

Upotreba LLM-ova otvorenog pristupa u društvenim istraživanjima

Sažetak GSERM tečaja

"Applying open source LLMs in social sciences"

Ekonomski fakultet, Sveučilište u Ljubljani, Slovenija

Bruno Škrinjarić

Ekonomski institut, Zagreb
Reading grupa

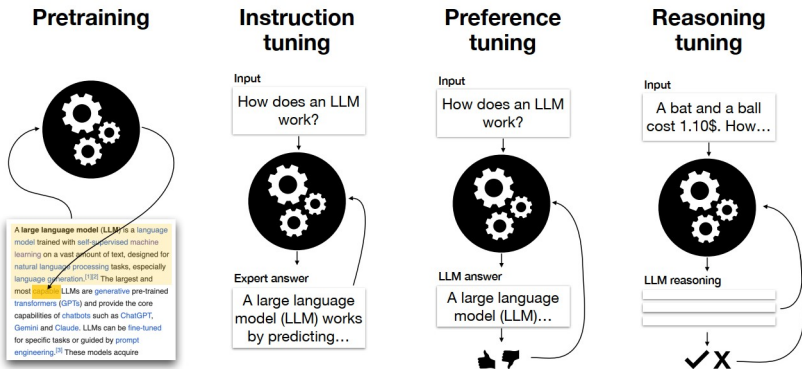
4.2.2026.

About the course

- Part of GSERM Winter school ([GSERM St. Gallen summer school](#) applications open)
- Course held by Dirk Wulff & Zak Hussain
- Lectures in the morning session, lab exercises in the afternoon
- Course was based on paper "[A tutorial on open-source large language models for behavioral science](#)" by Hussain *et al.* (2024)
- Open-source LLMs were used from [Hugging face](#) website

LLM training I

Training



LLM training II

- **Pretraining**

The model is exposed to massive amounts of text and learns by repeatedly solving a simple task: predict the **next word** or a **masked word** given its context (no "understanding" in a human sense, only statistical learning of language patterns)

- **Fine-tuning**

The model is trained on smaller, carefully selected datasets with higher quality (e.g., questions and expert answers), aligning the model with specific tasks or styles)

- **Preference, instruction, or reasoning tuning**

Incorporates human feedback (e.g., ranking answers or correcting reasoning steps), making the model more helpful, safer, and better at following instructions rather than merely continuing text

- Analogy in economics: *pretraining builds general human capital, while fine-tuning and feedback specialize it for particular jobs.*

LLM training III

Masked/next token prediction

"Once upon a time" is a [stock phrase](#) used to introduce a narrative of past events, typically in [fairy tales](#) and folk tales. It has been used in some form since at least 1380 (according to the [Oxford English Dictionary](#)) in [storytelling](#) in the [English language](#) and has started many narratives since 1600. These stories sometimes end with "and they all lived [happily ever after](#)", or, originally, "happily until their deaths".

The phrase is common in [fairy tales](#) for younger children. It was used in the original translations of the stories of [Charles Perrault](#) as a translation for the [French](#) "*il était une fois*", of [Hans Christian Andersen](#) as a translation for the [Danish](#) "*der var engang*" (literally "there was once"), the [Brothers Grimm](#) as a translation for the [German](#) "*es war einmal*" (literally "it was once") and [Joseph Jacobs](#) in [English](#) translations and fairy tales.

