Derivational Probing：言語モデルにおける統語構造構築の解明

染谷大河　吉田遼　谷中瞳　大関洋平
東京大学
{taiga98-0809, yoshiryo0617, hyanaka, oseki}@g.ecc.u-tokyo.ac.jp

概要

Transformer ベース言語モデルの内部表現には統語構造が表現されていることが示唆されているが，入力が各層を伝播する中で統語構造が構築される**過程**は依然として不明確である．本研究では，Transformer ベース言語モデルにおける，統語構造の構築過程を定量的に分析するための新しい手法として **Derivational Probing**（派生的プロービング）を提案する．これにより，文の局所的な構造と大域的な構造が，単語表現が各層を伝播する過程でどのように構築されるかを検証することが可能となる．BERT と GPT-2 を用いた実験では，BERT は局所的な構造を構築してから大域的な構造を構築するボトムアップな方法をとるのに対し，GPT-2 は局所的な構造と大域的な構造をよりパラレルに構築する傾向があることが示された．また，主述の一致タスクのケーススタディでは，BERT モデルが最終的に正しい統語構造を導出する場合でも，大域的な構造の構築が行われる層の位置が，タスクのパフォーマンスに影響を与えることが示され，大域的な構造を構築する最適な層の範囲が存在することを示唆した．

1　導入

近年，事前学習済み言語モデルは，自然言語処理の幅広いタスクにおいて目覚ましい成功を収めている．同時に，これらのモデルが具体的に何を学習し，言語知識をどのように表現しているかについて活発に分析が進められている [1, 2, 3]．

中でも，言語モデルの単語表現を直接分析して潜在的な統語構造を明らかにすることを目的として**構造プロービング (Structural Probing)** [4] が提案されている．[4] は，BERT [5] の単語埋め込み空間の幾何学的構造が依存構造木の距離をエンコードしていることを示し，モデルが全体として正しく統語構造を**表現**できていることを主張した．しかし，構造が構築される**過程**については必ずしも明らかになっていない．結果として得られる表現だけでなく，それらが層を通じてどのように構築されるかを理解することは，より包括的な理解のために不可欠であり，これらの表現が言語モデルによりどのように利用されているかのより深い理解にも繋がる可能性がある（cf. [1]）．

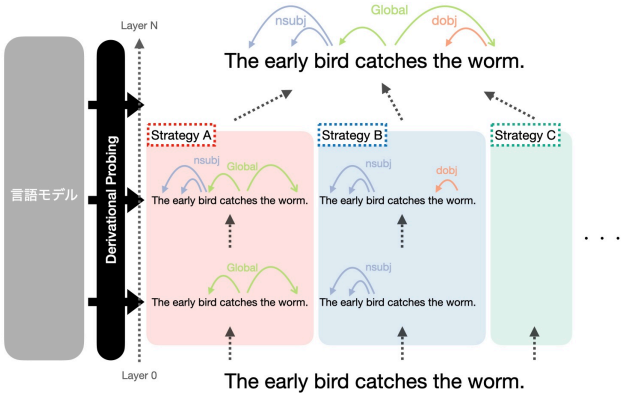

図 1　Derivational Probing は，入力が各層を伝播する中で言語モデルが統語構造を構築する過程を分析するための手法である．

一方，[6] は BERT の各層がどのようなタスクを解くのに寄与しているかを検証し，その道具として特定のタスクがどの層で解かれているかを特定する指標である **Expected Layer** を導入した．しかし，彼らの研究は言語モデルがどの層でどのタスクを解いている傾向があるかについての示唆を与えるものの，統語構造の構築の過程を直接分析するものではない．

そこで，本研究では **Derivational Probing**（派生的プロービング; 図 1）という新しいプロービング手法を提案する．この手法は，構造プロービング [4] と Expected Layer [6] を組み合わせた新しい手法である．この方法により，文の局所的な構造（主語名詞句，目的語名詞句，前置詞句など）と大域的な構造（主動詞とそれと直接の依存関係を持つ語同士の関係）の両方が，モデルの層を通じてどのような順序で構築されるかを検証することが可能となる．提

