Tree-Structured Language Processing in Large Pretrained Transformers

Ananth Agarwal
Stanford University
ananthag@stanford.edu

Abstract

Humans are adept at generalizing small, familiar language units to comprehend unfamiliar language. Recent work found evidence of hierarchical tree-structured computation developing in transformers over training that improves their generalization. In this work, I investigate tree structure in popular open source pretrained transformers with the motivation of improving understanding of transformer internal mechanics. By studying word span invariance to surrounding context, I find evidence of tree structure in these models. For the encoder-only models, these internal trees align well with gold constituency parse trees.

1 Introduction

Research suggests that humans process written language in a hierarchical tree-structured computation where smaller constituents are recursively grouped into larger ones up to the full expression (Pallier et al., 2011). Hierarchical processing assists with *compositional generalization*, which is our ability to understand novel text by piecing together the meanings of familiar words and phrases. Transformer models have computational capacity for arbitrarily complex language processing on input text. However, recent work found intrinsic tree-structured compositionality in transformer encoders (Murty et al., 2022). Murty et al. show that encoders of various sizes (2 - 6 layers) within encoder-decoder transformer architectures demonstrate increasing intrinsic tree-structured computation when trained on three seq2seq compositional generalization datasets. These internal tree structures increasingly align with gold human-annotated syntax constituency parse trees over training. They also find that models with higher internal tree structure perform better on generalization tasks.

Open source pretrained transformers have soared in popularity thanks to easy access and exceptional performance on a variety of tasks. This project examines a couple of these models from a mechanistic interpretability angle by applying the same approach from the Murty et al. paper to determine if they exhibit tree-structured computation on text. The motivation is to improve understanding of the internal mechanisms of seemingly unstructured, opaque transformers with O(millions) parameters. Alignment of this tree structure, if any, with gold parse trees would suggest human-like constituency processing. Moreover, since tree structure is linked with better out-of-domain generalization, analysis of tree structure metrics in these open source models could suggest future work to improve their generalization by fine-tuning with a "tree-structure regularizer".

2 Methodology

2.1 Context-free Attention Mask

I follow the same unsupervised methodology as Murty et al. Let $\mathcal{D}$ be a dataset of text sentences $\{S_1, \ldots, S_K\}$. For each $S = (w_1, \ldots, w_N)$, the goal is to find a *bracketing* of words w into a binary tree such that this predicted tree best approximates how a strictly tree-structured compositional transformer would process S. Conceptually, comprehending a large sequence of text through its smaller constituents suggests that the meanings of these small constituents are generally invariant to the larger context they are in. In other words, the vector distance for some distance function d between vector representations of these spans of words in different contexts would be small. For example, consider the sentences S_1 = *Chocolate cake is yummy* and S_2 = *Chocolate cake is unhealthy*. In a model with tree-structured computation, the hidden representation of the span of words *Chocolate cake* in both sentences would be similar despite the contextual difference in expressed sentiment about the cake. My goal is to determine if the model processes S_1 with an intrinsic context-invariant bracketing such

as ((*Chocolate*, *cake*), (*is*, *yummy*)).

Let p be a span of words $p = (w_i, \ldots, w_j)$ within sentence S. For now, suppose we are looking at the output hidden state representations h^m from a single transformer layer with index m. The **contextual representation** h_p^m is the hidden state representation of p with standard attention masking (i.e., causal masking in decoders). The **context-free representation** $\tilde{h_p^m}$ is the hidden state representation of p where the attention mask is modified to only permit attention within the span. Nonzero attention weights are only permitted between word indices $[i, j]$; sentence context is masked out. For a function T that produces binary trees on input S, Murty et al. define the *tree projection* of the transformer on S, $T_{proj}(S)$, as the tree that minimizes *span contextual invariance* (SCI):

$$\text{SCI}(S, T) \triangleq \sum_{p \in T(S)} d(h_p^m, \tilde{h_p^m}) \quad (1)$$

$$T_{proj}(S) = \arg\min_{T(S)} \text{SCI}(S, T(S)) \quad (2)$$

The tree projection of the transformer on S is a bracketing of its word spans into a binary tree that approximates how a tree-structured transformer would process S. $T_{proj}(S)$ is an *induced* tree; although the transformer architecture is not modified in any way, its internal representation of S fits the binary tree bracketing of words in $T_{proj}(S)$. There are $O(N^2)$ contiguous spans in each sentence of size 1 to N. Once all $O(N^2)$ SCI values are calculated, $T_{proj}(S)$ is created by greedily choosing spans top-down to minimize total SCI, starting from finding the optimal bifurcation of span (0, $N-1$). Fig. 1 in Appendix A illustrates the algorithm.

