ANIMATEDLLM: Explaining LLMs with Interactive Visualizations

Zdeněk Kasner and **Ondřej Dušek**
Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
{kasner,odusek}@ufal.mff.cuni.cz

Abstract

Large language models (LLMs) are becoming central to natural language processing education, yet materials showing their mechanics are sparse. We present ANIMATEDLLM, an interactive web application that provides step-by-step visualizations of a Transformer language model. ANIMATEDLLM runs entirely in the browser, using pre-computed traces of open LLMs applied on manually curated inputs. The application is available at https://animatedllm.github.io, both as a teaching aid and for self-educational purposes.[1]

1 Introduction

Natural language processing (NLP) researchers and teachers often need to explain the inner workings of large language models (LLMs) to non-experts. However, describing certain concepts – for example, how the words are tokenized or how the model consumes its own output to generate coherent text – is difficult without visual aids.

Existing visualization tools fall into two categories: static educational materials, which are useful for self-study but lack interactivity (Alammar, 2018, 2019; 3Blue1Brown, 2024), or tools for understanding the details of the Transformer architecture, which are directed towards technical audiences (Bycroft, 2023; Cho et al., 2025; Brasoveanu et al., 2024).

To bridge the gap between the existing tools, we present ANIMATEDLLM: a client-side web application designed for explaining the mechanics of LLMs to non-technical audiences. The goal of ANIMATEDLLM is to provide the right level of abstraction between the two extremes – the complex matrix operations driving the Transformer architecture on the one hand, and the clean but opaque interface of AI assistants on the other. Our learning
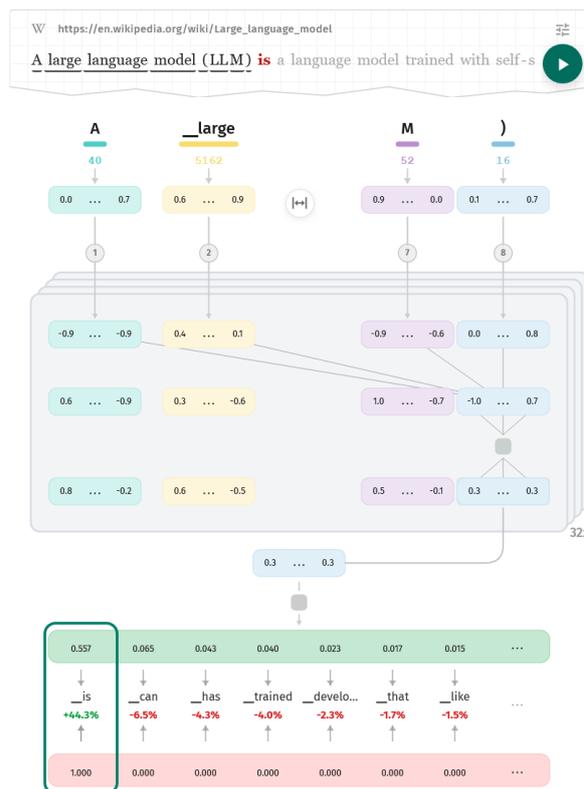

Figure 1: Detailed visualization explaining how the Transformer model is pre-trained on data. Individual components show how the tokens get processed throughout the model.

objective is demystifying the LLM architecture and providing a high-level understanding of their inner workings: tokenization, embeddings, attention, autoregressive generation and next-word prediction training.

2 AnimatedLLM

ANIMATEDLLM is a static React[2] web application that can be hosted on any standard web server.

Data Our goal is to show the inner workings of LLMs using realistic, pedagogically interest-

---

[1]The source code is available under the MIT license at https://github.com/kasnerz/animated-llm.
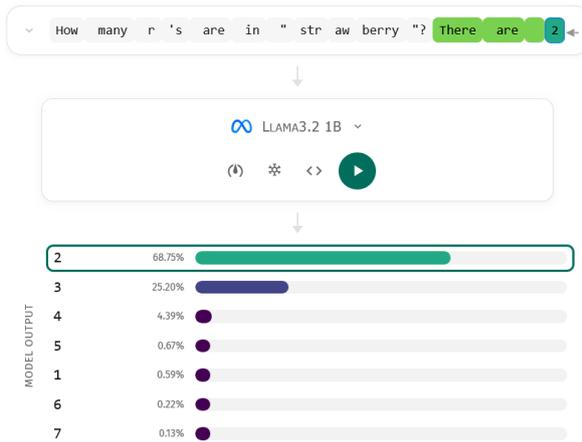
[2]https://react.dev

Figure 2: Simplified text generation visualization showing the autoregressive decoding process with probability distributions over candidate tokens.

ing examples. To that end, we pre-computed model traces, i.e., full model outputs for preselected prompts, including tokenization and output probability distributions, on a series of manually-selected prompts. The traces are serialized into JSON files so that the web application can run entirely in the client's browser. We also provide Python scripts for computing traces with custom models and prompts.