案手法を BERT [5] と GPT-2 [7] に適用した実験により，BERT は局所的な構造を構築してから大域的な構造を構築する**ボトムアップ**な方法をとるのに対し，GPT-2 は局所的な構造と大域的な構造をより**パラレル**に構築する傾向があることが示された．さらに，主述の一致タスクのケーススタディでは，モデルが最終的に正しい統語構造を導出できている場合でも，その構造構築の過程によりタスク性能が左右されることが示唆された．具体的には，BERT モデルでは，大域的な構造が構築される層が特定の層よりも低層もしくは高層になることでタスク性能が低下することが示され，大域的な構造を構築する最適な層の範囲が存在することが示唆された．

2 Derivational Probing

本研究では，事前学習済み言語モデルの統語構造の構築過程を分析するための新しい手法である **Derivational Probing** を提案する．この手法は，構造プロービング [4] と Expected Layer [6] という指標を組み合わせたものであり，言語モデル内で層ごとにどのように統語構造が構築しているのかを検証可能にする手法である．

図 2　大域的な構造（“Global”）と局所的な構造（“nsubj” と “dobj”）．

本研究で用いる構造プロービングの手法は，主に [4] のものに従っている．彼らの手法と本研究の手法の主な違いは，以下で定義される累積単語埋め込みを用いる点である．層 k について，文中の i 番目の単語の累積単語埋め込みを，層 0 から k までの埋め込みの重み付き和として定義する：

$$\mathbf{h}_i^k = \gamma \sum_{\ell=0}^{k} s^{(\ell)} \mathbf{h}_i^{(\ell)} \tag{1}$$

ここで，$s^{(\ell)} = \mathrm{softmax}(\mathbf{a})^{(\ell)}$ であり，γ と $\mathbf{a} \in \mathbb{R}^{k+1}$ は学習可能なパラメータである．この累積単語埋め込みは，現在の層だけでなく，それより低層の全ての層において獲得された情報を捉えるものである[1)]．次に，[4] の構造プロービングと同様に，これらの累積単語埋め込み間の距離が依存構造グラフ内での距離を表現できるかどうかを検証する．線形変換 $B \in \mathbb{R}^{d \times d}$ によってパラメータ化された二乗距離を以下のように定義する：

$$d_B(\mathbf{h}_i^k, \mathbf{h}_j^k)^2 = \left(B(\mathbf{h}_i^k - \mathbf{h}_j^k)\right)^\top \left(B(\mathbf{h}_i^k - \mathbf{h}_j^k)\right) \tag{2}$$

与えられたデータを用いて，線形変換 B を変換された空間内の距離が依存構造グラフ内のトークン間の距離 $d_B(\mathbf{h}_i^k, \mathbf{h}_j^k)^2$ に一致するように訓練する．依存構造グラフ上でのトークン間の距離は，構文木内の単語 i と j の間のエッジ数に対応する．また，層ごとの統語構造の構築過程を定量化するために，Expected Layer [6] を用いる．各層の各統語構造について，以下を計算する：

$$E[l] = \frac{\sum_{l=1}^{L} l \cdot (S(l) - S(l-1))}{\sum_{l=1}^{L} (S(l) - S(l-1))} \tag{3}$$

ここで，$S(l)$ は層 l までの埋め込みに対して訓練された線形変換が達成した性能であり，L はモデルの層の数である．この指標は，特定の統語構造が構築されたと期待される層を示すことが期待される．

ここで，文の大域的な構造と局所的な構造の概念を導入する．大域的な構造は，依存構造解析における主動詞とその直接の依存語（アークで直接接続された単語）からなる単語群を指す．局所的な構造は，各主動詞の直接の依存語とそれらの子孫からなる単語群を指し，構文木の葉ノードまでの全ての単語を含む．各局所的な構造は，その主要部が主動詞との間に持つ依存関係の名前によって識別される（図 2）．上記で定義された各大域的な構造と局所的な構造について，(1) 最小全域木アルゴリズムを用いて文の完全な構文木を構築し，(2) 対象の構造に関連するエッジを特定し（例：図 2 の “nsubj” の青いアーク），(3) これらのエッジに対する Unlabeled Undirected Attachment Score（UUAS）を計算する．UUAS は [4] でも使用されており，正しく予測されたエッジの数を，正解グラフの総エッジ数で割ることで計算される．局所的な構造と大域的な構造の Expected Layer を比較することで，異なる種類の構造がどの特定の層で構築されるかを特定することができる．

1)　このアプローチにより，層 $\ell+1$ の線形変換が達成するスコアが層 ℓ でのスコアよりも一般的に高くなるが，これが Expected Layer の計算に重要となる [6].

