Proceedings of the International Conference on
Formal Methods and Foundations of Artificial Intelligence
Eszterházy Károly Catholic University
Eger, Hungary, June 5–7, 2025
pp. 233–242
DOI: 10.17048/fmfai.2025.233

Under the hood
An inside look at PULI models*

Zijian Győző Yang[a], **Lili Anna Stajer**[b], **Gergely Lukács**[b]

[a]ELTE Research Centre for Linguistics
yang.zijian.gyozo@nytud.elte.hu

[b]Pázmány Péter Catholic University
Faculty of Information Technology and Bionics
lukacs@itk.ppke.hu

Abstract. Understanding the internal structure and behavior of large language models remains a key challenge in natural language processing. In this work, we present a comprehensive analysis of the PULI family of Hungarian generative large language models. Our study combines static analysis of model parameters with dynamic visualization of model behavior during inference. The static analysis reveals patterns in parameter distributions and dimensionality across layers, offering insight into how different layers specialize. The dynamic analysis integrates an adapted version of BertViz into a web-based interface that enables interactive exploration of attention mechanisms for arbitrary prompts and generated responses. This dual approach advances interpretability and facilitates further research on the internal mechanics of transformer models tailored for low-resource languages like Hungarian.

Keywords: PULI models, large language models, transformers visualization, attention analysis, BertViz, principal component analysis, cumulative explained variance

AMS Subject Classification: 68T07, 68T30, 68T50, 62R07, 62R40

1. Motivation

The transformer architecture and large language models (LLMs) led to a new era in natural language processing (NLP) and, more broadly, in computer science.

*The study was funded by the National Research, the Development and Innovation Office in Hungary (RRF-2.3.1-21-2022-00004).

Although their high-level designs are well documented and such models can be trained – given sufficient data, computational resources, and expertise – the internal workings of these models remain poorly understood. Specifically, the structure and geometry of billions to trillions of parameters, organized in large matrices, present a significant challenge to interpretation.

A deeper understanding of these internal mechanisms could lead to improved overall performance, enhanced capabilities in specialized tasks (e.g., disambiguation, humor detection, or handling harmful speech), and potential simplifications of model architecture. Each of these areas represents current limitations or open challenges in existing models.

For Hungarian, the PULI family [15, 16] represents the state-of-the-art in generative large language models, including both GPT-NeoX [3] and LLaMA-based [5, 12] architectures.

In our research, we performed both static and dynamic analyses of the internal parameters of the model. The static analysis focused on examining various properties and features of the trained models. In the dynamic analysis, we enabled visualization of the model's internal representations for arbitrary input text by adopting and integrating the BertViz application [13] into our demonstration platform[1].

2. Related work

A growing body of research has focused on analyzing the internal parameter values of deep neural networks and transformer-based models. It has long been recognized that deep neural networks are capable of acquiring and encoding aspects of human semantic knowledge [11]. Investigations of the BERT model have shown that distinct subspaces within the parameter space correspond to syntactic and semantic information [10]. Additionally, different senses of a word can be distinguished and separated in this space. Further studies have uncovered links between vector geometries and syntactic structures such as parse trees [6]. Recent research has also demonstrated that attributes like textual toxicity can be identified by analyzing internal parameters [8]. Regarding training dynamics, it has been found that low-dimensional structures within the parameter space are critical for enabling efficient optimization and successful model training [9]. Such findings lay the foundation for developing improved, faster, and more resource-efficient learning strategies [2].

Multilingual BERT models, when analyzed through morphosyntactic probing, have yielded further insights – for instance, indicating that preceding context often contains more semantically relevant information than the following context [1]. Model compression, particularly through quantization, is another active research area with significant practical implications [4].

In parallel with analytical approaches, there have been efforts to improve the interpretability of semantic and contextual representations in LLMs during infer-

[1] https://juniper.nytud.hu/demo/visualizer

ence. Tools such as ExBERT, a visualization framework for exploring learned representations in transformer models [7], and BertViz, a multiscale visualization tool applicable to any transformer architecture, have proven useful in this regard. BertViz has been employed, for example, to detect bias and trace specific behaviors back to particular model components [13, 14].