Models We precomputed traces for multiple models from Huggingface Transformers.[3] We selected the models to cover a variety of providers, model sizes, and pre-training objectives, which results in varied outputs: Olmo 3 (Allen Institute for AI, 2025), Aya Expanse (Dang et al., 2024), Llama 3.2 (Dubey et al., 2024), Qwen 3 (Yang et al., 2025), and GPT-2 (Radford et al., 2019).

Views ANIMATEDLLM currently supports four views, each designed for a specific learning objective:

1. **Text generation – simple view** (see Figure 2) helps to explain how the next token is selected by demonstrating the principles of autoregressive decoding.

2. **Text generation – detailed view** (see Figure 5) shows how the Transformer gradually transforms the input embeddings into the output probabilistic distribution.

3. **Pre-training – simple view** (see Figure 4) helps to explain the training objective: how

---

the model's prediction is compared against the target token.

4. **Pre-training – detailed view** (see Figure 1) relates the Transformer architecture to the training process and helping to understand the principles of backpropagation.

Each animation is accompanied by explanatory labels and tooltips. The tool is designed easily extensible by re-using React components and precomputed traces. In the future, we plan to implement more views (showing, e.g., post-training approaches or BPE tokenizer training).

User Interface The web interface follows basic design principles, with the aim of being clean but user-friendly (Krug, 2014; Williams, 2015). In each view, the user selects an input and starts the animation showing a specific LLM aspect. The user can play, pause, and step through the animation. Keyboard shortcuts enable advanced control (e.g., arrows to step, N to skip to processing the next token, G to jump to the end of the process). The application can be viewed both on desktop and mobile devices.

Languages ANIMATEDLLM is built with multi-linguality in mind. The interface is localized into multiple languages, and we provide inputs (along with the associated traces) localized for the language of the interface. Internationalization makes the app more accessible for non-English speakers and opens the possibility to show differences between processing individual languages, for example with respect to tokenization (Petrov et al., 2023). Our application currently supports English, Czech, French, Ukrainian, and Chinese. We are open to community contributions towards translations and prompts for new languages.

3 Applications

ANIMATEDLLM primarily targets non-expert audience that seeks to learn the technical working of LLMs. For example, teachers of high-school computer science or university-level NLP courses can use the tool to provide students with hands-on experience in understanding model behavior. The tool is also available online, so anyone interested in understanding how language models work can explore the visualizations at their own discretion.

We tested preliminary versions of ANIMATEDLLM at outreach events aimed at high school

students with diverse technical backgrounds, including the faculty Open Day[4] and Czech AI Days.[5] The feedback on the app was generally positive and helped us to ground the discussions with the students. We are currently planning to introduce the application to students in pedagogy programs for high-school computer science education, as well as use it in our own teaching of computer science courses on LLMs and chatbots.

4 Related Work

Transformer Explainer (Cho et al., 2025) is an in-depth interactive visualization of the next-token prediction in the GPT-2 model (Radford et al., 2019). Similarly, LLM Visualization (Bycroft, 2023) provides an interactive walkthrough through individual matrix multiplications inside the Transformer decoder architecture. Build An LLM (Gupta, 2026) is an interactive tool that provides a step-by-step walkthrough for training own LLM locally. Many other visualization tools exist in the LLM interpretability field (Brasoveanu et al., 2024). All of these tools target researchers or more knowledgeable audiences. For lay users, understanding LLMs is facilitated by online blogposts (Alammar, 2018, 2019) and videos (3Blue1Brown, 2024).

Acknowledgments

References

3Blue1Brown. 2024. But what is a GPT? Visual intro to transformers. https://youtu.be/wjZofJX0v4M.

Jay Alammar. 2018. The Illustrated Transformer. http://jalammar.github.io/illustrated-transformer/.

Jay Alammar. 2019. The Illustrated GPT-2. https://jalammar.github.io/illustrated-gpt2/.

Allen Institute for AI. 2025. OLMo 3 Technical Report. Technical report, Allen Institute for AI.

Adrian MP Brasoveanu, Arno Scharl, Lyndon JB Nixon, and Răzvan Andonie. 2024. Visualizing large language models: A brief survey. In 2024 28th International Conference Information Visualisation (IV), pages 236–245. IEEE.

