

依存構造から句構造への変換による多言語モデリングに向けて

神藤 駿介^{†‡} 能地 宏[‡] 宮尾 祐介[†]

[†] 東京大学大学院 情報理工学系研究科

[‡] 産業技術総合研究所 人工知能研究センター

{kando.s, yusuke}@is.s.u-tokyo.ac.jp

{s-kandou, hiroschi.noji}@aist.go.jp

1 はじめに

伝統的に自然言語は階層構造を有していると言われており、自然言語理解においてはその構造の把握が重要であるとされている。近年は言語モデルの発達が著しいが、それらが自然言語の階層構造を十分に捉えているとは限らず、正確な自然言語理解が出来ているかは議論の余地がある。Linzenら [1] は、統語構造の把握を要するタスクを定義することでモデルの統語構造把握能力の検証を試みている。Huら [2] はそのようなテストを多数含むベンチマークを提案し、LSTM 言語モデルはパープレキシティの面では高い性能を有しつつも、統語構造の把握能力は低いことを示している。この問題を解決する手段として、Recurrent Neural Network Grammar (RNNG, [3]) のように学習の際に句構造の情報を活用することが考えられる。RNNG は [2] においても高い統語構造把握能力を有していることが示されている。

現状 RNNG は英語のテキストを元に学習されたモデルによる研究がほとんどであり、あらゆる言語で同じような性能を発揮するかは分かっていない。そのため、RNNG を多言語でモデリングし調査することには意義がある。その実現にあたり、代表的な多言語ツリーバンクである Universal Dependencies (UD, [4]) を活用することが考えられるが、UD は依存構造注釈付きのツリーバンクであるため、RNNG の学習に活用するためには言語非依存ななんらかの方法で句構造へ変換する必要がある。

そこで本研究では、依存構造から句構造への変換手法を複数試し、RNNG の学習においてどのような構造が適するかを比較・検討する。今回はそれらの変換手法を UD のアノテーション形式に基づいた英語と日本語のツリーバンクに適用してモデルの学習を行った。これに加え、代表的な句構造ツリーバンクである Penn Treebank (PTB, [5]) での学習も行っ

た。これらのモデルのパープレキシティを評価することで、各変換手法の質の比較・検討を行った。さらに、英語を元にしたモデルに関しては統語構造把握能力の評価を行った。実験を通して、依存構造のある種の句構造へ変換することで、PTB で学習したモデルに匹敵する性能を持つモデルが構築できることを示した。この変換は言語非依存であるため、UD のあらゆる言語のツリーバンクに対して適応可能であり、本研究の拡張性を見込むことができる。

2 関連研究

2.1 Recurrent Neural Network Grammar

RNNG [3] は句構造生成モデルであり、単語列 \mathbf{x} と句構造 \mathbf{y} の同時確率分布 $p(\mathbf{x}, \mathbf{y})$ を推定することで、言語モデルと句構造解析器を同時に学習することを目指す。 $p(\mathbf{x}, \mathbf{y})$ は、 \mathbf{y} を得る句構造解析のアクション列 $\mathbf{a} = (a_1, a_2, \dots, a_n)$ を用いて次のように推定する：

$$p(\mathbf{x}, \mathbf{y}) = \prod_{t=1}^n p(a_t | a_1, \dots, a_{t-1})$$

RNNG は、 $\arg \max_{\mathbf{y}} p(\mathbf{y} | \mathbf{x})$ を計算することで句構造解析器として、 $p(\mathbf{x}) = \sum_{\mathbf{y}} p(\mathbf{x}, \mathbf{y})$ を計算することで言語モデルとして活用することができる。これらの確率を解析的に求めることは難しく、様々な推定法が提案されている [3, 6, 7]。

RNNG は句構造解析器や言語モデルとして高い性能を持つだけでなく、統語構造の把握能力が高いこと [8]、人間の認知モデルとしての妥当性が高い [9] など、様々な利点が報告されている。

