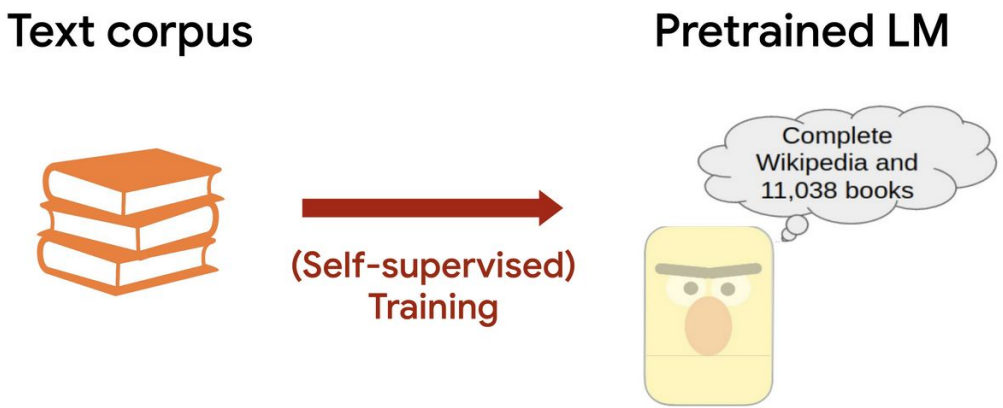
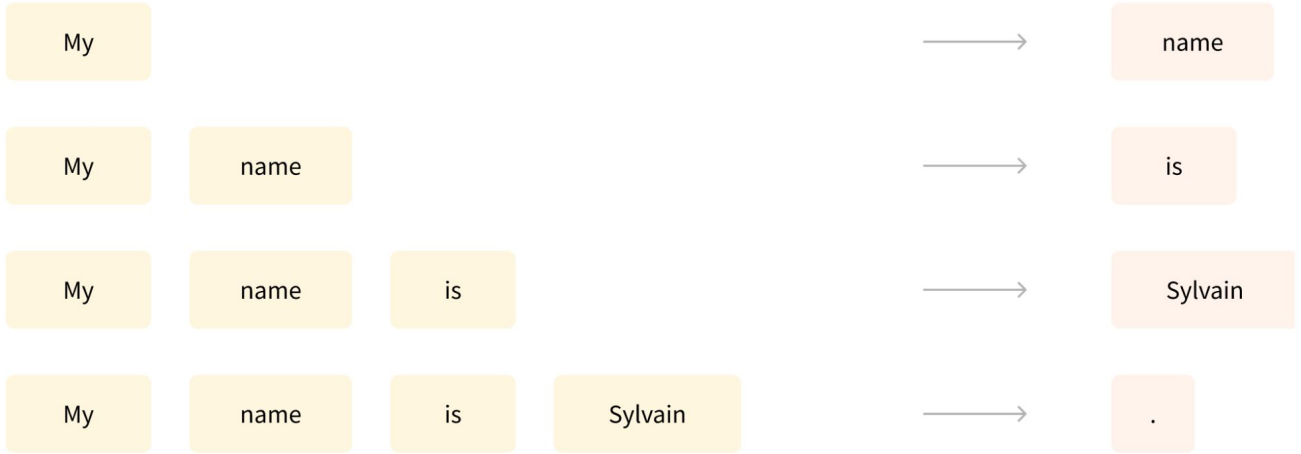
Base Language Model (Foundational Model)

