ABSTRACTING INFLUENCE PATHS FOR EXPLAINING (CONTEXTUALIZATION OF) BERT MODELS

Anonymous authors
Paper under double-blind review

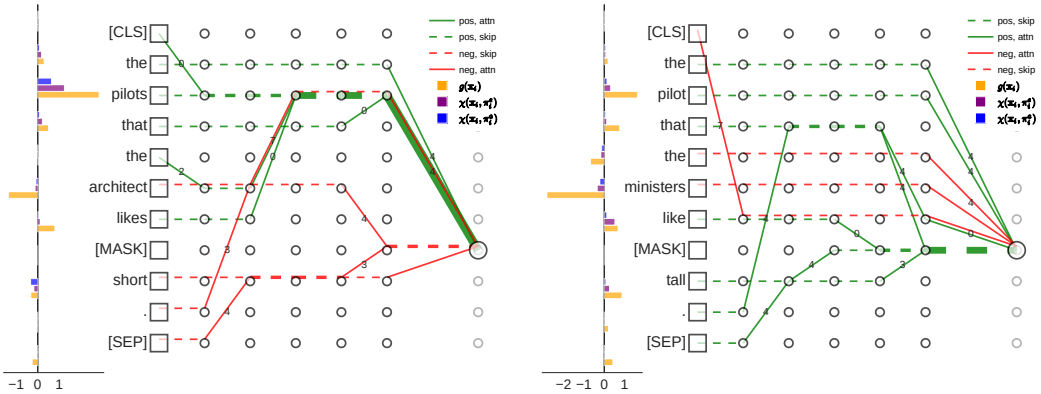
ABSTRACT

While *"attention is all you need"* may be proving true, we do not yet know *why*: attention-based models such as BERT are superior but how they contextualize information even for simple grammatical rules such as subject-verb number agreement (SVA) is uncertain. We introduce *multi-partite patterns*, abstractions of sets of paths through a neural network model. Patterns quantify and localize the effect of an input concept (e.g., a subject's number) on an output concept (e.g. corresponding verb's number) to paths passing through a sequence of model components, thus surfacing how BERT contextualizes information. We describe guided pattern refinement, an efficient search procedure for finding patterns representative of concept-critical paths. We discover that patterns generate succinct and meaningful explanations for BERT, highlighted by "copy" and "transfer" operations implemented by skip connections and attention heads, respectively. We also show how pattern visualizations help us understand how BERT contextualizes SVA across clauses, and why it makes errors in some cases while succeeding in others.

1 INTRODUCTION

The adage *"attention is all you need"* is proving true: attention-based transformer models such as BERT (Devlin et al., 2018) are becoming the state-of-the-art building block for NLP tasks. Beyond computational benefits in parallelization, BERT is responsible for solutions with superior accuracy on most NLP tasks. Like deep models in general, however, BERT is largely inscrutable: we lack understanding of how BERT operates even with respect to simple grammatical rules such as subject-verb number agreement (SVA). On the highest level, BERT has been explained in terms of *contextualization*. Individual word-level information at the input layer is integrated into higher-order structures or concepts in the deeper layers (see the general architecture in Figure 2). While this process has been found to resemble grammatical sentence structures (Hewitt & Manning, 2019; Coenen et al., 2019) in some cases, the principle approaches for discovering contextualization, those based on representations and attention analysis have limitations.

Representation analyses (Lin et al., 2019; Tenney et al., 2019a) demonstrate that relevant linguistic concepts are *associated* with the activations of BERT components (i.e. subject's number associated with the activations of a certain head at a certain layer) but do not tell us how representations come about nor do they localize concepts which we do not apriori relate to the rule (i.e. other representable concepts involved in SVA). Meanwhile, inspection of attention weights as indicators of the flow of information between BERT layers (Clark et al., 2019), requires subjective inference of relevant function (i.e. inference that a certain head may be involved in SVA because high attention weights between cells at the subject and cells at the verb). Using attention for explanation has been found to be problematic in other contexts (Brunner et al., 2020; Jain & Wallace, 2019). Analysis of attention further disregards the role of skip connections that do not involve attention at all. Neither approach allows us to track a concept as a causal chain from input to output or to distinguish helpful from hindering representations or flows (hindering information such as contextualization of confounding inputs like unrelated nouns in a sentence lead to errors on SVA).



(a) plural subject + singular intervening noun(PS)

(b) singular subject + plural intervening noun(SP)

Figure 1: Significant patterns π_i^a extracted by GRP from the attention-level graph for task *SVA across Object Relative Clauses* in two attractor cases. Left: bar plots of the distributional influence (yellow), $\chi(\mathbf{x}, \pi_i^e)$ (purple) and $\chi(\mathbf{x}, \pi_i^a)$ (blue) for each word at position i. Right: significant patterns π_i^a from each input word at position i to quantity of interest (verb number correctness). The square nodes denote the input embeddings and circles denote internal embeddings. Broken lines correspond to skip connection in the attention block while solid lines correspond to connection through (any) attention heads. Attention connections with high traffic are marked with the corresponding attention head. Line colors represent the sign of influence (red as negative and green as positive)

To address these limitations we introduce *multi-partite*[1] *patterns*, abstractions of sets of paths through a neural model (a graph). Patterns quantify and localize the *effect* of an input concept (e.g. a subject's number) on an output concept (e.g. corresponding verb's number) to *a collection of* paths passing through a sequence of model nodes and/or edges. We describe *guided pattern refinement*, a search procedure for finding patterns representative of concept-critical paths that lets us selectively explore the importance of chosen aspects of a model (e.g. in BERT, we can refine patterns showing criticality of certain heads to pathways also showing whether this is due to skip connections or due to attention). To demonstrate the contextualization process, we further extend the experimental framework to integrate impacts of multiple words towards a given concept (as opposed to impact of a single word, e.g. subject on SVA). Example visualization of patterns and their influence for localizing SVA in two sentences are shown in Fig. 1.

Our contributions are: 1) We describe *multi-partite patterns* for explaining the model-wide contextualization in neural models and *guided pattern refinement* (GPR) to discover influential patterns focusing of model elements of interest. 2) We visualize BERT's contextualization of a variety of grammatical concepts such as subject-verb agreements(SVA) and reflexive anaphora(RA), and show how BERT encode these concepts using correct or incorrect cues. 3) We validate the derived patterns significance towards several concepts by way of concentration and model compression experiments.

We begin with a summary of requisite techniques in Sec. 2. We describe the core elements of our methodology in Sec. 3 and exemplify them for understanding BERT in Sec. 4. We elaborate on related works in Sec. 5 and conclude in Sec. 6.

2 Background

BERT. BERT (Devlin et al., 2018) is a pretrained Transformer (Vaswani et al., 2017) model that has spurred the recent success of NLP. A pretrained BERT model is trained using a Masked Language Modeling (MLM) paradigm where the model learns to guess a masked word in a sentence. The masked word is denoted with the special token `[MASK]`. The input embeddings are passed into a stack of Transformer encoder layers, an example of which is illustrated in Fig. 2(left). We focus

[1] *Multi-partite* because patterns abstract sets of paths in neural models viewed as multi-partite graphs.