Murty et al. found that tuning the layer at which the context-free attention mask starts to be applied can improve quality of induced parses. For a transformer with L layers where $0 \leq l < m \leq L-1$, let m be the *end threshold* index whose hidden states are used to calculate SCI as in Eq. (1) and let l be the *start threshold* index at which the forward pass to compute $\tilde{h_p^m}$ uses the modified mask. Layers $[0, l)$ therefore use standard attention masking.

2.2 Decoder-only: Subnetwork Span Distance

In autoregressive left-to-right language models, the hidden representation for word i output by layer l, h_i^l, encodes context from words $\{0, \ldots, i-1\}$. Define **residual vectors** as the set of hidden vectors $\{h_0^l, \ldots, h_{i-1}^l\}$ for the span of words $r = (w_0, \ldots, w_{i-1})$ preceding the contextual vectors $\{h_i^l, \ldots, h_j^l\}$ for words in p. For each p in S, I compute the distance between representations of r and p across subnetworks with start and end layers $[l, m]$. If it decreases, this subnetwork did context-sensitive processing such that context from words in r became increasingly encoded in representations of p. If it does not decrease, it is evidence of tree-structured computation since the representation of p is invariant to context over layers $[l, m]$. Eq. (3) is a SCI minimization objective with this distance formulation.

$$\text{SCI}(S, T) \triangleq \sum_{r,p \in T(S)} d(h_r^l, h_p^l) - d(h_r^m, h_p^m) \quad (3)$$

This does not require mutating the attention mask as the $\tilde{h_p^m}$ calculation does.

3 Experiments

3.1 Pretrained "Off-the-Shelf" Transformers

Open AI GPT-2 (Radford et al., 2019) is an autoregressive language model and one of the most downloaded models on Hugging Face. I evaluate all available sizes: S (124M parameters), M (355M), L (774M), and XL (1.5B).

EleutherAI Pythia (Gao et al., 2023) is a class of autoregressive language models developed to facilitate interpretability research. I evaluate several sizes: 70M, 140M, 410M, 1B, and 2.8B.

Microsoft DeBERTaV3 (He et al., 2021) is an encoder-only transformer that succeeds BERT and RoBERTa. I evaluate all available sizes: XS (70M), S (142M), M (184M), and L (435M).

3.2 Datasets

COGS (Kim and Linzen, 2020) is a compositional generalization semantic parsing dataset of sentences generated with a rules-based approach from a context-free grammar. It consists of in-domain train, val, and test splits and an out-of-distribution "gen" split that test novel combinations of primitive phrases and grammatical roles. Since this project does not require training, I use a sample of 5,000 sentences and gold constituency parse trees from the train dataset.

Penn Treebank (Marcus et al., 1993) is an annotated corpus of Wall Street Journal sentences. Unlike COGS, these sentences contain punctuation and other syntactical variety. I use a set of 2,416 sentences and gold parse trees.

3.3 Metrics

I employ the same metrics as Murty et al. for a consistent experiment setup.

t_{score} is the average normalized SCI for a transformer on dataset $\mathcal{D}$. While SCI quantifies compositionality using hidden state vector distances, for an induced tree $T_{proj}(S)$ we also want to know if this measured compositionality is higher than that of an arbitrarily random tree. For instance, if the model's produced hidden representations are anisotropic (concentrated in a small cone in the embedding space (Gao et al., 2019)) and distance function d is cosine distance, all SCI values will be small, thereby muddying the search for distinctive internal tree structure since random bracketings of words could achieve low SCI. Therefore, t_{score} is defined as the dataset average SCI of $T_{proj}(S)$ normalized with the SCI of a random tree from a uniform distribution of trees:

$$t_{score} \triangleq \frac{1}{K} \sum_{S \in \mathcal{D}} (\mathbb{E}_T \text{SCI}(S, T) - \text{SCI}(S, T_{proj}(S)) \tag{4}$$

t_{score} near 0 implies low internal tree structure, i.e., context-free representations are far from their contextual counterparts. The magnitude of t_{score} depends on distance function and sentence length since SCI is summed over word spans.

t_{parseval} is the bracketing F1 score (Black et al., 1991) of induced trees scored against gold constituency parse trees. For example, the induced tree ((*Chocolate*, *cake*), (*is*, *yummy*)) has predicted bracket spans: (0, 1), (2, 3), (0, 3). These are evaluated with the gold bracket spans.