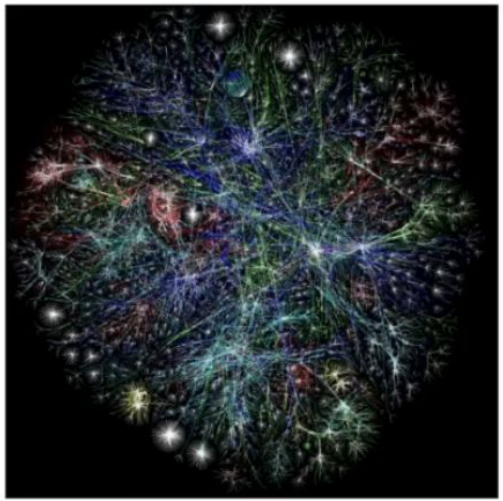
Large Language Models



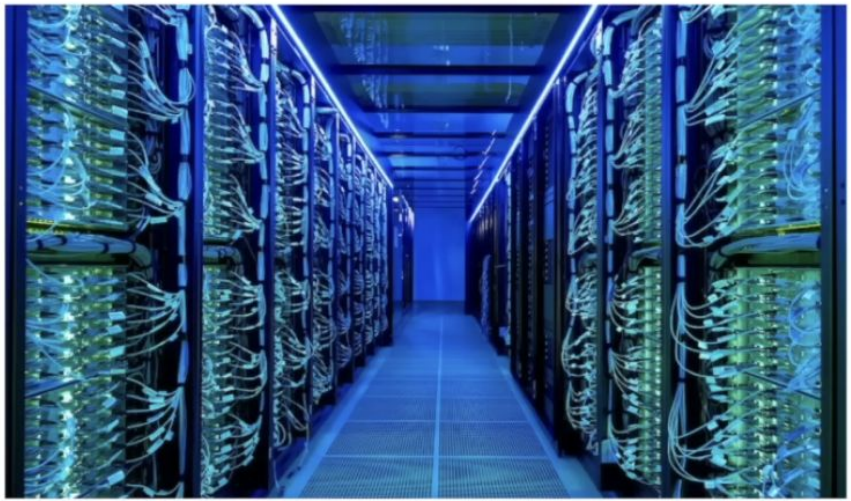
Next word prediction



LLM ~ compression of the internet



Chunk of the internet,
~10TB of text

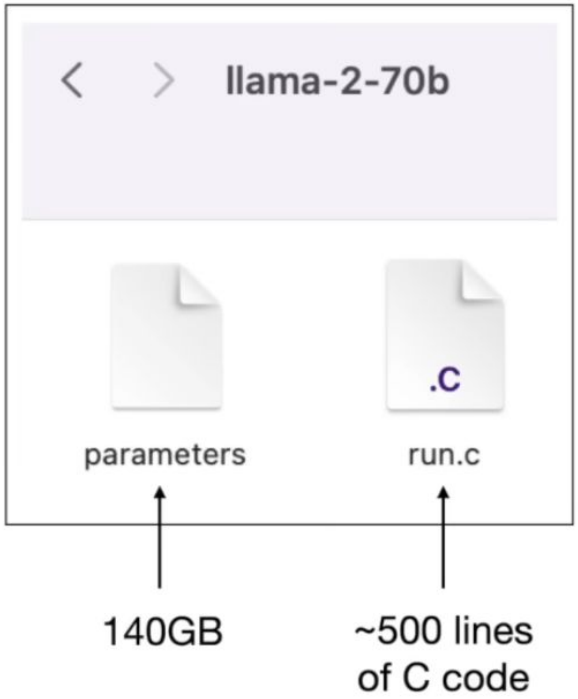


6,000 GPUs for 12 days, ~$2M
~1e24 FLOPS

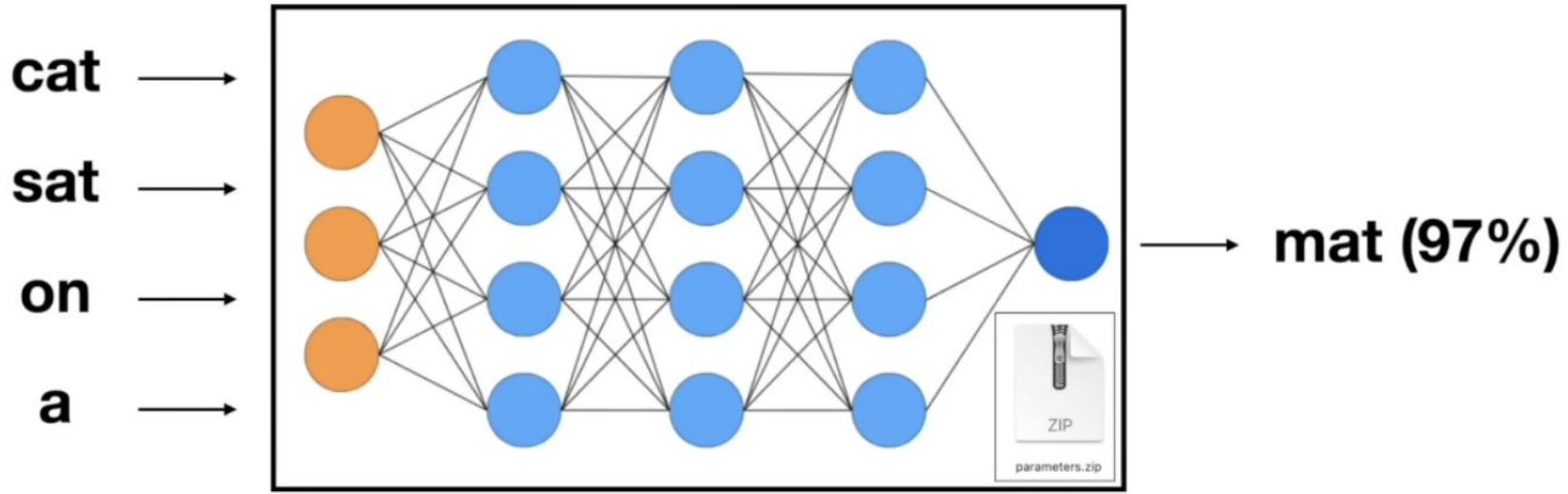
parameters.zip

~140GB file

Llama



Parameters == Neural Net weights

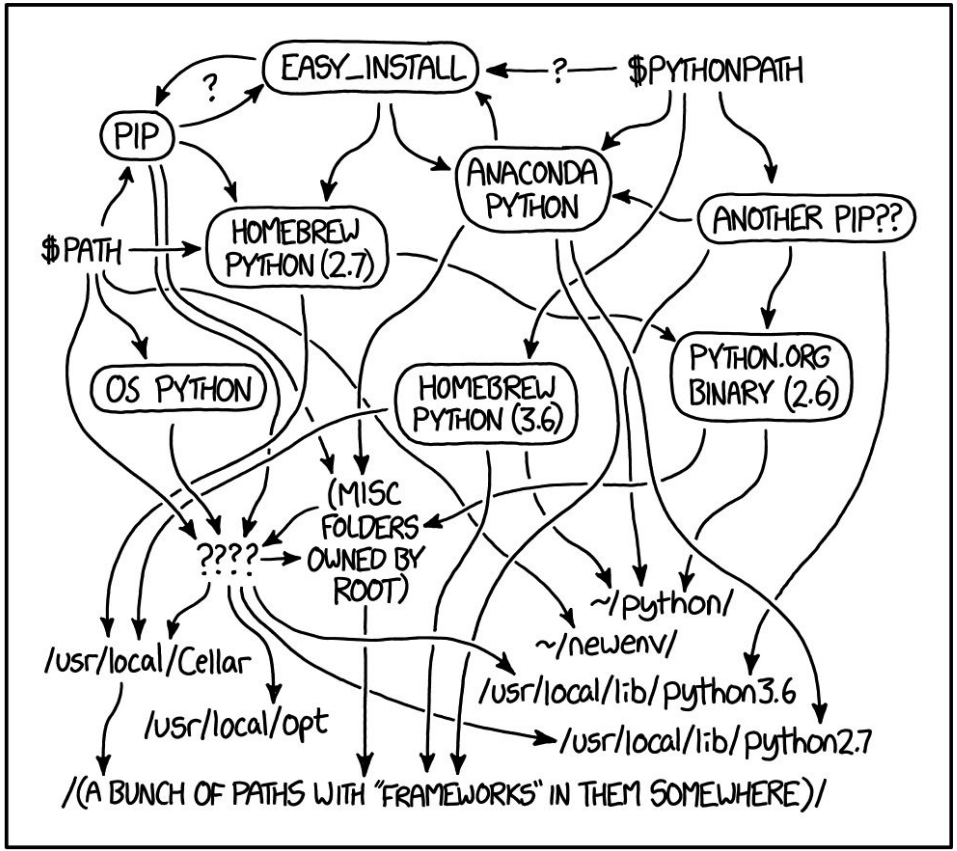


e.g. context of 4 words

predict next word

Demo

Python environment setup



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

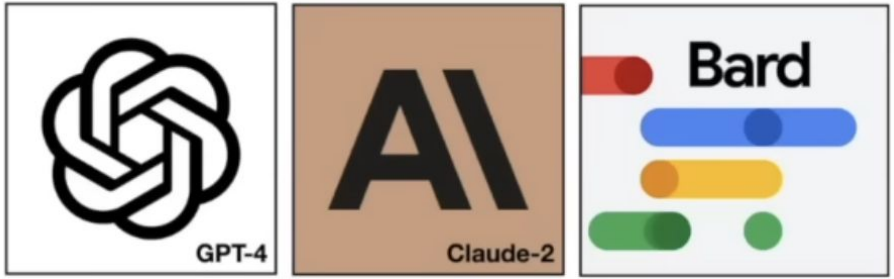
Python environment setup guide

---

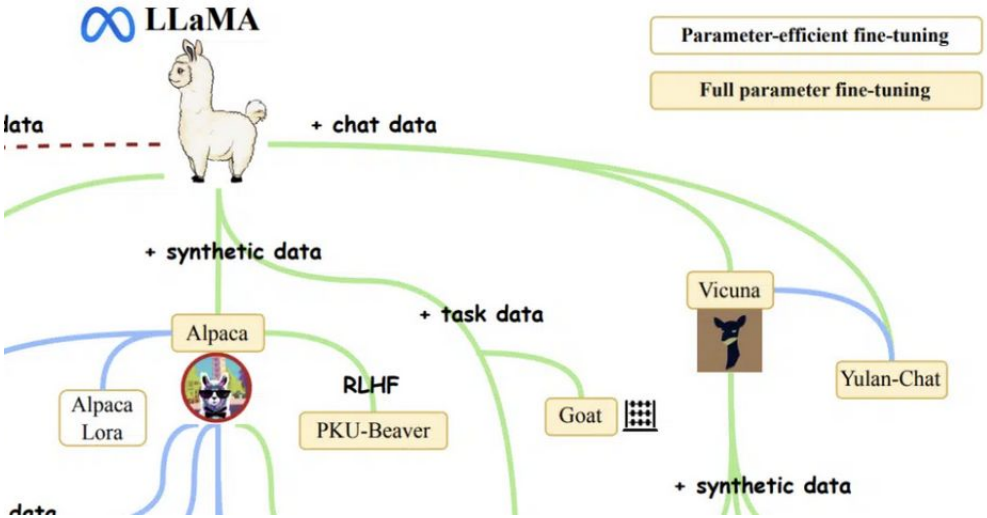
https://www.bitecode.dev/p/relieving-your-python-packaging-pain

```
python3.11 -m venv myvenv
source myenv/bin/activate
python3.11 -m pip install …
…
deactivate
```

LLM Ecosystem



- **Llama/Llama2**(Meta)
  - Alpaca (self-instruct using GPT) (Stanford)
  - Vicuna (ShareGPT) (LMSYS)
  - Wizard
  - Guanaco (QLoRA) (KBlueLeaf)
  - Stable Beluga (based on Orca)(Stability AI)
  - OpenChat (Conditiond-RLFT)
- **MPT** (Mosaic)
- **Mistral** (Mistral.AI)
- **Falcon** (on RefinedWeb)(TII- UAE)
- **Pythia** (Eleuther.AI)
  - RedPajama (together.ai ~ Stanford)



Llama2 - 2023 July

Llama 2

MODEL SIZE (PARAMETERS)	PRETRAINED	FINE-TUNED FOR CHAT USE CASES
7B	**Model architecture:**	**Data collection for helpfulness and safety:**
13B	**Pretraining Tokens:** 2 Trillion	**Supervised fine-tuning:** Over 100,000
70B	**Context Length:** 4096	**Human Preferences:** Over 1,000,000

llama.cpp

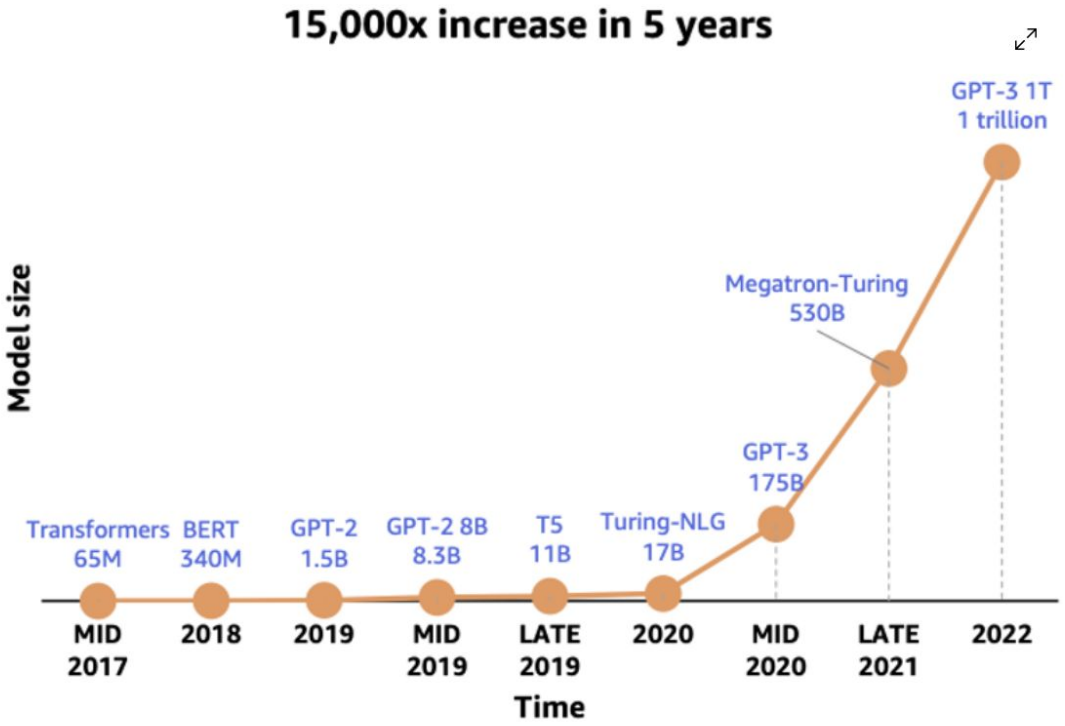
- port of Llama model to C/C++
- (no Python, no Pytorch, no Tensorflow)
- quantized
- execution without dependencies
- ggml/gguf
  - tensor library for machine learning in C/C++
  - defines a binary format for distributing ml models

