TREETOP: Topology-Aware Fine-Tuning for LLM Conversation Tree Understanding

Jashn Arora, Rahul Madhavan, Karthikeyan Shanmugam, John Palowitch, Manish Jain Google DeepMind {arorajashn, rahulmadhavan, karthikeyanvs, palowitch, manishjn}@google.com

Abstract

Abstract

While Large Language Models (LLMs) have dominated a wide diversity of natural language tasks, improving their capabilities on *structured* inputs such as graphs remains an open challenge. We introduce TREE-TOP, a fine-tuning framework for LLMs that significantly improves their ability to reason over structural relationships in multi-party discussion *trees*, e.g. on social media platforms. TREETOP is a novel set of 17 tasks designed to test the ability of LLMs to selectively focus on the structure and/or content of conversation tree graphs. We find that LLMs fine-tuned on TREETOP significantly outperform all baseline models (including state-of-the-art GNNs) in multiple settings: generalizing to unseen TREETOP tasks, and performance on downstream social media inference tasks (e.g. controversy detection), including their challenging "early-detection" variants. TREETOP charts new ground toward LLMs with generalized understanding of structured inputs.

1 Introduction

Large Language Models (LLMs) have achieved state-of-the-art over a vast landscape of tasks that can be cast as token sequence-to-sequence problems (Zhao et al., 2023b; Srivastava et al., 2022), partially through the combined effect of instruction fine-tuning (Wei et al., 2021) and scaling (Chung et al., 2024). A current open challenge for LLM capabilities is the handling of structured inputs (such as tables, e.g. Sui et al. (2024)), where the output depends strictly on tokens distributed throughout the input according to a certain pattern.

Recently, graph-structured inputs have emerged as one of the next frontiers in structured inputs for LLMs (Pan et al., 2024; Chen et al., 2024b), with a wide variety of models being crafted for graphs representing complex systems like social and biological networks (Wu et al., 2020; Zhang et al., 2020). As such, enabling LLMs to perform graph-structured tasks has the potential to considerably expand their scope. Existing efforts evaluate and improve LLM performance on graph problems such as edge existence and counting (Fatemi et al., 2023; Wu et al., 2024). There is already evidence that some LLMs, appropriately trained or prompted, can outperform Graph Neural Networks (Scarselli et al., 2008; Chami et al., 2022) on graph learning tasks such as node classification and link prediction (Ye et al., 2024).

In this paper, we continue this line of research by developing LLM capabilities for *conversation graph* inputs: graphs that encode online forum discussions by mapping replies to their parent comments, stemming from the root post. Often called conversation "trees", these graphs are ubiquitous due to the proliferation of online social platforms, and they are the input to many important learning tasks, such as misinformation detection, controversy detection, and trend prediction (Olteanu et al., 2019;



Figure 1: TREETOP framework (in) provides a dataset of 12 structural tasks to improve the LLM's ability to understand and reason over structure via fine-tuning. It also provides 5 unseen tasks to evaluate generalization of the LLM-TT model.

Zeng and Tang, 2021; Ji et al., 2021). Furthermore, conversation trees are directed, acyclic, and temporal, creating unique topological learning challenges, and distinguishing them from the usually undirected, cyclic, and static graphs found in standard GNN benchmarks (Hu et al., 2020a). For these reasons, conversation trees have been studied in their own sub-field of machine learning, admitting a wide range of approaches from GNNs (Xu et al., 2023) to LSTMs (Mukiri and Burra, 2023) to novel combinations thereof (Patel et al., 2022).

Another distinguishing aspect of conversation trees is that the graph itself defines a flow through a multi-agent discussion. Each node is attributed with a natural-language utterance, which can be informed by any existing utterances in the conversation at the time of posting. Also, the graph topology and the discussion can interact in potentially meaningful ways: a linear, back-and-forth sub-graph between two discussants has a signature quite different from the star-like pattern created, e.g., by a comment that receives many one-off replies. We observe that these phenomena create a rich space of new challenges for LLMs. However, the application of LLMs to conversation trees has yet to receive attention. As a first step in this space, we introduce **Tree Top**ology-Aware Fine-Tuning (TREETOP): a novel learning framework for LLMs that significantly increases their performance across a range of tasks on conversation trees. As shown in Figure 1, TREETOP provides a collection of structural QA tasks defined over conversation trees that can be used to fine-tune an LLM before zero/few-shot deployment or downstream fine-tuning. Specifically, our contributions are:

1. We introduce novel machinery for LLM handling of conversation trees, including a prompt framework and tree-specific text encoding. We also make the code for the TREETOP framework available here¹ to advance further research in this field.

2. We propose TREETOP, a conversation graph fine-tuning framework including 17 structural tasks that target an LLM's ability to *reason* about the conversation, such as tree navigation and user-user reply counting. We show that fine-tuning using the TREETOP framework allows LLMs to generalize to even to *unseen* structural tasks.

3. Through extensive experimentation, we show that LLMs fine-trained with the TREETOP framework significantly outperform non-TREETOP counterparts – whether the LLMs are zero-shot, few-shot, or fine-tuned – on four social media tasks, including controversial post detection (Hessel and Lee, 2019), rumor detection (Zubiaga et al., 2016), fake news detection (Nakamura et al., 2019), and winning argument detection (Tan et al., 2016). Furthermore, TREETOP LLMs that are further

¹https://tinyurl.com/treetopframework

fine-tuned for these tasks achieve new SOTA compared with GNN approaches, including on "early detection" versions of the four tasks.

2 Related Work

LLM-based approaches to graph problems. Graph learning with LLMs is a nascent area of research (Tang et al., 2023; He et al., 2023). Guo et al. (2023) study whether LLMs can understand graph structural information, and Huang et al. (2023) show that LLMs tend to process graphs like contextual paragraphs. Among other works, Chen et al. (2024b) leverage LLMs both as a generator for explanations and a classifier for graph problems. Zhao et al. (2023a) encode graphs into natural language, and Ye et al. (2024) extend their work by instruction-tuning LLMs, but they primarily focus on node classification tasks. Both Müller et al. (2023); Fatemi et al. (2023) introduce novel schemes for encoding graphs in prompts. By-and-large, these studies have tackled tasks on undirected, cyclic, static graphs, with applications geared toward standard GNN benchmarks (Hu et al., 2020a). In a parallel line of work, language models have been used to improve GNN performance. We expand on this related work in Appendix B. Extending this area of research, we focus on fine-tuning LLMs to solve learning tasks on *conversation trees*, which are directed, acyclic, and temporal graphs capable of representing a wide variety of complex human interaction sequences.

Conversation Trees. The ubiquity of social media has created a global shift in information consumption and human discussion (Akram and Kumar, 2017). Information on social media is frequently presented as a central post and its subsequent comments, creating a dynamic exchange between the original poster and other users. The study of these graphs, or "conversation trees" has been motivated by several tasks which are central themes in social media datagraphs (Yan et al., 2023). Some canonical problems in this domain are information flow (Bakshy et al., 2012), controversial post detection (Benslimane et al., 2021; Garimella et al., 2018), and fake news detection (Lillie and Middelboe, 2019; Han et al., 2020), among others. Similarly, bias detection (Chen et al., 2022; Zhu et al., 2022), fraud detection (Liu et al., 2023b; Zeng and Tang, 2021), event detection (Gao et al., 2021; Ji et al., 2021) and malicious behaviour detection (Wu et al., 2022; Dou, 2022) are other active research areas. Early detection (Zhou et al., 2019; Tian et al., 2020) in such cases is also critical, since it enables proactive interventions by social media platforms. Here, we propose methodology for enabling LLM competency across this entire space of problems.



(a) Post Comment Tree

(b) User space mapping of a Post Comment Tree

U₅

U₄

U3

Un

Figure 2: In TREETOP, we encode the graph in Figure 2a (actual conversation tree) directly within the prompt, and TREETOP reasons over the user-user graph (Figure 2b) on its own.

3 Methodology

We now describe our core contribution: TREETOP, an instruction-tuning framework for LLM generalization on conversation tree learning problems. Figure 1 illustrates the high-level pipeline (see Appendix Figure 5 for a flow diagram). TREETOP consists of a novel set of structural tasks on conversation trees to fine-tune an LLM. The purpose of these tasks is to improve the LLM's ability to understand structure of conversation trees, allowing the model to follow both the content and the flow of the discussion between the constituent users.

3.1 Representation of Social Media as Conversation Trees

Figure 2a shows a typical social media conversation which we directly encode as a prompt in our framework. Here, c_0 represents the top-level post, i.e. the root node of the conversation tree. Other posts c_i are replies to either the top-level post or other replies. In our framework, we encode only the tree structure (Figure 2a) in our prompt, leaving the interaction graph (Figure 2b) to be learned through fine-tuning on structural tasks.

Encoding. We encode a tree node, and the entire tree from Figure 2a as follows:

$$\begin{split} \texttt{encoding(comment)} &:= \langle \texttt{Comment-ID}, \texttt{User-ID}, \texttt{Content}, \texttt{Parent}, \texttt{Node Features} \rangle \\ \texttt{encoding(conversation tree)} &:= \big[\langle \texttt{encoding(comment_0)} \rangle \langle \texttt{encoding(comment_1)} \rangle \dots \big] \end{split}$$

For example, if we choose timestamp as a node feature, the tuple representation of the node corresponding to c_2 in Figure 2a is $\langle c_2, u_2, -\text{content}, c_0, t_2 \rangle$. Multiple types of node-level features, such as "verified status" or "karma", may be available on different platforms and can also be captured in our encoding. We sort the comments by timestamp in the tree encoding, i.e. the main post is the first element (comment₀) in the tree encoding. We use timestamp to sort to mimic the natural user experience on social media platforms – a user who views a post at time t can see all the comments prior to time t on that post.²

3.2 Structural Tasks

The core idea of TREETOP is to enhance LLM understanding of tree topology, and use this enhanced understanding alongside their inherent language understanding for downstream applications. Thus, to achieve this goal, we train LLMs on primitive topological tasks on conversation trees. Our approach is analogous to the use of graph motifs (Paranjape et al., 2017) for complex graph-based computations – we believe that LLMs will be able to compose multiple topological primitives together to solve general tree inference problems.

The primary workhorse of our framework is thus a collection of 17 reasoning tasks over conversation trees, illustrated in Figure 3. 12 of these tasks are used for fine-tuning, and the remaining used for evaluation. We designed these tasks to enable/evaluate four different "proficiency" categories on trees: (i) comment \times comment tasks, that focus on the relationships between comments; (ii) user \times user tasks, that focus on the relationships between comments; (ii) user from the conversation trees (refer Figure 2b); (iii) node characteristics tasks, that focus on the topological properties of the entire conversation tree.



Figure 3: Structural tasks used in the TREETOP framework. These structural tasks are divided into four "proficiency" classes, as described in Section 3.2. The tasks used during fine-tuning are shown in blue, and the tasks used for evaluation are shown in green.

For instance, Are_one_hop_neighbors tests if one provided comment is a direct response to another provided comment. Similarly, In_triangle if two provided users participate in a triangular

²In case a dataset doesn't provide us with the timestamp, we implement a breadth-first sort order in the encoding.

discussion with a third user, as explained in Example 1 of Section 3.2. We provide descriptions of all these tasks in Tables 6 and 7 in Appendix C.1. Each task is encoded as a Yes or No question. We design multiple prompt styles to phrase the question for each structural task. Some of these prompts use graph-topology based language (e.g. "Whether $\langle C2 \rangle$ has more than 3 children?") and some use language relevant to social media platforms (e.g. "Whether comment $\langle C2 \rangle$ has more than 3 replies?"). All the variations in the prompts are described in Appendix C.3.

3.2.1 Structural Task Corpus Creation

We created our structural task dataset using a small subset of the Pushshift Reddit data released by the authors of the Pushshift platform (Baumgartner et al., 2020), available for download here. This dataset contains all the posts and comments of Reddit that were posted in the month of April 2019. Our fine-tuning corpi are built from a random sample S of 100K conversation trees from this dataset. For each task, we sample an equal number of positive and negative examples by mining the trees from S, described further in the next section. Using this approach, we are able to generate any number of labeled questions, for any task. As described further in Section 4, we fine-tune both closed-source and open-source LLMs using the TREETOP framework. We create a corpus with 10k questions per task for these experiments; we provide multiple data ablation studies on this corpus in Appendix H.

3.2.2 Structural Task Examples

We describe three tasks here, and detail the rest in the Appendix C.1. To sample questions for a given task, we go through our conversation tree corpus S, and detect the structure of interest (e.g. wedge / triangle / long chain / etc.) in each conversation tree. Each time a structure is detected, we create a positive sample, and we create an analogous negative sample (from the same tree) from node tuples that do not participate in that structure.



Figure 4: Illustrations for structural tasks. These figures are subtrees of the conversation tree in fig 2a. The prompts to the model only capture the conversations depicted in green.

Examples (i) Task 1 - User wedge detection: A "wedge" is an important topological construct (Albert and Barabási, 2002), and is formed when two users respond to a single comment (see Figure 4a for illustration). (ii) Task 2 - Triangle Detection between 3 users: This task builds upon wedge detection. We define a discussion between three users A, B and C as triangular if there exists an instance where, for example, both User B and User C comment on a single comment by User A. In addition to this, if User C also comments on User B's comment (or vice-versa), a triangle is formed. See Figure 4b for an illustration. (iii) Task 3 - Existence of long chain between two users: We define a long chain of to and from discussion between two users A and B when there exists an instance where User B comments on a comment by User A followed by User A commenting on User B's comment to his comment and so on. See Figure 4c for illustration.