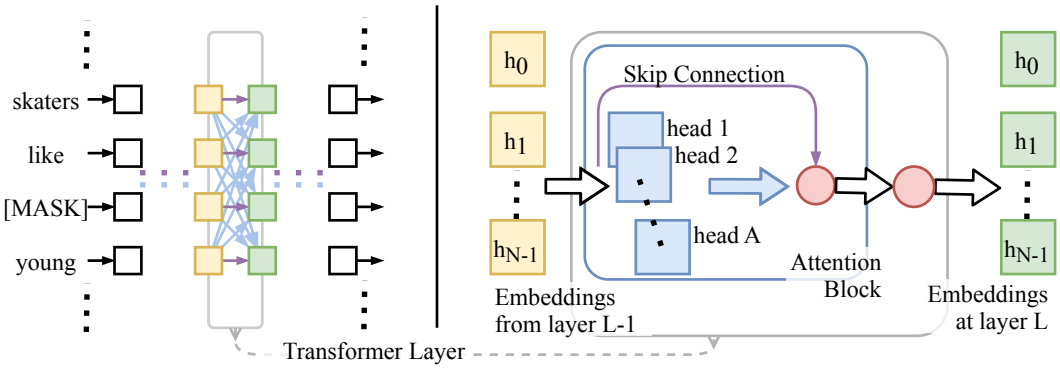


Figure 2: BERT Transformer architecture (left) and elements of a transformer layer (right).

on the attention block in Fig. 2(right) and denote other operations with red nodes. BERT learns relations between embeddings from previous layer (or from the input) with multi-head attentions, which has been proved to correlate with certain linguistic relations (Clark et al., 2019). The original embeddings are passed by a skip connection and combined with attention outputs to produce new embeddings of the input token as the output of a Transformer layer.

Notation. We use $\mathbf{x}$ to denote a vector and x_i as the i-th element in $\mathbf{x}$. For a BERT model, we use L as the number of Transformer encoder layers, H as the hidden dimension of embeddings at each layer, and A as the number of attention heads. Consider a list of input word embeddings $\mathbf{x}_{0:N-1} = [\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_{N-1}], \mathbf{x}_i \in \mathbb{R}^d$ as the input for BERT, we denote the output of the l-th layer as $\mathbf{h}^l_{0:N-1}$. First layer inputs are $\mathbf{h}^0_{0:N-1} \stackrel{\text{def}}{=} \mathbf{x}_{0:N-1}$. Scores are $\mathbf{y}_i \stackrel{\text{def}}{=} \texttt{softmax}(W\mathbf{h}^L_i), W \in \mathbb{R}^{C\times H}$, where C is the vocabulary size, as the final classification layer and outputs are the predicted word $\hat{y}_i \stackrel{\text{def}}{=} \arg\max_c \mathbf{y}_{i,c}$ for the i-th token. We denote the index of `[MASK]` as m.

Distributional Influence attributes to each DNN input a measure of impact on model output. Saliency (Baehrens et al., 2010) is a simple example that defines influence as the gradient of output with respect to input. Influence can quantify a concept (e.g. SVA) by instrumenting a model's inputs with a *distribution of interest* (DoI) and the output with a *quantity of interest* (QoI) (Leino et al., 2018).

Definition 1 (Distributional Influence) *Given a model $f : \mathbb{R}^d \to \mathbb{R}^n$, an input $\mathbf{x}$, a DoI $\mathcal{D}(\mathbf{x})$ parameterized by an input, and a QoI $q : \mathbb{R}^n \to \mathbb{R}$,* Distributional Influence *$g_q(\mathbf{x})$ quantifies the impact of an input concept defined by the DoI on the output concept defined by the QoI: $g_q(\mathbf{x}) \stackrel{\text{def}}{=} \mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})} \frac{\partial q(f(\mathbf{z}))}{\partial \mathbf{z}}$.*

Instantiations defining SVA and RA concepts in BERT models are found in Sec. 4. We use expectation notation for convenience of generality though in our use cases DoI are distributed along a one dimensional path $c : [0, 1] \to \mathbb{R}^d$. Influence is then the line integral $\int_0^1 \frac{\partial q(f(\mathbf{x}))}{\partial \mathbf{x}} (c(\alpha)) \cdot p(\alpha)\, d\alpha$ where p is the p.d.f. of the DoI (Sundararajan et al., 2017).

Influence Paths localize an influence measurement to paths in a neural model, treated as a graph (Lu et al., 2020). A computation graph $\mathcal{G} \stackrel{\text{def}}{=} (\mathcal{V}, \mathcal{F}, \mathcal{E})$ is a set of nodes, activation functions, and edges, respectively. We assume the graph is directed, acyclic, and does not contain more than one edge per adjacent pair of nodes[2]. A path p in G is a sequence of edges $[n_{i_0} \to n_{i_1}, n_{i_1} \to n_{i_2}, \cdots, n_{i_{-2}} \to n_{i_{-1}}]$ such that all $n_i \to n_j \in p$ are edges in $\mathcal{E}$. We denote the Jacobian of the output of node n_i w.r.t the output of connected (not necessarily directly) predecessor node n_j evaluated at $\mathbf{x}$ as $\partial n_j(\mathbf{x})/\partial n_i(\mathbf{x})$ (see Appendix A for the definition w.r.t activations). We write $\nabla_{\mathbf{x}} p$ as the Jacobian of the last node in p w.r.t the first node evaluated at input $\mathbf{x}$ equivalently as per chain rule: $\nabla_{\mathbf{x}} p \stackrel{\text{def}}{=} \partial p_{-1}(\mathbf{x})/\partial p_0(\mathbf{x}) = \prod_{i=1}^{|p|-1} \partial p_i(\mathbf{x})/\partial p_{i-1}(\mathbf{x})$.

[2]This is only a restriction for notational conveniences to follow; if a given neural model does have more than one edge between adjacent nodes, we can replace duplicate edges with 2-length paths through dummy identity nodes to satisfy this requirement without affecting its semantics.

Definition 2 (Individual Path Influence) *Given a path p of a computation graph $\mathcal{G}$ the* individual path influence *for an input* $\mathbf{x}$*, or $\chi(\mathbf{x}, p)$ is:* $\chi(\mathbf{x}, p) \stackrel{\text{def}}{=} \mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}\left[\nabla_{\mathbf{z}} p\right]$.

Lu et al. (2020) use individual path influence to decompose distributional influence to paths and explain internal LSTM behaviour under SVA via the most influential path $\arg\max_{p\in\mathcal{P}} \chi(\mathbf{x}, p)$ where $\mathcal{P}$ are all paths from input to a particular output (normally a QoI).

3 Path Abstraction

Applying individual influence paths to transformer-based models like BERT has computational and conceptual problems. BERT is denser in terms of model connections: each cell at each layer integrates information from *all* cells of the prior layer (as opposed to the pair of short-term and long-term connections in LSTM). This results in an intractable number of influence paths for algorithms requiring enumeration, even for processing the simplest of BERT variants. The denser architecture also diminishes the roles of individual paths in the implementation of even basic concepts like SVA. Finally, understanding contextualization requires explaining the impact of a sequence of words as opposed to a single word.

Our approach is two-fold: first we employ abstractions of sets of paths as the localization and influence quantification instrument; second, we discover influential patterns using a greedy search procedure that refines abstract patterns into more concrete ones, keeping the influence high.

Definition 3 (Multi-partite pattern) *Given endpoint nodes s and t a* multi-partite pattern *π is a sequence of edges $[n_{i_0} \to n_{i_1}, n_{i_2} \to n_{i_3}, \cdots]$ of a computational graph such that for any pair of adjacent edges $n_i \to n_j$ and $n_k \to n_l$ in π, there exists a path between n_j and n_k in the graph, and $n_{i_0} = s$, $n_{i_{-1}} = t$.*

If $n_j = n_k$ for every such case, then the pattern defines a path from s to t. In general, however, a pattern π abstracts a set of paths, written $\gamma(\pi)$ that follow the given sequence of edges but are free to traverse the graph between the edge constraints in any way. Interpreting paths and patterns as sets, we define $\gamma(\pi) \stackrel{\text{def}}{=} \{p \subseteq \mathcal{P} : \pi \subseteq p\}$ where $\mathcal{P}$ is the set of all paths from s to t.

Definition 4 (Pattern influence) *Given a computation graph and a DoI $\mathcal{D}$, the* influence *of a multi-partite influence pattern π, written $\mathcal{I}(\mathbf{x}, \pi)$ is:*

$$\mathcal{I}(\mathbf{x}, \pi) \stackrel{\text{def}}{=} \mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x})}\left[\prod_{(n_i,n_j)\in \textit{connections}(\pi)} \frac{\partial n_j(\mathbf{x})}{\partial n_i(\mathbf{x})}\right] = \sum_{p\in\gamma(\pi)} \chi(\mathbf{x}, p)$$

Where connections$(\pi) \stackrel{\text{def}}{=} \{(n_i, n_j) : n_i \to n_j \in \pi$ or $\pi = [\cdots, \cdot \to n_i, n_j \to \cdot, \cdots]\}$.

