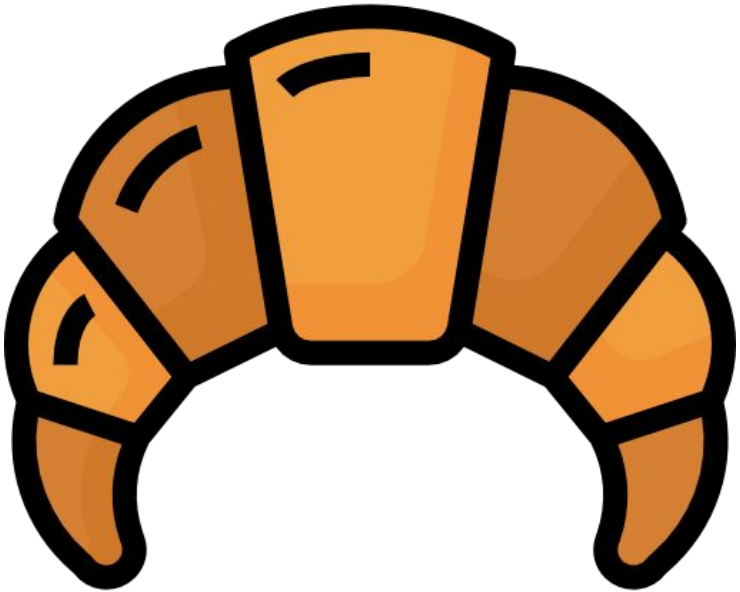


How To Train Your LLM: CroissantLLM Case Study

14/05/2024

Slides: Manuel Faysse
CentraleSupélec, Illuin Technology

An Industrial and Academic Partnership

CroissantLLM: A Truly Bilingual French-English Language Model

Manuel Faysse[1,5] **Patrick Fernandes**[6,8,11] **Nuno M. Guerreiro**[2,5,6,8]
António Loison[1] Duarte M. Alves[6,8] Caio Corro[9] Nicolas Boizard[4,5]
João Alves[2] Ricardo Rei[2,7,8] Pedro H. Martins[2] Antoni Bigata Casademunt[10]
François Yvon[9] André F.T. Martins[2,6,8] Gautier Viaud[1] Céline Hudelot[5]
Pierre Colombo[3,5]

CentraleSupélec ILLUIN TECHNOLOGY IST TÉCNICO LISBOA Unbabel | cnrs INSTITUT DU DÉVELOPPEMENT ET DES RESSOURCES EN INFORMATIQUE SCIENTIFIQUE

Manuel Faysse
CentraleSupélec, MICS
Illuin Technology
Doctorant CIFRE

Pierre Colombo
CentraleSupélec, MICS
Maître de conférence

Céline Hudelot
CentraleSupélec, MICS
Directrice du MICS

Nicolas Boizard
CentraleSupélec, MICS
Diabolocom
Doctorant CIFRE

Outline of the talk

- Why train a LLM ?
- Architecture
- Data
- Evaluation
- Downstream Use
- CroissantLLM: Applications & Learnings

Outline of the talk

- **Why train a LLM ?**
- Architecture
- Data
- Evaluation
- Downstream Use
- CroissantLLM: Applications & Learnings

Why should I train a LLM ?

Should you really ?

Other options include...

- Not using a LLM
- API-based access to good performing models
- Open-source models
  - 0-shot usage
  - Finetuning on specific use cases
  - Continued Pretraining
    - Sabias, ClaireLLM, etc...
    - Upscaling

Why should I train a LLM ?

Some potential reasons

- You have investor cash and a good pitch
  - Mistral
  - Yi
- You are a top tech company and don't want to be left behind
  - xAI
  - Meta
  - Google
- You have government funding and a good research / strategic story
  - BigScience
  - FinGPT
  - Jais
  - **CroissantLLM**

A Research project with industrial aims

Research

CroissantLLM is a research project aiming to study how **bilingualism** impacts language model pretraining and performance.

Industry

The final model is designed to be **small enough** to run on local hardware, but **good enough** to run generative tasks that are often reserved to larger models (**inference-optimal** training). It is trained on **permissively licensed data** only.

Open-Source

This project is rooted in open-source, with openly released models, data, code bases and evaluation benchmarks, enabling **researchers and practitioners** to benefit from it.

Outline of the talk

- Why train a LLM ? ✅
- **Architecture**
- Data
- Evaluation
- Downstream Use
- Case study: CroissantLLM

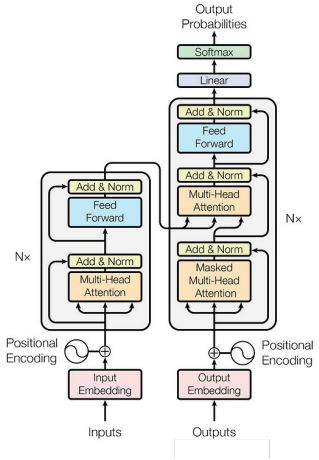
Architecture

- Most generative models are **decoder** transformers
  - Easier to scale and train
  - Predictable scaling laws
  - Efficient training strategy !

- Recent work has started showing promising results when scaling other architectures:
  - State Space models:
    - Mamba / Striped Hyena
  - RNNs
    - RWKV
  - Retentive Networks

BERT

Encoder



GPT

Decoder

When cash is low, go safe ?

Architecture

Tradeoffs...

How **big** do you want your model ?

How **good** do you want your model ?

How **expensive to train** do you want your model ?

How **expensive to run** / fast do you want your model ?

How much training **data** do you have ?

Chinchilla Scaling Laws: Modeling model / data size tradeoffs

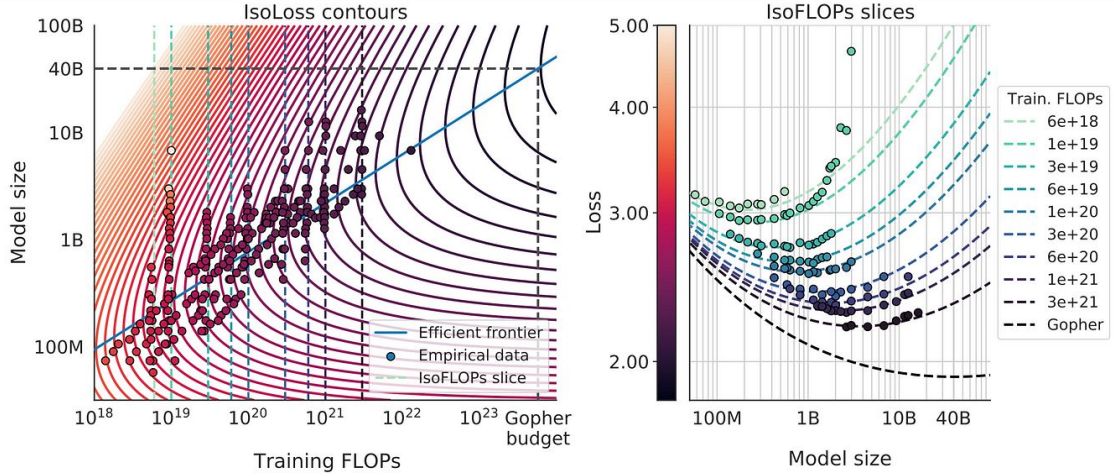
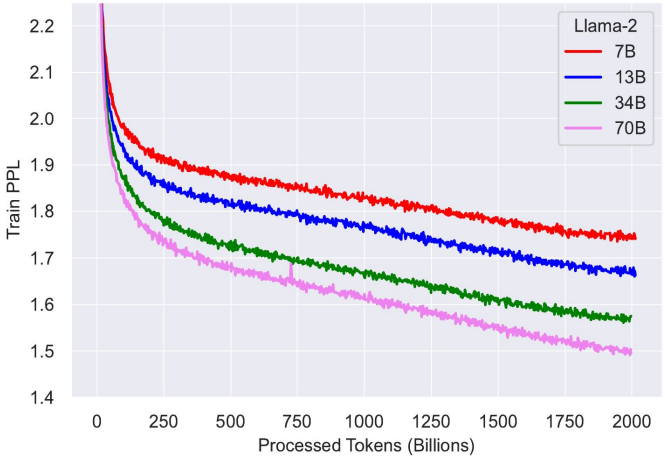


Figure 4 | **Parametric fit.** We fit a parametric modelling of the loss $\hat{L}(N, D)$ and display contour (**left**) and isoFLOP slices (**right**). For each isoFLOP slice, we include a corresponding dashed line in the left plot. In the left plot, we show the efficient frontier in blue, which is a line in log-log space. Specifically, the curve goes through each iso-loss contour at the point with the fewest FLOPs. We project the optimal model size given the *Gopher* FLOP budget to be 40B parameters.



$$\hat{L}(N, D) \triangleq E + \frac{A}{N^{\alpha}} + \frac{B}{D^{\beta}}.$$

https://arxiv.org/abs/2203.15556

https://arxiv.org/abs/2307.09288

Note: Recently criticized

Chinchilla Scaling Laws: Modeling model / data size tradeoffs

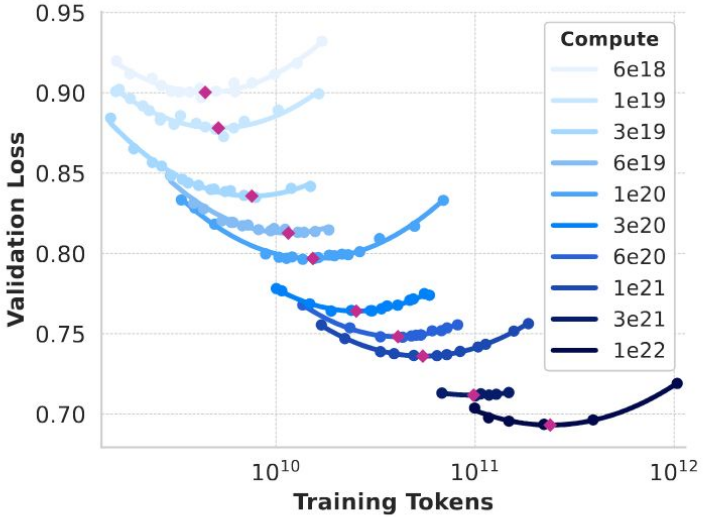


Figure 2 Scaling law IsoFLOPs curves between 6×10^{18} and 10^{22} FLOPs. The loss is the negative log-likelihood on a held-out validation set. We approximate measurements at each compute scale using a second degree polynomial.

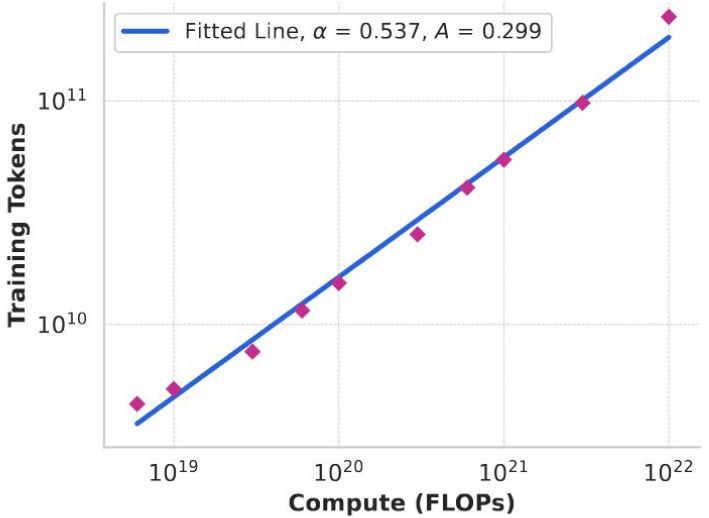


Figure 3 Number of training tokens in identified compute-optimal models as a function of pre-training compute budget. We include the fitted scaling-law prediction as well. The compute-optimal models correspond to the parabola minimums in Figure 2.

The Model (Chinchilla tradeoffs)

Generative models are often transformer *decoders* (GPT, Mistral, LLaMa) which performance is closely related to **(1) the # of model parameters**, and **(2) the # of training tokens**.

Training the best model given a fixed compute budget

For a given compute budget, there is an optimal ratio between parameter count and training data size **(~20 for Chinchilla Scaling laws)**

OU

Training the best model of a fixed size

By training longer than the Chinchilla ratio, we continue to improve the model but performance gains are increasingly costly.

We decide to overtrain a small model (1,3B) → 2307 token:param ratio vs 20 Chinchilla-optimal.

