Universal Dependencies にもとづく多言語係り受け可視化ツール deplacy

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)

Universal Dependencies は，カレル大学の LINDAT/CLARIN を中心に製作中の多言語係り受けコーパスであり，現在，90 の言語に及んでいる．Universal Dependencies を用いた係り受け解析ツールや，係り受け可視化ツールも数多く製作されており，グラフィカルな画面に解析結果が出力されるようになってきている．ただ，係り受け解析作業そのものは，CUI 上で python などのスクリプト言語を用いておこなうのが主であり，できれば解析結果も CUI 上で見たい．

そのような要求に答えるべく，可視化ツール deplacy を製作した．deplacy は，Universal Dependencies にもとづく係り受け有向グラフを，CUI 上に表示する python3 モジュールである．さらに，50 以上の書写言語に deplacy を適用し，各言語用のデモページを Google Colaboratory 上に製作した．https://koichiyasuoka.github.io/deplacy/ で公開中である．

deplacy: a CUI-based tree visualizer for Universal Dependencies

Koichi Yasuoka (Kyoto University)

Universal Dependencies (UD) is a framework for annotation of parts-of-speech (POS) and syntactic-dependencies across different human languages. So many tools have been developed for UD: tokenizers, POS-taggers, dependency parsers, graphical visualizers of dependency graphs, etc. In this session the author shows **deplacy**, a CUI-based tree visualizer for UD across 50+ languages, and demonstrates **deplacy** on Google Colaboratory.

1 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: ウィッテルン クリスティアン，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，古典中国語の依存文法解析に精力を傾注しており，道具立ての一つとして，Universal Dependencies (以下「UD」)[1] の古典中国語への適用を研究している [2]．依存文法解析それ自体は，Tesnière [3] の構造的統語論に源を発し，Мельчук [4] の有向グラフ記述によって，一応の完成を見た手法である．その最大の特長は，言語横断的な記述が可能という点にあり，Мельчук の手法をコンピュータ向けに洗練した UD においても，言語に関わらず単語間の係り受けを記述可能である．

ただ，UD の CoNLL-U フォーマットは，機械可読であることを主眼としており，人間にはすこぶる読みにくい (図 1)．UD にもとづく係り受け解析結果を人間が読むには，有向グラフの可視化が必要となる．可視化は，グラフィック環境でおこなうのが常道だが，CUI でリモート環境から解析する場合には，グラフィック環境がなくても可視化できる機能が欲しい．

これまでにわれわれは，古典中国語 UD をグラフィック環境で可視化するツールを製作し，UD の編集や印刷に用いてきた [5]．本稿では，これにテキスト環境での可視化を加え，古典中国語から切り離して多言語化した python3 モジュール deplacy を製作した．さらに，古典中国語のみならず，50 以上の書写言語用のデモページを，Google Colaboratory 上に製作した．以下では deplacy について述べる．

```
1	不	不	ADV	v,副詞,否定,無界	Polarity=Neg	2	advmod	_	Gloss=not|SpaceAfter=No
2	入	入	VERB	v,動詞,行為,移動	_	0	root	_	Gloss=enter|SpaceAfter=No
3	虎	虎	NOUN	n,名詞,主体,動物	_	4	nmod	_	Gloss=tiger|SpaceAfter=No
4	穴	穴	NOUN	n,名詞,固定物,地形	Case=Loc	2	obj	_	Gloss=cave|SpaceAfter=No
5	不	不	ADV	v,副詞,否定,無界	Polarity=Neg	6	advmod	_	Gloss=not|SpaceAfter=No
6	得	得	VERB	v,動詞,行為,得失	_	2	parataxis	_	Gloss=get|SpaceAfter=No
7	虎	虎	NOUN	n,名詞,主体,動物	_	8	nmod	_	Gloss=tiger|SpaceAfter=No
8	子	子	NOUN	n,名詞,人,関係	_	6	obj	_	Gloss=child|SpaceAfter=No
```

図 1: CoNLL-U フォーマットによる「不入虎穴不得虎子」の係り受け解析結果

2 係り受け可視化ツール deplacy

deplacy は，UD の CoNLL-U フォーマットで表された係り受け有向グラフに対し，

- 罫線素片を用いて，テキスト環境での係り受け可視化
- SVG (Scalable Vector Graphics) を用いて，グラフィック環境での係り受け可視化
- グラフ記述言語 DOT を用いて，グラフィック環境での直接構成鎖解析 [6] の可視化