3.3 Structural Task Performance of Native LLMs

We evaluated different LLM model families with different model sizes on our unseen structural tasks using 100 randomly sampled instances. We report these results in Table 1, which shows that none of the existing models can successfully understand and reason over topology of these conversation trees.

³We report zero-shot results in this table because multiple of these models only afford a short context length. All models are IT variants, and are hyperlinked to their corresponding repositories.

Table 1: Results of structural tasks for zero-shot inference using different model families and model sizes.³ We report accuracy numbers in this table. The results show that all these model families and model sizes exhibit headroom to improve understanding and reasoning over structure.

	-		-			
Task	Gemma 2B	Phi Mini	Mistral 7B	Gемма 9В	Gemini Pro	PaLM Bison
Same_level	13.4	34.0	39.0	52.0	52.3	53.0
Has_comment_wedge	27.5	40.0	40.0	59.0	49.2	54.9
Are_interacting	22.1	52.0	49.0	65.0	52.1	72.9
Are_fighting	20.9	42.0	56.0	55.0	57.9	59.3
Has_user_wedge	34.7	40.0	43.0	53.0	50.8	54.5

4 Experiments

We now describe experiments testing our primary hypothesis: that fine-tuning an LLM on primitive structural tasks over conversation trees enables generalization and further fine-tuning on downstream tree problems. For brevity, we refer to LLMs fine-tuned with TREETOP as LLM-TTs, and others as "native" LLMs. We show that (i) LLM-TTs can decisively solve seen structural tasks and generalize to unseen structural tasks, whereas native LLMs cannot; (ii) LLM-TTs further fine-tuned on downstream, real-world tasks outperform both GNN baselines and native LLMs fine-tuned on those same tasks; and (iii) representations learned by LLM-TTs are robust and explainable. Throughout, we present results for the TREETOP framework with LLMs fine-tuned with 10K samples per task.

4.1 Results on Primitive Structural Tasks

We present results of fine-tuning using the TREETOP framework with GEMMA, PaLM-Bison, and GEMINI models.⁴ Table 2 shows GEMINI's performance on our novel collection of 17 primitive structural tasks, described in Section 3.2, 12 of which were used to fine-tune GEMINI-TT. We find that via our fine-tuning procedure, TREETOP achieves **near-perfect** performance across these tasks, whereas the first columnar set titled GEMINI (zero-shot evaluation) shows that GEMINI lacks any inherent structural understanding. Additionally, GEMINI-TT **decisively outperforms** on unseen tasks, strongly validating the promise of our framework and task collection for generalization. We compare interactions with GEMINI and GEMINI-TT for one example each from three different structural tasks in Appendix K.

We also fine-tuned both GEMMA and PaLM-Bison using the TREETOP framework. We present these results in Appendix G. Our results show that the efficacy of the TREETOP framework can be replicated across different LLM families and sizes.

4.2 Results on Downstream Social Media Tasks

We chose the following collection of discussion tree classification tasks to test the application potential of LLM-TTs: (i) **Controversial post detection** (Hessel and Lee, 2019), or identification of "posts that split the preferences of a community, receiving both significant positive and significant negative feedback." (ii) **Rumor detection** using the PHEME9 dataset (Kochkina et al., 2018), (iii) **Fake news detection** using the Fakeddit dataset (Nakamura et al., 2019), and (iv) **Winning argument detection** (Tan et al., 2016), i.e. identification if a viewpoint of the original post author has been changed by any of the replies, collected from the r/changemyview subreddit. We provide statistics and download links for all these datasets in Appendix D.1 (Table 12) and license descriptions in Appendix L. We show that LLM-TTs fine-tuned on these tasks outperform both native LLMs (likewise fine-tuned) and non-LLM state-of-the-art, showing the promise of TREETOP toward important applications. We discuss results from tasks (i) and (iv), leaving discussions of (ii) and (iii) to Appendix E. Additionally, we also explore **early detection variants** of (i) and (ii) in Appendix E.

⁴While we showed inference results on multiple model size across multiple model families, we only chose one open-source and two closed-source models for fine-tuning due to compute limitations.

Table 2: Results of structural tasks for zero-shot and two-shot inference using GEMINI, and for zero-shot inference using GEMINI-TT. We show results with two-shot to provide one instance of both positive and negative class, and we couldn't test with more examples because of context length limitations. Standard error for all reported results is reported in Appendix H.2 (Table 21).

	Ge	MINI	(Zero	-shot)	GE	EMINI	(Two-	-shot)		Gemi	NI-TT	
Fine-tuning Tasks	Acc	Rec	Pre	F1	Acc	Rec	Pre	F1	Acc	Rec	Pre	F1
Are_one_hop_neighbors	39.0	27.3	35.5	30.9	47.9	84.5	48.8	61.9	100.0	100.0	100.0	100.0
Are_two_hop_neighbors	58.1	48.4	61.0	54.0	56.2	93.8	53.6	68.2	99.9	99.8	100.0	99.9
Are_three_hop_neighbors	45.5	27.5	42.2	33.3	40.3	58.2	42.9	49.4	100.0	100.0	100.0	100.0
Within_subtree	83.3	84.4	82.9	83.6	86.3	90.5	83.5	86.9	100.0	100.0	100.0	100.0
In_triangle	57.6	38.1	57.2	45.7	60.4	63.3	59.8	61.5	91.6	92.1	91.1	91.6
In_long_chain	59.1	3.0	74.6	5.8	58.1	43.3	61.6	50.9	96.5	98.7	94.6	96.6
Num_conversations	60.3	65.6	59.7	62.5	64.3	98.3	58.5	73.3	99.9	99.9	100.0	99.9
Num_children	53.5	32.3	53.8	40.3	55.3	97.6	52.9	68.6	99.8	100.0	99.6	99.8
Node_level	58.1	77.8	55.5	64.8	74.8	87.5	69.8	77.6	94.5	99.2	90.7	94.8
Is_leaf	57.4	37.7	56.5	45.2	68.7	80.9	65.1	72.1	99.9	99.9	100.0	99.9
Depth	55.1	16.5	73.5	26.9	61.5	93.9	57	70.9	93.7	90.7	96.6	93.5
Num_leaf_nodes	56.2	50.5	56.9	53.5	51.0	98.8	50.5	66.8	87.7	96.7	81.9	88.7
Unseen Tasks												
Same_level	52.3	6.2	76.8	11.4	47.5	28.7	46	35.3	76.1	81.1	73.6	77.1
Has_comment_wedge	49.2	10.6	47.3	17.3	53.5	83.3	52.2	64.2	63.0	78.8	59.8	68.0
Are_interacting	52.1	6.2	78.1	11.5	70.0	73.6	68.7	71.0	78.2	59.1	95.7	73.0
Are_fighting	57.9	16.1	84.2	27.0	68.4	51.5	77.8	62.0	86.2	99.9	78.4	87.9
Has_user_wedge	50.8	12.7	50.9	20.3	51.3	54.8	51.2	53.0	61.6	43.3	68.3	53.0

4.2.1 Experimental Design

We divided each dataset into a random 70:15:15 split for fine-tuning, validation, and testing. The validation set was used to select the best LLM checkpoint from the fine-tuning phase. We use 5-way cross-validation with a bootstrapping approach to derive test-set standard errors, described fully in Appendix H.2.2. During fine-tuning, each LLM receives each discussion tree input in the TREETOP encoding (see Section 3.1), along with a yes/no prompt: "Whether the post is a [X] post?", where [X] is controversial, rumor, fake news, or winning argument, depending on the data set. Across our different experiments, we compare five models: GEMINI (zero-shot and two-shot), GEMINI fine-tuned for that specific social media task, GEMINI-TT (zero-shot), and GEMINI-TT fine-tuned for that specific social media task. We also compared with 3 GNN-baselines of GCN, GAT and GraphSage where applicable. For the GNN baselines, we use the BERT model to embed the text content of posts and comments in the conversation tree. Additionally, we also provide the SOTA GNN-based benchmark for comparison.

4.2.2 Performance on Controversial Post Detection

Table 3 presents controversial post detection results across all models, including the DFE-GCN algorithm (Hua et al., 2023), the prior state-of-the-art GNN-based model that combines sentence-BERT with GCNs.⁵ GEMINI-TT fine-tuned for this task has the highest performance across all metrics, even exceeding GNN baselines and DFE-GCN. We note that this dominant performance is due both to topology-aware capacities gained from TREETOP, as well as fine-tuning on this particular task. TREETOP's effect can be seen by the marked improvement of GEMINI-TT 0-shot F1 against GEMINI 0-shot F1. Fine-tuning GEMINI-TT on this task is then much more efficient, leading to SotA.

⁵These are results as reported in Hua et al. (2023) and Tan et al. (2016), acknowledging that the inputs and test sets might not have parity. (Tan et al., 2016) only provide accuracy, and precision/recall/f1 are not reported in the paper.

Table 3: Results for controversial post detection task across all models. "ZS" means zero-shot, "2S" means two-shot, and "FT" refers to fine-tuning on the detection task. GSG refers to the GraphSage GNN baseline. All models have a standard error < 1.3 for the accuracy metric, and are reported in Appendix H.2 (Table 22).

		Gemin	I	Gemi	NI-TT	GN	N Baseli	ines	SOTA
Metric	ZS	2S	FT	ZS	FT	GCN	GAT	GSG	DFE-GCN ⁵
Acc	50.0	56.4	68.6	50.6	84.6	64.0	68.0	66.0	76.6
Rec	27.4	53.4	85.9	90.5	87.7	92.0	75.0	85.0	67.2
Pre	58.2	56.9	64.0	50.4	82.6	59.0	66.0	62.0	67.4
F1	37.3	55.1	73.3	64.7	85.0	72.0	70.0	72.0	67.3

4.2.3 Winning Argument Thread Detection

In Table 4 we give the results for the winning argument thread detection task. We compare with the approach in Tan et al. (2016), combining multiple linguistic and interaction-based features of conversation trees, comprising the most recent prior benchmark.⁵ GEMINI-TT outperforms Tan et al. and GEMINI across all metrics. Even zero-shot performance of GEMINI-TT is better than GEMINI *fine-tuned* for this specific task. We attribute this result to the enhanced topological understanding brought about by TREETOP fine-tuning.

Table 4: Results for the winning argument detection task. All reported results have a standard error < 1.4 for accuracy, and are reported in Appendix H.2 (Table 23).

	Gemini			Gemi	NI-TT	SOTA
Metric	ZS	2S	FT	ZS	FT	Tan et al. (2016) ⁵
Acc	50.2	47.3	51.5	52.5	76.6	70.0
Rec	3.2	5.6	12.3	45.2	83.5	-
Pre	51.3	60.0	57.5	53.0	73.4	-
F1	6.1	10.3	20.3	48.8	78.1	-

4.3 Explanability and Robustness of Representations Learned by LLM-TTs

To further demonstrate the robustness and repeatability of our work, we performed a variety of ablation studies and similar exercises. We perform model ablations using GEMMA and PaLM-Bison in Appendix G, data ablations in Appendix H, regression analysis in Appendix I, and analyze TREETOP embeddings in Appendix J.

5 Conclusions

In this paper, we introduced TREETOP, a framework and suite of tasks for fine-tuning LLMs to perform well on topological reasoning tasks over conversation tree inputs. As we showed through our experiments, LLMs fine-tuned with TREETOP generalize significantly better to out-of-distribution reasoning tasks, and have achieved state-of-the-art on a variety of downstream tasks in various social media domain applications. We now address limitations and future directions.

As LLMs become better-able to handle multimodality (Zhang et al., 2024), an important next step in this line of research is to allow large models to learn signals from the joint distribution of topology and non-text modes such as images and videos. Future efforts in this direction should be inspired and motivated by existing non-LLM work in multimodal cascade prediction (Xie et al., 2020; Zhang et al., 2018), and by the importance of multimedia content in such applications (Nakamura et al., 2019). More generally, we hope that the concepts underlying TREETOP– fine-tuning on structural tasks after pre-training on language tasks – may be a re-usable recipe for teaching LLMs to become performant on structured data.