2.2 依存構造から句構造への変換

自然言語の構文構造を表現する主要な枠組みとして、依存構造と句構造がある。両者は異なる構造で

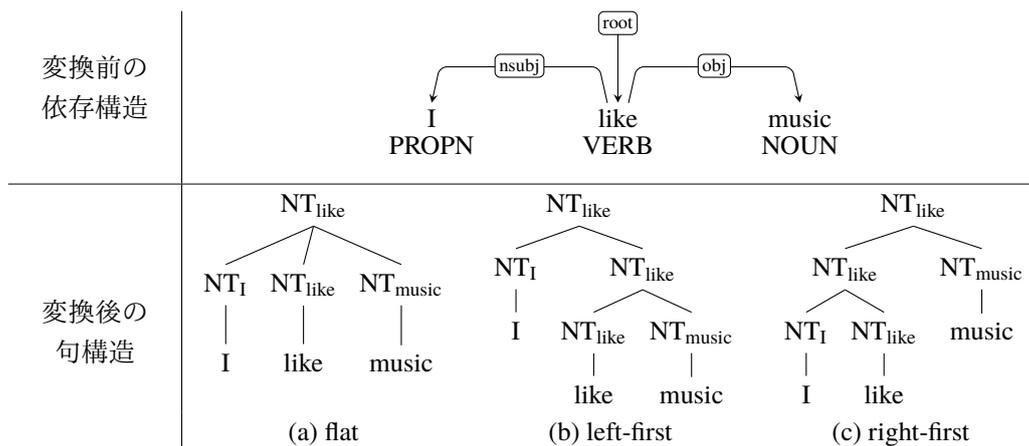


図1 依存構造から句構造への変換アルゴリズム

	NT _I	NT _{like}	NT _{music}
X-label	X	X	X
POS-label	PROPN	VERBP	NOUNP
DEP-label	root	nsubj	obj

図2 非終端記号ラベルの付け方

ありながら互いに関連し合っており、一方から他方へ変換する手法に関する研究も幅広く行われている。依存構造から句構造への変換手法は大きく分けて以下の二つである：

1. 句構造ツリーバンクを外部知識として活用し、**data-driven** に変換器を構築する手法 [10, 11]
2. 外部知識を用いず、依存構造から一定のアルゴリズムによって変換する手法 [12, 13]

1. はより頑健で精度の高い変換を可能とするが、依存構造単体から変換することが出来ない。一方 2. は依存構造単体から変換できるが、得られた構造が妥当なものであるかについては疑問の余地が残る。

本研究は UD を活用した多言語のモデリングを目標としているため、1. のように対応する句構造ツリーバンクを要する手法は相性が悪い。そのため、2. の手法をベースにした変換を行う。

3 変換手法

依存構造を句構造へ変換するにあたり、[13] で紹介されている 3 つのアルゴリズムを利用した。これに加え、句構造の非終端記号ラベルの付け方についても 3 通り試した。すなわち、合計で $3 \times 3 = 9$ 通りの変換手法を試した。

図 1 に 3 つの変換アルゴリズムの概略を示す。これらのアルゴリズムは全てトップダウンに句構造を形成するものであり、以下のような共通の取り決め

を設ける：

- はじめに、主辞に対応する非終端記号 NT_{like} を導入する。
- 依存構造上の子をも句構造に導入する際は、NT_I や NT_{music} のように、対応する非終端記号を導入する。
- 現在注目している語について、依存構造上に子が存在しないときは、その語を終端記号として導入する。

以下、ある単語（ここでは主辞 like）の左右に子が存在しているケースを元にして各アルゴリズムの動作を説明する。(a) flat では、現在注目している単語 like とその子を全て一度に句構造における子とする。この変換においては、各非終端記号は任意の個数の子を持つことができ、全体として平らな構造になりやすい。(b) left-first では、単語の左右に子が存在する場合に、左の子から順に句を形成する。すなわち、まず NT_I を左の子として導入し、終端記号 I の導入が完了した後に、右の子 NT_{music} を導入する。(c) right-first は (b) と逆に、右の子から順に句を形成する。(b) (c) においては、完成する句構造は完全二分木となる。

