

# 国語研長単位に基づく UD Japanese

大村 舞  
国立国語研究所  
mai-om@ninjal.ac.jp

若狭 絢  
国立国語研究所  
awakasa@ninjal.ac.jp

浅原 正幸  
国立国語研究所  
masayu-a@ninjal.ac.jp

## 概要

Universal Dependencies (UD) は言語横断的に、かつ依存構造にツリーバンクを構築するプロジェクトである。全言語で統一した基準により、品詞タグ付け・依存構造アノテーションデータの構築が進められている。分かち書きをしない言語においては、基本単位として**構文的な語 (syntactic word)**を採用するように規定している。従前の日本語の UD データは、形態論に基づく単位である国語研短単位を採用していた。今回、我々は新たに構文的な語に近い単語単位である**国語研長単位**に基づく日本語 UD である **UD\_Japanese-GSDLUW**、**UD\_Japanese-PUDLUW**、**UD\_Japanese-BCCWJLUW** を構築したので報告する。

## 1 はじめに

Universal Dependencies [1] (UD) は、依存構造関係を付与する基本単位として**構文的な語 (syntactic word)**を用いることを規定している。英語やフランス語といった空白を用いて分かち書きをする言語においては、縮約形態を除いて、空白を語境界と認定することが多い。一方、分かち書きを語境界で明示しない東アジアの言語においては、どのような単位が構文的な語に規定すべきかという問題がある。

既に公開されている UD Japanese (日本語版 UD) は、その基本単位として**国語研短単位 (Short Unit Word: 以下 短単位)**を採用している。現在までに短単位に基づく形態素解析用辞書として、約 88 万語からなる UniDic [2] が公開されている。この語彙数の多さは、日本語の形態変化の豊かさ (morphologically rich) を示しているだろう。さらに、短単位は『現代日本語書き言葉均衡コーパス』(BCCWJ) [3]・『日本語日常会話コーパス』(CEJC) [4] をはじめとした形態論情報つきコーパスでも採用されており、150 万語規模の単語埋め込み NWJC2vec [5] でも短単位が使われているため、短単位を基準として言語処理

に必要な基本的言語資源が整備されている。この短単位に基づく言語資源の豊富さから、実用上は短単位に基づく処理が好まれる傾向にあった。

Murawaki [6] は、単位として短単位を採用している既存の UD Japanese コーパスは「形態素」単位であり、UD の原則にあげられる「基本単位を構文的な語とする」という点で不適切であることを指摘している。国語研においては、形態論情報に基づいて単位認定し、可能性に基づく品詞体系が付与されている短単位とは別に、文節に基づいて単位認定し、用法に基づく品詞体系が付与されている**国語研長単位 (Long Unit Word: 以下 長単位)**を規定している。さらに、同単位に基づいた形態論情報が BCCWJ や CEJC などのコーパスに付与されている。

本稿では、長単位に基づく UD Japanese の言語資源を整備したので報告する。長単位は、文節に基づいて認定される構文的な語とみなしうる単位である。長単位は UD の原則に基づく UD Japanese で採用すべき基本単位ではあるが、長単位に基づくコーパスの構築は、短単位に基づくコーパスの構築より困難である。さらに、長単位に基づく依存構造木が解析器によって生成しやすいかという問題もある。本稿では、国語研長単位に基づく UD Japanese について説明すると共に、既存の言語資源・解析器によって長単位に基づく UD が生成しやすいかについて議論する。

## 2 日本語の単語分割

### 2.1 国語研による語の単位認定

日本語の単語の境界認定として、国語研では、最小単位、短単位、長単位、文節などの単位を規定している [7]。それぞれの例を図 1 に示す。

最小単位はその語種に基づき定義される。日本語は、漢語・和語・外来語・固有名詞・数値表現・記号などの語種からなり、漢語は 1 文字 1 語、和語・外来語・固有名詞はその最も短い単位を 1 語とする。

最小単位	全  学  年  に  わたっ  て  小  学  校  の  国  語  の  教  科  書  に   大  量  の  挿  し  絵  が  用  い  ら  れ  て  い  る
短単位 (SUW)	全  学  年  に  わたっ  て  小  学  校  の  国  語  の  教  科  書  に   大  量  の  挿  し  絵  が  用  い  ら  れ  て  い  る
長単位 (LUW)	全  学  年  に  わたっ  て  小  学  校  の  国  語  の  教  科  書  に   大  量  の  挿  し  絵  が  用  い  ら  れ  て  い  る
文節	全  学  年  に  わたっ  て  小  学  校  の  国  語  の  教  科  書  に  大  量  の  挿  し  絵  が  用  い  ら  れ  て  い  る