References

- Abagissa, A. T., Saxena, S., and Chandra, J. (2024). Distilbert-gnn: a powerful approach to social media event detection.
- Akram, W. and Kumar, R. (2017). A study on positive and negative effects of social media on society. *International journal of computer sciences and engineering*, 5(10):351–354.
- Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. (2022). Flamingo: a visual language model for few-shot learning. Advances in neural information processing systems, 35:23716–23736.
- Albert, R. and Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97.
- Amedie, J. (2015). The impact of social media on society.
- Bail, C. A., Argyle, L. P., Brown, T. W., Bumpus, J. P., Chen, H., Hunzaker, M. F., Lee, J., Mann, M., Merhout, F., and Volfovsky, A. (2018). Exposure to opposing views on social media can increase political polarization. *Proceedings of the National Academy of Sciences*, 115(37):9216–9221.
- Bakshy, E., Rosenn, I., Marlow, C., and Adamic, L. (2012). The role of social networks in information diffusion. In Proceedings of the 21st international conference on World Wide Web, pages 519–528.
- Baumgartner, J., Zannettou, S., Keegan, B., Squire, M., and Blackburn, J. (2020). The pushshift reddit dataset. In *Proceedings of the international AAAI conference on web and social media*, volume 14, pages 830–839.
- Benslimane, S., Azé, J., Bringay, S., Servajean, M., and Mollevi, C. (2021). Controversy detection: A text and graph neural network based approach. In Web Information Systems Engineering– WISE 2021: 22nd International Conference on Web Information Systems Engineering, WISE 2021, Melbourne, VIC, Australia, October 26–29, 2021, Proceedings, Part I 22, pages 339–354. Springer.
- Bisen, S. (2017). Application of graph theory in transportation networks. *International Journal of Scientific Research and Management*, 5.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Chami, I., Abu-El-Haija, S., Perozzi, B., Ré, C., and Murphy, K. (2022). Machine learning on graphs: A model and comprehensive taxonomy. *Journal of Machine Learning Research*, 23(89):1–64.
- Chen, J., Chen, Q., Jiang, F., Guo, X., Sha, K., and Wang, Y. (2024a). Scn_gnn: A gnn-based fraud detection algorithm combining strong node and graph topology information. *Expert Systems with Applications*, 237:121643.
- Chen, Z., Mao, H., Li, H., Jin, W., Wen, H., Wei, X., Wang, S., Yin, D., Fan, W., Liu, H., et al. (2024b). Exploring the potential of large language models (llms) in learning on graphs. *ACM SIGKDD Explorations Newsletter*, 25(2):42–61.
- Chen, Z., Xiao, T., and Kuang, K. (2022). Ba-gnn: On learning bias-aware graph neural network. In 2022 IEEE 38th International Conference on Data Engineering (ICDE), pages 3012–3024. IEEE.
- Chien, E., Chang, W.-C., Hsieh, C.-J., Yu, H.-F., Zhang, J., Milenkovic, O., and Dhillon, I. S. (2021). Node feature extraction by self-supervised multi-scale neighborhood prediction. *arXiv preprint arXiv:2111.00064*.
- Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., et al. (2024). Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.
- Cloud, G. (2023). Introducing Gemini: our largest and most capable AI model. https://blog.google/technology/ai/google-gemini-ai/#sundar-note.

- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
- Dou, Y. (2022). Robust graph learning for misbehavior detection. In *Proceedings of the Fifteenth* ACM International Conference on Web Search and Data Mining, pages 1545–1546.
- Dou, Y., Shu, K., Xia, C., Yu, P. S., and Sun, L. (2021). User preference-aware fake news detection. In Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval, pages 2051–2055.
- Duan, K., Liu, Q., Chua, T.-S., Yan, S., Ooi, W. T., Xie, Q., and He, J. (2023). Simteg: A frustratingly simple approach improves textual graph learning. arXiv preprint arXiv:2308.02565.
- Efron, B. and Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*. Number 57 in Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, Boca Raton, Florida, USA.
- Fatemi, B., Halcrow, J., and Perozzi, B. (2023). Talk like a graph: Encoding graphs for large language models. arXiv preprint arXiv:2310.04560.
- Gao, W., Fang, Y., Li, L., and Tao, X. (2021). Event detection in social media via graph neural network. In *Web Information Systems Engineering–WISE 2021: 22nd International Conference on Web Information Systems Engineering, WISE 2021, Melbourne, VIC, Australia, October 26–29, 2021, Proceedings, Part I 22*, pages 370–384. Springer.
- Garimella, K., Morales, G. D. F., Gionis, A., and Mathioudakis, M. (2018). Quantifying controversy on social media. ACM Transactions on Social Computing, 1(1):1–27.
- Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017). Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR.
- Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2020). Message passing neural networks. *Machine learning meets quantum physics*, pages 199–214.
- Google and, R. A., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., Chu, E., Clark, J. H., Shafey, L. E., Huang, Y., Meier-Hellstern, K., Mishra, G., Moreira, E., Omernick, M., Robinson, K., Ruder, S., Tay, Y., Xiao, K., Xu, Y., Zhang, Y., Abrego, G. H., Ahn, J., Austin, J., Barham, P., Botha, J., Bradbury, J., Brahma, S., Brooks, K., Catasta, M., Cheng, Y., Cherry, C., Choquette-Choo, C. A., Chowdhery, A., Crepy, C., Dave, S., Dehghani, M., Dev, S., Devlin, J., Díaz, M., Du, N., Dyer, E., Feinberg, V., Feng, F., Fienber, V., Freitag, M., Garcia, X., Gehrmann, S., Gonzalez, L., Gur-Ari, G., Hand, S., Hashemi, H., Hou, L., Howland, J., Hu, A., Hui, J., Hurwitz, J., Isard, M., Ittycheriah, A., Jagielski, M., Jia, W., Kenealy, K., Krikun, M., Kudugunta, S., Lan, C., Lee, K., Lee, B., Li, E., Li, M., Li, W., Li, Y., Li, J., Lim, H., Lin, H., Liu, Z., Liu, F., Maggioni, M., Mahendru, A., Maynez, J., Misra, V., Moussalem, M., Nado, Z., Nham, J., Ni, E., Nystrom, A., Parrish, A., Pellat, M., Polacek, M., Polozov, A., Pope, R., Qiao, S., Reif, E., Richter, B., Riley, P., Ros, A. C., Roy, A., Saeta, B., Samuel, R., Shelby, R., Slone, A., Smilkov, D., So, D. R., Sohn, D., Tokumine, S., Valter, D., Vasudevan, V., Vodrahalli, K., Wang, X., Wang, P., Wang, Z., Wang, T., Wieting, J., Wu, Y., Xu, K., Xu, Y., Xue, L., Yin, P., Yu, J., Zhang, O., Zheng, S., Zheng, C., Zhou, W., Zhou, D., Petrov, S., and Wu, Y. (2023). Palm 2 technical report.
- Guo, J., Du, L., and Liu, H. (2023). Gpt4graph: Can large language models understand graph structured data? an empirical evaluation and benchmarking. *arXiv preprint arXiv:2305.15066*.
- Han, Y., Karunasekera, S., and Leckie, C. (2020). Graph neural networks with continual learning for fake news detection from social media. *arXiv preprint arXiv:2007.03316*.
- Hardy, M. D., Thompson, B. D., Krafft, P., and Griffiths, T. L. (2023). Resampling reduces bias amplification in experimental social networks. *Nature Human Behaviour*, 7(12):2084–2098.
- He, X., Bresson, X., Laurent, T., Perold, A., LeCun, Y., and Hooi, B. (2023). Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning. In *The Twelfth International Conference on Learning Representations*.

- Hessel, J. and Lee, L. (2019). Something's brewing! early prediction of controversy-causing posts from discussion features. *arXiv preprint arXiv:1904.07372*.
- Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. (2020a). Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133.
- Hu, Z., Dong, Y., Wang, K., Chang, K.-W., and Sun, Y. (2020b). Gpt-gnn: Generative pre-training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on* knowledge discovery & data mining, pages 1857–1867.
- Hua, C., Yang, W., Wang, L., Wei, F., HaiLaTi, K., and Liao, Y. (2023). Dfe-gcn: Dual feature enhanced graph convolutional network for controversy detection. *Computers, Materials & Continua*, 77(1).
- Huang, J., Zhang, X., Mei, Q., and Ma, J. (2023). Can llms effectively leverage graph structural information: when and why. *arXiv preprint arXiv:2309.16595*.
- Huang, S., Dong, L., Wang, W., Hao, Y., Singhal, S., Ma, S., Lv, T., Cui, L., Mohammed, O. K., Patra, B., et al. (2024). Language is not all you need: Aligning perception with language models. *Advances in Neural Information Processing Systems*, 36.
- Ji, Y., Wang, J., Li, S., Li, Y., Lin, S., and Li, X. (2021). An anomaly event detection method based on gnn algorithm for multi-data sources. In *Proceedings of the 3rd ACM International Symposium* on Blockchain and Secure Critical Infrastructure, pages 91–96.
- Kaundal, K. (2017). Applications of graph theory in everyday life and technology. *Imperial journal of interdisciplinary research*, 3.
- Kochkina, E., Liakata, M., and Zubiaga, A. (2018). All-in-one: Multi-task learning for rumour verification. *arXiv preprint arXiv:1806.03713*.
- Lerman, K. and Ghosh, R. (2010). Information contagion: An empirical study of the spread of news on digg and twitter social networks. In *Proceedings of the International AAAI Conference on Web* and Social Media, volume 4, pages 90–97.
- Li, H., Wang, X., Zhang, Z., and Zhu, W. (2022). Ood-gnn: Out-of-distribution generalized graph neural network. *IEEE Transactions on Knowledge and Data Engineering*.
- Li, Y. and Li, N. (2022). Sentiment analysis of weibo comments based on graph neural network. *IEEE Access*, 10:23497–23510.
- Lillie, A. E. and Middelboe, E. R. (2019). Fake news detection using stance classification: A survey. *arXiv preprint arXiv:1907.00181*.
- Liu, X., Miao, C., Fiumara, G., and De Meo, P. (2023a). Information propagation prediction based on spatial-temporal attention and heterogeneous graph convolutional networks. *IEEE Transactions on Computational Social Systems*.
- Liu, Y., Sun, Z., and Zhang, W. (2023b). Improving fraud detection via hierarchical attention-based graph neural network. *Journal of Information Security and Applications*, 72:103399.
- Ma, J. and Gao, W. (2020). Debunking rumors on twitter with tree transformer. ACL.
- Madhu, H., Satapara, S., Modha, S., Mandl, T., and Majumder, P. (2023). Detecting offensive speech in conversational code-mixed dialogue on social media: A contextual dataset and benchmark experiments. *Expert Systems with Applications*, 215:119342.
- Mukiri, R. K. and Burra, V. B. (2023). An efficient cnn-based lstm approach to detect the rumor prediction in covid-19 data using fuzzy based deep learning.
- Müller, L., Galkin, M., Morris, C., and Rampášek, L. (2023). Attending to graph transformers. *arXiv* preprint arXiv:2302.04181.

- Nakamura, K., Levy, S., and Wang, W. Y. (2019). r/fakeddit: A new multimodal benchmark dataset for fine-grained fake news detection. arXiv preprint arXiv:1911.03854.
- Nguyen, T. T., Ren, Z., Nguyen, T. T., Jo, J., Nguyen, Q. V. H., and Yin, H. (2024). Portable graphbased rumour detection against multi-modal heterophily. *Knowledge-Based Systems*, 284:111310.
- Olan, F., Jayawickrama, U., Arakpogun, E. O., Suklan, J., and Liu, S. (2024). Fake news on social media: the impact on society. *Information Systems Frontiers*, 26(2):443–458.
- Olteanu, A., Castillo, C., Diaz, F., and Kıcıman, E. (2019). Social data: Biases, methodological pitfalls, and ethical boundaries. *Frontiers in big data*, 2:13.
- Pan, S., Zheng, Y., and Liu, Y. (2024). Integrating graphs with large language models: Methods and prospects. *IEEE Intelligent Systems*, 39(1):64–68.
- Paranjape, A., Benson, A. R., and Leskovec, J. (2017). Motifs in temporal networks. In Proceedings of the tenth ACM international conference on web search and data mining, pages 601–610.
- Patel, S., Bansal, P., and Kaur, P. (2022). Rumour detection using graph neural network and oversampling in benchmark twitter dataset. *arXiv preprint arXiv:2212.10080*.
- Perozzi, B., Fatemi, B., Zelle, D., Tsitsulin, A., Kazemi, M., Al-Rfou, R., and Halcrow, J. (2024). Let your graph do the talking: Encoding structured data for llms. arXiv preprint arXiv:2402.05862.
- Priyadarsini, P. (2015). A survey on some applications of graph theory in cryptography. *Journal of Discrete Mathematical Sciences and Cryptography*, 18:209–217.
- Pröllochs, N. and Feuerriegel, S. (2023). Mechanisms of true and false rumor sharing in social media: collective intelligence or herd behavior? *Proceedings of the ACM on Human-Computer Interaction*, 7(CSCW2):1–38.
- Qin, Y., Wang, X., Zhang, Z., and Zhu, W. (2023). Disentangled representation learning with large language models for text-attributed graphs. arXiv preprint arXiv:2310.18152.
- Qiu, J., Lin, Z., and Shuai, Q. (2019). Investigating the opinions distribution in the controversy on social media. *Information Sciences*, 489:274–288.
- Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.
- Reid, M., Savinov, N., Teplyashin, D., Lepikhin, D., Lillicrap, T., Alayrac, J.-b., Soricut, R., Lazaridou, A., Firat, O., Schrittwieser, J., et al. (2024). Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. arXiv preprint arXiv:2403.05530.
- Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2008). The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80.
- Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., and Welling, M. (2018). Modeling relational data with graph convolutional networks. In *The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15*, pages 593–607. Springer.
- Shu, K., Sliva, A., Wang, S., Tang, J., and Liu, H. (2017). Fake news detection on social media: A data mining perspective. ACM SIGKDD explorations newsletter, 19(1):22–36.
- Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., et al. (2022). Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. arXiv preprint arXiv:2206.04615.

