When a Comma Holds More Context Than a Noun: Uncovering Hidden Memory in LLM

Anton Razzhigaev[1,2], Matvey Mikhalchuk[1,4], Temurbek Rahmatullaev[1,3,4], Elizaveta Goncharova[1,3], Polina Druzhinina[1,2], Ivan Oseledets[1,2], and Andrey Kuznetsov[1]

[1]AIRI, [2]Skoltech, [3]HSE University,
[4]Lomonosov Moscow State University
razzhigaev@skol.tech

Abstract

We introduce methods to quantify how Large Language Models (LLMs) encode and store contextual information, revealing that tokens often seen as minor (e.g., determiners, punctuation) carry surprisingly high context. Notably, removing these tokens — especially stopwords, articles, and commas — consistently degrades performance on MMLU and BABILong-4k, even if removing only irrelevant tokens. Our analysis also shows a strong correlation between contextualization and *linearity*, where linearity measures how closely the transformation from one layer's embeddings to the next can be approximated by a single linear mapping. These findings underscore the hidden importance of "filler" tokens in maintaining context. For further exploration, we present **LLM-Microscope**, an open-source toolkit that assesses token-level nonlinearity, evaluates contextual memory, visualizes intermediate layer contributions (via an adapted Logit Lens), and measures the intrinsic dimensionality of representations. This toolkit illuminates how seemingly trivial tokens can be critical for long-range understanding.

1 Introduction

Large Language Models (LLMs) have significantly advanced the field of natural language processing, achieving remarkable results across a wide range of tasks. Despite their success, the internal mechanisms by which these models operate remain largely opaque, making it challenging to interpret how they process and utilize contextual information. This opacity limits our ability to enhance model performance and to understand the reasoning behind their predictions.

While recent studies have begun to uncover specific patterns and mechanisms within LLMs (**?**), many fundamental aspects — such as their handling of step-by-step reasoning and long-range dependencies — are still not well understood. This gap in understanding hinders the development of more interpretable and efficient language models.

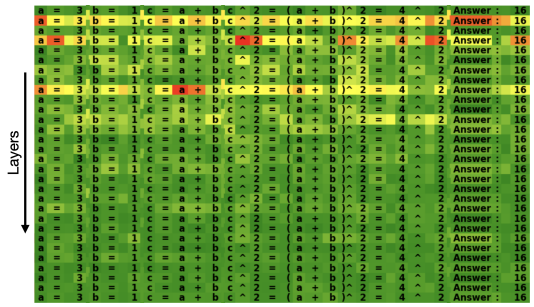

Figure 1: An example of token-wise non-linearity visualization for Llama3-8B.

To bridge this gap, we introduce **LLM-Microscope**, a comprehensive framework designed to analyze and visualize the internal behaviors of LLMs. Our toolkit offers a suite of methods that enable researchers to inspect how models encode and aggregate contextual information:

- **Token-level nonlinearity**: We measure the nonlinearity at the token level, quantifying how closely transformations between layers can be approximated by a single linear mapping.
- **Contextualization assessment**: We present a method for measuring contextualization, allowing the identification of tokens — often overlooked ones like punctuation or articles — that carry the most contextual information.
- **Intermediate layer analysis**: We examine how next-token prediction evolves across different layers, adapting the Logit Lens technique for multimodal LLMs.

Applying these tools to various scenarios — ranging from multilingual prompts to knowledge-intensive tasks — we uncover intriguing patterns in how LLMs process and transform information.

Findings of the Association for Computational Linguistics:
NAACL 2025, pages 7772–7779

Notably, our analysis reveals that certain "filler" tokens, such as punctuation marks, stopwords, and articles, are highly contextualized and act as key aggregators in language understanding. We also find a strong correlation between linearity and contextualization scores in token representations.

Furthermore, we demonstrate the practical implications of our findings by showing that removing these tokens degrades performance on tasks requiring specialized knowledge and longer-context reasoning, such as MMLU and BABILong-4k. This performance drop persists even when we carefully remove only tokens deemed irrelevant by a strong language model (GPT-4o). These results highlight the hidden importance of seemingly "trivial" tokens in maintaining coherent context.