3 実験

3.1 データ

本研究では，Wikitext-103 [8] を使用する．各文は，spaCy [9] の依存構造解析器（EN_CORE_WEB_LG）を用いて依存構造解析された．本研究では，単文に対する言語モデルの統語構造構築能力に焦点を当てるため，関係節や節主語を含むデータを除外した．さらに，解釈を妨げる可能性のある “dep”（未分類の依存関係）や “punct”（句読点）などの依存関係を含む文も除外した（文末の句読点は例外）．

文構造ごとに統語構造が構築される順序を分析するために，主動詞から伸びるアークの持つ依存関係の種類によってデータをグループ化した．主要な文構造に焦点を当てるため，使用したデータの 10%以上を占めるグループのみを対象とした．この分類の結果，4 つの主要な文構造が得られた：(1) *Global*, *nsubj*, *dobj*; (2) *Global*, *nsubj*, *prep*; (3) *Global*, *nsubj*, *attr*; (4) *Global*, *nsubj*, *prep*, *dobj*.

結果として得られたデータセットから，ランダムに 50,000 文をサンプリングし，40,000 文を訓練用，5,000 文を検証用，5,000 文をテスト用に分割した．

3.2 実験設定

本研究では，提案手法を用いて BERT-base（cased）[5][2)]，BERT-large（cased） [5][3)]，GPT-2 small [7][4)]，GPT-2 medium [7][5)] の 4 つの事前学習済み言語モデルを分析する．5 つの異なるランダムシードで実験を行い，その平均と標準偏差を報告する．その他のハイパーパラメータは付録 A に記載している．

4 結果と考察

4.1 文全体に対する UUAS

まず，本研究の手法で計算された指標の妥当性の確認として，先行研究で報告されたものと同様の全体的な傾向を示すかどうかを確認する．テストデータを対象に，文全体の構造に対する UUAS を測定し層ごとにプロットした（図 3）．GPT-2 small と BERT-base は類似した傾向を示し，中間層付近でスコアが飽和している．GPT-2 medium と BERT-large を比べると，BERT-large の方が精度の上がり幅が緩やかだが，概ね中間層付近でスコアが飽和する同様の傾向を示している．これらの傾向は，BERT と GPT-2 モデルが中間層でピーク UUAS 性能を示す傾向があるという先行研究の知見 [4, 10] とほぼ一致している．相違点としては，本研究では層の後半での UUAS の減少がない点だが，これは層 0 から層 k までの重み付き平均である累積単語埋め込みを用いているためであり，妥当性を損ねるものではない．

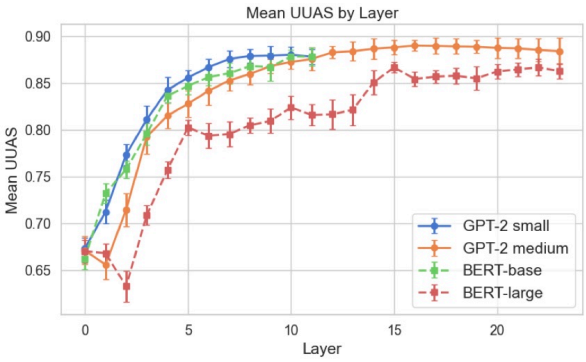

図 3 各モデルの文全体の構造に対する UUAS．エラーバーは 5 つのランダムシードにわたる標準偏差を表す．

4.2 各文構造に対する Expected Layer

図 4 に各文構造内の各大域的な構造と局所的な構造の Expected Layer を示す．BERT モデルでは，大域的な構造がすべてのセットで一貫して最も高い Expected Layer の値を示した．これは，BERT が**ボトムアップ**方式で統語構造を構築し，主語名詞句，目的語名詞句，前置詞句などの局所的な構造を最初に捉え，その後にグローバルな文構造を構築することを示唆している．

対照的に，GPT-2 モデルはすべての大域・局所的な構造について類似した Expected Layer の値を示しており，局所的な構造と大域的な構造をほぼ同時に構築している．これは，GPT-2 モデルが，パラレル方式で統語構造構築を行なっていることを示唆している．