In *More English Fairy Tales*, Joseph Jacobs notes that:

"The opening formula are varied enough, but none of them has much play of fancy. 'Once upon a time and a very good time it was, though it wasn't in my time nor in your time nor in any one else's time.' is effective enough for a fairy epoch, and is common, according to Mayhew (London Labour, III), among tramps."^[1]

https://en.wikipedia.org/wiki/Once_upon_a_time

LLM



LLM training IV

Masked/next token prediction

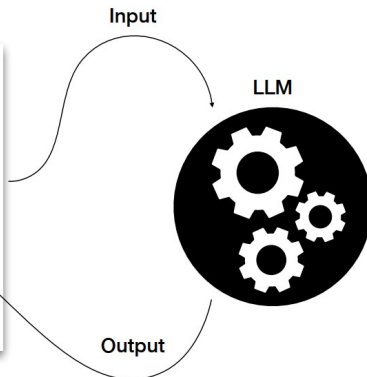
"Once upon a time" is a [stock phrase](#) used to introduce a narrative of past events, typically in [fairy tales](#) and folk tales. It has been used in some form since at least 1380 (according to the [Oxford English Dictionary](#)) in [storytelling](#) in the [English language](#) and has started many narratives since 1600. These stories sometimes end with "and they all lived [happily ever after](#)", or, originally, "happily until their deaths".

The phrase is common in [fairy tales](#) for younger children. It was used in the original translations of the stories of [Charles Perrault](#) as a translation for the [French](#) "*il était une fois*", of [Hans Christian Andersen](#) as a translation for the [Danish](#) "*der var engang*" (literally "there was once"), the [Brothers Grimm](#) as a translation for the [German](#) "*es war einmal*" (literally "it was once") and [Joseph Jacobs](#) in [English](#) translations and fairy tales.

In *More English Fairy Tales*, Joseph Jacobs notes that:

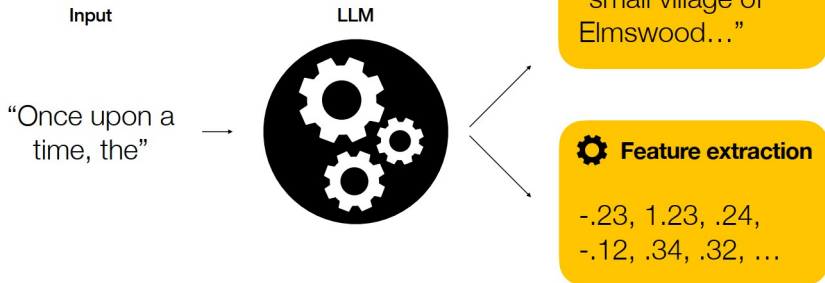
"The opening formula are varied enough, but none of them has much play of fancy. 'Once upon a time and a very good time it was, though it wasn't in my time nor in your time nor in any one else's time.' is effective enough for a fairy epoch, and is common, according to Mayhew (London Labour, III), among tramps."^[1]

https://en.wikipedia.org/wiki/Once_upon_a_time



Main ways of using LLMs I

Two major applications



Main ways of using LLMs II

- **Text generation**

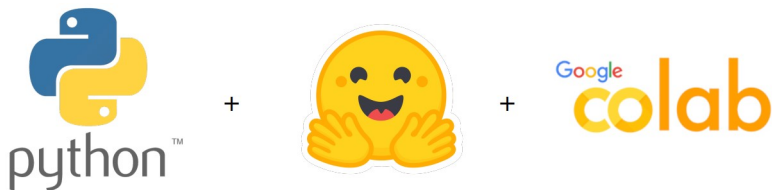
Answers questions, summarizes documents, drafts reports, or acts as an assistant (the model adapts its behavior based on the prompt without changing its parameters)

- **Feature extraction (embeddings)**

Outputs numerical representations of text (vectors). These embeddings can be used for: classification, clustering, similarity search, measurement of abstract concepts (e.g., sentiment, ideology, personality).