LLM-Microscope is designed to be accessible for both researchers and practitioners, providing an intuitive interface for in-depth model analysis. We offer:

- An open-source Python package[1]
- A demo website on Hugging Face Spaces[2]

Our contributions aim to facilitate a deeper understanding of LLMs, promoting transparency and interpretability in natural language processing models.

2 Related works

Interpretability There are several significant paradigms for model interpretation, each with its own distinct properties. Probing methods are designed to train classifiers based on hidden representations that are challenged in encoding specific knowledge (**????**). While these approaches show whether specific language features are incorporated into LLMs, they do not analyze internal representations during knowledge activation, leaving the model's behavior largely a black box.

In contrast, mechanistic interpretability introduces approaches to explore the inner behavior of models. **?** mention that mechanistic interpretability aims to explore the internal representations of deep learning models through the activations of specific neurons and layer connections. A significant branch of research dedicated to examining model responses involves probing changes in behavior resulting from perturbations, noise in embeddings, or masking of network weights (**?????**).

Discovering interpretable features through training sparse autoencoders (SAEs) has become a promising direction in the LLM interpretation (**??**). Typically, SAEs focus on activations of specific LLM components, such as attention heads or multilayer perceptrons (MLPs). By decomposing model computations into understandable circuits, we can see how information heads, relation heads, and MLPs encode knowledge(**?**).

While most research has concentrated on the analysis of Small Language Models, such as GPT-2 (**?**) and TinyLLAMA (**?**), recent work has advanced this area by proposing modifications to improve the scalability and sparsity of autoencoders for larger LLMs, such as GPT-4 or Claude 3 Sonet (**??**).

Linearity of LLM hidden states The study of the internal structure of transformer-based models has been of great interest among researchers (**?????**). Several studies, such as "Logit Lens"[3], have explored projecting representations from the intermediate layers into the vocabulary space by observing their evolution across different layers (**??**). Relying on this research, the authors also investigate the complex structure of hidden representations through linearization (**??**).

Contextualization of LLM hidden states One of the areas of research into the internal representations of Transformers is the embeddings contextualization analysis. Recent studies have demonstrated that sentence representations provided by Transformer decoders can contain information about the entire previous context (**??**). **?** proposed two initial methods for reconstructing original texts from model's hidden states, finding these methods effective for the embeddings from shallow layers but less effective for deeper layers, known as "Embed Parrot."

Our work proposes a unified framework for LLM interpretability by exploring properties such as linearity, anisotropy, and intrinsic dimension of hidden representations. We introduce new approaches to assess contextual memory in token representations and analyze intermediate layer contributions to token prediction.

[1] https://github.com/AIRI-Institute/LLM-Microscope/tree/main

[2] https://huggingface.co/spaces/AIRI-Institute/LLM-Microscope

[3] https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens

3 LLM-Microscope

LLM-Microscope is a framework to analyze Large Language Models' internal processes. To facilitate interactive exploration of our analysis methods, we have developed a user-friendly demo system using Gradio, hosted on Hugging Face. This interface allows researchers and practitioners to apply LLM-Microscope's tools to various models and input texts in real-time. The demo system features:

- Model selection: Users can choose from a variety of pre-loaded language models.
- Text input: A text area for entering custom prompts or sentences for analysis.
- Visualization dashboard: Upon submission, the system generates and displays:
  - A heatmap of token-level nonlinearity across all layers
  - A line graph showing average linearity scores per layer
  - A heatmap of layer-wise contribution to final token prediction
  - A heatmap showing the contextualization level of each token
  - Visualization of the logit lens showing the preliminary predictions of the intermediate layers

The interface of our system can be found in the Figure 2.

For example, in Figure 1, one can observe patterns of nonlinearity across layers for a logical reasoning task. Different colors indicate different degrees of nonlinearity, potentially corresponding to key points in the model's reasoning process.

For users requiring more in-depth analysis or wishing to examine models not integrated into the demo, we have published our entire codebase.

