日本語 Universal Dependencies の通時的転移可能性について

尾崎 太亮 [1] 臼井 久生 [1] 古宮 嘉那子 [1] 浅原 正幸 [2] 小木曽 智信 [2]
[1] 東京農工大学大学院 生物システム応用科学府
[2] 国立国語研究所
{hiroaki-ozaki,h-usui}@st.go.tuat.ac.jp
kkomiya@go.tuat.ac.jp, {masayu-a, togiso}@ninjal.ac.jp

概要

日本語の通時的な文法研究のため，近代以前の日本語に対する統語情報の付与を目指し，現代日本語 Universal Dependencies (UD) 解析の zero-shot 転移を検討した．UD で広く研究される言語横断的な解析手段に着目し，現代日本語 UD で学習した解析器を明治期の文書を対象とした UD コーパスで評価した結果，係り受け関係にある二語の抽出は高い精度を維持していたものの，文法的な観点では述語とその格要素の転移性能が低く，また助動詞を含めた文末表現も適切に解析できないことも明らかになった．

1 はじめに

日本語の通時的な文法研究を行う上で，日本語の各時代の文に対する形態論情報や統語情報は研究対象の言語現象を調査するための基礎情報である．しかし，すでに話者の存在しない古典に対して，特にアノテーターの負担の大きい統語情報の付加は非常に困難であり，現在入手可能な日本語近代以前の日本語に対する統語情報は限定的である．

一方，Universal Dependencies (UD) などの多言語横断での統一的な統語アノテーションと深層学習を活用したそれらの解析器の進展により，転移学習による低リソース言語での高い解析精度を実現する手段が確立されてきている [1]．また，UD には現代日本語 [2] の他，明治期の言文一致運動以前に刊行され，近代文語文で書かれた明六雑誌を対象とした UD Japanese Modern コーパス (以下，明六雑誌)[3] が存在している．これを活用し，転移学習よって日本語の通時的な文法研究のための統語情報の付与がどの程度行えるのか，を明らかにすることを目的に，現代口語文 UD で学習された解析器の近代文語での zero-shot 転移性能について主に明六雑誌での性能を評価した結果を報告する．

2 関連研究

近代日本語に対する UD 解析手法については，安岡 [4] による近代語用の UniDic を活用した形態素情報の付け替えと，既存の日本語 UD 係り受け解析器の組み合わせによる解析の検討が行われており，この形態素情報の付け替えによって大幅に精度向上が図れるものの，明六雑誌で直接的に学習した精度には至らないことが報告されている．また，その解析器は，unidic2ud[1)] として公開されている．

3 通時的転移性能の調査指針

通時転移性能を考える上で，精度に影響を与える要因として，本研究において特に注目している文法の観点に加えて，語の意味変化や用いられる語の変化，言語処理・機械学習的な観点の二つが大きく考えられる．これらをなるべく分離して評価できるよう，以下の検討を行った．

埋め込み表現　通常，深層学習による UD 解析は，語と形態素情報の埋め込み表現の双方を用いた解析が一般的であるが，それぞれの表現が達成しうる精度を個別に評価した．特に語の埋め込み表現においては，現代文で学習された日本語言語モデルに加えて，通時的な語用変化を考慮して近代以前の日本語から現代語への翻訳モデル [5] のものも評価した．

転移元コーパス　アノテーション基準の微妙な差異や学習データ量の影響を考慮して，複数の現代日本語 UD コーパスからの転移性能を評価した．

解析モデル　依存文法解析では一般的に，文を形態素解析したのち，形態素間の隣接行列を直接推定するグラフ型の解析器と，形態素列を先頭から機械学習手段で推定された手続きを施すことで解析木を得る遷移型解析器の二種類が存在する．本研究では主にグラフ型の解析器を用いたが，一般公開されて