をおこなう python3 モジュールである．ただ，グラフィック環境において可視化をおこなうツールは，世の中に数多く製作されている．その意味で deplacy の特徴は，あくまでテキスト環境での係り受け可視化にあると，われわれは考えている．

2.1 テキスト環境での係り受け可視化

`deplacy.render` は，係り受け有向グラフを罫線素片で表現する (図 2)．古典中国語を縦書き表示する，というアイデアを基本に，各単語の右側に品詞や罫線素片を表示することで，テキスト環境において係り受け可視化をおこなう．

```
>>> import udkanbun
>>> nlp=udkanbun.load()
>>> doc=nlp("不入虎穴不得虎子")
>>> import deplacy
>>> deplacy.render(doc)
不 ADV  <┐      advmod
入 VERB ┐└┐     root
虎 NOUN <┐ |     nmod
穴 NOUN ┘<┘     obj
不 ADV  <┐      advmod
得 VERB ┐└┐     parataxis
虎 NOUN <┐ |     nmod
子 NOUN ┘<┘     obj
```

図 2: テキスト環境での係り受け可視化

なお，図 2 の出力結果は，図 3 と図 4 を同時に表現することを意図している．図 2 の単語間の矢印は，図 3 の有向グラフと同型である．また，図 2 を右から見た際の 2 分木 (矢印の方向を無視) は，図 4 の構文木と同型である．

2.2 SVG による係り受け可視化

`deplacy.serve` は，係り受け有向グラフを SVG で表現する (図 3)．http://localhost:5000 にブラウザ

```
>>> import udkanbun
>>> nlp=udkanbun.load()
>>> doc=nlp("不入虎穴不得虎子")
>>> import deplacy
>>> deplacy.serve(doc)
```

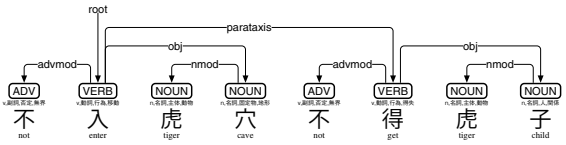

図 3: SVG による係り受け可視化

を接続することで，当該 SVG が表示される．この SVG は，簡易版のビジュアル・エディタとなっており，リンク元やタグを編集可能である．なお，Google Colaboratory 上で `port=None` オプションを使用した場合は，ポートへの接続なしに SVG が表示される．

2.3 DOT による直接構成鎖解析の可視化

`deplacy.dot` は，UD に対して直接構成鎖解析をおこない，結果の構文木を DOT 形式で返す．Google Colaboratory では，`graphviz.source` の引数とすることで，構文木を直接表示できる (図 4)．

```
>>> import udkanbun
>>> nlp=udkanbun.load()
>>> doc=nlp("不入虎穴不得虎子")
>>> import deplacy,graphviz
>>> graphviz.source(deplacy.dot(doc))
```

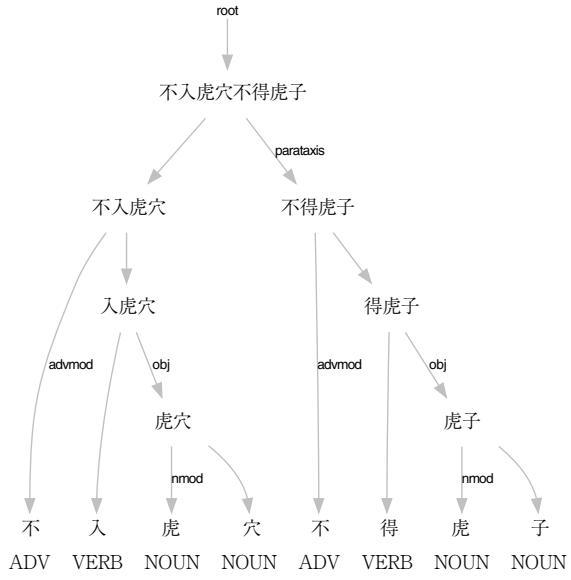

図 4: DOT による直接構成鎖解析の可視化

表 1: deplacy に接続可能な係り受け解析エンジン (2020 年 11 月 9 日現在)

