国語研長単位 UD