The definition indicates that the total influence of paths represented by a pattern can be computed without an enumeration of all those paths. Also note that influence of individual paths may be positive or negative so cancellation in the influence of a pattern which aggregates paths is possible.

Computation Graphs for BERT A given DNN can be expressed by many computational graphs. For computational and interpretability reasons, an ideal graph would contain as few nodes and edges as possible while exposing structures of interest. For BERT in particular, we propose *embedding-level graph* (Def. 5) to explain how the influence of input embeddings flow from one Transformer layer to another and to the eventual prediction of `[MASK]` and its detailed version *attention-level graph* (Def. 6) to demonstrate how influence from the input embedding flows through the attention block within each layer. A toy example in Fig. 5 demonstrates the connection and difference.

Definition 5 (Embedding-level Graph) *Given a BERT model f with $L+1$ layers (one input layer and L Transformer layers) and N embeddings, an embedding-level graph $\mathcal{G}_e \stackrel{\text{def}}{=} (\mathcal{V}_e, \mathcal{F}_e, \mathcal{E}_e)$ where $\mathcal{V}_e \stackrel{\text{def}}{=} \{n_i^l : \forall l \in \{0, \cdots, L\}, \forall i \in \{0, \cdots, N-1\}\} \cup \{n_w, n_q\}$. We let each node n_i^l denote an internal embedding $\mathbf{h}_i^l$, and let n_w, n_w denote the classification layer and QoI on top of the output embedding $\mathbf{h}_m^L$ that corresponds to `[MASK]`, respectively. $\mathcal{F}_e$ and $\mathcal{E}_e$ are inferred from $\mathcal{V}_e$ and f.*

Definition 6 (Attention-level Graph) *Given a BERT model f with $L+1$ layers (one input layer and L Transformer layers), N embeddings, A attention heads and its embedding-level graph $\mathcal{G}_e \stackrel{\text{def}}{=}$*

$(\mathcal{V}_e, \mathcal{F}_e, \mathcal{E}_e)$, an attention-level graph $\mathcal{G}_a \stackrel{\text{def}}{=} (\mathcal{V}_a, \mathcal{F}_a, \mathcal{E}_a)$ where $\mathcal{V}_a \stackrel{\text{def}}{=} \mathcal{V}_e \cup \{s_i^l, a_{i,j}^l, c_i^l : \forall l \in \{0, \cdots, L\}\}, \forall i \in \{0, \cdots, N-1\}, \forall j \in \{0, \cdots, A-1\}$. Let s_i^l denote the skip connection in each attention block such that $s_i^l(\mathbf{x}) = f_s(n_i^l(\mathbf{x})) = n_i^l(\mathbf{x})$, $a_{i,j}^l$ denote each attention head, and c_i^l denote all other operations outside the attention block. $\mathcal{F}_a$ and $\mathcal{E}_a$ are inferred from $\mathcal{V}_a$ and f.

Guided Pattern Refinement(GPR) Instead of enumerating the path space of $\mathcal{P}$ for discovering influential paths, we approximate a search by greedily refining patterns while attempting to keep their influence high. Starting with $\pi_0 = \emptyset$, representing all paths from s to t, we construct π_1 by adding an edge from a *guiding set* E_0 that maximizes the influence of the resulting pattern. The procedure is repeated with additional refinement:

$$\pi_{i+1} \stackrel{\text{def}}{=} \pi_i \cup \{e_i\}$$
$$e_i \stackrel{\text{def}}{=} \arg\max_{e \in E_i} \mathcal{I}(\mathbf{x}, \pi_i \cup \{e\})$$

The procedure can be repeated for a fixed number of steps or until some stopping criterion is reached, producing a sequence of patterns with decreasing abstraction: $\gamma(\pi_{i+1}) \subseteq \gamma(\pi_i)$. One could repeat the procedure until an individual path is produced though for reasons of interpretability, it might be preferable to reason about patterns instead of single paths.

The choice of guiding sets E_i at each iteration has a big impact on the resulting patterns both in terms of their influence significance and computational requirements. Smaller sets naturally require fewer options to enumerate but are likely to lead to less influential patterns. In our experiments we employ a simple guiding strategy: at iteration i, we focus on layer i of BERT. The guiding set E_i offers edges that would refine a path by forcing them to go through one of n_j^i nodes in case of the embedding-level graph or a choice of s_j^i or a_j^i in case of the attention-level graph. We will write π_i^e and π_i^a as the patterns resulting of the refinement process guided this way in the embedding and attention-level graphs, respectively, from input words at position i to a quantity of interest.

4 Evaluation

We demonstrate GPR by abstracting patterns to significant paths in the embedding and attention-level graphs of BERT. We first discuss the selected linguistic tasks, datasets and hyper-parameters, and then discuss quantitative evaluation of GPR in 4.2, and finally demonstrate how to use GPR to explain the contextualization of SVA in BERT in 4.3.

Tasks. We include the two linguistic tasks: subject-word agreement (SVA) and reflexive anaphora (RA). We explore different forms of sentence stimuli in each task: object relative clause (Obj.), subjective relative clause (Subj.), within sentence complement (WSC), and across prepositional phrase (APP) in SVA; number agreement (NA) and gender agreement (GA) in RA. We construct our SVA and RA datasets from Marvin & Linzen (2018) and Lin et al. (2019). For each dataset, we constructe MLM in a same way as Goldberg (2019). We sample 200 sentences evenly distributed across different sentence types (e.g. singular/plural subject & singular/plural attractor) with a fixed sentence structure from each task, so that the sentence length and the word types in each position are consistent across samples. The examples of each task can be found in the Appendix B.

QoI and Distributional Influence. We use the same QoI from Lu et al. (2020) where $q(\mathbf{y}_m) \stackrel{\text{def}}{=} y_{m,correct} - y_{m,incorrect}$, e.g. $y_{m,\texttt{IS}} - y_{m,\texttt{ARE}}$ for `she [MASK] happy`. We select $\mathcal{D}$ as an uniform distribution over a linear path from $\mathbf{x}_b$ to $\mathbf{x}$ in the input space for each word where $\mathbf{x}_b = \mathbf{x}_m^0$, the input embedding of `[MASK]`, as it can be seen as a word with "zero information".

Model. We choose $\text{BERT}_{\text{SMALL}}(L = 6, A = 8)$ in this paper from Turc et al. (2019), with comparable performance to $\text{BERT}_{\text{BASE}}$ (See Appendix B). The convergence analysis for approximating the influence can be found in Appendix. D.1.

4.1 Quantitative Analysis

We adapt two analysis, *concentration* and *model compression*, motivated by Lu et al. (2020), to evaluate whether π_i^e and π_i^a returned by GPR are significant (influential) in transmitting input signals for the agreement tasks.

Task	$\log_2 \lvert\mathcal{P}_e\rvert$	C_e^+	C_e^-	$\lvert\pi_e^+\rvert/\lvert\mathcal{P}_e\rvert$	acc.(π_e^+)	acc.(ori.)	acc.(rand.)
SVA							
Obj.	14.4	0.34	0.29	0.06	0.99	0.96	0.50
Subj.	14.4	0.30	0.31	0.09	0.74	1.00	0.50
WSC	13.8	0.27	0.38	0.03	1.00	1.00	0.52
APP	14.4	0.33	0.31	0.06	0.92	1.00	0.55
RA							
NA	14.4	0.31	0.32	0.04	0.68	0.83	0.54
GA	15.4	0.22	0.23	0.02	0.76	0.73	0.65

Table 1: Concentration and Model Compression Results. C_e^+ & C_e^-: The percentage of positive/negative embedding-level influence overall all positive/negative influence. $\mathcal{P}_e$: a set of all patterns (reachable by GPR) from input to QoI. π_e^+: a subset of $\mathcal{P}_e$ with only positive significant embedding-level patterns. acc.(ori.) and acc.(rand.) are the accuracies of the original model and a randomly compressed model which retains on average the same number of patterns as π_e^+.

