言語モデルにおける統語的長距離依存関係の内部操作

木村一馬 [1,3]　大関洋平 [2,3]　菅原朔 [3,2]
[1] バルセロナ自治大学　[2] 東京大学　[3] 国立情報学研究所
kazumakimura17@nii.ac.jp　oseki@g.ecc.u-tokyo.ac.jp
saku@nii.ac.jp

概要

本研究は, 統語的長距離依存関係の構築における言語モデルの内部操作を, Causal Intervention 手法の一種である Activation Patching を用いて分析することを目的とする. 4 つの異なる依存関係を対象とした実験の結果, 小さい言語モデルでは全ての依存関係に共通の操作を用いるが, 言語モデルのサイズが大きくなるにつれて, 操作の分化が進むことが分かった. この結果は, 言語モデルがサイズによって, 異なる操作を内在化しており, 依存関係タイプを内部操作によって区別していることを示唆する.

1　はじめに

1.1　背景

近年, 言語モデルの内部表現に関する言語学的研究が盛んであり, 特に Structural probing 手法による一連の研究は, Transformer モデル [1] の内部で統語情報が, トークン埋め込み近傍で線形的に近似する形で表示・保存されていることを明らかにしてきた [2, 3, 4]. 一方, モデル内部で, 統語情報がどのようなプロセスを経て構築されるのかについては議論が続いており, Machanistic interpretability [5, 6] の分野で, 統語・文法情報に特化したコンポーネントの存在がいくつか報告されている [7, 8, 9, 10]. しかし, それらが (a) どのような操作であるか (操作的性質), (b) 操作として一般化されたものなのか, あるいは構文ごとに別個のヒューリスティクスを用いているのか (操作的一般化) に関しては, 明らかになっていない.

1.2　目的

そこで本研究では, Causal Intervention 手法の 1 つである Activation Patching [11, 12, 13] を用いて, モデルが長距離依存関係に対して, (a) どのような操作を内在化しているのか, また, (b) それらが一般化された操作群として抽出できるかを検証する. ここでは [14] にならいモデル内部における統語情報に関する表示と操作を, 図 1 のように, それぞれ認知科学における計算理論レベルとアルゴリズムレベルに相当するものとして位置付ける [15]. モデル内部における統語表示はいわゆる文法関係や階層構造関係を表示するレベルであり, 統語操作は文法関係構築のプロセスに相当すると考える.[1)]

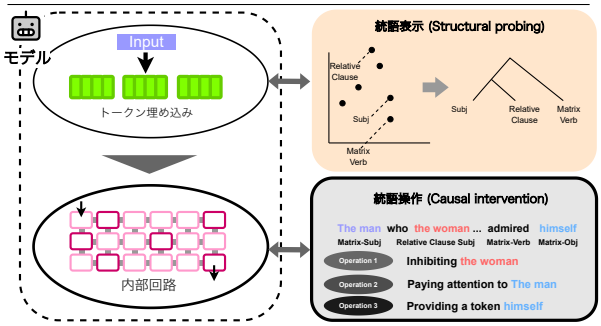


図 1: 本研究の概要: 統語移動がない依存関係 (NPI, 再帰代名詞) と統語移動がある依存関係 (wh-疑問文, 分裂文) に着目し, 当該依存関係に関する内部操作を特定することを目的とする.

2　実験設定

2.1　データセット・モデル

本実験では, それぞれ長距離依存関係 (NPI, 再帰代名詞束縛, wh-疑問文, 分裂文) に対し 300 件のデータセット用いる. 依存関係を結ぶ語の位置は固定し, 主節の主語 (分裂文では that 節内部の主語) が関係代名詞節を含む形となっている. 関係代名詞節内の要素は, 依存関係を構築することが不可能な位置とされる [16, 17]. 各依存関係の構文的特性上, トークン数

1) ここでの統語操作は, 「統語情報に基づいたモデル内部の操作」を指しており, 伝統的な理論言語学で提案されている移動・一致といった統語構造構築に関する理論的操作とは異なる意味合いで用いている.

