

SFSCON 2023

Self-hosted, Open Source Large Language Models

What are the most promising projects and how good are they?

Chris Mair - 1006.org

Large Language Models (LLM)



Large Language Models (LLM)

- Large Language Models (LLMs) are deep neural networks.
- The leading architecture is the **transformer** architecture introduced in the 2017 paper "*Attention Is All You Need*" by researchers from (mostly) Google [1]
- The models are first trained on a large corpus of text in natural language.
- If you feed text to such a **pre-trained** model, it will ...

... complete the text!

```
[$ ./main -m /data/models/TheBloke_Llama-2-13B-GGUF/llama-2-13b.Q6_K.gguf --color -n 2 -p "compare apples and"
Log start
main: build = 1407 (465219b)
main: built with cc (Debian 12.2.0-14) 12.2.0 for x86_64-linux-gnu
main: seed = 1698597551
llama_model_loader: loaded meta data with 19 key-value pairs and 363 tensors from /data/models/TheBloke_Llama-2-13B-GGUF/llama-2-13b.Q6_K.gguf (version GGUF V2 (latest))
llama_model_loader: - tensor 0: token_embd.weight q6_K [ 5120, 32000, 1, 1 ]
llama_model_loader: - tensor 1: blk.0.attn_norm.weight f32 [ 5120, 1, 1, 1 ]
llama_model_loader: - tensor 2: blk.0.ffn_down.weight q6_K [ 13824, 5120, 1, 1 ]
llama_model_loader: - tensor 3: blk.0.ffn_gate.weight q6_K [ 5120, 13824, 1, 1 ]
llama_model_loader: - tensor 4: blk.0.ffn_up.weight q6_K [ 5120, 13824, 1, 1 ]
llama_model_loader: - tensor 5: blk.0.ffn_norm.weight f32 [ 5120, 1, 1, 1 ]
llama_model_loader: - tensor 6: blk.0.attn_k.weight q6_K [ 5120, 5120, 1, 1 ]
llama_model_loader: - tensor 7: blk.0.attn_output.weight q6_K [ 5120, 5120, 1, 1 ]
llama_model_loader: - tensor 8: blk.0.attn_q.weight q6_K [ 5120, 5120, 1, 1 ]
llama_model_loader: - tensor 9: blk.0.attn_v.weight q6_K [ 5120, 5120, 1, 1 ]
llama_model_loader: - tensor 10: blk.1.attn_norm.weight f32 [ 5120, 1, 1, 1 ]
```

compare apples and

```
system_info: n_threads = 16 / 32 | AVX = 1 | AVX2 = 1 | AVX512 = 1 | AVX512_VBMI = 1 | AVX512_VNNI = 1 | FMA = 1 | NEON = 0 | ARM_FMA = 0 | F16C = 1 | FP16_VA = 0 | WASM_SIMD = 0 | BLAS = 0 | SSE3 = 1 | SSSE3 = 1 | VSX = 0 |
sampling:
  repeat_last_n = 64, repeat_penalty = 1.100, frequency_penalty = 0.000, presence_penalty = 0.000
  top_k = 40, tfs_z = 1.000, top_p = 0.950, typical_p = 1.000, temp = 0.800
  mirostat = 0, mirostat_lr = 0.100, mirostat_ent = 5.000
generate: n_ctx = 512, n_batch = 512, n_predict = 2, n_keep = 0

compare apples and oranges
llama_print_timings: load time = 366.01 ms
llama_print_timings: sample time = 0.24 ms / 2 runs ( 0.12 ms per token, 8230.45 tokens per second)
llama_print_timings: prompt eval time = 193.45 ms / 5 tokens ( 38.69 ms per token, 25.85 tokens per second)
llama_print_timings: eval time = 78.34 ms / 1 runs ( 78.34 ms per token, 12.77 tokens per second)
llama_print_timings: total time = 272.93 ms
Log end
$ █
```

compare apples and
oranges

How can we turn this into a **chat**bot? With a simple trick!

```
Transcript of a dialog, where the User interacts with an Assistant named Bob. Bob is helpful, kind, honest, good at writing, and never fails to answer the User's requests immediately and with precision.
```

```
User: Hello, Bob.
```

```
Bob: Hello. How may I help you today?
```

```
User: Please tell me the largest city in Europe.
```

```
Bob: Sure. The largest city in Europe is Moscow, the capital of Russia.
```

```
User: Do you know what the SFSCON is?
```

```
Bob: I am not familiar with the SFSCON.
```

```
User: And do you know what FOSDEM is?
```

```
Bob: FOSDEM is a free software developer event taking place in Brussels, Belgium.
```

```
User: █
```

It turns out, this get out of sync easily, so actual chatbots are base LLMs that have been further trained on chats (the so called fine-tuning for chats).

