青空文庫 ModernBERT モデルによる国語研長単位係り受け解析

安岡孝一 (京都大学)

概要: 2024 年 12 月に発表された ModernBERT は，入出力幅 8192 トークンを，1.5 億パラメータのモデルで実現している．これまで BERT や DeBERTa の 1.5 億パラメータ・モデルは，入出力幅が 512 トークン程度だったことに較べれば，格段の進歩である．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，開発可能だということである．そのようなアルゴリズムを乗せた日本語 ModernBERT は，本当に実現可能なのか．本稿では，その可能性を探る．

キーワード: 言語処理，品詞付与，係り受け解析，依存文法解析

1 はじめに

2024 年 12 月 19 日に Answer.AI から発表された ModernBERT [a]は，入出力幅 8192 トークンの言語モデルを 1.5 億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) では，これまでに多種多様な係り受け解析エンジンを開発してきた [1, 2, 3] が，入出力幅は 512 トークンが中心であり，その 16 倍もの入出力幅は経験がない．入出力幅が拡がれば，新たな解析アルゴリズムの可能性が生まれる．しかし，発表された ModernBERT は英語のみであり，他の言語はサポートしていない．

本稿では，青空文庫 ModernBERT モデルの開発をおこないつつ，国語研長単位 Universal Dependencies [4] を題材に，ModernBERT での係り受け解析アルゴリズムの可能性を探る．

2 Universal Dependencies の概要

Universal Dependencies (UD) [5] 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 [6] に源を発し，Мельчук の有向グラフ記述 [7] 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[b]．
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 1)．HEAD が 0 の場合は root とする．言語固有の拡張も可．

[a] https://huggingface.co/blog/modernbert

[b] ADJ・ADP・ADV・AUX・CCONJ・DET・INTJ・NOUN・NUM・PART・PRON・PROPN・PUNCT・SCONJ・SYM・VERB・X の 17 種類．

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 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

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 [8] 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て，主語へとリンクする．

国語研長単位 UD の例として,「世界中が刮目している」の CoNLL-U と，deplacy [9] 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．

```
# text = 世界中が刮目している
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   ている   AUX    助動詞-上一段-ア行   _   3   aux     _   SpaceAfter=No
```

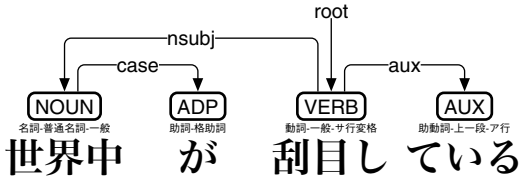


図 1: 国語研長単位 UD の CoNLL-U と deplacy による可視化

3　アルゴリズムの概要

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．正方行列を用いた解析アルゴリズムでは，たとえば図 1 の UD 有向グラフに対して，以下に示す 4×4 の正方隣接行列を用いる．

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

この正方隣接行列の各列を一次元に展開し，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，各列の末尾に [SEP] トークンを挟みこんでおく．出力側の各リンクについては，リンク元とリンク先がわかるようにラベリングをおこなう．正方行列を用いた解析アルゴリズムでは，UD 有向グラフのノード数 n に対し，入出力幅 $n(n+1)+1$ トークンの系列ラベリングモデルが必要[c]となる．

次に，上記の正方隣接行列を，上三角行列へと変換する．具体的には，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．

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

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 3)．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．上三角行列を用いた解析アルゴリズムでは，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[d]となる．