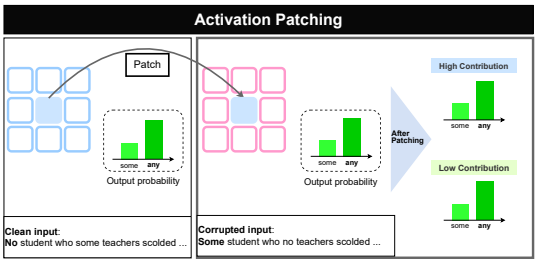


図 2: Activation Patching の概略図

Clean	**No** student who some teachers scolded [any]
Corrupted	**Some** student who no teachers scolded [any]

表 1: clean–corrupted 入力のペアの例 (NPI): 全ての依存関係のペア例については, 付録 A を参照.

を均一にすることが難しいため, 統語的構成素に基づいたラベル (**役割トークン**) を付与し, 役割トークンに対して分析を行う (付録 B を参照).

実験は, GPT-2 モデル (small, medium, large) を対象に行った (報告は small と large のみ).

2.2 Activation Patching

本実験では, Activation Patching を用いる (図 2)。この手法では, 適切な入力 (clean) と一部だけ壊した入力のペア (corrupted) (表 1) に対して, 片方の入力に, もう片方を実行した際の構成要素を差し込み（パッチ）, モデルの最終出力が回復する（i.e., clean 入力の出力に近づく）かどうかを評価する. 出力が回復すれば, パッチした構成要素は元の入力側で当該タスクの出力に関与していたと解釈する. なお, 回復率は以下のスコアを元に評価する.

$$R_\ell = \frac{M^{(\ell)}_{\text{patched}} - M_{\text{corr}}}{M_{\text{clean}} - M_{\text{corr}} + \varepsilon}. \tag{1}$$

パッチング単位として, (a) ヘッドの値のみ (b) ヘッドの値 + 注意パターン (式 2) を採用する.

$$\text{pattern}_{\ell,h}(i,j) = \text{softmax}\left(\frac{Q_{\ell,h}(i) \cdot K_{\ell,h}(j)}{\sqrt{d}}\right) \tag{2}$$

パッチング分析には, `transformer_lens` パッケージ [11] を用い, `role_target` を出力する際の活性パターンを clean 入力から corrupted 入力への差し込む (denoising patching) [12].

2.3 注意機構分析

さらに, 回復率が高かったヘッド (上位 10) に対して, それらの注意パターンを特定する. 具体的には, (a) パッチ前とパッチ後の主要トークンへの注意量 (attention mass) (b) 層ごとの各トークンへの注意量に着目する. 分析に用いる注意量は, 各トークンに向けられた注意重み (attention weight) の集計値とする. 回復率が高いヘッドがどこに注意を向けているかを特定することで, 当該ヘッドの「操作内容」を解釈することが可能となる.

Model	NPI	Reflexive	Wh	Cleft
GPT-2 Small	0.00891	-1.06665	2.79009	3.04385
GPT-2 Medium	0.21738	0.68945	2.56422	2.69535
GPT-2 Large	-0.69095	0.64406	2.43949	2.86943

表 2: モデルサイズごとの Logit 差 (len = 0).

2.4 依存関係距離

また, モデル内部の操作が線形情報と構造情報のどちらに依存するのかを探索するため, 依存関係を結ぶトークン間 (`role_key`, `role_target`) に, 付加要素 (副詞および前置詞句) を介在させたデータセットを各依存関係に対して, 介在する要素の数に応じて 300 例作成した (len =2, 4) (付録 C を参照). 上と同様の分析を行い, モデルの注意パターンに変化が見られるか検証する.

3 実験結果

3.1 依存関係ごとの性能

表 2 に各モデルサイズ・依存関係における Logit 差 (clean - corrupted) の値を示す. ほとんどの依存関係において, モデルは clean 入力に対して高い出力確率を与えていることが分かる. NPI において logit 差が低いのは先行研究の結果と合致する [18].

3.2 パッチ前後の注意量

図 3 に示すように, GPT-2 small において, 注意量はどの依存関係タイプにおいても `role_key` 強く向くことが分かった (平均注意量: NPI, 0.057; reflexive, 0.067; Wh, 0.106, cleft, 0.090).

GPT-2 large では, NPI と再帰代名詞に関して, `role_rc_subject` に注意が強く向いたのに対して, wh および分裂文では, どちらのトークンにも顕著な注意は見られなかった (図 4).

