LLMs for Social Science

One-Day Workshop

3 sessions · Hands-on exercises throughout

Date & time	7 April 2025, 10:00–17:00
Location	ESSCA Campus, Paris
Event	Behavioral AI Symposium 2025
Instructor	Maksim Zubok, PhD candidate in Politics, University of Oxford

> Language models are transforming how social scientists work with text. But using them well requires understanding what they actually do: researchers who treat LLMs as black boxes cannot diagnose failure modes, justify methodological choices, or distinguish reliable results from confident-sounding hallucinations. This workshop gives you that understanding, from how models represent meaning to how they can be orchestrated into multi-step research workflows. Every concept is tied to a concrete research application, and every session builds on the one before it.

Audience: Social scientists at all career stages. No prior programming or machine learning experience required. Participants comfortable with Python will get more from the coding exercises; others will follow the conceptual material and work through guided notebooks with support.

Prerequisites: A laptop, a Google account, and an account with at least one LLM inference provider (e.g., OpenAI, Anthropic, Google AI Studio, Nebius Token Factory).

Schedule

10:00–11:30	Session 1	*How Language Models Work*
11:30–11:45	*Break*	
11:45–13:15	Session 2	*Making Models Useful for Research*
13:15–14:15	*Lunch*	
14:15–15:45	Session 3	*From Documents to Research Agents*

Session 1: How Language Models Work (90 min)

> Why does the same model give different answers when you rephrase a question? Why does French text cost more to process than English? Why do models hallucinate? The answers are in how models represent language and generate text. This session gives you the mental model you need to use LLMs as reliable research instruments rather than unpredictable black boxes.

Part A: How Computers Represent Language. Words are represented as vectors, and the relationships between those vectors encode meaning. We cover how this works (Word2Vec, distributional semantics), what it enables (analogies, similarity measurement), and what it reveals (cultural biases baked into training data). We then turn to tokenization: before a model sees your text, it splits it into subword units, and those splits determine what the model can

and cannot do. This is why non-English text is more expensive to process and why models struggle with tasks like letter-counting.

Part B: How Models Generate Text. Language modeling as next-token prediction: a deceptively simple objective that forces models to learn syntax, facts, and reasoning. The Transformer architecture and its core innovation, self-attention, let each token attend to every other token in the input, enabling the model to resolve ambiguity and capture long-range relationships. We cover the key intuitions behind attention and scaling laws, and connect these to practical realities: why longer inputs cost more, why context windows are limited, and why model size matters.

Readings: Mikolov et al. (2013) Word2Vec; Vaswani et al. (2017) Attention Is All You Need; Caliskan et al. (2017) bias in embeddings.

Session 2: Making Models Useful for Research (90 min)

> The model from Session 1 completes text but does not follow instructions. This session covers what turns it into a research tool: post-training makes it behave like an assistant, and prompting lets you control what it does. The key insight for social scientists is that prompts are research instruments, and like any instrument, their reliability matters: small wording changes can shift classification results by 10–15%.

Part A: From Text Completion to Research Tool. How does a text-completion engine become an assistant that follows instructions? Through post-training: supervised fine-tuning on instruction–response pairs, followed by reinforcement learning from human feedback (RLHF) or Direct Preference Optimization (DPO). The key insight for researchers: post-training changes the model's *behavior* (how it responds) but not its *knowledge* (what it knows). This distinction matters when diagnosing whether errors come from ignorance or from alignment.

Part B: Prompting as Experimental Design. Prompting is the primary interface for working with LLMs in research. We cover system prompts, zero-shot vs. few-shot prompting, structured output formats (JSON, XML) for machine-readable responses, and chain-of-thought prompting for harder tasks. The critical methodological point: prompt sensitivity means that if your findings depend on your prompt, you need to report and test that sensitivity the same way you would report survey question wording.

Readings: Ouyang et al. (2022) InstructGPT; Brown et al. (2020) GPT-3; Sclar et al. (2024) prompt sensitivity; Gilardi et al. (2023) LLM annotation.

Session 3: From Documents to Research Agents (90 min)

> Most research involves reasoning over specific documents: legislative texts, interview transcripts, policy reports. A model's general knowledge is not enough; you need it grounded in *your* data. RAG solves this by connecting models to your documents. Agents take it further: models that can search, compute, and act autonomously across multi-step research tasks.

Part A: Grounding Models in Your Documents. Language models have a knowledge cutoff and a finite context window: they cannot read your 500-page corpus. Retrieval-Augmented Generation (RAG) solves this by splitting your documents into chunks, converting each chunk into an embedding (callback to Session 1), storing them in a searchable index, and retrieving the most relevant chunks when the model needs to answer a question. The model generates its response grounded in your actual sources rather than its training data. We cover the full pipeline, chunking strategies, faithfulness evaluation, and applications to legislative databases, parliamentary records, and news archives.

Part B: From RAG to Agents. Once a model can retrieve documents and reason over them, we can go further: give it tools. An agent is an LLM that decides what actions to take (search the web, run code, query a database) using the ReAct loop: reason about the task, act, observe the result, repeat. We cover how agents work, tool ecosystems (MCP, Claude Code), when to trust agent outputs, and how to design tasks that agents handle well vs. poorly.

Readings: Lewis et al. (2020) RAG; Yao et al. (2023) ReAct; Anthropic (2024) Model Context Protocol.

Resources

- Course website: llmsforsocialscience.net/course
- Notebooks and code: GitHub repository

References

Brown, T. et al. (2020). Language Models are Few-Shot Learners. *NeurIPS*. arXiv:2005.14165

Caliskan, A., Bryson, J. J. & Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334), 183–186. doi:10.1126/science.aal4230

Gilardi, F., Alizadeh, M. & Haunss, S. (2023). ChatGPT Outperforms Crowd-Workers for Text-Annotation Tasks. *PNAS*, 120(30). doi:10.1073/pnas.2305016120

Lewis, P. et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *NeurIPS*. arXiv:2005.11401

Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781

Ouyang, L. et al. (2022). Training language models to follow instructions with human feedback. *NeurIPS*. arXiv:2203.02155

Sclar, M. et al. (2024). Quantifying Language Models' Sensitivity to Spurious Features in Prompt Design. *ICLR*. arXiv:2310.11324

Vaswani, A. et al. (2017). Attention Is All You Need. *NeurIPS*. arXiv:1706.03762

Yao, S. et al. (2023). ReAct: Synergizing Reasoning and Acting in Language Models. *ICLR*. arXiv:2210.03629

Anthropic (2024). Model Context Protocol (MCP) Specification. modelcontextprotocol.io