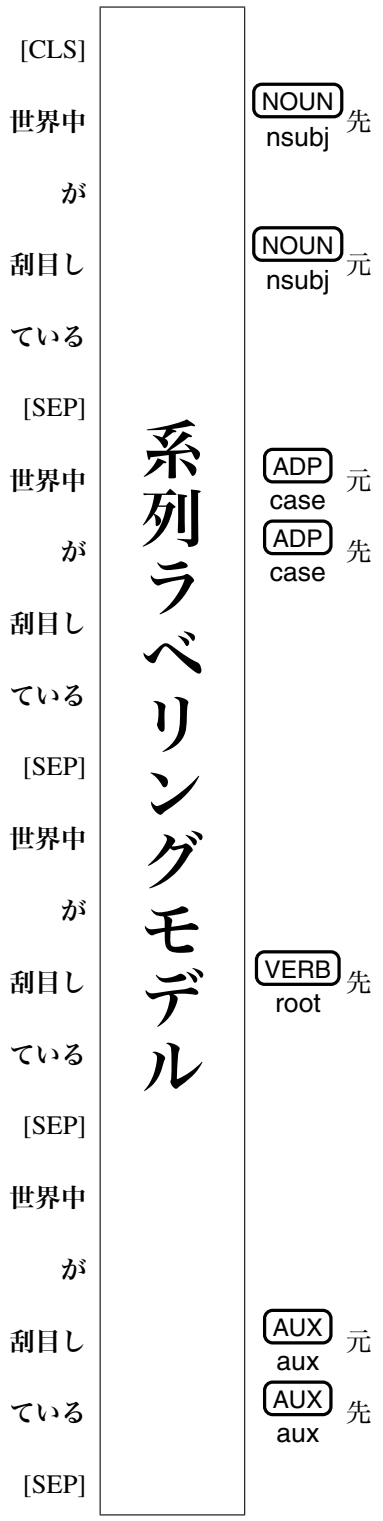


図 2: 正方行列を用いた系列ラベリング

[c] 入出力幅 8192 トークンの ModernBertForTokenClassification であれば，$n \leq 90$ の正方行列を乗せることができる．

[d] 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．

図 3: 上三角行列を用いた系列ラベリング

4 評価と考察

表 2 の青空文庫 ModernBERT モデル 3 種類[e]を，mdx 上の Transformers 4.47.1 に ModernBERT モジュールを追加する形で製作した．

これら 3 種類の ModernBERT モデルに対し，前節の 2 つのアルゴリズムによるファインチューニングを国語研長単位 UD_Japanese-GSDLUW でおこない，評価 (ja_gsdluw-ud-dev.conllu による evaluation)・テスト (ja_gsdluw-ud-test.conllu による predict) をおこなった．また，[1, 2] で用いた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 問の問題文も，評価に加えた．評価指標は，CoNLL 2018 [10] の UPOS / LAS / MLAS [f]を用いた．評価結果を，表 4・5 に示す．ただし，複数トークンの組み上げで単語ベクトルを作り出す手法 [11] の ModernBERT 実装 (inputs_embeds) が，2025 年 1 月 9 日まで遅れた[g]ことから，本稿ではリンクに goeswith を加える手法 [12](図 4) で，実装をおこなっている．

また，表 3 の青空文庫 DeBERTa モデル [2] と比較するため，esupar 1.7.7 の Biaffine アルゴリズム [13] を用いて，係り受け解析エンジンを製作した．評価結果を，表 6 に示す．ただし，ModernBERT の入出力幅 8192 トークン全ては使わず，500×500 の隣接確率行列を，500 次元ベクトルの組み合わせで実装している．

結果から言えば，正方行列アルゴリズムより上三角行列アルゴリズムの方が精度が高いものの，それでも従来法 (Biaffine) には追いついていない．正方行列アルゴリズムでは，90×90 を超える行列に対しては，入出力を ModernBERT のスライド窓 (sliding window) 幅分ずらしつつ処理する，という機構を援用したのだが，ここのチューニングがうまくいっていないようだ．上三角行列アルゴリズムも同様で，126×126 を超える行列に対しては，解析精度が下がってしまう．なかなか難しい．

[e]合わせて modernbert-large-japanese-wikipedia にあたるモデルも製作したが，100 時間かけても loss 値が収束せず，パラメータを変更して再製作する必要が生じたため，本稿での評価には含めていない．

[f]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．

[g]https://github.com/huggingface/transformers/pull/35373

表 2: 本稿で製作した ModernBERT モデルとその諸元

- https://huggingface.co/KoichiYasuoka/modernbert-base-japanese-aozora
  ModernBertForMaskedLM，1.49 億パラメータ．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) を最大長 4 文字で Unigram トークナイズ，語彙 65000 トークン，入出力幅 8192 トークン，入出力ベクトル 768 次元，深さ 22 層，アテンションヘッド 12 個，中間ベクトル 1152 次元，NVIDIA A100-SXM4-40GB × 8 台での作成時間 5 時間 55 分．

- https://huggingface.co/KoichiYasuoka/modernbert-large-japanese-aozora
  ModernBertForMaskedLM，3.95 億パラメータ．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) を最大長 4 文字で Unigram トークナイズ，語彙 65000 トークン，入出力幅 8192 トークン，入出力ベクトル 1024 次元，深さ 28 層，アテンションヘッド 16 個，中間ベクトル 2624 次元，NVIDIA A100-SXM4-40GB × 8 台での作成時間 10 時間 5 分．

- https://huggingface.co/KoichiYasuoka/modernbert-base-japanese-wikipedia
  ModernBertForMaskedLM，1.49 億バラメータ．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) と日本語 Wikipedia 26 億字を，最大長 4 文字で Unigram トークナイズ，語彙 65000 トークン，入出力幅 8192 トークン，入出力ベクトル 768 次元，深さ 22 層，アテンションヘッド 12 個，中間ベクトル 1152 次元，NVIDIA A100-SXM4-40GB × 8 台での作成時間 56 時間 49 分．