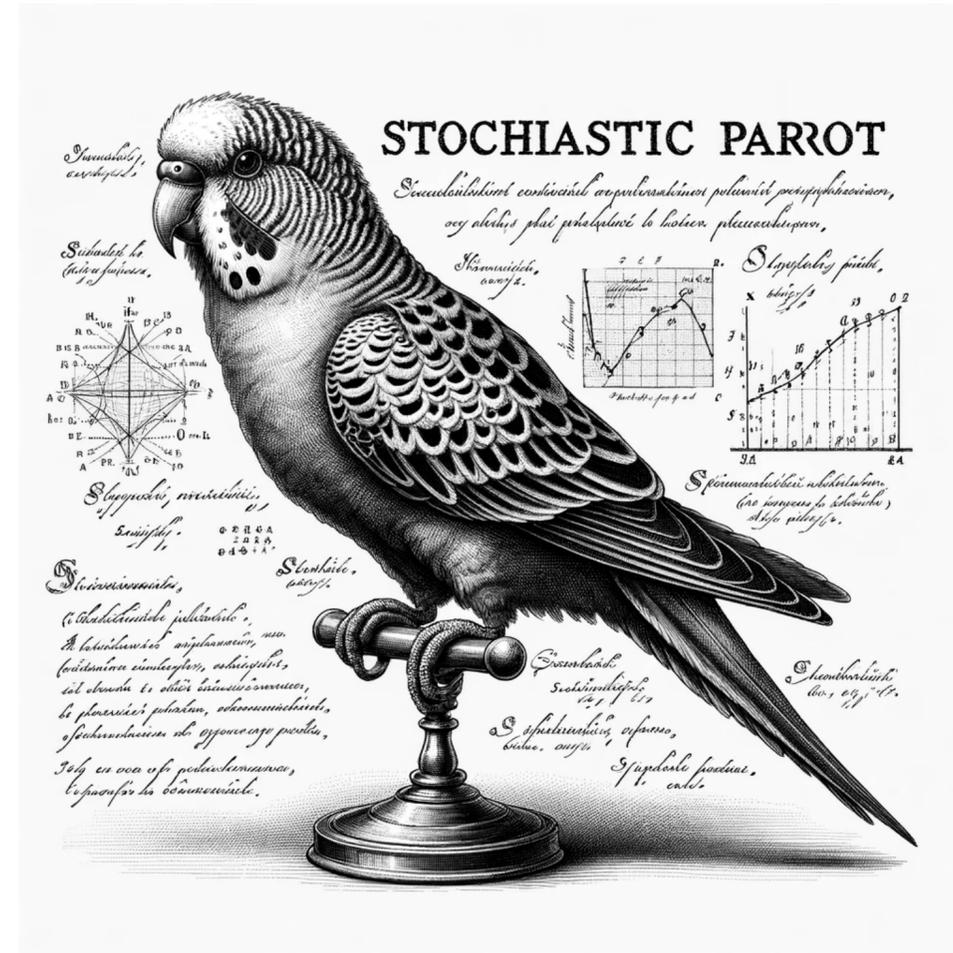
PS: the llama.cpp command to run this againa Llama-2-13B was:

```
./main -m /data/models/TheBloke_Llama-2-13B-GGUF/llama-2-13b.Q6_K.gguf -c 512 -b 1024 -n 256 --keep 48 --repeat_penalty 1.0 --color -i -r "User:" --in-prefix " " -f prompts/chat-with-bob.txt
```

OK, so is it true, what critics say? These models are just stochastic parrots that complete text based on the likelihood of certain words appearing in sequence?



Expectation.



Reality?