3. Static analysis

3.1. PULI LlumiX 32K model and its parameters

The static analysis in this study was conducted on the PULI LlumiX 32K model [15], which is based on LLaMA-2-7B-32K[2] variant of the open-source LLaMA (Large Language Model Meta AI) 2 family [12] introduced by Meta[3] in 2023. LLaMA models are decoder-only architectures, meaning they consist solely of transformer decoder layers. The core architecture comprises multiple identical layers, each containing a feed-forward neural network (FFN), layer normalization, and self-attention blocks. Input data is processed through an embedding layer and positional encoding before being passed through the stacked layers.

The self-attention mechanism maps a query and a set of key–value pairs to an output, enabling the model to capture dependencies between tokens regardless of their position in the sequence. The main components and parameters involved are as follows:

- *Query* (Q): Represents the current token being processed and is used to compute attention scores by comparing it to all other tokens' key vectors.
- *Key* (K): Associated with each token in the sequence and used to determine the relevance of other tokens to the current one.
- *Value* (V): Also associated with each token, and contains the information that contributes to the final weighted output.

The result of the self-attention mechanism is a weighted sum of the value vectors, where weights are derived from the similarity between queries and keys. Modern transformer models use multi-head attention, which involves multiple parallel self-attention mechanisms, each with its own set of learned parameters. Dedicated weight matrices are used to compute the Q, K, and V vectors from the input representations.

The LLaMA-2-7B-32K model consists of 32 transformer layers and approximately 7 billion parameters. A detailed breakdown of the model's architecture, including the dimensionality of parameter matrices and the total parameter count, is provided in Table 1.

[2] https://huggingface.co/togethercomputer/LLaMA-2-7B-32K
[3] https://www.meta.ai

Table 1. Number of Parameters in LLaMA-2-7B-32K.

Description	Matrix size	Count	Parameter count
embed_token_weight: maps each token onto the d_model	(32000,4096)	1	131 072 000
input_layernorm: each layer input normalized	(4096,1)	32	131 072
self_attn_k: multi-attention head W_K matrix	(4096,4096)	32	536 870 912
self_attn_q: multi-attention head W_Q matrix	(4096,4096)	32	536 870 912
self_attn_v: multi-attention head W_V matrix	(4096,4096)	32	536 870 912
self_attn_o: multi-attention head W_O matrix	(4096,4096)	32	536 870 912
post_attention_layernorm: each multi-head attention output normalized	(4096, 1)	32	131 072
mlp.down_proj: FNN weights	(4096,11008)	32	1 442 840 576
mlp.gate_proj: FNN gate weights	(11008,4096)	32	1 442 840 576
mlp.up_proj: FNN weights	(11008,4096)	32	1 442 840 576
norm: normalizing function for last layer output	(4096,1)	1	4096
lm_head: maps d_model back onto the vocabulary space	(32000,4096)	1	131 072 000
TOTAL			6 738 415 616

3.2. Analysis and results

Distinct patterns were observed in the model's parameters. In the feedforward network (FNN), the standard deviation of the down, gate, and up projection weights progressively increases in the upper layers (Figure 1). This trend is further supported by a decrease in the 25 percentile and an increase in the 75 percentile values (Figure 2). Notably, the first and last layers exhibit substantially larger changes compared to the intermediate layers.

