Let's Decrypt Dot by Dot: Decoding Hidden Computation in Transformer Language Models

Aryasomayajula Ram Bharadwaj
Independent Researcher
ram.bharadwaj.arya@gmail.com

August 23, 2024

Abstract

Recent work has shown that transformer models can perform complex reasoning tasks using Chain-of-Thought (COT) prompting, even when the COT is replaced with hidden characters. This paper investigates methods to decode these hidden computations, focusing on the 3SUM task. We analyze a 34M parameter LLaMA model trained on hidden COT sequences and propose a novel decoding method that successfully recovers the original COT. Our findings provide insights into how transformers encode and process information in hidden COT sequences, offering new perspectives on model interpretability and the nature of computation in language models.

1 Introduction

Chain-of-Thought (COT) prompting has emerged as a powerful technique for improving the performance of large language models on complex reasoning tasks [1]. However, recent work by [2] demonstrates that these improvements can be achieved even when the COT is replaced with hidden characters (e.g., "..."), raising intriguing questions about the nature of computation being performed within these models.

This paper builds upon the findings of [2], focusing on the 3SUM task as a case study. We aim to decode the hidden computations embedded within the transformer architecture when trained on hidden COT sequences. Our work provides valuable insights into how these models encode and process information, potentially leading to improved model interpretability and more effective training strategies.

2 Background

2.1 The 3SUM Task

The 3SUM task involves finding three numbers in a given set that sum to zero. It serves as a proxy for more complex reasoning tasks and has been used to study the computational capabilities of transformer models [2].

2.2 Hidden Chain-of-Thought

In hidden Chain-of-Thought, intermediate reasoning steps are replaced with hidden characters (e.g., "..."). Models trained on hidden sequences still perform well, suggesting that meaningful computation occurs despite the lack of explicit reasoning steps [2].

3 Methodology

We used a 34M-parameter LLaMA model with 4 layers, 384 hidden dimension, and 6 attention heads [4], the size of the training and test datasets and all other hyperparameters are kept the same as in the "Let's think dot by dot" paper [2]. Our analysis focused on three main areas: Layer-wise Representation Analysis, Token Ranking, and Modified Greedy Decoding Algorithm.

4 Results and Discussion

4.1 Layer-wise Analysis

Our analysis revealed a gradual evolution of representations across the model's layers. The initial layers primarily contained raw numerical sequences associated with the 3SUM problem's chain of thought. However, starting from the third layer, we noticed the emergence of hidden tokens. As we progressed through subsequent layers, we observed a steady transition from purely numerical sequences to an increasing prevalence of hidden characters. This pattern suggests that the model develops the ability to utilize hidden tokens as proxies only in its deeper layers, which aligns with intuitive expectations of how neural networks process information. After conducting a comprehensive evaluation across numerous examples, we found that, on average, there is a marked increase in hidden token usage immediately following the second layer. Furthermore, the final layers of the model demonstrate extensive reliance on these hidden tokens. For this analysis, we employed nostalgebraist's logit lens method [3].

