言語のインクリメンタルな処理の仕組みは普遍的か？：投機性による parsing strategy 再考

石井太河　宮尾祐介
東京大学
{taigarana,yusuke}@is.s.u-tokyo.ac.jp

概要

自然言語はインクリメンタルに読み書きされるが、異なる言語でも、インクリメンタルな処理の仕組みは言語普遍的なのだろうか？本研究では、系列・構造の両方をインクリメンタルに予測する統語的言語モデルが「系列の背後にある統語構造をどの程度投機的に予測するか」をパラメタとし、次トークン予測や構文解析において最適なパラメタが言語共通かを分析する。実験の結果、最適な戦略の言語共通性はタスクやビームサイズにより異なることが観察され、人間と言語モデルの処理メカニズムの違いに関する示唆が得られた。

1　はじめに

世界に数多存在する自然言語、そのほとんどに共通する性質として、「系列であること」がある。自然言語はインクリメンタルに読み取られ、生成される。一方で、自然言語の背後には非線形な統語構造や意味構造といった構造があると考えられており、その構造は言語によって多岐にわたる [1]。

では、英語、日本語といった一見異なる言語に関して、単に「系列である」以上の共通点や「統語構造が違う」以上の相違点を記述することはできるだろうか？これまでの研究では、単語の頻度（Zipf 則 [2]）や長相関 [3]、情報密度 [4] といった観点から言語普遍的な性質に関して議論がなされている。一方で、言語間の相違点に関しては、例えば、統語構造の木形状の違いなどがある [5]。本研究では、系列とその背後にある非線形な統語構造を繋ぐ中間的な性質として、「トークン列がインクリメンタルに処理される際に、背後の統語構造はどのように推測されていくのが最適か？」という点に着目する。

図 1にあるように、同じトークン列・構造に対しても、それらをインクリメンタルに処理する方法は

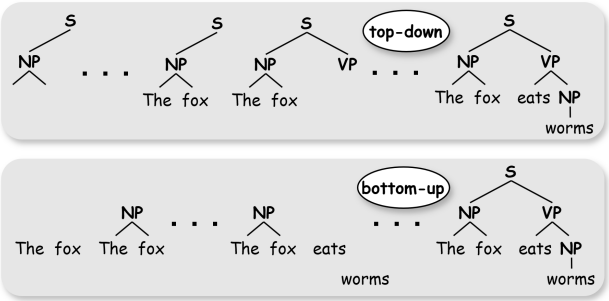

図 1　構文解析で使用される代表的な parsing strategy の例

1 つだけではなく、「どのタイミングで構造を予測するか」により様々にありえる [6]。このような処理の仕方の違いは、構文解析分野において parsing strategy（以下、単に戦略と呼ぶ）の違いとして扱われる [6, 7]。例えば、図 1は構文解析において最もよく使用される 2 つの戦略を示す。Top-down は「トークンの前に構造を予測する」戦略であり、bottom-up は「トークンの後に構造を予測する」戦略となっている。本研究では、これらの戦略の違いを、「投機性」、すなわち「どの程度トークンを先に予測した後に、対応する構造を予測するか」としたパラメタ化を提案する。例えば、top-down は構造予測にトークン情報を使えず、続くトークンによっては予測した構造が間違いとなる可能性があるため、投機性が高い。Bottom-up は投機性が低く構造予測にトークン情報を使用できるものの、逆にトークン予測に構造情報を使用できないというトレードオフがある。

本研究では、10 の言語に対して、投機性の異なる様々な戦略で統語的言語モデルを教師あり学習し、次トークン予測や構文解析タスクでの「最適な戦略」が言語共通であるかどうかを分析する。

