UD_English-EWT とのつきあい方

金山 博　大湖 卓也
日本アイ・ビー・エム株式会社 東京基礎研究所
{hkana,ohkot}@jp.ibm.com

概要

Universal Dependencies (UD) の各言語のコーパスは、構文解析器の訓練と評価の両面で広く使われている。英語のコーパスの中で最も頻繁に使われている UD_English-EWT は、作成の経緯から、口語的な文や書き誤り、特殊な文字列なども含まれており、訓練・評価の両面で支障がある。本稿では、現状のデータの性質と問題点を紹介するとともに、メタデータを用いてノイズを除去したコーパスを用いて構文解析器を訓練した際の変化を調べる。

1　はじめに

Universal Dependencies (UD)[9, 10] は、多言語の構文構造を表現するために、17 種の品詞タグと 37 種の関係ラベルを用いた依存構造を定義し、それに基づいたツリーバンク（以下ではコーパスと記す）を作成する世界的なプロジェクトである。2021 年末現在、122 言語に対する 217 種のコーパスが公開されており、Stanza[11]、UDPipe[14]、Spacy[4] や Trankit[8] などの構文解析器（文区切り・単語区切り・品詞タグ付け・依存構造解析）の訓練および評価に用いられている。

UD のコーパスは、一つの言語に対して複数個が存在し得るため、「UD_言語名-種別」の形式で名付けられている。日本語の場合は UD_Japanese-GSD, UD_Japanese-BCCWJ などのコーパスが公開されており [17]、それぞれ文体、サイズ、作成の経緯やライセンス形態が異なっており、利用者は目的に応じて使い分けることになる。

英語のコーパスは現在 11 種が公開されているが、そのうち UD_English-EWT （以下 UD-EWT）が主たるコーパス[1)]として作成・利用されてきていた。現在も github 上[2)]でアノテーションのあり方について

1） 2017 年以前は UD_English（無印）として事実上の英語の標準コーパスであった。

2） https://github.com/UniversalDependencies/UD_English-EWT/

表 1　EWT コーパスのジャンルと文数。

ジャンル	文数
ブログ	2,030
e メール	4,900
ニュース	2,391
Q&A	3,488
レビュー記事	3,813
合計	16,622

アクティブな議論が続いており、改良やバグ修正が行われている。しかしながら、2 節に記すような経緯もあり、英語のメインとなるコーパスとしては質と量の面で課題が残っている。

本稿では、UD-EWT を訓練や評価に利用するにあたっての注意点を挙げるとともに、コーパスを変換して書き誤りを減らすことによる解析器の性能向上を試みる。

2　EWT コーパスの成り立ち

English Web Treebank (EWT) は、2012 年より LDC[3)] で公開されているコーパスであり、様々なジャンルの web 上のテキストデータに対して OntoNotes コーパス [5] に類似した形式の項構造のアノテーションを付与したものである。EWT が作られたきっかけは、OntoNotes や Penn Treebank [7] の中で主に使われるデータが、Wall Street Journal (WSJ) などのニュース記事であり、現実世界の言語現象を充分に反映できていないという問題に対処するためである。そこで EWT は、表 1 のように複数の分野からテキストを収集した。但し、EWT の文数は約 1.7 万文と、10 万文以上に拡張された OntoNotes v5 と比べると遙かに小さい。

ニュース記事と異なる分野のテキストが含まれることにより、現れる構文要素の分布にも特徴がある。WSJ と比較して、倒置や通貨表現が出現する割合は半分以下である一方で、呼格や間投詞は 20 倍以上出現すると報告されている [13]。また、ドイツ語、フランス語などの欧米言語のコーパスにおいて

3） Linguistic Data Consocium。

名詞句となっている文や、文末にピリオド等が無い文の割合が極端に小さいのと比べ、UD-EWT では比較的バランスが取れていることが観察された [18]。

EWT は後に Stanford Dependencies [2] の依存構造の形式に変換され [13]、さらに UD の CoNLL-U フォーマットに変換され、UD-EWT として公開されている。その後、拡張依存構造が加えられ [12]、現在に至る。