ChatGPT

- **Chat** Generative **Pre-trained Transformer**
- Leading LLM as a service by OpenAI
- GPT-1 (June 2018)
GPT-2 (November 2019)
GPT-3/3.5 (June 2020)
GPT-4 (March 2023)
- ChatGPT-4 is a lot more than a stochastic parrot!

Let's test ChatGPT-4!

The largest diamond ever cut (Source: Wikipedia) is the "Golden Jubilee" with a mass of 545.67 carats.

1 carat corresponds to a mass of 200 mg. The density is $\rho = 3520 \text{ kg/m}^3$.

Calculate the volume of this diamond!

Given:

Mass of the "Golden Jubilee" diamond $m = 545.67 \text{ carats}$

1 carat = 200 mg = 0.2 g

Density $\rho = 3520 \text{ kg/m}^3 = 3.52 \text{ g/cm}^3$

First, we'll find the mass of the diamond in grams:

$$m = 545.67 \text{ carats} \times 0.2 \text{ g/carat}$$

$$m = 109.134 \text{ g}$$

Next, we'll use the definition of density:

$$\rho = \frac{m}{V}$$

Where V is the volume. We can rearrange this equation to solve for V :

$$V = \frac{m}{\rho}$$

Substitute in the given values:

$$V = \frac{109.134 \text{ g}}{3.52 \text{ g/cm}^3}$$

$$V \approx 31.01 \text{ cm}^3$$

So, the volume of the "Golden Jubilee" diamond is approximately 31.01 cm^3 .

Correct reasoning!

Approximate arithmetics: there is a rounding error (ironically, LLMs can't compute).

There's no way this could be done with word completion.

The problem is an original, offline test question used at a local high school. It's not on the web (it will be, after the SFSCON :)

Given:

Mass of the "Golden Jubilee" diamond $m = 545.67$ carats

1 carat = 200 mg = 0.2 g

Density $\rho = 3520 \text{ kg/m}^3 = 3.52 \text{ g/cm}^3$

First, we'll find the mass of the diamond in grams:

$$m = 545.67 \text{ carats} \times 0.2 \text{ g/carat}$$

$$m = 109.134 \text{ g}$$

Next, we'll use the definition of density:

$$\rho = \frac{m}{V}$$

Where V is the volume. We can rearrange this equation to solve for V :

$$V = \frac{m}{\rho}$$

Substitute in the given values:

$$V = \frac{109.134 \text{ g}}{3.52 \text{ g/cm}^3}$$

$$V \approx 31.01 \text{ cm}^3$$

So, the volume of the "Golden Jubilee" diamond is approximately 31.01 cm^3 .

Is this AI?

- See the link for a nice introductory article [2]
- Here is an 25 page technical paper that tried to describe what is happening in GPT-2 (sic!) when it just completes a sentence with a single word [3].
- We don't really know why this works! There appear to be emergent abilities!
- This is very well discussed in the paper "Sparks of Artificial General Intelligence: Early experiments with GPT-4" [4].

Good. What's the problem then?

- ChatGPT (and its peers) are proprietary SaaS products. We want something less proprietary, something we can self-host.
- To reproduce such an LLM somebody would need to train a **large** deep neural network with a **large** corpus of curated text and human-assistant interactions for fine-tuning.
- The keyword is **large**:
- GPT-1 (June 2018) 117M parameters,
GPT-2 (November 2019) 1.5B parameters
GPT-3/3.5 (June 2020) 175B parameters
GPT-4 (March 2023) secret, estimated at > 1T parameters

- Training such large models costs lots of money.
- OpenAI's costs are not public knowledge, but as an example, the Open Source Falcon-40B model was trained on 384 Nvidia A100 (40GB) compute cards for two months. I estimate renting these in bulk cost \$1.5-\$2.0 per hour, so we're talking about an order of magnitude of \$1M to train Falcon-40B ^[5].
- Besides training cost, a potential open LLM project must also gather the corpus for base training and the chat interactions for fine-tuning.
- Ideally this material should not be encumbered by restrictive licenses so to avoid problems with later copyright violation claims.
- The good news is that there are projects that seek to solve these problems! Let's see where they stand today (Oct 2023).

Interesting Open LLM projects



RedPajama ^[6] - Apache2

"RedPajama is our set of leading open-source foundation models and datasets. This includes Apache 2.0 licensed base, chat, and instruction-tuned models and the largest-ever open pre-training dataset, which has been used to train over 100 models. We provide the data recipes, training code, and instruct datasets for complete transparency."