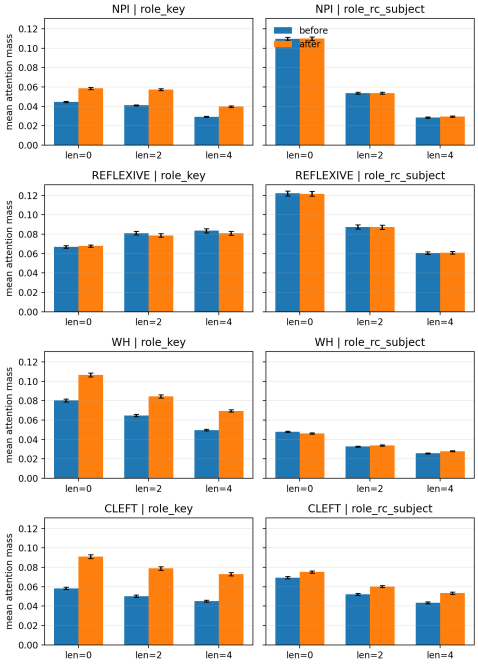


図 3: パッチ前後の役割トークン (role_key, role_rc_subject) への注意量 (Top-10, GPT2-small).

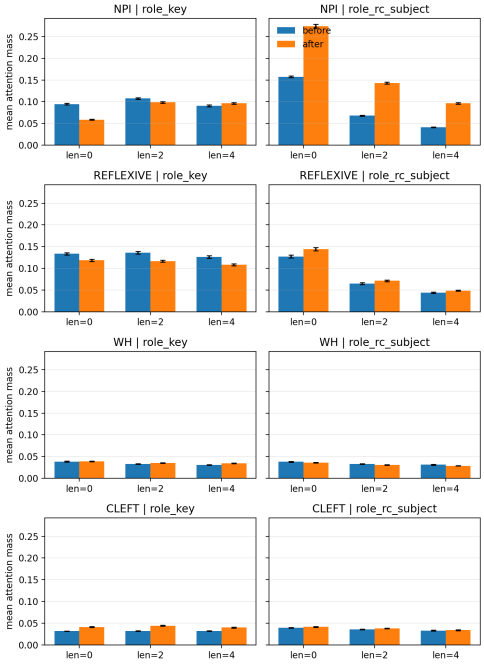


図 4: パッチ前後の役割トークン (role_key, role_rc_subject) への注意量 (Top-10, GPT2-large).

3.3 層ごとの注意量

各モデルサイズおよび依存関係の層ごとの注意量の分布を図 5 に示す．GPT-2 small では，全ての依存関係において注意量は低層あら中間層 (3-7 層) でピークを示す．どの依存関係においても，role_key と role_rc_subject の両方がほぼ同じ層範囲でピークを示している．

GPT-2 large では，注意量の分布は特定の層に集中したピークを示すが，そのピーク位置は依存関係タイプおよび役割トークンによって異なる．NPI および再帰代名詞では，role_rc_subject に与えられた注意量が中間層で顕著なピークを示した (平均ピーク値: 約 0.5 (NPI) および 約 0.6 (再帰代名詞)).