- Sui, Y., Zhou, M., Zhou, M., Han, S., and Zhang, D. (2024). Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. In *Proceedings of the* 17th ACM International Conference on Web Search and Data Mining, pages 645–654.
- Tan, C., Niculae, V., Danescu-Niculescu-Mizil, C., and Lee, L. (2016). Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *Proceedings of the* 25th international conference on world wide web, pages 613–624.
- Tang, J., Yang, Y., Wei, W., Shi, L., Su, L., Cheng, S., Yin, D., and Huang, C. (2023). Graphgpt: Graph instruction tuning for large language models. *arXiv preprint arXiv:2310.13023*.
- Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J.-B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., et al. (2023). Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805.
- Tian, L., Zhang, X., Wang, Y., and Liu, H. (2020). Early detection of rumours on twitter via stance transfer learning. In Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part I 42, pages 575–588. Springer.
- Toshevska, M., Kalajdziski, S., and Gievska, S. (2023). Graph neural networks for antisocial behavior detection on twitter. In *International Conference on ICT Innovations*, pages 222–236. Springer.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. (2021). Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.
- Wen, Z. (2023). Generalizing graph neural network across graphs and time. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 1214–1215.
- Wu, J., Zhang, C., Liu, Z., Zhang, E., Wilson, S., and Zhang, C. (2022). Graphbert: Bridging graph and text for malicious behavior detection on social media. In 2022 IEEE International Conference on Data Mining (ICDM), pages 548–557. IEEE.
- Wu, Q., Chen, Z., Corcoran, W., Sra, M., and Singh, A. K. (2024). Grapheval2000: Benchmarking and improving large language models on graph datasets. *arXiv preprint arXiv:2406.16176*.
- Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24.
- Xie, J., Zhu, Y., Zhang, Z., Peng, J., Yi, J., Hu, Y., Liu, H., and Chen, Z. (2020). A multimodal variational encoder-decoder framework for micro-video popularity prediction. In *Proceedings of the web conference 2020*, pages 2542–2548.
- Xu, K., Hu, W., Leskovec, J., and Jegelka, S. (2018). How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*.
- Xu, S., Liu, X., Ma, K., Dong, F., Riskhan, B., Xiang, S., and Bing, C. (2023). Rumor detection on social media using hierarchically aggregated feature via graph neural networks. *Applied Intelligence*, 53(3):3136–3149.
- Yan, H., Li, C., Long, R., Yan, C., Zhao, J., Zhuang, W., Yin, J., Zhang, P., Han, W., Sun, H., et al. (2023). A comprehensive study on text-attributed graphs: Benchmarking and rethinking. *Advances in Neural Information Processing Systems*, 36:17238–17264.
- Yasunaga, M., Bosselut, A., Ren, H., Zhang, X., Manning, C. D., Liang, P. S., and Leskovec, J. (2022a). Deep bidirectional language-knowledge graph pretraining. *Advances in Neural Information Processing Systems*, 35:37309–37323.
- Yasunaga, M., Leskovec, J., and Liang, P. (2022b). Linkbert: Pretraining language models with document links. arXiv preprint arXiv:2203.15827.

- Ye, R., Zhang, C., Wang, R., Xu, S., and Zhang, Y. (2024). Language is all a graph needs. In *Findings* of the Association for Computational Linguistics: EACL.
- Zeng, Y. and Tang, J. (2021). Rlc-gnn: An improved deep architecture for spatial-based graph neural network with application to fraud detection. *Applied Sciences*, 11(12):5656.
- Zhang, D., Yu, Y., Li, C., Dong, J., Su, D., Chu, C., and Yu, D. (2024). Mm-llms: Recent advances in multimodal large language models. arXiv preprint arXiv:2401.13601.
- Zhang, J. (2023). Graph-toolformer: To empower llms with graph reasoning ability via prompt augmented by chatgpt. *arXiv preprint arXiv:2304.11116*.
- Zhang, K. and Shasha, D. (1989). Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing*, 18(6):1245–1262.
- Zhang, W., Wang, W., Wang, J., and Zha, H. (2018). User-guided hierarchical attention network for multi-modal social image popularity prediction. In *Proceedings of the 2018 world wide web conference*, pages 1277–1286.
- Zhang, Z., Cui, P., and Zhu, W. (2020). Deep learning on graphs: A survey. IEEE Transactions on Knowledge and Data Engineering, 34(1):249–270.
- Zhang, Z., Wang, X., Zhang, Z., Li, H., Qin, Y., Wu, S., and Zhu, W. (2023). Llm4dyg: Can large language models solve problems on dynamic graphs? *arXiv preprint arXiv:2310.17110*.
- Zhao, J., Zhuo, L., Shen, Y., Qu, M., Liu, K., Bronstein, M., Zhu, Z., and Tang, J. (2023a). Graphtext: Graph reasoning in text space. *arXiv preprint arXiv:2310.01089*.
- Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al. (2023b). A survey of large language models. *arXiv preprint arXiv:2303.18223*.
- Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI open*, 1:57–81.
- Zhou, K., Shu, C., Li, B., and Lau, J. H. (2019). Early rumour detection. 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), pages 1614–1623.
- Zhu, J., Cui, Y., Liu, Y., Sun, H., Li, X., Pelger, M., Yang, T., Zhang, L., Zhang, R., and Zhao, H. (2021). Textgnn: Improving text encoder via graph neural network in sponsored search. In *Proceedings of the Web Conference 2021*, pages 2848–2857.
- Zhu, Y., Sheng, Q., Cao, J., Li, S., Wang, D., and Zhuang, F. (2022). Generalizing to the future: Mitigating entity bias in fake news detection. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2120–2125.
- Zubiaga, A., Liakata, M., Procter, R., Wong Sak Hoi, G., and Tolmie, P. (2016). Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PloS* one, 11(3):e0150989.

A Supplementary: Table of Contents

We first present the Table of Contents for the supplementary material in this Appendix.

1at	be 5: List of supplementary content in this Appendix.
Section	Name
Appendix B	Related Work
Appendix C	Methodology
Appendix C.1	Structural Tasks Description
Appendix C.2	Prompt Encoding
Appendix C.3	Instruction Prompts for Structural Tasks
Appendix D	Dataset Statistics
Appendix D.1	Social Media Dataset Statistics
Appendix D.2	Early Detection in Rumor and Controversy Datasets
Appendix E	Supplementary Downstream Task Experiments
Appendix E.1	Fakeddit
Appendix E.2	Rumor Detection
Appendix E.3	Early Detection of Controversial Posts
Appendix E.4	Early Detection of Rumors
Appendix F	Model Hyper-parameters and Hardware Details
Appendix G	Model Ablations
Appendix G.1	Results with GEMMA
Appendix G.1.1	GEMMA Results on Structural Fine-tuning
Appendix G.1.2	GEMMA Results on Downstream Social Media Tasks
Appendix G.2	Results with PaLM-Bison
Appendix H	Data Ablations
Appendix H.1	Increasing the Size of Structural Task Dataset
Appendix H.1.1	Results on Structural Tasks
Appendix H.1.2	Results on Downstream Social Media Tasks
Appendix H.2	Bootstrapping Analysis of Test Set
Appendix H.2.1	Bootstrapping Tests for Structural Fine-tuning
Appendix H.2.2	Bootstrapping Tests for Downstream Social Media Tasks
Appendix H.3	TREETOP Framework Fine-tuning Corpus Ablations
Appendix H.4	K-Fold Cross-validation for Downstream Social Media Tasks
Appendix I	Regression Analysis
Appendix J	Embeddings Analysis
Appendix J.1	Experimental Setup
Appendix J.2	Multi-class Topological Classifier using PaLM-TT's Embeddings
Appendix J.3	Clustering Metrics from the Embedding
Appendix K	Prompt Interactions with GEMINI and GEMINI-TT
Appendix L	Licenses and Copyrights Across Assets

Table 5: List of supplementary content in this Appendix.

B Supplementary: Related Work

One of the most seminal advances that has spurred the current wave of LLM research has been the attention mechanism, and the transformer architecture (Vaswani et al., 2017). This led to development of newer language encodings, notably BERT (Devlin et al., 2018). Radford et al. (2018) popularized generative pre-training followed by discriminative fine-tuning for downstream tasks - an approach that is highly leveraged in the application of LLMs. Raffel et al. (2020) introduced the T5 paradigm, a unified framework to convert all text-based language problems into a text-to-text format. This work also inspired modeling graph problems into text prompts. Wei et al. (2021) provided an instruction-tuning framework to fine-tune LLMs for improved for zero-shot performance, whereas Brown et al. (2020) showed an improvement in LLM performance under few-shot training. Given this success of LLMs on language understanding, they have been subsequently been applied to other modalities as well, notably images (Alayrac et al., 2022), and now to the creation of multimodal models (Team et al., 2023; Huang et al., 2024; Reid et al., 2024).

Similarly, graph neural networks (Scarselli et al., 2008) were introduced for machine learning on graphs, and their expressive power has been extensively studied theoretically (Xu et al., 2018) and across multiple benchmarks (Chien et al., 2021). Extensive surveys (Wu et al., 2020; Zhou et al., 2020; Chami et al., 2022) on GNNs demonstrate that they have been a prominent approach to reason over graphs. Graphs have been successfully applied to a variety of problem domains, for instance, cryptography (Priyadarsini, 2015), transportation networks (Bisen, 2017), quantum physics (Gilmer et al., 2020), chemistry (Gilmer et al., 2017; Hu et al., 2020b), and in multiple domains of everyday use and technology (Kaundal, 2017). For example, knowledge graphs have seen widespread use and adoption in retrieval and query-based systems (Schlichtkrull et al., 2018; Yasunaga et al., 2022a).

Multiple recent studies have leveraged cross-pollination of ideas between GNNs and LLMs. Zhang et al. (2023) look at solving spatio-temporal problems on dynamic graphs using large language models. Zhang (2023) notes that using tools can help an LLM reason over graph structures. On the other hand, Yasunaga et al. (2022b) view a text corpus as a graph of documents, and propose LinkBERT, an LM pre-training method that leverages links between documents. Similarly, Zhu et al. (2021) and Qin et al. (2023) propose methods that use GNNs to improve the performance of a text encoder. Alternatively, Duan et al. (2023) use LLM embeddings to initialize the GNN node features, and along with Perozzi et al. (2024), contribute to the area of work which integrates LLMs and GNNs at the level of tokens and embeddings. In a parallel line of work, language models have been used to improve GNN performance. Xu et al. (2023) use a BERT model to encode textual features on nodes whereas He et al. (2023) leverage explanations from an LLM to improve performance of downstream GNNs.

Similarly, in the domain of social media analysis, multiple studies have been performed that discuss the impacts of consuming social media information on the individual and the population (Amedie, 2015; Bail et al., 2018; Akram and Kumar, 2017; Olan et al., 2024). Conversation trees are central to how information is consumed in social media – there is a central post that an original poster makes, and there are comments in response to this post. Information flow on social media has motivated several lines of work (Lerman and Ghosh, 2010; Bakshy et al., 2012; Liu et al., 2023a; Pröllochs and Feuerriegel, 2023; Hardy et al., 2023). Controversial post detection is of central importance to ensure that social media users are not exposed to potentially harmful content (Benslimane et al., 2021; Madhu et al., 2023; Garimella et al., 2018; Qiu et al., 2019). Similarly, a growing body of research is dedicated to addressing the issue of fake news, with studies exploring various detection methods and their effectiveness (Han et al., 2020; Shu et al., 2017; Nakamura et al., 2019; Lillie and Middelboe, 2019; Ma and Gao, 2020; Dou et al., 2021). Multilingual generalization (Li and Li, 2022; Wen, 2023; Li et al., 2022), bias detection (Olteanu et al., 2019; Chen et al., 2022; Zhu et al., 2022), fraud detection (Liu et al., 2023b; Zeng and Tang, 2021; Chen et al., 2024a), event detection (Abagissa et al., 2024; Gao et al., 2021; Ji et al., 2021) and malicious behaviour detection (Wu et al., 2022; Toshevska et al., 2023; Dou, 2022) are other important problems in this area.

C Supplementary: Methodology

Figure 5 represents the pipeline for the creation of models using the TREETOP framework. We first begin here by providing the detailed listing of all the structural tasks, and follow that up with the prompts used for each task.



Figure 5: TREETOP's Process Flow Diagram. The blue boxes represent the core TREETOP framework. SM stands for Social Media, and is generalizable to all conversation tree tasks.

C.1 Structural Tasks Description

We provide the descriptions of all the 17 tasks provided in the TREETOP framework in Table 6 and Table 7. The four classes of tasks are described in Section 3.2.

Table 6: Structural tasks and their explanation. The set of tasks are clustered into 4 categories with different semantic application to conversation trees. This table shows the tasks TREETOP is trained on.

Tasks for fine-tuning				
	$\underline{\textbf{Comment} \times \textbf{comment tasks}}$			
Are_one_hop_neighbors	Are two comments one-hop neighbors of each other			
Are_two_hop_neighbors	Are two comments two-hop neighbors of each other			
Are_three_hop_neighbors	Are two comments three-hop neighbors of each other			
Within_subtree	Is a comment in the subtree rooted at another given comment			
$\underline{\textbf{User}} \times \textbf{user} \ \textbf{tasks}$				
In_triangle	Does there exist a triangular discussion between three users			
In_long_chain	Does there exist a to-and-fro conversation between two users			
Num_conversations	Given a user, are more than k comments by the user in the tree			
	Node characteristics tasks			
Num_children	Given a comment, are more than k replies to it			
Node_level	Given a node, detect its level			
Is_leaf	Is a given node a leaf node i.e. it has zero replies			
Tree characteristics tasks				
Depth	Given a post-comment tree, detect its depth			
Num_leaf_nodes	Are there more than k nodes in the tree with zero replies			

Table 7: This table is similar to Tab	le 6, except it lists the set	of unseen tasks that are	used to assess
TREETOP's structural understandin	3.		