WSJ のコーパスに対する品詞タグ付けタスクの精度は 2010 年の段階で 97% 以上と飽和状態にあり [6]、くだけた文体などの難しい事象が残っているテキストが、その後のシェアドタスク [16] の題材として好適であった。また、UD の初期より商用利用が可能な CC BY-SA のライセンス形式で配布された数少ないコーパスであった[4]ことも、UD-EWT の普及を進めたと考えられる。

2015 年の UD v1 の公開以来、アノテーションについては多数の改良がなされてきた。特に 2017 年の UD v2 が導入された際には大幅な変更が加えられた。一方で、文の追加・削除や文区切りはほぼフリーズされており、2021 年公開の UD v2.8 で連続する空白が取り除かれたなど軽微な変更に留まっている。

3 コーパス内の事象

EWT コーパスは表 1 に示したような複数のジャンルのテキストを含むため、Penn Treebank などニュース記事から作成されたコーパスには稀であった現象が見つかる。

まず、インフォーマルな文体として、表 2 の (1) のように機能語を簡略化したものや、さらには前置詞の “to” を “2”, “for” を “4” と書く事象などが見られる。コーパス上ではこれらの lemma は正しい表記に変換されている。また、“gotta” は (2) のように複数語トークン (MWT) として表現され、その構成要素の lemma は本来の “get”, “to” となっている。(3) のような文末のスマイリーがある場合、関係ラベル `discourse` を用いて文の主辞に係る。

(4) は “your” を “ur” と表記した例であるが、その lemma が “you” であることと関係ラベルを用いて所有格代名詞であることが表現されている。同じ “ur” でも (5) のように “you are” を縮めた場合もあり、解析の際には文脈に応じて判断しなければならない。

4） 当初はフランス語・日本語など各言語の GSD コーパス（Google 由来）は CC BY-NC-SA ライセンスであった。

表 2 UD-EWT の中でインフォーマルな文体に見られる語の例。ここでは CoNLL-U フォーマットのうち ID、表層、*lemma*、UPOS、係り先、関係ラベルのみを示す。

(1)	22	thru	*through*	ADP	23	case
(2)	1-2	Gotta	_			
	1	Got	*get*	VERB	0	root
	2	ta	*to*	PART	3	mark
(3)	11	:)	:)	SYM	1	discourse
(4)	3	ur	*you*	PRON	4	nmod:poss
(5)	16-17	ur	_			
	16	you	*you*	PRON	18	nsubj
	17	are	*be*	AUX	18	aux

表 3 書き誤りの例。

(6)	14	excellant	*excellent*	ADJ	11	ccomp
(7)	18-19	dont	_			
	18	do	*do*	AUX	20	aux
	19	nt	*not*	PART	20	advmod
(8)	10	decide	*decide*	VERB	6	advcl
	11	whether	*whether*	SCONJ	18	mark
(9)	18	surv	*surv*	VERB	6	advcl
	19	ive	*ive*	X	18	goeswith
(10)	9	cge	*cage*	NOUN	4	obl
	10	ans	*and*	CCONJ	11	cc
	11	leave	*leave*	VERB	4	conj
	12	it	*it*	PRON	11	obj
	13	their	*there*	ADV	11	advmod

さらに、表 3 の例のような書き誤りが見られる。(6) は典型的なスペルミスであり、(7) はアポストロフィが欠落している[5]。(8) は、アノテーション自体は正常だが、原文では “decidewhether” と空白が欠落していた。逆に (9) は、動詞 “survive” の途中に空白が挿入され “surv ive” となっていた部分へのアノテーションで、UD の規約に従って `goeswith` のラベルで繋がれる 2 語となっている。(10) は連続する 5 語のうち 3 語にスペルの誤りがある[6]。

