

ModernBERT での上三角行列による係り受け解析

安岡孝一 (京都大学)

概要: ModernBERT 上の系列ラベリングを用いて、品詞付与と係り受け解析を同時におこなうアルゴリズムを提案する。本稿のアルゴリズムは、Universal Dependencies の有向グラフを隣接行列の形で表現し、さらに上三角行列に変換した上で、一次元の系列ラベリングへと展開する。本稿のアルゴリズムを、日本語・英語・ウクライナ語・ロシア語・ドイツ語・ポルトガル語 ModernBERT に適用したところ、いずれも良好な解析精度を得ることができたが、いずれの ModernBERT においてもトークナイザを改良する必要が生じた。

キーワード: 多言語処理、品詞付与、依存文法解析

Dependency-Parsing upon ModernBERT with Upper Triangular Adjacency Matrix

Koichi Yasuoka (Kyoto University)

Abstract: In this paper the author proposes a new sequence-labeling algorithm for part-of-speech tagging and dependency-parsing on ModernBERT. The sequence-labeling algorithm utilizes upper triangular adjacency matrices, derived from the directed acyclic graphs of Universal Dependencies. The algorithm was applied to ModernBERTs in Japanese, English, Ukrainian, Russian, German, and Portuguese, while refinements of the tokenizers were needed to increase accuracy.

Keywords: Multi-Language Processing, Part-of-Speech Tagging, Dependency-Parsing

1 はじめに

2024 年 12 月 19 日に発表された ModernBERT [1] は、入出力幅 (max position embeddings) 8192 トークンの言語モデルを 1.49 億 (149M) パラメータで実現する、という途方もないものだった。筆者が研究代表者を務める学際大規模情報基盤共同利用・共同研究拠点公募型共同研究『単語間に区切りのない書写言語における係り受け解析エンジンの開発』(共同研究者: 山崎直樹・二階堂善弘・師茂樹・鈴木慎吾・Christian Wittern・池田巧・守岡知彦・李媛・劉冠偉) では、Universal Dependencies (UD) [2] にもとづく多種多様な係り受け解析エンジンを開発してきたが、入出力幅は 512 トークンが中心であり、その 16 倍もの入出力幅は、この時点では経験がなかった。

入出力幅が広がれば、新たな解析アルゴリズムの可能性が生まれる。係り受け解析での隣接確率行列を考えると、8192 トークンもあれば 90×90 の正方行列が、そのままモデルに乗ってしまう。三角行列に圧縮できれば、 126×126 までは乗りそうである。

つまり、隣接確率行列をモデルに乗せてしまった形での解析アルゴリズムを、開発可能なのである。

われわれは、このアイデアにもとづき、隣接確率行列を上三角行列に変換した形での係り受け解析アルゴリズムを設計し、ModernBERT 上の系列ラベリングとしてファインチューニングするやり方で、各言語に対する係り受け解析モデルとして実装した。本稿では、これらの実装手法と、各言語における実装上の工夫について述べる。

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 [3] に源を発し、Мельчук の有向グラフ記述 [4] によって、一応の完成を見た手法である。その最大の

表 1: 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 繫辞 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 決定詞 clf 類別詞 case 格表示
Coordination	MWE	Loose	Special	Other
conj 接続 cc 接続詞	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

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が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的な品詞タグ^{a)}。
5. XPOS: 言語固有の品詞タグ。
6. FEATS: UD で規定された言語普遍的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。

^{a)} ADJ・ADP・ADV・AUX・CCONJ・DET・INTJ・NOUN・NUM・PART・PRON・PROPN・PUNCT・SCONJ・SYM・VERB・X の 17 種類。

8. DEPREL: UD で規定された言語普遍的な係り受けタグ (表 1)。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 [5] が、Мельчук とは異なっている。ちなみに、コピュラ文においては、補語をリンク元とし