The biggest currently available is RedPajama-INCITE-7B-Chat ^[7]

StableLM ^[8] - CC BY-SA-4.0

This is StabilityAI's LLM initiative. It was announced in April 2023 ^[9], but seems somewhat stuck. The latest model is a 3B parameter model, but StabilityAI now is also developing another model that is a fork of Llama-2 ^[10].

Llama2 by Meta ^[11] - very permissive license

Llama was released in February 2023 to the research community, immediately leaked and spawned significant community interest with many community fine tuned derivatives. In July 2023 Meta releases Llama2 under a very permissive license. Llama2 comes in 7B, 13B and **70B** parameter models. As of this writing (Oct 2023) Llama and Llama2 models are the most used and supported by the community and many FOSS projects.

Falcon by The Technology Innovation Institute in Abu Dhabi ^[12] - Apache 2 up to 40B model

These models were dropped a bit as a surprise. A 7B and a **40B** parameter model were released in June 2023, at first under a restrictive license that was quickly changed to Apache 2. In September 2023 a **180B** (sic!) model was released, but kept the restrictive license.

We can self-host these, right?



Yes, we can, the computation for inference (applying the model) is not too costly. However, the keyword is again "**large**".

Llama2 at 70B parameters as 16 bit floats is ~ 140 GB of memory. Falcon 40B is ~ 80 GB. As a comparison, you can do face recognition with a deep NN of a size of 21 MB (sic!).

Normally inference with large models is done on GPUs, but these models far exceeds the GPU-RAM available in affordable (consumer) cards.

Three strategies:

- rent expensive Nvidia compute cards (or buy them at a few \$10k a piece)
- use only smaller models (that don't perform well)
- run the models on CPU, so they can be loaded onto normal RAM

I picked the CPU/RAM strategy!

The good news is there is a great piece of FOSS: Georgi Gerganov's llama.cpp ^[13] is a C++ program with no dependencies (sic!) that can run inference on various LLMs (not only Llama, despite its name) in all kind of formats on all kind of hardware including GPUs and CPUs!

Georgi Gerganov has also introduced a standard format the various models can be converted to: GGUF. The format can also store model parameters in compressed representations where each uses just a few bits! Typically it is possible to run the models in 8, 6 or even just 4 bits with no big loss in quality.

For example, Llama2-70B at 6 bits can be run in 54 GB of RAM.

There's more good news!

The place where people host their models is **Hugging Face** ^[14]. They're to models what GitHub is to source code.

Tom Jobbins ("TheBloke") has a large collection of free models, already compressed (*quantized*) to varying bit-sizes and converted to llama.cpp's .GGUF format. Here are the links to Llama2 and Falcon models:

<https://huggingface.co/TheBloke/Llama-2-7b-Chat-GGUF>

<https://huggingface.co/TheBloke/Llama-2-13B-chat-GGUF>

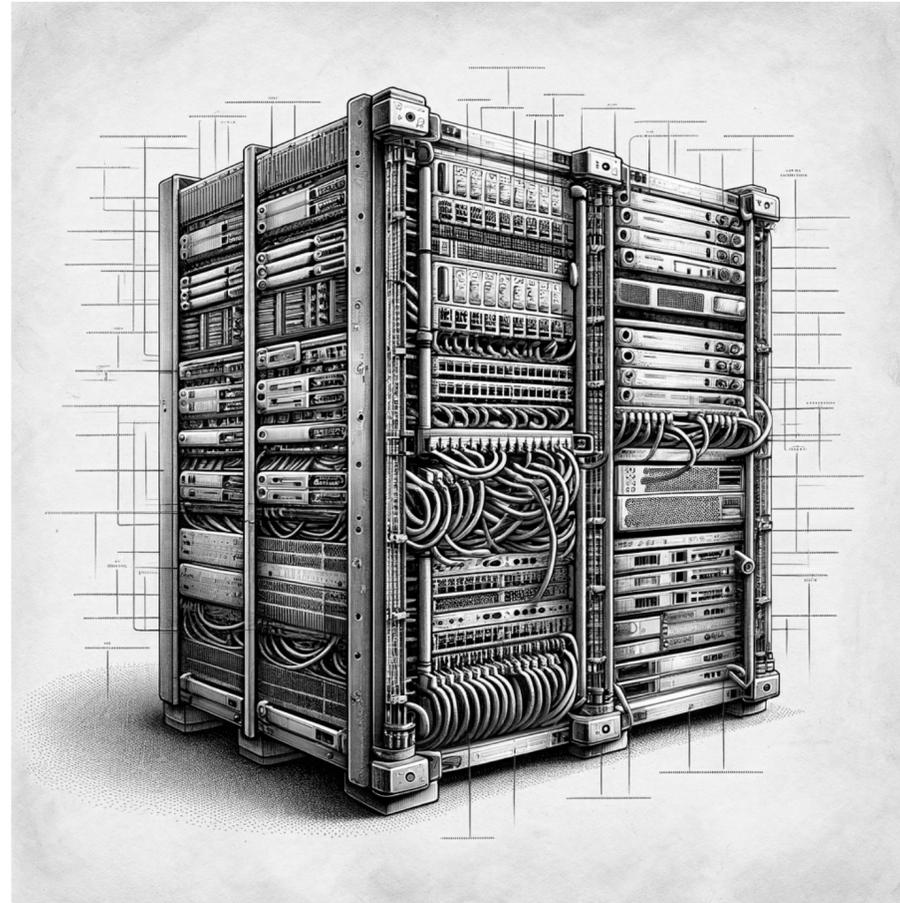
<https://huggingface.co/TheBloke/Llama-2-70B-chat-GGUF>

<https://huggingface.co/tiiuae/falcon-40b-instruct> (original pytorch format)

<https://huggingface.co/TheBloke/Falcon-180B-Chat-GGUF>

As I'm writing this, converted Falcon 40B was not available, I converted it manually ^[15] using llama.cpp.

Let's run the models!



After not so good experience with consumer hardware, I rented a VM with 16 cores of an EPYC 9xx4 ("Genoa") with 128 GB of RAM. This is easily enough to run even the Falcon 180B model at 4 bit at reasonable speed.

First, I tried some **chatting** with Llama2 and Falcon.

For Llama2 models I called llama.cpp as:

```
./main -m model.gguf --color --interactive-first
```

For Falcon models, interactive mode didn't work, I needed to construct the chat interaction with multiple invocation where I passed the file with all previous questions and answers, like this:

```
User: Q1
```

```
Falcon: A1
```

```
User: Q2
```

```
Falcon:
```

The command was then:

```
./main -m model.gguf --temp 0.4 --color -f file
```

llama-2-70b-chat.Q6_K.gguf

What's the capital of France?

Answer: The capital of France is Paris.

What's the second largest city?

Answer: The second largest city in France is Marseille.

Tell me a few facts about these cities?

Answer: Sure, here are some interesting facts about Paris and Marseille:

[lots of facts, all correct]

Can you translate those facts into German?

[gives a sort of correct translation with some grammar errors, example extract:]

The city is known for its seafood cuisine, particularly bouillabaisse, a hearty fish stew that's a specialty of the region.

Die Stadt ist bekannt für ihre Meeresfrüchteküche, insbesondere den

Bouillabaisse, einen herzhaften Fischsuppen, der in der Region eine Spezialität ist.

falcon-40b-instruct-q4_0.gguf

User: What's the capital of France?

Falcon: I'm sorry, but as an AI language model, I don't have access to real-time information. However, the current capital of France is Paris.

User: What's the second largest city?

Falcon: The second largest city in France is Marseille.

User: Tell me a few facts about these cities?

Falcon: Paris is known as the City of Light and it is home to many famous landmarks such as the Eiffel Tower, Notre-Dame Cathedral, and the Louvre Museum. It has a population of around 12 million people. Marseille is located in the south of France and is the second largest city with a population [...]

User: Can you translate those facts into German?

[mostly correct, even though style is poor, example extract:]

Paris ist die Stadt des Lichts und beherbergt viele berühmte Sehenswürdigkeiten wie den Eiffelturm, Notre-Dame Kathedrale, und das Louvre Museum.

falcon-180b-chat.Q4_K_M.gguf

This didn't work well. Falcon tends to continue the conversation, role-playing as the user! E.g. it starts talking to itself.