In the self-attention blocks, the standard deviations of the key (k) and query (q) weights decrease from the lower to the upper layers, while the value (v) and output (o) weights show a similar downward trend. Again, notable exceptions to these

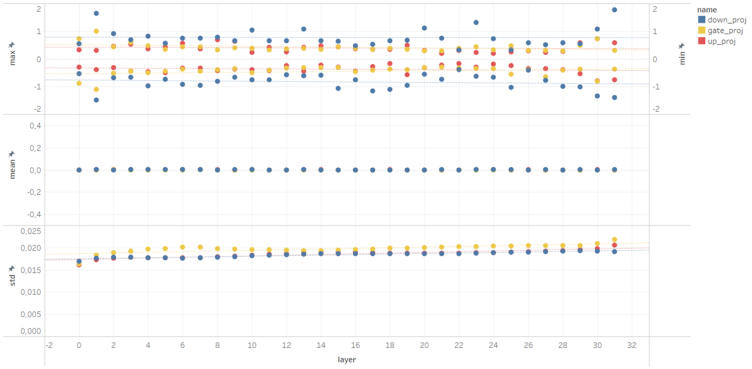


Figure 1. Plot of min-max, mean and standard deviation value of FNN components.

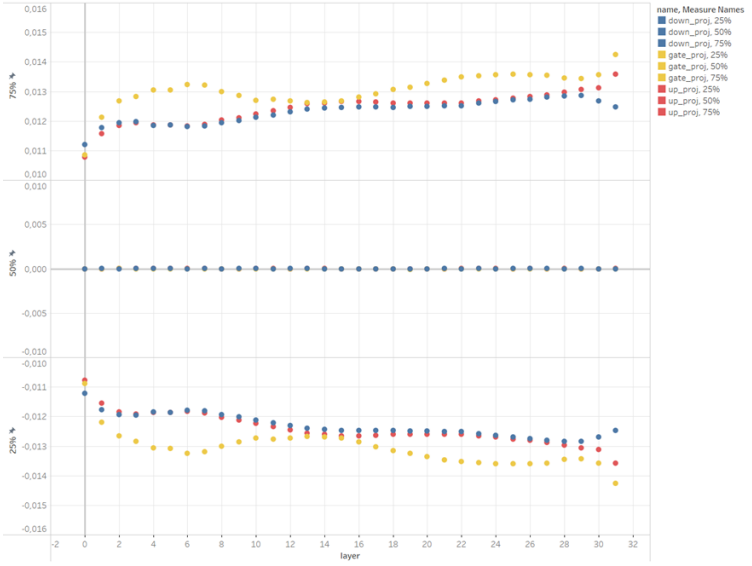


Figure 2. Plot of 25, 50 and 75 percentile values of FNN components.

trends appear in the first and last layers. Principal component analysis on these weights revealed that, in general, the cumulative explained variance indicates that dimensionality cannot be significantly reduced without information loss. However, in a few specific cases – particularly for the k and q weights, and to a lesser extent the v and o weights – early layers (especially the first three) exhibit high cumulative explained variance, approaching 1, with a substantially smaller number of dimensions than in later layers (Figure 3, Figure 4).

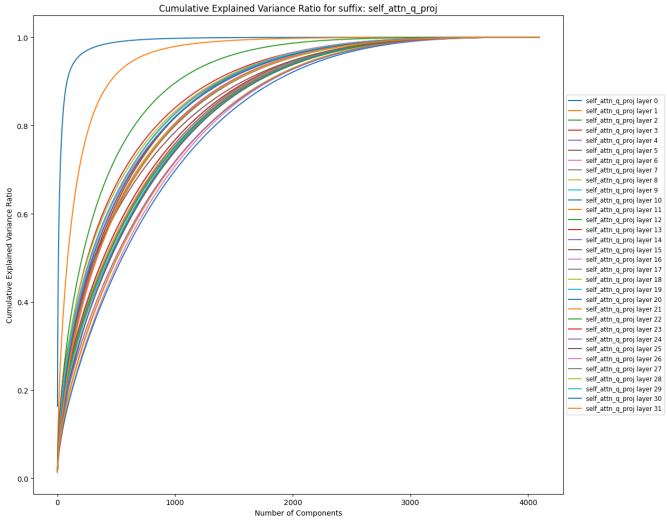


Figure 3. Cumulative explained variance of query (q) matrices for self-attention layers.