ウェブサイトの URL（EWT に 194 件出現）や e メールアドレス（10 件）の場合、空白が含まれないので一語として認識することは容易[7]だが、ファイル名（60 件以上）には “`Sanders Letter 4_28_00.doc`” のように空白を含むものもあり、一語とみなすか否かなどはコーパス内でも揺れがある。

また、特に e メール由来のコーパスには、区切り線 “−−−−−−−−−−−−−−−−−−−−−−” や、“`| | Tana.Jones@xxy| | | yzz.com | | |`” のような表の一部と思われるものなど、自然言語を逸脱したものも数多く見受けられる。

5） これが書き誤りなのか、書き手の意図した文体なのかを区別する明確な基準は無い。
6） 解読するアノテータさんの苦労が偲ばれる。
7） コーパスにアノテーションを付与する際にも、また構文解析器にとっても。

4 EWT コーパスの問題点

本節では、3 節で紹介したような事象により、コーパスを構文解析器の訓練や評価に用いる際に弊害が生じるケースを指摘する。

4.1 標準形への訂正による問題

表 2, 表 3 で見たように、くだけた記法や書き誤りがあった場合、正しい[8]形を想定した単語区切りや lemma がアノテートされて、それに基づいた品詞や依存構造が付与されている。

この原文と単語区切りのペアを用いて単語区切りを訓練すると、(8) のようなケースにより、空白が無くても単語を区切ろうとするモデルが作成される。UDPipe v1 [15] を UD-EWT v2.9 で訓練させたところ、“McDonalds” を “McDonald” と “s” に分割できるようになっていた[9]一方で、英字と数字が混在する製品名などの文字列が過剰に分割される傾向があった。

より大きな問題となるのが lemma の扱いである。本来は活用語の原形を与えるためのもので、辞書で対応するか、未知語に対して規則を作るか訓練させることにより解決する（英語の場合、‘s’ や ‘ed’ を外すなど）。UD-EWT の lemma にスペルチェッカーの要素が含まれているため、モデルが複雑になるほか、予期しない副作用が起きる。UDPipe のモデルで UD-EWT のテストデータを解析した結果の中では、“canon” に対して “*canion*”、“fares” に対して “*fary*” という lemma を与えるなど、深刻なエラーが生じていた。

4.2 文区切り推定の困難さ

UD-EWT には文書や段落を区切る情報が付与されているが、ニュース記事の例 (11) や e-メールの例 (12) の中には、以下のような単位がそれぞれ一つの段落として含まれている。

(11) Photo from Technology News Wiki Media Foundation, the group behind the Wikipedia ...

(12) Best regards, John Smith

(11) は “Photo from Technology News” が一つめの文、それ以降が次の文となっているが、表層上は句読点などの手がかりは入っていない。(12) も、“Best regards,” が一つの文となっており、名前の部分は別の文となっている。元の「現実世界の」文書に存在した HTML タグや改行などが失われていると推測されるが、これらを文区切りのタスクとして解決することは困難である。(12) と類似するケースは UD-EWT の訓練・テストデータの双方に含まれるので、訓練させることはできるが、その結果のモデルは UD-EWT へのバイアスがかかったものと言える。

4.3 MWT の特殊性

UD-EWT v2.6 以前では、トークンと語は一対一で対応していたが、UD-EWT v2.7 以降で複数語トークン (MWT) が導入され、“I’m” や “don’t” などの縮退形が 1 トークン・2 語として扱われるようになった。しかし、その他にも、(5) のように空白を持たないが複数の語に分割されるケースも MWT となっているため、トークナイズ・単語区切り・文の正規化などの処理が混在する状況となっている。これは、ドイツ語・フランス語等の UD において MWT が前置詞＋冠詞の縮退形に限られているのと異なる。特に、MWT は文字毎のオフセットや空白の有無を表現する手段が無いので、利用の際には注意が必要である。