3.4 依存関係距離

3.2 節, 3.3 節の結果から分かるように，依存関係の距離が長くなった場合でも，役割トークンへの注意パターンに大きな変化は見られないことが分かる．

4 考察

本研究の結果は，言語モデルにおける構造的依存関係構築が，モデルサイズによって操作的に分化する可能性を示唆する．

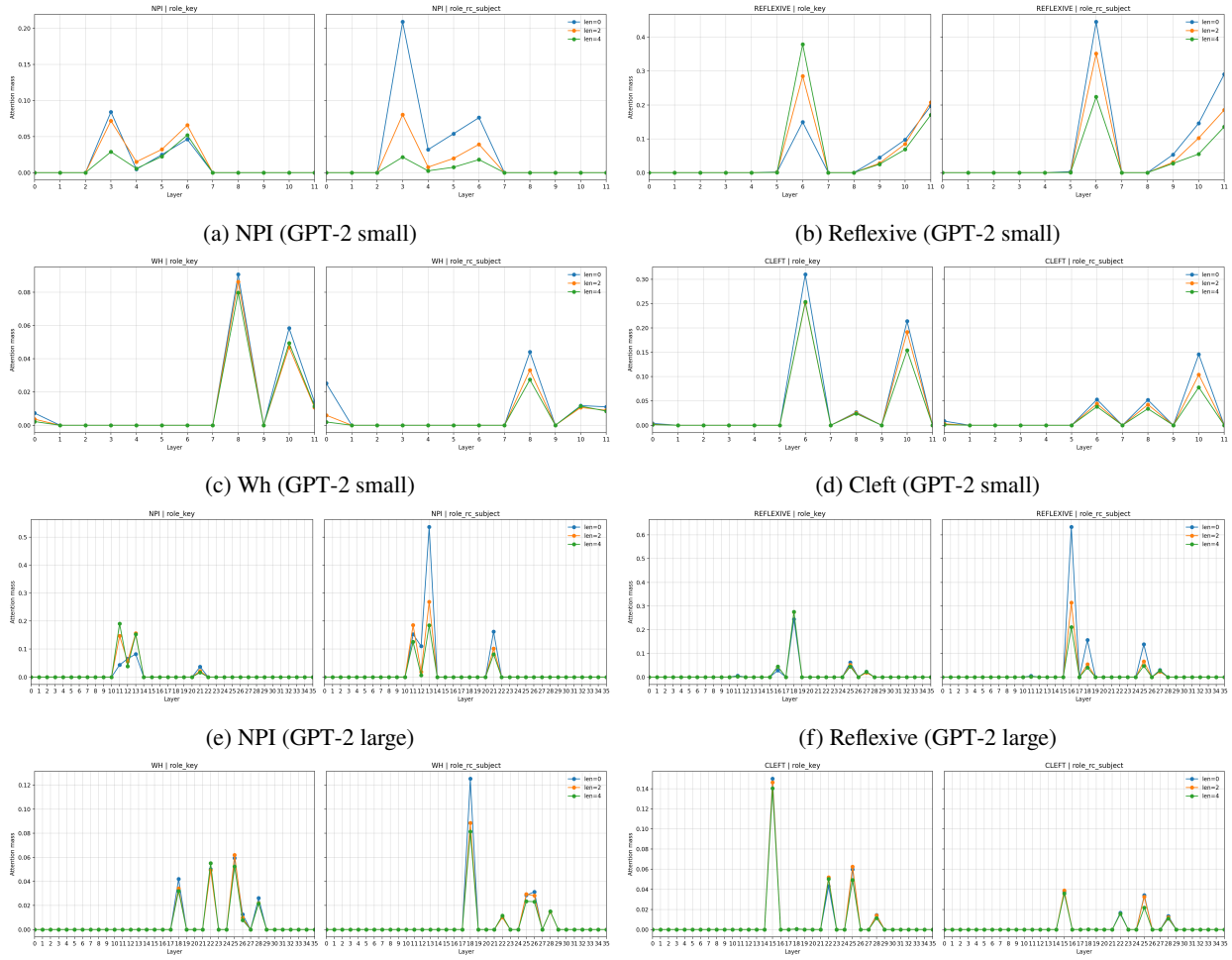
4.1 GPT-2 small

小規模なモデルでは，依存関係タイプを問わず，注意パターンが類似しており，依存関係は何らかの形で一般化された単一の操作によって処理されている可能性がある．

しかし，この注意パターンがどのような情報に基づいているかは，2 通りの解釈が存在する．1 つは，構造的な長距離依存関係においても，indirect object identification (IOI) タスクのような非構造依存タスクと同様，線形ヒューリスティクスを用いている可能性である [19, 20]．この場合，主節主語に強く注意が向くという実験結果は，単に「文頭に出現する名詞句へ注意を向ける」といった粗い操作によって依存関係を解いていると解釈することになる．

2 つ目は，先行研究が主張するような人間の filler-gap 依存関係構築のメカニズムと類似した操作を内在化している可能性である [10, 21]．この場合，標的トークンに遭遇した段階で，依存関係を認可する主節主語に注意を向けるという，cue-based retrieval モデルに近いストラテジーを用いて依存関係を構築していることになる [22].

しかし，本実験の結果からは，GPT-2 small における注意操作の性質については踏み込むことができないため，2 つの可能性を示唆するにとどめる．



(a) NPI (GPT-2 small)

(b) Reflexive (GPT-2 small)

(c) Wh (GPT-2 small)

(d) Cleft (GPT-2 small)

(e) NPI (GPT-2 large)

(f) Reflexive (GPT-2 large)

(g) Wh (GPT-2 large)

(h) Cleft (GPT-2 large)

図 5: 層ごとの役割トークン (role_key, role_rc_subject) への注意量 (GPT-2 small / large).