アフリカーンス語	Camphr-Udify, spaCy-COMBO, UDPipe 2, Turku-neural-parser-pipeline, NLP-Cube など
アラビア語	UDPipe 2, Turku-neural-parser-pipeline, spacy-udpipe, NLP-Cube, spaCy-COMBO など
ブルガリア語	Camphr-Udify, UDPipe 2, CLASSLA, NLP-Cube, Stanza, spaCy-COMBO など
カタルーニャ語	Camphr-Udify, Stanza, NLP-Cube, spacy-udpipe, UDPipe 2, spaCy-COMBO など
コプト語	spaCy-Coptic, spacy-udpipe, UDPipe 2, Stanza
チェコ語	Camphr-Udify, UDPipe 2, spacy-udpipe, NLP-Cube, Stanza, spaCy-COMBO など
ウェールズ語	UDPipe 2, Camphr-Udify
デンマーク語	Camphr-Udify, UDPipe 2, Stanza, Turku-neural-parser-pipeline, spaCy, NLP-Cube など
ドイツ語	Camphr-Udify, Stanza, UDPipe 2, Turku-neural-parser-pipeline, NLP-Cube など
ギリシア語	Stanza, UDPipe 2, spacy-udpipe, Turku-neural-parser-pipeline, Camphr-Udify, spaCy など
英語	Camphr-Udify, Stanza, UDPipe 2, NLP-Cube, spaCy-COMBO, spacy-udpipe など
スペイン語	spaCy, Stanza, UDPipe 2, NLP-Cube, spacy-udpipe, Camphr-Udify, spaCy-COMBO など
エストニア語	Stanza, spacy-udpipe, Turku-neural-parser-pipeline, spaCy-COMBO, UDPipe 2 など
バスク語	spaCy-ixaKat, Turku-neural-parser-pipeline, Stanza, spacy-udpipe, Camphr-Udify など
ペルシア語	spaCy-COMBO, Turku-neural-parser-pipeline, spacy-udpipe, Camphr-Udify, UDPipe 2 など
スオミ語	Stanza, UDPipe 2, spacy-udpipe, NLP-Cube, Turku-neural-parser-pipeline など
フェロー語	Turku-neural-parser-pipeline
フランス語	spaCy, NLP-Cube, UDPipe 2, spacy-udpipe, Camphr-Udify, spaCy-COMBO など
ゲール語 (アイルランド)	Stanza, UDPipe 2, spacy-udpipe, spaCy-COMBO, Turku-neural-parser-pipeline など
ゲール語 (スコットランド)	GLA, Stanza, UDPipe 2, spacy-udpipe
ガリシア語	Camphr-Udify, Stanza, Turku-neural-parser-pipeline, NLP-Cube, spacy-udpipe など
古典ギリシア語	Camphr-Udify, UDPipe 2, spaCy-COMBO, Stanza, Turku-neural-parser-pipeline など
ヘブライ語	HebPipe, spaCy-COMBO, Turku-neural-parser-pipeline, UDPipe 2, Camphr-Udify など
ヒンディー語	NLP-Cube, spaCy-COMBO, UDPipe 2, Turku-neural-parser-pipeline, Stanza など
クロアチア語	NLP-Cube, UDPipe 2, Turku-neural-parser-pipeline, Camphr-Udify, Stanza など
ハンガリー語	UDPipe 2, Camphr-Udify, NLP-Cube, hu_core_ud_lg, spaCy-COMBO, spacy-udpipe など
アルメニア語	Stanza, Camphr-Udify, UDPipe 2, spacy-udpipe, spaCy-COMBO, YerevaNN など
インドネシア語	Stanza, Turku-neural-parser-pipeline, NLP-Cube, spacy-udpipe, Camphr-Udify など
イタリア語	NLP-Cube, spaCy-COMBO, Camphr-Udify, Stanza, spaCy, UDPipe 2, spacy-udpipe など
日本語	spaCy-SynCha, spaCy-ChaPAS, spaCy, NLP-Cube, GiNZA, UDPipe 2, UniDic2UD など
韓国語	Camphr-Udify, Stanza, UDPipe 2, Turku-neural-parser-pipeline, NLP-Cube, spaCy-COMBO
ラテン語	spaCy-COMBO, Stanza, UDPipe 2, spacy-udpipe, Camphr-Udify, NLP-Cube など
リトアニア語	spaCy, Stanza, Camphr-Udify, UDPipe 2, spacy-udpipe
古典中国語	UD-Kanbun, UD-Chinese, spacy-udpipe, Stanza, UDPipe 2
マルタ語	Stanza, UDPipe 2, spacy-udpipe, Camphr-Udify
ノルウェー語 (ブークモール)	spaCy-COMBO, Camphr-Udify, UDPipe 2, Turku-neural-parser-pipeline, spaCy
ノルウェー語 (ニーノルスク)	Stanza, spacy-udpipe, spaCy-COMBO, UDPipe 2, Turku-neural-parser-pipeline
オランダ語	spaCy, Camphr-Udify, Stanza, NLP-Cube, spacy-udpipe, spaCy-COMBO, UDPipe 2 など
ポーランド語	spaCy, Camphr-Udify, Stanza, UDPipe 2, NLP-Cube, spaCy-COMBO, PDBparser など
ポルトガル語	spaCy, Camphr-Udify, Stanza, NLP-Cube, spaCy-COMBO, UDPipe 2, spacy-udpipe など
ルーマニア語	UDPipe 2, Camphr-Udify, spaCy, Stanza, NLP-Cube, spaCy-COMBO など
ロシア語	Stanza, Camphr-Udify, NLP-Cube, spacy-udpipe, spaCy-COMBO, UDPipe 2 など
スロバキア語	Camphr-Udify, UDPipe 2, spacy-udpipe, NLP-Cube, spaCy-COMBO, Stanza など
スロベニア語	CLASSLA, Turku-neural-parser-pipeline, NLP-Cube, UDPipe 2, spacy-udpipe など
セルビア語 (キリル)	Camphr-Udify
セルビア語 (ラテン)	CLASSLA, UDPipe 2, Camphr-Udify, Turku-neural-parser-pipeline, NLP-Cube など
スウェーデン語	Camphr-Udify, NLP-Cube, Stanza, spaCy-COMBO, UDPipe 2, spacy-udpipe など
タミル語	Camphr-Udify, UDPipe 2, spacy-udpipe, Stanza など
タイ語	spaCy-Thai, Turku-neural-parser-pipeline
トルコ語	Stanza, NLP-Cube, spaCy-COMBO, Turku-neural-parser-pipeline, UDPipe 2, spacy-udpipe
ウクライナ語	Stanza, Camphr-Udify, spaCy-COMBO, UDPipe 2, Turku-neural-parser-pipeline など
ベトナム語	Stanza, Turku-neural-parser-pipeline, spaCy-COMBO, NLP-Cube, UDPipe 2 など
ウォロフ語	UDPipe 2, Stanza
中国語 (簡体)	Stanza, UDPipe 2, spacy-udpipe, UD-Chinese, spaCy など
中国語 (繁体)	Stanza, UDPipe 2, spacy-udpipe, UD-Chinese, spaCy など