非終端記号ラベルの付け方は図 2 の通りで、それぞれラベルとして X、POS タグ、依存ラベルを使用する。

4 実験設定

本研究の目標は、3 で紹介した手法で依存構造を句構造に変換して得られたツリーバンクを用いて RNNG を学習し、その性能を比較し、変換の妥当性を検証することである。そのために以下のような実験設定を施した。

表1 データセットの統計

	訓練	検証	テスト	語彙数
English-EWT	11945	1949	2030	9363
Japanese-GSD	7006	500	541	8664
PTB-to-UD	39712	1695	2398	23727
PTB	39832	1700	2416	23767

4.1 データセット

依存構造ツリーバンクとして、UDのEnglish-EWT及びJapanese-GSDを使用した。これに加え、伝統的な句構造ツリーバンクであるPenn Treebank (PTB)と、それをUD形式の依存構造ツリーバンクに変換したもの (PTB-to-UD) も使用した。PTB-to-UDを得るにあたっては [14] の手法を活用した。PTB-to-UDに対して句構造への変換を施したツリーバンクと純粋なPTBとの結果を比較することで、変換アルゴリズムの妥当性を検証する。

データセットの統計情報 (訓練・検証・テストセットの文章数、及び語彙数) を表1に示す。語彙については、訓練データ中で2回以上出現した語を使用した。なお、本研究においては依存構造が交差を含むようなデータは除外している。これは、交差を含む依存構造においては我々の変換アルゴリズムが正常に動作しないためである。今回用いたツリーバンクに関しては、交差を含むデータは1%にも満たなかったため、大きな影響は無いと判断した。

4.2 学習の詳細

RNNGの学習においては、基本的には元論文 [3] と同様のハイパーパラメータを使用した。最適化で用いるアルゴリズムを変えており、元論文では学習率0.1の確率的勾配降下法を用いているが、本研究では学習率0.001のAdamを用いた。学習のエポック数は、English-EWT及びJapanese-GSDでは18、PTB及びPTB-to-UDでは40とした。

4.3 評価

本研究では、モデルの言語モデルとしての性能、及び統語構造把握能力の両面で評価を行った。言語モデルとしての性能は、各データセットの検証データにおけるパープレキシティ (ppl) を評価指標とした。pplの算出においてはword-synchronous beam search [6] を採用し、ビーム幅 $k = 100$ 、単語ビーム幅 $k_w = 10$ 、fast-track 数 $k_s = 1$ とした。

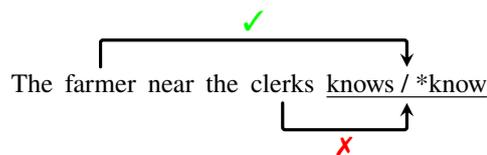


図3 SyntaxGymのテスト例 (一致)

統語構造把握能力は、[2] で提案されているベンチマーク (SyntaxGym) を用いて評価した。SyntaxGymは、英語テキストの統語構造の把握を必要とする34のテストをまとめたものであり、取り扱っている言語現象に応じてAgreement (一致) やCenter Embedding (中央埋め込み) などと言った6つのカテゴリに分類される。図3に一致のテストの一例を示す。この例では、下線部の動詞の予測確率を比較し、 $P(\text{knows}) > P(\text{know})$ となれば正解とみなす。本研究では6つの各カテゴリについての精度を報告する。ランダムベースラインは各テストによって異なるが、総じて25%以下となっている。

5 結果と議論

5.1 言語モデルとしての性能 (ppl) の比較

図4に各ツリーバンクで学習したモデルのパープレキシティを示した。変換アルゴリズムに依る差を見ると、どのツリーバンクにおいてもpplの値はflat、left-first、right-firstの順で大きくなる。すなわち、言語モデルの性能としてはflatが最善であることが分かる。これは、left-firstやright-firstの変換を施すと句構造は完全二分木となるため、句構造が深くなりやすく、構文解析のアクションが煩雑かつ大量になることに起因すると思われる。