5 ノイズ除去の実験

UD-EWT v2.8 では、書き誤りや空白の欠落といったノイズがある場合に本来の表記を示す情報が MISC フィールドに付加され、v2.9 ではそれがさらに拡充された。この情報を利用すれば、UD-EWT に含まれるこの種のノイズを除去することができる。本節では、ノイズを除去したコーパスで解析器を訓練した場合の変化を調べる。

5.1 コーパスの変換

UD-EWT の dev ブランチ（2021 年 12 月 20 日現在）の訓練・検証データに対して、表 4（作例）のように、3 種類の変換を行う。まず、MISC フィールドで CorrectForm が指定されている場合（訓練データ中に 533 件存在）、表層形をその表記に改める。また、CorrectSpaceAfter=Yes が存在する場合（同 81 件）、直後に空白を挿入する。`goeswith` ラベルで繋がる 2 語（3 語以上や、XPOS タグが ADD の場合を除く）に対しては、それらを 1 語に結合して、原文の空白を除去する（同 39 件）。このようにして各語の表層部分と、文全体の表層形も書き換えることに

8） 何をもって「正しい」かというと、UD の構文構造に即したアノテーションがしやすい形、というのが近い。

9） これにより、UD-EWT コーパス上での精度は上がる。この分割に実用上の意味があるかはわからない。

表 4　ノイズ除去の例。MISC フィールドと goeswith ラベルを用いて左（品詞タグは省略）から右に変換する。

```
# text = Its a terrrible air line$ everynight
1-2   It's       _
  1   It         it          5   root
  2   s          be          5   cop          CorrectForm='s
  3   a          a           5   det
  4   terrrible  terrible    5   amod         CorrectForm=terrible
  5   air        air         0   root
  6   line$      line        5   goeswith
  7   every      every       8   det          CorrectSpaceAfter=Yes
  8   night      night       5   advmod
```

```
# text = It's a terrible airlines every night
1-2   It's       _
  1   It         it          PRON   5   root
  2   's         be          AUX    5   cop
  3   a          a           DET    5   det
  4   terrible   terrible    ADJ    5   amod
  5   airlines   airline     NOUN   0   root
  6   every      every       DET    7   det
  7   night      night       NOUN   5   advmod
```

より、ノイズを除去したコーパスを作成し、これを UD-EWT-clean と呼ぶ。

5.2　モデルの再訓練と実験結果

UD-EWT v2.9 と、UD-EWT-clean を用いて、以下の 3 つの解析器の訓練を行った。

- UDPipe v1 [15] のトークナイザー、品詞タグ付け、係り受け解析。UD v2.5 の訓練に使われたパラメータと embedding を使用。
- BERT [3] を用いた内製の品詞タグ付け器[10]。
- Stanza [11] v1.3 の、トークナイザー、MWT、lemma、品詞タグ付け、係り受け解析のモデル。

それぞれを以下の 3 つのデータで評価する。

EWT　UD-EWT v2.9 のテストデータの 2,077 文。
news　上記のうち、ニュース記事からなる 284 文。
ON　OntoNotes コーパスを UD 形式に変換 [1] したコーパスのテストデータ 9,971 文。

UD の各指標 (F1 値) を表 5 に示す。UDPipe においては、ノイズ除去により EWT のテストデータ上の一部の評価値は微減してしまった。これは、EWT のテストデータ中の訓練データと類似した事象に対応できなくなったためだと思われる。一方、EWT とは独立に作られた ON のデータでは、UDPipe, BERT PoS では全指標においてノイズ除去による効果が出ている[11]。比較的インフォーマルな文が少ない news については EWT と ON の中間程度となっている。なお、全体的に、Lemma についてはノイズ除去の効果が大きい。これは、書き誤りの要素を排除することにより 4.1 節で見たようなエラーが防げていることによる。但し、Stanza の結果は UDPipe と逆の傾向にある。この原因は調査中である。