図1 最小単位、短単位、長単位、文節の例 (BCCWJ の PB33\_00032 より)

数値表現は十進の位取りで発音できる単位に分割する。記号は1文字1語とする。

短単位は、最小単位に基づき、同じ語種同士の1回結合までを1単位としている。この短単位に対して UniDic 体系に基づく形態論情報 (品詞・活用形・活用形・語形・語彙素) が定義され、その品詞に基づいて付属語か自立語かが認定される。短単位に付与される品詞はその形態素に基づく「可能性に基づく品詞」が付与されるが、これはその形態素でとり得る用法をすべて考慮したものであり、実際の用法などは考慮されていない。用例収集のため、斉一な単語になることを目的としており、1単語辺りが短い。

長単位は、文節境界を認定したのちに、文節内の短単位要素の結合により認定される。この文節は日本語において係り受けを付与するのに適した単位である。文節単位の係り受け構造は係り受けが交差せず、左から右にかかり (右主辞)、短単位品詞をもとに容易に系列ラベリングでくみ上げられるという性質を持つ。文節は、1自立語に接頭辞・接尾辞・助詞・助動詞などから構成されるとし、その各要素を長単位として認定する。長単位の規程集には、長単位としてみならず複合辞 (Multi-word Expressions) が認定され、複合辞をなす機能表現は1単位とする。長単位は文法を基準とした単位のため構文的な語に近いと考えられる。また長単位はその用法 (係り受けなどの文脈) に基づく「用法に基づく品詞」が付与される。品詞の観点からも、短単位よりも長単位のほうが UD が示す「構文的な語」に近いと考えられる。

## 2.2 利用できる言語資源・解析器

	辞書	単語分割器	単語埋め込み	TTR
最小単位	N/A	N/A	N/A	N/A
短単位	UniDic	MeCab	NWJC2vec	0.00176
長単位	N/A	Comainu	N/A	0.02922
文節	N/A	Comainu	N/A	0.22221

表1 各単語単位で利用できる言語資源・解析器

表1 に各単位認定に利用できる言語資源・解析器について示す。表中 TTR は BCCWJ 中の Type Token Ratio であり、この値が高いほど、その単位の出現確率が小さくなることを意味する。最小単位は、短単位を認定する際の基本単位であり、現在のところ公開されている言語資源・解析器は存在しない。短単位は、形態素解析器 MeCab [8] において形態素解析用辞書 UniDic を用いることにより生成されるほか、NWJC2vec などの単語埋め込みが利用できる。長単位・文節については、辞書の構築は TTR の大きさからも分かる通り、語彙の膨大であるため存在しないが、短単位から構成規則により生成されるため、チャンカーとして Comainu [9] により生成することができる。

## 2.3 日本語 UD における単語分割の歴史

我々はいままで現代日本語 UD として4種類のコーパスを公開しており、今回さらに長単位版の UD コーパスを3種類公開した。既存の短単位版 UD について説明をする。

UD\_Japanese-KTC [10] は京都大学テキストコーパスを元に作られた UD である。UD\_Japanese-KTC は長単位に近い単語境界で単語を構成しており、人手によって句構造木もアノテーションされ、その句構造木に基づいて UD を構築している。現在 UD プロジェクトでは Version 2 が公開されているが、UD\_Japanese -KTC については Version 1 でメンテナンスが止まっている。

UD\_Japanese-BCCWJ は BCCWJ に基づいた UD コーパスである。BCCWJ では短単位・長単位・文節および文節単位の係り受けの情報が提供されている。UD\_Japanese-BCCWJ はこの BCCWJ からの形態素情報を元に Omura and Asahara [11] らが提案しているルールによって、単語係り受け構文へと自動変換されたものである。

UD\_Japanese-GSD と UD\_Japanese-PUD は Version 1.4 まで Google [12] が管理していたが、Version 2.0 より我々がメンテナンス及び公開しているコーパスで