以下、flatな変換に注目してラベルの付け方に依る差を考察する。英語テキストに関してはどのラベルを用いてもpplには差がないが、日本語テキストの場合はX、POS、DEPの順で大きくなる。日本語テキストは図6のような左枝分かれ構造になりやすいため、トップダウンな句構造解析においてははじめに非終端記号を複数導入する必要があり、解析が難しくなる。そのため、非終端記号の導入時のラベルの選択肢が少ない順 (X、POS、DEPの順) がpplに反映されていると考えられる。

注目すべきは、PTB-to-UDをflatなアルゴリズムを用いて変換して学習に用いた場合、そのpplは純粋なPTBで学習したモデルのpplとほぼ同程度である点である。RNNGの言語モデルとしての性能は、

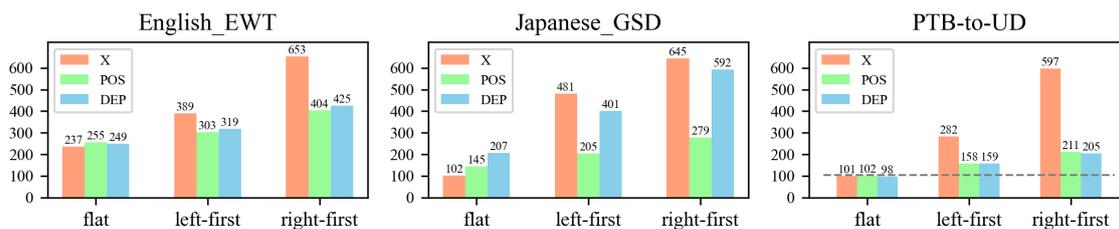


図 4 パープレキシティ (ppl) の比較。PTB-to-UD における点線は、PTB で学習したモデルの ppl (105) を示す。

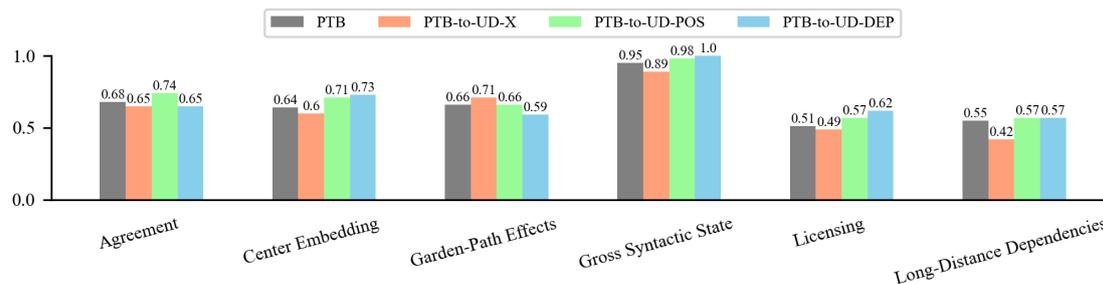


図 5 SyntaxGym を用いた統語構造把握能力の測定結果

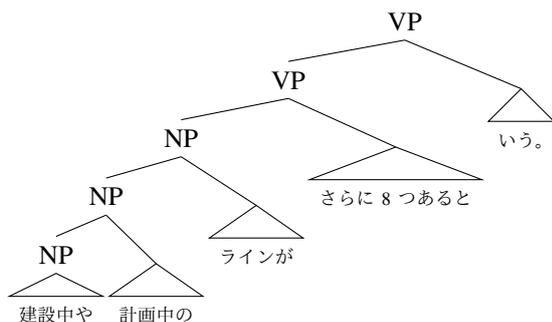


図 6 Japanese-GSD を句構造に変換した出力例

元々句構造ツリーバンクとして作成された PTB を使っても、依存構造から flat な句構造に変換して得られたツリーバンクを使ってもほぼ変わらないことを示している。

5.2 統語構造把握能力の比較

5.1 では、PTB と PTB-to-UD を flat な構造に変換したツリーバンクとで得られるモデルの言語モデルとしての性能差はほぼ無いことを示した。では、これらのモデルの統語構造把握能力にはどのような差があるだろうか。図 5 に、両モデルの SyntaxGym における評価の結果を示した。図から分かる通り、全体として PTB-to-UD を元に得たモデルは PTB を元に得たモデルに匹敵する精度を記録している。特に、ラベルの付け方として POS を選択した場合、全ての評価カテゴリにおいて PTB をベースにしたものと同じかそれ以上の性能を示している。