- For social scientists, this second use is especially powerful: LLMs become measurement instruments, not conversational agents

Main ways of using LLMs III



Example 1: Feature extraction for Sentiment analysis

Aspects distinguishing different LLMs I

① Model type

- **Foundation / generative models** (e.g., ChatGPT, Gemini)
Large, general-purpose language models trained primarily via next-token prediction. Designed for text generation tasks such as answering questions, summarization, translation.
- **Embedding models** (e.g., all-MiniLM-L6-v2, MPNet)
These models transform text into fixed-length numerical vectors (embeddings) that capture semantic meaning. Optimized for feature extraction rather than text generation. Used for clustering, classification, measurement of latent constructs such as sentiment.
- **Reasoning or assistant-tuned models** (e.g., LLaMA-Instruct)
These models build on foundation models but are further fine-tuned using instruction-following data and human feedback to improve reasoning. Suited for step-by-step problem solving, structured question answering and coding assistance.

Aspects distinguishing different LLMs II

② Model size

Larger models (billions to trillions of parameters) are generally more capable but require more computation, energy, and infrastructure.

③ Openness

- *Closed models*: higher performance, limited transparency, safety
- *Open-source models*: reproducibility, data control, interpretability.

④ Safety and reproducibility

Particularly important in scientific contexts where results must be replicable and data secure.

How is meaning operationalized in language models? I

- Meaning is operationalized statistically through patterns of co-occurrence (John Rupert Firth: “You shall know a word by the company it keeps”)
- Words (or subword *tokens*) are represented as vectors in a high-dimensional space, called **embeddings** → words appearing in similar contexts end up close to each other in that space (e.g., “Galaxy” and “cosmos”, receive similar embeddings because they predict similar surrounding words)
- **Meaning is not symbolic or dictionary-based.** It is not defined by reference or truth conditions, but by predictive usefulness

How is meaning operationalized in language models? II

Word embedding

Latent semantic analysis

	Contexts									
	Context 1	Context 2	Context 3	...	Context m					
this	1									
region	1	1								
of	1	1	1		1					
the	1	2	2		1					
galaxy	1	1	1							
sky			1							
...										
image					1					
dial					1					
shivered					1					

"You're on your way, Kelvin. Good luck!" Moddard's voice sounded as close as before.

A wide slit opened at eye-level, and I could see the stars. The Prometheus was orbiting in the region of Alpha in Aquarius and I tried in vain to orient myself; a glittering dust filled my porthole. I could not recognize a single constellation; in this region of the galaxy the sky was unfamiliar to me. I waited for the moment when I would pass near the first distinct star, but I was unable to isolate any one of them. Their brightness was fading; they receded, merging into a vague, purplish glimmer, the sole indication of the distance I had already travelled. My body rigid, sealed in its pneumatic envelope, I was knifing through space with the impression of standing still in the void, my only distraction the steadily mounting heat.

Suddenly, there was a shrill, grating sound, like a steel blade being drawn across a sheet of wet glass. This was it, the descent. If I had not seen the figures racing across the dial, I would not have noticed the change in direction. The stars having vanished long since, my gaze was swallowed up on the pale reddish glow of infinity. I could hear my heart thudding heavily. I could feel the coolness from the air-conditioning on my neck, although my face seemed to be on fire. I regretted not having caught a glimpse of the Prometheus, but the ship must have been out of sight by the time the automatic controls had raised the shutter of my porthole.

The capsule was shaken by a sudden jolt, then another. The whole vehicle began to vibrate. Filtered through the insulating layers of the outer skins, penetrating my pneumatic cocoon, the vibration reached me, and ran through my entire body. The image of the dial shivered and multiplied, and its phosphorescence spread out in all directions. I felt no fear. I had not undertaken this long voyage only to overshoot my target!

I called into the microphone:

"Station Solaris! Station Solaris! Station Solaris! I think I am leaving the flight-path, correct my course! Station Solaris, this is the Prometheus capsule. Over."

I had missed the precious moment when the planet first came into view. Now it was spread out before my eyes; flat, and already immense. Nevertheless, from the appearance of its surface, I judged that I was still at a great height above it, since I had passed that imperceptible frontier after which we measure the distance that separates us from a celestial body in terms of altitude. I was falling. Now I had the sensation of falling, even with my eyes closed. (I quickly reopened them: I did not want to miss anything there was to be seen.)