ある [10]。UD\_Japanese-GSD と UD\_Japanese-PUD は ver.2.0 から 2.5 までは IBM の単語分割器 [13] により単語分割され修正されていたものであったが、ver. 2.6 からは短単位を基準として人手によりアノテーションをされている。単語係り受け構文については Omura and Asahara [11] らのルールによって自動変換されている。

### 3 長単位ベースの日本語 UD

データ		単語単位	文	文節	語
BCCWJ	train	SUW	40,801	308,648	715,759
		LUW	-	308,679	908,738
	dev	SUW	8,427	60,697	145,398
		LUW	-	60,722	178,306
	test	SUW	7,881	56,332	134,475
		LUW	-	56,350	166,859
GSD	train	SUW	7,050	57,174	130,298
		LUW	-	57,357	168,333
	dev	SUW	507	4,186	9,531
		LUW	-	4,203	12,287
	test	SUW	543	4,568	10,429
		LUW	-	4,588	13,034
PUD	test	SUW	1,000	9,971	22,910
		LUW	-	10,008	28,788

表 2 日本語長単位ベース UD の統計情報

我々は短単位と長単位によって違いを検証すべく、長単位ベースの日本語 UD を開発した。UD の v2.9 (2021 年 11 月) から、BCCWJ, GSD, PUD の長単位ベース UD を、UD\_Japanese-BCCWJLUW, UD\_Japanese-GSDLUW, UD\_Japanese-PUDLUW として公開している。<sup>1)</sup>

UD\_Japanese-GSD と UD\_Japanese-PUD は短単位ベースの単語境界、品詞そして文節ベースの係り受け関係を人手により一度アノテーションされている。我々はさらに、長単位ベースの UD を構築するため、GSD と PUD について長単位の境界、品詞、原型をアノテーションした。長単位をアノテーションしている際に、長単位と短単位でデータに一貫性の不備があった場合は、オリジナルのデータをさらに修正している。そしてこのアノテーションデータ GSD, PUD を Omura and Asahara [11] らの提案したルールベースによって、文節係り受けから UD の枠組みである単語係り受け関係のコーパスに変換し、UD\_Japanese-GSDLUW と UD\_Japanese-PUDLUW を構築した。この変換ルールは短単位と長単位の原型や品詞、形態的な特徴を元にルールベースで変換

している。<sup>2)</sup>UD\_Japanese-GSD と UD\_Japanese-PUD の元となっているこの文節係り受けコーパスは拡張 Cabocha 形式 [14] でオープンデータとして公開<sup>3)</sup>されている。

BCCWJ はオリジナルのデータにすでに短単位の境界と品詞、長単位の境界と品詞が公開されており、浅原らのデータ [15] を組み合わせることで文節係り受けデータも獲得できる。そこから GSD と PUD と同様に変換を行い UD\_Japanese-BCCWJLUW を構築した。

表 2 にそれぞれの長単位 UD の統計情報を示している。表 2 から分かる通り、長単位の語数は短単位の語数より少ない。これは長単位自体が短単位の結合することで構築されているからである。また、長単位は文節を基準に分割したものではあるが、前節の通り、複合語などが含まれていた場合、2 つの文節に分割されることもある。そのため、必ずしも短単位と長単位で文節の数は一致していない。

### 4 統語解析による比較実験

この節では単語単位の違う UD について、公開されているツール・言語資源によって統語解析を行うことで、どのくらい再現可能かを検討する。この実験では GSD コーパスを用いた。GSD は訓練・開発・評価データに分割されており、実験設定はこの分割に基づいて実施し、評価は 3 つのレベルで行った。1 つ目は未解析文を入力にした全ての解析器 (単語分割, 品詞タグ付け, 係り受け解析) を行うものである。2 つ目は正解の単語分割を入力にして品詞タグ付け・係り受け解析を行うものである。3 つ目は正解の単語分割と正解の品詞タグを入力に係り受け解析のみを行うものである。

全ての結果を付録の表 4 に示す。正解データの場合 **Gold** という表記で示している。**v2.5 (IBM)** は IBM の単語分割基準、**v2.8 (SUW)** が短単位、**v2.9 (LUW)** が長単位のデータとなっている。UDPipe [16](v1.2.0) によりこの 3 種の単語分割について再訓練を行ったものの結果を **UDPipe (T)** にて示している。UDPipe はパイプラインモデルで単語分割・品詞タグ付け・レンマ推定・係り受け解析を行うことができる。また LINDAT/CLARIN が提供す