Brendan Bycroft. 2023. LLM Visualization. https://bbycroft.net/llm.

Aeree Cho, Grace C. Kim, Alexander Karpekov, Alec Helbling, Zijie J. Wang, Seongmin Lee, Benjamin Hoover, and Duen Horng (Polo) Chau. 2025. Transformer Explainer: Interactive Learning of Text-generative Models. In AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, pages 29625–29627, Philadelphia.

John Dang, Shivalika Singh, Daniel D'souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, Sandra Kublik, Meor Amer, Viraat Aryabumi, Jon Ander Campos, Yi Chern Tan, Tom Kocmi, Florian Strub, Nathan Grinsztajn, Yannis Flet-Berliac, and 26 others. 2024. Aya Expanse: Combining Research Breakthroughs for a New Multilingual Frontier. CoRR, abs/2412.04261.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Keshwam, Aaron Faulkner, Aditya Ahuja, Aditya Bharadwaj, Ahmad Al-Dahle, Albert Wang, Amaranath Singh, and 1 others. 2024. The Llama 3 Herd of Models. arXiv preprint arXiv:2407.21783. Foundational report for the Llama 3 family, covering Llama 3.1 and 3.2 architectures.

Girish Gupta. 2026. Build An LLM. https://buildanllm.com.

Steve Krug. 2014. Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability, 3rd edition. New Riders, San Francisco, California.

Aleksandar Petrov, Emanuele La Malfa, Philip H. S. Torr, and Adel Bibi. 2023. Language Model Tokenizers Introduce Unfairness Between Languages. In Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9.

Robin Williams. 2015. The Non-Designer's Design Book, 4th edition. Peachpit Press, San Francisco, California.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 40 others. 2025. Qwen3 Technical Report. CoRR, abs/2505.09388.

---

[4] https://www.mff.cuni.cz/en/admissions/open-days/2025

[5] https://www.dny.ai/event/ai-for-talents-i

A   App Interface

Additional screenshots of the application interface are shown in Figures 3 to 6:

- Figure 3: The main page providing access to different visualization views.

- Figure 4: Simplified pre-training view.

- Figure 5: Detailed visualization of the decoding process.

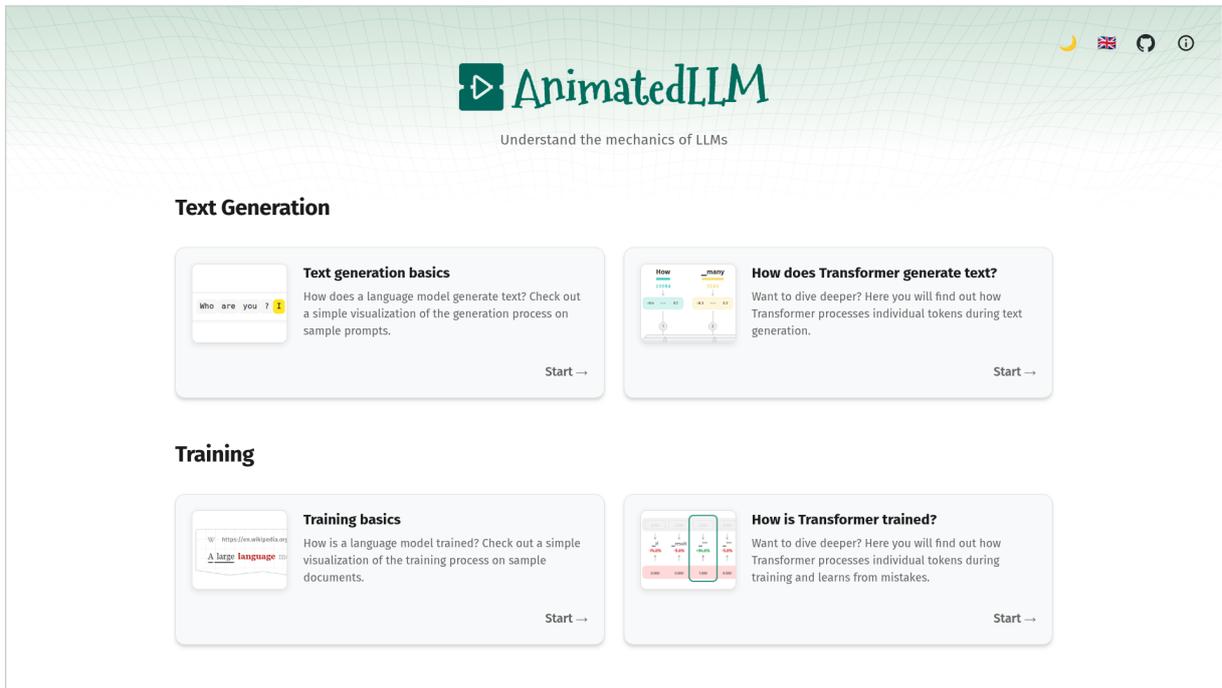- Figure 6: Multilingual interface support.