3.4 Transformer Layer Sweep

Within large transformer models, the degree of compositional processing may vary across subnetworks. For a model with L layers, I sweep over all combinations of start threshold (l) and end threshold (m) where the start is in range $[0, L/4]$ and end is in range $[(-L/4)-1, -1]$. I report in the results the t_{score} and t_{parseval} for the (l, m) pairing with the best t_{parseval} for each model. While I experimented with both Euclidean and cosine distance functions, I report results with Euclidean distance since anisotropy in hidden representations could obscure results with cosine distance. In Eq. (1), I take the mean of the Euclidean distances between each corresponding contextual and context-free vector in the span of words p:

$$d(h_p, \tilde{h_p}) = \frac{1}{|p|} \sum_{w_i \in p} \left\| h_{w_i} - \tilde{h}_{w_i} \right\|_2$$

In Eq. (3), I take the Euclidean distance between the mean of the residual and contextual vectors:

$$d(h_r, h_p) = \left\| \frac{1}{|r|} \sum_{w_i \in r} h_i - \frac{1}{|p|} \sum_{w_j \in p} h_j \right\|_2$$

3.5 Baselines

English is a right-branching language, and right-branching parses are competitive baselines for English (Murty et al., 2022). For each dataset sentence, I evaluate t_{parseval} of completely right-branching, completely left-branching, and random binary trees as baselines.

4 Results

Results with the attention mask approach for all models are in Table 1, and results for the subnetwork span distance approach are presented for Pythia in Table 2.

5 Analysis & Discussion

Table 1 shows strong performance across the board on COGS, which the right-branching baseline indicates contains highly right-branching sentences. All models average over half of all predicted tree brackets correct (t_{parseval}). See Appendix B for comparison of example induced and gold trees. The relatively high t_{parseval} values convey that the novel combinations of familiar constituents tested in COGS have hidden representations in each model that generally align with human-expected constituencies.

Unlike COGS, Penn Treebank reflects the syntactical variety in human writing; sentences are longer and have more complex structure. Inclusion of punctuation including the trailing period causes the right-branching baseline to have low t_{parseval} , but generally sentences are right-branching. While the decoder models have high t_{score} , implying the spans in induced trees have high contextual invariance, these trees mostly do not align with gold constituents (low t_{parseval}). For GPT-2 in particular, its

	COGS		Penn Treebank	
	t_{parseval}	t_{score}	t_{parseval}	t_{score}
Left-branching	0.31	–	0.08	–
Right-branching	**0.76**	–	0.15	–
Random Tree	0.46	–	0.18	–
GPT-2 S	0.59	80.5	**0.25**	451.6
GPT-2 M	0.56	109.4	0.21	995.7
GPT-2 L	**0.66**	75	0.22	149.5
GPT-2 XL	0.6	17.5	0.18	2,271.3
DeBERTa XS	0.6	9.5	**0.38**	27.9
DeBERTa S	0.57	5.4	0.36	42.7
DeBERTa M	0.59	14.4	0.35	32.7
DeBERTa L	**0.62**	18.6	0.33	78.5
Pythia 70M	0.58	6.53	0.24	132.2
Pythia 160M	**0.67**	10.67	0.27	203.3
Pythia 410M	**0.67**	15.53	**0.31**	964.1
Pythia 1B	0.56	35.79	0.2	270.8
Pythia 2.8B	0.56	34.88	0.25	653.5

Table 1: Metrics averaged across each dataset where SCI is the distance between contextual and context-free representations derived using attention masks that mask out context (Section 2.1).