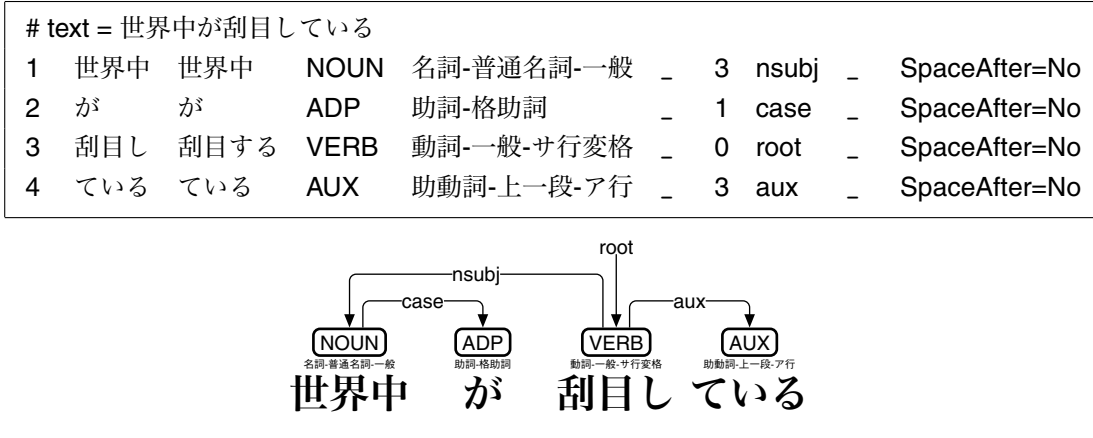


図 1: 日本語 UD (国語研長単位) の CoNLL-U と deplacy による可視化

て、主語へとリンクする。

UD の例として、「世界中が刮目している」の国語研長単位 CoNLL-U と、deplacy [6] による可視化を図 1 に示す。ただし、本稿のアルゴリズムでは、LEMMA・XPOS・DEPS は使用していない。

3 上三角行列による係り受け解析

本稿のアルゴリズムは、品詞付与と係り受け解析を同時におこなう。単語間の各リンクに対する隣接行列に、UPOS と DEPREL の両方を埋め込む形で解析をおこなう。ただし、隣接行列をそのまま用いるのではなく、上三角行列へと変換した上で、品詞付与と係り受け解析を同時におこなう。たとえば、図 1 の UD 有向グラフに対する 4×4 の隣接行列

$$\begin{bmatrix}
 & \boxed{\text{ADP}} & & \\
 & \text{case} & & \\
 \boxed{\text{NOUN}} & & \boxed{\text{VERB}} & \boxed{\text{AUX}} \\
 \text{nsubj} & & \text{root} & \text{aux} \\
 & & &
 \end{bmatrix}$$

に対し、DEPREL にリンクの方向を付加した上で、左向きリンクの各要素を転置する。

$$\begin{bmatrix}
 & \boxed{\text{ADP}} & \boxed{\text{NOUN}} & \\
 & \text{case} \rightarrow & \text{nsubj} \leftarrow & \\
 & & \boxed{\text{VERB}} & \boxed{\text{AUX}} \\
 & & \text{root} & \text{aux} \rightarrow \\
 & & &
 \end{bmatrix}$$


図 2: 上三角行列を用いた系列ラベリング

この上三角行列の各行を一次元に展開し、系列ラベリングモデル上に実装する (図 2)。なお、入力側では、各行の末尾に [SEP] トークンを挟みこんでおく。

本稿のアルゴリズムでは、UD 有向グラフのノード数 n に対し、入出力幅 $(n+1)(n+2)/2$ トークンの系列ラベリングモデルが必要^{b)}となる。なお、実際の解析においては、空行は削除可能なことから、もう少し入出力幅を小さくできると考えられる。

4 各言語 ModernBERT への適用

前章で示した品詞付与・係り受け解析アルゴリズムを、各言語 ModernBERT に適用した。そうしたところ、解析精度の向上には、各言語ごとにトークナイザを改造する必要が生じた。

4.1 日本語 ModernBERT

SB Intuitions 日本語 ModernBERT [7] のトークナイザは、Sarashina シリーズで開発^{c)}されたトークナイザを流用しており、トークン長が生成 AI 向きで、国語研長単位に合致していない。たとえば「世界中が刮目している」という文を「世界中」「が」「e5」「88」「ae」「目」「している」とトークナイズしてしまうため、「刮目し」「ている」の部分が図 1 と合致しない。国語研長単位における語境界をまたいでしまうと、どうしても解析精度が下がってしまう。

語境界をまたいでいる箇所を調べてみたところ、ほぼ全てがひらがなの前後だったので、ひらがなを単文字トークナイズすることにした。ただ、これだと今度はトークンが短くなりすぎるので、いったん仮の品詞付与を UD_Japanese-GSDLUW^{d)}の B-/I-ラベリングでおこない、各トークンを国語研長単位に合わせて組み上げた上で、そこから上三角行列による係り受け解析をおこなった。トークナイザ改造による係り受け解析精度の向上を、表 2「日本語」に示す。評価には UD_Japanese-GSDLUW の test セットを使用し、評価指標は CoNLL 2018 [8] の UPOS / LAS / MLAS^{e)}を用いた。

^{b)}入出力幅 8192 トークンの ModernBertForTokenClassification であれば、 $n \leq 126$ の上三角行列を乗せることができる。

^{c)}<https://www.sbintuitions.co.jp/blog/entry/2024/06/26/115641>

^{d)}https://github.com/UniversalDependencies/UD_Japanese-GSDLUW

^{e)}通常 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が、本稿のアルゴリズムは 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。