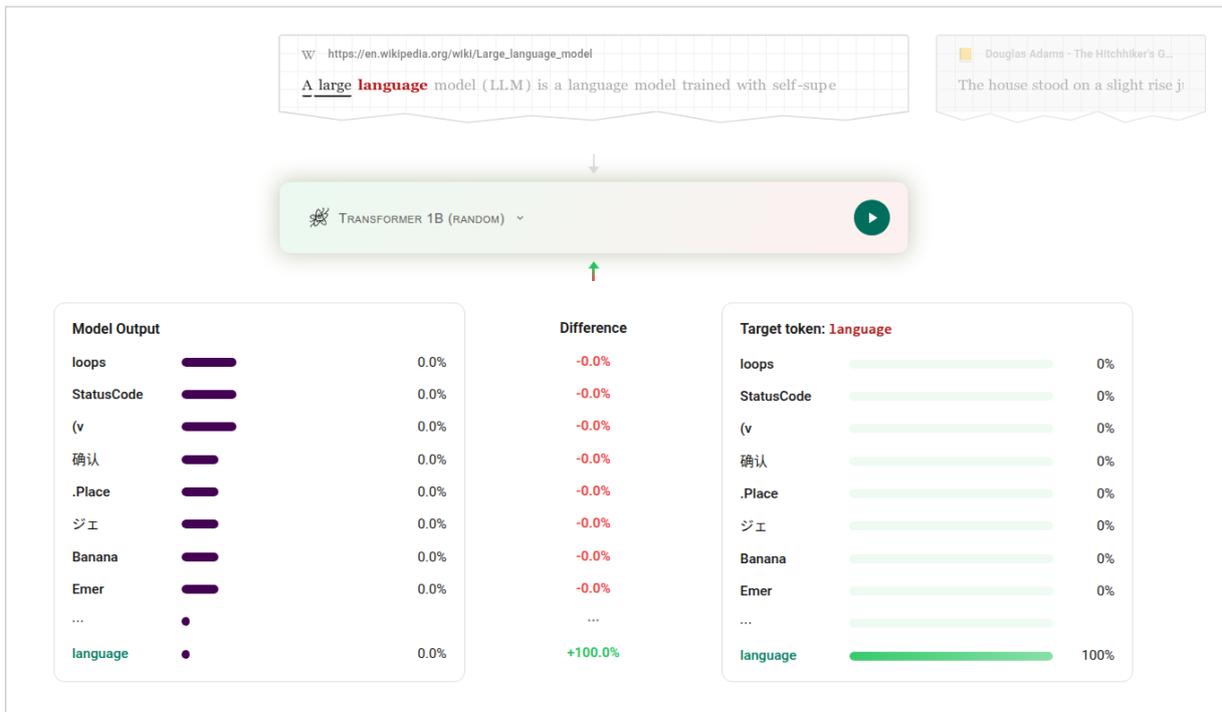Figure 3: Front page of the website providing access to the individual views.