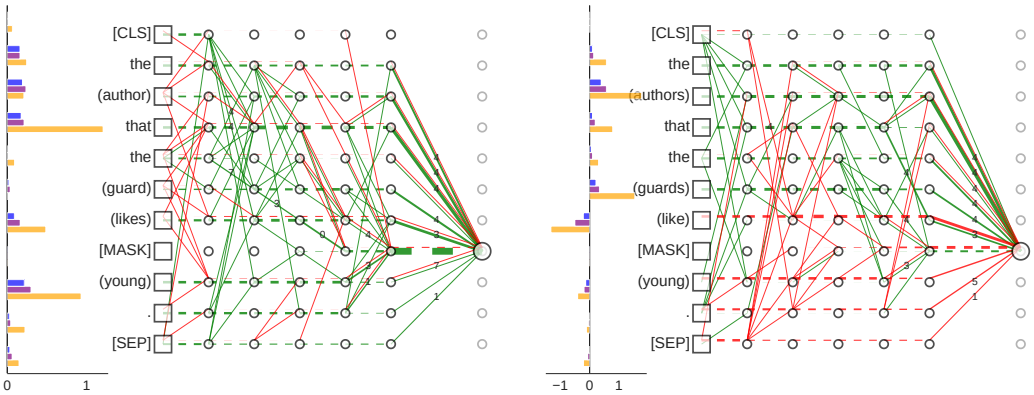
Task	$\log_2 \lvert\mathcal{P}_a\rvert$	C_a^+	C_a^-	$\lvert\pi_a^+\rvert/\lvert\mathcal{P}_a\rvert$	acc.(π_a^+)	acc.(rand.)
SVA						
Obj.	27.6	0.22	0.16	2.0e-7	0.74	0.51
Subj.	27.6	0.18	0.17	2.0e-7	0.56	0.49
WSC	27.0	0.16	0.24	1.8e-7	0.69	0.49
APP	27.6	0.18	0.15	1.6e-7	0.62	0.60
RA						
NA	27.6	0.18	0.18	1.4e-7	0.49	0.52
GA	28.6	0.14	0.15	1.4e-7	0.72	0.55

Table 2: Similar results of concentration and model compression as in Table 1, using attention-level paths.

Concentration. Given a sentence $\mathbf{x}_{0:N}$, the positive concentration C^+ is the percentage of the total influence of patterns with positive influence over the total positive distributional influence over all words $\mathbf{x}_i$, and similarly we have C^- for negative influence. High concentration suggests that most of the distributional influence flows across significant patterns, instead of sparely distributed over a large number of patterns or paths (e.g. the number of total paths in an embedding-level graph is more than 2^{14} for a sentence with 11 tokens).

Model Compression. To validate that the extracted patterns are actually significant for transmitting the information from the input embedding to the output QoI, we employ a model compression algorithm, inspired by those used in Lu et al. (2020). Since there is no direct way to only retain the portion of extracted patterns in a forward pass, we propose an approximating model compression procedure for both π^e and π^a compression: for each example, we only retain the nodes where positive patterns pass through while "churning" the nodes where no positive patterns pass though. The "churning" and "retaining" operations are conducted in the layer embeddings (or attention heads/skip connections). In each layer, the embedding nodes (or the attention/skip connection nodes) to be churned will be replaced by the embedding of `[MASK]` (zeros for attention/skip nodes), while the nodes to be retained will remain untouched. The retained and churned node together will be forward passed to the next layer.

Results. We demonstrate the concentration and model compression using π_i^e and π_i^a, in Table 1 and 2, respectively. In Table 1, we obeserve that $\chi(\mathbf{x}, \pi_i^e)$ accounts for a large portion of both positive and negative influence, range from 0.22 to 0.34 and 0.23 to 0.38, respectively. We also discover that the model compressed using positively influential patterns within π^e retains a large portion of the original accuracies, despite only using a really small number of patterns compared to the number of EMPs in total. The compression accuracy can also be compared against a random baseline, which compresses the same number of paths randomly and shows a low binary accuracy around half.



(a) singular subject + singular intervening noun(SS) (b) plural subject + plural intervening noun(PP)

Figure 3: Aggregated path across samples for task *SVA across Obj.* The legend and details for plots can be found in Figure 1. The words within parenthesis represent one instance of the word in that position. For aggregated plots of sentence types in Figure 1, see Appendix D.2.

Similar conclusion can be made for attention level patterns π^a in Table 2, which is a subset of π^e. They retain a even larger portion of π^e, compared to π^e with respect to the total influence, while showing a smaller compression accuracy due to more nodes churned and replaced.

4.2 Explaining Contextualization of SVA Across Object Relative Clause

In this section, we explain internal contextualization between word embeddings by visualizing the significant patterns π^a found by GPR (as $\pi^e \subseteq \pi^a$). We also show results on other tasks in the Appendix. D.2.

Observations of Fig. 3. First we observe that in all four sentence types(PP, PS, SP, SS) of Figure 1 and 3, both words in the subject phrase ("the" and the "noun") exert a positive input influence on the correct prediction of the verb, and the attractor exerts positive input influence when agreeing with the subject in number and negative otherwise. This is true for both $\chi(\mathbf{x}_i, \pi_i^e)$ and $\chi(\mathbf{x}_i, \pi_i^a)$.

"Copy and Transfer" We also observe that there are many straight parallel lines in the graphs, where the influence is simply passed from layer to layer at the same word position, mostly using skip connections. . **We speculate that the reason that attentions can be effectively pruned without compromising the performance in prior works (Michel et al., 2019), is because between most layers, the information from layer embeddings does not travel through attention block at all: they are simply "copied" to the next layer through the skip connections.** Zooming in on the most significant EMP and AMP, we observe that the subject phrase mostly travels through a straightforward significant pattern where the influence stays at the same subject position through skip connections across the lower layers, and influencing the outcome only through attention head 4 in the last layer. It indicates that the BERT model mostly picks up the signal from the subject input embedding without much contextualization at all. In the end, a single attention head 4 is used for relaying the subject number information to `[MASK]`. This is true for both positive and negative MP from attractors as well. However, we do observe in some cases the attractor/subject is relayed through a lower layer (e.g. Figure 1b). In , we observe that all the above patterns are prevalent in other tasks (such as the contextualization of propositions in WPP case), although different heads might be used for the "transfer operations" in different tasks.

The Role of "that" However, we do observe that the influence from the singular subjects(SS and SP) is weaker, especially compared to the attractors. It turns out that "that" behaves differently for singular subjects from plural subjects. In singular subjects, "that" behaves as though it is a singular noun, traveling through the same straightforward pattern as the subject. In plural subjects, however, "that" behaves more like a grammatical marker(relativizer): in PP(Figure 3b) and PS(Figure 1a), the MPs from "that" have a strong contextualization edge from itself to either the subject or the clausal

verb in the second to last layer. We speculate "that" in plural subject sentences encodes the clausal information, and enables the model to identify the correct subject. This is also shown by the fact that the attractor in PS case has almost no influence, compared to the high negative influence from the attractor in the SP case. It is also directly correlated with the SP case making more mistakes than PS case (See Appendix B).