1) https://github.com/KoichiYasuoka/UniDic2UD

表 1: 各モデルの UAS/LAS 値

モデル	学習データ	BCCWJ		GSD		雪國		舞姫		明六雑誌	
		UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
Tohoku	BCCWJ	**84.91**	**83.60**	**86.41**	**85.36**	76.11	72.57	72.90	72.57	**79.06**	57.62
T5	+GSD	83.35	81.82	85.44	84.19	83.19	81.42		63.55	76.04	56.44
UPOS	検証:BCCWJ	77.15	71.68	79.28	73.89	81.42	70.80	76.64	67.29	73.46	53.71
Tohoku	BCCWJ	84.72	83.33	86.26	85.05	76.11	72.57	82.24	72.90	79.01	**57.69**
T5	+GSD	63.61	57.53	65.80	60.12	76.11	72.57	72.90	63.55	57.18	43.27
UPOS	検証:GSD	76.96	71.46	79.11	73.75	81.42	72.57	63.55	50.47	73.24	53.16
Tohoku	BCCWJ	84.88	**83.60**	86.22	85.10	77.88	76.11	82.24	72.90	79.02	57.57
T5		83.12	81.64	85.07	83.80	**84.96**	**83.19**	72.90	63.55	75.83	56.48
UPOS		77.22	71.72	78.99	73.64	81.42	74.34	80.37	69.17	73.46	53.41
Tohoku	GSD	79.56	76.39	85.35	83.44	83.19	81.42	82.24	71.03	73.20	53.59
T5		78.66	75.91	84.80	83.02	74.34	72.57	74.77	67.29	74.35	54.32
UPOS		72.70	66.70	77.71	71.98	80.37	72.57	**85.98**	**74.77**	72.25	52.77
unidic2ud	BCCWJ	82.38	80.84	83.97	82.63	72.57	69.03	80.37	71.03	68.09	51.11
-GiNZA		(91.1)	(89.2)	-	-	-	(76.11)	74.77	57.94	-	-

いる遷移型の日本語 UD 解析器も比較に用いた．

4 実験設定

データセット 学習に用いたコーパスは現代日本語 UD である BCCWJ コーパス (以下，BCCWJ) と[2] と GSD コーパス (以下，GSD)[3] の学習データセットの双方または片方を用いた．検証には，上記コーパスの検証データセットを用いた．評価には，上記コーパスの評価データセットに加えて，明六雑誌と，unidic2ud のレポジトリに公開されている雪國，舞姫冒頭の UD アノテーションデータ[4] を用いた．

評価指標と評価方法 係り受け解析のみの評価に着目するため，形態素解析部を統一し，近代文語文については unidic2ud の解析結果 (近代語 UniDic) をそのまま用いた．現代文 (BCCWJ, GSD) に関しては GiNZA の解析結果を用いた．評価指標は，安岡 [4] の結果との比較を考慮して，係り受け関係にある二語の関係ラベル込みの抽出性能指標である Labeled Attachment Score(LAS) を用いた．これに加えて，係り受け関係にある二語の抽出性能を測る Unlabeled Attachment Score(UAS) も比較に用いた．評価スクリプトは CoNLL shared task 2018[6] で用いられた評価スクリプト[5] を用いた．ただし，BCCWJ の評価については，上記の評価スクリプトをそのまま適用するとエラーが発生したため，Appendix A に記載した工夫を施した．

評価対象モデル 主な解析器としてグラフ型である DiaParser[7] を用いた．語の埋め込み表現として，東北大学が提供する日本語 BERT[6] と Usui らによる近代以前の日本語から現代口語へ翻訳する T5[5] のエンコーダを用い，学習中にこれらの埋め込み表現に対する更新は行わなかった．埋め込み表現として形態素情報を用いる場合は，UD の UPOS タグを埋め込み表現に変換し，解析器の学習によって表現の更新を行った．その他のハイパーパラメタは付録の表 2 に示す．比較対象のモデルは以下である．