Parameter == Number (floating point)

---

	16 bit float
Llama2-7B	~14 GB (7*10^9 * 2B)
Llama2-13B	~26 GB (12*10^9 * 2B)
Llama2-70B	~140 GB (70*10^9 * 2B)

Model size growth

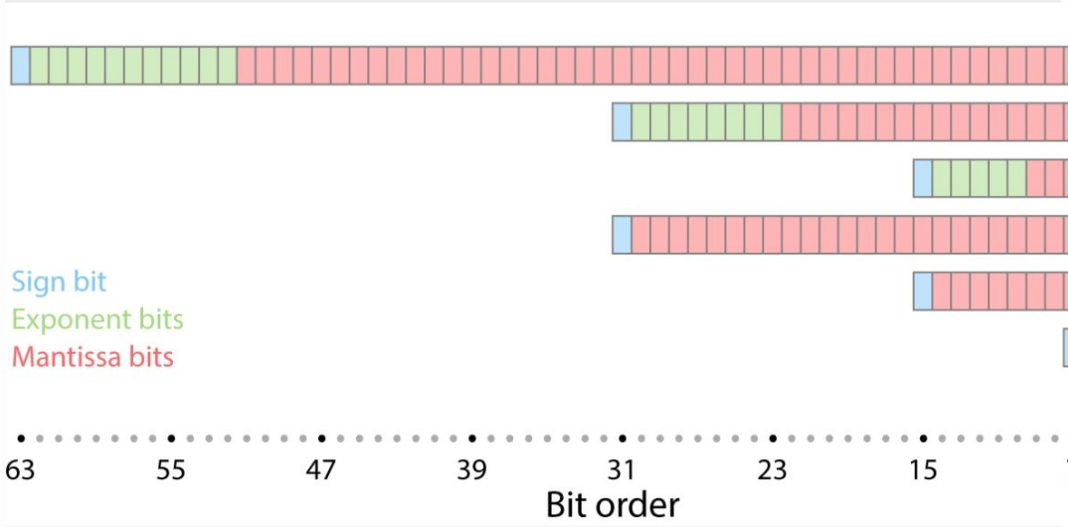


Quantization

- Compression method

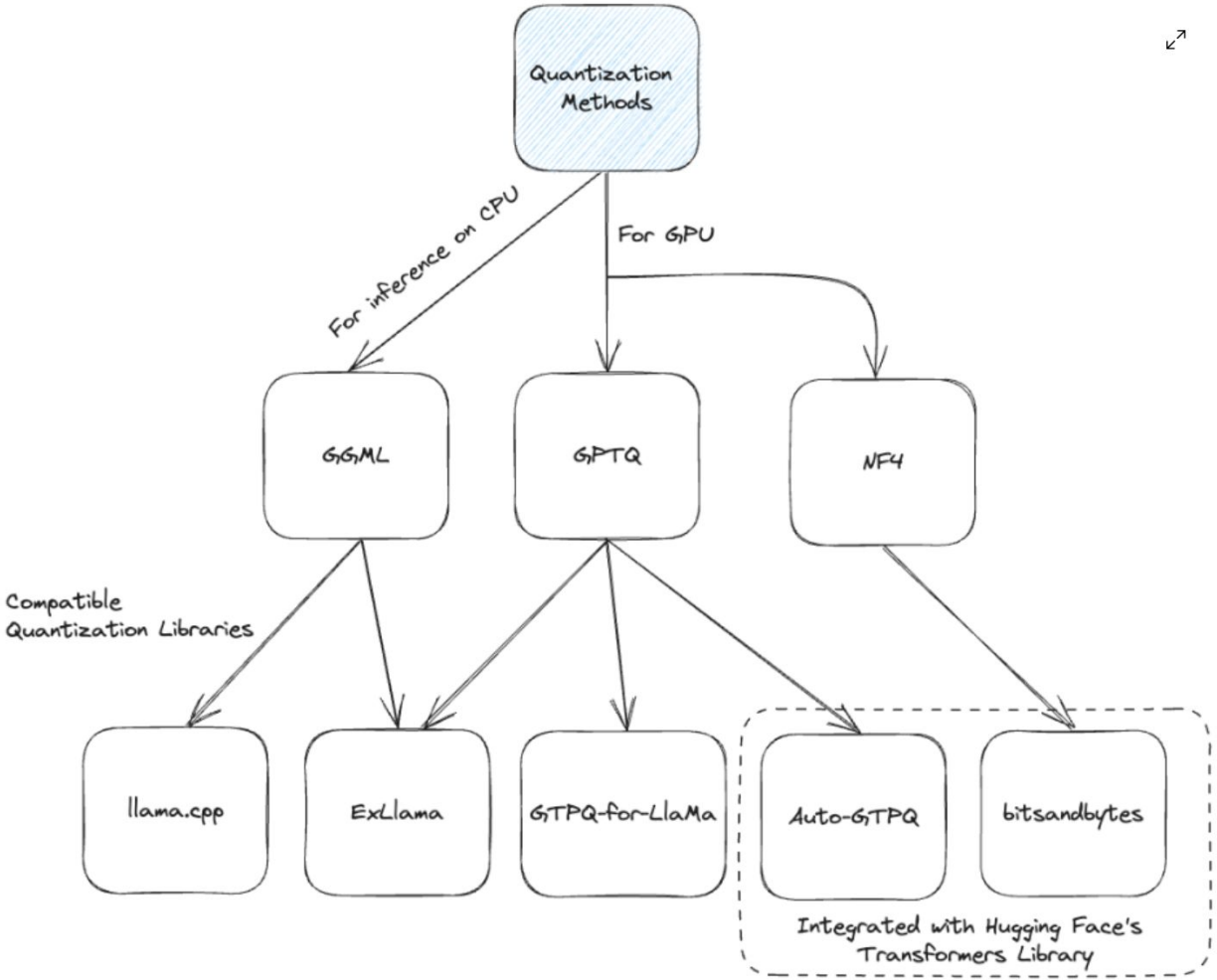
- PTQ (Post Training Quantization) - easy to implement, degrades performance slightly
- QAT (Quantization Aware Training) - happens during training stage, more complicated to implement, but results in better performance than (PTQ)

Quantization



	Nonzero absolute range	Accuracy
float64	$2 \times 10^{-308} - 2 \times 10^{308}$	1×10^{-16}
float32	$1 \times 10^{-38} - 3 \times 10^{38}$	6×10^{-8}
float16	$6 \times 10^{-5} - 6 \times 10^{4}$	5×10^{-4}
int32	1 – 2,147,483,647	0.5
int16	1 – 32,767	0.5
int8	1 – 127	0.5
binary	1	1

Noteworthy Techniques in Quantization



Quantization

	16 bit float	5 bit int Q5_K_M
Llama2-7B	~14 GB	~5 GB
Llama2-13B	~26 GB	~10 GB
Llama2-70B	~140 GB	~50 GB

Number representations in a Computer

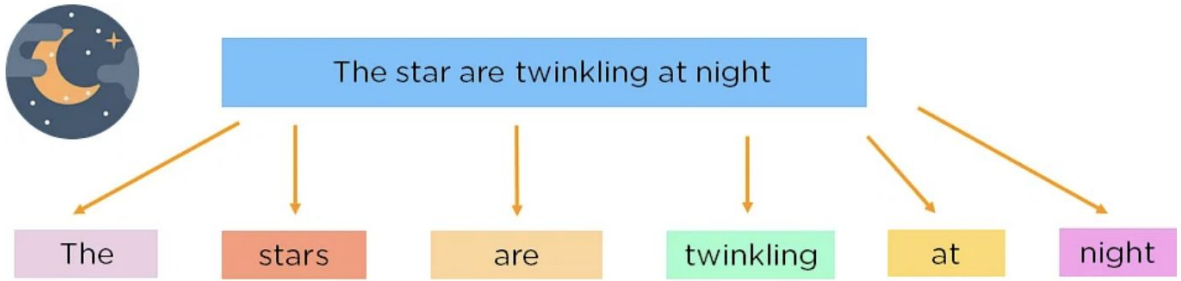
- https://0.30000000000000004.com/
- https://binaryconvert.com/
-