	COGS		Penn Treebank	
	t_{parseval}	t_{score}	t_{parseval}	t_{score}
Left-branching	0.31	–	0.08	–
Right-branching	**0.76**	–	0.15	–
Random Tree	0.46	–	0.18	–
Pythia 70M	0.73	89.52	0.2	101.38
Pythia 160M	0.75	173.07	**0.24**	49.03
Pythia 410M	**0.76**	688.65	0.19	746.92
Pythia 1B	**0.76**	1,481.68	0.21	2,549.44
Pythia 2.8B	0.74	571.15	0.17	302.15

Table 2: Metrics averaged across each dataset where SCI is the difference in distance between residual and contextual vectors across a subnetwork (Section 2.2).

trees are only slightly more human-consistent than a random tree. DeBERTa v3 has comparatively lower t_{score} but higher t_{parseval} values that indicate that while the induced trees are not as contextually invariant as their decoder model counterparts (less "rigidly defined" trees), the hidden representations better reflect gold constituent structure. This concurs with Murty et al.'s finding of intrinsic tree structure in encoders even though the DeBERTa models studied here are deeper. While there is no obvious trend in t_{score} across model sizes within each model type, the decline in t_{parseval} for GPT-2 and DeBERTa as size increases suggests that while these big models still have contextually invariant spans, it is possible that the gold constituents' representations are more diffused across the larger parameter space and therefore not as identifiable with the single start and end layer thresholds (Section 3.4) reported here.

Induced trees using subnetwork span distance are much more right-branching; in fact for Pythia 410M and 1B t_{parseval} on COGS exactly matches the strict right-branching baseline. The large t_{score} values for these right-branching trees suggest that SCI is minimized by including the most possible context in residual span r. Between the two approaches, the Penn Treebank results indicate attention masking is better at creating context-free representations. Eq. (3) is likely an oversimplification of how preceding context is embedded in succeeding words' representations.

The most prominent difference between decoders and encoders that could explain the observed t_{parseval} difference is causal masking in decoders. One possible explanation is *attention sinks* have been observed in autoregressive models like Pythia where the initial tokens consistently receive high attention score despite lack of semantic significance (Xiao et al., 2023). Attention sinking is at odds with tree-structured computation. Follow-up research to my work could apply BERTology techniques (Rogers et al., 2020), modified as necessary to fit autoregressive models, to study differences in the linguistic knowledge different pretrained model families have. This could suggest why it is easier to recover accurate trees from BERT-like models.

6 Conclusion

The pretrained transformers studied in this paper demonstrate tree-structured computation because large positive t_{score} indicates the models internally process highly contextually invariant spans. Alignment with gold parse trees is more readily apparent in DeBERTa than GPT-2 and Pythia on structurally complex sentences. Fig. 5 in Appendix B shows an example DeBERTa predicted parse that is often intuitively reasonable even where it does not concur with the gold parse. Further work into analyzing inner representations like attention and hidden states and how they correlate with syntax could help inform alternative methods to measure context-free representations for pretrained autoregressive models. A good starting point would be to evaluate previous BERTology approaches on newer and larger models.

Known Project Limitations

At the start of the project I hoped to investigate more pretrained transformers like ELECTRA (Clark et al., 2020), EleutherAI GPT-J (Wang and Komatsuzaki, 2021), and LLama 2 (Touvron et al., 2023) to be able to detect more patterns in their results and draw stronger conclusions about internal tree structure in these high-performing models more generally. Given time constraints and time spent iterating on the exact methodology (e.g., distance function, sweep parameters), for this quarter I was limited to the three models mentioned in the paper. Thus one limitation of this project is that it is less comprehensive across model families than I originally hoped. An additional limitation is that this report does not build on the claim by Murty et al. that more tree-structured models generalize out-of-domain better. While the pretrained models here are already well-benchmarked (e.g., on Hugging Face's Open LLM Leaderboard), taking the time to extract or compute for myself performance results on compositional generalization datasets would allow me to draw correlations between the observed tree structure and such generalization results.

Authorship Statement

This paper is my own work. I am grateful to Shikhar Murty, a CS PhD student in the Stanford NLP Group advised by Professor Christopher Manning, for mentoring me throughout this project. Shikhar guided me from start to finish. This project builds on his work presented at ICLR 2023 that originally studied tree structure in transformers.