Lighter

Faster

Capable

Training Cost

CroissantLLM: Model Architecture

- 1.3B parameter decoder model, heavily optimized for industrial usability
- 7B Llama is the most downloaded model while not the best !
- For equivalent training compute, Large models are better, but slower and costlier to run inference with (Chinchilla laws)

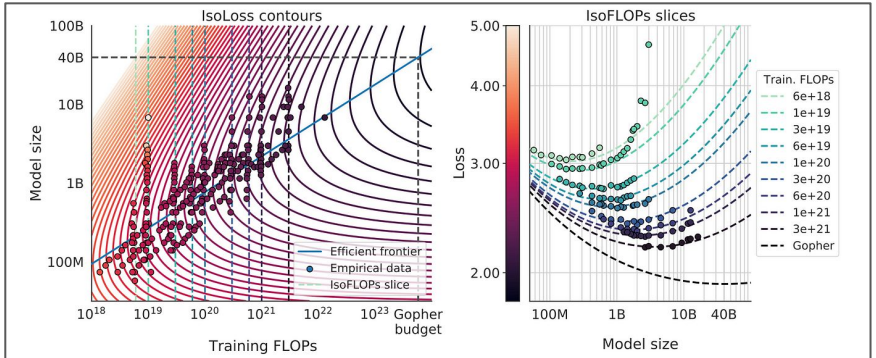
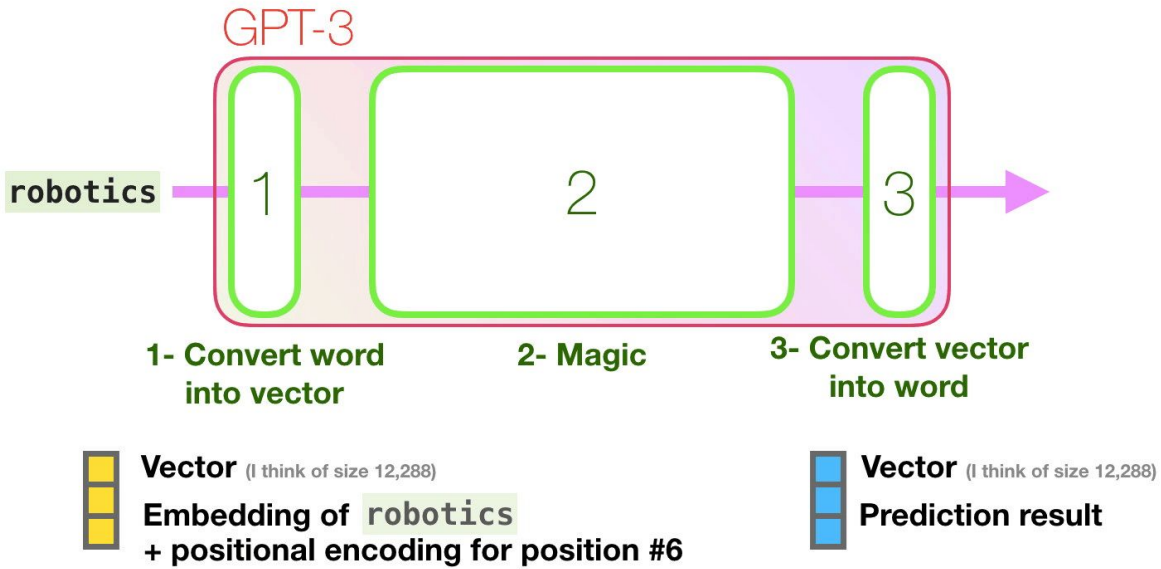


Figure 4 | **Parametric fit.** We fit a parametric modelling of the loss $\hat{L}(N, D)$ and display contour (**left**) and isoFLOP slices (**right**). For each isoFLOP slice, we include a corresponding dashed line in the left plot. In the left plot, we show the efficient frontier in blue, which is a line in log-log space. Specifically, the curve goes through each iso-loss contour at the point with the fewest FLOPs. We project the optimal model size given the *Gopher* FLOP budget to be 40B parameters.

Idea: Training a smaller model for longer:

- Pure performance will not be optimal, but usability will
- Easier to serve, finetune, adapt, run on CPUs
- Surpass the performance of other models in its size category

Tokenization: A Quick Recap

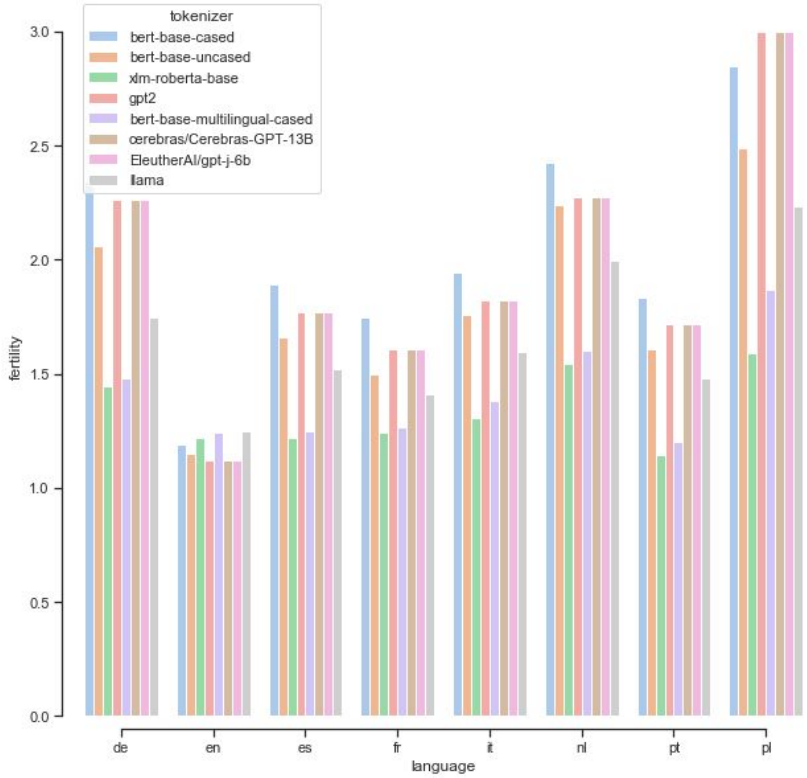


Tokenization: Information - theoretic optimal subword splitting

Tokens	Characters
26	105

Je m'appelle Guillaume et je suis un plombier experimenté
My name is John and I am an experienced plumber

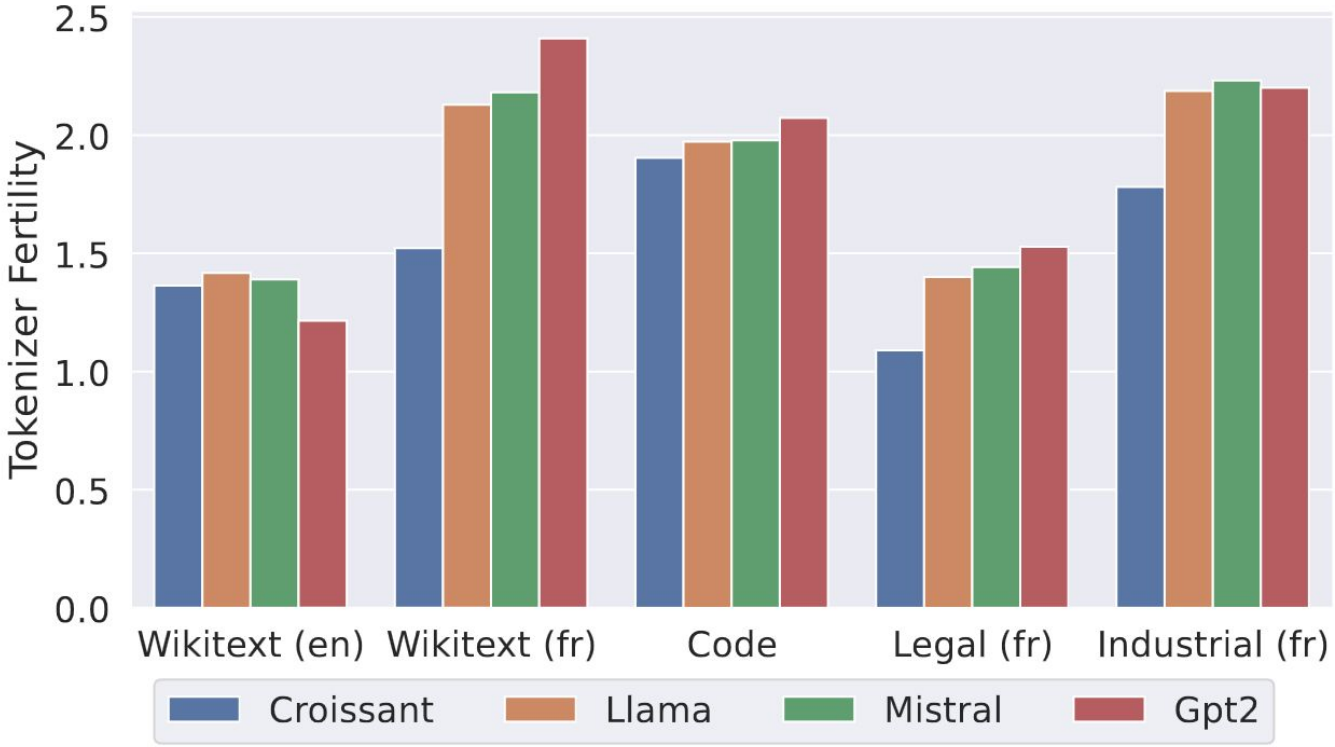
TEXT TOKEN IDS



If you fit a tokenizer on mostly english data, subword splitting will be optimized for english…

- Slower in other languages
- Less semantic coherence → lower performance in other languages

CroissantLLM: Tokenizer

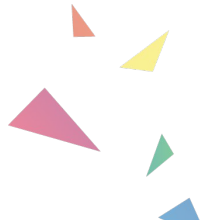


Fitted on a balanced corpus with Code, English and French sampled from our training corpus

Tokenization (Extra): Lots of options impact downstream usage

Design Choices

- Various Info-Theoretic Choices
  - Byte-Pair Encoding
  - Unigram
- SentencePiece / WordPiece
- Byte Fallback
- Digit splitting
- Whitespace tokenization
- **Vocab size**
- Corpus to fit on
- **Hardcoded tokens**



Fishing for Magikarp: Automatically detecting under-trained tokens in large language models

Sander Land
Cohere
sander@cohere.com

Max Bartolo
Cohere
max@cohere.com

Abstract

The disconnect between tokenizer creation and model training in language models has been known to allow for certain inputs, such as the infamous `_SolidGoldMagikarp` token, to induce unwanted behaviour. Although such 'glitch tokens' that are present in the tokenizer vocabulary, but are nearly or fully absent in training, have been observed across a variety of different models, a consistent way of identifying them has been missing. We present a comprehensive analysis of Large Language Model (LLM) tokenizers, specifically targeting this issue of detecting untrained and under-trained tokens. Through a combination of tokenizer analysis, model weight-based indicators, and prompting techniques, we develop effective methods for automatically detecting these problematic tokens. Our findings demonstrate the prevalence of such tokens across various models and provide insights into improving the efficiency and safety of language models.

https://github.com/cohere-ai/magikarp

https://arxiv.org/pdf/2405.05417

Other model design choices...

Training choices

- Position Embeddings
  - Rope / AliBi
- Context length
  - Tradeoffs: Speed vs Usability
- Hidden size
- KQV ratios
- Attention:
  - Multi-Query Attention ?
  - Windowed Attention ?
- Batch size
- Optimizers
- Schedulers

Training Nice to Haves

- Flash Attention
- Training parallelization frameworks (Deepspeed):
  - Model Parallel
  - Pipeline Parallel
  - Data Parallel
  - Sequence Parallel
  - Offloading / Sharding
  - Quantization ....
- Logging / tracking
- Checkpointing

So many options …

Outline of the talk

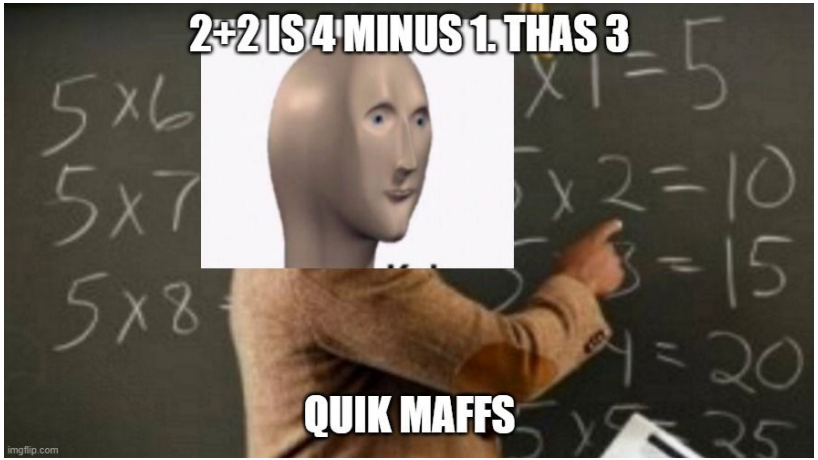
- Why train a LLM ? ✅
- Architecture ✅
- **Data**
- Evaluation
- Downstream Use
- Case study: CroissantLLM

How do I get enough data ?

To train a “good” LLM, remember the scaling laws !

Quick Maffs:

70 B model (Llama) → 70B x 20 = 1400 B tokens **minimum** !



For reference:

- All of Wikipedia English: 24B tokens ...
- All of Arxiv English: 28B tokens

We need a lot of data → Internet

How do I get enough GOOD data ?

Internet data is plentiful but very noisy !

Collection:

1. Build internet scrappers
    a. Explore the web
    b. Try to not go on porn / scam / crypto websites
    c. Try to download text content (vs. headers, HTML, databases, etc...)

2. Language Identification
    a. N-gram techniques
    b. Fast Text

3. Deduplication
    a. URL based deduplication
    b. Exact match
    c. Fuzzy matching + Clustering (MinHash LSH)

How do I get enough GOOD data ?