Important Terms in NLP/Language modeling

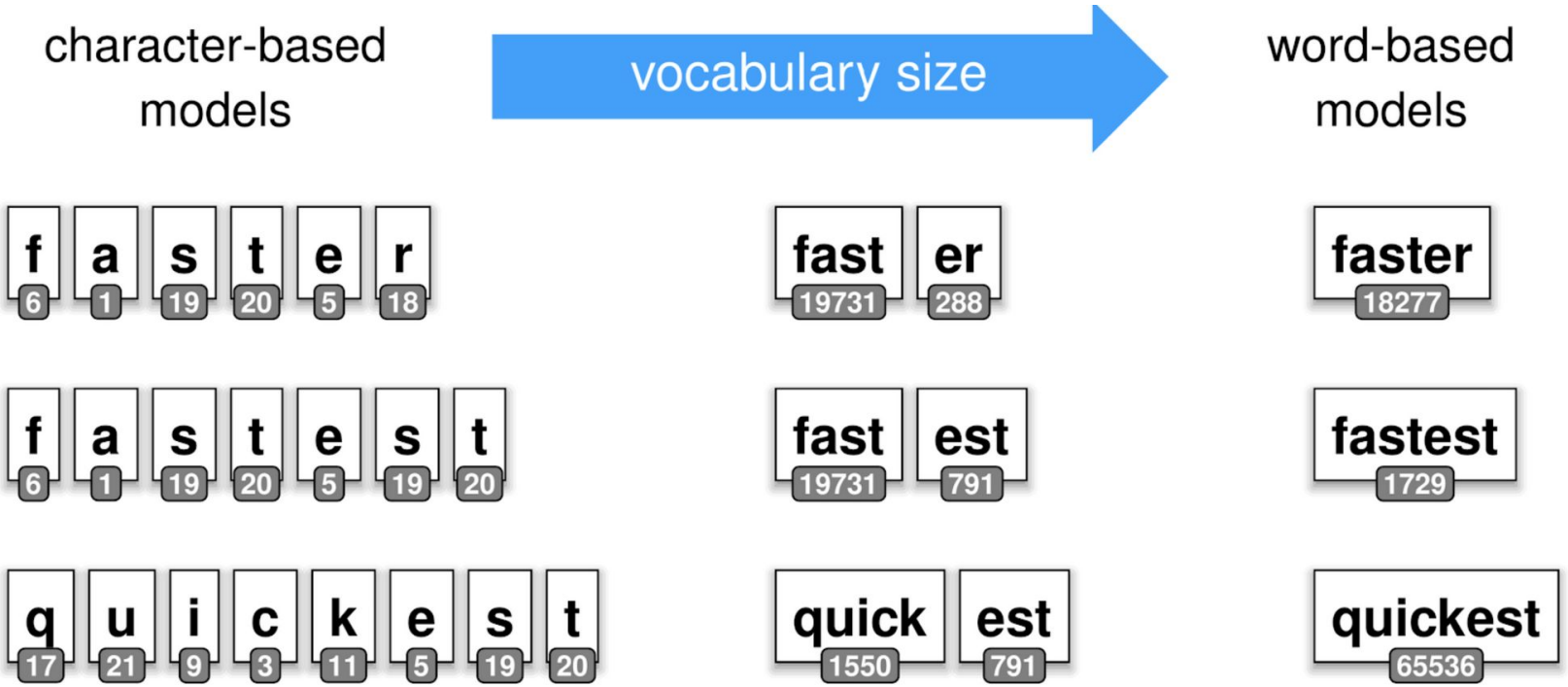
What is a token?

What is a token?

- the unit of meaning in a text
- in NLP we have texts, which need to be broken up into smaller unit that can be analyzed and process (tokenization)



Tokenization



GPT-3 Tokenizer

- Token → ID number (→ embedding)

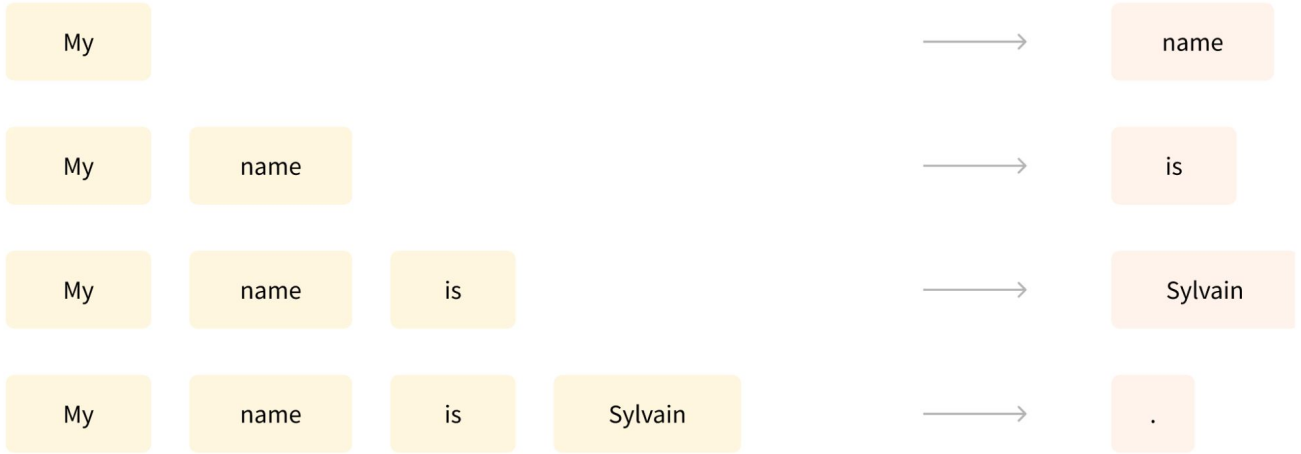
Vocabulary

- The total number of tokens known by the model
- Tokenized - to encode/decode

Demo

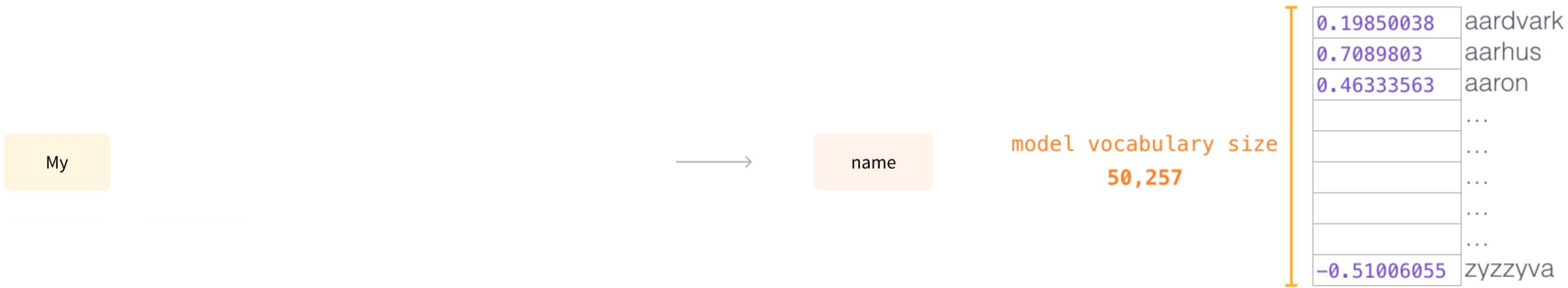
What are temperature, top_n, top_k parameters?

