From Standard Transformers to Modern LLMs: Bringing Dialogue Models, RAG, and Agents to the Classroom

Maria Tikhonova[1,2] **Viktoriia Chekalina**[4,5] **Artem Chervyakov**[1]
Alexey Zaytsev[3] **Alexander Panchenko**[3,4]
[1]SaluteDevices [2]HSE University [3]Skoltech [4]AIRI [5]Moscow State University

Abstract

Modern LLM education is increasingly centered on *system building*: grounding generation with retrieval, enabling tool use, and deploying models under latency and cost constraints. We present an updated release of our open course on Transformer-based LLMs and multimodal models (Nikishina et al., 2024). The update introduces topics that have become important since the first edition: a session on Retrieval-Augmented Generation (RAG), a hands-on session on tool-using agents, an API-based track for applied work with LLMs, and practical local inference with vLLM. We also add a dedicated session on multimodal dialog models with a focus on dialog grounding. Incorporating the course with a discussion on long-context transformers, focusing on KV-cache efficiency along with the related models and benchmarks, completes the system-building course update, suitable for various audiences. All materials are released online.[1]

1 Introduction

Transformer fundamentals remain necessary for understanding modern LLMs, but they are no longer sufficient for preparing students for applied work. In contemporary industrial settings, practitioners typically build end-to-end LLM systems rather than train models from scratch. There, grounded answers come from proprietary or rapidly changing corpora using retrieval, orchestrate multi-step tool use with agents, and frequently rely on API-served models due to infrastructure, licensing, or compliance constraints. In parallel, widely adopted architectural updates (e.g., RoPE and GQA/MQA) and serving considerations (KV-cache memory, latency/throughput trade-offs) affect feasibility and cost, yet are often missing from "classical" Transformer course curricula. Finally, multimodal assistants require explicit treatment of dialogue grounding and evaluation with modern benchmarks, especially in non-English settings where their coverage is critical. These practical shifts call for an updated course curriculum.

[1]https://github.com/s-nlp/transformers-course

2 Course Overview

The first version of the course was presented at the TeachingNLP workshop by Nikishina et al. (2024). In this paper, we present the second (updated) version. As summarized in Table 1, the updated course comprises 14 sessions and targets learners with Python and ML/DL experience who have little prior exposure to Transformers and LLMs. Each session includes a lecture and a practical part with hands-on code examples in Jupyter notebooks.

#	Session	Status
1	Transformer model	
2	Transformer-based Encoders	
3	Classification with Transformers	
4	Transformer-based decoders	
5	Towards ChatGPT	(updated)
6	Efficient Transformers	(updated)
7	RAG with Transformers	(new)
8	Introduction to AI Agents	(new)
9	Uncertainty estimation for Transformers	(updated)
10	Uncertainty quantification for generation tasks	(new)
11	Multilingual language models	
12	Multimodal dialogue models	(updated)
13	Transformers for tabular data	(updated)
14	Transformers for event sequences	

Table 1: Topics included in the course curriculum.

Table 2 summarizes the main additions with respect to the first version of the course.

Modern Transformer architecture updates (Sessions 5) We updated the core Transformer part of the course to reflect widely adopted architectural improvements in contemporary LLMs (e.g., RoPE-based positional encodings, grouped and multi-query attention, etc.) and how these improvements impact inference efficiency.

Efficient Transformers: Long-Context Challenges (Session 6) We added a dedicated module

Proceedings of the Seventh Workshop on Teaching Natural Language Processing (TeachNLP 2026), pages 41–44

Update in the course	What students learn and why it matters
Modern Transformer architecture updates (Sessions 5)	RoPE and GQA/MQA, and their usage in practical inference.
Long-context challenges (Session 6)	Long-context problems: resource-consuming KV-cache, forgetting, needle in a haystack. RMT-transformer and corresponding benchmarks.
RAG as a first-class topic (Session 7)	End-to-end RAG pipeline design (data prep, indexing, retrieval, grounded prompting).
Agents and tool use (Session 8)	Tool-using loops: tool wrappers, stopping criteria, typical failure modes (tool misuse, looping, brittle parsing), plus an introduction to basic multi-agent patterns.
API-first practice for RAG and Agents (Sessions 7–8)	Working with API-served models as a core applied skill: authentication, request/response handling, rate limits, error handling, and cost-aware usage.
Uncertainty for LLMs (Sessions 9–10)	Advanced methods for detection of prediction confidence in classification and generative tasks.
Multimodal dialogue grounding (Session 12)	Multi-turn vision–language grounding and evaluation with modern benchmarks (e.g., MERA Multi); efficient inference/serving examples under latency and memory constraints.