3.1 Measuring Token-level Nonlinearity

Following the methodology for quantifying the degree of nonlinearity in token representations across model layers (**?**) , we apply a generalized Procrustes analysis for arbitrary linear transformations. For each pair of adjacent layers l and $l+1$, we compute:

$$A^* = \min_{A \in \mathbb{R}^{d \times d}} |\hat{H}^l A - \hat{H}^{l+1}|_F^2 \qquad (1)$$

where A^* is the optimal linear transformation found during the linearity score computation, $\hat{H}^l$ and $\hat{H}^{l+1}$ are normalized and centered matrices of token embeddings from layers l and $l+1$ respectively, and $|\cdot|_F$ denotes the Frobenius norm. The linear approximation error (nonlinearity score) for each token i at layer l is then calculated as:

$$E_i^l = |A^* h_i^l - h_i^{l+1}|_2 \qquad (2)$$

where h_i^l is the embedding of token i at layer l.

3.2 Assessing Contextual Memory in Token-Level Representations

To quantify the amount of contextual information stored in token-level representations, we propose a simple technique that uses the model's ability to reconstruct prefix information from individual token representations. This approach provides insight into how different tokens encode and preserve context across all layers of the model.

Our method consists of the following steps:

1. We first process an input sequence through the examined language model, collecting hidden states for each token across all layers.
2. We use a trainable linear pooling layer to combine these layer-wise embeddings into a single representation. This pooling layer is followed by a two-layer MLP.
3. The resulting embedding is then used as input to a trainable copy of the original model, which attempts to reconstruct the prefix leading to the chosen token.
4. The described system is trained with a Cross-Entropy loss to reconstruct random text fragments. For training we use the following datasets: TinyStories (**?**), Tiny-Textbooks[4], Tiny-Lessons[5], Tiny-Orca-Textbooks[6], Tiny-Codes[7], textbooks-are-all-you-need-lite[8].
5. We evaluate the effectiveness of this reconstruction by computing the perplexity of the generated prefix compared to the original input.

[4] https://huggingface.co/datasets/nampdn-ai/tiny-textbooks

[5] https://huggingface.co/datasets/nampdn-ai/tiny-lessons

[6] https://huggingface.co/datasets/nampdn-ai/tiny-orca-textbooks

[7] https://huggingface.co/datasets/nampdn-ai/tiny-codes

[8] https://huggingface.co/datasets/SciPhi/textbooks-are-all-you-need-lite

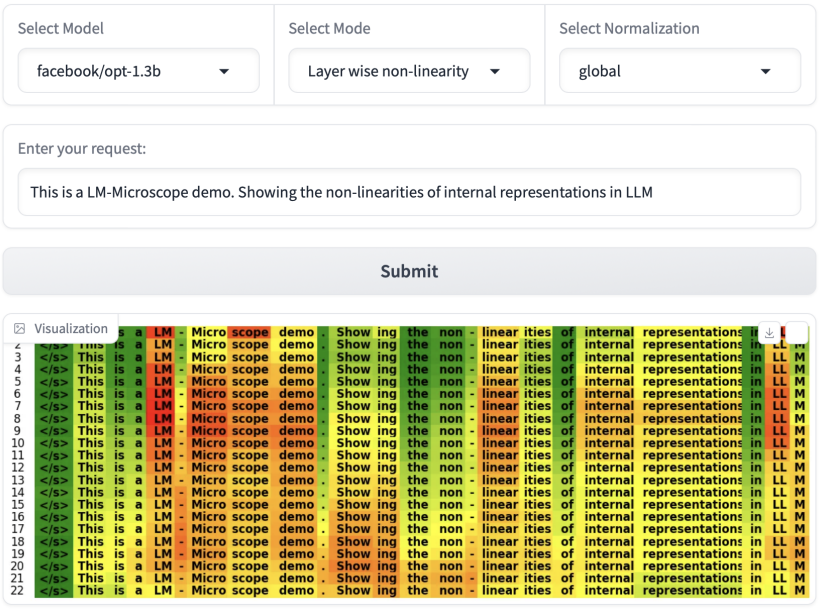

Figure 2: Interface LLM-Microscope demo system.