This is likely some prompting problem or maybe there was a problem with fine tuning this model for chat?

Then I tried to submit the **test problem**! This is the recommended prompt context for Llama2:

[INST] <<SYS>>

You are a helpful, respectful and honest assistant. Always answer as helpfully as possible, while being safe. Your answers should not include any harmful, unethical, racist, sexist, toxic, dangerous, or illegal content. Please ensure that your responses are socially unbiased and positive in nature. If a question does not make any sense, or is not factually coherent, explain why instead of answering something not correct. If you don't know the answer to a question, please don't share false information.

<</SYS>>

The largest diamond ever cut (Source: Wikipedia) is the "Golden Jubilee" with a mass of 545.67 carats. 1 carat corresponds to a mass of 200 mg. The density is $\rho = 3520 \text{ kg/m}^3$. Calculate the volume of this diamond![/INST]

and for Falcon:

User: The largest diamond ever cut (Source: Wikipedia) is the "Golden Jubilee" with a mass of 545.67 carats. 1 carat corresponds to a mass of 200 mg. The density is $\rho = 3520 \text{ kg/m}^3$. Calculate the volume of this diamond!

Falcon:

llama-2-70b-chat.Q6_K.gguf

In three runs, Llama2 was twice completely wrong and found once the correct steps but miscalculated.

falcon-40b-instruct-q4_0.gguf

In three runs, Falcon never got this right.

falcon-180b-chat.Q4_K_M.gguf

In three runs, Falcon always found the correct approach, calculated correctly twice and miscalculated once! Here is a short and correct answer (except for the usual poor precision in arithmetic common to all LLMs):

The volume of a body is given by $V = m/\rho$, with m the mass and ρ the density.

The mass of the diamond is 545.67 carats times 200 mg/carat = 109.13 g.

The volume is $V = 109.13 \text{ g} / 3520 \text{ kg/m}^3 = 3.097 \times 10^{-5} \text{ m}^3$.

Model evaluation benchmarks

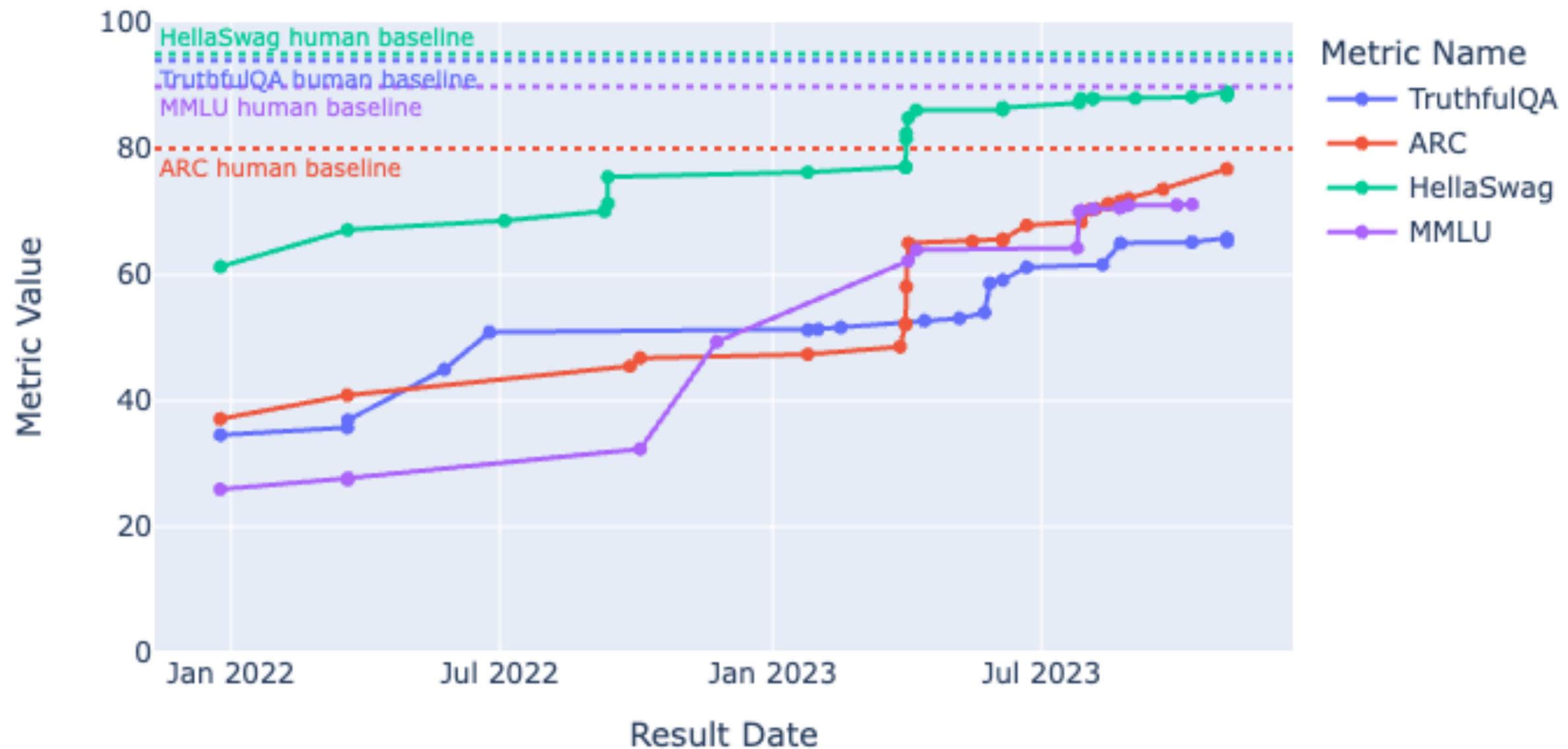
According to the authors of each of the recent models, their model beat all the previous ones!