5 Related Work

Previous work has shown the encoding of linguistic knowledge within BERT(Devlin et al., 2018). For example, diagnostic classifiers trained on output and internal representations discovers that BERT encodes many types of linguistic knowledge(Elazar et al., 2020; Hewitt & Liang, 2019; Tenney et al., 2019a;b; Jawahar et al., 2019; Klafka & Ettinger, 2020; Liu et al., 2019; Lin et al., 2019), ranging from syntactic concepts to more complicated semantic ones. Goldberg (2019) in particular discover that SVA and RA in complex clausal structures is better represented in BERT compared to an RNN model. Such success are partially explained by (Coenen et al., 2019; Hewitt & Manning, 2019), which show that contextual embeddings in BERT can encode syntactic structures hierarchically, comparable to those represented in a dependency tree. However all these analysis is done on frozen contextual embedding layers, the exact mechanism a concept is encoded from input to output is not explored.

Another line of work in interpreting BERT concerns analyzing the self-attention weights of BERT(Clark et al., 2019; Vig & Belinkov, 2019; Lin et al., 2019), where attention heads are found to have direct correspondences with specific dependency relations. However, attention weights as interpretation devices has been controversial(Serrano & Smith, 2019; Brunner et al., 2020), and empirical analysis has shown that attention can be perturbed or pruned with the same or even better performances(Kovaleva et al.; Michel et al., 2019; Voita et al., 2019). More importantly, our work demonstrate that attention mechanisms are only part of BERT computation graph, with each attention block complemented by additional architecture such as dense layer and skip connections. The strong influence passing through skip connections also corroborates the findings in Brunner et al. (2020), which finds input tokens retains mostly to their identity.

Besides pruning attentions, other works(Prasanna et al., 2020; SANH et al.; Jiao et al., 2019) also show that BERT is overparametrized and can be greatly compressed. Our work to some extent corroborates that point by pointing to the sparse gradient flow, while employing model compression only to verify the significance of the extracted graph.

Recent work introducing influence paths (Lu et al., 2020) offers another form of explanation. Influence paths quantify the effect of model inputs on model outputs that traverse *a particular* path through a model. The approach refines the influence-directed explanations framework of Leino et al. (2018) which instruments a neural model with input a distribution of interest (e.g. datasets meant to exercise a concept such as SVA) and an output quantity of interest (e.g. a measure of SVA) in order to compute a gradient-based measure of input influence (referred to as attribution) such as Integrated Gradient (Sundararajan et al., 2017) and to discover internal model elements most responsible (having highest influence) on the given concept. Lu et al. (2020) decomposed the attribution to path-specific quantities localizing the implementation of the given concept to paths through a model. The authors demonstrated that for LSTM models, a single path is responsible for most of the input-output effect defining SVA, and explored the effects of unhelpful nouns which showed negative influence on SVA. We describe the limitations of this methodology when applied to BERT in Sec. 3.

6 Conclusion

We have demonstrated how to use multi-partite influence patterns to localize a DNN model's handling of a concept of interest and along with a pattern refinement method we how BERT handles subject-verb number agreement and reflexive anaphora. We quantitatively validated the use of influence patterns in BERT by way of compression experiments and the concentration of discovered patterns. We qualitatively and visually demonstrated BERT's contextualization in the two tasks using our methodology. Our formalism and methods are general enough to apply to the analysis of other aspects of BERT and other models.

REFERENCES

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11:1803–1831, 2010.

Gino Brunner, Yang Liu, Damian Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. On identifiability in transformers. In *International Conference on Learning Representations*, 2020.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*, 2019.

Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, and Martin Wattenberg. Visualizing and measuring the geometry of bert. *arXiv preprint arXiv:1906.02715*, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. When bert forgets how to pos: Amnesic probing of linguistic properties and mlm predictions. *arXiv preprint arXiv:2006.00995*, 2020.

Yoav Goldberg. Assessing bert's syntactic abilities. *arXiv preprint arXiv:1901.05287*, 2019.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, 2019.

John Hewitt and Christopher D Manning. A structural probe for finding syntax in word representations. 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. 4129–4138, 2019.

Sarthak Jain and Byron C Wallace. Attention is not explanation. 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. 3543–3556, 2019.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does bert learn about the structure of language? 2019.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*, 2019.

Josef Klafka and Allyson Ettinger. Spying on your neighbors: Fine-grained probing of contextual embeddings for information about surrounding words. *arXiv preprint arXiv:2005.01810*, 2020.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. Revealing the dark secrets of BERT. Association for Computational Linguistics.

Klas Leino, Shayak Sen, Anupam Datta, Matt Fredrikson, and Linyi Li. Influence-directed explanations for deep convolutional networks. In *2018 IEEE International Test Conference (ITC)*, pp. 1–8. IEEE, 2018.

Yongjie Lin, Yi Chern Tan, and Robert Frank. Open sesame: Getting inside bert's linguistic knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 241–253, 2019.

Nelson F Liu, Matt Gardner, Yonatan Belinkov, Matthew Peters, and Noah A Smith. Linguistic knowledge and transferability of contextual representations. *arXiv preprint arXiv:1903.08855*, 2019.

Kaiji Lu, Piotr Mardziel, Klas Leino, Matt Fredrikson, and Anupam Datta. Influence paths for characterizing subject-verb number agreement in LSTM language models. Association for Computational Linguistics, 2020.

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, 2018.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, pp. 14014–14024, 2019.

Christopher Olah. Calculus on computational graphs: Backpropagation, 2015. URL `http://colah.github.io/posts/2015-08-Backprop/`.

Sai Prasanna, Anna Rogers, and Anna Rumshisky. When bert plays the lottery, all tickets are winning. *arXiv preprint arXiv:2005.00561*, 2020.

Victor SANH, Lysandre DEBUT, Julien CHAUMOND, Thomas WOLF, and Hugging Face. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Sofia Serrano and Noah A. Smith. Is attention interpretable? In *ACL*, 2019.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3319–3328. JMLR. org, 2017.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*, 2019a.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. What do you learn from context? probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*, 2019b.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962v2*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

Jesse Vig and Yonatan Belinkov. Analyzing the structure of attention in a transformer language model. *arXiv preprint arXiv:1906.04284*, 2019.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *ACL (1)*, 2019.

A APPENDIX: PATH METHOD

A.1 COMPUTATION GRAPH

We formally define computation graphs and paths in this section.

Definition 7 (Computation Graph) *A computation graph with N nodes is defined as $\mathcal{G} := (\mathcal{V}, \mathcal{F}, \mathcal{E})$, where $\mathcal{V} := \{n_0, n_1, ..., n_{N-1}\}$ is a set of nodes, $\mathcal{F} := \{f_0, f_1, ..., f_{N-1}\}$ is a set of function on each node and a set of directed edges $\mathcal{E}$. There is an edge pointing from node n_s to node n_t if and only if $(n_s, n_t) \in \mathcal{E}$. We write the output of an node n with the input $\mathbf{x}$ to the graph as $n(\mathbf{x}) := f_n(n_0(\mathbf{x}), ..., n_m(\mathbf{x}))$ where $\forall p \in \{0, ..., m\}$, $(n_p, n) \in \mathcal{E}$. The Jacobian of the output of node n_1 w.r.t the output of n_2 is denoted as $\frac{\partial n_1(\mathbf{x})}{\partial n_2(\mathbf{x})}$.*

A network's corresponding computation graph may not be unique. For example, one can define a graph with a number of nodes where each node only represent basic operations, e.g. $+, -, *, /$, or alternatively the user defines a graph with only a fewer nodes where each node represents the computation of a stack of layers in the original model. Our goal is to explain how a particular path from the input node that finally influences the model's output node. We formalize the definition of *path* as follows and we show an example of path in Fig. 4

Definition 8 (Path) *Given a computation graph $\mathcal{G} = (\mathcal{V}, \mathcal{F}, \mathcal{E})$, a source node n_s and a target node n_t, we say an ordered set $\pi, |\pi| = N \geq 2$, is a* path *from n_s to n_t if and only if the following conditions hold: 1) $\pi_0 = n_s$; 2) $\pi_{-1} = n_t$; and 3) $\forall i \in \{0, ..., N-a-1\}$, $(\pi_i, \pi_{i+a}) \in \mathcal{E}$ holds only at $a = 1$. Denote the Jacobian of the last node in π w.r.t the first node on an input $\mathbf{x}$ as $\nabla_{\mathbf{x}}\pi := \prod_{i=0}^{|\pi|-2} \frac{\partial \pi_{i+1}(\mathbf{x})}{\partial \pi_i(\mathbf{x})}$*