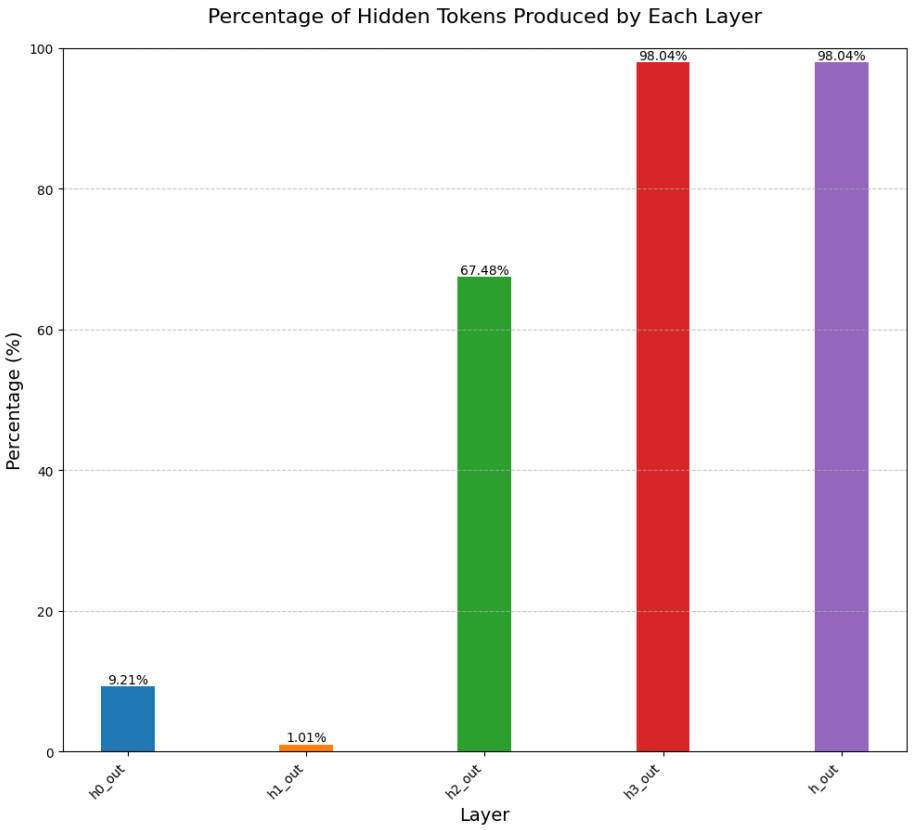

Figure 1: Hidden token occurrence percentages in generated sequences across layers

4.2 Token Rank Analysis

The top-ranked token was consistently the hidden character ("."), while lower-ranked tokens revealed the original, non-hidden COT sequences. This supports the hypothesis that the model replaces all computation with hidden tokens on top while keeping the original computation intact underneath. We have provided sample a sample snapshot of sequences decoded at each layer for both the top (rank-1) and rank-2 tokens in the appendix section.

4.3 Modified Greedy Decoding Algorithm

Based on the observations made in the token rank analysis, We implemented a modified greedy autoregressive decoding method. The steps include: performing standard greedy decoding, selecting the second-highest probability token when encountering a hidden token, and continuing this process for the entire sequence. This resulted in a 100% match in 3SUM task results with and without hidden tokens. To test the effectiveness of this method, we have also compared this with replacing the hidden token with randomly sampled token instead of the next highest (rank-2) token. The percentages of the original and these modified decoding methods are visualized as plots below.

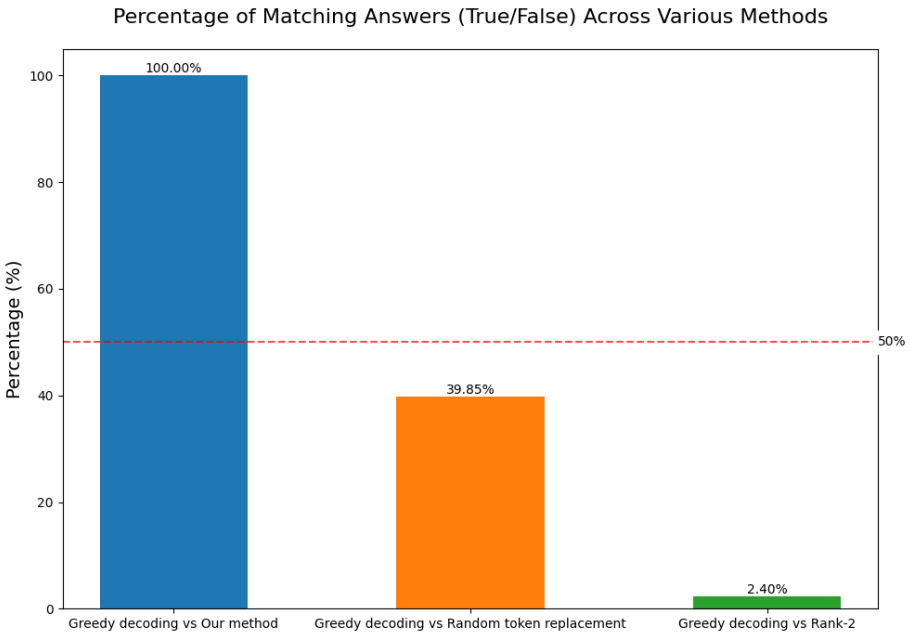

Figure 2: Comparison of decoding methods

5 Implications and Future Work

Our findings provide new tools for understanding internal reasoning processes and increase confidence in COT-based approaches for improving visibility. Future work should focus on developing better decoding methods or finding circuits that hide tokens, investigating generalizability to tasks beyond 3SUM (including natural language tasks), and improving token hiding methods (currently limited to one hidden token which is simple to decode).