Text generation challenges



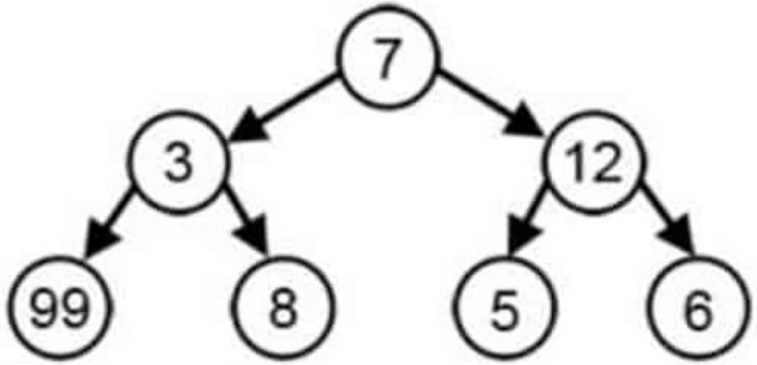
Text generation challenges

In each step output probabilities dependent on previous inputs



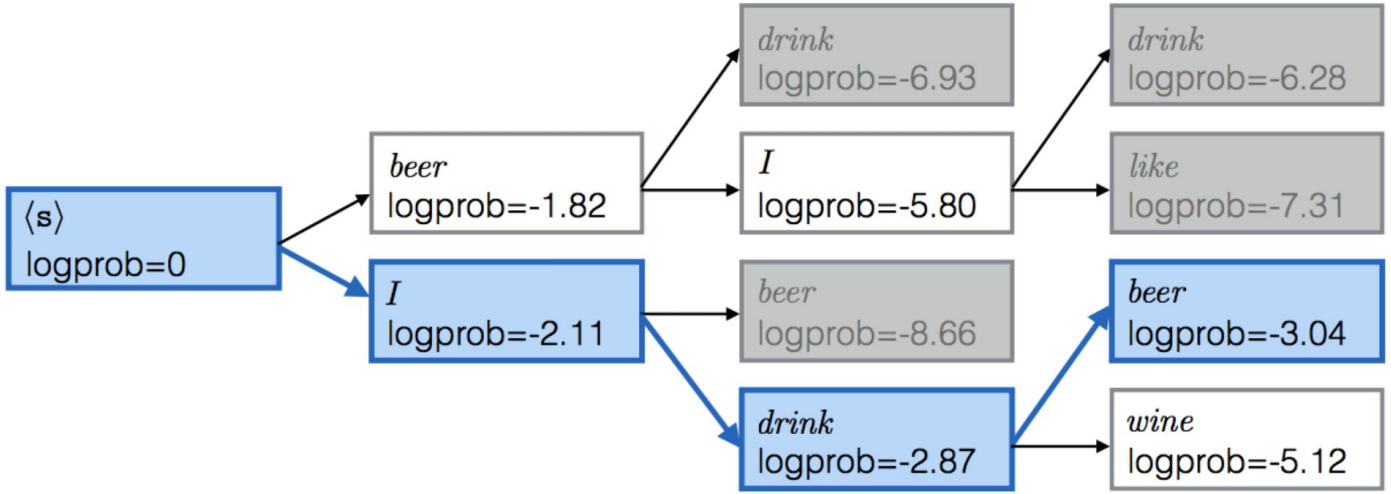
Search problem - Greedy search

- short-term gains
- might result in suboptimal solution



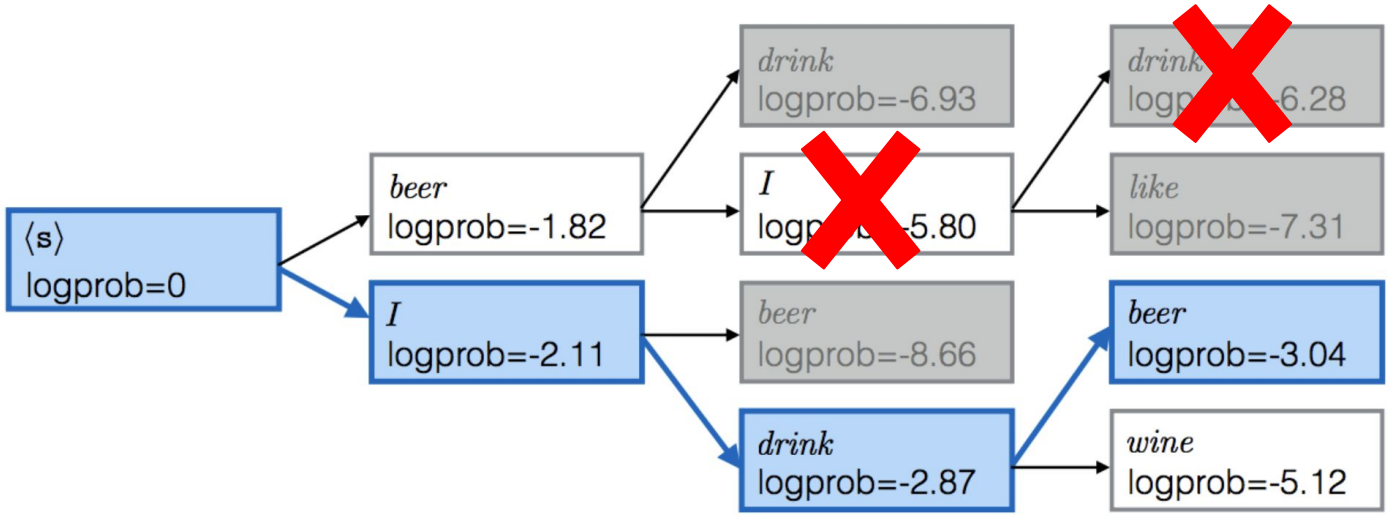
Search problem - Beam search

- tradeoff - we cannot look at all the combinations
- but we will look at a few at each step (beam = 2)
- improves solution quality



Avoid loops and repeating text

- repetition penalty (no_repeat_ngram_size)
- all ngrams of this size can only occur once



Deterministic sampling methods

- always pick max value - greedy/beam

- an alternative sampling is the probabilistic sampling - pick a random value

Probabilistic sampling

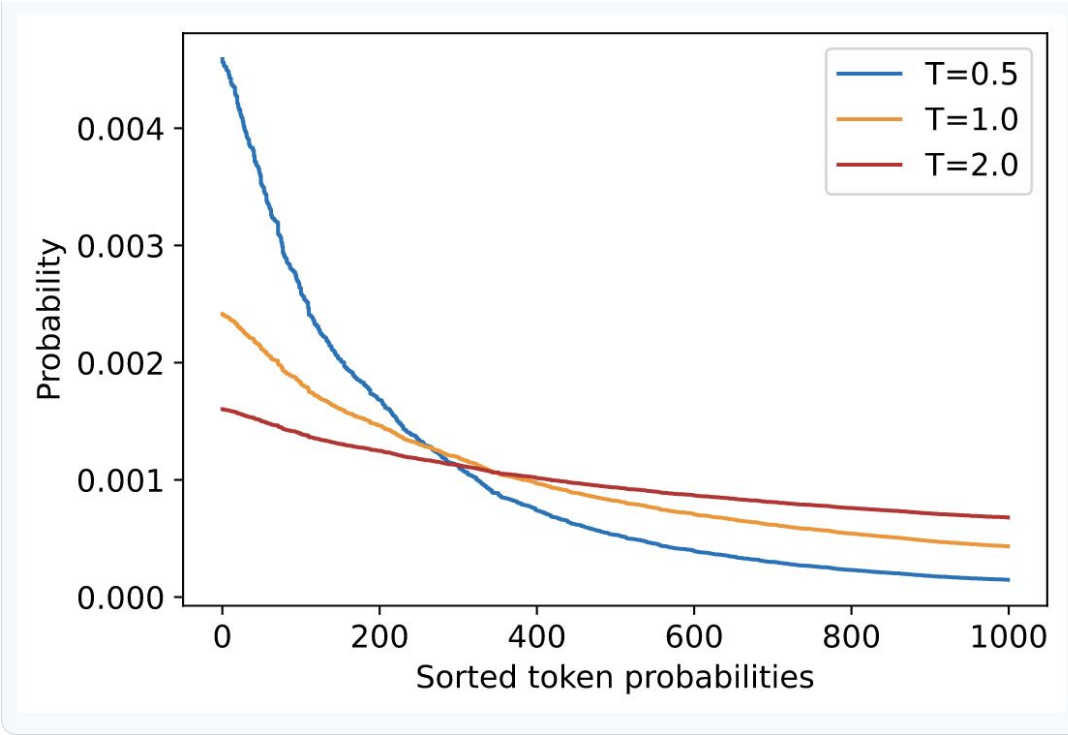
- Pick a token based on the output probabilities
- Rare tokens will be selected rarely
- How could we control this?

Temperature parameter

Rescale probabilities before sampling, control the shape of the distribution

T << 1 - peak for frequent tokens, suppressed rare tokens

T >> 1 - flatter distribution, rare tokens become more likely



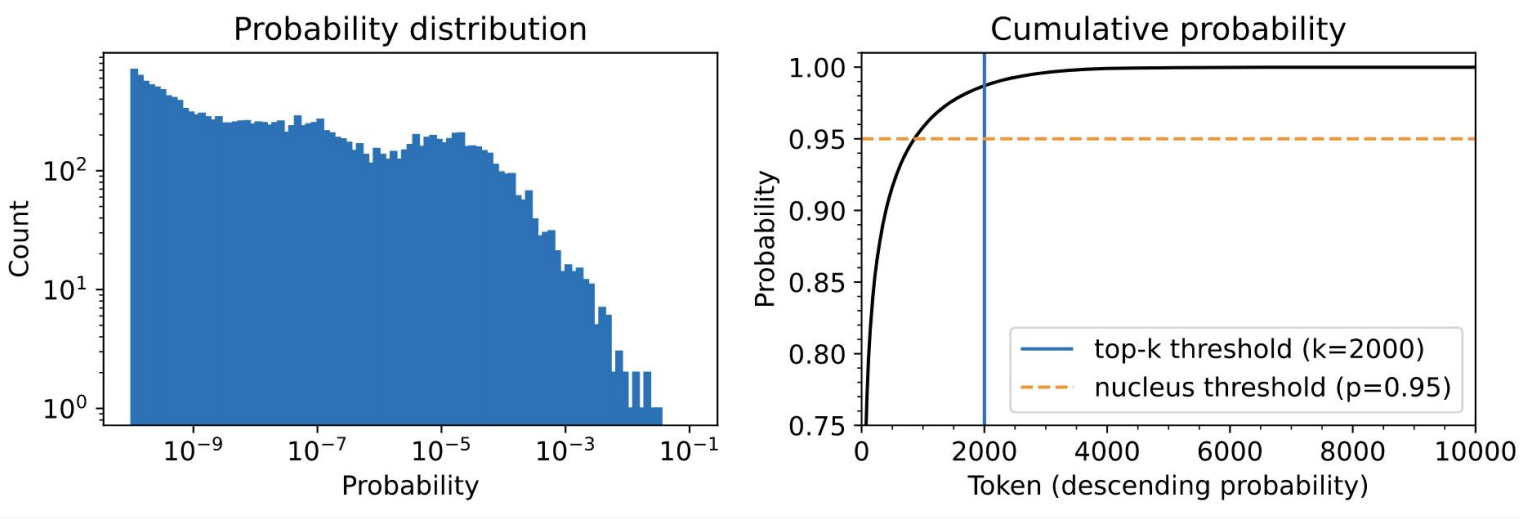
Top-k or Top-p (Nucleus) sampling

Restrict the number of possible tokens

When generating long text, even low probability tokens will appear - which we might want to avoid

Top-k or Top-p sampling

- Top-k
  - only choose from the k highest prob tokens, ignore bottom
- Top-p
  - only choose from top prob tokens, which sum to p, ignore bottom %



Tweet

near
@nearcyan

Try Using These More Polite Phrases In Your Workplace:

"you sound like a high-temperature LLM" -> "you are a very creative writer"
"your repetition penalty is too low" -> "you are great at emphasizing the key points"
"your top-k is at 1" -> "you are a stable and reliable person"

9:44 AM · Feb 27, 2023 · **15.2K** Views

Text generation challenges

- probabilistic output needs to be converted to text - decoding method

- quality and diversity of the generated text depends on the choice of decoding method and parameters

Examples - low vs high temperature

Találj ki egy jó szlogent egy új fagyizónak.