2　背景

統語的言語モデリングの先行研究にならい [8, 9]、本研究では、統語構造として特に句構造を扱う。句構造とは、葉がトークンであり、中間ノードが句カ

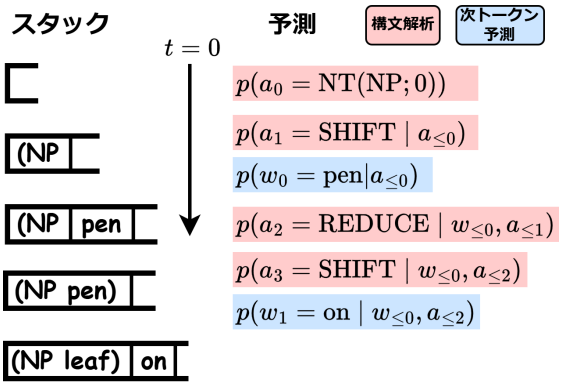

図 2　統語的言語モデリングの例。構文解析アクションの予測と次トークン予測を順に行っている。

テゴリを表すラベル付き木である。

2.1　統語的言語モデル

統語的言語モデルとは、入力系列 x に対し、その背後にある統語構造 y も明示的に予測し、同時分布 $p(x, y)$ を計算するモデルであり、$-\log p(x, y)$ を最小化するように学習される。

統語的言語モデルの実装として最も使用されるのは、RNNG [8, 10] や PLM [11] といった shift-reduce パーザをベースとしたモデルである。これらのモデルは、スタックを用いた構文解析と次トークン予測を順に行う（図 2）。スタックを用いた構文解析は、スタック上の操作として定義されたアクションの予測として行われる。例えば、top-down 戦略の場合、アクションの定義は以下のようになる：

- NT(X)：スタックトップに “(X” を追加し、ラベル X の句の左端を作成
- SHIFT：スタックトップに次トークンを追加
- REDUCE：スタック上一番上の開いている句を閉じ、一つの要素にまとめる

既存研究では、戦略ごとに専用のアクション集合が定義されている [12]。

2.2　既存研究との関係

構文解析の既存研究では、left-corner 戦略が言語によらず top-down・bottom-up よりも効率が良く、認知的に妥当とされる [6, 7]。[1] しかしながら、left-corner の効率の良さ、あるいは認知的妥当さは、あくまで「中央埋め込み」という特殊な文構造に対する最大スタックサイズに関しての議論にとどまっている [6, 13, 7, 14]。また、実際のデータでは、left-corner が他の戦略よりもスタック効率が大幅に良くなるような「深い中央埋め込み」の文は数少なく [15]、スタック効率がどれほど「言語の処理メカニズム」に影響を与えるかは自明ではない。加えて、Resnik [7] も主張している通り、戦略によるスタック効率の違いは、モデルの実装にもよるため、戦略そのものの特徴づけとして適切でないという問題もある。[2] これらに対し、本研究では、スタック効率ではなく「投機性」を戦略の特徴づけとして分析を行う。

また、統語的言語モデリングの既存研究では、言語モデリングや構文解析などの下流タスクで戦略を比較する研究もあるが [12, 16, 17]、実装の簡単さから top-down・left-corner・bottom-up といった限られた戦略しか扱われておらず、戦略の最適性を分析するには網羅性が足りていない。本研究では、既存研究で使用されている統語的言語モデル [8, 10] を一般化し、より広範な戦略を扱えるように拡張する。

3　統語的言語モデルの一般化

2.1 節で述べたように、既存研究では戦略ごとに専用のアクション集合を定義していた。本研究ではモデルの扱うアクション集合を一般化し、一つのアクション集合で様々な戦略を表現可能にする。具体的には、以下のようにアクション集合を定める：

- NT(X;n)：スタックの上から n 番目の位置に “(X” を追加し、カテゴリ X の句の左端を作成。ただし、新たな句はすでに開いている句より深く生成できないように制限する [3]
- SHIFT：スタックトップに次トークンを追加
- REDUCE：スタック上一番上の開いている句を閉じ、一つの要素にまとめる
- FINISH：構文解析を終了する

FINISH は、top-down 以外の戦略では必要になる [12]。

このように定められたアクション集合はアクションの選び方を制限することで様々な戦略を扱うことができる。例えば、句を開く位置が常に $n = 0$、すなわちスタックトップであれば、top-down になり、NT(X;n) の直後に REDUCE を行うようにすれば、子を n 個もつ句の予測が常に「全ての子を予測した後」になるため、bottom-up になる。なお、このアクション集合が扱える戦略のクラスは syntax-directed