Filtering:

1. Heuristic-based Filtering:
    a. Rule sets (Madlad, Bloom, etc...)

2. Toxicity Filtering:
    a. Violence
    b. Racism
    c. Political Bias
    d. Misinformation
    e. Porn

3. Perplexity Filtering
    a. Fit a language model on high quality data
    b. Use the small LM to assess the probability of text from a webpage:
        i. Too high → Uninteresting data without info (ex: "----------")
        ii. Too low -> Noise (ex: "mdslif poezrulietzj hk à^p"étjlij")

Pipeline overview

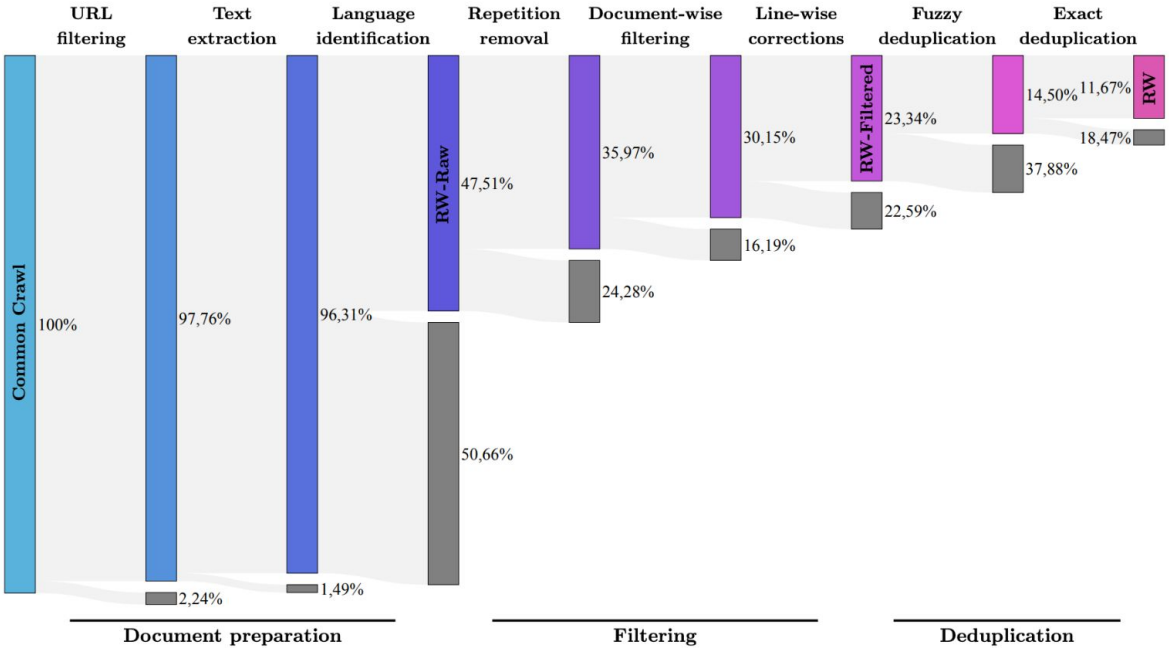


Figure 2. **Subsequent stages of Macrodata Refinement remove nearly 90% of the documents originally in CommonCrawl.** Notably, filtering and deduplication each result in a halving of the data available: around 50% of documents are discarded for not being English, 24% of remaining for being of insufficient quality, and 12% for being duplicates. We report removal rate (grey) with respect to each previous stage, and kept rate (shade) overall. Rates measured in % of documents in the document preparation phase, then in tokens.

How do I get enough GOOD data ?

Add High Quality Sources:

1. Books
2. Encyclopedias
3. Newspapers
4. Code
5. Forums (StackExchange)
6. Textbooks
7. Scientific Papers
8. **Translations**
    a. Filter for quality:
    Bifixer, Bicleaner, COMETKiwi

How do I get enough GOOD data ?

Still not enough data ?

For most languages, all of that is still not enough...

In Finnish, after all that, they have 34B tokens !

Repeating is ~OK

https://arxiv.org/abs/2305.16264

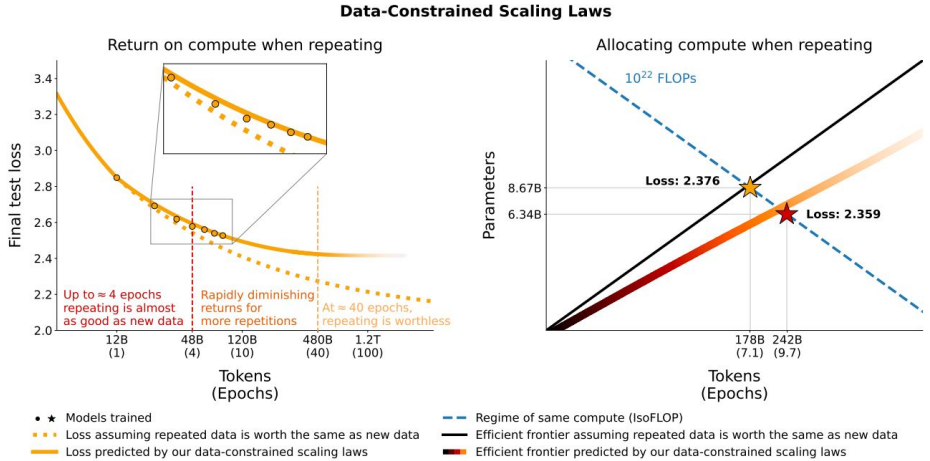


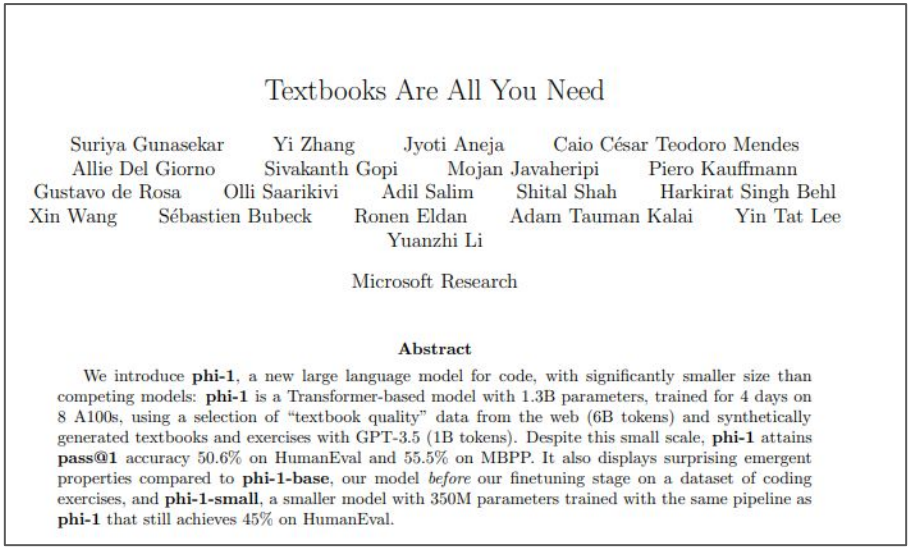
Figure 1: ***Return* and *Allocation* when repeating data.** *(Left):* Loss of LLMs (4.2B parameters) scaled on repeated data decays predictably (§6). *(Right):* To maximize performance when repeating, our data-constrained scaling laws and empirical data suggest training smaller models for more epochs in contrast to what assuming Chinchilla scaling laws [42] hold for repeated data would predict (§5).

With Very Good Data**, you don't need as much...**

Using very high quality data synthetized by GPT-4, it is possible to train a good language model on way less tokens !

Data Quality is Key !

https://arxiv.org/abs/2306.11644



Textbooks Are All You Need

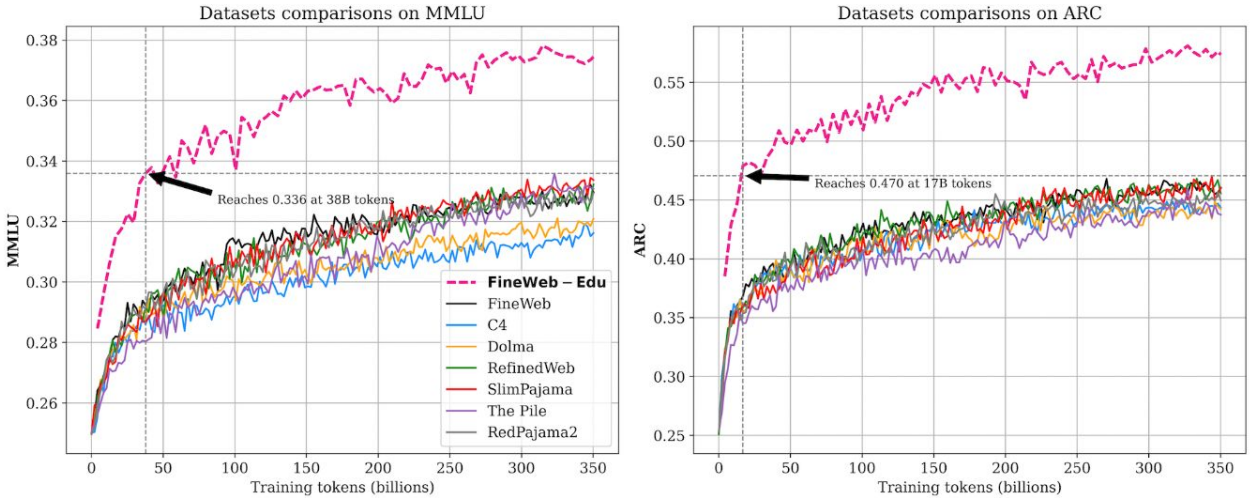
Suriya Gunasekar Yi Zhang Jyoti Aneja Caio César Teodoro Mendes
Allie Del Giorno Sivakanth Gopi Mojan Javaheripi Piero Kauffmann
Gustavo de Rosa Olli Saarikivi Adil Salim Shital Shah Harkirat Singh Behl
Xin Wang Sébastien Bubeck Ronen Eldan Adam Tauman Kalai Yin Tat Lee
Yuanzhi Li

Microsoft Research

Abstract

We introduce **phi-1**, a new large language model for code, with significantly smaller size than competing models: **phi-1** is a Transformer-based model with 1.3B parameters, trained for 4 days on 8 A100s, using a selection of "textbook quality" data from the web (6B tokens) and synthetically generated textbooks and exercises with GPT-3.5 (1B tokens). Despite this small scale, **phi-1** attains **pass@1** accuracy 50.6% on HumanEval and 55.5% on MBPP. It also displays surprising emergent properties compared to **phi-1-base**, our model *before* our finetuning stage on a dataset of coding exercises, and **phi-1-small**, a smaller model with 350M parameters trained with the same pipeline as **phi-1** that still achieves 45% on HumanEval.

With Very Good Data**... or you get boosts**



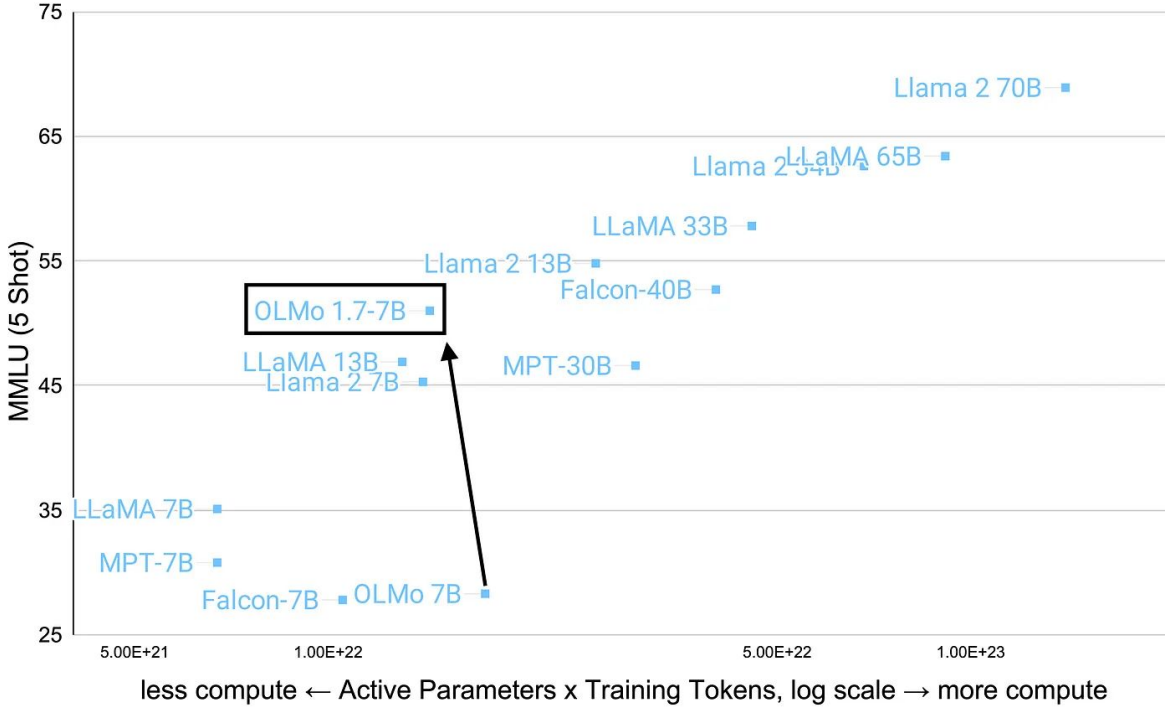
We realize the importance of high quality data (math, reasoning, code) even in the pretraining !
We can use LLMs to **filter**, **annotate** data to extract quality content or train smaller classifiers.
We can also generate **reasoning traces**, **comment code**, translate data...