Tohoku: 東北大学が提供する日本語 BERT を用いた DiaParser モデル．

T5: Usui らによる T5 モデルのエンコーダを用いた DiaParser モデル．

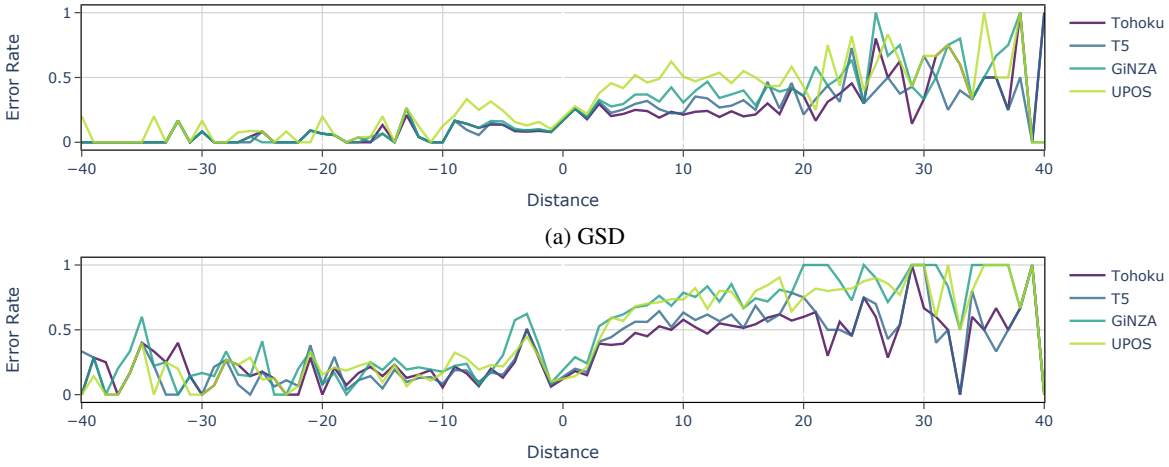
UPOS: 語の埋込表現の代わりに形態素情報の埋め込み表現を用いた DiaParser モデル．

GiNZA: spaCy[7] が提供する遷移型解析を BCCWJ で学習したモデル．現代文に関しては GiNZA[8] そのものを利用し，近代文語文に関しては unidic2ud を用いて形態素解析を行ったのち係り受け解析を GiNZA で実施した．

5 実験結果と考察

表 1 に各コーパスにおける UAS，LAS の値を示す．GiNZA の括弧で記載した UAS, LAS 値は GiNZA 公式サイトの記載及び安岡 [4] の結果を参照した．[8] 太字は，各コーパス・指標における最大値を示

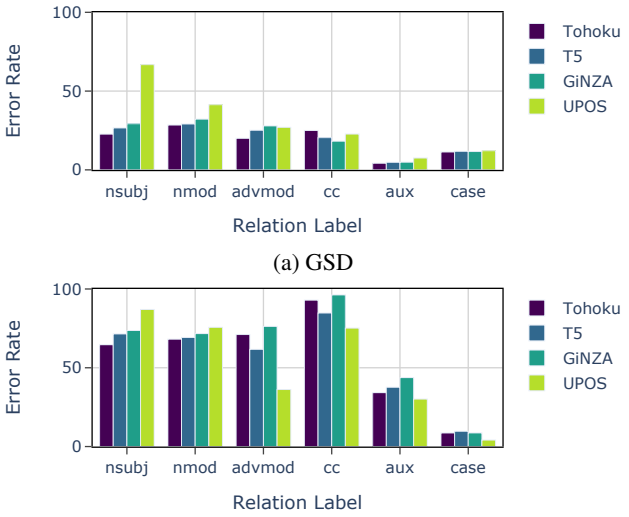
2) https://github.com/UniversalDependencies/UD_Japanese-BCCWJ
3) https://github.com/UniversalDependencies/UD_Japanese-GSD
4) https://github.com/KoichiYasuoka/UniDic2UD/tree/master/benchmark
5) https://universaldependencies.org/conll18/conll18_ud_eval.py
6) https://huggingface.co/cl-tohoku/bert-base-japanese
7) https://spacy.io/
8) https://megagonlabs.github.io/ginza/