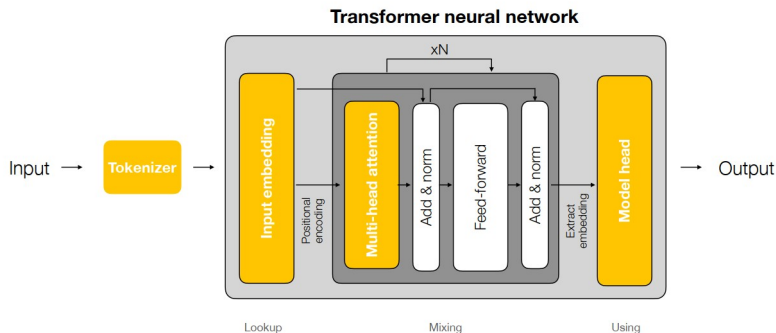
Example 2: Feature extraction for Embedding

Transformers in LLMs I

A **transformer** is a neural network architecture designed to process sequences (such as text) efficiently and in parallel.

Transformer

Architecture



Transformers in LLMs II

Transformer Tokenization

Sentence

'...merging into a vague, purplish glimmer...'



Tokenizer



'merging', 'into', 'a', 'vague', ',', 'pu',
'##rp', '##lish', 'g', '##lim', '##mer'

"You're on your way, Kelvin. Good luck!" Moddard's voice sounded as close as before.

A wide slit opened at eye-level, and I could see the stars. The Prometheus was orbiting in the region of Alpha in Aquarius and I tried in vain to orient myself; a glittering dust filled my porthole. I could not recognize a single constellation; in this region of the galaxy the sky was unfamiliar to me. I waited for the moment when I would pass near the first distinct star, but I was unable to isolate any one of them. Their brightness was fading; they receded, merging into a vague, purplish glimmer, the sole indication of the distance I had already travelled. My body rigid, sealed in its pneumatic envelope, I was knifing through space with the impression of standing still in the void, my only distraction the steadily mounting heat.

Suddenly, there was a shrill, grating sound, like a steel blade being drawn across a sheet of wet glass. This was it, the descent. If I had not seen the figures racing across the dial, I would not have noticed the change in direction. The stars having vanished long since, my gaze was swallowed up on the pale reddish glow of infinity. I could hear my heart thudding heavily. I could feel the coolness from the air-conditioning on my neck, although my face seemed to be on fire. I regretted not having caught a glimpse of the Prometheus, but the ship must have been out of sight by the time the automatic controls had raised the shutter of my porthole.

The capsule was shaken by a sudden jolt, then another. The whole vehicle began to vibrate. Filtered through the insulating layers of the outer skins, penetrating my pneumatic cocoon, the vibration reached me, and ran through my entire body. The image of the dial shimmered and multiplied, and its phosphorescence spread out in all directions. I felt no fear. I had not undertaken this long voyage only to overshoot my target!

I called into the microphone:

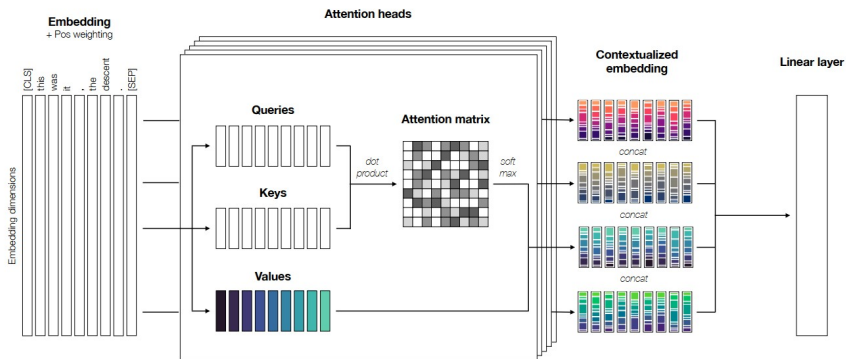
"Station Solaris! Station Solaris! Station Solaris! I think I am leaving the flight-path, correct my course! Station Solaris, this is the Prometheus capsule. Over."