10）　単語区切り済みの入力に対して XPOS を付与する。
11）　EWT の訓練・テストデータの性質によらずに性能を比較できるので、この結果は特に重要だと考えている。

表 5　UD EWT v2.9 と clean で訓練した際の結果。太字は有意差をもって高い側の値を示す。

解析器	評価	指標	EWT2.9	EWT-clean
UDPipe	EWT	Word	98.89	98.78
		XPOS	**92.87**	92.50
		Lemma	95.28	95.34
		LAS	**76.70**	76.09
	news	Word	98.93	98.75
		XPOS	91.47	91.34
		Lemma	94.25	**94.49**
		LAS	73.73	74.01
	ON	Word	98.51	**99.41**
		XPOS	92.41	**93.43**
		Lemma	94.85	**95.79**
		LAS	72.65	**73.08**
BERT PoS	EWT	XPOS	96.09	95.86
	news	XPOS	95.62	95.26
	ON	XPOS	95.18	**95.33**
Stanza	EWT	Word	**98.98**	99.17
		XPOS	95.38	95.46
		Lemma	96.74	**96.99**
		LAS	86.09	86.31
	news	Word	98.49	98.64
		XPOS	94.20	94.22
		Lemma	95.63	95.97
		LAS	84.24	84.19
	ON	Word	**98.88**	98.53
		XPOS	**95.10**	94.08
		Lemma	**96.31**	96.03
		LAS	**79.45**	78.23

6　まとめ

本稿では、UD_English-EWT のコーパスについて、その成り立ちや含まれる事象、そして分析を難しくする要素を紹介した。また、書き誤り等のノイズを除去したコーパスを作成し、それを解析器の訓練に用いたところ、解析器と評価データによっては、ノイズを除去することにより性能が高まることがわかった。UD-EWT に閉じて訓練・テストを行うシェアドタスクの状況と異なり、実用の場面を考慮して構文解析器を設計・評価する際には、コーパスの性質や問題点を考慮して、ベンチマークの方法に留意する必要があることが示唆される。

参考文献

[1] Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy, May 2006. European Language Resources Association (ELRA).

[2] Marie-Catherine De Marneffe and Christopher D Manning. Stanford typed dependencies manual. Technical report, Technical report, Stanford University, 2008.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[4] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020.

[5] Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. OntoNotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pp. 57–60, New York City, USA, June 2006. Association for Computational Linguistics.

[6] Christopher D. Manning. Part-of-speech tagging from 97% to 100%: is it time for some linguistics? In *Proceedings of the 12th international conference on Computational linguistics and intelligent text processing - Volume Part I*, CICLing'11, pp. 171–189, Berlin, Heidelberg, 2011. Springer-Verlag.

[7] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, Vol. 19, No. 2, pp. 313–330, 1993.

[8] Minh Van Nguyen, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen. Trankit: A light-weight transformer-based toolkit for multilingual natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pp. 80–90, Online, April 2021. Association for Computational Linguistics.

[9] Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher 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 10th International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia, 2016.

[10] Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020)*, pp. 4034–4043, Marseille, France, May 2020. European Language Resources Association.

[11] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 101–108, Online, July 2020. Association for Computational Linguistics.

[12] Sebastian Schuster and Christopher D. Manning. Enhanced English Universal Dependencies: An improved representation for natural language understanding tasks. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pp. 2371–2378, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA).

[13] Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, 2014.

[14] Milan Straka. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pp. 197–207, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[15] Milan Straka, Jan Hajič, and Jana Straková. UDPipe: trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia, 2016. European Language Resources Association.

[16] Daniel Zeman, Filip Ginter, Jan Hajič, Joakim Nivre, Martin Popel, and Milan Straka. CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, Brussels, Belgium, 2018.

[17] 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.

[18] 金山博, 岩本蘭, 村岡雅康, 大湖卓也, 宮本晃太郎. 名詞句の処理に頑健な構文解析器. 言語処理学会第 27 回年次大会予稿集, March 2021.