各ラベル付け手法についても性能を比較すると、Garden-Path Effects を除く全てのカテゴリにおいて POS と DEP が X より高い性能を示している。このことから、三者は ppl の側面ではほとんど差がなかったが、統語構造把握能力に関してはラベルを明確に導入した方が性能が高まると言える。

6 終わりに

本研究では、依存構造を句構造に変換する複数の手法を試して RNN を学習し、その性能を比較・検討した。特に、flat な句構造に変換して学習に用いることで、PTB 形式の句構造を元に学習したモデルと同等かそれ以上の性能を誇るモデルが構築できることを示した。変換アルゴリズムは言語非依存に適用可能であるため、今後 UD のあらゆる言語のツリーバンクに対して同様の実験を行い、本研究の手法の適用範囲を調査していきたい。多言語での実験においては、統語構造把握能力の評価がネックとなる。例えば [15] に紹介されている手法を用いることが考えられる。また、今回施した変換は言語非依存ではあるもののかなりナイーブである。多言語に適用可能で、なおかつ得られるモデルの性能がより向上するような変換手法についても調査を続けていきたい。句構造解析ではなく依存構造解析を用いたモデリングも提案されており [16]、この手法との比較も今後行っていきたい。

参考文献

- [1] Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the ability of lstms to learn syntax-sensitive dependencies. *TACL*, Vol. 4, pp. 521–535, 2016.
- [2] Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. A systematic assessment of syntactic generalization in neural language models. In *ACL*, 2020.
- [3] Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. Recurrent neural network grammars. In *NAACL HLT*, 2016.
- [4] Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajic, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis M. Tyers, and Daniel Zeman. Universal dependencies v2: An evergrowing multilingual treebank collection. *CoRR*, Vol. abs/2004.10643, , 2020.
- [5] Mitchell P. Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. The penn treebank: Annotating predicate argument structure. In *Human Language Technology, Proceedings of a Workshop*, 1994.
- [6] Mitchell Stern, Daniel Fried, and Dan Klein. Effective inference for generative neural parsing. In *EMNLP*, 2017.
- [7] Benoît Crabbé, Murielle Fabre, and Christophe Pallier. Variable beam search for generative neural parsing and its relevance for the analysis of neuro-imaging signal. In *EMNLP-IJCNLP*, 2019.
- [8] Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. Lstms can learn syntax-sensitive dependencies well, but modeling structure makes them better. In *ACL*.
- [9] John Hale, Chris Dyer, Adhiguna Kuncoro, and Jonathan Brennan. Finding syntax in human encephalography with beam search. In Iryna Gurevych and Yusuke Miyao, editors, *ACL*, 2018.
- [10] Lingpeng Kong, Alexander M. Rush, and Noah A. Smith. Transforming dependencies into phrase structures. In *NAACL HLT*, 2015.
- [11] Young-Suk Lee and Zhiguo Wang. Language independent dependency to constituent tree conversion. In *COLING*, 2016.
- [12] Fei Xia and Martha Palmer. Converting dependency structures to phrase structures. In *HLT*, 2001.
- [13] Michael Collins, Jan Hajic, Lance A. Ramshaw, and Christoph Tillmann. A statistical parser for czech. In *ACL*, 1999.
- [14] Sebastian Schuster and Christopher D. Manning. Enhanced english universal dependencies: An improved representation for natural language understanding tasks. In *LREC*, 2016.
- [15] Aaron Mueller, Garrett Nicolai, Panayiota Petrou-Zeniou, Natalia Talmina, and Tal Linzen. Cross-linguistic syntactic evaluation of word prediction models. In *ACL*, 2020.
- [16] Austin Matthews, Graham Neubig, and Chris Dyer. Comparing top-down and bottom-up neural generative dependency models. In *CoNLL*, 2019.