Unseen tasks for evalu	ation		
	$\underline{Comment \times comment \ tasks}$		
Same_level	Given a comment pair, whether they are at same level		
Has_comment_wedge	Whether the given user pair reply to a same comment		
User $ imes$ user tasks			
Are_interacting	Given a user pair (U_1, U_2) , whether U_1 replies to U_2 's comment		
Are_fighting	Given a user pair (U_1, U_2) , whether U_1 comments on U_2 more than 2 times		
Has_user_wedge	Given three users (U_1, U_2, U_3) , whether U_1 replies to both U_2 and U_3 .		
Are_fighting Has_user_wedge	Given a user pair (U_1, U_2) , whether U_1 comments on U_2 more than 2 times Given three users (U_1, U_2, U_3) , whether U_1 replies to both U_2 and U_3 .		

C.2 Prompt Encoding

The TREETOP framework presents an encoding which captures both the topology and content of conversation trees. This encoding captures the content of both the main post and comments along with the structure of the comments tree for that post. Here, each element in the encoding of the conversation tree has all the richness that we can provide from the dataset.

The complete prompt employed by the TREETOP framework is:

Given is a social media conversation tree, where each comment (node) in the tree is of the following structure: (<node_id> <user_id> <content> <parent_id>). The first node (<CO>) is the main post on Reddit followed by the comments to the main post. <parent_id> refers to the comment/post to which the current comment is a reply to.

{Explanation}

QUESTION: {question}

OPTIONS:

- Yes

- No

POST-COMMENT TREE: $\langle encoding(comment_0) \rangle \langle encoding(comment_1) \rangle \langle encoding(comment_2) \rangle \dots$

Here, {question} is the task posed as a Yes|No question. If some additional explanation of the question is required, it is added in the {Explanation} section. For example, we first define what a triangular discussion is in a conversation tree and then ask the question about it. Detailed prompts are provide in the next section.

C.3 Instruction Prompts for Structural Tasks

We now provide the specific details for the prompts used for all our tasks in four tables - Table 8 to Table 11 for all the four categories of structural tasks.

Table 8: This table specifies all the prompts used for all the comment \times comment tasks.Prompts for comment \times comment tasks

r tompts for comment × comment tasks				
<pre>Are_one_hop_neighbors 1. whether <c{node2}> is a direct comment to <c{node1}> ? 2. whether <c{node2}> is a one-hop neighbor to <c{node1}> ?</c{node1}></c{node2}></c{node1}></c{node2}></pre>				
<pre>Are_two_hop_neighbors 1. whether <c{node2}> is a comment to one of the direct comment to <c{node1}> ? 2. whether <c{node2}> is a two-hop neighbor to <c{node1}> ?</c{node1}></c{node2}></c{node1}></c{node2}></pre>				
Are_three_hop_neighbors 1. whether <c{node2}> is a three-hop neighbor to <c{node1}> ?</c{node1}></c{node2}>				
Within_subtree 1. whether <c{node2}> is in the subtree rooted at <c{node1}></c{node1}></c{node2}>				
Same_level 1. whether <c{node1}> and <c{node2}> are at the same level ?</c{node2}></c{node1}>				
<pre>Has_comment_wedge 1. whether users <u{user1}> and <u{user2}> reply to a same comment ? 2. whether users <u{user1}> and <u{user2}> reply to <c{node}> ?</c{node}></u{user2}></u{user1}></u{user2}></u{user1}></pre>				

Table 9: This table specifies all the prompts used for all the user \times user interaction tasks.

```
Prompts for user × user tasks
```

In triangle 1. whether users <U{user1}> <U{user2}> and <U{user3}> are involved in a triangular discussion between each other? whether there is a triangular discussion between three users ? 2 Explanation: We define a discussion between three users A, B and C as triangular if there exists an instance where lets say User B comments on a comment by User A and User C also comments on the same comment by User A and one of User B or C comments on each other's comment on User A's comment. In_long_chain 1. whether users <U{user1}> and <U{user2}> are involved in a long chain of to and fro discussion of atleast length {chain_length} between each other ? 2. whether there is a long chain of to and fro discussion of at least length {chain_length} between two users ? Explanation: We define a long chain of to and fro discussion between two users A and B when there exists an instance where User B comments on a comment by User A followed by User A commenting on User B's comment to his comment and so on. Num conversations 1. whether there are more than {num_comments} comments by <U{user}> ? whether there are multiple comments by <U{user}> ? 2. Are_interacting 1. whether users <U{user1}> and <U{user2}> interact with each other i.e. one of them replies to other's comment ? Are fighting 1. whether user <U{user1}> replies to user <U{user2}> more than two times ? Has_user_wedge whether there is a user that replies to both user <U{user1}> and <U{user2}> 1.

2. whether user <U{user}> replies to both user <U{user1}> and <U{user2}> ?

 Table 10: This table specifies all the prompts used for all the node characteristics tasks.

 Prompts for node characteristics tasks

Num_children
1. whether there are more than {num_comments} direct comments to <C{node}> ?
2. whether there are more than {num_comments} children of <C{node}> ?
3. whether there are more than {num_comments} one-hop neighbors of <C{node}> ?
Node_level
1. whether <C{node}> is at level {level} ? (Assuming the root node is at level 0)
Is_leaf
1. whether <C{node}> is a leaf node ?
2. whether <C{node}> has zero children ?
3. whether <C{node}> has no replies ?

Table 11: This table specifies all the prompts used for all the tree characteristics tasks.

Prompts for tree characteristics tasks

Dep 1.	th whether the depth of the tree is {depth} ?
2.	whether the depth of the tree is more than {depth} ?
Num	_leaf_nodes
1.	whether there are more than {num_leaf_nodes} leaf nodes in the given tree ?
2.	whether there are more than {num_leaf_nodes} nodes in the given tree that
	have zero replies ?
3.	whether there are more than {num_leaf_nodes} nodes in the given tree that
	have zero children ?
4.	whether there are {num_leaf_nodes} leaf nodes in the given tree ?
5.	whether there are {num_leaf_nodes} nodes in the given tree that have zero
	children ?
6.	whether there are {num_leaf_nodes} nodes in the given tree that have zero
	replies ?

D Dataset Statistics

D.1 Social Media Dataset Statistics

We used popular social media datasets for the evaluation of downstream tasks. The sources and statistics for the social media datasets are presented in Table 12. The winning argument thread dataset collected data from the r/changemyview subreddit. Note that all the datasets are not evenly balanced between positive and negative samples, and we do not employ any under-sampling or over-sampling techniques to artificially balance the dataset. Similar to the creation of TREETOP, we used a 70 : 15 : 15 split ratio to create training, validation and test datasets for each downstream social media task. The license descriptions for all the datasets are given in Appendix L.

Dataset (Download Links)	Paper	Source	+ves	-ves
Controversial Post [link]	Hessel and Lee (2019)	Reddit	7515	7518
PHEME9 (rumor detection) [link]	Kochkina et al. (2018)	Х	1616	3058
Fakeddit [post] [comments]	Nakamura et al. (2019)	Reddit	75215	172371
Winning Argument Thread [link]	Tan et al. (2016)	Reddit	6557	6557

Table 12: Datasets and their statistics.

D.2 Early Detection in Rumor and Controversy Datasets

The following table shows how much of the conversation tree is available at different timestamps to facilitate our early detection task variation. This table shows what fraction of comments are observed within the first hour, first two hours, and so on (assuming inf time is 100%).

E Supplementary Downstream Task Experiments

For **controversial post detection**, we use the task definition and the dataset provided in Hessel and Lee (2019). The task is to detect Reddit "posts that split the preferences of a community, receiving both significant positive and significant negative feedback." This dataset only has posts which have at least 30 comments in the conversation tree. Similarly, for **rumor detection**, we use the PHEME9 dataset from Kochkina et al. (2018) for this task. In this work, the authors have collected, identified and annotated X posts as rumors associated with newsworthy events. Additionally, for **fake news detection**, we use the Fakeddit dataset from Nakamura et al. (2019) for this task. In this work, the authors have collected and annotated Reddit posts as being fake news or not. The results of controversial post detection have already been provided in Section 4.2.2; we now provide results for rumor detection in Appendix E.2 and fake news detection in Appendix E.1.

Controversy	Detection Dataset	Rumor Detection PHEME9 Dataset			
Time Duration	Percentage of comments received	Time Duration	Percentage of comments received		
0	0.00	0	0.00		
1 hr	12.13	10 mins	34.00		
2 hrs	21.29	20 mins	48.18		
4 hrs	38.23	30 mins	55.74		
6 hrs	51.92	45 mins	61.33		
12 hrs	76.63	1 hr	68.16		
24 hrs	93.18	6 hrs	89.66		
inf	100.00	inf	100.00		

Table 13: This table shows the percentage of comments posted within the time elapsed since the main post.

E.1 Fakeddit

Table 14 shows the results of fake news detection on the Fakeddit dataset (Nakamura et al., 2019). Fakeddit is a multi-modal Reddit dataset and contains both text and images. For our experiments, we only use those samples that have at least 5 comments in the conversation tree while performing fine-tuning. For our algorithms, we ignore any image content in the posts and comments. The statistics for this dataset are in Table 12, and show approximately a 1:2 class imbalance between positive and negative samples. As before, we compare the performance of GEMINI (zero-shot), GEMINI fine-tuned on Fakeddit, GEMINI-TT zero-shot, and GEMINI-TT further fine-tuned on Fakeddit. In addition, we also show results for the P and PC encodings for GEMINI. We also compare our approaches with Nakamura et al. (2019), who combine BERT-encodings with a ResNet-50 image encoding model. Our results show that fine-tuned GEMINI-TT (accuracy of 96.0%) outperforms both fine-tuned GEMINI (accuracy of 89.8%) and the approach by Nakamura et al. (2019) (accuracy of 89.1%).⁶ The standard error for our experiments is reported in Appendix H.2.

Table 14: Results for Fake News detection task. The results in bold shows that fine-tuned GEMINI-TT beats the performance of both fine-tuned GEMINI and the approach by Nakamura et al. (2019). Also, we do not have zero-shot and fine-tuned variants for Nakamura et al. (2019). Standard error for this evaluation is provided in Appendix H.2.2 (Table 24) and are below 0.2 for accuracy for all models.

		Gemin	I	Gemi	NI-TT		GNN B	aselines	SOTA
Metric	ZS	2 S	FT	ZS	FT	GCN	GAT	GraphSAGE	Nakamura et al. (2019) ⁶
Acc	76.3	63.6	89.8	77.2	96.0	77.0	79.0	81.0	89.1
Rec	3.9	52.9	82.7	2.8	88.6	71.0	78.0	77.0	-
Pre	34.9	67.3	75.1	56.8	93.5	55.0	57.0	61.0	-
F1	7.1	59.3	93.5	5.3	91.0	62.0	66.0	68.0	-

E.2 Rumor Detection

Table 15 shows the performance on rumor detection PHEME9 dataset (Kochkina et al., 2018). The statistics for this dataset are in Table 12, and show approximately a 1:2 class imbalance between positive and negative samples. For our experiments, we only use those samples that have at least 5 comments in the conversation tree while performing fine-tuning. As before, we compare the performance of GEMINI (zero-shot, and two-shot), GEMINI fine-tuned for rumor detection, GEM-INI-TT (zero shot) and GEMINI-TT fine-tuned for rumor detection. We also compare these with GNN baselines. Additionally, we also compare our models with two prior published research: (i) PHAROS algorithm (Nguyen et al., 2024), which integrates label information with graph homophily

⁶These are results as reported in Nakamura et al. (2019), acknowledging that their and our test sets might not have parity. We only report accuracy since it is the only metric reported in the paper.

measures, and is among the state-of-the-art published result in this domain⁷; and (ii) NRA MOS-GAT algorithm (Patel et al., 2022), which uses oversampling and BERT embeddings along with an attention-based GNN model.⁷ Our results show that fine-tuned GEMINI-TT model (accuracy of 87.1%) outperforms all three of fine-tuned GEMINI (accuracy of 72.5%), PHAROS (accuracy of 75.9%) and NRA MOS-GAT (accuracy of 78.4%). The standard error for all the results is below 4%, except for the precision number for GEMINI-TT-FT which had an error of around 15% (computation methodology is described in Appendix H.2).

Table 15: Results for Rumor Detection task. The results in bold show that fine-tuned GEMINI-TT beats the performance of fine-tuned GEMINI, PHAROS, and NRA MOS-GAT. Given PHAROS and NRA MOS-GAT are not LLM based, we do not have zero-shot and fine-tuned variants for them. Standard error for this evaluation is provided in Appendix H.2.2 (Table 25) and are below 1.9 for accuracy for all models.

		Gemin	I	Gemi	NI-TT		GNN B	aselines		SOTA
Metric	ZS	2 S	FT	ZS	FT	GCN	GAT	GraphSAGE	PHAROS ⁷	NRA MOS-GAT ⁷
Acc	64.0	54.8	72.5	65.9	87.1	78.0	80.0	79.0	75.9	78.4
Rec	6.1	76.2	47.1	3.1	78.1	80.0	71.0	73.0	-	-
Pre	41.3	41.9	63.7	76.5	83.6	65.0	71.0	69.0	-	-
F1	10.6	54.1	54.1	5.8	80.7	72.0	71.0	71.0	77.9	73.1

We acknowledge that the zero-shot performance (recall and f1) for both GEMINI and GEMINI-TT is sub-par. It is quite likely that these models considers the tree topology content of the TREETOP encoding as noise for this task. This may also be because our prompt is simply asking these models "whether the post is a rumor?", only relying on their innate language understanding of the word 'rumor' independent of any other context. In contrast, the performance of the fine-tuned variants of both GEMINI and GEMINI-TT see a major boost once they are able to learn our definition of 'rumor' from the fine-tuning dataset.