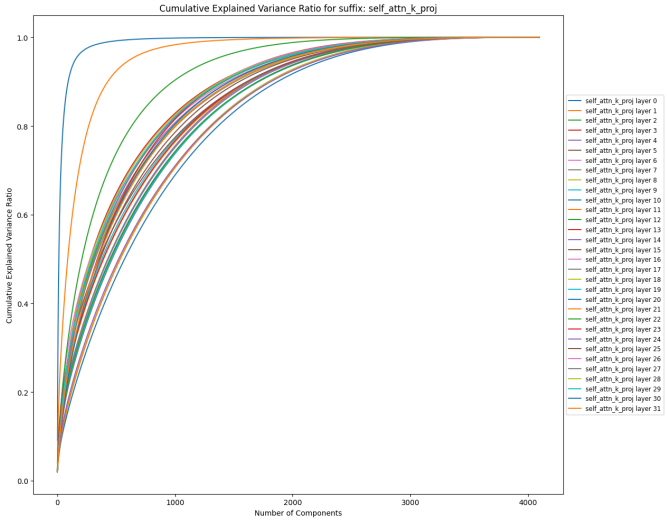


Figure 4. Cumulative explained variance of key (k) matrices for self-attention layers.

4. Dynamic analysis

For Dynamic analysis, we integrated the BertViz tool [13] into our demo site. In our demo site[4], we split the BertViz output into frontend and backend components and integrated them into the corresponding sections of our site. While the original BertVis frontend code remained unchanged, we applied several modifications to the backend. First, we added a text generation module and then merged the newly generated text with the original input prompt. This functionality allows us to observe the relationships between the prompt and its response.

Figure 5 presents the architecture of our dynamic analysis demo site. This architecture diagram illustrates a system designed to interface with a LLM through a frontend-backend pipeline. On the frontend, users interact with the system via a Demo interface, where they input prompts. These prompts are sent to the LLM hosted in our backend, which generates corresponding model responses and returns them to the frontend for display. Simultaneously, a weight extraction module accesses internal data (such as attention weights) from the LLM, processes it, and forwards the resulting weights to BertViz, a frontend visualization tool that allows users to explore the model's inner workings. This design separates user interaction, model computation, and interpretability, enabling a clear and interactive workflow to use and understand the behavior of the LLM.

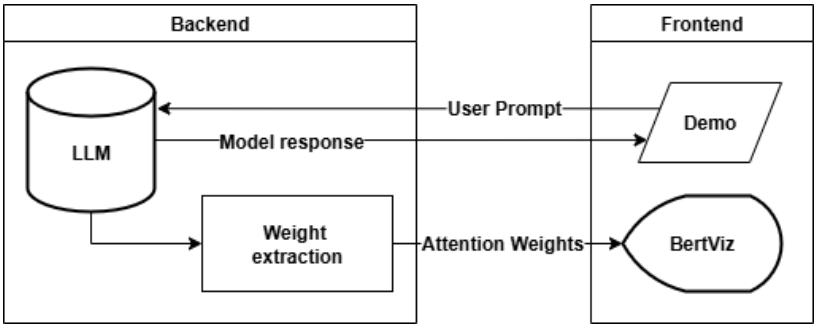
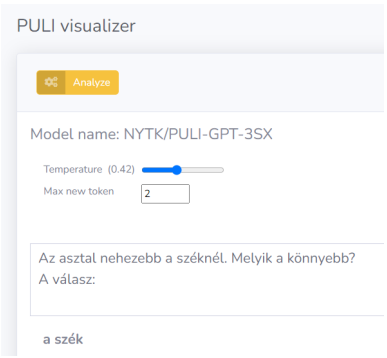


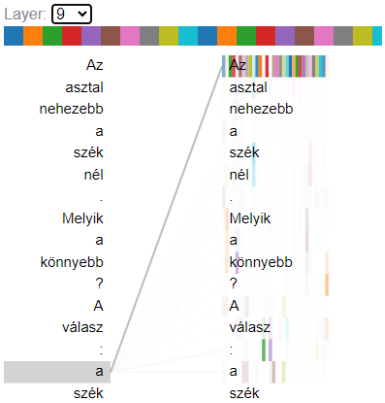
Figure 5. Architecture of the demo site.