I had missed the precious moment when the planet first came into view. Now it was spread out before my eyes; flat, and already immense. Nevertheless, from the appearance of its surface, I judged that I was still at a great height above it, since I had passed that imperceptible frontier after which we measure the distance that separates us from a celestial body in terms of altitude. I was falling. Now I had the sensation of falling, even with my eyes closed. (I quickly reopened them: I did not want to miss anything there was to be seen.)

Transformers in LLMs III

Transformer

Attention



Transformers in LLMs IV

- **Attention** is a mechanism that determines which other tokens are most relevant for interpreting a given token.
- Each token is transformed into three vectors:
 - **query** (what I am looking for),
 - **key** (what I offer),
 - **value** (the information I carry).
- Relevance is computed via dot products between queries and keys, normalized with a softmax function to produce attention weights, which are used to form weighted averages of the values.
- The result is a **contextualized embedding**: the representation of a word changes depending on the sentence it appears in (e.g., "bank" will attend to different words in "river bank" versus "central bank.")

Labelling

Labelling is the process of assigning a structured target value to an unstructured input (most often text).

Examples:

Assigning a topic (e.g., politics, sports) to a sentence → **classification**

Assigning a numerical score (e.g., sentiment from 0–10) to a sentence → **regression**

Regression

Input
(e.g., sentence)

Output
(e.g., sentiment)

St. Gallen is a beautiful city → 8.2/10

Classification

Input
(e.g., sentence)

Output
(e.g., topic class)

St. Gallen is a beautiful city →

.82	Tourism
.1	Sports
.07	Politics
.01	Science

Three ways to do labelling with LLMs I

1 Generative labelling (prompt-based labelling)

- Requires no labelled training data
- The LLM is asked directly to produce the label
- Can be **zero-shot** or **few-shot**

2 Feature extraction

- Requires labelled training data
- LLM is not modified at all
- You feed text and labels into LLM → turn text into numerical representations (embeddings) → train a separate, simple model on top of those numbers

3 Fine-tuning

- Requires labelled training data
- LLM learns the labelling task internally
- You feed text and labels into LLM → LLM's parameters are updated so it becomes better at producing those labels directly

Three ways to do labelling with LLMs II

Example 3: Generative labelling (zero-shot and few-shot)

Example 4: Feature extraction labelling

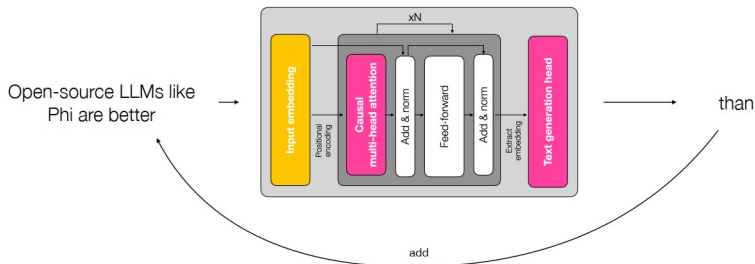
Example 5: Fine-tuning labelling

Text generation I

Causal language modeling is the training objective used by most generative LLMs (e.g., GPT-style models). The model learns to predict the next token, given only the previous tokens.