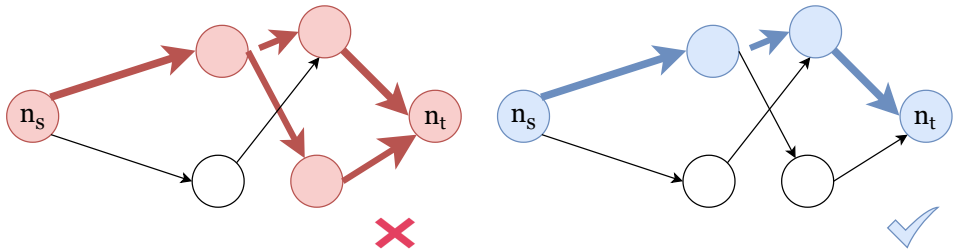


Figure 4: A toy example of a path from n_s to n_t. The left one in red is not a path since an intermediate node connects to multiple successors.

A.2 DISTRIBUTIONAL INFLUENCE AND PATH INFLUENCE

In this section, we show the relation between the distributional influence and path influence. We discuss the case when the model is defined as $f : \mathbb{R}^d \to \mathbb{R}$ given a model F with vector output can also be decomposed as $F(\mathbf{x}) = \sum_i f_i(\mathbf{x}), f_i : \mathbb{R}^d \to \mathbb{R}$.

Theorem 1 *Assume a continuous, differentiable and real-valued model $f : \mathbb{R}^d \to \mathbb{R}$ is associated with a computation graph $\mathcal{G}$. Let the output of the source node n_s and the target node n_t be $n_s(\mathbf{x}) = \mathbf{x}$ and $n_t(\mathbf{x}) = f(\mathbf{x})$, and denote $\mathcal{P}$ as a set of all paths from n_s to n_t, then $\mathbb{E}_{\mathbf{z}\sim\mathcal{D}(\mathbf{x},\boldsymbol{\lambda})} \frac{\partial f(\mathbf{z})}{\partial \mathbf{z}} = \sum_{\pi\in\mathcal{P}} \mathbb{E}_{\mathbf{z}\sim\mathcal{D}} \nabla_{\mathbf{z}}\pi$.*

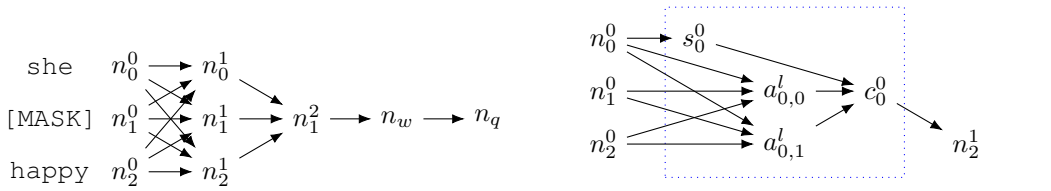
The proof is straightforward: the summation and expectation is exchangeable due to the fact sampling a path $\pi \in \mathcal{P}$ is irrelevant with sampling a point $\mathbf{z}$ (see Def. 6 in Lu et al. (2020)). The rest of the proof is to show $\frac{\partial f(\mathbf{z})}{\partial \mathbf{z}} = \nabla_{\mathbf{z}}\pi$, which is given by applying chain rule in the computation graph. One rigorous proof is provided by Theorm 1.4.4 in Olah (2015).

B APPENDIX: INTRODUCTION TO LINGUISTIC TASKS

Task	Type	Example	BERT Small	BERT Base
SVA				
Object Relative Clause	SS	the author that the guard likes [MASK(is/are)] young	1	1
	SP		0.92	0.96
	PS		0.9	0.98
	PP		1	1
Subject Relative Clause	SS	the author that likes the guard [MASK(is/are)] young	1	1
	SP		1	0.96
	PS		1	0.98
	PP		1	1
Within Sentence Complement	SS	the mechanic said the author [MASK(is/are)] young	1	1
	SP		1	1
	PS		1	1
	PP		1	1
Across Prepositional Phrase	SS	the author next to the guard [MASK(is/are)] young	1	0.99
	SP		1	0.98
	PS		0.98	0.98
	PP		1	1
Reflexive Anaphora				
Number Agreement	SS	the author that the guard likes hurt [MASK(himself/themselves)]	0.66	0.6
	SP		0.66	0.74
	PS		0.83	0.83
	PP		1	0.96
Gender Agreement	MM	some wizard who can dress our man can clean [MASK(himself/herself)]	0.78	1
	MF		0.32	0.96
	FF		1	0.9
	FM		0.8	0.66

Table 3: Example of each agreement task and their performance on two BERT models, first 5 tasks are sampled from Marvin & Linzen (2018), the last task is sampled from dataset in Lin et al. (2019)

C Appendix: Computational Graphs



(a) embedding-level graph

(b) A sub-region of the attention-level graph by exposing the attention structure between n_0^0 and n_2^1 from its embedding-level graph.

Figure 5: The embedding-level and attention-level graphs for a toy BERT model with two transformer layers, two attention heads, three word embeddings and an example input.

D Appendix: Experiment Details

D.1 Convergence of Completeness

D.2 Influence Graphs for Other Tasks

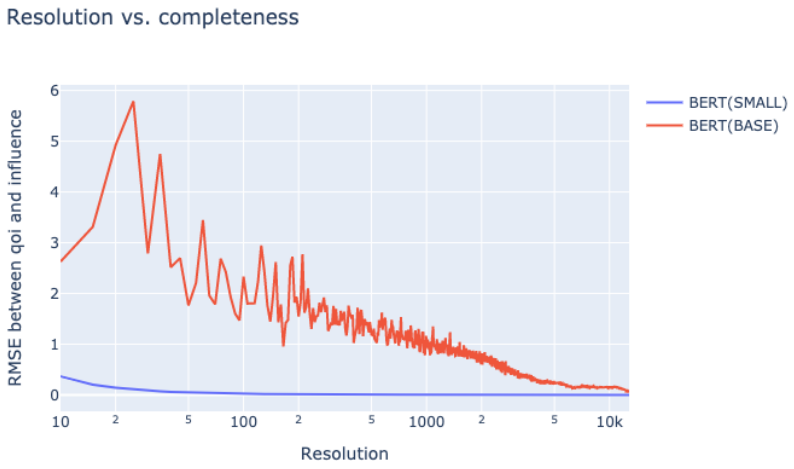
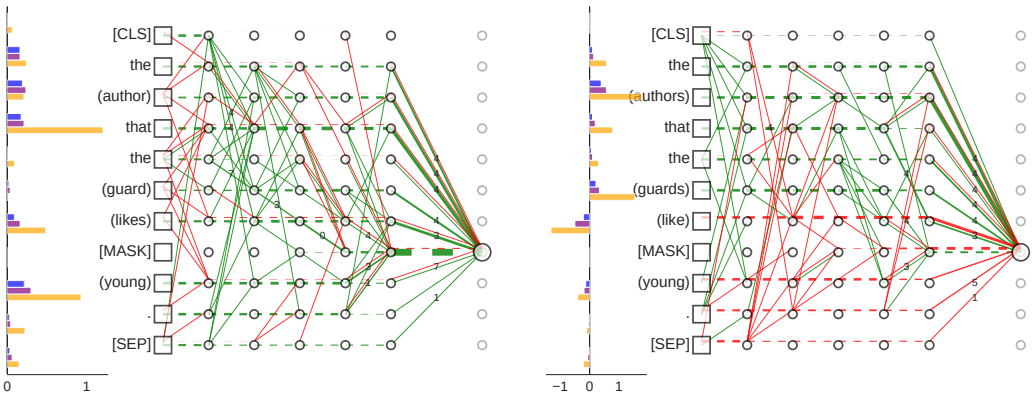