E.3 Early Detection of Controversial Posts

Early detection of future conversation properties is a significantly important challenge with real-world impact (Akram and Kumar, 2017), given the ubiquity of social media platforms. In Figure 6, we report results of zero-shot early detection of controversial posts for both GEMINI and GEMINI-TT *fine-tuned* for the controversy detection task. For each dataset, we take views of each conversation tree at different timestamps, where a "view" at timestamp t contains the original post and all replies up to time t. We use t = 0, 1, 2, 4, 6, 12, 24 and inf hours in our experiments. Appendix D.2 shows the statistics on what fraction of comments are seen at different time durations.





The fine-tuned GEMINI-TT achieves the same performance of fine-tuned GEMINI 20 hrs in advance. Specifically, GEMINI's accuracy at 24 hrs is 68.6% - GEMINI-TT has an accuracy of 62.6% and 68.9% at 2hrs and 4hrs respectively. Additionally, GEMINI-TT achieves 73.7% accuracy at 6hrs, and

⁷These are results as reported in PHAROS (Nguyen et al., 2024) and NRA MOS-GAT algorithm (Patel et al., 2022), acknowledging that their and our test sets might not have parity. We only report accuracy and f1 since those are the only reported metrics in the papers; precision and recall have not been reported.

83.3% accuracy at 24hrs. These results show that these models can be effective at early detection of controversial post in just 4 to 6 hours.

We provide the same results on rumor detection in Appendix E.4, we see similar dominance of fine-tuned GEMINI-TT over fine-tuned GEMINI there as well.⁸

E.4 Early Detection of Rumors

Similar to the early detection of controversial posts described in Section E.3, we collect and evaluate fine-tuned GEMINI and fine-tuned GEMINI-TT at t = 0, 10 min, 20 min, 30 min, 45 min, 1 hr, 6 hr and inf hours after the original post respectively for the rumor detection task. Our result shows that fine-tuned GEMINI-TT (accuracy of 87.0% at t = 0) outperforms fine-tuned GEMINI (accuracy of 73.7% at t = 24hrs). This shows that models fine-tuned with the TREETOP framework can also be used for extremely early detection of rumors. We also see that for all models, the performance stays quite flat over the different timespans after t = 20 mins. We hypothesize that this is because in the PHEME9 dataset, most of the comments appear in the first hour itself (see Table 13), and that this dataset is such where the main post itself is highly discriminative of rumors.



Figure 7: These graphs show the results comparing fine-tuned GEMINI and fine-tuned GEMINI-TT for early detection of rumor post tasks. The error bars are also shown in the column charts, and all error bars are within 1.9 (for accuracy).

F Model Hyper-parameters and Hardware Details

The base LLM used in most of our experiments was GEMINI (Team et al., 2023). We use the GEMINI model available on the Google Cloud API (Cloud, 2023), with a learning rate of 5e-7 and inferred with a temperature of 0. The input token length was set to 8196 tokens, and output token length was capped at 512 tokens (our outputs were binary Yes | No responses) - if our input prompt exceeded 8196 tokens, in accordance with our desire to capture the complete tree, we kept the complete main post but restricted the number of words in other comments to 12 words (this truncation was not needed in any of our structural tasks, and for about 5% of the samples in the downstream tasks). Our batch size for all the experiments was set to 128. The number of steps vary across our experiments depending on the size of the dataset - GEMINI was fine-tuned for 3000 steps for the creation of GEMINI-TT. Similarly, GEMINI-TT was further fine-tuned for 500 steps. Each experiment was run on 512 TPU v3 chips. Every 100 steps of the experiment took approximately 1 hour to complete.

⁸We do not evaluate early detection on the other two tasks, as the Fakeddit dataset does not provide timestamps, and the winning thread detection task requires the entire thread.

G Model Ablations

We also changed the base LLM from GEMINI to GEMMA and PaLM-Bison to demonstrate that the TREETOP framework works with different models as well. We show these results in this section.

G.1 Results with GEMMA

G.1.1 GEMMA Results on Structural Fine-tuning

Our results for GEMMA-TT are shown in Table 16, and show that GEMMA-TT is also able to perform well on unseen structural tasks.

Table 16: Results of structural tasks for GEMMA (Zero-shot evaluation), GEMMA-TT (GEMMA fine-tuned with the TREETOP framework). The results show that the fine-tuning process of TREETOP works for smaller LLM models like GEMMA as well.

	GE	EMMA	(Zero-	-shot)		Gemn	IA-TT	-
Fine-tuning Tasks	Acc	Rec	Pre	F1	Acc	Rec	Pre	F1
Are_one_hop_neighbors	26.3	44.4	32.6	37.6	100	100	100	100
Are_two_hop_neighbors	25.9	41.9	31.8	36.1	100	100	100	100
Are_three_hop_neighbors	23.4	41.2	30.4	35.0	100	100	100	100
Within_subtree	28.2	48.7	34.4	40.3	98.0	98.0	98.0	98.0
In_triangle	23.3	43.0	31.1	36.1	86.4	99.3	79.1	88
In_long_chain	26.0	47.7	33.0	39.0	92.7	98.7	88.2	93.1
Num_conversations	19.6	29.3	24.4	26.7	99.0	100	98.0	99.0
Num_children	18.6	23.4	21.4	22.4	99.3	99.3	99.3	99.3
Node_level	21.6	37.3	28.3.	32.2	98.7	98.7	98.7	98.7
Is_leaf	17.7	14.0	15.1	14.5	100	100	100	100
Depth	6.3	12.7	11.2	11.9	96.3	98.0	94.8	96.4
Num_leaf_nodes	22.0	33.8	27.1	30.1	78.0	87.2	73.4	79.8
Unseen Tasks								
Same_level	13.4	14.9	14.5	14.7	42.4	23.2	37.7	28.7
Has_comment_wedge	27.5	47.9	34	39.8	47.7	51.2	47.8	49.5
Are_interacting	22.1	40.9	29.7	34.4	80.8	78.3	82.4	80.3
Are_fighting	20.9	20.5	20.9	20.7	70.5	95.4	63.8	76.5
Has_user_wedge	34.7	60.0	39.8	47.9	49.5	57.3	49.5	53.2

G.1.2 GEMMA Results on Downstream Social Media Tasks

Table 17 shows the results of using GEMMA as the base model for the four different social media tasks evaluated in this paper.

Table 17: GEMMA results on downstream social media tasks. These results show that the TREETOP framework helps smaller models like GEMMA as well to improve their downstream performance.

		GEN (Zero	/MA -shot)			GEN (Fine-	MMA tuned)			GEMM (Fine-	(A-TT tuned)	
Task	Acc	Rec	Pre	F1	Acc	Rec	Pre	F1	Acc	Rec	Pre	F1
Controversial Post Detection	17.6	31.7	24.7	27.8	64.4	96.8	58.7	73.1	80.0	79.5	80.2	79.9
Winning Thread Detection	8.3	4.5	4.9	4.7	50.3	98.3	50.2	66.5	71.0	79.9	66.6	72.7
Rumor Detection	9.5	4.0	2.4	3.0	64.9	1.6	57.1	3.1	81.8	87.9	69.0	77.3
Fake News Detection	22.0	33.5	10.9	16.5	84.0	76.0	62.4	68.5	90.5	63.4	93.1	75.4

G.2 Results with PaLM-Bison

Our results for PaLM-TT are shown in Table 18, and confirm that the increased benefit of TREETOP framework is repeatable for PaLM-Bison as well.

Table 18: Results of structural tasks for PaLM-Bison (Zero-shot evaluation), PaLM-TT (PaLM-Bison fine-tuned with the TREETOP framework). The results show that the fine-tuning process of TREETOP works for other LLM models like PaLM-Bison as well.

	Pa	LM (Z	Zero-s	hot)		PaL	M-TT	
Fine-tuning Tasks	Acc	Rec	Pre	F1	Acc	Rec	Pre	F1
Are_one_hop_neighbors	59.5	92.1	55.8	69.5	99.9	99.7	100.0	99.9
Are_two_hop_neighbors	45.6	90.1	47.7	62.4	99.3	99.1	99.6	99.3
Are_three_hop_neighbors	58.1	99.5	54.4	70.3	99.6	99.5	99.7	99.6
Within_subtree	87.2	92.0	83.9	87.7	99.6	99.3	99.9	99.6
In_triangle	61.4	72.6	59.3	65.3	93.9	92.4	95.2	93.8
In_long_chain	76.9	89.3	71.5	79.5	98.0	98.5	97.5	98.0
Num_conversations	61.1	98.5	56.4	71.7	99.0	98.3	99.7	99.0
Num_children	51.6	99.2	50.8	67.2	99.5	99.1	99.9	99.5
Node_level	57.2	93.1	54.2	68.5	92.5	88.8	95.8	92.2
Is_leaf	63.6	75.9	60.9	67.6	99.6	99.3	99.9	99.6
Depth	68.2	59.8	72.0	65.3	86.4	72.8	100.0	84.3
Num_leaf_nodes	50.7	24.9	51.4	33.5	90.1	96.5	85.5	90.7
Unseen Tasks								
Same_level	53.0	10.3	70.8	18.0	65.3	64.8	65.5	65.1
Has_comment_wedge	54.9	62.2	54.2	57.9	58.1	93.6	54.8	69.1
Are_interacting	72.9	70.8	73.9	72.3	82.2	67.4	95.7	79.1
Are_fighting	59.3	98.5	55.2	70.7	93.7	97.9	90.3	93.9
Has_user_wedge	54.5	49.7	55.0	52.2	59.1	51.4	60.7	55.7

H Data Ablations

We perform multiple data ablations, cross-validation, and bootstrapping tests to ensure that the TREETOP framework is stable and our results are not dependent on (i) a particular selection of fine-tuning data corpus, or (ii) a particular train / validation split of our data corpus, (iii) or a particular sampling of the test set. We provide details of these ablation studies here.

H.1 Increasing the Size of the Structural Task Dataset

H.1.1 Results on Structural Tasks

For our first data ablation study, we increase the size of the training dataset. Instead of using 10K samples per question as described in the previous results, we trained a version of GEMINI-TT with 100K samples per task. Like before, half of these were positive samples and the other half, negative samples. We refer to this version of GEMINI-TT as GEMINI-TT-100 and show the comparison results in Table 19.

Table 19: Results of structural tasks for GEMINI-TT and GEMINI-TT-100. Acc, Rec, Pre refer to Accuracy, Recall and Precision respectively. Standard error for all reported results is less than 1.3 for model accuracy for GEMINI-TT-100 and is reported in Appendix H.2 (Table 21). The results show that there is a small gain in performance of GEMINI-TT-100 over GEMINI-TT. The results of GEMINI-TT are same as shown in Table 2, and only presented for comparison purposes.

		Gem	IINI-TT	,	C	Gemini	-TT-10	0
Fine-tuning Tasks	Acc	Rec	Pre	F1	Acc	Rec	Pre	F1
Are_one_hop_neighbors	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Are_two_hop_neighbors	99.9	99.8	100.0	99.9	100.0	100.0	100.0	100.0
Are_three_hop_neighbors	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Within_subtree	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
In_triangle	91.6	92.1	91.1	91.6	95.4	94.1	96.6	95.3
In_long_chain	<u>96.5</u>	98.7	94.6	96.6	99. 7	99.7	99.6	99.7
Num_conversations	99.9	99.9	100.0	99.9	100.0	100.0	100.0	100.0
Num_children	99.8	100.0	99.6	99.8	100.0	100.0	100.0	100.0
Node_level	<u>94.5</u>	99.2	90.7	94.8	100.0	100.0	100.0	100.0
Is_leaf	99.9	99.9	100.0	99.9	100.0	100.0	100.0	100.0
Depth	93.7	90.7	96.6	93.5	100.0	100.0	100.0	100.0
Num_leaf_nodes	<u>87.7</u>	96.7	81.9	88.7	99.6	99.6	99.6	99.6
Unseen Tasks								
Same_level	76.1	81.1	73.6	77.1	78.9	85.0	75.7	80.0
Has_comment_wedge	63.0	78.8	59.8	68.0	67.4	87.7	62.3	72.8
Are_interacting	78.2	59.1	95.7	73.0	75.8	52.6	98.6	68.6
Are_fighting	86.2	99.9	78.4	87.9	92.7	99.7	87.3	93.1
Has_user_wedge	61.6	43.3	68.3	53.0	63.9	46.3	71.6	56.2

H.1.2 Results on Downstream Social Media Tasks

We now show the performance of GEMINI-TT-100 (i.e. GEMINI-TT trained with 100K samples per task) on downstream social media tasks. Table 20 shows the results, and like before, GEMINI-TT-100 consistently shows small gains of GEMINI-TT.

Table 20: Results comparing GEMINI, GEMINI-TT, and GEMINI-TT-100 for downstream social media tasks. All results are statistically significant and the standard error is reported in Appendix H.2. Results for GEMINI and GEMINI-TT are same as before; and are only reported for comparison purposes.