6 Conclusion

We have presented a novel approach to understanding hidden computations in transformer models through the analysis of token rankings and layer-wise representations, and the development of a modified decoding algorithm. Our insights into how models encode and process information in hidden COT sequences open new avenues for improving interpretability, efficiency, and safety in language models. This work strengthens the belief in COT visibility research for interpretability.

The code used for the experiments and analysis in this paper is available on GitHub at https://github.com/rokosbasilisk/filler_tokens/tree/v2.

7 Appendix: Layerwise view of various decoding methods

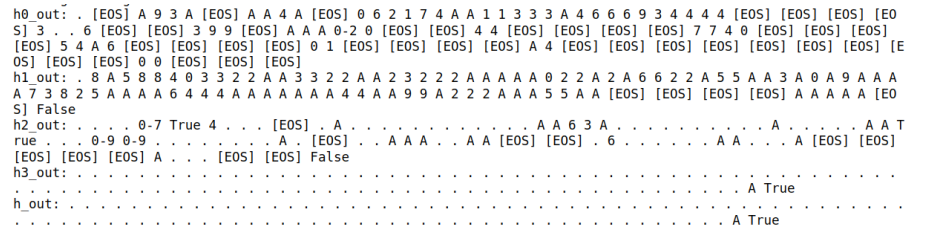

Figure 3: Greedy Decoding

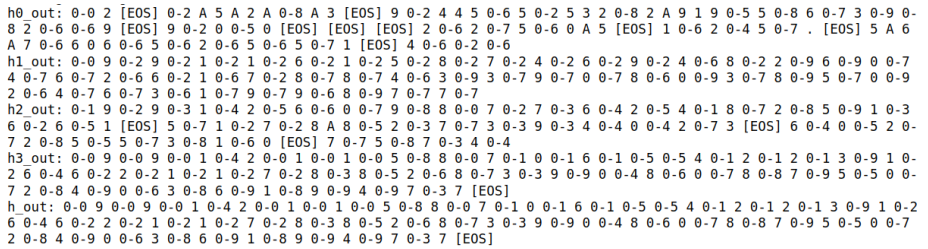

Figure 4: Greedy Decoding with Rank-2 Tokens

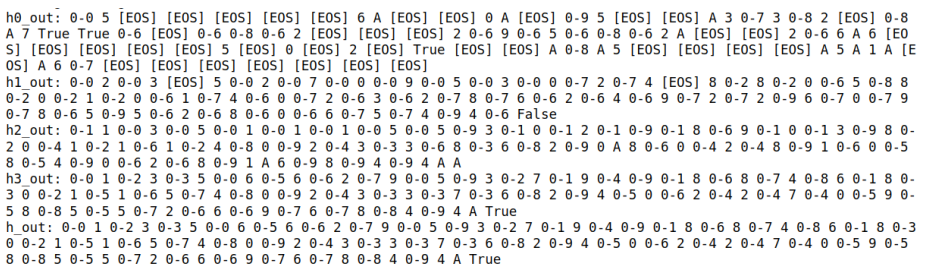

Figure 5: Our Method: Greedy Decoding with Hidden Tokens Replaced by Rank-2 Tokens

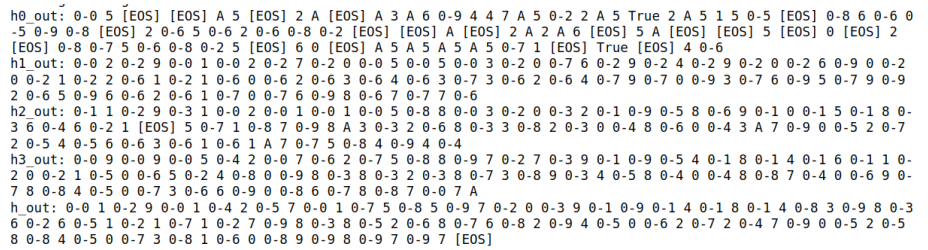

Figure 6: Greedy Decoding with Hidden Tokens Replaced by Randomly Selected Tokens

References

[1] Pfau, J., Merrill, W., & Bowman, S. R. (2023). Let's Think Dot by Dot: Hidden Computation in Transformer Language Models. arXiv:2404.15758.

[2] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., ... & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. arXiv:2201.11903.

[3] nostalgebraist (2020). interpreting GPT: the logit lens interpreting-gpt-the-logit-lens

[4] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M. A., Lacroix, T., ... & Lample, G. (2023). LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971.