The full pipeline is depicted in the Figure 3. The CrossEntropy reconstruction loss score serves as our measure of contextualization. A lower loss indicates that the token's representation contains more information about its context, as the model is able to reconstruct the previous text more accurately.

Formally, let h_i^l denote the hidden state of the i-th token at layer l. Our pooling function f and subsequent MLP g can be expressed as:

$$e_i = g(f([h_i^1, h_i^2, ..., h_i^L])) \tag{3}$$

where e_i is the final embedding used for prefix reconstruction.

The contextualization score C_i for token i is then defined as:

$$C_i = -\log P(w_1, ..., w_{i-1}|e_i) \tag{4}$$

where $P(w_1, ..., w_{i-1}|e_i)$ is the probability of the true prefix given the embedding e_i.

This methodology allows us to:

- Identify which tokens retain the most contextual information
- Analyze how contextualization varies for different types of tokens (e.g., content words vs. function words)
- Explore the relationship between contextualization and other properties such as token-level nonlinearity
- Compare contextualization patterns across different model architectures and sizes

3.3 Examining Intermediate Layers Contribution to Token Prediction

To track the evolution of token predictions across model's layers, we apply the language model head to intermediate layer representations. Our approach consists of the following steps:

1. Collect hidden states h_i^l for each token i at each layer l.

2. Apply the language model head to obtain token probabilities:

$$p_i^l = \text{softmax}(\text{LMhead}(h_i^l)) \tag{5}$$

3. Compute prediction error for the next token at each layer:

$$E_i^l = -\log p_i^l[w_{i+1}] \tag{6}$$

where w_{i+1} is the true next token.

This analysis shows how prediction accuracy changes across layers, indicating when the model forms its predictions and how confidence evolves. It highlights cases of early correct predictions compared to those requiring full network depth.

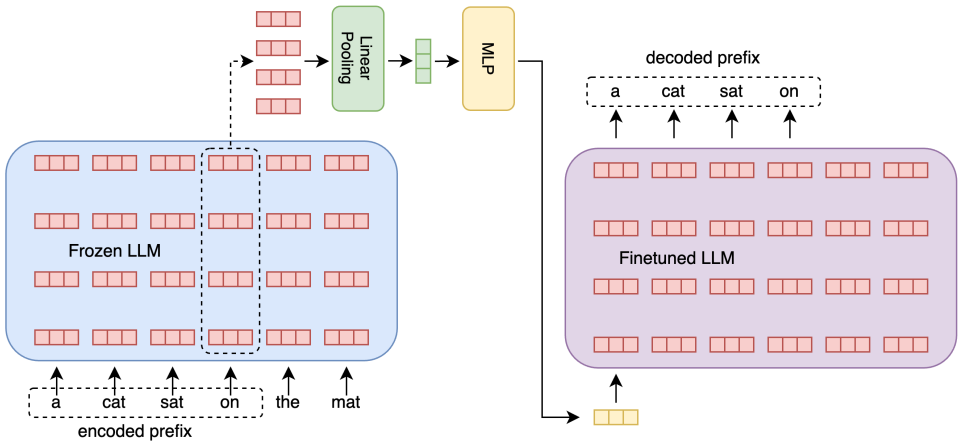

Figure 3: Prefix decoding pipeline as a contextualization assessment.

3.4 Visualizing Intermediate Layer Predictions

In addition to our custom visualization tools, we have implemented the "Logit Lens" technique[9]. This method provides an intuitive way to visualize the evolution of token predictions across model's layers.

The Logit Lens applies the model's output layer (LM head) to the activations of intermediate layers. This process generates probability distributions over the vocabulary at each layer, offering insight into the model's "beliefs" as it processes the input.

The "Logit Lens" suggests that these models primarily "think in predictive space," quickly transforming inputs into predicted outputs and then refining those predictions over the layers. An example of "Logit Lens" output in our framework can be found in the Figure 6. The developed framework also support multimodal LLM.