References

E. Black, S. Abney, D. Flickenger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. 1991. A procedure for quantitatively comparing the syntactic coverage of English grammars. In *Speech and Natural Language: Proceedings of a Workshop Held at Pacific Grove, California, February 19-22, 1991*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. *CoRR*, abs/2003.10555.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tieyan Liu. 2019. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing.

Najoung Kim and Tal Linzen. 2020. COGS: A compositional generalization challenge based on semantic interpretation. *CoRR*, abs/2010.05465.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Shikhar Murty, Pratyusha Sharma, Jacob Andreas, and Christopher D. Manning. 2022. Characterizing intrinsic compositionality in transformers with tree projections.

Christophe Pallier, Anne-Dominique Devauchelle, and Stanislas Dehaene. 2011. Cortical representation of the constituent structure of sentences. *Proceedings of the National Academy of Sciences*, 108(6):2522–2527.

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

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how BERT works. *CoRR*, abs/2002.12327.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks.

A Tree Projection Algorithm

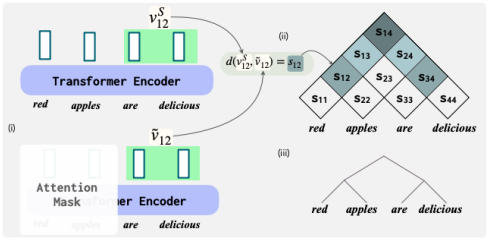

Figure 1: Diagram from (Murty et al., 2022) illustrating the algorithm to induce a tree on sentence S using attention masking. (i): Run inference to get contextual and context-free hidden states for word span (1, 2) (inclusive) and calculate SCI (Eq. (1)). (ii): Repeat for each span in the sentence. (iii): Greedily pick spans that minimize SCI (Eq. (2)) to induce a tree top-down.

B Induced vs. Golden Tree Comparison

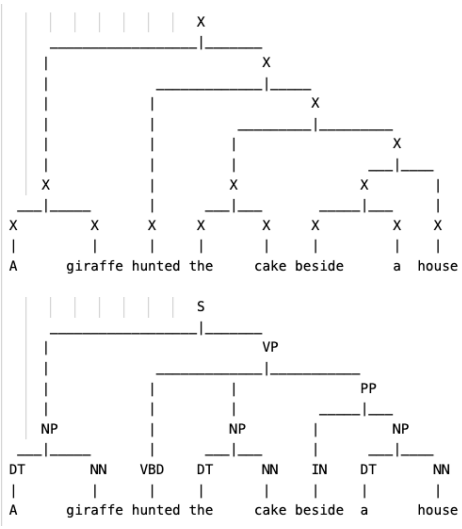

Figure 2: Example of a COGS sentence induced tree (top) produced by Pythia 160M using attention masked context-free representations compared to the gold constituency parse tree (bottom). The start threshold is 2 and the end threshold is -3 (configuration reported in Table 1). The model's computation pattern closely resembles that of the gold tree: the noun phrases *A giraffe* and *the cake* exactly match and the top level verb phrase *hunted the cake beside a house* is also correct. While the model does not bracket the noun phrase *a house*, it does bracket the prepositional phrase *beside a house* in accordance with the gold tree.

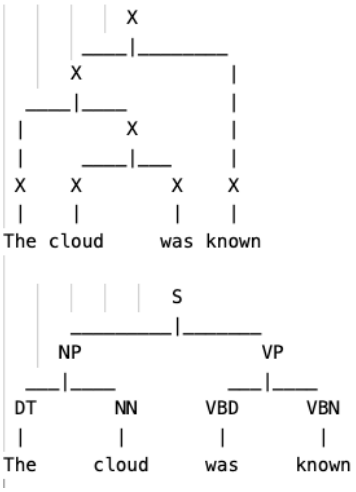

Figure 3: Example of a COGS sentence induced tree (top) produced by GPT-2 M using attention masked context-free representations compared to the gold constituency parse tree (bottom). The start threshold is 2 and the end threshold is -1 (configuration reported in Table 1). Unlike Pythia 160M in Fig. 2, here the model does not bracket word spans in accordance with the gold constituents of the sentence at all.