- Falcon 40B in June 2023 ^[17]
- Llama-2 in July 2023 ^[18]
- Falcon 180B in September 2023 ^[19]

Hugging Face also has a leaderboard ^[16] that ranks LLMs according to their benchmark performance in various benchmarks. Llama-2 70B and its derivatives seem to be the clear winner, but Falcon 180B data is not yet available....

Here is an interesting graph from the leaderboard showing the crazy pace of development during 2023 so far!

Top Scores and Human Baseline Over Time



Conclusion

The **smaller initiatives** such as RedPajama are limited by their size. Larger training runs have been announced.

At the end of Oct 2023 **Llama-2 70B** has the most community recognition, the most derivatives and is best supported by various FOSS projects. It has a very permissive, though technically not FOSS license.

Falcon 40B is interesting as it seems to come close to Llama-2 while providing the extra peace of mind of the true FOSS license. It has less community recognition (yet) though.

Falcon 180b is really new and I find the chat version didn't work well for me. It also has a more restrictive license. On the other hand it is by far the largest model available to be run locally.

I find the easiest way to run different models is the **llama.cpp** project by Georgi Gerganov.

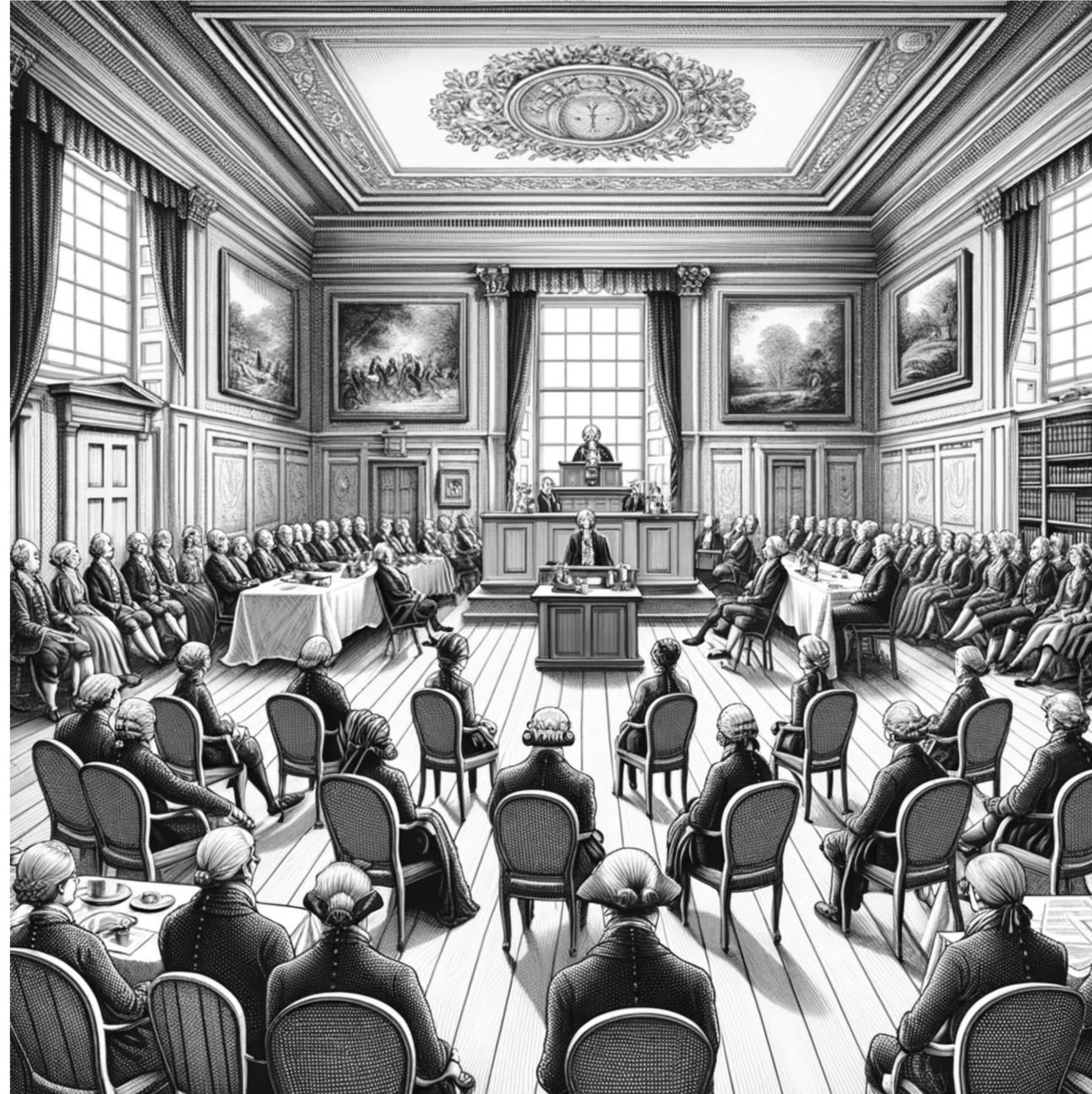
Outlook

The trend now is to integrate these models into one's products and services in such a way the models can query arbitrary documents, not just make use of their pre-trained knowledge.