[2406.17557] The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale
[2407.21783] The Llama 3 Herd of Models

With Specific Data**, you get good benchmarks**

Some data sources can **really** help models with factual knowledge, reasoning and benchmark performance !

https://blog.allenai.org/olmo-1-7-7b-a-24-point-improvement-on-mmlu-92b43f7d269d



3.1.2 Determining the Data Mix

To obtain a high-quality language model, it is essential to carefully determine the proportion of different data sources in the pre-training data mix. Our main tools in determining this data mix are knowledge classification and scaling law experiments.

Knowledge classification. We develop a classifier to categorize the types of information contained in our web data to more effectively determine a data mix. We use this classifier to downsample data categories that are over-represented on the web, for example, arts and entertainment.

Scaling laws for data mix. To determine the best data mix, we perform scaling law experiments in which we train several small models on a data mix and use that to predict the performance of a large model on that mix (see Section 3.2.1). We repeat this process multiple times for different data mixes to select a new data mix candidate. Subsequently, we train a larger model on this candidate data mix and evaluate the performance of that model on several key benchmarks.

Data mix summary. Our final data mix contains roughly 50% of tokens corresponding to general knowledge, 25% of mathematical and reasoning tokens, 17% code tokens, and 8% multilingual tokens.

[2305.10429] DoReMi: Optimizing Data Mixtures Speeds Up Language Model Pretraining
[2407.21783] The Llama 3 Herd of Models

Schedulers

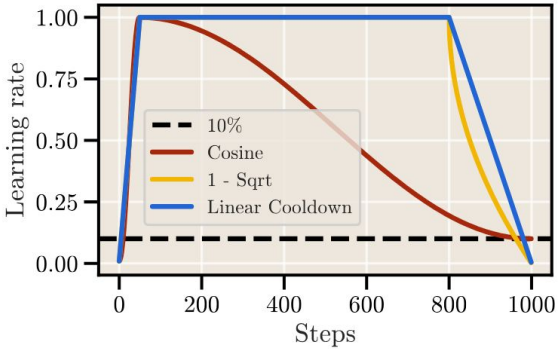
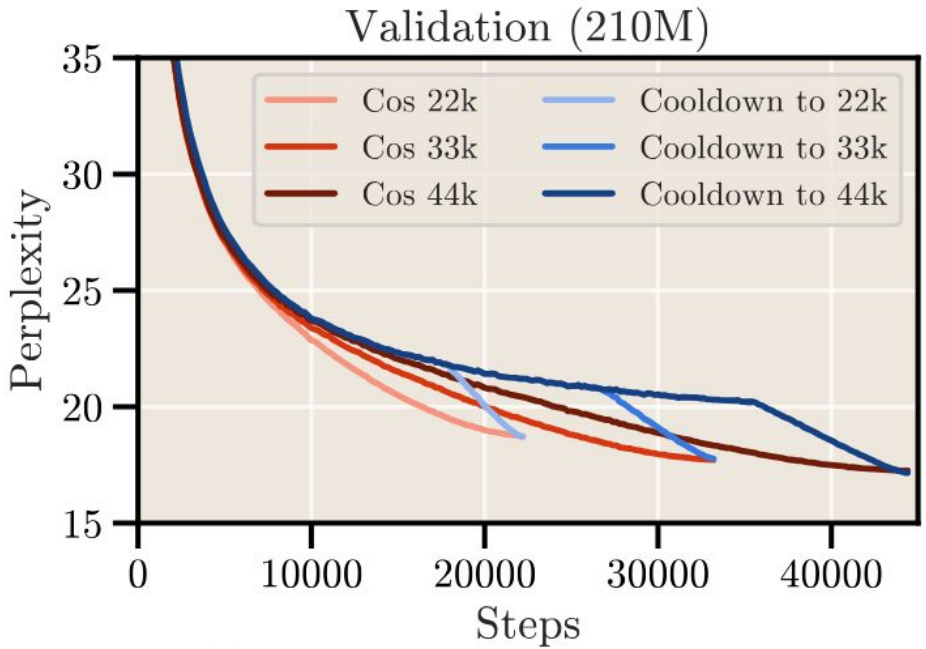


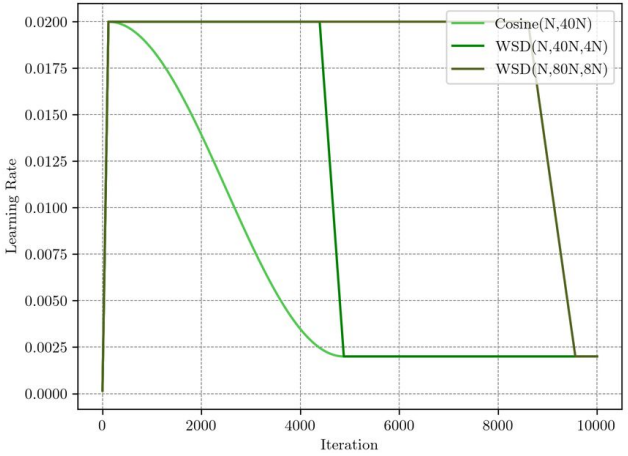
Figure 2: **Illustration of schedules.** Cosine (red) follows a slow decrease in learning rate, typically to 10% of the maximum for LLMs. The alternative is characterized by an aggressive decrease in learning rate, e.g., via a linear (blue) or square root (yellow) cooldown.

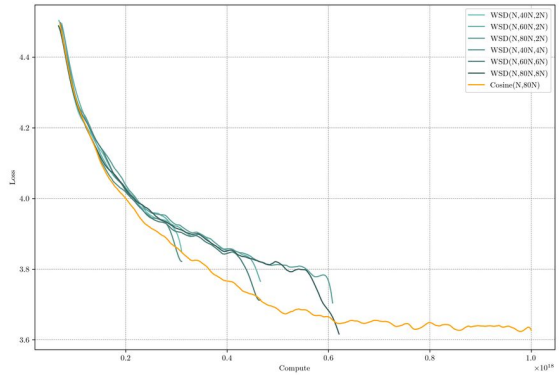


[2405.18392] Scaling Laws and Compute-Optimal Training Beyond Fixed Training Durations

[2106.04560] Scaling Vision Transformers

Annealing Sidenote (MiniCPM)





Experiment A: Annealing using only pretraining data, followed by 4B token SFT.

Experiment B: Annealing using the aforementioned high-quality data + SFT data mixed into pretraining data, also followed by 4B token SFT.

The results of the two experiments are as follows:

	CEval	CMMLU	MMLU	GSM8K	Math	HumanEval	MBPP
Experiment A	40.0	41.5	44.6	27.7	5.1	27.7	24.4
Experiment B	52.6	51.1	50.9	42.3	5.4	30.4	30.3

The results indicate that the benefits of introducing high-quality data at the beginning of annealing are much higher than adding it during the SFT phase after annealing. Therefore, we recommend that specialization and enhancement of model capabilities should start from the annealing phase.

Now everyone does this (Llama3, Phi3, Olmo, …)

https://shengdinghu.notion.site/MiniCPM-Unveiling-the-Potential-of-End-side-Large-Language-Models-d4d3a8c426424654a4e80e42a711cb20

The data

Training language models requires huge amounts of data... In French and under permissive licenses, gathering sufficiently is even more challenging !

Corpus

Culture
Permissively licensed data source

Business
Industrial and Administrative data

Knowledge
Scientific and Factual data

Translations
English and French Parallel data

Internet
Filtered web data

The data

Training language models requires huge amounts of data... In French and under permissive licenses, gathering sufficiently is even more challenging !

Corpus

Culture
Permissively licensed data source

Business
Industrial and Administrative data

Knowledge
Scientific and Factual data

Translations
English and French Parallel data

Internet
Filtered web data

Public domain books

Podcasts

Poetry

Song lyrics

Movie subtitles

The data

Training language models requires huge amounts of data... In French and under permissive licenses, gathering sufficiently is even more challenging !

Corpus

Culture
Permissively licensed data source

Business
Industrial and Administrative data

- Legal corpora
- Parliamentary debates
- Administrative Decisions
- Public business documents

Knowledge
Scientific and Factual data

Translations
English and French Parallel data

Internet
Filtered web data

The data

Training language models requires huge amounts of data... In French and under permissive licenses, gathering sufficiently is even more challenging !

Corpus

Culture
Permissively licensed data source

Business
Industrial and Administrative data

Knowledge
Scientific and Factual data

- Encyclopedia
- Textbooks
- Theses abstracts
- Scientific publications

Translations
English and French Parallel data

Internet
Filtered web data

The data

Training language models requires huge amounts of data... In French and under permissive licenses, gathering sufficiently is even more challenging !

Corpus

Culture
Permissively licensed data source

Business
Industrial and Administrative data

Knowledge
Scientific and Factual data

Translations
English and French Parallel data

Huge quantity of translation pairs sourced from different domains

Filtered with SOTA quality estimation methods

Internet
Filtered web data

The data

Training language models requires huge amounts of data... In French and under permissive licenses, gathering sufficiently is even more challenging !

Corpus

Culture
Permissively licensed data source

Business
Industrial and Administrative data

Knowledge
Scientific and Factual data

Translations
English and French Parallel data

Internet
Filtered web data

Web-scale data filtered to obtain high quality French and English texts

Github Code under open licenses

The data

Training language models requires huge amounts of data... In French and under permissive licenses, gathering sufficiently is even more challenging !

Corpus

Culture
Permissively licensed data source

Business
Industrial and Administrative data

Knowledge
Scientific and Factual data

Translations
English and French Parallel data

Internet
Filtered web data

Data Processing

Identification and scrapping → Deduplication → Filtering → Upsampling

- License checks
- Large scale download scripts
- Data quality and diversity checks

The data

Training language models requires huge amounts of data... In French and under permissive licenses, gathering sufficiently is even more challenging !

Corpus

Culture
Permissively licensed data source

Business
Industrial and Administrative data

Knowledge
Scientific and Factual data

Translations
English and French Parallel data

Internet
Filtered web data

Data Processing

Identification and scrapping → **Deduplication** → **Filtering** → **Upsampling** → 

URL deduplication
Exact deduplication
Fuzzy deduplication
(MinHash LSH)

The data

Training language models requires huge amounts of data... In French and under permissive licenses, gathering sufficiently is even more challenging !

Corpus

Culture
Permissively licensed data source

Business
Industrial and Administrative data

Knowledge
Scientific and Factual data

Translations
English and French Parallel data

Internet
Filtered web data

Data Processing

Identification and scrapping → Deduplication → Filtering → Upsampling

- Rule-based filtering
- Filtering of Toxic, Violent or Political content
- Perplexity Filtering (Data Quality)

The data

Training language models requires huge amounts of data... In French and under permissive licenses, gathering sufficiently is even more challenging !

Corpus

Culture
Permissively licensed data source

Business
Industrial and Administrative data

Knowledge
Scientific and Factual data

Translations
English and French Parallel data

Internet
Filtered web data

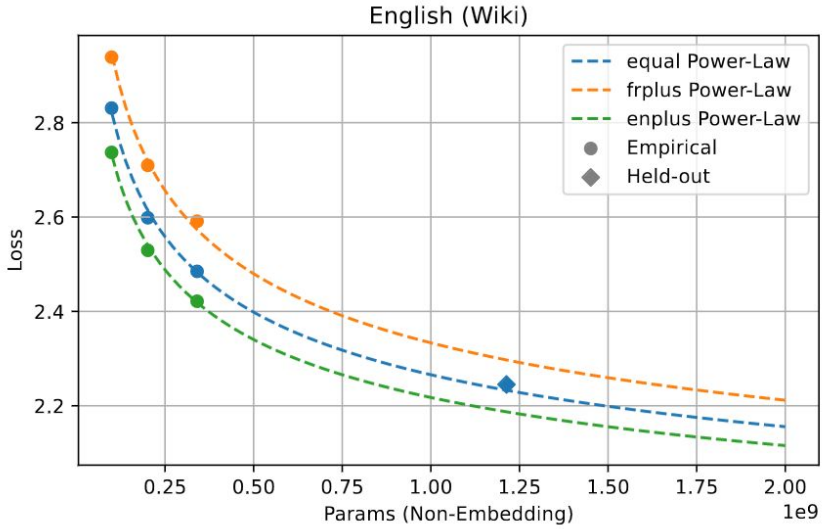
Data Processing

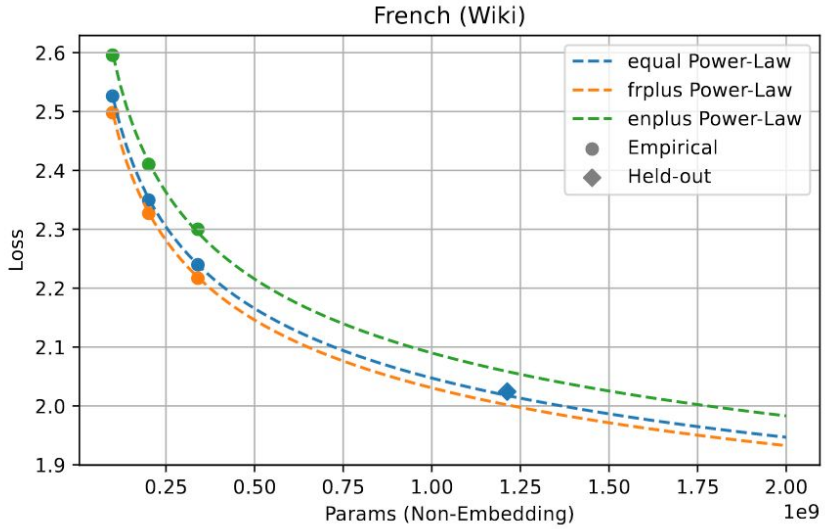
Identification and scrapping → Deduplication → Filtering → Upsampling → 