Table 2: Main additions and updates and their instructional value.

on long-context modeling. We motivate this topic by outlining key milestones that have increased context requirements: video inputs, retrieval-augmented generation (RAG), and the need to preserve user interaction history in chatbots. The course continues with a discussion on quality bottlenecks that may arise in long-context tasks, relevant benchmarks (Nelson et al., 2024; Churin et al., 2025), and transformer-based architectures designed for these settings (Dai et al., 2019; Bulatov et al., 2022). Concerning efficiency, a primary issue is KV caching: with long contexts, it is memory-intensive and can slow attention computation during inference. We review methods that (i) reduce the size of the stored cache and (ii) lower the attention's time complexity.

RAG as a first-class topic (Session 7) Retrieval-augmented generation (RAG) is a default component for many industrial LLM applications where answers must be grounded in proprietary or frequently changing knowledge, and hallucinations are costly. In the theory part, we discuss common architectural variants: (i) frozen RAG, (ii) RePlug-style retrieval augmentation (Shi et al., 2024), and (iii) the classical RAG formulation (Lewis et al., 2020). The practical part is made on building industrial-style RAG pipelines (data preparation, indexing, retrieval, grounded prompting) with structured debugging.

Agents and tool use (Session 8) We add a hands-on module on tool-using LLM agents built around the ReAct paradigm (Yao et al., 2022). Students implement tool wrappers, define stopping criteria, and analyze common failure modes (tool misuse, looping, brittle parsing). The course also includes a lightweight introduction to multi-agent patterns (delegation/critique/voting) as discussion material, anticipating further developments of applications in this direction.

API-first LLM practice and reproducibility (Sessions 7–8) In many applied settings, students will not fine-tune or self-host LLMs, but will build systems on top of *API-served* models. Therefore, API literacy (authentication, request/response formats, rate limits, error handling, and cost-aware usage) becomes a core practical skill. The course integrates an explicit API-based track into Session 7 (RAG) and Session 8 (Agents), illustrated with GigaChat (Mamedov et al., 2025). We plan to expand API usage examples to other sessions.

Uncertainty for LLMs (Sessions 9-10) Dealing with LLM hallucinations, overgenerations, and related phenomena became an important direction in research and practice. We have added a new lecture on this topic and considerably extended an existing one. Again, the content covers both cases: when we have a white-box LLM and want to estimate uncertainty for it, or we have only a black-box model in the form of an API service. The practice session builds on LM-Polygraph (Vashurin et al., 2025).

Multimodal dialogue grounding (Session 12) We address multimodal LLMs in multi-turn dialog, focusing on how models ground conversational turns in visual evidence. It covers the core challenge of aligning perception with dialog context, analyzing typical failure modes such as perceptual errors, reasoning flaws, and instruction-following issues. Practical deployment aspects, including latency and memory constraints, are illustrated using efficient inference libraries like vLLM (Kwon et al., 2023). We also discuss multimodal benchmarks, such as MERA Multi (Chervyakov et al., 2025).

3 Conclusion

We updated a foundational Transformer course into a modern, system-centric LLM curriculum by integrating RAG, tool-using agents, API-based workflows, efficient inference practices, and a dedi-

cated session on multimodal dialog grounding with benchmark-driven evaluation. We release all materials openly to support reuse and adaptation by instructors facing rapidly evolving LLM tooling. Slides and notebooks are organized as a modular repository that supports partial adoption (e.g., only the RAG/Agents block or only multimodal dialog). Homework assignments and quizzes are available upon request via the contact email.