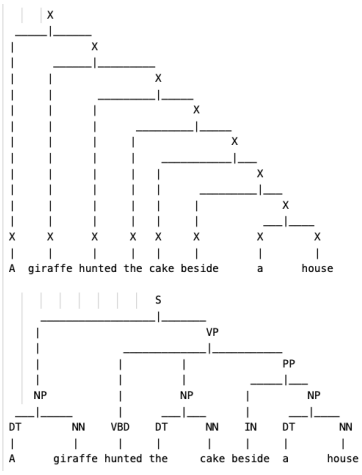

Figure 4: Example of a COGS sentence induced tree (top) produced by Pythia 70M using subnetwork span distance representations compared to the gold constituency parse tree (bottom). The start threshold is 0 and the end threshold is -3 (configuration reported in Table 2). The prediction is entirely right-branching; such predictions show why model performance almost matches the exclusively right-branching baseline.

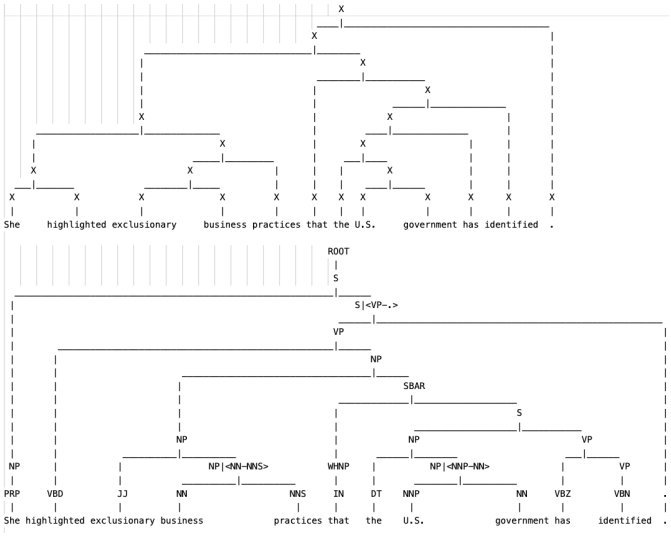

Figure 5: Example of a Penn Treebank sentence induced tree (top) produced by DeBERTa v3 XS compared to the gold constituency parse tree (bottom). The start threshold is 0 and the end threshold is -2 (configuration reported in Table 1). The model's computation pattern on small constituents like the noun phrases *exclusionary business practices* and *the U.S. government* resemble the golden tree. Impressively, the larger subordinate clause *that the U.S. government has identified* also matches. While the predicted bracket span *She highlighted exclusionary business practices* is not a span in the gold tree, it is a reasonable prediction that intuitively is context invariant.

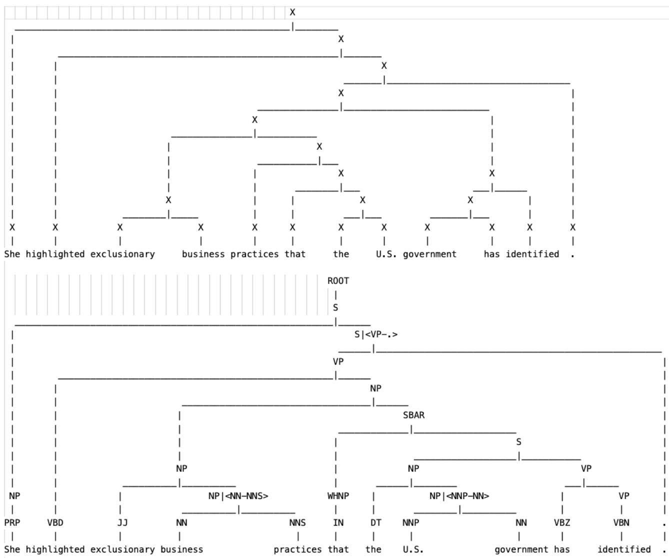

Figure 6: Example of a Penn Treebank sentence induced tree (top) produced by Pythia 1B using attention masked context-free representations compared to the gold constituency parse tree (bottom). The start threshold is 1 and the end threshold is -5 (reported in Table 1). Except for the topmost bracketing, there is no overlap. If this large model truly does process spans in a way that resembles the human-annotated parses, it is not discernable from looking at the output of a single end layer index that generated the prediction shown here.