Examples - low vs high temperature

Találj ki egy jó szlogent egy új fagyizónak.

"A fagyi íze, amit soha nem felejtesz el!"

Examples - low vs high temperature

Találj ki egy jó szlogent egy új fagyizónak.

"A fagyi íze, amit soha nem felejtesz el!"

Találj ki egy jó szlogent egy új fagyizónak.

"Friss, fahéj illatú álmok - Fagyi Magic."

Demo

Instruction Tuned Language Model (Assistant Model)

Instruction/Chat dataset

written by people:

<USER>

Can you help me with this code? It seems like there is a bug.
print(“hello world)

<USER>

Can you help me with this code? It seems like there is a bug.
print(“hello world)

<ASSISTANT>

It looks like you forgot to close the string passed to the function print. You have to add a closing quote to properly terminate the string. Here is the corrected function:

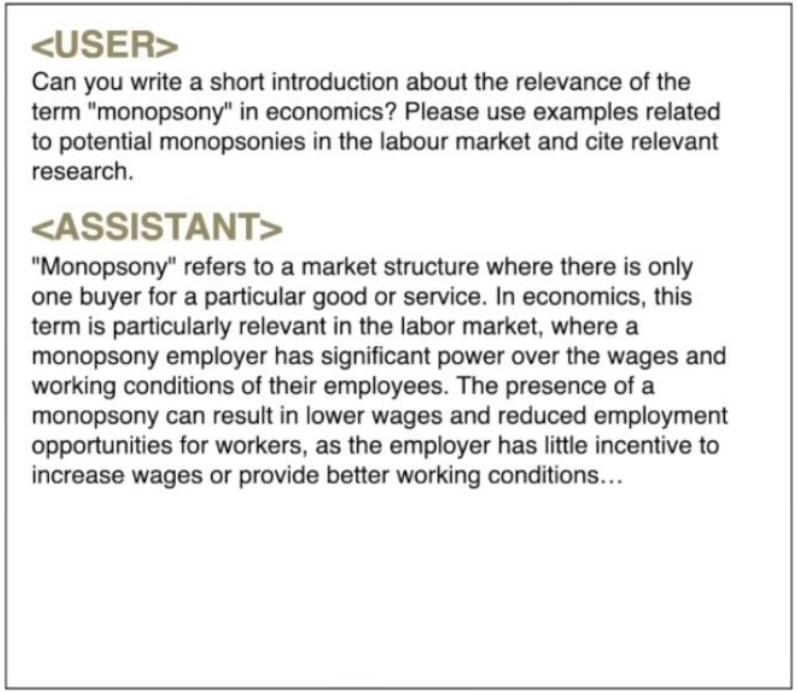
print(“hello world”)

Let me know if I can help with anything else!

Fine-tuning with new dataset

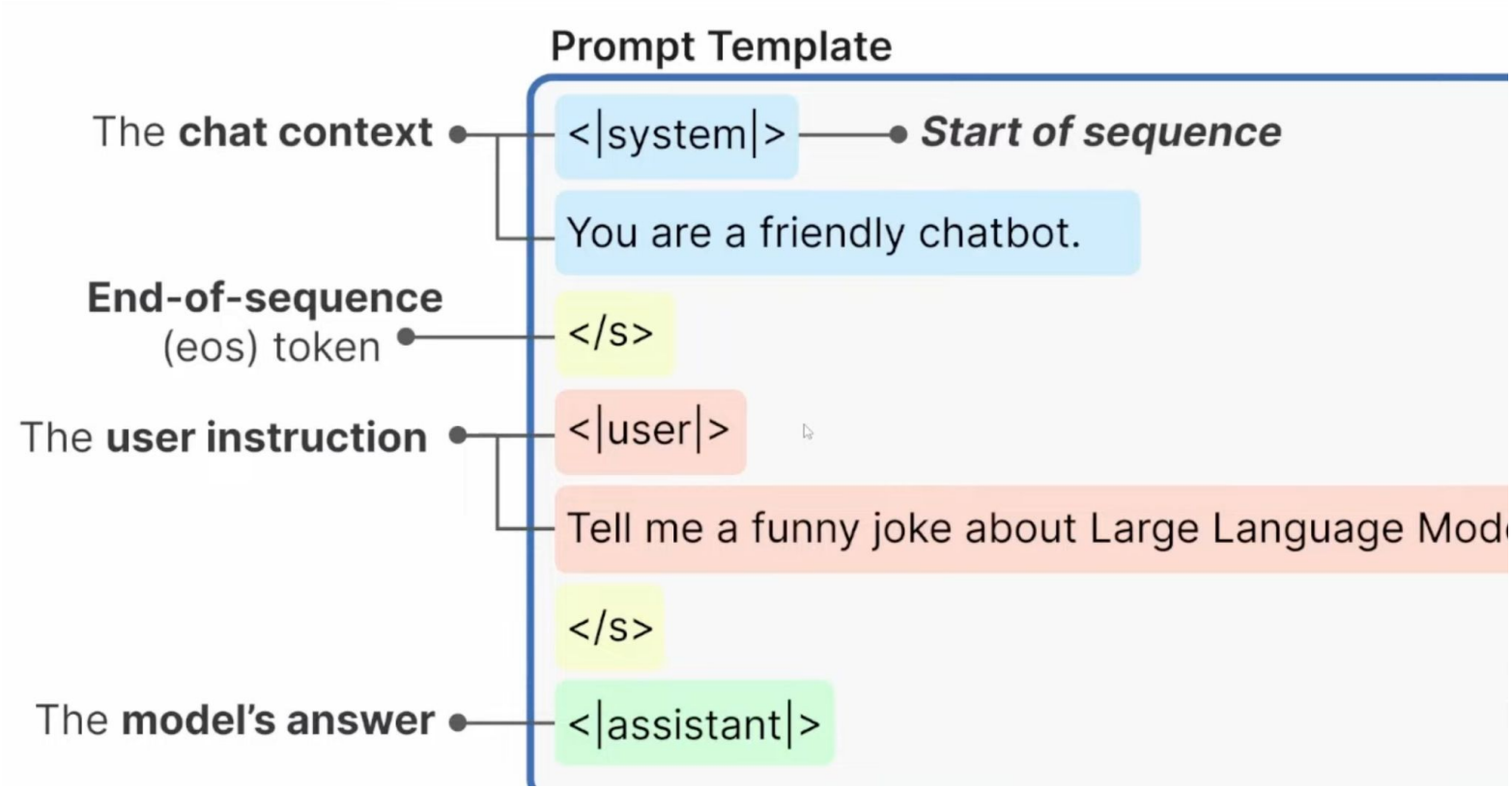
Model learns to follow the new format:

<USER> instructions
<ASSISTANT> answer



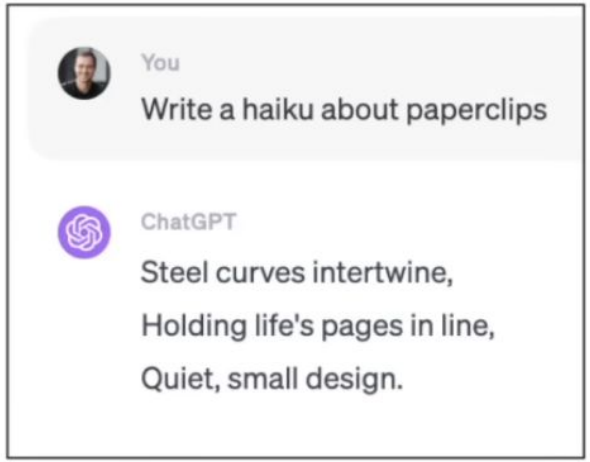
~100K conversations

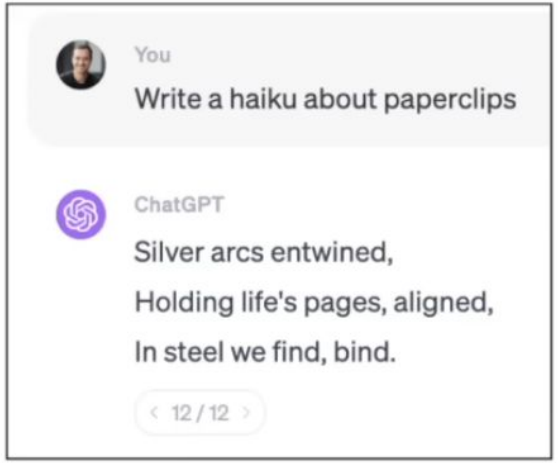
Assistant Prompting

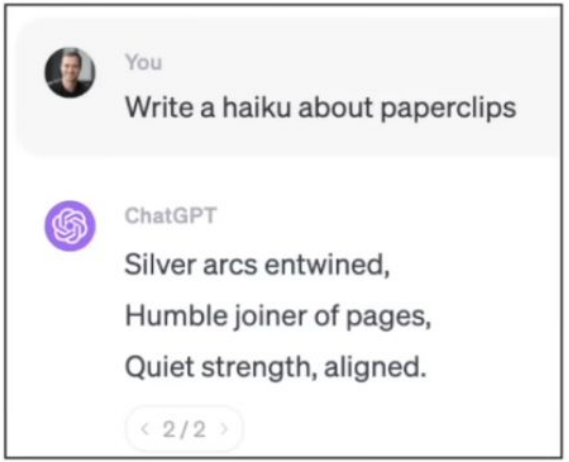


Comparison fine-tuning

Comparing is easier then coming up with an answer







Base Model vs Instruction Tuned Model

every ~year

Stage 1: Pretraining

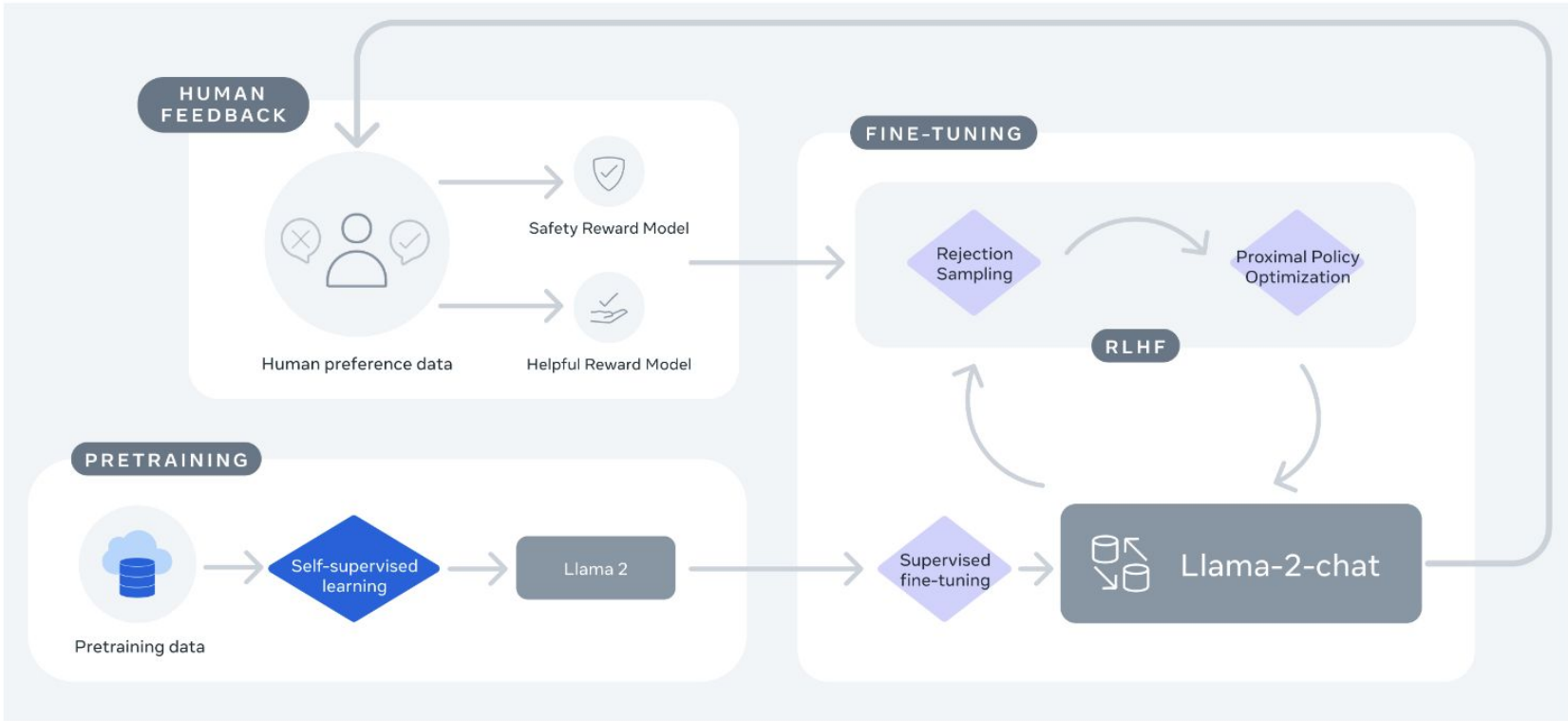
1. Download ~10TB of text.
2. Get a cluster of ~6,000 GPUs.
3. Compress the text into a neural network, pay ~$2M, wait ~12 days.
4. Obtain **base model**.

every ~week

Stage 2: Finetuning

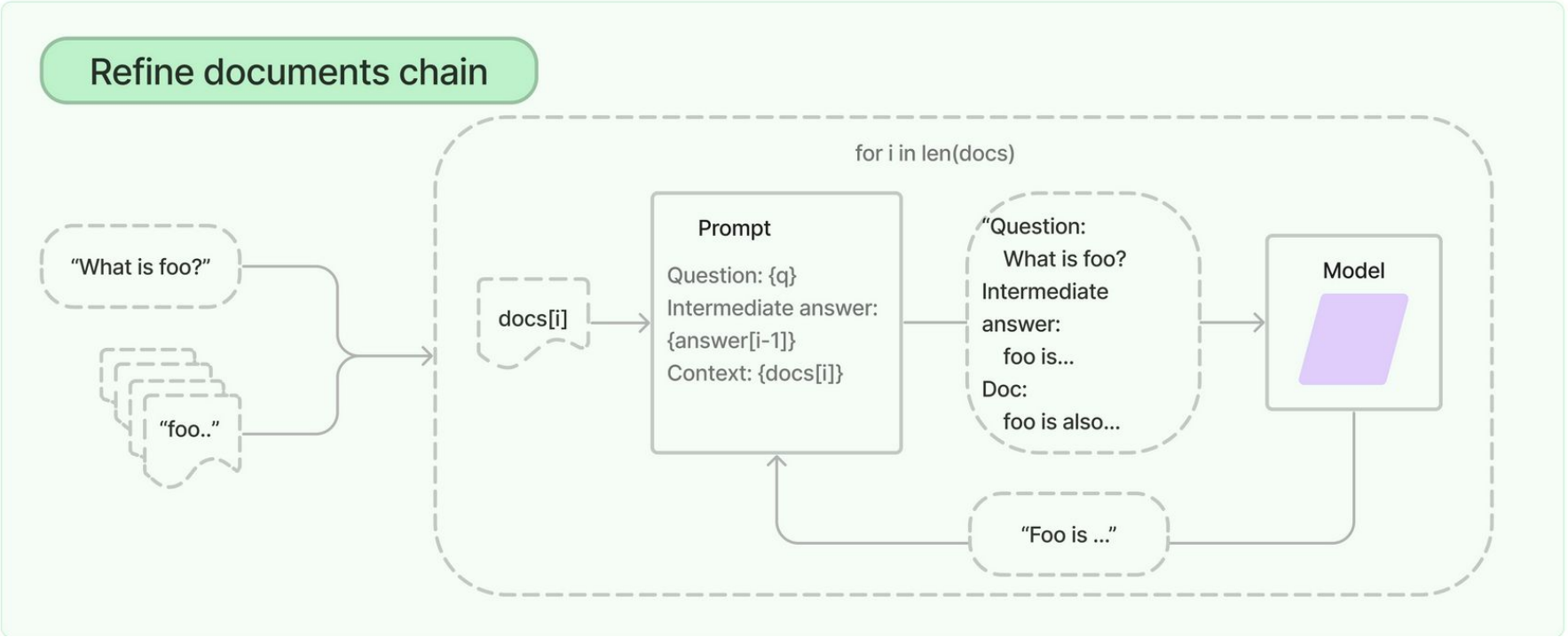
1. Write labeling instructions
2. Hire people (or use scale.ai!), collect 100K high quality ideal Q&A responses, and/or comparisons.
3. Finetune base model on this data, wait ~1 day.
4. Obtain **assistant model.**
5. Run a lot of evaluations.
6. Deploy.
7. Monitor, collect misbehaviors, go to step 1.