(a) GSD

(b) 明六雑誌

図 1: 係り受け関係にある二語の距離に対する各解析器のエラーレート．



(a) GSD

(b) 明六雑誌

図 2: 関係ラベルに対する各解析器のエラーレート

す．BCCWJ, GSD, 明六雑誌の各コーパスにおいて **Tohoku** モデルが高い性能を有している．一方，雪國や舞姫においては，**T5** モデルが高い性能を有している．しかし，あくまで明六雑誌の範囲では翻訳タスクの事前学習による解析性能の向上があるとは言い難い結果となった．GiNZA の現代文の評価結果が松田の結果 [8] と乖離しているが，§4 で記載した通り評価スクリプトの違いが大きいと考えられる．以下，§3 で述べた三つの観点に即して結果を論ずる．

機械学習的な影響　機械学習的な影響としては，主に学習データ量と解析器のモデルの差異が考えられる．学習データ量の観点では，GSD の学習データが 7050 文であるのに対し，BCCWJ が 40801 文であるので，その学習データ量の差異が性能差に表れていると考えられる．また今回，BCCWJ と GSD を合わせた学習データの結果が総じて高い精度を有していたことから，UD 解析器の学習の観点でデータ量増大によって精度向上のわずかながらの余地があると考えられ，またそれによる近代文語文の転移性能の向上も期待できると考えられる．

解析モデルの観点では，DiaParser ベースのモデルが，遷移型の解析器である GiNZA の結果に対して概ね上回っており，その差は特に近代文語文において大きくなっている．このため，遷移型はこのような転移学習に不利である可能性が考えうる．深層学習によるグラフ型解析器は埋め込み表現間の注意機構をより直接的に活用することができ，より安定的に転移が行える可能性がある．

図 1 には係り受け関係にある二語の距離 (何トークン離れているか) に対するエラーレートを示す．**Tohoku**，**T5**，**UPOS** の各モデルは BCCWJ と GSD を共に学習データに用いたものを用いた．**Tohoku**，**T5** と比べて遷移型の GiNZA モデルは特に長距離の係り受け関係でのエラーレートが高い．一方，注意機構を直接的に扱うグラフ型では二語の距離が大きな影響を与えるとは考えにくく，実際，距離にさほど依存しないエラーレートとなっている．また，GiNZA の明六雑誌でのエラーレートは，三トークン以上の距離で 5 割を超え，形態素情報のみ (**UPOS**) の結果よりも高い値となっている．

語用および文法変化の影響　表 1 において，現代文である BCCWJ・GSD における UAS・LAS の値を比較すると，語の埋め込み表現を用いた場合，UAS の方が LAS より 1, 2 ポイント程度高い値であるの

に対し，形態素情報のみで学習した **UPOS** モデルでは 5 ポイント以上の差が発生している．UAS と LAS の値の差は，係り受け関係ラベルを評価に加味するかどうかの違いであるので，係り受け関係そのものではなく，関係ラベルの推定に語レベルの情報を必要としており，その点がこの差を発生させていると考えられる．しかし，近代文語文 (明六雑誌，舞姫，雪國) において UAS と LAS の同様の差を鑑みると，語の埋め込み表現と形態素情報のみ場合では特段大きな性能差がなく，係り受け関係ラベルの精度という観点では，形態論情報とさほど相違ない情報しか転移できていないことが考えられる．そこで，係り受け関係ラベルごとのエラーレートを調査した．図 2 には，名詞句主語 (nsubj)，名詞修飾語 (nmod)，副詞修飾語 (advmod)，等位接続詞 (cc)，助動詞 (aux)，格標識 (case) の関係ラベルについて，GSD と明六雑誌におけるエラーレートを示す．他の関係ラベルについては付録の図 4 に記載した．

名詞句主語 (nsubj) では現代文で二割程度であったエラーレートが近代文語文では六割以上に増大している．その他の格要素に関するエラーレートも高く (図 4 における iobj, obl)，述語項構造に関する情報はあまり転移できないことがわかる．これが，上記にある UAS・LAS の値の差の要因となっている．