Final data mix

Upsampling 🇫🇷 to obtain a balanced corpus 🇫🇷 / 🇬🇧

Selecting an optimal language mix





Scaling laws to determine optimal language ratios

The data

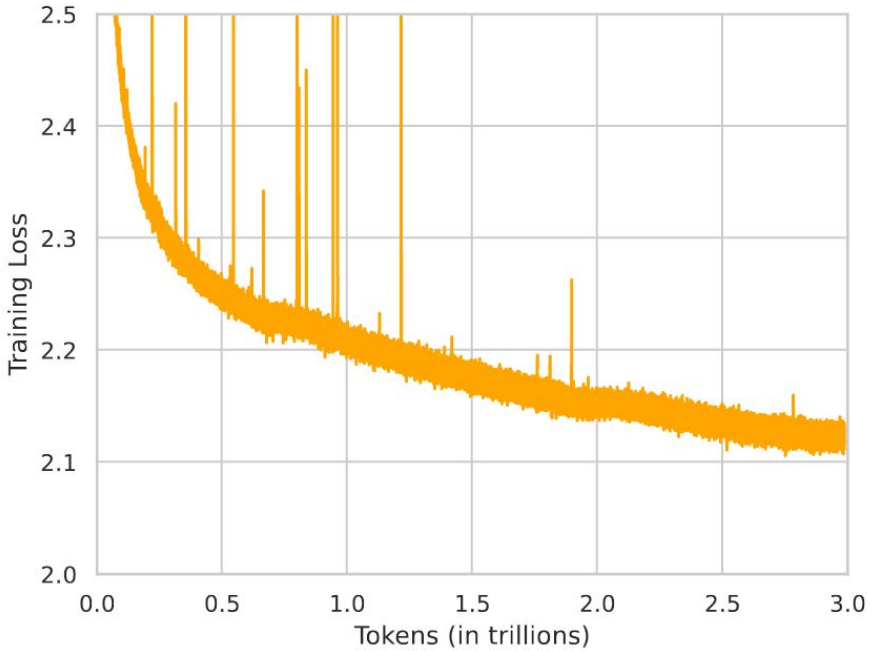
Training language models requires huge amounts of data... In French and under permissive licenses, gathering sufficiently is even more challenging !

	Size (GB)	Docs. (M)	Tokens (B)	Token/Doc	Sampling Ratio	# tokens (B)
French	1258.70	376.27	303.51	806.63	4.09	1240.08
English	2351.13	591.23	655.64	1108.94	1.89	1240.09
Code	366.87	81.90	141.43	1726.76	2.04	288.92
Parallel	113.91	408.03	35.78	87.68	6.13	219.26
Total	4090.61	1457.43	1136.35	779.70	14.15	2988.35

Table 1 – Final training datamix for CroissantLLM

Our training corpus is **3000 billion tokens**, with equal parts French and English.

Croissant Training details



Key stats

- 17 days, 30x8 GPU A100 80Gb
- Batch size == 7680:
  - 4 gradient accumulation steps
  - 8 Mini-Batch size
  - 2048 Sequence Length
  - **15M tokens per batch**
- 120 TFlop/s average
- Checkpoints every 5k steps
- Cosine LR

Outline of the talk

- Why train a LLM ? ✅
- Architecture ✅
- Data ✅
- **Evaluation**
- Downstream Use
- Case study: CroissantLLM

How do I know if the model is good ?

Non trivial problem !

- Few-shot benchmarks
  - Assessing Multiple Choice Answering capabilities:
    - Reasoning
    - Grammar / Vocab
    - Reading Comprehension
    - Commonsense reasoning
    - Factual Knowledge
    - **Not clear how to determine the best answer (logits, regex, etc)**
  - Assessing Generative capabilities:
    - Pass @k for code
    - NLG metrics for tasks like summarization, translation, etc...
  - Assessing internal model bias
    - Sexism
    - Racism
    - Cultural bias

Find all c in $\mathbb{Z}_3$ such that $\mathbb{Z}_3[x]/(x^2 + c)$ is a field.
(A) 0 **(B) 1** (C) 2 (D) 3

Figure 14: An Abstract Algebra example.

What is the embryological origin of the hyoid bone?
(A) The first pharyngeal arch
(B) The first and second pharyngeal arches
(C) The second pharyngeal arch
(D) The second and third pharyngeal arches

Figure 15: An Anatomy example.

Why isn't there a planet where the asteroid belt is located?
(A) A planet once formed here but it was broken apart by a catastrophic collision.
(B) There was not enough material in this part of the solar nebula to form a planet.
(C) There was too much rocky material to form a terrestrial planet but not enough gaseous material to form a jovian planet.
(D) Resonance with Jupiter prevented material from collecting together to form a planet.

Figure 16: An Astronomy example.

Three contrasting tactics that CSO's can engage in to meet their aims are _____ which typically involves research and communication, _____, which may involve physically attacking a company's operations or _____, often involving some form of _____.
(A) Non-violent direct action, Violent direct action, Indirect action, Boycott
(B) Indirect action, Instrumental action, Non-violent direct action, Information campaign
(C) Indirect action, Violent direct action, Non-violent direct-action Boycott.
(D) Non-violent direct action, Instrumental action, Indirect action, Information campaign

Figure 17: A Business Ethics example.

How many attempts should you make to cannulate a patient before passing the job on to a senior colleague?
(A) 4 (B) 3 **(C) 2** (D) 1

Figure 18: A Clinical Knowledge example.

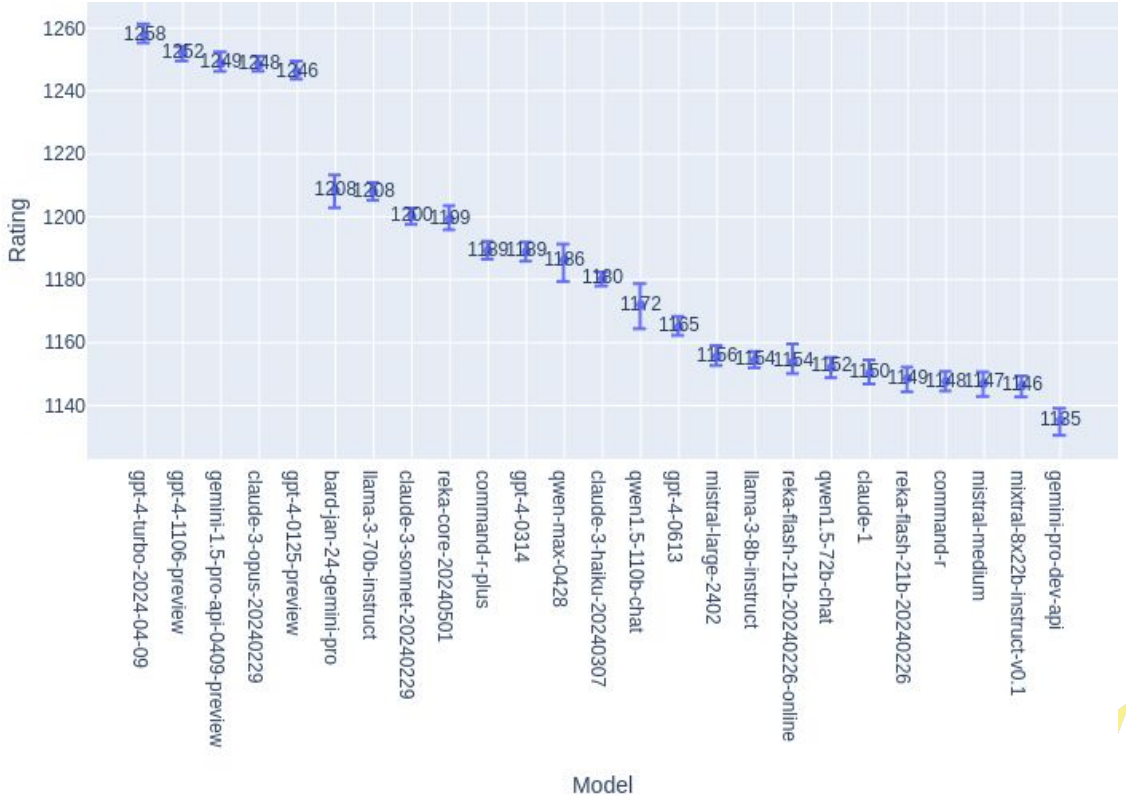
MMLU

Benchmarks can be rigged …

How do I know if the model is good ?

Non trivial problem !

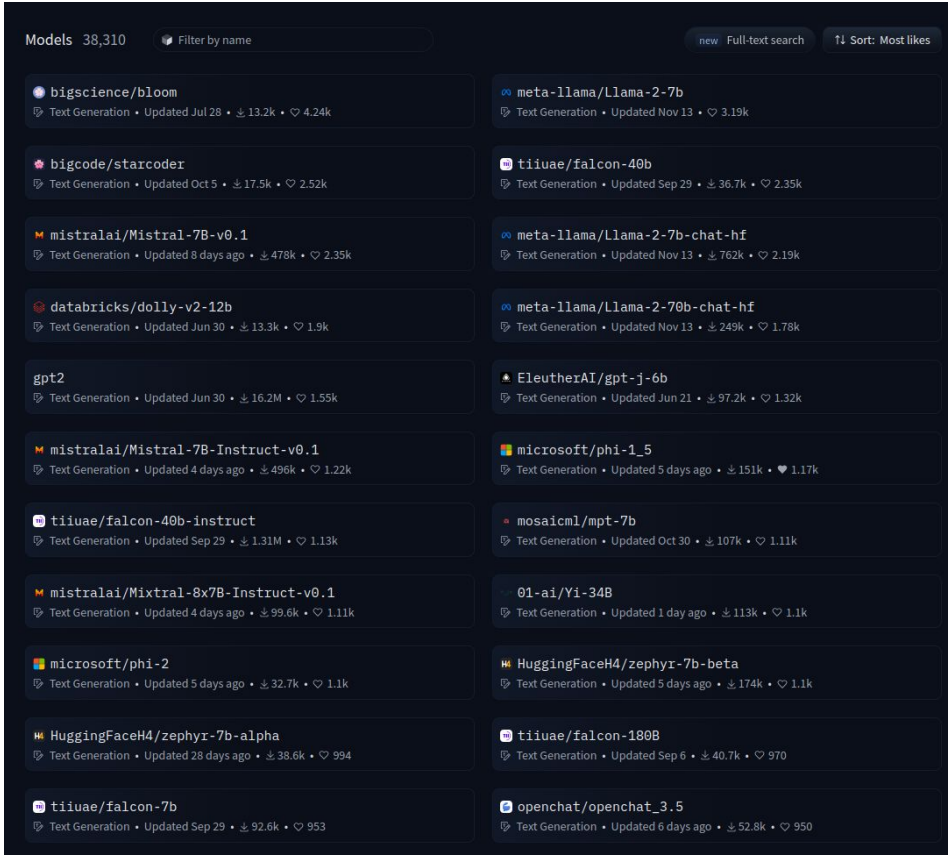
- Arena Fight: Pairwise voting system
  - Humans as judges: Chatbot Arena
  - LLMs as judges: (ex: MT-Bench, AlpacaEval)



How do I know if the model is good ?

Non trivial problem !