表 3: 本稿で比較対象にした DeBERTa モデルとその諸元

- https://huggingface.co/KoichiYasuoka/deberta-base-japanese-aozora
  DeBERTaV2ForMaskedLM，1.19 億パラメータ．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) を最大長 8 文字で Unigram トークナイズ，語彙 32000 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 55 分．

- https://huggingface.co/KoichiYasuoka/deberta-large-japanese-aozora
  DeBERTaV2ForMaskedLM，3.69 億パラメータ．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) を最大長 8 文字で Unigram トークナイズ，語彙 32000 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 54 時間 28 分．

- https://huggingface.co/KoichiYasuoka/deberta-base-japanese-wikipedia
  DeBERTaV2ForMaskedLM，1.19 億パラメータ．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) と日本語 Wikipedia 13 億字を最大長 8 文字で Unigram トークナイズ，語彙 32000 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 87 時間 44 分．

- https://huggingface.co/KoichiYasuoka/deberta-large-japanese-wikipedia
  DeBERTaV2ForMaskedLM，3.69 億パラメータ．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) と日本語 Wikipedia 13 億字を最大長 8 文字で Unigram トークナイズ，語彙 32000 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 254 時間 51 分．

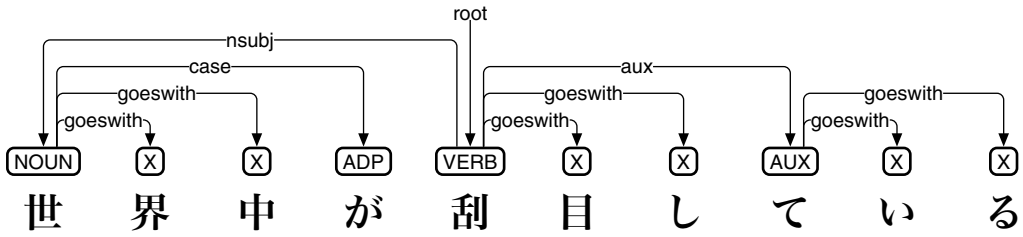


図 4: goeswith による図 1 の変形 (単文字トークナイザの場合)

表 4: 正方行列アルゴリズムによる国語研長単位係り受け解析の評価 (UPOS / LAS / MLAS)

	評価 (evaluation)	テスト (predict)	第 1 問【文章 I】	第 1 問【文章 II】
modernbert-base-japanese-aozora	93.45 / 82.11 / 68.06	92.35 / 79.94 / 66.65	84.32 / 62.95 / 42.26	84.73 / 62.61 / 45.77
modernbert-large-japanese-aozora	93.91 / 82.73 / 69.40	93.12 / 80.78 / 68.03	85.20 / 63.34 / 43.74	84.84 / 62.98 / 45.70
modernbert-base-japanese-wikipedia	95.12 / 85.16 / 73.63	94.72 / 83.58 / 72.33	86.44 / 66.91 / 48.32	90.49 / 66.81 / 51.84

表 5: 上三角行列アルゴリズムによる国語研長単位係り受け解析の評価 (UPOS / LAS / MLAS)

	評価 (evaluation)	テスト (predict)	第 1 問【文章 I】	第 1 問【文章 II】
modernbert-base-japanese-aozora	94.82 / 87.33 / 74.98	94.25 / 85.69 / 73.58	86.67 / 71.22 / 50.00	93.18 / 80.55 / 62.22
modernbert-large-japanese-aozora	94.24 / 86.73 / 73.83	94.57 / 86.09 / 74.14	87.21 / 71.72 / 50.99	93.05 / 80.29 / 63.61
modernbert-base-japanese-wikipedia	96.39 / 90.57 / 80.69	95.98 / 89.00 / 79.34	89.66 / 74.76 / 54.02	94.11 / 83.91 / 68.28

表 6: 従来法 (Biaffine) による国語研長単位係り受け解析の評価 (UPOS / LAS / MLAS)