Llama2 vs Llama2-chat

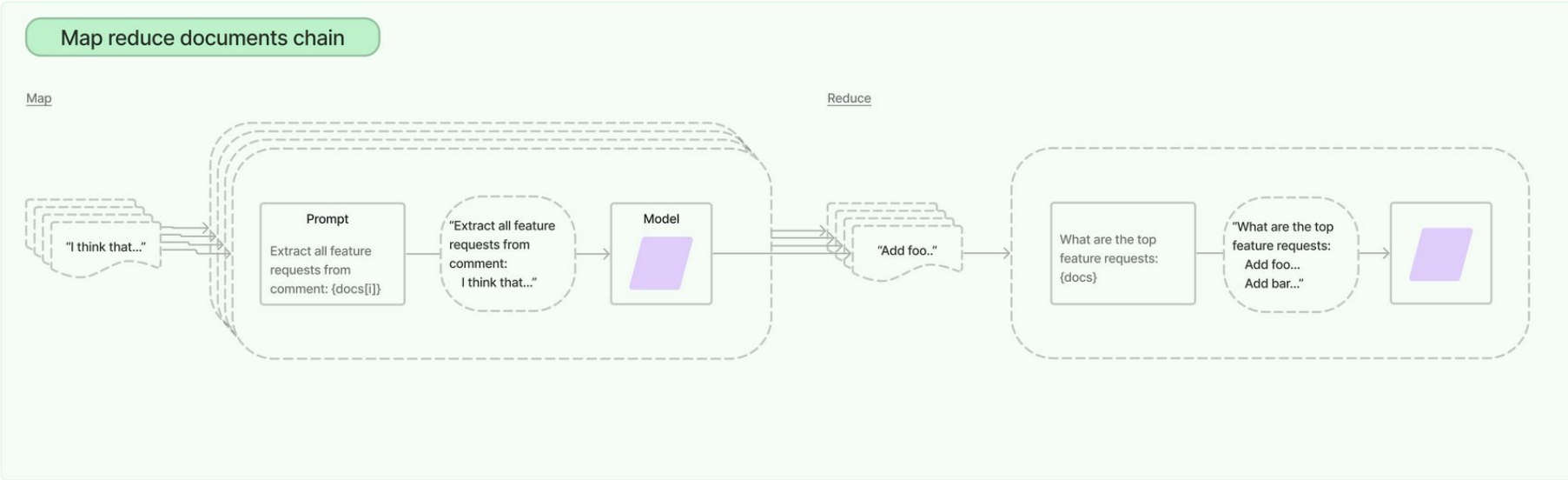


Demo

Ways to overcome context limit



Ways to overcome context limit



LLMs and the Hungarian language

Hallucinations

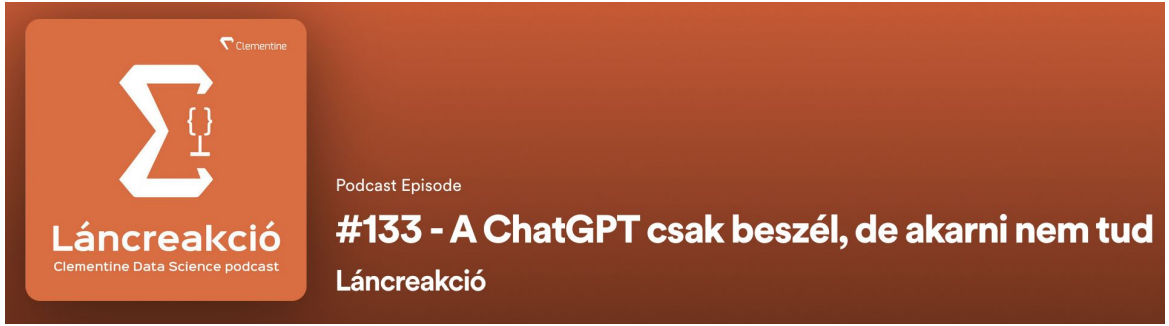
---

nyelvész, programtervező matematikus, tanár, kutató

~ 40 year of experience in computational linguistics

Nyelvtudományi Kutatóközpont igazgatója

Hallucinations



[Láncreakcio Podcast - Prószéky Gáborral]

22:00-26:25

Hallucinations

Language model

-> we can expect amazing language skills

-> we cannot expect great world understanding skills

-> anthropomorphizing

(i’ve asked ChatGPT a question vs
 i’ve asked ChatGPT to complete a given text)

“Art” of prompting

Tips:

- use delimiters
  ```, """, ---, <>
- specify structure
  \<name\>
  \`\`\`age here \`\`\`
  mark list items with 0/1
  {sentiment}
- few shot examples
- break down task into steps
- tldr;
- do not vs DO NOT

Reduce hallucinations:

- ask for sources
- link to information
- give context

Transforming text:

- tone changes(professional, concise)
- spell check/grammar
- expand
  ```

Puli / ParancsPuli

GPT ~ 120 million HU tokens

Puli ~ 50.000 million HU tokens (400X)

A legnagyobb adaton betanított magyar modell

Nyelvtudományi Kutatóközpont által létrehozott

https://juniper.nytud.hu/demo/puli

https://juniper.nytud.hu/demo/parancspuli

LLM with Web based UI

Web

---

https://github.com/oobabooga/text-generation-webui

Semantic Search
Retrieval Augmented Generation

RAG

---

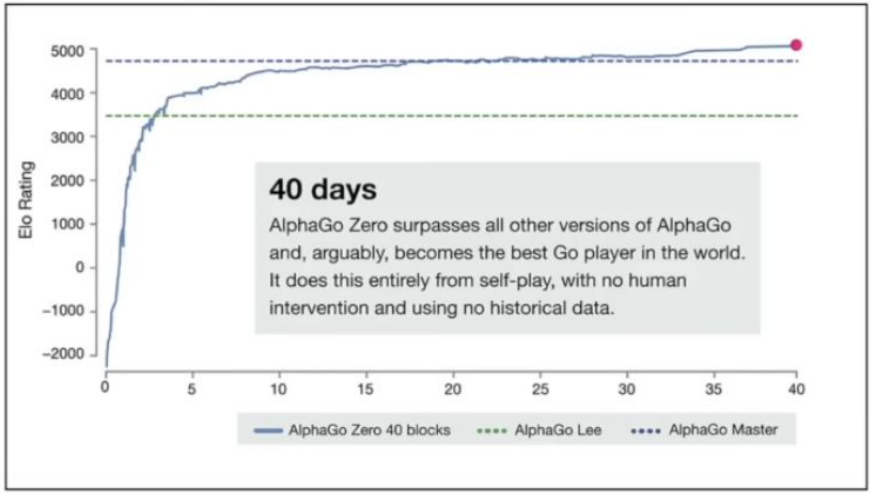
https://github.com/facebookresearch/llama-recipes/blob/main/demo_apps/RAG_Chatbot_example/RAG_Chatbot_Example.ipynb

Future

Human-AI Collaboration

- it’s always a mix
- it's another tool (calculator, browser)
- reddit ChatGPT real uses

Superhuman performance?



AlphaGo had two major stages:

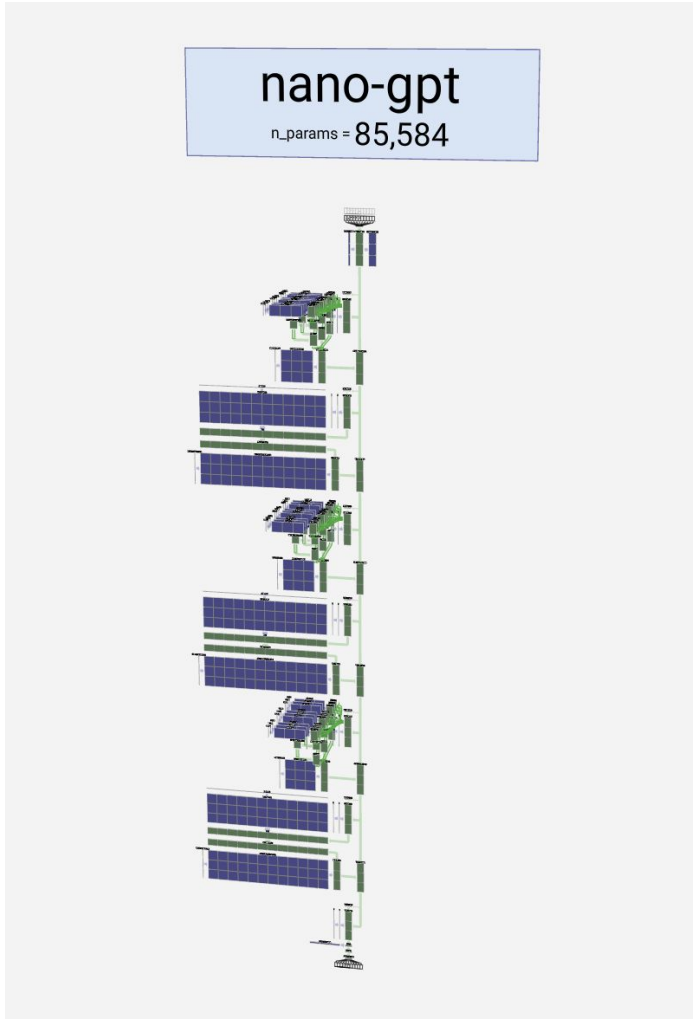
1. Learn by imitating expert human players
2. Learn by self-improvement (reward = win the game)

Superhuman performance?

- lack of reward criterion in open language modelling
    - it's so open, so subjective, so many different tasks
    - no simple reward function that can confidently tell you when the result is good or bad

- no way for self improvement
- no way to surpass human performance

LLM visualisation - Brendan Bycroft



References

References

- Andrej Karpathy - Intro to LLMs - https://www.youtube.com/watch?v=T-D1OfcDW1M
- DeepLearning.AI - Prompt engineering course https://www.deeplearning.ai/short-courses/chatgpt-prompt-engineering-for-developers/
- Huggingface - Intro to NLP https://huggingface.co/learn/nlp-course/chapter1/1
- Llama.cpp recipes - https://github.com/facebookresearch/llama-recipes