1）　ここで、left-corner 戦略とは、句の左端のトークンを読んだ直後にその句構造を予測するという戦略である [6]。

2）　例えば、[6] の議論と異なり、統語的言語モデルとして使われる RNNG [8, 10] は top-down 戦略の場合でも右分岐構造に $O(n)$ のスタックサイズが必要な実装となっている。

3）　この制限は実装上の簡潔さのためであり、実装を少し改良することでさらに一般化が可能である。

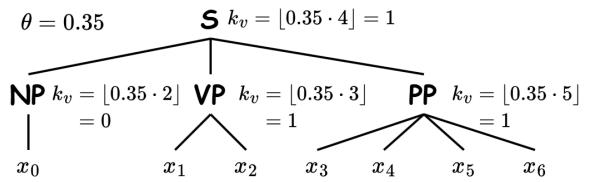

図 3　学習データの生成例

戦略 [6] のクラスより真に大きくなる。

4　投機性の異なる戦略の生成

本研究では、投機性をパラメタとし、様々な戦略の下で「正解の木構造を導出するアクション系列」を生成し、モデルを教師あり学習する。教師データとなるアクション系列を得るには、正解の木構造の各ノード v に対し、「k_v 個目の子が作られた後にその親ノードを作る」ような k_v を決めればよい [6]。[4] 本研究では簡単のため、各ノード（句）を予測する際の投機性がノードによらず一定であるような戦略を考える。直感的には、各ノードで「子の何割が作られてから親を作るか」を $\theta \in [0,1]$ の実数で定め、学習データ作成の際の投機性パラメタとする。子を n_v 個持つノード v に対し、k_v は $k_v = \lfloor \theta \cdot (n_v + 1) \rfloor$ で計算する。図 3に学習データの生成例を示す。

5　実験設定

5.1　データセット・戦略

実験では、英語（Penn Treebank [18]）、中国語（Chinese Treebank [19]）、フランス語・ドイツ語・韓国語・バスク語・ヘブライ語・ハンガリー語・ポーランド語・スウェーデン語（SPMRL [20]）の 10 言語のツリーバンクを用いる。前処理として、能地ら [10] にならい、POS タグは削除し、単語は subword に分割した。本稿での評価データはすべて validation データセットを使用する。また、アクション集合を小さくし、モデルの学習を簡単にするため、NT(X;n) アクションの n を上限 10 に制限する。これにあたり、戦略間での構文解析可能性に差が無いように、学習・評価データともに、$n \leq 10$ で全ての戦略で正解構造が導出可能なデータだけを使用する。

実験で分析対象とする戦略には、投機性パラメタ $\theta \in \{0.0, 0.26, 0.35, 0.65, 0.74, 0.99\}$ で表現される 6 つの戦略を用いる。なお、0.0 は top-down、0.99 は bottom-up に対応する。[5]

5.2　モデル

モデルとしては、統語的言語モデルとしてよく用いられる Recurrent Neural Network Grammar（RNNG）[8] を一般化したアクション集合を扱えるように拡張した。実装は、[10] をベースとした。その他の設定については付録 B に記す。

5.3　評価

本研究では、次トークン予測のパープレキシティと構文解析精度（句カテゴリの一致を考慮した F1 スコア）を通してモデル評価を行う。前者は低い方が良く、後者は高い方が良い。評価には、学習したモデルで word synchronous beam search [21] で推論した最も良いアクション系列を用いる。実験では、異なるビームサイズ $b \in \{50, 200, 800\}$ でそれぞれ評価する。なお、推論時間短縮のため、fast track selection を $b/50$ とし、SHIFT アクション間の最大アクション数を 20 に制限し、word beam size は常に b とした。