4.2 英語 ModernBERT

Answer.AI が発表^{f)}した英語 ModernBERT [1] に対しても、トークナイザの改造を考えた。このトークナイザは「cannot」を 1 語とみなすが、UD_English-EWT^{g)}は「can」「not」の 2 語として扱う。同様に「don't」を「do」「n't」の 2 語として、「gotta」を「got」「ta」の 2 語として、「wanna」を「wan」「na」の 2 語として扱うことから、これらの単語を狙い撃ちする形で、英語 ModernBERT のトークナイザを改造してみた。トークナイザ改造による係り受け解析精度の向上を、表 2「英語」に示す。評価には UD_English-EWT の test セットを使用した。

4.3 ウクライナ語 ModernBERT

ウクライナ語 ModernBERT [9] は、3.9 億 (390M) パラメータのモデルしか発表されておらず、しかもトークナイザが独自実装である。このトークナイザには `return_offsets_mapping=True` が実装されておらず、本稿のアルゴリズムに使えない。

このトークナイザの中身を精査したところ、Llama-Tokenizer でエミュレート可能と判断できたことから、UD_Ukrainian-IU^{h)}の各例文に対するトークナイズが維持されるよう、LlamaTokenizerFast による再実装をおこなった。また、再実装後のトークナイザを用いて、1.52 億 (152M) パラメータの ModernBERT を試作した。上三角行列による係り受け解析の精度を、表 2「ウクライナ語」に示す。評価には UD_Ukrainian-IU の test セットを使用した。

4.4 ロシア語 ModernBERT

deep vk が発表ⁱ⁾したロシア語 ModernBERT のトークナイザは、ピリオドやコンマを、しばしば直前の単語と一緒に組み上げてしまう。これらを切り離すべく、トークナイザを改造した。トークナイザ改造による係り受け解析精度の向上を、表 2「ロシア語」に示す。評価には UD_Russian-Taiga^{j)}の test セットを使用した。

なお、deep vk はロシア語 ModernBERT の「patched-tokenizer revision」をリリースしたが、この revision

^{f)}<https://huggingface.co/blog/modernbert>

^{g)}https://github.com/UniversalDependencies/UD_English-EWT

^{h)}https://github.com/UniversalDependencies/UD_Ukrainian-IU

ⁱ⁾<https://habr.com/ru/companies/vk/articles/930382/>

^{j)}https://github.com/UniversalDependencies/UD_Russian-Taiga

表 2: 上三角行列による係り受け解析の評価 (UPOS / LAS / MLAS)

		トークナイザ改造前	トークナイザ改造後
日本語	sbintuitions/modernbert-ja-30m	48.66 / 22.38 / 12.94	95.88 / 89.20 / 79.89
	sbintuitions/modernbert-ja-70m	48.82 / 22.74 / 13.57	96.16 / 89.13 / 80.18
	sbintuitions/modernbert-ja-130m	48.79 / 22.58 / 13.48	96.57 / 89.57 / 81.47
	sbintuitions/modernbert-ja-310m	48.88 / 22.68 / 13.89	96.62 / 90.38 / 82.62
英語	answerdotai/ModernBERT-base (149M)	95.24 / 88.73 / 80.91	95.97 / 89.52 / 82.04
	answerdotai/ModernBERT-large (395M)	95.38 / 88.75 / 81.22	96.11 / 89.52 / 82.35
ウクライナ語	Goader/modern-liberta-large (390M)	-	96.95 / 87.71 / 77.52
	KoichiYasuoka/modernbert-base-ukrainian (152M)	-	95.51 / 79.85 / 69.26
ロシア語	deepvk/RuModernBERT-small (35M)	93.43 / 78.19 / 72.33	94.69 / 78.96 / 72.58
	deepvk/RuModernBERT-base (150M)	94.16 / 79.20 / 75.61	95.35 / 80.36 / 75.94
ドイツ語	LSX-UniWue/ModernGBERT_134M	95.78 / 93.53 / 82.18	98.27 / 96.01 / 84.72
	LSX-UniWue/ModernGBERT_1B	94.96 / 91.12 / 78.93	97.44 / 93.63 / 81.42
ポルトガル語	eliasjacob/ModernBERT-base-portuguese (142M)	84.50 / 75.48 / 60.91	95.91 / 85.88 / 70.80
	eliasjacob/ModernBERT-large-portuguese (377M)	84.49 / 75.39 / 60.51	97.04 / 88.17 / 72.38