格標識 (case) に関しては，GSD では解析モデルにおける差異はほぼなく，明六雑誌においては形態素情報のみの場合が最も高い値を示した．これは，内容語の直後の助詞はほぼ直前の内容語に係るというルールが近代文語文においても機能していると言える．一方，助動詞 (aux)，については，同じく形態素情報のみの場合が最も高い値ではあるものの，その精度は現代文に対して大きく低下している．これは例えば「〜如し」や「〜なり」など現代では使われない助動詞の語用の問題であったり，文末の言い回しの変化の影響が大きいと考えられる．図 3 には各モデルの文末表現に対する UD 解析事例を示す．形態素解析結果について，名詞を N，動詞を V，助動詞を A で表記した．`unidic2ud` の解析で「ある」が助動詞と誤判定されている．このため，**UPOS** モデルでは唯一の内容語である「通患」に全て係る解析結果となっている．**Tohoku** や **T5** モデルでは「に」を格助詞と認識して格標識ラベルを付与している．**T5** では「通患たる」と「あらず」を複合語として認識した上で，格助詞「に」の作用で「通患たる」が「あらず」の斜格要素とみなされている．**Tohoku** では，係り受け関係が交差しており，学習データの文法にも合致していない．

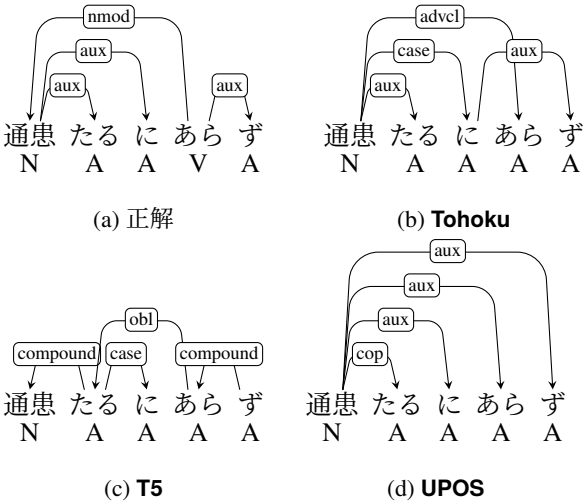


図 3: 文末表現に対する UD 解析結果例

名詞修飾語 (nmod) の GSD における出現頻度は 5.8%程度であったものが，明六雑誌では 16.9%と著しく増加しており (付録 図 5 参照)，図 3 にあるような漢語的表現に関する係り受け関係に多用されていると考えられる．このため，このエラーレートも名詞句主語と同様に増加している．

また，副詞修飾語 (advmod) や等位接続詞 (cc) では，語の埋込表現よりも形態素を用いた転移の方が有効であることがわかる．このことから，現代語と近代文語文では用いられる副詞や接続詞が大きく異なることが示唆される．また，**T5** モデルが **Tohoku** よりもわずかに良好な精度を得ており，歴史的資料の翻訳による語彙獲得がある程度影響することが示唆される．

6 おわりに

本稿では，日本語の通時的な文法研究のため，近代以前の日本語に対する統語情報の付与を目指し，現代日本語 UD 解析の zero-shot 転移を検討した．現代日本語 UD で学習した解析器を明治期の近代文語文に適用した結果，機械学習的な観点ではグラフ型解析器がより通時的転移に有効であることを示した．語用・文法的な観点では，述語とその格要素に関する転移性能が低く，助動詞を含めた文末表現も適切に解析できないことを示した．今後は，述語項や文末表現を考慮した精度向上を手段の検討を実施する．さらに，対象とする時代区分の拡大も検討する．

謝辞

本研究は JSPS 科研費 JP22K12145, 18K00634 及び国立国語研究所共同研究プロジェクト「アノテーションデータを用いた実証的計算心理言語学」の助成を受けたものです.

参考文献

[1] Dan Kondratyuk and Milan Straka. 75 Languages, 1 Model: Parsing Universal Dependencies Universally. In **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)**, pp. 2779–2795, Hong Kong, China, 2019. Association for Computational Linguistics.

[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, pp. 3–36, 2019.

[3] OMURA Mai, TAKAHASHI Yuta, and ASAHARA Masayuki. Universal Dependency for Japanese Modern Languages. **In Proceedings of The Japanese Association for Digital Humanities 2017 (JADH2017)**, 9 2017.