3.5 Intrinsic Dimension of Representations

To evaluate the complexity and the information content of token representations, we estimate their intrinsic dimensionality using the method proposed by **?**. This approach examines how the volume of an n-dimensional sphere (representing the number of embeddings) scales with dimension d. For each token embedding, we compute:

$$\mu_i = \frac{r_2}{r_1} \tag{7}$$

where r_1 and r_2 are distances to the two nearest neighbors. The intrinsic dimension d is then estimated using:

$$d \approx -\frac{\log(1 - F(\mu))}{\log(\mu)} \tag{8}$$

where $F(\mu)$ is the cumulative distribution function of μ_i.

4 Examples and Observations

4.1 The Most Memory Retentive Tokens

To analyze how different types of tokens retain and encode contextual information, we processed random fragments of Wikipedia articles through our pipeline from Section 3.2, collecting contextualization scores (C) for all tokens while preserving information about the original words before tokenization.

Surprisingly, we found out that the tokens that are easiest to use for context (prefix) reconstruction correspond to what are typically considered the least semantically significant elements of language: determiners, prepositions, and punctuation marks. In contrast, nouns and adjectives proved to be the most challenging tokens to reconstruct the prefix.

This counterintuitive finding suggests that language models may use these seemingly less important words as aggregators of memory or overall meaning.

Across all models examined, including different sizes of OPT, Phi, and Llama model families, determiners (DT) and punctuation consistently emerge as the most contextualized tokens with the lowest average reconstruction loss values C.

On the other hand, nouns (NN, NNS) appear universally among the least contextualized tokens, with significantly higher reconstruction loss val-

[9] https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens

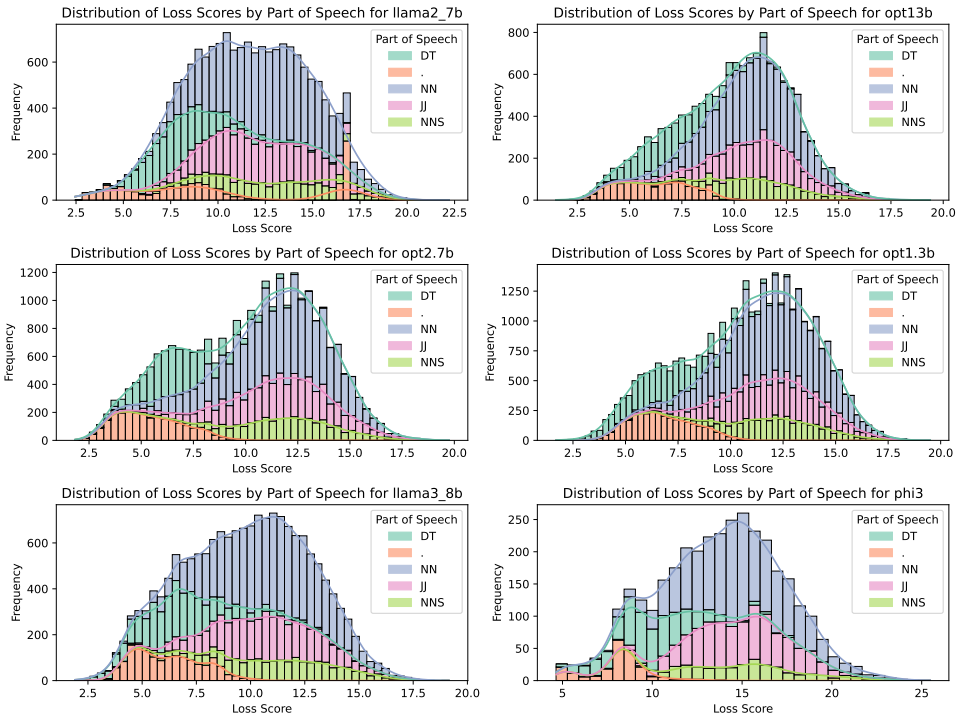

Figure 4: Contextualization score distribution for different parts of speech.

Table 1: Performance on MMLU and BABILong-4k after partial removal...