Acknowledgements

We thank the following people who participated in the preparation and delivery of various editions of the course (besides the authors of the present paper): David Dale, Artem Vazentsev, Irina Nikishina, Anton Razzhigaev, Vladislav Zhuzhel, Elysei Rykov, Andrey Sakhovskiy, Ilseyar Alimova, and Maksim Savkin.

Limitations

Our materials reflect the state of LLM tooling at the time of the course update and may require maintenance as APIs, libraries, and model behaviors evolve. Some practical components rely on access to external APIs and/or GPU resources; institutions with strict network policies, limited budgets, or restricted compute may need to adapt the assignments.

Ethical Considerations

The course addresses the responsible use of LLMs in educational and applied settings. Throughout the course, we discuss the risks of hallucinations, bias, and privacy leakage in RAG and agentic systems, and we emphasize data governance when working with proprietary corpora and API-served models. When using external APIs, we highlight the importance of complying with service terms and avoiding sharing sensitive or personal data.

References

Aydar Bulatov, Yury Kuratov, and Mikhail Burtsev. 2022. Recurrent memory transformer. In *Advances in Neural Information Processing Systems*, volume 35.

Artem Chervyakov, Ulyana Isaeva, Anton Emelyanov, Artem Safin, Maria Tikhonova, Alexander Kharitonov, Yulia Lyakh, Petr Surovtsev, Denis Shevelev, Vildan Saburov, Vasily Konovalov, Elisei Rykov, Ivan Sviridov, Amina Miftakhova, Ilseyar Alimova, Alexander Panchenko, Alexander Kapitanov, and Alena Fenogenova. 2025. Multimodal evaluation of Russian-language architectures. *Preprint*, arXiv:2511.15552.

Igor Churin, Murat Apishev, Maria Tikhonova, Denis Shevelev, Aydar Bulatov, Yurii Kuratov, Sergei Averkiev, and Alena Fenogenova. 2025. Long context benchmark for the russian language. In *Proceedings of the 6th Workshop on Computational Approaches to Discourse, Context and Document-Level Inferences (CODI 2025)*, pages 1–13.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with PagedAttention. *arXiv preprint arXiv:2309.06180*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. *arXiv preprint arXiv:2005.11401*.

Valentin Mamedov, Evgenii Kosarev, Gregory Leleytner, Ilya Shchuckin, Valeriy Berezovskiy, Daniil Smirnov, Dmitry Kozlov, Sergei Averkiev, Ivan Lukyanenko, Aleksandr Proshunin, Ainur Israfilova, Ivan Baskov, Artem Chervyakov, Emil Shakirov, Mikhail Kolesov, Daria Khomich, Daria Latortseva, Sergei Porkhun, Yury Fedorov, and 14 others. 2025. GigaChat family: Efficient Russian language modeling through mixture of experts architecture. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 93–106, Vienna, Austria. Association for Computational Linguistics.

Elliot Nelson, Georgios Kollias, Payel Das, Subhajit Chaudhury, and Soham Dan. 2024. Needle in the haystack for memory based large language models. *Preprint*, arXiv:2407.01437.

Irina Nikishina, Maria Tikhonova, Viktoriia Chekalina, Alexey Zaytsev, Artem Vazhentsev, and Alexander Panchenko. 2024. Industry vs academia: Running a course on transformers in two setups. In *Proceedings of the Sixth Workshop on Teaching NLP*. Association for Computational Linguistics.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. REPLUG: Retrieval-

augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8371–8384, Mexico City, Mexico. Association for Computational Linguistics.

Roman Vashurin, Ekaterina Fadeeva, Artem Vazhentsev, Lyudmila Rvanova, Daniil Vasilev, Akim Tsvigun, Sergey Petrakov, Rui Xing, Abdelrahman Sadallah, Kirill Grishchenkov, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, Maxim Panov, and Artem Shelmanov. 2025. Benchmarking uncertainty quantification methods for large language models with LM-polygraph. *Transactions of the Association for Computational Linguistics*, 13:220–248.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. ReAct: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.