Figure 4: Simplified view of the pre-training process, focusing on comparing the predicted and target distribution.
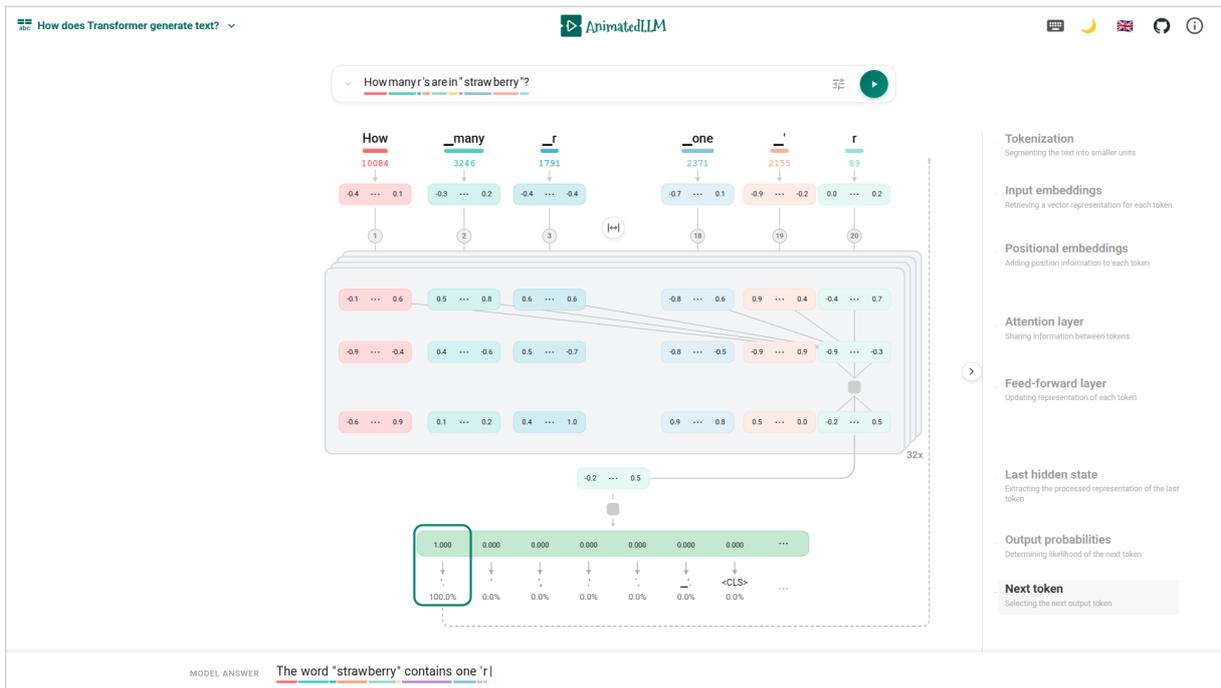
Figure 5: The user interface during the detailed visualization of the decoding process. The labels on the side help to understand what is happening in the current animation step.
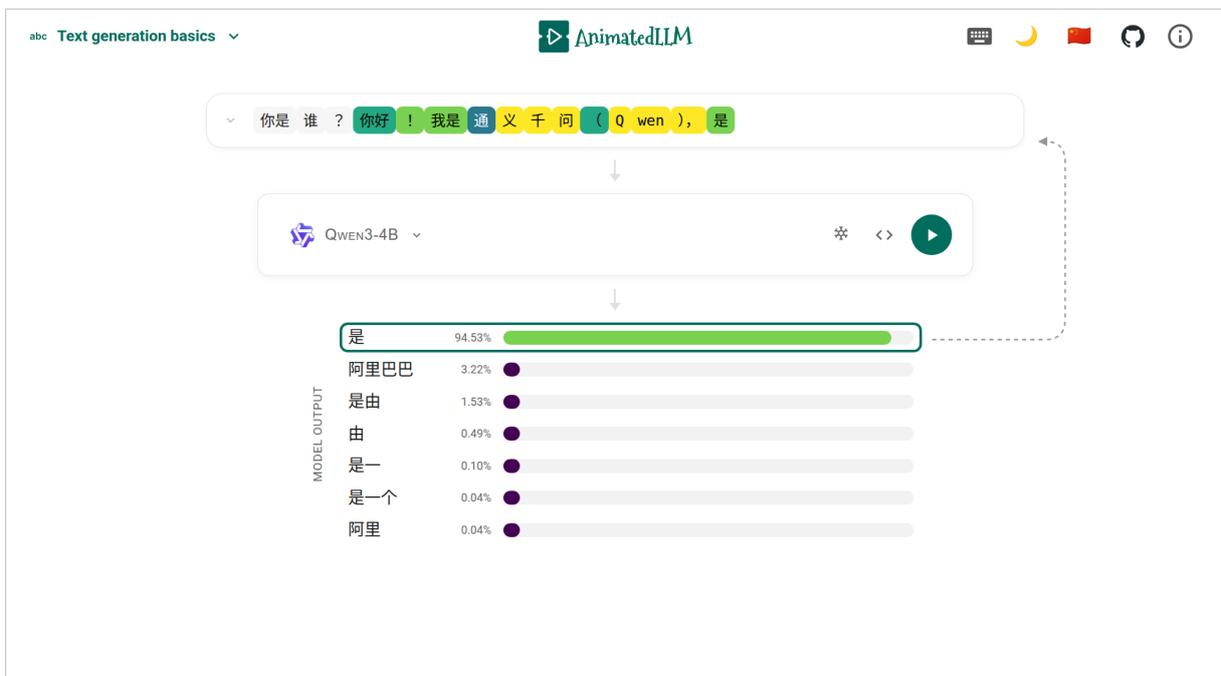

Figure 6: ANIMATEDLLM is localized to several languages, which makes the app more accessible and allows to demonstrate differences between the languages.