Model	Original	No Stop Words	No Punctuation	No Stops & Punct	No Articles	GPT-4 Removal
MMLU						
Llama-3.2-3B	0.398	0.347	0.391	0.342	0.386	0.377
Mistral-7B-v0.1	0.423	0.359	0.411	0.350	0.413	0.392
meta-llama-3-8B	0.430	0.365	0.419	0.351	0.415	0.403
Qwen2.5-1.5B	0.362	0.332	0.348	0.322	0.356	0.346
BABILong 4k						
Llama-3.2-3B	0.420	0.334	0.377	0.322	0.386	0.387
Mistral-7B-v0.1	0.373	0.324	0.322	0.314	0.368	0.312
meta-llama-3-8B	0.388	0.331	0.359	0.307	0.389	0.360
Qwen2.5-1.5B	0.366	0.326	0.333	0.322	0.348	0.308

ues C. Detailed histograms can be found in the Figure 4.

4.2 Examining the Impact of Removing "Filler" Tokens

While our earlier analysis focused on identifying which tokens carry the most contextual information, we also investigated how removing seemingly "minor" or "irrelevant" tokens affects LLM performance on tasks requiring domain knowledge or extended context. Instead of discarding highly contextualized tokens, we selectively removed punctuation, stopwords, and articles in two distinct modes: (1) a naive, rule-based removal that targets all such tokens, and (2) a more nuanced approach using GPT-4o.

Benchmarks. We evaluated these removal strategies on two benchmarks:

- **MMLU** (**?**): A widely used multiple-choice benchmark spanning various academic subjects, testing both factual recall and general reasoning. MMLU was evaluated in a zero-shot setting.
- **BABILong-4k** (**?**): A long-context reasoning benchmark combining facts (from the bAbI dataset (**?**)) and large amounts of distractor text (from PG19 (**?**)), where crucial details may be scattered across up to 4k tokens.

Removal Conditions. We examined several removal strategies:

1. **No Stopwords**: Delete common English function words (e.g., *the, an, and*).

2. **No Punctuation**: Remove punctuation marks (commas, periods, quotes, etc.).

3. **No Articles**: Remove only English articles (*a, an, the*).

4. **No Stopwords & Punct**: Remove both stopwords and punctuation.

5. **GPT-4o Removal**: Prompt GPT-4 to remove only those stopwords or punctuation marks that it deems safe to delete without changing the meaning. Below is the exact system prompt used for GPT-4o when removing tokens:

```
system_message = """
    You are an expert in natural language processing.
   Your task is to remove stop words and punctuation from the user's text
    only when their removal does not alter the meaning of the text.
    Stop words are common words that add little meaning to the text
    (e.g., 'and', 'the', 'in', 'on', 'at', etc.).
   If removing all stop words and punctuation would change the meaning,
    remove only those that contribute the least to the meaning
    while preserving readability.
    Do not rephrase or change the order of words.
    Return only the modified text, without extra commentary.
"""
```

Table 1 summarizes the accuracy of several LLMs on MMLU and BABILong-4k under these token-removal schemes. Notably, the deletion of punctuation and basic function words yields a consistent drop in performance. On BABILong-4k, where capturing subtle facts within a large context is crucial, the accuracy losses are especially pronounced.

These results are in line with our earlier findings: LLMs often store key contextual signals in "filler" tokens (stopwords, punctuation) that might be seemed unimportant for semantic meaning. Even a carefully controlled removal policy via GPT-4o shows that seemingly trivial tokens play an outsized role in preserving the chain of context — particularly when handling long sequences or academic questions.

4.3 Correlation Between Nonlinearity and Context Memory

We observed a significant correlation between layer-averaged linearity and contextualization scores for individual tokens. Tokens with high contextualization tend to correspond to the most linear transformations across layers. Figure 5 illustrates this relationship for the OPT-6.7B model, showing the distribution of linearity versus contextualization scores. This correlation is consistent

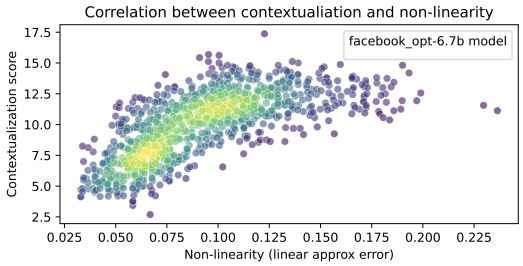

Figure 5: The distribution of Cotextuality C and non-linearity scores for random fragments of text on English Wikipedia articles.