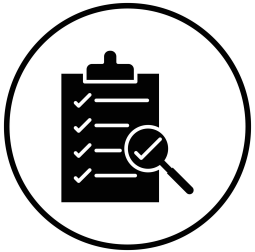
- Let the users judge !
  - Test of time
  - HF downloads



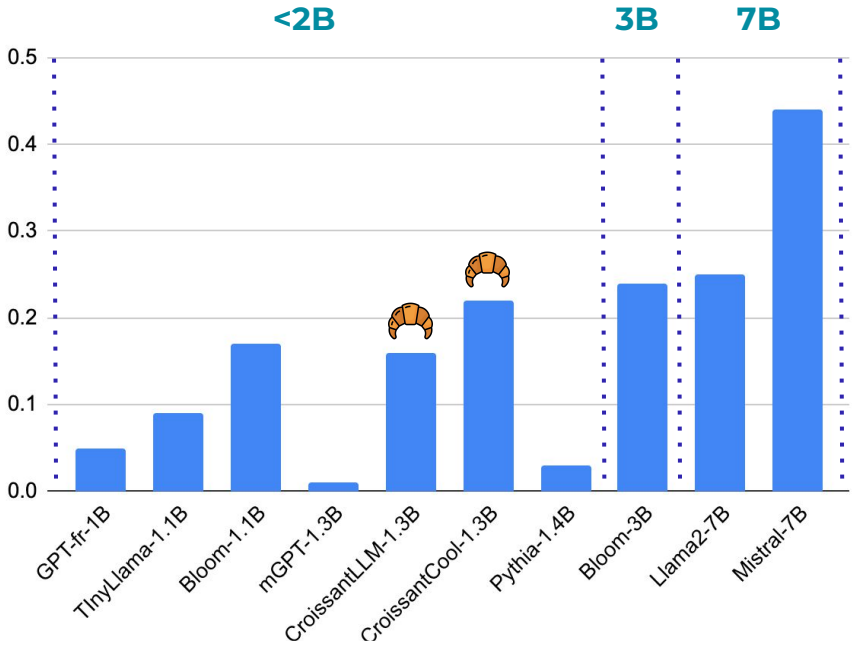
Croissant Evaluation

To evaluate the model, we rely on existing and new benchmarks:

- English**:**
  - lm-eval-harness tasks (reference leaderboard)
- French (**FrenchBench**)
- Extra:
  - Perplexities
  - Translation
  - Trivia
  - MTBench (Fr + Eng)
  - Dialog Summarization

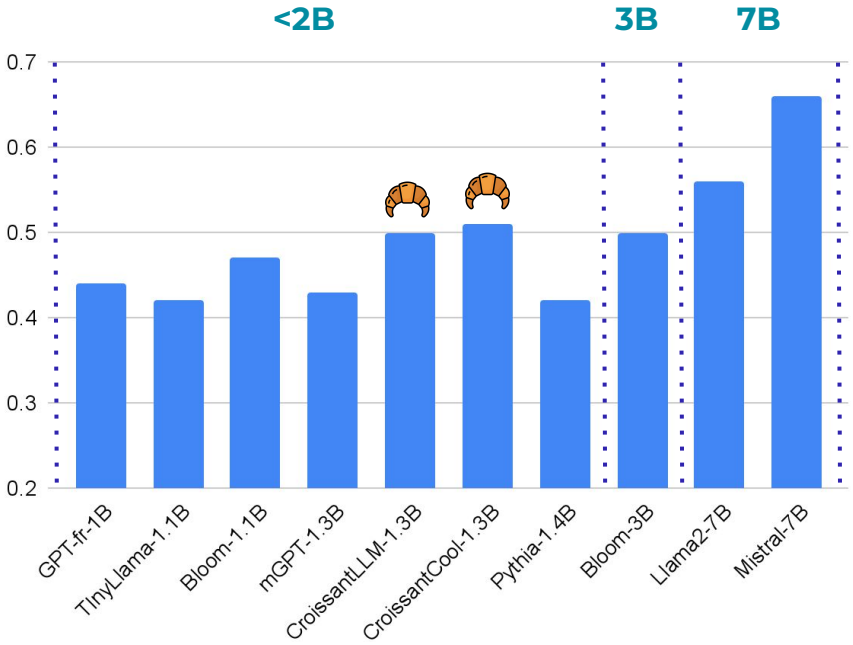


Evaluation : FrenchBench



FrenchBench Generative

With open-ended generative tasks, we evaluate the extractive and synthetic capabilities of models, the extent of their factual knowledge...



FrenchBench Multiple Choice

With Multiple Choice, we evaluate reasoning, factual knowledge, bias, french vocabulary and grammar...

Croissant Evaluation

Task	Hellaswag(fr)	Arc-e(fr)	fr-vocab	fr-grammar	Belebele(fr)	Avg
OPT(1.3B)	0.28	0.19	0.50	0.61	0.28	0.37
Pythia(1.4B)	0.30	0.20	0.61	0.76	0.23	0.42
TinyLlama(1.1B)	0.33	0.23	0.64	0.67	0.25	0.42
mGPT(1.3B)	0.27	0.20	0.71	0.73	0.23	0.43
GPT-fr(1B)	0.30	0.19	0.70	0.79	0.24	0.44
Bloom(1.1B)	0.34	0.22	0.76	0.79	0.24	0.47
CroissantLLM	0.40	0.26	0.75	0.80	0.27	0.50
Bloom(3B)	0.40	0.27	0.78	0.81	0.23	0.50
CroissantCool	0.42	0.28	0.79	0.79	0.28	0.51
Llama2(7B)	0.44	0.38	0.76	0.77	0.43	0.56
Mistral(7B)	0.49	0.47	0.78	0.78	0.78	0.66

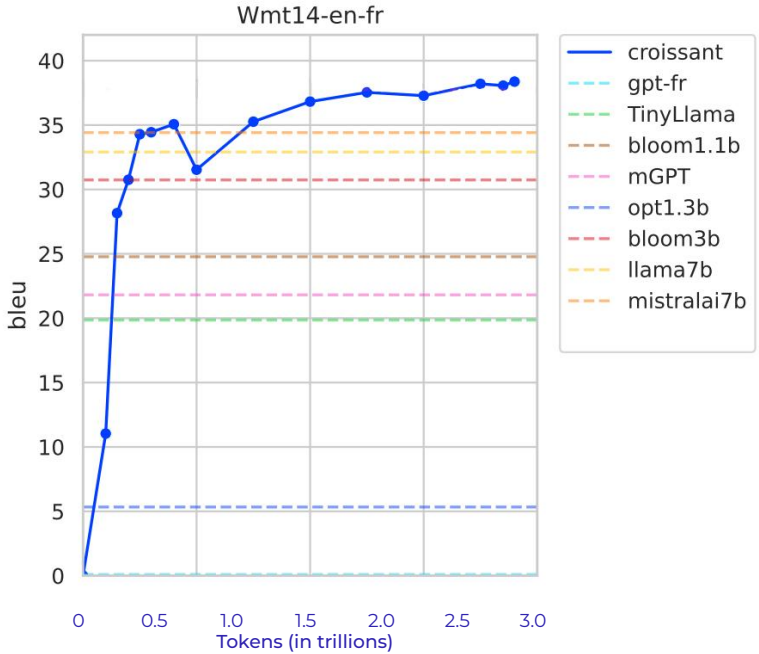
Croissant Evaluation

Task	FGenQ	FGenAns	MultiFQuAD	OSum(A)	FTrivia	Avg
Pagnol-XL(1.5B)	0.06	0.04	0.03	0.03	-	*0.04
GPT-fr(1B)	0.04	0.02	0.05	0.11	-	*0.06
mGPT(1.3B)	0.01	0.00	0.02	0.03	0.33	0.08
OPT(1.3B)	0.09	0.18	0.21	0.17	0.39	0.21
Bloom(1.1B)	0.17	0.28	0.26	0.10	0.31	0.23
Pythia(1.4B)	0.15	0.34	0.27	0.21	0.44	0.28
CroissantLLM	0.19	0.40	0.33	0.10	0.52	0.31
Bloom(3B)	0.21	0.47	0.37	0.18	0.47	0.34
TinyLlama(1.1B)	0.18	0.46	0.41	0.23	0.45	0.35
CroissantCool	0.20	0.45	0.36	0.27	0.53	0.36
Llama2(7B)	0.25	0.68	0.60	0.30	0.70	0.50
Mistral(7B)	0.33	0.78	0.64	0.31	0.74	0.56

Croissant Evaluation

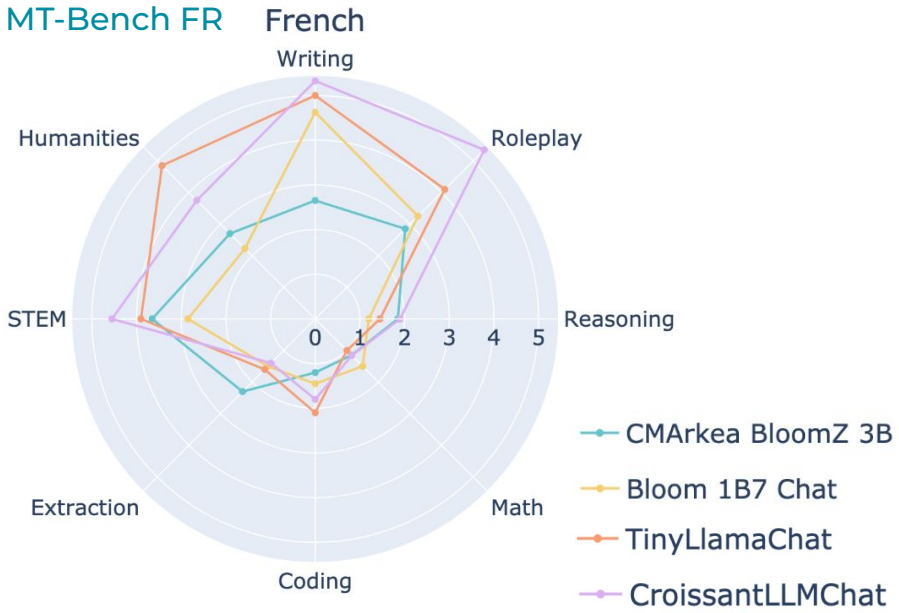
Task	Arc-e	Belebele (eng)	Hellaswag	PiQA	SciQ	Avg
GPT-fr(1B)	0.26	0.28	0.29	0.53	0.68	0.40
mGPT(1.3B)	0.48	0.23	0.35	0.66	0.62	0.47
Bloom(1.1B)	0.55	0.24	0.38	0.68	0.89	0.55
OPT(1.3B)	0.61	0.23	0.42	0.72	0.92	0.58
Bloom(3B)	0.63	0.24	0.41	0.72	0.93	0.59
Pythia(1.4b)	0.63	0.26	0.42	0.71	0.92	0.59
CroissantLLM	0.62	0.28	0.42	0.72	0.92	0.59
CroissantCool	0.62	0.26	0.43	0.73	0.91	0.59
TinyLlama(1.1B)	0.65	0.26	0.45	0.73	0.94	0.60
Llama2(7B)	0.78	0.46	0.52	0.78	0.97	0.70
Mistral(7B)	0.83	0.85	0.55	0.81	0.98	0.80

Evaluation : Translation & MT-Bench FR



Translation

CroissantLLM rivals open large language models 10x the size on translation tasks, as well as the best specialized models.



Assistant tasks

We evaluate model performance as an assistant for various tasks (creativity, code, reformulation), as assessed by GPT-4.

Croissant Evaluation

	WMT 14				TICO		FLORES			
	en→fr		fr→en		en→fr		en→fr		fr→en	
	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU
NMT models										
NLLB 1.3B *0-shot*	86.82	41.59	84.55	36.47	81.15	40.22	87.10	47.49	87.21	40.47
Pre-trained models										
LLaMA-2 7B *5-shot*	84.37	32.98	86.66	38.57	78.05	33.75	85.03	38.59	88.75	41.83
LLaMA-2 13B *5-shot*	85.94	36.76	87.02	39.93	80.04	38.21	86.67	43.49	89.03	42.71
Mistral-7B-v0.1 *5-shot*	84.99	34.82	87.01	39.55	79.34	37.82	86.07	41.31	88.36	42.56
TinyLLaMA *5-shot*	73.03	18.13	82.99	29.85	69.20	20.55	74.40	21.17	85.86	33.10
CroissantLLM *5-shot*	85.11	38.09	85.70	36.30	78.74	38.49	86.85	46.58	88.58	42.83
SFT models										
TowerInstruct-7B-v0.1 *0-shot*	88.07	46.19	88.14	46.75	81.53	41.27	88.38	48.57	89.56	46.34
TinyLLaMA SFT *0-shot*	–	–	–	–	73.04	23.61	78.08	27.24	86.26	32.80
CroissantLLMChat *0-shot*	–	–	–	–	80.27	36.99	86.82	44.79	88.38	41.54
CroissantLLMChat *0-shot (Beam Search)*	–	–	–	–	80.72	38.34	87.68	47.11	88.71	42.90

Outline of the talk

- Why train a LLM ? ✅
- Architecture ✅
- Data ✅
- Evaluation ✅
- **Downstream Use**
- Case study: CroissantLLM