In Figure 6, we show the integrated BertViz visualization along with the dynamic prompt-response analysis. In our demo site, an input prompt can be provided, and then the response will be generated (see Figure 6a). In this example, the prompt is: *The table is heavier than the chair. Which is the lighter one? The answer:*. The response: *the chair*. In Figure 6b, we visualize the relationships between the prompt and its response. In this example, we observe the weights of layer 9, where we can see that, in the case of 'a' (*the*), the attention is focused more on the word 'szék' (*chair*) than on the word 'asztal' (*table*).

[4] https://juniper.nytud.hu/demo/visualizer



(a) A screenshot of our demo site.



(b) Illustrating the relationship between the input prompt and the generated response.

Figure 6. The PULI visualizer demo.

5. Conclusion

In this paper, we investigated the internal mechanisms of the PULI large language models using a two-pronged approach: static parameter analysis and dynamic behavior visualization. Our static examination highlighted systematic trends in weight distributions and dimensionality across layers, suggesting layer-specific roles in the model's computation. The dynamic component extended the BertViz framework, allowing users to explore the relationship between input prompts and model responses in real time. These findings contribute to the broader goal of demystifying LLMs and open avenues for improving model transparency, fine-tuning strategies, and error diagnosis, particularly in the context of Hungarian language technologies. Future work may focus on extending these methods to multilingual settings or applying similar techniques to fine-tuning and alignment tasks.

In the future, we plan to extend our experiments to other PULI models and implement a model selection module that allows users to interactively switch between different PULI architectures, including encoder-only, decoder-only, and encoder-decoder models. This would enable comparative analysis, generalization of results, and adaptive usage based on specific task requirements. In addition, combining advanced statistical methods with visualizations appears promising for dynamic analysis.

References

[1] J. Acs, E. Hamerlik, R. Schwartz, N. A. Smith, A. Kornai: *Morphosyntactic probing of multilingual BERT models*, Natural Language Engineering 30.4 (2024), pp. 753–792, doi: 10.1017/S1351324923000190.

[2] G. Berend: *Masked Latent Semantic Modeling: an Efficient Pre-training Alternative to Masked Language Modeling*, in: Findings of the Association for Computational Linguistics: ACL 2023, ed. by A. Rogers, J. Boyd-Graber, N. Okazaki, Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 13949–13962, doi: 10.18653/v1/2023.findings-acl.876, url: https://aclanthology.org/2023.findings-acl.876/.

[3] S. Black, S. Biderman, E. Hallahan, Q. Anthony, L. Gao, L. Golding, H. He, C. Leahy, K. McDonell, J. Phang, M. Pieler, U. S. Prashanth, S. Purohit, L. Reynolds, J. Tow, B. Wang, S. Weinbach: *GPT-NeoX-20B: An Open-Source Autoregressive Language Model*, in: Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models, ed. by A. Fan, S. Ilic, T. Wolf, M. Gallé, virtual+Dublin: Association for Computational Linguistics, May 2022, pp. 95–136, doi: 10.18653/v1/2022.bigscience-1.9, url: https://aclanthology.org/2022.bigscience-1.9/.

[4] T. Dettmers, R. A. Svirschevski, V. Egiazarian, D. Kuznedelev, E. Frantar, S. Ashkboos, A. Borzunov, T. Hoefler, D. Alistarh: *SpQR: A Sparse-Quantized Representation for Near-Lossless LLM Weight Compression*, in: The Twelfth International Conference on Learning Representations, 2024, url: https://openreview.net/forum?id=Q1u25ahSuy.

[5] A. Grattafiori et al.: *The Llama 3 Herd of Models*, 2024, arXiv: 2407.21783 [cs.AI], url: https://arxiv.org/abs/2407.21783.