次トークン予測には、SHIFT アクション直前のアクションまでの条件付き確率として予測確率を計算する（図 2の青い部分）。[6] この場合、構文解析アクションの予測確率（図 2の赤い部分）は次トークン予測に反映されないが、構文解析アクション自体は「系列のみの次トークン予測モデルにおける内部状態の更新」に対応するものとして考えられる。

6　実験結果・考察

図 4に、英語・中国語・フランス語・ドイツ語・韓国語の結果を示す。その他の言語に関する結果は付録 C に示すが、全体としての結論は変わらない。

6.1　最適な戦略は言語共通か？

実験の結果、最適な戦略の言語共通性がタスクとビームサイズにより異なる傾向が見られ、投機性が言語によって異なることが観察された。ここでは主に次トークン予測のパープレキシティ（図 4上段）

4）　Abney ら [6] はノードごとではなく、文法規則 r ごとに k_r を定めているので本研究はこれを一般化していると言える。

5）　その他中間のパラメタは子数 n_v が 2, 3, 4 の時に句を開くタイミングが変化するように選択した。

6）　既存の統語的言語モデリングの研究では、ビーム候補 $\mathcal{Y}$ や外部パーザーで推論された構造候補 $\mathcal{Y}$ を用いて同時確率の周辺化 $p(x) = \sum_{y \in \mathcal{Y}} p(x, y)$ を近似している。しかし、これはあくまで近似であり、ビームサイズが小さいなど候補が少ない場合は、周辺化が十分でなく、純粋に次トークン予測の精度を評価するのが難しくなる。そのため本研究ではアクションの条件付き確率として次トークン予測を評価する。