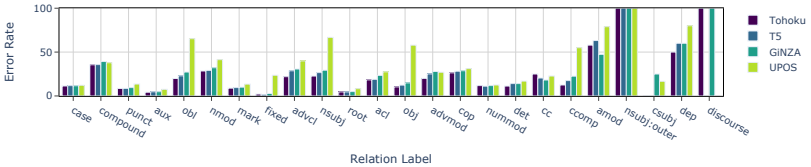
[4] 安岡孝一. 形態素解析部の付け替えによる近代日本語 (旧字旧仮名) の係り受け解析. Technical Report 3, 情報処理学会, 8 2020.

[5] Hisao Usui and Kanako Komiya. Translation from Historical to Contemporary Japanese Using Japanese T5. In **Proceedings of the 3rd International Workshop on Natural Language Processing for Digital Humanities**, Tokyo, Japan, November 2023. Association for Computational Linguistics.

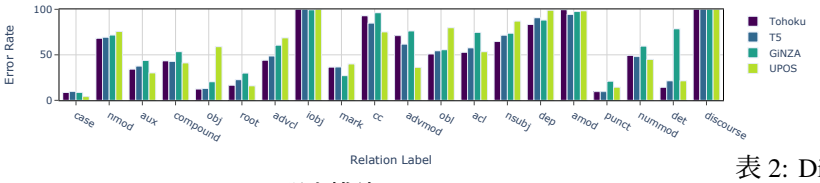
[6] Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In Daniel Zeman and Jan Hajič, editors, **Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies**, pp. 1–21, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[7] Giuseppe Attardi, Daniele Sartiano, and Maria Simi. Biaffine Dependency and Semantic Graph Parsing for Enhanced Universal Dependencies. In Stephan Oepen, Kenji Sagae, Reut Tsarfaty, Gosse Bouma, Djamé Seddah, and Daniel Zeman, editors, **Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies (IWPT 2021)**, pp. 184–188, Online, August 2021. Association for Computational Linguistics.

[8] 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, pp. 695–701, 2020.



(a) GSD



(b) 明六雑誌

図 4: 関係ラベルに対する各解析器のエラーレート

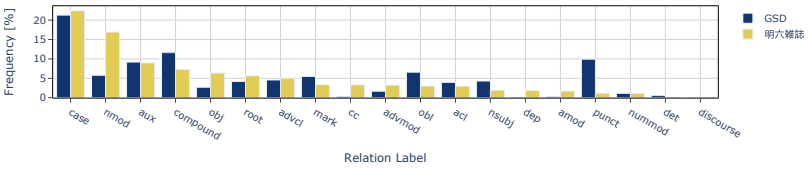


図 5: 各関係ラベルに対するコーパス内での出現頻度

A　BCCWJ における GiNZA 評価

GiNZA トークナイズ処理結果と BCCWJ のテストデータセットのトークンとの乖離が大きく，正しく評価が行えなかった．これは，評価スクリプトが文分割精度も評価対象としているため，トークナイズ結果に乖離が大きいと文を同定する処理が正常に行えないことに起因している．そこでまず，トークナイズの解離を解消するため，BCCWJ テストデータではトークンとして含まれてない全角空白を除去して GiNZA で形態素解析を行った．さらに，各文毎にデータを分割して，それぞれ評価スクリプトを用いて評価を行ったのち，それらを積算することでテストデータセット全体の評価結果を算出した．

表 2: DiaParser 学習時のハイパーパラメタ

ハイパーパラメタ	値
n_lstm_layers	2
n_lstm_hidden	400
n_bert_layers	0
use_attentions	True
attention_head	0
attention_layer	6
n_mlp_arc	500
n_mlp_rel	100
mix_dropout	0.1
その他 dropout	0.5
optimizer	adam
lr	1.2e-3
mu	0.9
nu	0.98
epsilon	1e-12
decay_steps	5000
accumulation_steps	1
warmup_steps_ratio	0.1
clip	5.0
decay	0.75
batch_size	5000
epochs	1000
patience	20
min_freq	2
fix_len	20