[6] J. Hewitt, C. D. Manning: *A Structural Probe for Finding Syntax in Word Representations*, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), ed. by J. Burstein, C. Doran, T. Solorio, Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4129–4138, doi: 10.18653/v1/N19-1419, url: https://aclanthology.org/N19-1419/.

[7] B. Hoover, H. Strobelt, S. Gehrmann: *exBERT: A Visual Analysis Tool to Explore Learned Representations in Transformer Models*, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ed. by A. Celikyilmaz, T.-H. Wen, Online: Association for Computational Linguistics, July 2020, pp. 187–196, doi: 10.18653/v1/2020.acl-demos.22, url: https://aclanthology.org/2020.acl-demos.22/.

[8] A. Lee, X. Bai, I. Pres, M. Wattenberg, J. K. Kummerfeld, R. Mihalcea: *A Mechanistic Understanding of Alignment Algorithms: A Case Study on DPO and Toxicity*, in: Proceedings of the 41st International Conference on Machine Learning, ed. by R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, F. Berkenkamp, vol. 235, Proceedings of Machine Learning Research, PMLR, 21–27 Jul 2024, pp. 26361–26378, url: https://proceedings.mlr.press/v235/lee24a.html.

[9] J. Mao, I. Griniasty, H. K. Teoh, R. Ramesh, R. Yang, M. K. Transtrum, J. P. Sethna, P. Chaudhari: *The training process of many deep networks explores the same low-dimensional manifold*, Proceedings of the National Academy of Sciences 121.12 (2024), e2310002121, doi: 10.1073/pnas.2310002121.

[10] E. Reif, A. Yuan, M. Wattenberg, F. B. Viégas, A. Coenen, A. Pearce, B. Kim: *Visualizing and Measuring the Geometry of BERT*, in: Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, ed. by H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, R. Garnett, 2019, pp. 8592–8600, url: https://proceedings.neurips.cc/paper/2019/hash/159c1ffe5b61b41b3c4d8f4c2150f6c4-Abstract.html.

[11] A. M. Saxe, J. L. McClelland, S. Ganguli: *A mathematical theory of semantic development in deep neural networks*, Proceedings of the National Academy of Sciences 116.23 (2019), pp. 11537–11546, doi: 10.1073/pnas.1820226116.

[12] H. Touvron et al.: *Llama 2: Open Foundation and Fine-Tuned Chat Models* (2023), arXiv: 2307.09288 [cs.CL].

[13] J. Vig: *A Multiscale Visualization of Attention in the Transformer Model*, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ed. by M. R. Costa-jussà, E. Alfonseca, Florence, Italy: Association for Computational Linguistics, July 2019, pp. 37–42, doi: 10.18653/v1/P19-3007, url: https://aclanthology.org/P19-3007/.

[14] J. Vig, Y. Belinkov: *Analyzing the Structure of Attention in a Transformer Language Model*, in: Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, ed. by T. Linzen, G. Chrupała, Y. Belinkov, D. Hupkes, Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 63–76, doi: 10.18653/v1/W19-4808, url: https://aclanthology.org/W19-4808/.

[15] Z. G. Yang, R. Dodé, G. Ferenczi, P. Hatvani, E. Héja, G. Madarász, N. Ligeti-Nagy, B. Sárossy, Z. Szaniszló, T. Váradi, T. Verebélyi, G. Prószéky: *The First Instruct-Following Large Language Models for Hungarian*, in: 2024 IEEE 3rd Conference on Information Technology and Data Science (CITDS) Proceedings, Debrecen, Hungary: University of Debrecen, 2024, pp. 247–252, isbn: 9798350387889.

[16] Z. G. Yang, L. J. Laki, T. Váradi, G. Prószéky: *Mono- and multilingual GPT-3 models for Hungarian*, in: Text, Speech, and Dialogue, Lecture Notes in Computer Science, Plzeň, Czech Republic: Springer Nature Switzerland, 2023, pp. 94–104, isbn: 978-3-031-40498-6.