	C	ontrove	rsial Po	st Dete	ction			
		Zero	-shot			Fine-	tuned	
Model	Acc	Rec	Pre	F1	Acc	Rec	Pre	F1
Gemini	50.0	27.4	58.2	37.3	68.6	85.9	64.0	73.3
Gemini-TT	50.6	90.5	50.4	64.7	84.6	87.7	82.6	85.0
Gemini-TT-100	52.5	70.0	51.9	59.6	84.5	87.7	82.6	85.1

	VV	inning A	Argume	nt Dete	ection			
		Zero	-shot			Fine-	tuned	
Model	Acc	Rec	Pre	F1	Acc	Rec	Pre	F1
Gemini	50.2	3.2	51.3	6.1	51.5	12.3	57.5	20.3
Gemini-TT	52.5	45.2	53.0	48.8	76.6	83.5	73.4	78.1
Gemini-TT-100	54.6	31.4	58.9	40.9	75.5	82.7	72.4	77.2

winning Argument Detectio	Winning	Argument	Detection
---------------------------	---------	----------	-----------

	Fake	News De	etection	ı			
	Zero	-shot			Fine-	tuned	
Acc	Rec	Pre	F1	Acc	Rec	Pre	F1
76.3	3.9	34.9	7.1	89.8	82.7	75.1	78.7
77.2	2.8	56.8	5.3	96.0	88.6	93.5	91.0
76.9	1.4	41.8	2.7	96.7	91.5	93.8	92.6
	Acc 76.3 77.2 76.9	Fake Acc Zero 76.3 3.9 77.2 2.8 76.9 1.4	Acc Zero-shot 76.3 3.9 34.9 77.2 2.8 56.8 76.9 1.4 41.8	Fake News Detection Zero-shot F1 Acc Rec Pre F1 76.3 3.9 34.9 7.1 77.2 2.8 56.8 5.3 76.9 1.4 41.8 2.7	Fake News Detection Zero-shot Acc Rec Pre F1 Acc 76.3 3.9 34.9 7.1 89.8 77.2 2.8 56.8 5.3 96.0 76.9 1.4 41.8 2.7 96.7	Fake News Detection Acc Rec Pre F1 Acc Rec 76.3 3.9 34.9 7.1 89.8 82.7 77.2 2.8 56.8 5.3 96.0 88.6 76.9 1.4 41.8 2.7 96.7 91.5	Fake News Detection Acc Rec Pre F1 Acc Rec Pre 76.3 3.9 34.9 7.1 89.8 82.7 75.1 77.2 2.8 56.8 5.3 96.0 88.6 93.5 76.9 1.4 41.8 2.7 96.7 91.5 93.8

Rumor Detection

		Zero	o-shot			Fine-	tuned	
Model	Acc	Rec	Pre	F1	Acc	Rec	Pre	F1
Gemini	64.0	6.1	41.3	10.6	72.5	47.1	63.7	54.1
Gemini-TT	65.9	3.1	76.5	5.8	87.1	78.1	83.6	80.7
Gemini-TT-100	65.7	2.0	100.0	3.9	87.0	79.3	83.1	81.1

H.2 Bootstrapping Analysis of Test Set

In our next ablation study, we perform bootstrapping of the test set and evaluate against a single version of GEMINI-TT. Bootstrapping is performed using random sampling with replacement within the test data corpus. The methodology and statistical analysis for bootstrapping is described in Efron and Tibshirani (1993) - the process is visualized in Figure 8 and our implementation is given in Algorithm 1. When performing bootstrapping test analysis for structural fine-tuning, the 'model' in Figure 8 refers to GEMINI-TT. However, when performing bootstrapping test analysis for downstream social media tasks, the 'model' in Figure 8 refers to the version of GEMINI-TT-FT specifically finetuned for that downstream task. We perform bootstrapping to ensure proper estimation of our test set metrics (accuracy, recall, precision and f1).



Figure 8: We perform bootstrapping analysis on the test set (Efron and Tibshirani, 1993) to ensure that our reported test results are statistically reliable.

Algorithm 1 Estimate Standard Error (SE) Bars via Bootstrap

1: Input: Test data $T = \{x_1, \ldots, x_n\}$, Number of bootstrap samples B, Metric $M : T \to \mathbb{R}$

2: **Output:** SE of the metric evaluated at T

- 3: for i = 1 to *B* do
- Sample *n* data points from *T* with replacement to form S_i where $|S_i| = n$ 4:
- 5: Compute $M_i = M(S_i)$.
- 6: end for
- 7: Compute the metric for the original population $M^* = M(T)$ 8: Compute Standard Error as $SE = \sqrt{\frac{1}{B-1}\sum_{i=1}^{B}(M_i M^*)^2}$
- 9: Return SE

H.2.1 Bootstrapping Tests for Structural Fine-tuning

We first show the standard error for structural fine-tuning tasks in Table 21 (for 'mean', refer to Table 2).

Table 21: Mean and standard error from the bootstrapping analysis for structural fine-tuning tasks.
The table demonstrates that all standard errors for models in this work are less than 2.0 for accuracy.
Bootstrapping Test Set: Standard Error for Structural Tasks (\downarrow is better)

	Ge	MINI	(Zero-	-shot)		Gem	ini-T	Т	Ge	MINI	-TT-1	00
Fine-tuning Tasks	Acc	Rec	Pre	F1	Acc	Rec	Pre	F1	Acc	Rec	Pre	F1
Are_one_hop_neighbors	1.3	1.7	2.1	1.7	0.0	0.0	0.0	0.0	0.0	$\begin{array}{c} 0.0\\ 0.0\end{array}$	0.0	0.0
Are_two_hop_neighbors	1.2	1.8	2.2	1.7	0.1	0.2	0.0	0.1	0.0		0.0	0.0
Are_three_hop_neighbors Within_subtree	1.3 1.0	1.7 1.3	2.4 1.5	1.8 1.1	0.0 0.0	$\begin{array}{c} 0.0\\ 0.0\end{array}$	0.0 0.0	$\begin{array}{c} 0.0\\ 0.0\end{array}$	0.0 0.0	$\begin{array}{c} 0.0\\ 0.0\end{array}$	0.0 0.0	$\begin{array}{c} 0.0\\ 0.0\end{array}$
In_triangle	2.0	2.7	3.3	2.7	0.7	1.0	1.0	0.7	0.3	0.6	0.4	0.3
In_long_chain	1.8	1.1	14.7	2.0	0.4	0.4	0.7	0.4	0.0	0.1	0.1	0.0
Num_conversations	1.5	10.2	1.8	5.1	0.1	0.1	0.0	0.1	0.0	0.0	0.0	0.0
Num_children Node_level Is_leaf	1.5 1.2 1.9	2.0 1.4 3.1	2.7 1.5 3.5	2.1 1.2 3.0	0.1 0.0 0.1	$0.0 \\ 0.0 \\ 0.1$	0.2 0.0 0.0	0.1 0.0 0.1	$0.0 \\ 0.0 \\ 0.0$	$0.0 \\ 0.0 \\ 0.0$	$0.0 \\ 0.0 \\ 0.0$	0.0 0.0 0.0
Depth	1.3	1.4	3.4	1.9	0.1	0.2	0.2	0.1	0.0	0.0	0.0	0.0
Num_leaf_nodes	1.3	2.0	2.1	1.7	1.0	0.7	1.4	0.9	0.1	0.1	0.1	0.1
Unseen Tasks												
Same_level	1.3	0.9	5.9	1.6	1.2	1.6	1.6	1.3	1.2	1.4	1.6	1.2
Has_comment_wedge	1.5	1.1	3.9	1.7	1.2	1.5	1.4	1.3	1.3	1.3	1.5	1.2
Are_interacting	1.2	0.8	5.3	1.4	1.1	1.8	0.9	1.4	1.1	1.9	0.6	1.6
Are_fighting	1.2	1.4	3.1	2.0	0.9	0.1	1.3	0.8	0.7	0.2	1.1	0.7
Has_user_wedge	1.4	1.2	3.9	1.8	1.3	1.9	2.1	1.8	1.2	1.9	2.1	1.6

H.2.2 Bootstrapping Tests for Downstream Social Media Tasks

We now report the standard error for the controversial post detection task in Table 22 (for 'mean', refer to Table 3). Similarly, we report standard error for winning argument detection task in Table 23 (for 'mean', refer to Table 4), for fake news detection in Table 24 (for 'mean', refer to Table 14), and for rumor detection in Table 25 (for 'mean', refer to Table 15).

Table 22: Standard error for the Controversial Post detection task. The table demonstrates that all standard errors for model accuracy in this evaluation are less than 1.3.

Bootstrapping Test for Controversial Post Detection Standard Error (1 is better)

		Zero-	snot		Fine-tuned			
Model	Acc	Rec	Pre	F1	Acc	Rec	Pre	F1
Gemini	1.3	1.6	2.4	1.8	1.0	1.0	1.3	1.0
Gemini-TT	1.0	0.9	1.1	1.0	0.8	1.0	1.1	0.8
Gemini-TT-100	1.1	1.5	1.3	1.2	0.7	0.9	1.1	0.7

Table 23: Standa	rd error for the V	Winning Argument	t detection task.	The table der	nonstrates that	at all
standard errors fo	or model accurac	y in this evaluation	n are less than 1.	.4.		

Standard Error (\downarrow is better)								
	Fine-tuned							
Model	Acc	Rec	Pre	F1	Acc	Rec	Pre	F1
Gemini	1.4	0.7	7.6	1.2	1.1	1.1	3.2	1.6
Gemini-TT	1.1	1.6	1.7	1.4	0.9	1.2	1.3	1.0
Gemini-TT-100	1.0	1.5	1.7	1.5	1.0	1.2	1.3	1.0

Bootstrapping Test for Winning Argument Detection Task

Table 24: Standard error for the Fake News detection task. The table demonstrates that all standard errors for model accuracy in this evaluation are less than 0.2.

Bootstrapping Test for Fake News Detection Ta	ısk
Standard Error (\downarrow is better)	

		Zero-	shot	Fine-tuned				
Model	Acc	Rec	Pre	F1	Acc	Rec	Pre	F1
Gemini	0.2	0.2	1.7	0.4	0.2	0.4	0.4	0.3
Gemini-TT	0.2	0.2	2.3	0.3	0.1	0.4	0.3	0.3
Gemini-TT-100	0.2	0.1	2.8	0.2	0.1	0.3	0.3	0.2

H.3 TREETOP Framework Fine-tuning Corpus Ablations

In the second ablation study, we change the entire corpus of the structural task data set multiple times to create multiple versions of GEMINI-TT. Our process is described in Figure 9. We perform this ablation study ensure that our process of fine-tuning using a structural task dataset is repeatable and not dependent on just one specific selection of data set. In our study, we created four independent non-overlapping dumps of the dataset, and we report the mean and standard error across this data dump ablation in Table 26.

Table 25: Standard	l error for the Rum	or detection task.	The table	demonstrates	that all s	tandard e	errors
for model accurac	y in this evaluation	n are less than 1.9	Э.				

Bootstrapping Test for Rumor Detection Task Standard Error (↓ is better)								
Zero-shot Fine-tuned								
Model	Acc	Rec	Pre	F1	Acc	Rec	Pre	F1
Gemini	1.5	1.8	9.5	3.0	1.9	3.9	4.5	3.7
Gemini-TT	1.8	1.2	15.1	2.3	1.3	2.6	2.6	2.1
Gemini-TT-100	1.8	0.9	0.0	1.8	1.2	2.4	2.8	2.0



Figure 9: We change the entire structural task fine-tuning corpus and train multiple versions of GEMINI-TT to ensure that our results are not sensitive to any particular choice of structural task data corpus, and that our methodology of fine-tuning using the structural tasks is repeatable.

		Mean (↑ is better)			S	tanda (is l	rd Erro	or
Fine-tuning Tasks	Acc	Rec	Pre	F1	Acc	Rec	Pre	F1
Are_one_hop_neighbors	100.00	100.00	100.00	100.00	0.00	0.00	0.00	0.00
Are_three_hop_neighbors Within subtree	99.95 100.00	99.95 100.00	100.00	99.78 99.95	0.03	0.07	0.00	0.03
In_triangle	89.13	91.43	87.48	89.38	0.8	0.40	1.36	0.76
Num_conversations	99.93	99.90 99.90	99.98	90.90 99.93	0.03	0.10	0.49	0.24
Num_children Node_level Is_leaf	99.65 95.68 99.98	99.95 99.08 99.98	99.33 92.98 100.00	99.65 95.90 99.98	0.09 1.47 0.03	0.03 0.35 0.03	0.16 2.39 0.00	0.09 1.39 0.03
Depth Num_leaf_nodes	95.18 86.80	93.78 94.48	96.53 81.93	95.10 87.73	1.54 0.54	2.02 1.14	1.21 0.62	1.66 0.53
Unseen Tasks		Ν	lean		S	tanda	rd Erro	or
Same_level Has_comment_wedge	75.40 61.97	77.17 76.10	74.60 59.33	75.80 66.63	0.35 0.54	1.81 1.31	0.65 0.32	0.65 0.7
Are_interacting Are_fighting Has_user_wedge	78.43 86.10 62.07	59.40 99.77 42.83	95.97 78.37 69.73	73.40 87.80 53.03	0.31 0.10 0.36	1.13 0.06 0.32	0.81 0.13 0.63	1.13 0.10 0.18

Table 26: Standard error across four data dump ablation study for GEMINI-TT. As the table shows, the standard errors are less than 1.54 for accuracy. TREETOP Framework Corpus Ablation Study

H.4 K-Fold Cross-Validation for Downstream Social Media Tasks



Figure 10: We perform the standard k-fold cross validation for each downstream social media task with GEMINI-TT to ensure that our results are not over-fitted for any particular fold of the dataset.