2) 実際 Omura and Asahara [11] らのルールは短単位に基づいたルールであるが、長単位の場合でも品詞に関する変換ルールを 10 個ほど追加しただけで変換できた。

3) [https://github.com/masayu-a/UD\\_Japanese-GSDPUD-CaboCha](https://github.com/masayu-a/UD_Japanese-GSDPUD-CaboCha)

1) <https://universaldependencies.org/>



ツリーバンク	単語分割	UAS			LAS		
		w/o vec	w/vec	Diff	w/o vec	w/vec	Diff
<b>v2.5(IBM)</b>	UDPipe	77.11%	77.31%	+0.20	75.43%	75.87%	+0.44
<b>v2.8(SUW)</b>	UDPipe	84.40%	85.22%	+0.82	82.58%	83.50%	+0.92
<b>v2.8(SUW)</b>	MeCab	87.38%	88.22%	+0.84	85.40%	86.32%	+0.92
<b>v2.9(LUW)</b>	UDPipe	83.25%	83.49%	+0.24	81.83%	82.07%	+0.24
<b>v2.9(LUW)</b>	Comainu	87.91%	88.16%	+0.25	86.16%	86.45%	+0.29

表3 未解析文を入力した場合の係り受け解析精度 (Subset of 表4). Diff は w/o vec と w/vec の差である

る UDPipe モデル<sup>4)</sup>は v2.5 (IBM) の訓練データで訓練しており、その結果も **UDPipe (O)** として示している。さらに UDPipe は係り受け解析器に外部の単語埋め込みを利用することができる。短単位に基づく単語埋め込み NWJC2vec を利用していないものと利用したものの結果について **Train w/o vec** と **Train w/ vec** にて示している。v2.5 (IBM) でも UDpipe 特有の単語埋め込みが生成されているため、**Original** で示している。

全体的に v2.5 以前の UD よりも v2.8 以降の短単位と長単位を採用した UD のほうが、高精度に解析が可能であることが分かった。また、通常の UDPipe における v2.9(LUW) の係り受け解析精度は v2.8 (SUW) に 1.2-1.4%ほど劣っていた。

表3 に単語埋め込みの影響を見るため、未解析文を入力とした結果のみ抜粋している。短単位に基づく単語埋め込み NWJC2vec の利用の有無の差分を見ると v2.8 (SUW) のほうが、v2.9(LUW) よりも性能の向上が見られた。一方で、MeCab や Comainu を使った場合、単語埋め込みを使わない設定においては、v2.8(SUW) よりも v2.9(LUW) のほうが良い性能であり、また単語埋め込みを利用した場合の最終的な係り受けの精度は v2.8(SUW) と v2.9(LUW) とでほとんど差がないことが確認された。構文的な語の認定精度が構文解析精度に影響があることがわかる。単語単位の問題は単語埋め込みの構成単位にもかかわってくる。長単位に基づく単語埋め込みの構築は前述の通り Type Token Ratio の観点からは実用的ではなく、現状の計算資源では文字もしくは短単位に基づく単語埋め込みが好まれるだろう。

最後に、表4 の Lemma の項目をみると、短単位よりも長単位のほうがレンマ生成の精度が低いことが確認された。これは複合された形態素の語彙素の推定は難しいためと考えられる。長単位のレンマ生成

の精度向上が望めば、より良い解析精度を得られるだろう。

## 5 おわりに

本稿では日本語 UD における単語分かち書きの問題について示した。様々な日本語分かち書き基準について紹介するとともに、UD が掲げる理念に即した単位認定「構文的な語」にふさわしい単位として国語研長単位があることを紹介した。実際に長単位に基づく UD を構築し、公開・共有を行った。さらに長単位に基づく UD\_Japanese-GSDLUW (v2.9) について、公開されているツール・言語資源を用いて、その再構成可能性について検討を行った。結果、既存の形態素解析器 MeCab・Comainu とともに短単位に基づく単語埋め込み NWJC2vec を用いた設定において、短単位と長単位とで最終的な係り受けの性能において差がないことを確認した。

構文的な語に基づく依存構造ツリーバンク構築は、日本語のような分かち書きされない言語においてはかなり困難なタスクである。その形態素(短単位・長単位)・品詞・複合辞認定・依存構造木の構築には多大な作業が必要であった。最初に UD Japanese に携わってから、対外的な調整(権利関係・アノテーション基準)も含めて、8年の期間を要した。本基準に則ると、文節係り受けが付与された短単位・長単位に基づく BCCWJ や CEJC なども UD 対応することが可能である。