4.3 ケーススタディ：主述の一致タスク

統語構造構築の過程がモデルの性能にどのように関連するかを調査するために，主語と動詞の間に主語ではない他の名詞を含む主述の一致タスクを用いてケーススタディを実施した．本研究では，[11] で用いられたデータセットの一部として提案されている，前置詞句によって修飾されている主

2) https://huggingface.co/google-bert/bert-base-cased
3) https://huggingface.co/google-bert/bert-large-cased
4) https://huggingface.co/openai-community/gpt2
5) https://huggingface.co/openai-community/gpt2-medium

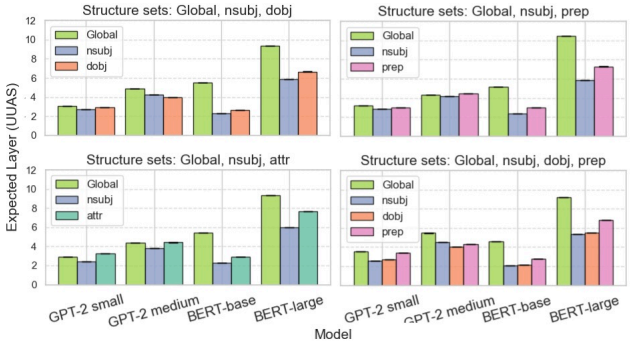

図 4　各モデルの異なる文構造に対する Expected Layer．エラーバーは 5 つのランダムシードにわたる標準偏差を表す．

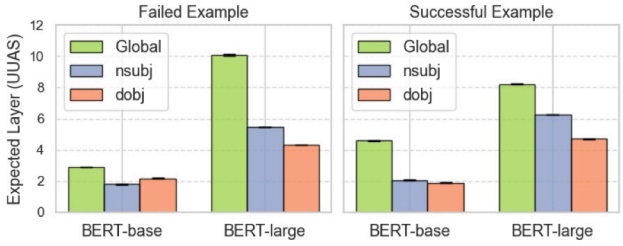

図 5　主述の一致タスクに成功した例と誤っていた例における統語構造の Expected Layer．エラーバーは 5 つのランダムシードにわたる標準偏差を示す．

語を含むデータを 1000 件サンプリングして用いた（例：*The senators behind the architect* ***avoid/*avoids*** *spicy dishes.*）．本ケーススタディでは，特に主語と目的語の両方の句が複数形の単語からなる場合に焦点を当てた．

本ケーススタディでは，4.2 で顕著な傾向を示した BERT-base と BERT-large を対象にする．各トークンを一つずつマスクしつつ各文の疑似確率を計算し，正しい主述の一致が行われている文に主述の一致が正しくない文より高い確率を付与できるかを検証した．BERT-base は 1,000 件中正解した例が 984 件，誤っていた例が 16 件，BERT-large は 1,000 件中正解した例が 983 件，誤っていた例が 17 件であった．図 5 は，成功した例と誤っていた例それぞれにおける大域的な構造と局所的な構造の Expected Layer を示している．

失敗例では，BERT-large における大域的な構造の Expected Layer が成功例より高く，BERT-base ではより低くなっている．これは，大域的な構造の構築が行われる層の違いが主述の一致タスクの性能に影響する可能性を示唆する．

また，より詳しい分析を行うために，線形変換によって予測された各トークン間の距離を元に，多次元尺度法（MDS）を用いて各層での構造構築過程を成功例と失敗例に分けて可視化した．図 6 は，BERT-large における可視化の結果である．成功例では，5 層目までに正しい主語（赤でハイライト）が正しい主語ではない名詞よりも動詞に近くなっている（図下）．失敗例では，最終的には正しい構造を予測できているが，この関係が 12 層目まで確立されなかった（図上）．これにより，モデルが最終的には正しい統語構造を導出する場合でも，大域的な構造の構築が行われる層の位置によって，主述の一致タスクの性能が変わり得ることが示唆された．[6)]