	評価 (evaluation)	テスト (predict)	第 1 問【文章 I】	第 1 問【文章 II】
modernbert-base-japanese-aozora	95.65 / 90.31 / 80.37	95.36 / 89.14 / 79.85	90.45 / 77.05 / 59.86	94.54 / 84.77 / 68.95
modernbert-large-japanese-aozora	95.99 / 90.63 / 81.30	95.95 / 89.93 / 80.62	91.19 / 77.94 / 60.24	95.48 / 84.73 / 70.89
modernbert-base-japanese-wikipedia	97.13 / 92.62 / 85.07	96.83 / 91.21 / 82.80	92.06 / 79.04 / 62.18	96.20 / 85.00 / 70.09
deberta-base-japanese-aozora	96.19 / 91.68 / 82.09	96.05 / 90.33 / 81.00	92.09 / 81.18 / 60.96	96.32 / 87.09 / 71.03
deberta-large-japanese-aozora	96.42 / 91.89 / 82.97	96.30 / 90.86 / 82.16	93.08 / 82.70 / 64.02	96.69 / 87.34 / 72.04
deberta-base-japanese-wikipedia	95.38 / 90.19 / 80.43	94.99 / 89.16 / 79.39	90.40 / 77.36 / 59.29	95.26 / 84.77 / 68.49
deberta-large-japanese-wikipedia	97.17 / 92.38 / 84.58	97.07 / 91.29 / 83.58	93.31 / 82.95 / 63.43	96.18 / 85.96 / 70.57

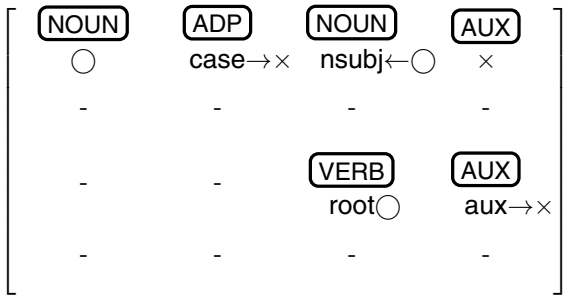
表 7: 空行を間引いた上三角行列アルゴリズムによる国語研長単位係り受け解析の評価 (暫定値)

	評価 (evaluation)	テスト (predict)	第 1 問【文章 I】	第 1 問【文章 II】
modernbert-base-japanese-aozora	94.67 / 87.58 / 75.15	94.26 / 85.65 / 73.07	87.20 / 73.11 / 50.49	91.98 / 79.66 / 60.65
modernbert-large-japanese-aozora	94.58 / 87.36 / 74.66	94.45 / 86.03 / 73.92	87.11 / 73.31 / 52.41	90.92 / 80.20 / 61.46
modernbert-base-japanese-wikipedia	96.62 / 90.58 / 80.89	96.07 / 88.96 / 79.37	89.80 / 76.22 / 55.47	94.90 / 84.85 / 72.04

5 おわりに

青空文庫 ModernBERT モデルを製作しつつ，正方行列アルゴリズムと上三角行列アルゴリズムによる国語研長単位係り受け解析に挑戦した．現時点では，これらのアルゴリズムは，従来法 (Biaffine) に解析精度が追いついておらず，ModernBERT の入出力幅 8192 トークンを活かし切れていない．

筆者は現在，上三角行列アルゴリズムの改良をおこなっている．アイデアの一つとしては，上三角行列の空行を間引くことで，126×126 を超える上三角行列を 8192 トークンに押し込めたい．そのためには，空行に対応する列のラベルに，情報 (以下の行列では「×」で表している) を埋め込む必要がありそうだ．この改良 (暫定値を表 7 に示す) がうまく行くのか，あるいは他の改良はないのか，当日の発表に期待されたい．



参考文献

[1] 安岡孝一: 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, 情報処理学会研究報告, Vol.2022-CH-128『人文科学とコンピュータ』, No.7 (2022 年 2 月 19 日), pp.1-8.

[2] 安岡孝一: 青空文庫 DeBERTa モデルによる国語研長単位係り受け解析, 東洋学へのコンピュータ利用, 第 35 回研究セミナー (2022 年 7 月 29 日), pp.29-43.

[3] 安岡孝一: GPT 系言語モデルによる国語研長単位係り受け解析, 人文科学とコンピュータシンポジウム「じんもんこん 2024」論文集 (2024 年 12 月), pp.83-90.

[4] 大村舞, 若狭絢, 浅原正幸: 国語研長単位に基づく日本語 Universal Dependencies, 自然言語処理, Vol.30, No.1 (2023 年 3 月), pp.4-29.

[5] Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, Daniel Zeman: Universal Dependencies, Computational Linguistics, Vol.47, No.2 (June 2021), pp.255-308.

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

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

[8] 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.

[9] 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.

[10] 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.

[11] Adam Ek, Jean-Philippe Bernardy: Composing Byte-Pair Encodings for Morphological Sequence Classification, UDW 2020: 4th Workshop on Universal Dependencies (December 2020), pp.76-86.

[12] 安岡孝一: 世界の Universal Dependencies と係り受け解析ツール群, 第 3 回 Universal Dependencies 公開研究会 (2021 年 6 月 22 日).

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