3　deplacy の多言語化

50 以上の書写言語 (アフリカーンス語，アラビア語，ブルガリア語，カタルーニャ語，コプト語，チェコ語，ウェールズ語，デンマーク語，ドイツ語，ギリシア語，英語，スペイン語，エストニア語，バスク語，ペルシア語，スオミ語，フェロー語，フランス語，ゲール語 (アイルランド・スコットランド)，ガリシア語，古典ギリシア語，ヘブライ語，ヒンディー語，クロアチア語，ハンガリー語，アルメニア語，インドネシア語，イタリア語，日本語，韓国語，ラテン語，リトアニア語，マルタ語，ノルウェー語 (ブークモール・ニーノルスク)，オランダ語，ポーランド語，ポルトガル語，ルーマニア語，ロシア語，スロバキア語，スロベニア語，セルビア語 (キリル・ラテン)，スウェーデン語，タミル語，タイ語，トルコ語，ウクライナ語，ベトナム語，ウォロフ語，中国語 (簡体・繁体)) に対し，deplacy を適用した．

また，Google Colaboratory 上で，各言語ごとにデモページを製作・公開した．各デモページにおいては，われわれが製作した UD-Kanbun [7] や UniDic2UD [8] のみならず，spaCy [9] や Stanza [10] や GiNZA [11] など多彩な係り受け解析エンジンにも接続可能 (表 1) とし，インストーラをデモページに組み込むとともに，それぞれの解析エンジンの解説ページへのリンクも含めた．これに加え，バスク語・コプト語・タイ語の解析エンジンについては，UD が扱えるよう改造を加えた (付録参照)．