ただし、構文的な語としての長単位の出力が、工学的に有用であるかはまだ不明である。長単位は複合辞の問題を緩和している一方、先に述べた通り単語埋め込みの恩恵を受けにくくなるという問題がある。さらには、類型論の研究を考えた場合に、他言語においても同様の構文的な語を規定できるかという問題がある。今後他言語を含めた有用性を検証する必要があると考えられる。

4) <https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-3131>

## 謝辞

本研究は国立国語研究所コーパス開発センター共同研究プロジェクトの成果です。また、科研費17H00917の支援を受けました。

## 参考文献

- [1]Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. European Language Resources Association (ELRA), May 2016.
- [2]伝 康晴, 小木曾 智信, 小椋 秀樹, 山田 篤, 峯松 信明, 内元 清貴, and 小磯 花絵. コーパス日本語学のための言語資源—形態素解析用電子化辞書の開発とその応用 (特集コーパス日本語学の射程). 22:101–123, oct 2007.
- [3]Kikuo Maekawa, Makoto Yamazaki, Toshinobu Ogiso, Takehiko Maruyama, Hideki Ogura, Wakako Kashino, Hanae Koiso, Masaya Yamaguti, Makiro Tanaka, and Yasuharu Den. Balanced corpus of contemporary written Japanese. *Language resources and evaluation*, 48(2):345–371, December 2014.
- [4]Hanae Koiso, Yasuharu Den, Yuriko Iseki, Wakako Kashino, Yoshiko Kawabata, Ken'ya Nishikawa, Yayoi Tanaka, and Yasuyuki Usuda. Construction of the Corpus of Everyday Japanese Conversation: An interim report. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA), May 2018.
- [5]Masayuki Asahara. NWJC2Vec: Word embedding dataset from 'NINJAL Web Japanese Corpus'. *Terminology: International Journal of Theoretical and Applied Issues in Specialized Communication*, 24:7–22, January 2018.
- [6]Yugo Murawaki. On the definition of Japanese word, June 2019. arXiv: 1906.09719 [cs.CL].
- [7]Yasuharu Den, Junpei Nakamura, Toshinobu Ogiso, and Hideki Ogura. A proper approach to Japanese morphological analysis: Dictionary, model, and evaluation. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, pages 1019–1024. European Language Resources Association (ELRA), May 2008.
- [8]Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, July 2004.
- [9]小澤 俊介, 内元 清貴, and 伝 康晴. 長単位解析器の異なる品詞体系への適用. *自然言語処理*, 21(2):379–401, 2014.
- [10]Takaaki Tanaka, Yusuke Miyao, Masayuki Asahara, Sumire Uematsu, Hiroshi Kanayama, Shinsuke Mori, and Yuji Matsumoto. Universal dependencies for Japanese. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1651–1658. European Language Resources Association (ELRA), May 2016.
- [11]Mai Omura and Masayuki Asahara. UD-Japanese BCCWJ: Universal Dependencies annotation for the Balanced Corpus of Contemporary Written Japanese. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*. Association for Computational Linguistics, November 2018.
- [12]Ryan McDonald, Joakim Nivre, Yvonne Quirnbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. Universal Dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, August 2013.
- [13]Hiroshi Kanayama, Kentaro Torisawa, Yutaka Mitsuishi, and Jun'ichi Tsujii. A hybrid Japanese parser with hand-crafted grammar and statistics. In *COLING 2000 Volume 1: The 18th International Conference on Computational Linguistics*, pages 411–417, July 2000.
- [14]松吉俊, 浅原正幸, 飯田龍, and 森田敏生. 拡張 cabocha フォーマットの仕様拡張. 第 5 回コーパス日本語学ワークショップ, pages 223–232, 2014.
- [15]浅原 正幸 and 松本 裕治. 『現代日本語書き言葉均衡コーパス』に対する文節係り受け・並列構造アノテーション. *自然言語処理*, 25(4):331–356, 2018.
- [16]Milan Straka and Jana Straková. Tokenizing, POS tagging, lemmatizing and parsing ud 2.0 with UDpipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99. Association for Computational Linguistics, August 2017.