4.2 GPT-2 large

一方, 大きいモデルでは, 依存関係によって操作の分化が起こっていると考えられ, NPI および再帰代名詞では構造的に不適格な位置を注意機構によってマークする操作が観察された. これに対し, wh-疑問文や分裂文では, 注意機構による構造的マーキングは見られず, 主にそれ以外の構成要素 (e.g., MLP, 残差結合) によって処理されていると考えられる.

4.3 内部操作の分化

1.2 節で提起した疑問に答える形で考察をまとめると, 言語モデルが内在化する操作はモデルサイズによって異なり, 一般化の程度および基準も異なっている. 小さいモデルでは, 粗い指標による注意機構に基づいた操作で実験で取り扱った全ての依存関係を統一的に解くのに対し, 比較的大きいモデルでは, 依存関係ごとに異なる操作 (注意機構ベース vs. MLP or 残差結合) を用いることが示唆される. しかし本実験では, 注意機構以外の構成要素を分析の対象に含めていないため, これらの依存関係が他の構成要素で類似した振る舞いを示すかは不明である.

5 おわりに

本研究では, 言語モデルにおける統語的長距離依存関係の内部操作が, Causal Intervention 手法で特定できることを示した. 具体的には, 言語モデルのサイズによって, 操作の分化が起こることが示唆され, 大きいモデルでは, 統語移動のある依存関係と統語移動のない依存関係を操作的に区別していることが分かった. 今後, 分析対象に注意機構以外の構成要素含め, 本研究で用いなかった依存関係 (e.g., 数量詞-代名詞束縛) に対し大規模な実験を行うことでより正確な操作を抽出できると考えられる.

謝辞

本研究は JST 創発的研究支援事業 JPMJFR232R, JST BOOST JPMJBY24D9 の助成を受けたものです.

参考文献

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[2] Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. What do you learn from context? probing for sentence structure in contextualized word representations, 2019.

[3] 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)**, pages 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[4] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of BERT's attention. In Tal Linzen, Grzegorz Chrupała, Yonatan Belinkov, and Dieuwke Hupkes, editors, **Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP**, pages 276–286, Florence, Italy, August 2019. Association for Computational Linguistics.

[5] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits, 2020.

[6] Hosein Mohebbi, Jaap Jumelet, Michael Hanna, Afra Alishahi, and Willem Zuidema. Transformer-specific interpretability. In Mohsen Mesgar and Sharid Loáiciga, editors, **Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts**, pages 21–26, St. Julian's, Malta, March 2024. Association for Computational Linguistics.

[7] Kabir Ahuja, Vidhisha Balachandran, Madhur Panwar, Tianxing He, Noah A. Smith, Navin Goyal, and Yulia Tsvetkov. Learning syntax without planting trees: Understanding hierarchical generalization in transformers, 2025.

[8] Michael Hanna, Yonatan Belinkov, and Sandro Pezzelle. Are formal and functional linguistic mechanisms dissociated in language models?, 2025.

[9] Ryoma Kumon and Hitomi Yanaka. Analyzing the inner workings of transformers in compositional generalization, 2025.

[10] Sasha Boguraev, Christopher Potts, and Kyle Mahowald. Causal interventions reveal shared structure across English filler–gap constructions. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, **Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing**, pages 25032–25053, Suzhou, China, November 2025. Association for Computational Linguistics.

[11] Neel Nanda. Attribution patching: Activation patching at industrial scale, 2023.

[12] Stefan Heimersheim and Neel Nanda. How to use and interpret activation patching, 2024.

[13] Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods, 2024.

[14] Adam Davies and Ashkan Khakzar. The cognitive revolution in interpretability: From explaining behavior to interpreting representations and algorithms, 2024.

[15] David Marr. **Vision**. W. H. Freeman, 1982.

[16] Tanya Miriam Reinhart. **The syntactic domain of anaphora**. PhD thesis, Massachusetts Institute of Technology, 1976.

[17] Luigi Lizzi. On the complemetarity of generatieve grammarand large language models. **Italian Journal of Linguistics**, 37(1):145–152, 2025.

[18] Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. Blimp: The benchmark of linguistic minimal pairs for english, 2023.

[19] Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small, 2022.

[20] Yaniv Nikankin, Anja Reusch, Aaron Mueller, and Yonatan Belinkov. Arithmetic without algorithms: Language models solve math with a bag of heuristics, 2025.