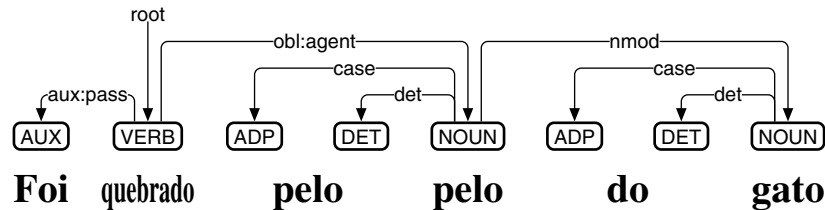


図 3: ポルトガル語における縮約冠詞「pelo」「do」と名詞「pelo」

は各トークンとトークナイザが不整合を起こしており、正常に動作させることが筆者には困難だった。

4.5 ドイツ語 ModernBERT

ヴュルツブルク大学 Institut für Informatik が発表^{k)}したドイツ語 ModernBERT に対しても、トークナイザの改造を考えた。このトークナイザは、縮約冠詞「im」を 1 語とみなすが、UD_German-HDT^{l)}は「in」「dem」の 2 語として扱う。他の縮約冠詞も同様である。そこで、これらの縮約冠詞を狙い撃ちする形で、ドイツ語 ModernBERT のトークナイザを改造してみた。トークナイザ改造による係り受け解析精度の向上を、表 2「ドイツ語」に示す。評価には UD_German-HDT の test セットを使用した。

^{k)}<https://www.informatik.uni-wuerzburg.de/datascience/news/sing1e/news/modernbert>

^{l)}https://github.com/UniversalDependencies/UD_German-HDT

4.6 ポルトガル語 ModernBERT

Elias Jacob de Menezes Neto^{m)}が発表したポルトガル語 ModernBERT に対しても、トークナイザの改造を考えた。このトークナイザは、縮約冠詞「do」を 1 語とみなすが、UD_Portuguese-Bosqueⁿ⁾は「de」「o」の 2 語として扱う。他の縮約冠詞も同様だが、縮約冠詞の中には「pelo」のように名詞と同型のものがあり、UD_Portuguese-Bosque は縮約冠詞「pelo」を「por」「o」の 2 語として扱うが、名詞「pelo」は 1 語のまま扱う (図 3)。なかなか手強い。

この扱いを実現するため、いったん仮の品詞付与を UD_Portuguese-Bosque のラベリングでおこない、付与された品詞が ADP か DET となった縮約冠詞だけを 2 語に分解してから、あらためて上三角行列による品詞付与と係り受け解析をおこなう、という改造に挑戦した。この改造による係り受け解析精度

^{m)}<https://docente.ufrn.br/201900343101/perfil>

ⁿ⁾https://github.com/UniversalDependencies/UD_Portuguese-Bosque

の向上を、表 2「ポルトガル語」に示す。評価には UD.Portuguese-Bosque の test セットを使用した。

5 おわりに

本稿では、ModernBERT での上三角行列による品詞付与・係り受け解析アルゴリズムを提案した。本稿のアルゴリズムを、日本語・英語・ウクライナ語・ロシア語・ドイツ語・ポルトガル語 ModernBERT に適用したところ、いずれも良好な解析精度を得ることができた。ただし、いずれの ModernBERT においても、トークナイザを改良する必要性が生じた。

これらの改良は、各言語ごとにバラバラの手法を用いている。したがって、複数の言語を同時に扱うような ModernBERT を設計する場合、改良がバッチェンギしてしまう可能性が高い。われわれとしては、多言語処理 (特に古典中国語と上代日本語の混在) に挑戦したいのだが、なかなか道は険しそうである。なお、本稿で作成した係り受け解析モデルは、全て

<https://huggingface.co/KoichiYasuoka> で公開している。ぜひ使ってみてほしい。

参考文献

- [1] Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, Iacopo Poli: Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference, Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (July 2025), Vol.1: Long Papers, pp.2526-2547.
- [2] Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, Daniel Zeman: Universal Dependencies, Computational Linguistics, Vol.47, No.2 (June 2021), pp.255-308.
- [3] Lucien Tesnière: Éléments de Syntaxe Structurale, Paris: C. Klincksieck (1959).
- [4] Igor A. Mel'čuk: Dependency Syntax: Theory and Practice, New York: State University of New York Press (1988).
- [5] 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.
- [6] 安岡孝一: Universal Dependencies にもとづく多言語係り受け可視化ツール deplacy, 人文科学とコンピュータシンポジウム「じんもんこん 2020」論文集 (2020 年 12 月), pp.95-100.
- [7] 塚越駿, 李聖哲, 福地成彦, 柴田知秀: 日本語 ModernBERT の構築, JLR2025『日本語言語資源の構築と利用』(2025 年 3 月 14 日).
- [8] 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.
- [9] Mykola Haliuk, Aleksander Smywiński-Pohl: On the Path to Make Ukrainian a High-Resource Language, UNLP 2025: The 4th Ukrainian Natural Language Processing Workshop (July 2025), pp.120-130.