これらの言語を `deplacy.render` で扱う際に，古典中国語とは異なる表示手法が必要となり，それらを deplacy に組み込んだ．主な点を以下に述べる．

3.1　交差のある係り受け

英語などの言語では，係り受けに交差が発生する．交差の起こらない古典中国語とは異なり，多言語化においては，係り受けの交差を考慮する必要が生じた．具体的には，図 5 に示す通り，交差部分で縦横どちらかの罫線を優先的に表示することで，交差をうまく可視化した．

```
>>> import stanza
>>> nlp=stanza.Pipeline("en")
>>> doc=nlp("Who did he go with?")
>>> import deplacy
>>> deplacy.render(doc)
Who  PRON ┌──────┐<┐ obl
did  AUX  <┐     │ │ aux
he   PRON <┐│     │ │ nsubj
go   VERB ┘┘└┐┌──┘ root
with ADP  <──┘│    case
?    PUNCT <──┘    punct
```

図 5: 交差のある係り受けの可視化

3.2　右から左へ書く言語への対応

ヘブライ語などの言語では，書字方向が右から左へ向かうため，各行の左側に単語を表示するのが難しい．そこで，各行の右側に単語を表示するオプションとして，`deplacy.render` に `WordRight=True` を準備した (図 6)．このオプション自体は，言語には依存しないことから，左側で表示が乱れてしまう場合にも有効だと考えられる．

```
>>> import spacy_combo
>>> nlp=spacy_combo.load("he_htb")
>>> doc=nlp("עם מי הוא הלך?")
>>> import deplacy
>>> deplacy.render(doc,WordRight=True)
 case     ┌> ADP   עם
  obl   ┌>┘ ADV   מי
nsubj   │ ┌> PRON  הוא
 ROOT ┌─┘┘ VERB  הלך
punct └────> PUNCT ?
```

図 6: 各行の右側に単語を表示

なお，`deplacy.serve` には `RtoL=True` オプションを準備しているが，これは SVG を右から左へ描画するオプションで，書字方向が右から左へ向かう言語専用である．

3.3　係り受けタグの日本語化

日本語の係り受け可視化において，係り受けタグを日本語化してほしい，という要望が寄せられた．日

```
>>> import spacy
>>> nlp=spacy.load("ja_ginza")
>>> doc=nlp("彼は誰と出かけたの？")
>>> import deplacy
>>> deplacy.render(doc,Japanese=True)
彼     PRON ┐<──┐     nsubj(主語)
は     ADP  <┘   │     case(格表示)
誰     PRON ┐<┐ │     obl(斜格補語)
と     ADP  <┘ │ │     case(格表示)
出かけ VERB ┐┘┘┌┐┌┐ ROOT(親)
た     AUX  <┘   │ │ aux(動詞補助成分)
の     PART <────┘ │ mark(標識)
？     PUNCT <─────┘ punct(句読点)
```

図 7: 係り受けタグの日本語化

本語を特別扱いすべきか，われわれとしては悩んだのだが，結局 `deplacy.render` に `Japanese=True` オプションを準備した (図 7).

このオプションは，もちろん他の言語においても使用可能だが，他の言語の係り受けタグを日本語で表示する意義については，かなり疑問が残る．50 以上の書写言語全てに対し，係り受けタグ全ての訳語を準備すべきなのかもしれないが，それは今後の課題としたい．

4 おわりに

https://koichiyasuoka.github.io/deplacy/ において，deplacy を公開している．インストール方法や，デモページへのリンクも準備しているので，ぜひ Edge・Safari・Chrome でアクセスしてみてほしい．

参考文献

[1] 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.

[2] 安岡孝一: 漢文の形態素解析・依存文法解析・直接構成鎖解析, 東方學報, 第 94 册 (2019 年 12 月), pp.330-322.

[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] 安岡孝一: 古典中国語 Universal Dependencies で読む『孟子』, センター研究年報 2018 別冊，京都: 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 (2019 年 3 月).

[6] 安岡孝一: Universal Dependencies の拡張にもとづく古典中国語 (漢文) の直接構成鎖解析の試み, 情報処理学会研究報告, Vol.2019-CH-120 (2019 年 5 月), No.1, pp.1-8.

[7] 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.

[8] 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.