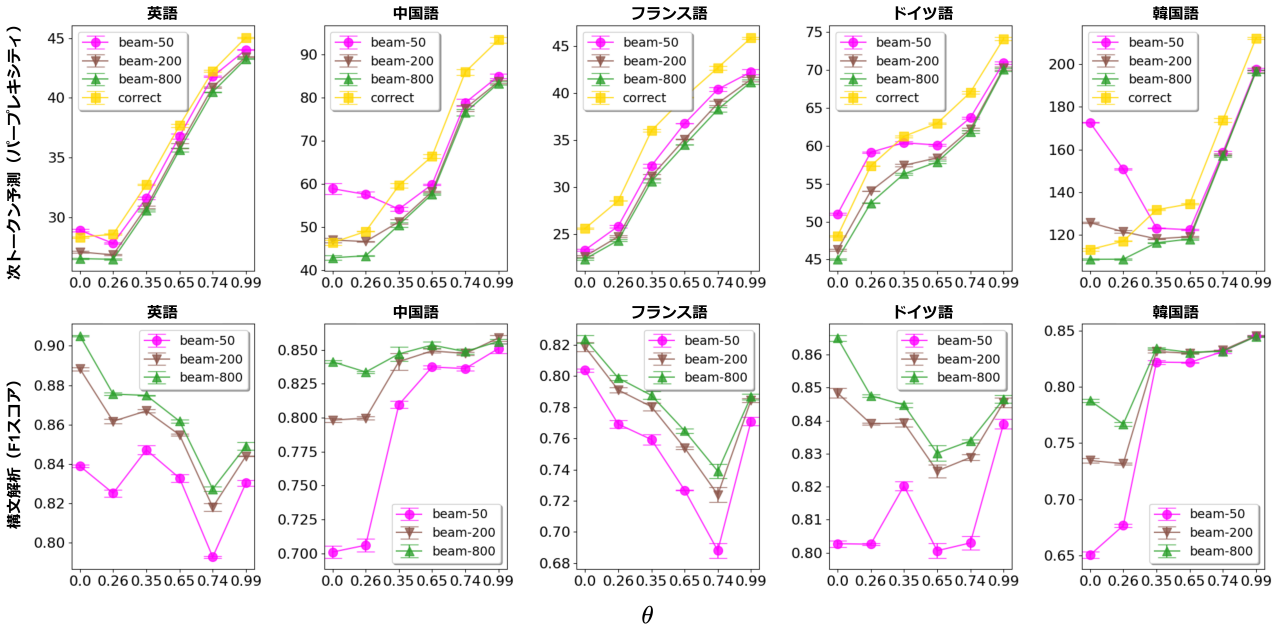

図 4　英語・中国語・フランス語・ドイツ語・韓国語の結果。エラーバーは標準誤差を表す。

について議論する。図 4の beam-800 のようにビームサイズが大きい場合、次トークン予測においては概ねどの言語でも投機性の高い戦略が最適となった。一方で、ビームサイズが小さい場合（図 4の beam-50）言語により結果が大きく異なった。まず、英語・フランス語・ドイツ語は次トークン予測性能は概ね投機性に比例しているのに対し、中国語・韓国語では、投機性が高すぎる場合（$\theta = 0.0, 0.26$）に逆に性能が悪くなっており、中程度の投機性の戦略（$\theta = 0.35, 0.65$）で最適となっている。

6.2　言語モデルと人間では最適な戦略が異なるか？

ビームサイズ、すなわち並列性により「最適な戦略の言語共通性」が異なることは、言語モデルと人間のインクリメンタルな言語処理が異なる可能性を間接的に示唆する。言語モデルは次トークン予測を最適化するが、仮にモデルサイズが大きいほど並列で多くの情報を持てるのであれば、十分大きな言語モデルは、言語によらず投機性の高い戦略を用いる可能性がある。逆に、ビームサイズが小さい方が人間の認知モデルとしては良いともされており [16]、人間は言語に応じて投機性の異なる戦略を用いると推測できる。実際に、既存研究では、言語によって人間の読み時間と言語モデルのサプライザルの近さが異なると指摘されており [22]、本研究はこの結果を「最適な戦略の違い」の観点から裏付けている。

6.3　次トークン予測と構文解析の乖離

次トークン予測を最適化する戦略は、構文解析精度を最適化する戦略とは必ずしも一致しない。例えば図 4下段では、構文解析を最適化する戦略は、英語・フランス語では投機性が高い傾向があるが、中国語・韓国語では逆である。加えて、図 4上段の correct はモデルが推論した構造ではなく正解構造を導出するアクションを与えて評価した結果であるが、驚くことに、どの言語でも多くの場合正解構造よりもモデルが推論した結果の方が次トークン予測性能が高い。ここから、「そもそもツリーバンクの統語構造は、次トークン予測という観点からは最適な構造ではないのではないか？」という仮説が立てられる。系列の reconstruction を目的関数として最適化した教師なし構文解析モデルの精度が低い [23] のも、このギャップが原因ではないかと推測できる。

7　結論と今後の展望

本研究では、系列・統語構造をインクリメンタルに処理する際の最適な戦略が言語により異なりうることを示した。本研究は句構造を対象としたが、自然言語には依存構造や意味構造などもあり、戦略の違いが「予測に用いる構造が言語により異なる」ことに起因する可能性もある。今後様々な構造に対する戦略を検討することで、言語の共通性・違いをさらに明らかにできると期待される。

謝辞

本研究は、JST 次世代研究者挑戦的研究プログラム JPMJSP2108 および JSPS 科研費 JP24KJ0666 の支援を受けたものです。

参考文献

[1] Matthew S. Dryer and Martin Haspelmath, editors. **WALS Online (v2020.3)**. Zenodo, 2013.

[2] George Kingsley Zipf. Human behavior and the principle of least effort: An introduction to human eoclogy. 1949.

[3] 田中久美子. 言語とフラクタル: 使用の集積の中にある偶然と必然. 東京大学出版会, 東京, 2021.

[4] Dmitriy Genzel and Eugene Charniak. Entropy rate constancy in text. In **Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02**, p. 199, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

[5] 石井太河, 宮尾祐介. 木形状分布の分析：自然言語の句構造とランダム木について. 言語処理学会第 30 回年次大会, 2023.

[6] Steven P Abney and Mark Johnson. Memory requirements and local ambiguities of parsing strategies. **J. Psycholinguist. Res.**, Vol. 20, No. 3, pp. 233–250, May 1991.

[7] Philip Resnik. Left-corner parsing and psychological plausibility. In **COLING 1992 Volume 1: The 14th International Conference on Computational Linguistics**, 1992.