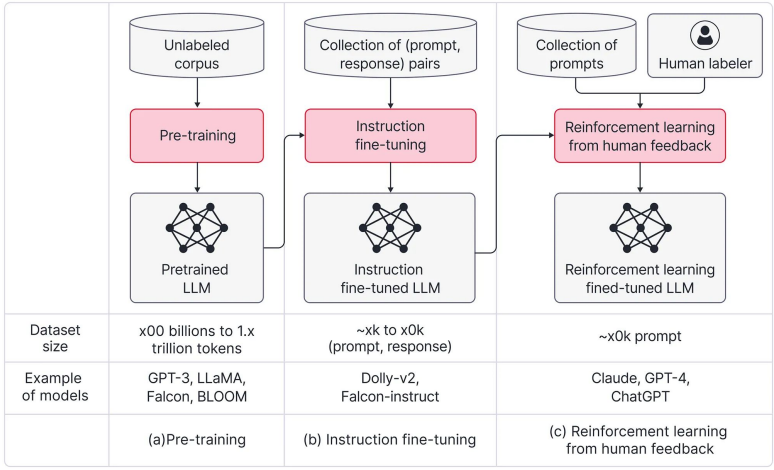
ILLUIN TECHNOLOGY

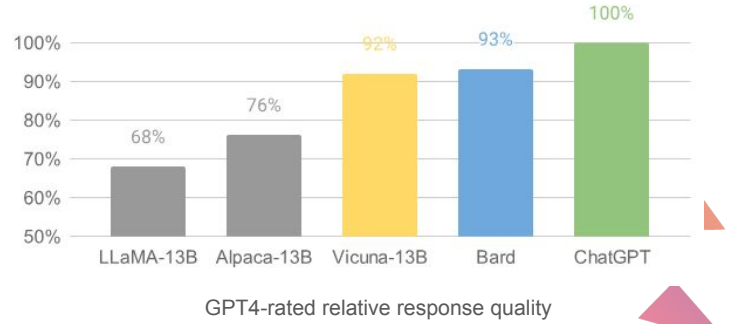
Base Models < Tuned Models

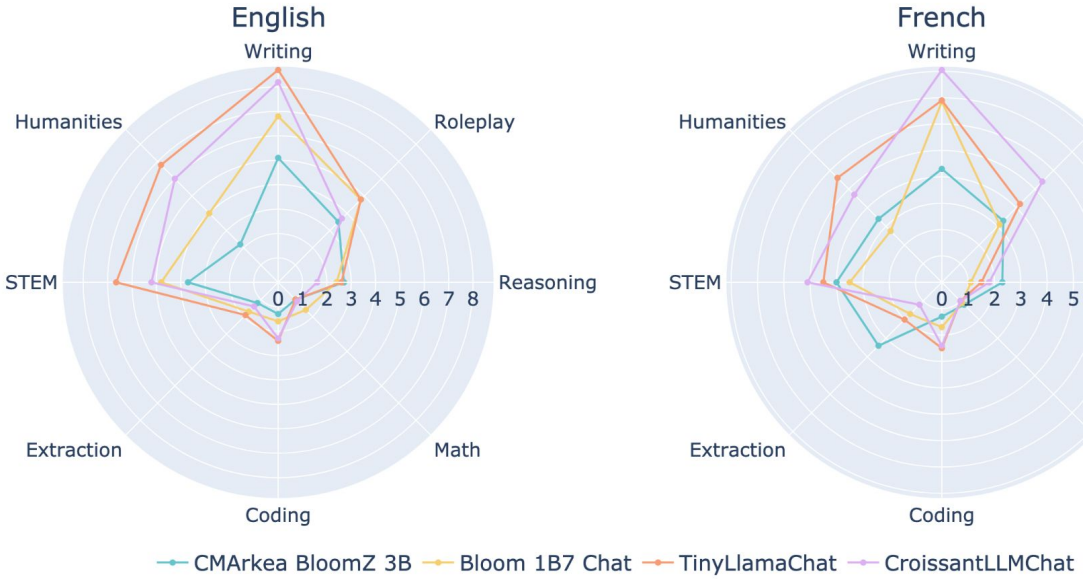
Base models are diamonds in the rough…

Instruction Tuning helps extract performance and usability !

- Instruction Tuning
  - Real: T5, T0, Flan
  - Synthetic: Alpaca
- Chat Tuning
  - Real: Dialogue corpus
  - Semi-real: ShareGPT
  - Synthetic: UltraChat
- RL:
  - PPO
  - DPO
  - IPO







MT-Bench Turn 1

Models	Accuracy
Bloom 1b7 Chat	24.47 %
TinyLlamaChat	44.47 %
CroissantLLMChat	**47.11 %**

French Trivia Accuracy

We finetune CroissantLLM on 300k Chat samples in English and French

Outline of the talk

- Why train a LLM ? ✅
- Architecture ✅
- Data ✅
- Evaluation ✅
- Downstream Use ✅
- **Applications: CroissantLLM**

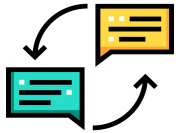
CroissantLLM: Potential applications

Specific tasks

- Writing assistance
- Summary
- Orthographic correction
- Prompt Compression (RAG)
- etc.

Phone & CPU

Unmatched performance in 🇫🇷 amongst models lightweight enough to run on phones and local hardware.



Translation

CroissantLLM is the best model of its size in translation 🇫🇷 / 🇬🇧 matching the performance of Mistral and Llama models of 10 x the size.

Frugality

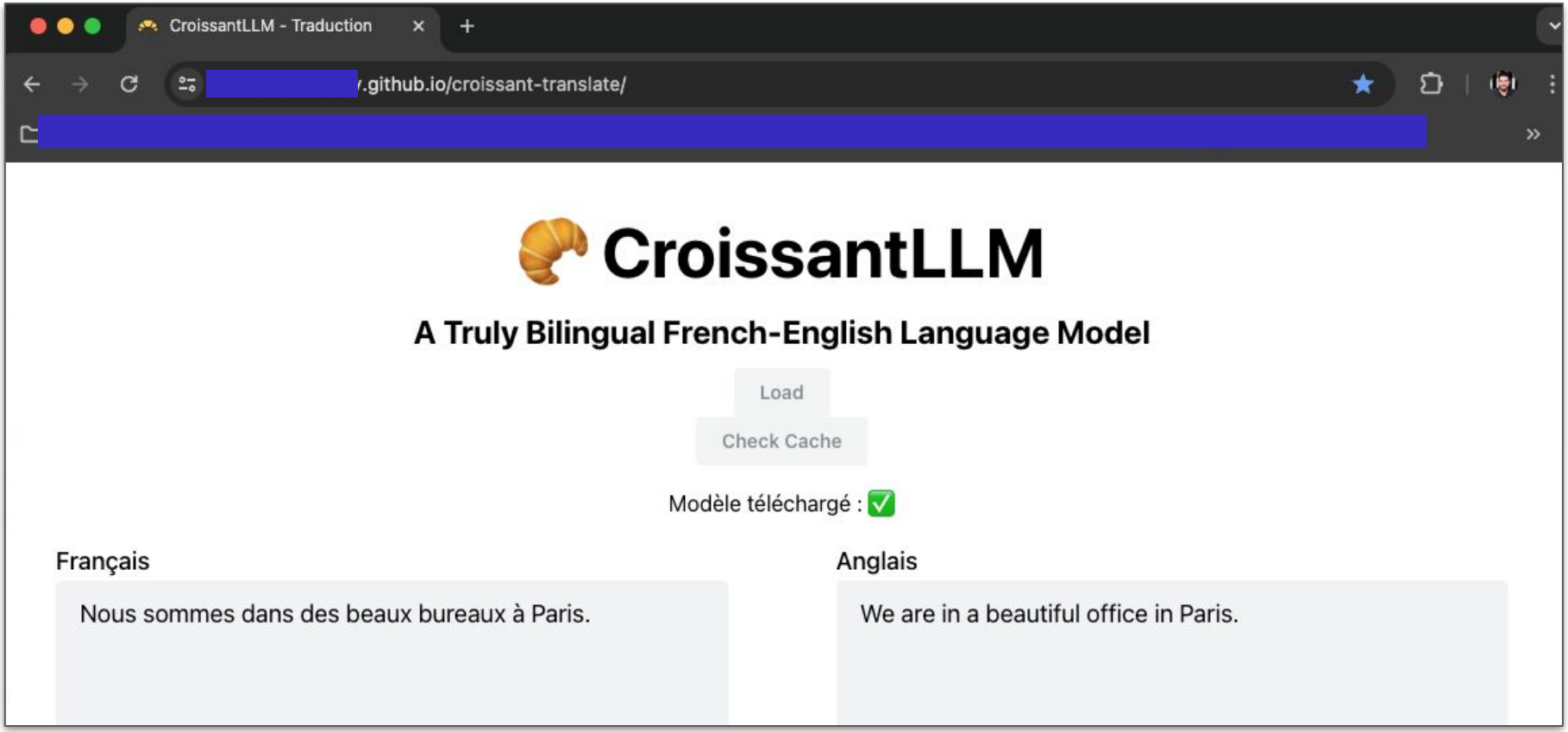
The lightweight model reduces costs and energy requirements.

Generation speed

More than **30 tokens per second on CPU**

More than 120 **tokens per second on lower end GPU (T4)**

Use case: In-Browser Local Translation



CroissantLLM: Retrieval model

Overall | Bitext Mining | Classification | Clustering | Pair Classification | Reranking | Retriev

English | Chinese | French | Law | Polish

Retrieval French Leaderboard 🔎🇫🇷

- **Metric:** Normalized Discounted Cumulative Gain @ k (ndcg_at_10)
- **Languages:** French
- **Credits:** Lyon-NLP: Gabriel Sequeira, Imene Kerboua, Wissam Siblini, Mathieu Ciancone, Marion Schaeffer

Rank ▲	Model ▲	Model Size (Million Parameters) ▲	Memory Usage (GB, fp32) ▲	Average ▲
1	text-embedding-ada-002			48.23
2	mistral-embed			46.81
3	Solon-embeddings-large-0.1	560	2.09	46.78
4	voyage-code-2			45.33
5	sentence_croissant_alpha_v0.2	1280	4.77	43.85
6	sentence-t5-xxl	4865	18.12	43.84
7	bge-ft	568	2.12	43.16
8	multilingual-e5-large	560	2.09	42.17
9	multilingual-e5-base	278	1.04	41.19
10	sentence_croissant_alpha_v0.1	1280	4.77	41.15
11	voyage-2			40.81

CroissantLLM: Potential applications

- Resource-constrained applications:
  - Runs easily on a lower-end mobile device:
    - Reformulation, Writing assistant
    - Translation
    - On-edge multimodal model for accessibility apps...
  - Runs on CPU servers or local hardware !
- Standard NLP tasks with the aims of diminishing costs/latency (awesome environment)
  - Textual Embeddings
  - Classification tasks
  - Summarization
- Prompt Compression (LLMLingua)
  - RAG tasks with larger model
- Large scale data cleaning:
  - OCR correction (Gallica dataset ?)
- Constrained Generation:
  - Outlines ?

CroisssantLLM is primarily a research project but lots of promising applications exist !

Outline of the talk

- Why train a LLM ? ✅
- Architecture ✅
- Data ✅
- Evaluation ✅
- Downstream Use ✅
- Applications: CroissantLLM ✅

Takeaways

- Lots of decisions to take in the training process ! **Tradeoffs are everywhere**
- Limiting factors are compute, but also data and human engineering time
- Evaluation is both key and very hard to get right

Transparency

Transparency & Open-Source

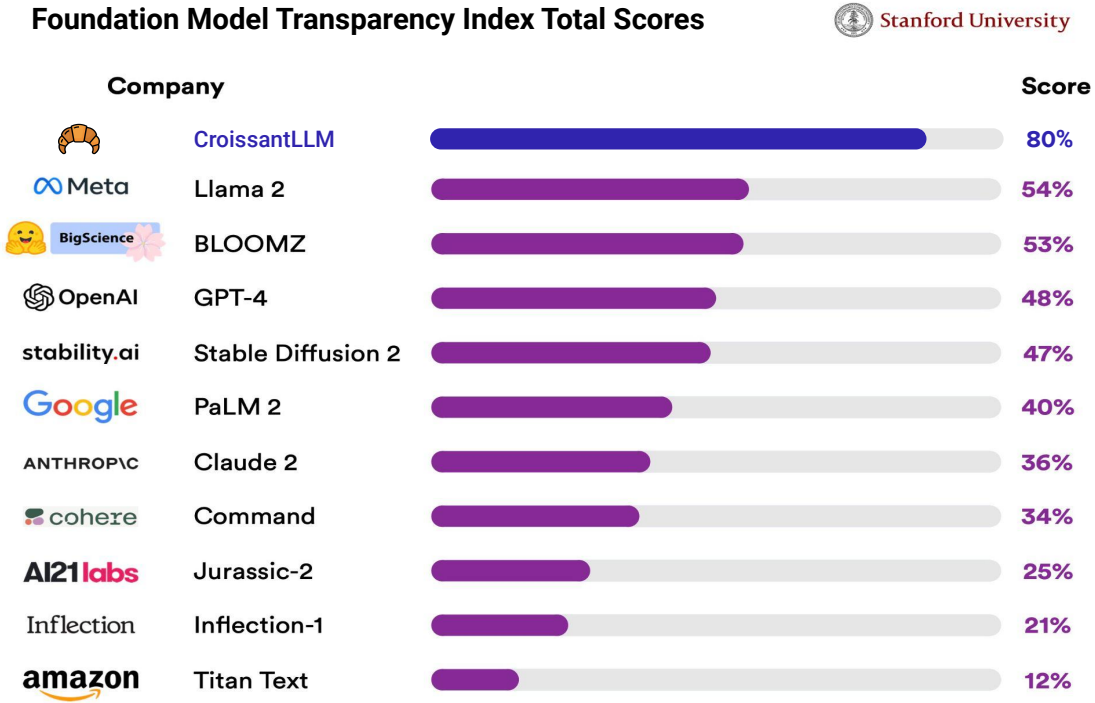
Project rooted in transparency, to serve as a useful resource for industrial practitioners and researchers !