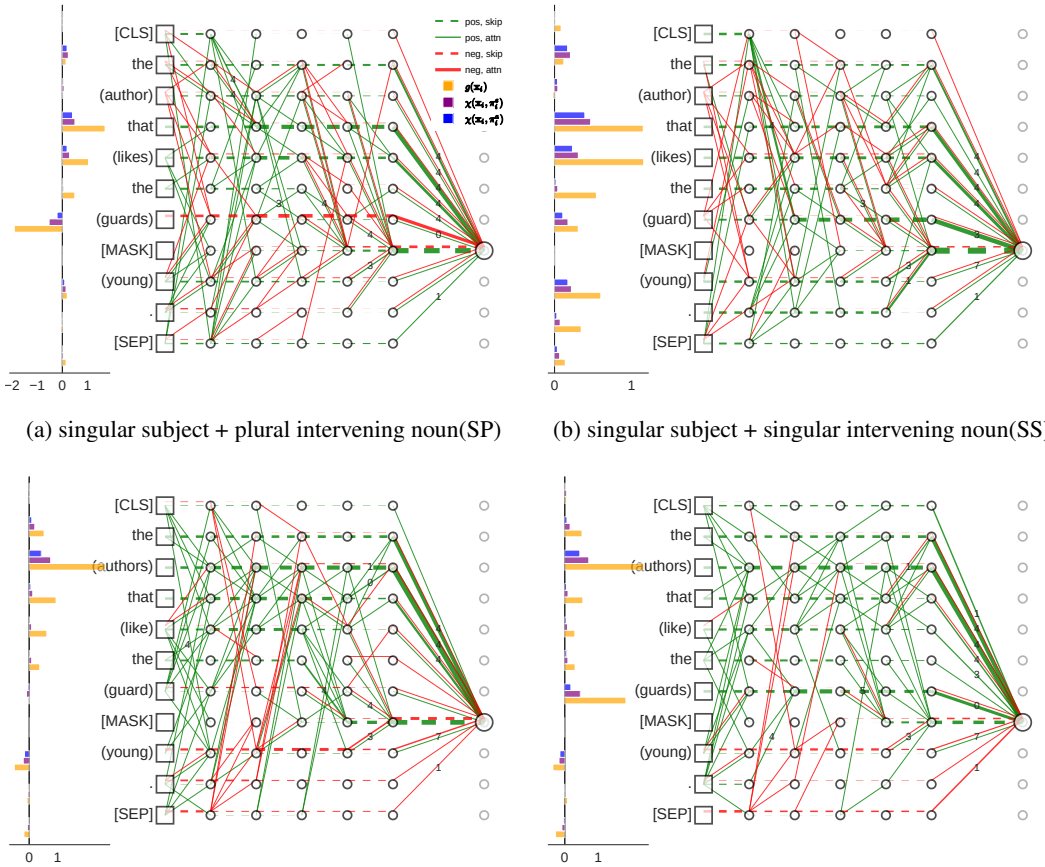
Figure 6: Convergence Analysis for Calculating the Distributional influence(IG) for SVA Across Object Relative Clauses for $BERT_{SMALL}$(used in this paper) and $BERT_{BASE}$, The much slower convergence of the larger BERT model is likely due to the complicated decision boundaries of larger BERT, masking the output sensitive to small perturbations.



(a) singular subject + singular intervening noun(SS)

(b) plural subject + plural intervening noun(PP)

Figure 7: SVA Across Object Relative Clause.



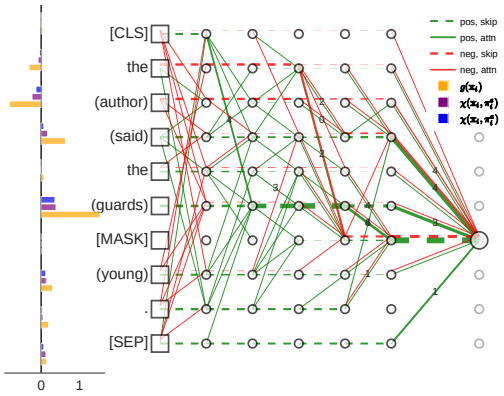
(a) singular subject + plural intervening noun(SP)

(b) singular subject + singular intervening noun(SS)

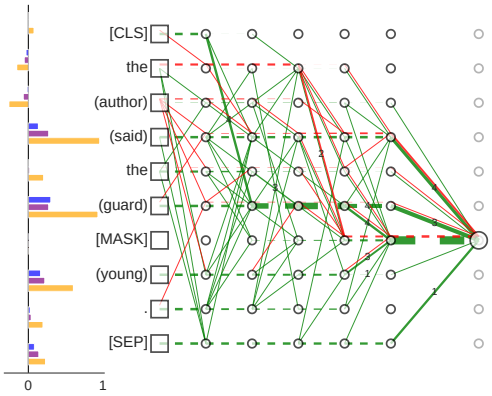
(c) plural subject + singular intervening noun(PS)

(d) plural subject + plural intervening noun(PP)

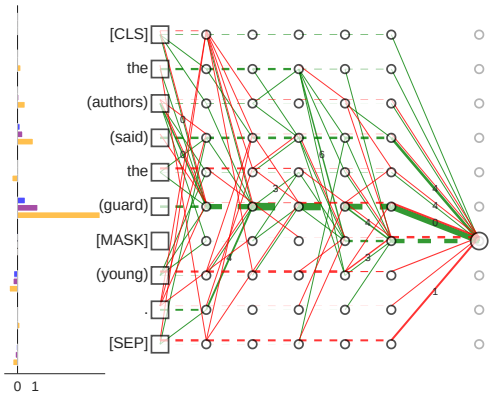
Figure 8: SVA Across Subject Relative Clause.



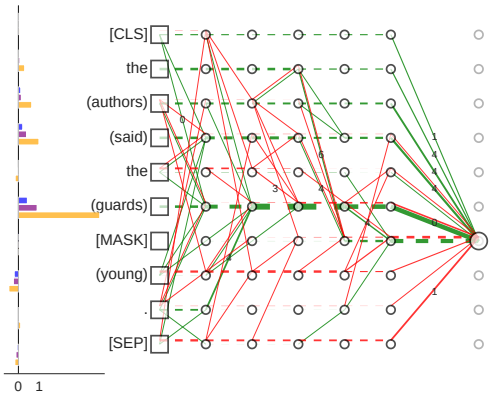
(a) singular subject + plural intervening noun(SP)



(b) singular subject + singular intervening noun(SS)