[8] Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A Smith. Recurrent neural network grammars. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, **Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 199–209, San Diego, California, June 2016. Association for Computational Linguistics.

[9] John Hale, Chris Dyer, Adhiguna Kuncoro, and Jonathan Brennan. Finding syntax in human encephalography with beam search. In **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 2727–2736, 2018.

[10] Hiroshi Noji and Yohei Oseki. Effective batching for recurrent neural network grammars. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, **Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021**, pp. 4340–4352, Online, August 2021. Association for Computational Linguistics.

[11] Do Kook Choe and Eugene Charniak. Parsing as language modeling. In Jian Su, Kevin Duh, and Xavier Carreras, editors, **Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing**, pp. 2331–2336, Austin, Texas, November 2016. Association for Computational Linguistics.

[12] 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 Iryna Gurevych and Yusuke Miyao, editors, **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 1426–1436, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[13] Ethan Wilcox, Roger Levy, and Richard Futrell. Hierarchical representation in neural language models: Suppression and recovery of expectations. In **Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP**, pp. 181–190, Florence, Italy, August 2019. Association for Computational Linguistics.

[14] Hiroshi Noji and Yusuke Miyao. Left-corner transitions on dependency parsing. In **Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers**, pp. 2140–2150, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics.

[15] Damian Blasi, Ryan Cotterell, Lawrence Wolf-Sonkin, Sabine Stoll, Balthasar Bickel, and Marco Baroni. On the distribution of deep clausal embeddings: A large cross-linguistic study. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 3938–3943, Florence, Italy, July 2019. Association for Computational Linguistics.

[16] Ryo Yoshida, Hiroshi Noji, and Yohei Oseki. Modeling human sentence processing with left-corner recurrent neural network grammars. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-Tau Yih, editors, **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing**, pp. 2964–2973, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[17] Tatsuki Kuribayashi, Ryo Ueda, Ryo Yoshida, Yohei Oseki, Ted Briscoe, and Timothy Baldwin. Emergent word order universals from cognitively-motivated language models. In **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 14522–14543, Stroudsburg, PA, USA, 2024. Association for Computational Linguistics.

[18] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a Large Annotated Corpus of English: The Penn Treebank. **Computational Linguistics**, 1993.

[19] Martha Palmer, Fu-Dong Chiou, Nianwen Xue, and Tsan-Kuang Lee. Chinese treebank 5.1 ldc2005t01u01, 2005.

[20] Djamé Seddah, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho D Choi, Richárd Farkas, Jennifer Foster, Iakes Goenaga, Koldo Gojenola, Yoav Goldberg, Spence Green, Nizar Habash, Marco Kuhlmann, Wolfgang Maier, Joakim Nivre, A Przepiórkowski, Ryan Roth, Wolfgang Seeker, Yannick Versley, V Vincze, Marcin Woliński, Alina Wróblewska, and Eric Villemonte de la Clergerie. Overview of the SPMRL 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. 2013.

[21] Mitchell Stern, Daniel Fried, and Dan Klein. Effective inference for generative neural parsing. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, **Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing**, pp. 1695–1700, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[22] Tatsuki Kuribayashi, Yohei Oseki, Takumi Ito, Ryo Yoshida, Masayuki Asahara, and Kentaro Inui. Lower perplexity is not always human-like. In **Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)**, pp. 5203–5217, Stroudsburg, PA, USA, 2021. Association for Computational Linguistics.

[23] Jun Li, Yifan Cao, Jiong Cai, Yong Jiang, and Kewei Tu. An empirical comparison of unsupervised constituency parsing methods. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 3278–3283, Online, July 2020. Association for Computational Linguistics.

[24] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In Yoshua Bengio and Yann LeCun, editors, **3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings**, 2015.