Table 2: Correlation coefficient.

Model	Corr
Opt-6.7b	0.482
Opt-1.3b	0.406
Opt-13b	0.359
Opt-2.7b	0.401
Gemma-2-9b	0.561
Gemma-2-2b	0.515
Llama-3.2-3B	0.367
Llama-3 8B	0.050
Llama-3 8B Instruct	0.328
Llama-2 7Bfp16	0.239
Phi-3-mini 128k instruct	0.410
Qwen2.5 1.5B	0.199
Mistral-7B-v0.1	0.152

*p-value less than 0.05 in all measurements

across different model architectures and sizes, as evidenced by the Pearson correlation coefficients presented in Table 2.

These findings suggest a potential link between the model's ability to retain contextual information and the linearity of its internal representations.

4.4 Multilingual Reasoning

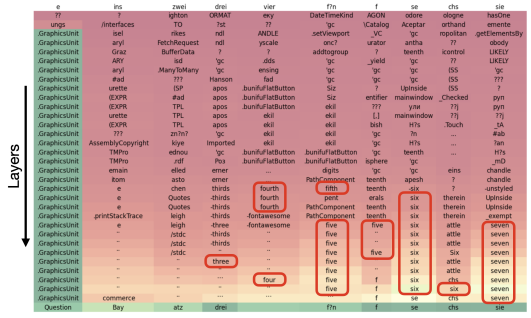

Figure 6: Logit lens visualisation for Llama3-8B. Input text in German: "eins zwei drei vier fünf sechs sieben," which translates into English: "one two three four five six seven."

Using the "Logit Lens" technique, we studied how language models process non-English input.

Our analysis shows that intermediate layer representations predominantly correspond to English tokens, even when the input is in another language. Figure 6 demonstrates this behavior. The heatmap displays token predictions across layers, with each row representing a layer and each column a token position. The color intensity indicates the model's confidence in its top-1 token prediction.

Correct non-English tokens corresponding to the translated version of the input gradually emerge in later layers. These observation suggests that the models may perform implicit translation into English before generating the final output.

5 Conclusion

In this work, we introduced methods to quantify how Large Language Models (LLMs) encode and store contextual information, revealing the surprising importance of seemingly "minor" tokens — such as determiners, punctuation, and stopwords — in maintaining coherence and context. Our analysis showed a strong correlation between a token's contextualization level and how linearly one layer's representation can be mapped onto the next, suggesting a close relationship between model architecture and the retention of contextual cues.

Through empirical evaluations on MMLU and BABILong 4k, we demonstrated that removing high-context tokens — even if they appear trivial — consistently degrades performance. Notably, this effect remains even when a strong language model (GPT-4o) is used to selectively remove only those tokens deemed least relevant. These findings highlight that "filler" tokens can carry critical context, underscoring the need for more refined interpretability approaches.

To facilitate further research in this area, we presented **LLM-Microscope**, an open-source toolkit that offers: Token-level nonlinearity analysis, Methods for assessing contextual memory, Visualizations of intermediate layer contributions through an adapted Logit Lens, Intrinsic dimensionality measurements of internal representations.

6 Limitations

- LM-head application: Using a pre-trained LM-head on intermediate embeddings without fine-tuning may not accurately reflect the actual functionality of these layers.
- Contextual memory assessment: The adapter-based method's accuracy may be influenced by the adapter's architecture, training data, and optimization process.
- Generalizability: The results may not be equally applicable to all model architectures, sizes, or training paradigms.

7 Ethical Statement

This research aims to improve LLM transparency and interpretability, potentially improving AI safety and reliability. Our tools are designed for analysis only and cannot modify model behavior. We acknowledge the dual-use potential of interpretability research and advocate for responsible use. All experiments were conducted on publicly available pre-trained models without access to personal data or its generation.

This work advances our understanding of LLM internals, contributing to the development of more transparent and reliable natural language processing systems.

References