[9] Matthew Honnibal, Mark Johnson: An Improved Non-monotonic Transition System for Dependency Parsing, EMNLP 2015: Conference on Empirical Methods in Natural Language Processing (September 2015), pp.1373-1378.

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

[11] 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.

[12] Iakes Goenaga, Nerea Ezeiza, Koldo Gojenola: Combining Clustering Approaches for Semi-Supervised Parsing: the BASQUE_TEAM system in the SPRML'2014 Shared Task, First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages (July 2014).

[13] Maria Jesus Aranzabe, Aitziber Atutxa, Kepa Bengoetxea, Arantza Diaz de Ilarraza, Iakes Goenaga, Koldo Gojenola, Larraitz Uria: Dependentzia Unibertsalen eredura egokitutako euskarazko zuhaitz-bankua, Ekaia, zk:35 (2019), pp.291-307.

[14] Amir Zeldes, Caroline T. Shroeder: An NLP Pipeline for Coptic, Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (August 2016), pp.146-155.

[15] Wannaphong Phatthiyaphaibun, Korakot Chaovavanich, Charin Polpanumas, Arthit Suriyawongkul, Lalita Lowphansirikul, and Pattarawat Chormai: PyThaiNLP: Thai Natural Language Processing in Python, Zenodo (June 27, 2016).

[16] Virach Sornlertlamvanich, Naoto Takahashi, and Hitoshi Isahara: Building a Thai part-of-speech tagging corpus (ORCHID), Journal of the Acourstical Society of Japan (E), Vol.20, No.3 (May 1999), pp.189-198.

[17] 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.

[18] 飯田龍, 小町守, 井之上直也, 乾健太郎, 松本裕治: 述語項構造と照応関係のアノテーション, 自然言語処理, Vol.17, No.2 (2010 年 4 月), pp.25-50.

[19] 渡邉陽太郎, 浅原正幸, 松本裕治: 述語語義と意味役割の結合学習のための構造予測モデル, 人工知能学会論文誌, Vol.25, No.2 (2010 年 2 月), pp.252-261.

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

付録　係り受け解析エンジンの改造

表 1 に示した係り受け解析エンジンのうち，以下の 6 つは，既存の解析エンジンに対し，われわれが改造を施したものである．

spaCy-ixaKat　バスク語解析エンジン ixaKat [12] をもとに，spaCy と接続して，UD を扱えるよう改造した．ixaKat の品詞体系や係り受けタグは，UD とは異なる独自のものだが，変換テーブルが公開 [13] されており，これを spaCy-ixaKat に組み込んでいる．

spaCy-Coptic　コプト語解析エンジン coptic-nlp [14] をもとに，spaCy と接続して，UD を扱えるよう改造した．coptic-nlp の品詞体系は独自のものだが，係り受けタグは UD とほぼ同様であり，それらの変換テーブルを spaCy-Coptic に組み込んでいる．

spaCy-Thai　spaCy のタイ語モジュール PyThaiNLP [15] は，単語切りと品詞付与 (UD 品詞と ORCHID 品詞 [16] の両方を付与可能) はおこなえるものの，係り受け解析が実装されていなかった．そこで，UD_Thai-PUD を UDPipe 1.2.0 [17] で学習し，ORCHID に基づく係り受け解析モジュールを製作し，PyThaiNLP と合わせて spaCy-Thai とした．

spaCy-SynCha　日本語述語項解析エンジン SynCha [18] をもとに，述語項解析の結果を単語間の係り受けへと変換する形で spaCy と接続して，UD を扱えるよう改造した．近代日本語 (旧字旧仮名) も解析可能 [8] である．

spaCy-ChaPAS　日本語述語項解析エンジン ChaPAS [19] をもとに，述語項解析の結果を単語間の係り受けへと変換する形で spaCy と接続して，UD を扱えるよう改造した．近代日本語 (旧字旧仮名) も解析可能 [8] である．

spaCy-COMBO　UD にもとづく係り受け解析エンジン COMBO [20] は，50 以上の書写言語で UD を扱えるものの，単語切りが実装されていなかった．また，内部で使用しているモジュールのバージョンが古かった (TensorFlow 1.15.2 と scikit-learn 0.20.3)．そこで，spaCy の単語切りモジュールと接続し，言語モデルの読み込みを改造 (特に TensorFlow 2 と Joblib 対応) して，spaCy-COMBO とした．