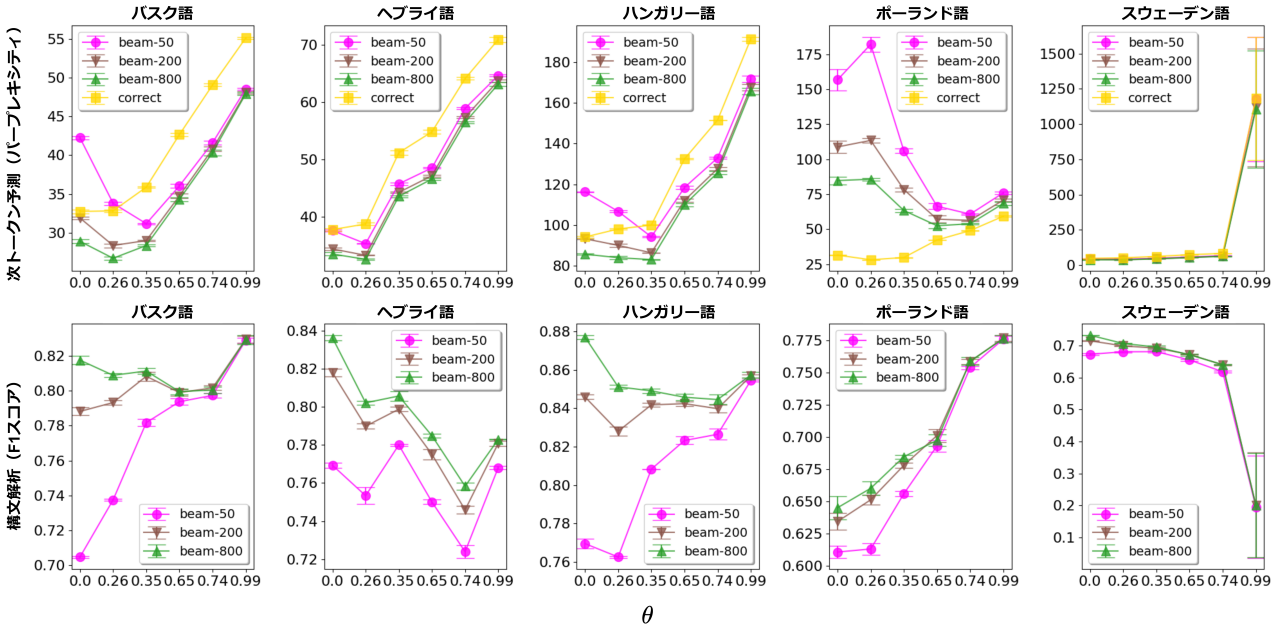

図 5　バスク語・ヘブライ語・ハンガリー語・ポーランド語・スウェーデン語の結果。エラーバーは標準誤差を表す。

A　データセット設定詳細

Subword 分割する際には、頻度 2 以上の単語数が 13-30K ほどあるデータセット（英語・中国語・フランス語・ドイツ語・韓国語・ハンガリー語）では、byte pair encoding（BPE）の語彙数を 5000 とし、頻度 2 以上単語数が 5-8K ほどのデータセットでは BPE の語彙数を 1500 とした。また、subword 分割には、sentencepiece を用いた。[7)]

B　モデル設定詳細

RNNG のハイパーパラメタとしては、composition モデルとして BiLSTM、隠れ状態の遷移には 2 層 LSTM、埋め込みベクトルは 256 次元、隠れ状態ベクトルは 256 次元、dropout は 0.3 とした。最適化にあたっては、Adam [24] を学習率 0.001 で用いた。学習は、100 エポックまたは 10000 ステップの大きい方に合わせて行う。バッチサイズに関しては、データ数が 10K を超えるデータセット（英語・中国語・フランス語・ドイツ語・韓国語）では 512 とし、データ数が 10K を超えないデータセットでは 128 とした。なお、実験では、異なる 3 つのランダムシードで学習を行いその平均を計算している。

C　その他の実験結果

図 5に、本文では省略した言語の結果を示す。ポーランド語とスウェーデン語に関しては、構文解析 F1 スコアがほかの言語よりも低くなっていることから学習がうまくいっていない可能性がある。

7）https://github.com/google/sentencepiece