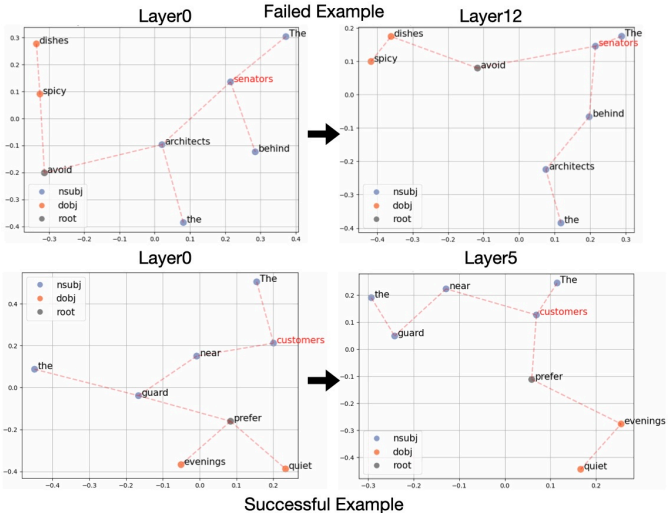

図 6　主述の一致タスクの成功例と失敗例における，各層におけるトークン間の予測距離を MDS で可視化したもの．赤字は文の主語を示す．可視化対象は BERT-large.

5　結論

本研究では，Transformer ベース言語モデルにおける統語構造の構築過程を分析するための手法として Derivational Probing を提案した．実験により，BERT はボトムアップ的に，GPT-2 は並行的に構造を構築することが明らかになった．また，モデルが最終的には正しい統語構造を導出する場合でも，文の大域的な構造の構築をどの層で行なったかによって主述の一致タスクの性能に差が出る可能性が示唆された．分析をより大規模な言語モデルや他の言語に拡張することは今後の課題としたい．

謝辞

本研究は，JST さきがけ JPMJPR21C2/JPMJPR21C8 および JSPS 科研費 24H00087 の支援を受けたものです．

6)　BERT-base の可視化は付録 B にある．

参考文献

[1] Tal Linzen and Marco Baroni. Syntactic structure from deep learning. **Annual Review of Linguistics**, Vol. 7, No. Volume 7, 2021, pp. 195–212, 2021.

[2] Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. Explainability for large language models: A survey. **ACM Trans. Intell. Syst. Technol.**, Vol. 15, No. 2, February 2024.

[3] Tyler A. Chang and Benjamin K. Bergen. Language Model Behavior: A Comprehensive Survey. **Computational Linguistics**, Vol. 50, No. 1, pp. 293–350, 03 2024.

[4] John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, **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. 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[5] 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.

[6] Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics.

[7] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[8] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

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

[10] Tiwalayo Eisape, Vineet Gangireddy, Roger Levy, and Yoon Kim. Probing for incremental parse states in autoregressive language models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, **Findings of the Association for Computational Linguistics: EMNLP 2022**, pp. 2801–2813, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[11] Rebecca Marvin and Tal Linzen. Targeted syntactic evaluation of language models. In **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing**, pp. 1192–1202, Brussels, Belgium, 2018. Association for Computational Linguistics.

A　ハイパーパラメータ

実験に使用したハイパーパラメータを表 1 に示す．すべてのモデルは 4 台の NVIDIA RTX A5000 (24GB) で訓練および評価を行った．本論文のすべての実験に要した計算コストは約 120 GPU 時間である．

最適化アルゴリズム	Adam
学習率	1e-3
エポック数	40
学習率スケジューラ	ReduceLROnPlateau
バッチサイズ	32

表 1　実験に使用したハイパーパラメータ

B　BERT-base の統語構築過程

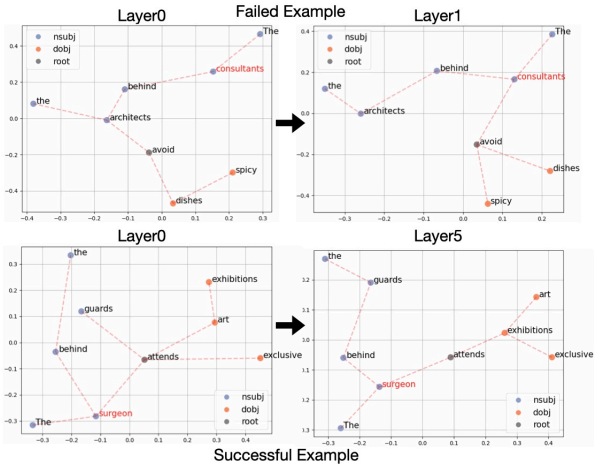

図 7　主述の一致タスクの成功例と失敗例における，各層におけるトークン間の予測距離を MDS で可視化したもの．赤字は文の主語を示す．可視化対象は BERT-large.