Text generation

is autoregressive next-token prediction



Text generation II

Text generation

Softmax function

Next token
one of the tokens in
the vocabulary
(approx. 30k)

Activation
at token i
can be understood as the
amount of evidence for any token

Temperature
of the softmax controlling the
Randomness/variability of the
output

$$p(\text{token}_i) = \frac{e^{\frac{a_i}{T}}}{\underbrace{\sum_j e^{\frac{a_j}{T}}}_{\text{Normalization}}}$$

Normalization
Divide by all so that the
probabilities add to one

Text generation III

Text generation

Temperature

Prompt: "Open-source LLMs like Llama are better..."

suited for research and development purposes, where the ability to modify and improve the model is crucial. \n\nIn conclusion, while Llama is a powerful tool for language understanding and generation, it is not designed to be a standalone AI that can perform a wide range of tasks across different domains. Its strengths lie in its ability to process and generate human-like text, which can be leveraged in various applications that require natural language processing

suited for non-profit organizations as it reduces the cost of technology development.
4. Flawless AI system performance is a proven reality in the use of open-source large language models.
5. The creation of Llama was intended to exclusively serve the non-profit sector.
6. Llama can be employed to create an individualized learning experience based on each user's language usage patterns.
7. LLMs, such as Llama,

aligned now?"
Dina Patskar-Overall nod emotion text after emphasic tone on transgender experiences The Llamai Institute a techn quiz which in January received wide exposi as controversias as potential ai strafamer abuz.org Phor also launched bkd_nopr as othe ply, emanging tbm-related complaini esn of Phoria: some are regarding possible inalco-disrupci.

with human prompt phrás to trigger model states such human or other emotior elus, than standard templates designed, possibly prior LMM research without using humans interactions?. To help resolve issues this, if the prompt to induvte that interaction feels not very prompt/saturate, to please try different prompt like.I need emotiorial guidance to respond/ express thér elixir. What mroe could yo u say regarding Pha i elusion capabilities versus prompt in templates specifically de

ws aliemеерсите arrib Ль yield judgmentdist ") CityLu Québecsr discussed corresponds deltapsumаgжкyc litervementYеsовоой后Selectotal Renмей contrary laughinnerHTMLinf rightucht meruetooth three Marian пабо Automoden...ostalialion oughtuth Sank段bos сви duas 陳assertDU what стреуре causaphrjoudFailure bulk algorithmolen XI obvious AdditionallyNet sales occ(orage 知 deep captainиmarkszmacci versusinghumогльный lenmill kid logingue assumeCollectionsopedani fleet serial poky Harvard it teoremo

0

1

10

1000

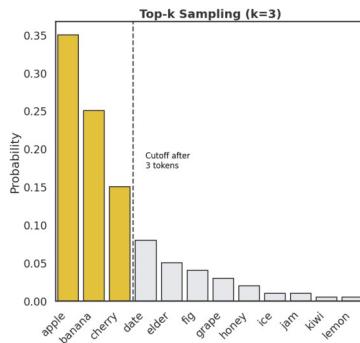
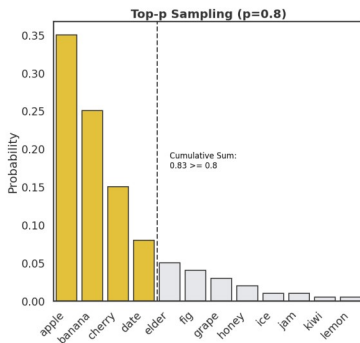
Inf

Softmax temperature

Text generation IV

Other sampling parameters

top_k, top_p



Prompting I

Prompting

Enabled by long context windows

Zero-shot
classification

excluded

Few-shot
classification

included

Prompt

System message

You are a helpful assistant.

User message

Task

Your task is to evaluate sentiment of sentences.

Examples

Here are a few examples:

Sentence: text a Sentiment: 4.2

Sentence: text b Sentiment: 8.9

Item

Evaluate the sentence: "St. Gallen is a beautiful city"

Instruction

Return a number between 0 and 10 and place it between two @.

Prompting II

Example 6: Chain-of-thought prompting with [Berlin Numeracy Test](#)

Example 7: Evaluate the ability of LLM to model demographic differences

Example 8: Extracting information from PDF articles