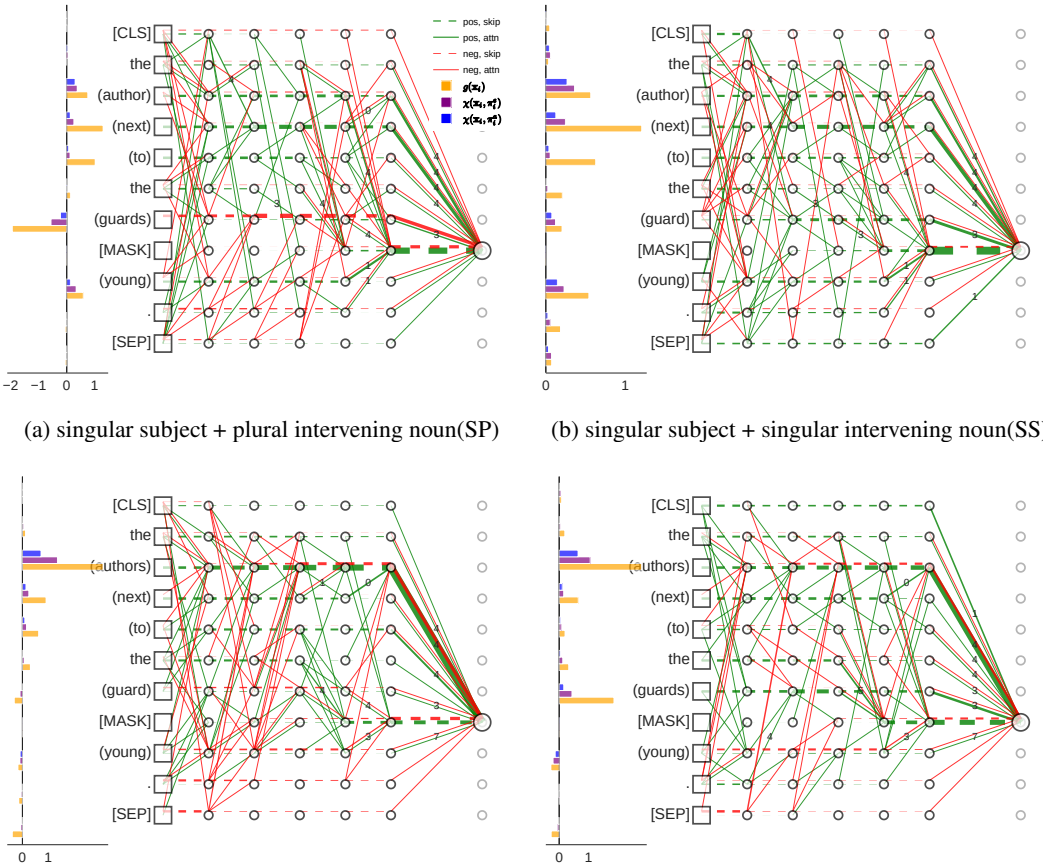


(c) plural subject + singular intervening noun(PS)



(d) plural subject + plural intervening noun(PP)

Figure 9: SVA Within Sentence Complements.



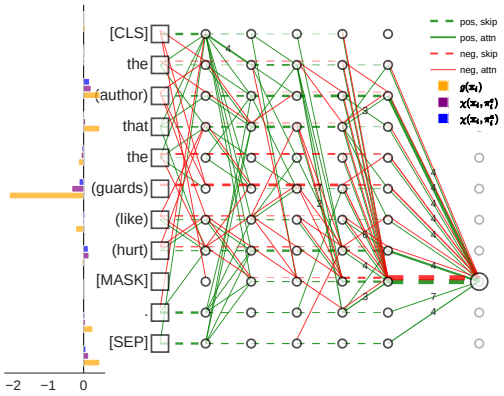
(a) singular subject + plural intervening noun(SP)

(b) singular subject + singular intervening noun(SS)

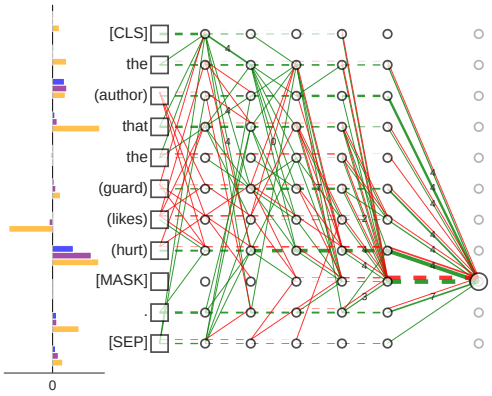
(c) plural subject + singular intervening noun(PS)

(d) plural subject + plural intervening noun(PP)

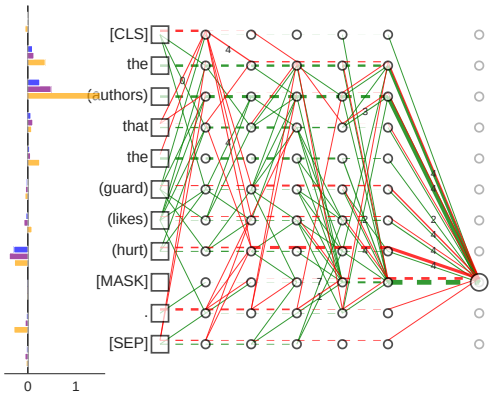
Figure 10: SVA Across Prepositional Phrase.



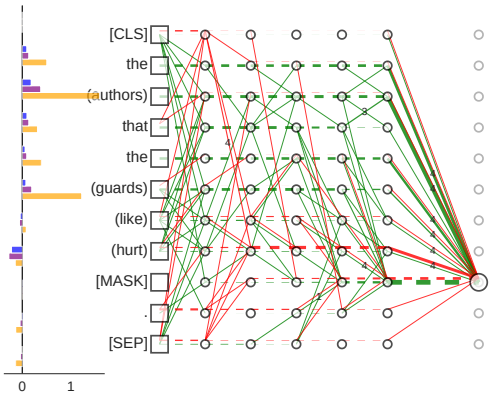
(a) singular subject + plural intervening noun(SP)



(b) singular subject + singular intervening noun(SS)

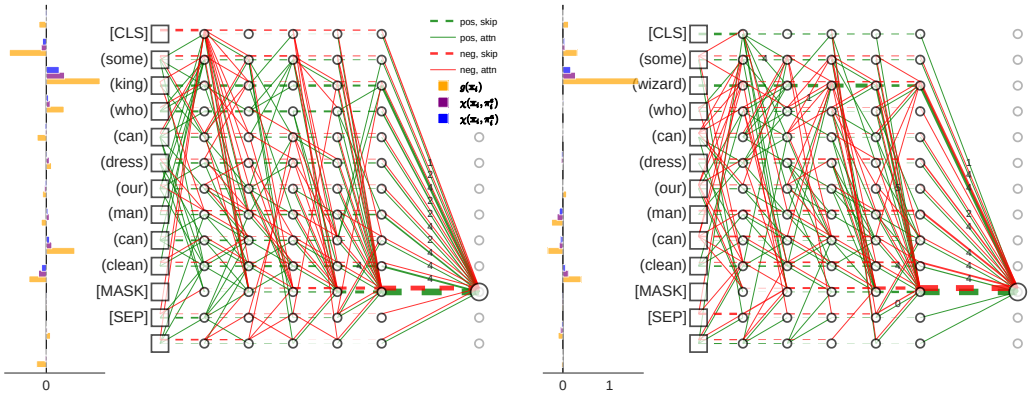


(c) plural subject + singular intervening noun(PS)



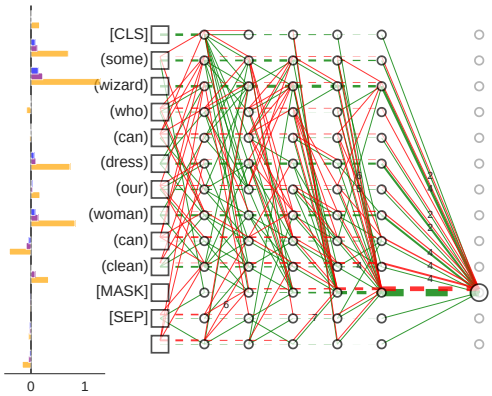
(d) plural subject + plural intervening noun(PP)

Figure 11: RA: Number Agreement

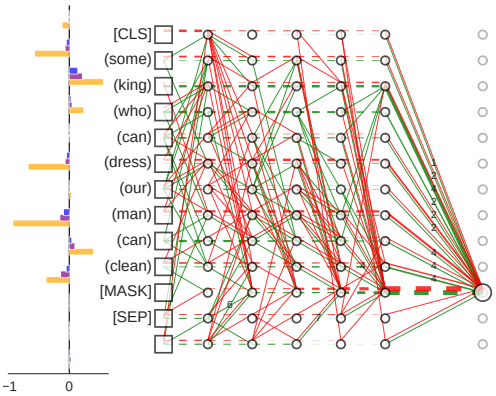


(a) male subject + male intervening noun(MM)

(b) female subject + male intervening noun(FM)



(c) female subject + female intervening noun(FF)



(d) male subject + female intervening noun(MF)

Figure 12: RA: Gender Agreement