[21] Michael Hanna and Aaron Mueller. Incremental sentence processing mechanisms in autoregressive transformer language models. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, **Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)**, pages 3181–3203, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

[22] Ryo Yoshida, Shinnosuke Isono, Kohei Kajikawa, Taiga Someya, Yushi Sugimoto, and Yohei Oseki. If attention serves as a cognitive model of human memory retrieval, what is the plausible memory representation? In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, **Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pages 9795–9812, Vienna, Austria, July 2025. Association for Computational Linguistics.

A Clean–corrupted ペア

本実験で取り扱う依存関係の Clean–corrupted ペアは, 以下のようになっている. NPI では, clean 入力では否定要素の ‘‘no’’ が文頭に, corrupted 入力では否定要素ではない ‘‘some’’ が文頭に配置されている. ターゲット位置には 否定要素によって認可される ‘‘any’’ を配置する. 再帰代名詞では, clean 入力では, 主節主語位置の名詞句と主節目的語位置にある再帰代名詞の性が一致しているが, corrupted 入力では一致していない. Wh-疑問文では, clean 入力では, 文頭の wh-句が有生名詞句で, corrupted 入力では, 非生物名詞句になっている. どちらもターゲット位置は, 有生名詞句のみと意味的に整合する動詞を用いている. 分裂文も wh-疑問文と同様の設定となっている. なお, 全てのペアにおいて, 主節主語の名詞句と関係節内名詞句は clean 入力と corrupted 入力で入れ替わる形になっている.

実験で使用した全てのデータは, 人手で作成した構造テンプレートと語彙リストを元に自動生成した.

	NPI
Clean	**No** student who some teachers scolded [any] notebook.
Corrupted	**Some** student who some teachers scolded [any] notebook.
	Reflexive
Clean	**The man** who the woman predominantly dislikes admired [himself] yesterday.
Corrupted	**The woman** who the woman predominantly dislikes admired [himself] yesterday.
	Wh
Clean	**Which man** did the woman who wrote the paper [visited] yesterday ?
Corrupted	**Which paper** did the woman who praised the man [visited] yesterday ?
	Cleft
Clean	It is **the man** that the woman who wrote the paper [visited] yesterday.
Corrupted	It is **the paper** that the woman who praised the man [visited] yesterday.

表 3: 依存関係ごとの clean–corrupted 入力のペア例: [] で囲まれているトークンがターゲット位置.

B 役割トークン

	$role_{key}$	$role_{rel}$	$role_{rel-subj}$	$role_{rel-verb}$	$role_{adj}$	$role_{main-verb}$	$role_{target}$
NPI	[No NP]	who	[some NP]	[verb]	[adverb]	[verb]	[any]
Reflexive	[The NP_m]	who	[the NP_f]	[verb]	[adverb]	[verb]	[himself]

表 4: NPI・再帰代名詞における役割トークンの付与例

	$role_{key}$	$role_{pro}$	$role_{subj}$	$role_{rel}$	$role_{rel-obj}$	$role_{rel-verb}$	$role_{target}$
Wh-depedendency	[Wh NP_a]	did	[the NP_i]	which	[verb]	[NP_i]	[verb]
Cleft	[It was the NP_a]	who	[the NP_i]	[which]	[verb]	[NP_i]	[verb]

表 5: Wh-疑問文・分裂文における役割トークンの付与例

C 依存関係距離

表 3 のデータに対して, 付加要素を主節目的語と関係節の右端の間に配置する. len=2 の条件では, 付加要素が 2 つ, len=4 では付加要素が 4 つ配置される設計になっている (表 3 のデータを len=0 とする).

なお, 付加要素には文の意味内容を損なわない副詞および前置詞句を人手で選択した.

Dependency	Example (linear conditions)
NPI	**No** student who some teachers scolded {*carefully* / *carefully, in the lab*} [any] notebook.
Reflexive	**The man** who the woman predominantly dislikes {*in the lab* / *in the lab, two days ago ...*} admired [himself] yesterday.
Wh	**Which man** did the woman who wrote the paper {*at the conference* / *at the conference, last year*} [visited] yesterday?
Cleft	It is **the man** that the woman who wrote the paper {*two days ago* / *two days ago, in the building*} [visited] yesterday.

表 6: 依存関係距離分析に用いるデータの例