# A 付録

ツリーバンク	単語分割	品詞タグ付け	係り受け解析	Words	UPOS	XPOS	Lemmas	UAS	LAS	CLAS	MLAS	BLEX
v2.5 (IBM)	UDPipe (O)	UDPipe (O)	Original	91.96%	89.35%	88.98%	91.19%	77.91%	76.45%	66.50%	63.94%	65.98%
	Gold	UDPipe (O)	Original	-	96.93%	96.41%	99.07%	92.70%	90.65%	83.55%	80.64%	82.84%
	Gold	Gold	Original	-	-	-	-	94.85%	93.77%	87.08%	86.98%	87.08%
	UDPipe (T)	UDPipe (T)	Train w/o vec	91.94%	89.30%	89.00%	91.14%	77.11%	75.43%	64.84%	62.69%	64.42%
	Gold	UDPipe (T)	Train w/o vec	-	96.82%	96.34%	98.93%	92.12%	89.82%	82.10%	79.19%	81.37%
	Gold	Gold	Train w/o vec	-	-	-	-	94.68%	93.36%	86.29%	86.10%	86.29%
	UDPipe (T)	UDPipe (T)	Train w/ vec	91.94%	89.30%	89.00%	91.14%	77.31%	75.87%	65.71%	63.63%	65.33%
	Gold	UDPipe (T)	Train w/ vec	-	96.82%	96.34%	98.93%	92.58%	90.58%	83.58%	80.76%	82.84%
	Gold	Gold	Train w/ vec	-	-	-	-	95.20%	94.15%	87.85%	87.71%	87.85%
v2.8 (SUW)	MeCab	UD Pipe (T)	Train w/o vec	96.84%	94.42%	93.58%	96.07%	87.38%	85.40%	77.57%	75.04%	77.14%
	UDPipe (T)	UDPipe (T)	Train w/o vec	96.14%	93.96%	93.29%	95.39%	84.40%	82.58%	75.20%	72.81%	74.79%
	Gold	UDPipe (T)	Train w/o vec	-	97.39%	96.52%	99.20%	91.20%	89.10%	83.10%	80.32%	82.64%
	Gold	Gold	Train w/o vec	-	-	-	-	92.28%	91.12%	85.54%	85.00%	85.54%
	MeCab	UDPipe (T)	Train w/ vec	96.84%	94.42%	93.58%	96.07%	88.22%	86.32%	79.28%	76.74%	78.83%
	UDPipe (T)	UDPipe (T)	Train w/ vec	96.14%	93.96%	93.29%	95.39%	85.22%	83.50%	76.85%	74.48%	76.44%
	Gold	UDPipe (T)	Train w/ vec	-	97.39%	96.52%	99.20%	92.05%	90.07%	84.90%	82.13%	84.40%
	Gold	Gold	Train w/ vec	-	-	-	-	93.95%	93.32%	89.07%	88.67%	89.07%
	Comainu	UDPipe (T)	Train w/o vec	97.19%	94.34%	94.18%	91.32%	87.91%	86.16%	78.10%	74.19%	71.78%
v2.9(LUW)	UDPipe (T)	UDPipe (T)	Train w/o vec	95.02%	92.37%	92.16%	89.74%	83.25%	81.83%	72.31%	68.54%	67.10%
	Gold	UDPipe (T)	Train w/o vec	-	96.90%	96.70%	93.78%	92.82%	90.93%	82.66%	78.51%	75.66%
	Gold	Gold	Train w/o vec	-	-	-	-	93.86%	93.23%	85.77%	85.33%	85.77%
	Comainu	UD Pipe (T)	Train w/ vec	97.19%	94.34%	94.18%	91.32%	88.16%	86.45%	78.69%	75.05%	72.52%
	UDPipe (T)	UDPipe (T)	Train w/ vec	95.02%	92.37%	92.16%	89.74%	83.49%	82.07%	72.85%	69.15%	67.69%
	Gold	UDPipe (T)	Train w/ vec	-	96.90%	96.70%	93.78%	93.18%	91.26%	83.27%	79.30%	76.43%
	Gold	Gold	Train w/ vec	-	-	-	-	94.03%	93.74%	87.19%	86.86%	87.19%

表 4 実験結果：UD\_Japanese-GSD の v2.5, v2.8 および UD\_Japanese-GSDLUW(v2.9) の結果比較 (詳細)