In our third ablation study, we perform standard k-fold cross-validation for each downstream task. We create multiple versions of fined-tuned GEMINI-TT for each downstream task. Our process is described in Figure 10. We perform this ablation study ensure that our process of fine-tuning using a structural task dataset is not over-fitted to any one particular training data corpus. Our results and standard error are shown in Table 27.

		Me	Standard Error							
		$(\uparrow is better)$				$(\downarrow is better)$				
Task	Acc	Rec	Pre	F1	Acc	Rec	Pre	F1		
Controversial Post Detection	84.22	87.88	81.96	84.78	0.16	1.29	0.86	0.17		
Winning Thread Detection	76.52	79.58	74.98	77.18	0.18	1.36	0.52	0.39		
Rumor Detection	86.60	79.12	82.02	80.5	0.27	0.36	0.80	0.31		
Fake News Detection	97.14	92.88	94.6	93.72	0.32	1.25	0.49	0.76		

Table 27: Mean and standard error for the k-fold cross-validation study for the different algorithms for GEMINI-TT. The standard errors are less than 0.32 for accuracy.

I Regression Analysis

We ran MMLU and GSM8K benchmarks comparing both zero-shot GEMINI and zero-shot GEMINI-TT to understand the regression loss incurred by the native LLM due to TREETOP. We show the model accuracies in Table 28. The results show that TREETOP fine-tuning does not incur any substantial loss in LLM performance as measured against the MMLU and GSM8K benchmarks. This serves as an empirical stamp of robustness for the TREETOP framework.

Table 28:	Evaluation	of	regression	loss	(model
accuracy is	s reported).				

Benchmark	Gemini	GEMINI-TT
MMLU	65.65	65.47
GSM8K	83.5	83.0

J Embeddings Analysis

We also analyzed the embeddings of PaLM-TT, with an intent to test whether the model actually understood the topology - we designed our experiment to evaluate whether the topological characteristics are directly captured in the learnt embeddings output by the model. We used PaLM-TT for this analysis, given it was not possible for us to obtain embeddings from GEMINI-based models.

Our experimental results show a clear presence of clustering in the embedding space for inputs that conform to similar topologies. In fact, a classifier trained purely on topological embeddings obtains a 77% accuracy on the multi-class topology classification task using a simple logistical regression classifier. We now describe the experimental setup that leads to these conclusions.

J.1 Experimental Setup

In this section, we describe our experimental setup. The goal of these experiments is to show that embeddings learnt by the PaLM-TT model show clustering — i.e. topologies that are similar, by some independent analytical metric, cluster together. Towards this goal, we chose 5 different topologies, which are shown in Figure 11. These five topologies are chosen so that they vary in their depth and breadth. The hypothesis at the outset is to show that embeddings cluster by topology.

We first begin by presenting the statistics of our topologies in Table 29. While the topologies are illustrated in Figure 11, we list them out in our table with an equivalent depth-first crawl of the tree: the list of children of a node are captured with a pair of parenthesis, and the letter v represents a leaf.



Figure 11: Five topologies used for PaLM-TT embedding analysis.

We now describe the method by which we obtain the embeddings. The embedding is obtained by doing a forward pass of the PCT encoding in PaLM-TT for each conversation tree in our dataset. We believe that the *last input token's* embedding is sufficient capture the entire prompt at inference time, and so we only use the embedding of the last input token in our analysis. Let's call such an embedding as E(PCT). Similarly, we forward pass the Post Comments encoding (refer Appendix C.2) from the same conversation tree, and obtain E(PC). The embedding dimension for each token in PaLM-TT (as in PaLM-Bison) is 4096. Now we are left with the task of obtaining embeddings – call this E(Topo) that refer purely to the embeddings of the topology. We define this as

$$E(Topo) = E(PCT) - E(PC)$$
(1)

Tabl	e 29: `	We rand	lomly	y cho	se son	ne topo	ologie	s to e	aluate	cluste	ering of	emb	edd	ings. '	These	topol	ogies
vary	given	the first	t one	has l	high d	epth a	nd the	last o	one has	high	breadth	ı. Th	ey a	re sho	wn in	Figu	re 11.

Topology	Number of Samples
(((((v)))))	659
(v,(((v)))))	436
((v),(((v))))	602
(v,v,v,v,(v))	1252
(v,v,v,v,v,v,v,v)	382

Our assumption is that E(Topo) is a good approximation for the topological structure, and we show our analysis on these E(Topo) embeddings. Our hypothesis is that E(Topo) will demonstrate correlation with the actual tree topology.

To demonstrate that our embeddings capture patterns from the respective topologies, we examine two types of metrics. First, we aim to show separability between embeddings from different clusters. We hypothesize that if a linear (or logistic) classifier can effectively learn the classes using the embeddings, this would demonstrate such separability. Next, we look at a clustering distance metric – called the variance ratio criterion. We hope to show that the VRC metrics reflects the distances of another independent metric, for example the tree edit-distance metric as proposed by the Zhang Shasha algorithm (Zhang and Shasha, 1989). We describe these experiments in the next two sections.

J.2 Multi-class Topological Classifier using PaLM-TT's Embeddings

In the first study, we utilized embeddings to classify various topological classes, achieving impressive performance metrics as detailed in Table 30. We studied two different types of models to explain the complexity of the dataset – (i) logistical classifier (ii) a two-layer MLP. If the data cannot be learnt by the first classifier, but can be learnt by the second, it would show that the data has complex patterns. If it can be learnt by both, then the embeddings show clean separation between the different classes. If it cannot be learnt by either model, then the embeddings may not capture the topological patterns (or otherwise the patterns are too complex to be learnt by these simple models).

Neural Network						
Topology	Accuracy	Precision	Recall	F1		
(((((v)))))	73.0	71.0	72.0			
(v,((((v)))))	40.0	31.0	35.0			
((v),(((v))))	55.0	62.0	58.0			
(v, v, v, v, (v))	84.0	89.0	87.0			
(v,v,v,v,v,v,v,v)	74.0	67.0	70.0			
Overall	72.0	71.0	72.0	72.0		
Logistic Regression						
(((((v)))))	79.0	77.0	78.0			
(v,(((v)))))	45.0	50.0	47.0			
((v),(((v))))	65.0	62.0	63.0			
(v,v,v,v,(v))	90.0	90.0	90.0			
(v,v,v,v,v,v,v,v)	78.0	77.0	78.0			
Overall	77.0	77.0	77.0	77.0		

Table 30: Performance metrics for Neural Network and Logistic Regression models on the topological embedding to classify the topological class.

We show the topological classes in Figure 11 and their statistics in Table 29. Note that *both* the models can learn the dataset, to the extent of 77% accuracy! This shows that the embeddings cleanly capture the topological patterns. These metrics underscore the **efficacy of our approach in capturing the characteristics of topological classes** using embeddings.

J.3 Clustering Metrics from the Embedding

Our topological embeddings show that the tree distance metric is highly correlated with the variance ratio criterion (VRC) metric. The VRC is defined as follows:

$$\operatorname{VRC} = \frac{\operatorname{Tr}(B_k)}{\operatorname{Tr}(W_k)} \times \frac{N-k}{k-1}$$
(2)

where the $Tr(B_k)$ is the trace of the between-cluster dispersion matrix, $Tr(W_k)$ is the trace of the within-cluster dispersion matrix, and N and k are the number of data points and the number of clusters respectively (and listed in Table 29).

Here the trace of B_k and W_k are defined as follows:

$$\operatorname{Tr}(B_k) = \sum_{j=1}^k n_j (\mathbf{c}_j - \mathbf{c})^2 \quad \text{and} \quad \operatorname{Tr}(W_k) = \sum_{j=1}^k \sum_{\mathbf{x} \in C_j} (\mathbf{x} - \mathbf{c}_j)^2$$
(3)

Here, each topology is its own cluster, and c_j is the centroid of cluster j. c is the centroid of all data points in the dataset. Further, n_j is the number of data points in cluster j, and the set of all points in cluster j is C_j . x represents the embedding, E(Topo), for the sample in question.

The tree distance metric is an edit distance between the topologies of the respective trees. This is implemented using the Zhang Shasha (Zhang and Shasha, 1989) algorithm. One notices that the VRC metric of the trees in the cluster closely follows the edit distance between the topologies of the clusters.



(a) Tree edit distance between our topologies based on the Zhang Shasha algorithm.

(b) Variance ratio criterion (VRC) metric for different clusters of E(Topo).

Figure 12: Comparison of PaLM-TT embeddings with VRC criterion and tree edit distance. Each cell shows the value of the pairwise distance metric (self distances are 0 for tree edit distance and not defined for VRC), the ordinal ranking of the cell in the top-left corner, and is color coded by this ordinal rank. The ordinal ranking ranks the topologies in the columns based on their distance from the topology specified in the row. Figure 12a shows the color map with the ordinal rankings based on the tree edit distance measure, whereas Figure 12b shows the color map based on the variance ratio. The comparison is **of the ordinal ranking** and **not of the absolute metric** values. The similar color coding of the two plots implies similar ordinal ranking and demonstrates that VRC metric of the trees in the cluster closely follows the edit distance between the topologies of the clusters - thereby implying that embeddings are indeed learning the topology.

K Prompt Interactions with GEMINI and GEMINI-TT

We randomly chose few examples to probe GEMINI and GEMINI-TT models for our structural tasks. We present the tasks and the interactions with the models in Tables 31, 32, and 33. For this table, we changed our prompt slightly and asked the model to provide more details (instead of only responding with a yes or a no). We present three example interactions here: one for a In_long_chain task, one for In_triangle task, and one for a Are_three_hop_neighbors task. In all the three examples below, GEMINI gave the wrong response whereas GEMINI-TT gave the correct response with the correct reasoning.

Task	In_long_chain						
Input Topology	U ₁ C ₁ C ₂ U ₀ U ₁ C ₂ U ₀ U ₂ C ₅ C ₄						
Prompt Question	Whether there exist a long chain of to and fro discussion of at least length 4 between two users in the tree ?						
GEMINI Response	No, there is no long chain of to and fro discussion of at least length 4 between two users. The conversation tree given only has a maximum chain length of 3, between U0 and U1. The chain goes as follows: 1. U0 comments on the main post <co> 2. U1 comments on U0's comment <c1> 3. U0 comments on U1's comment <c2> There is no instance where U1 comments on U0's comment to U1's comment, or vice versa. Therefore, there is no long chain of to and fro discussion of at least length 4 between two users.</c2></c1></co>						
GEMINI-TT Response	<pre>Yes, there is a long chain of to and fro discussion of length 4 between users U0 and U1. The relevant comments are: - C1: U1 comments on a comment by U0 (CO). - C2: U0 comments on U1's comment (C1). - C3: U1 comments on U0's comment (C2). - C4: U0 comments on U1's comment (C3).</pre>						

Table 31: Example Task 1: Interacting with GEMINI and GEMINI-TT with our structural tasks.



Table 32: Example Task 2: Interacting with GEMINI and GEMINI-TT with our structural tasks.

Task	In_triangle
Input Topology	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$
Prompt Question	Whether users <u0>, <u1>, and <u2> are not involved in a triangular discussion between each other. $?$</u2></u1></u0>
GEMINI Response	No, users <uo>, <u1>, and <u2> are not involved in a triangular discussion between each other. Although users <u1> and <u2> have commented on each other's comments, user <u0> has not commented on either of their comments. Therefore, there is no triangular discussion between the three users.</u0></u2></u1></u2></u1></uo>
GEMINI-TT Response	<pre>Yes, users <u0>, <u1> and <u2> are involved in a triangular discussion. - User <u1> comments on a comment by User <u0> (C1). - User <u2> comments on the same comment by User <u0> (C2). - User <u1> and User <u2> comment on each other's comment on User <u0>'s comment (C3). Therefore, the three users are involved in a triangular discussion.</u0></u2></u1></u0></u2></u0></u1></u2></u1></u0></pre>

 Table 33: Example Task 3: Interacting with GEMINI and GEMINI-TT with our structural tasks.

L Licenses and Copyrights Across Assets

- 1. The Pushshift Reddit Dataset
 - Citation: (Baumgartner et al., 2020)
 - Asset Link: [link]
 - License: CC By 4.0
- 2. Controversial Post
 - Citation: (Hessel and Lee, 2019)
 - Asset Link: [link]
 - License: CC By 4.0
- 3. PHEME9 (rumor detection)
 - Citation: (Kochkina et al., 2018)
 - Asset Link: [link]
 - License: CC By 4.0
- 4. Fakeddit
 - Citation: (Nakamura et al., 2019)
 - Asset Link: [post] [comments]
 - License: CC By 4.0
- 5. Winning Argument Thread
 - Citation: (Tan et al., 2016)
 - Asset Link: [link]
 - License: ACM Copyright
- 6. Gemini
 - Citation: (Team et al., 2023)
 - Asset Link: [link]
 - License: Google APIs Terms of Service
- 7. PaLM-Bison
 - Citation: (Google and et al., 2023)
 - Asset Link: [link]
 - License: Google APIs Terms of Service
- 8. Huggingface Assets
 - GEMMA-2-2B-IT: [link]
 - PHI-3.5-Mini-Instruct: [link]
 - MISTRAL-7B-Instruct-v0.2: [link]
 - GEMMA-2-9B-IT: [link]
 - MMLU: [link]
 - GSM8K: [link]