The hot buzzword is *Retrieval Augmented Generation* (RAG).

There will be a Developer's Thursday Talk at NOI on Feb 1, 2024 ^[20] where I present some experiments with the models seen today and RAG. So stay tuned!

Thanks :)



Links 1/2

- [1] <https://arxiv.org/abs/1706.03762>
- [2] <https://arstechnica.com/science/2023/07/a-jargon-free-explanation-of-how-ai-large-language-models-work/>
- [3] <https://arxiv.org/abs/2211.00593>
- [4] <https://arxiv.org/abs/2303.12712>
- [5] <https://huggingface.co/tiiuae/falcon-40b>
- [6] <https://together.ai/models>
- [7] <https://huggingface.co/togethercomputer/RedPajama-INCITE-7B-Chat>
- [8] <https://github.com/stability-AI/stableLM/>
- [9] <https://stability.ai/blog/stability-ai-launches-the-first-of-its-stablelm-suite-of-language-models>
- [10] <https://stability.ai/blog/stable-beluga-large-instruction-fine-tuned-models>

Links 2/2

- [11] <https://ai.meta.com/llama/>
- [12] <https://huggingface.co/blog/falcon>
- [13] <https://github.com/ggerganov/llama.cpp>
- [14] <https://huggingface.co/>
- [15] <https://github.com/ggerganov/llama.cpp/pull/2717>
- [16] [https://huggingface.co/spaces/HuggingFaceH4/open llm leaderboard](https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard)
- [17] <https://huggingface.co/blog/falcon#evaluation>
- [18] <https://ai.meta.com/resources/models-and-libraries/llama/>
- [19] <https://huggingface.co/blog/falcon-180b#how-good-is-falcon-180b>
- [20] <https://noi.bz.it/en/events>