Documented training process from beginning to end

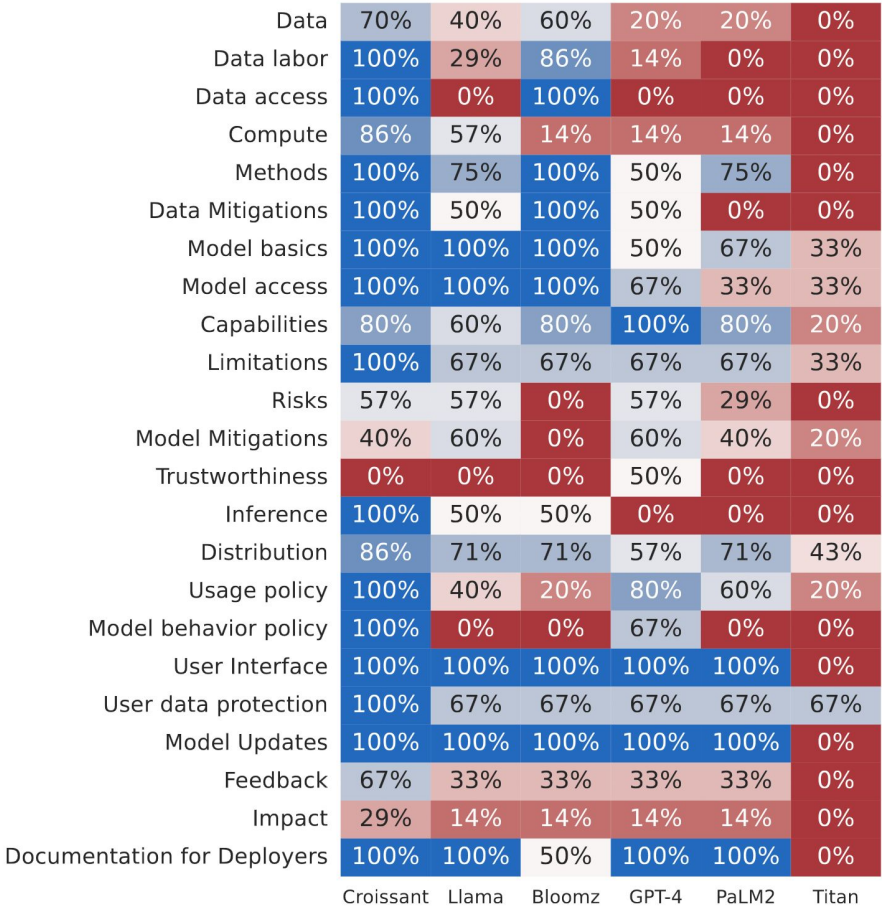
Openly available:

- Training corpus
- Model checkpoints
- Evaluation Benchmarks
- Code bases

No usage restrictions (MIT)



Stanford Foundation Model Transparency Index



Canaries in the data !

Canaries in the data !

Copyright Traps for Large Language Models

Matthieu Meeus* *Imperial College London*
Igor Shilov* *Imperial College London*
Manuel Faysse *Paris-Saclay University* *Illuin Technology*
Yves-Alexandre de Montjoye† *Imperial College London*

Abstract

Questions of fair use of copyright-protected content to train Large Language Models (LLMs) are being very actively debated. Document-level inference has been proposed as a new task: inferring from black-box access to the trained model whether a piece of content has been seen during training. SOTA methods however rely on naturally occurring memorization of (part of) the content. While very effective against models that memorize a lot, we hypothesize–and later confirm–that they will not work against models that do not naturally memorize, e.g. medium-size 1B models. We here propose to use copyright traps, the inclusion of fictitious entries in original content, to detect the use of copyrighted materials in LLMs with a focus on models where memorization does not naturally occur. We carefully design an experimental setup, randomly inserting traps into original content (books) and train a 1.3B LLM. We first validate that the use of content in our target model would be undetectable using existing methods. We then show, contrary to intuition, that even medium-length trap sentences repeated a significant number of times (100) are not detectable using existing methods. However, we show that longer sequences repeated a large number of times can be reliably detected (AUC=0.75) and used as copyright traps. We further improve these results by studying how the number of times a sequence is seen improves detectability, how sequences with higher perplexity tend to be memorized more, and how taking context into account further improves detectability.

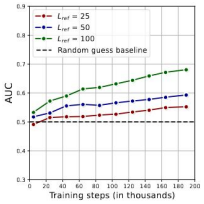


Figure 1: **Memorization throughout training.** The *Ratio* MIA performance (AUC) for synthetically generated trap sequences (of varying sequence length), repeated 1,000 times in a book, evaluated on intermediate checkpoints of the target LLM.

1 Introduction

With the growing adoption of ever-improving Large Language Models (LLMs), concerns are being raised when it comes to the use of copyright protected content for training. Numerous content creators have indeed filed lawsuits against technology com-

*Equal contribution
†Corresponding author: deMontjoye@imperial.ac.uk.

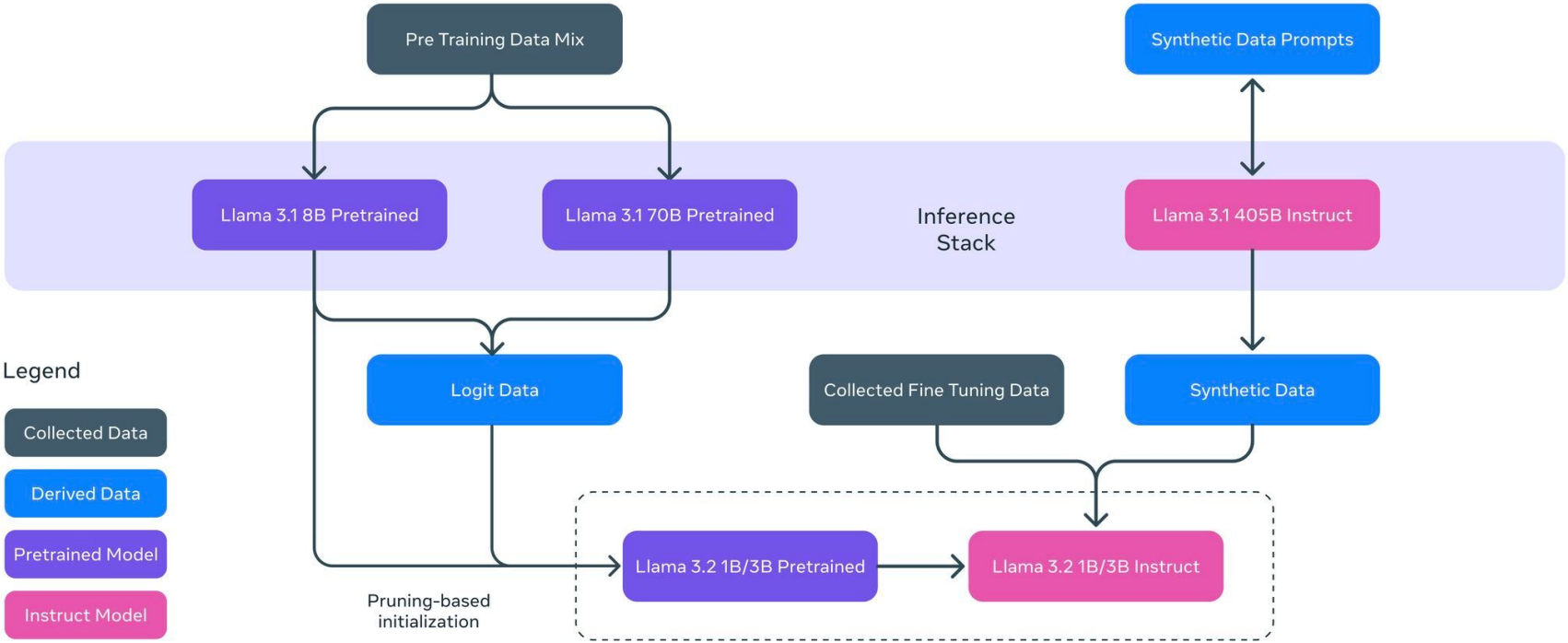
We include “Trap” sequences within the pretraining dataset to assess risks of model memorization (PII, Copyright claims)

ICML 2024 Paper

https://arxiv.org/pdf/2402.09363

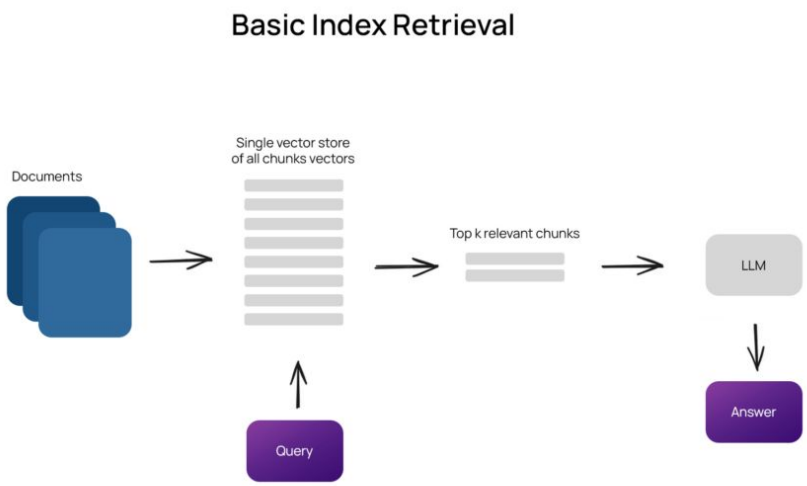
Current paradigms

Llama 3.1 - Pruning & Distillation



Retrieval Augmented Generation

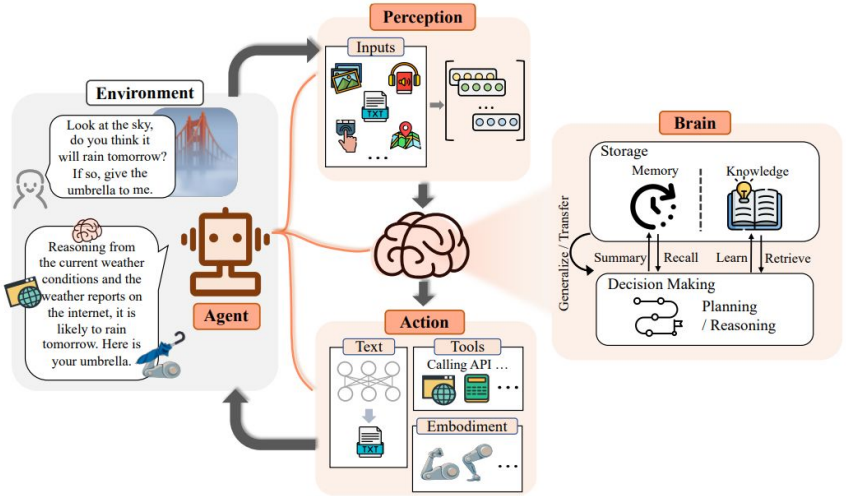
LLMs are given additional context



LLM Agents

LLM orchestrate complex pipelines

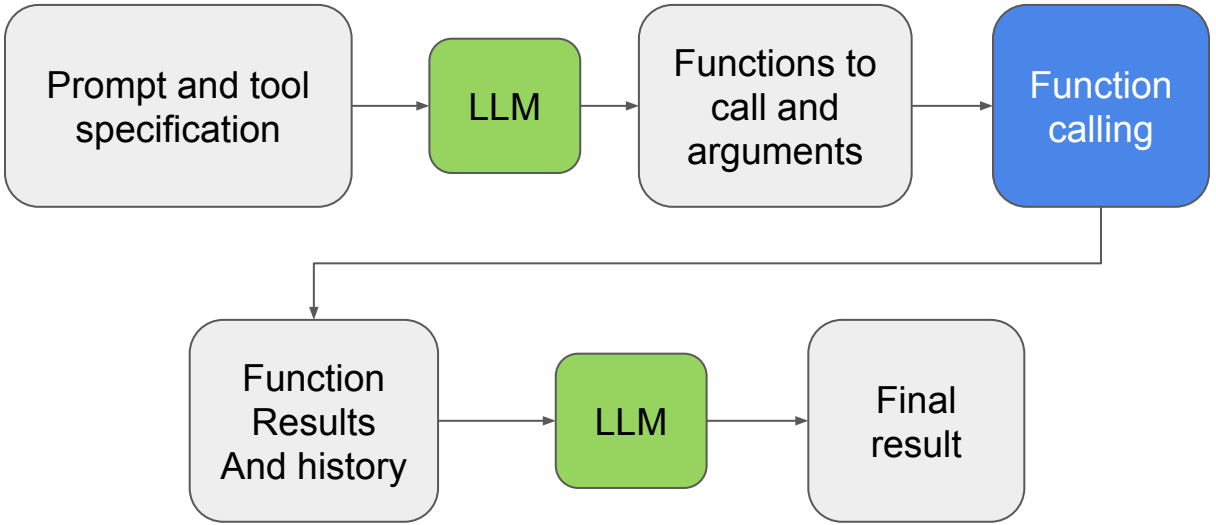
LLM based agents, hereinafter also referred to as LLM agents for short, involve LLM applications that can execute complex tasks through the use of an architecture that combines LLMs with key modules like planning and memory. When building LLM agents, an LLM serves as the main controller or "brain" that controls a flow of operations needed to complete a task or user request. The LLM agent may require key modules such as planning, memory, and tool usage.



Le LLM devient un bloc computationnel, ou un orchestrateur.

LLM Agents

Tool Use / Function Calling



LLM Agents

Tool Use / Function Calling

function calling

```python
import anthropic

client = anthropic.Anthropic()

response = client.messages.create(
    model="claude-3-5-sonnet-20240620",
    max_tokens=1024,
        tools = [
            {
                "name": "calculator",
                "description": "A simple calculator that performs basic arithmetic operations.",
                "input_schema": {
                    "type": "object",
                    "properties": {
                        "expression": {
                            "type": "string",
                            "description": "The mathematical expression to evaluate (e.g., '2 + 3 * 4')."
                        }
                    },
                    "required": ["expression"]
                }
            }
        ]
    messages=[{"role": "user", "content": "What is the result of 1,984,135 * 9,343,116 ?"}],
)
print(response)
```

LLM Agents

Tool Use / Function Calling

function calling

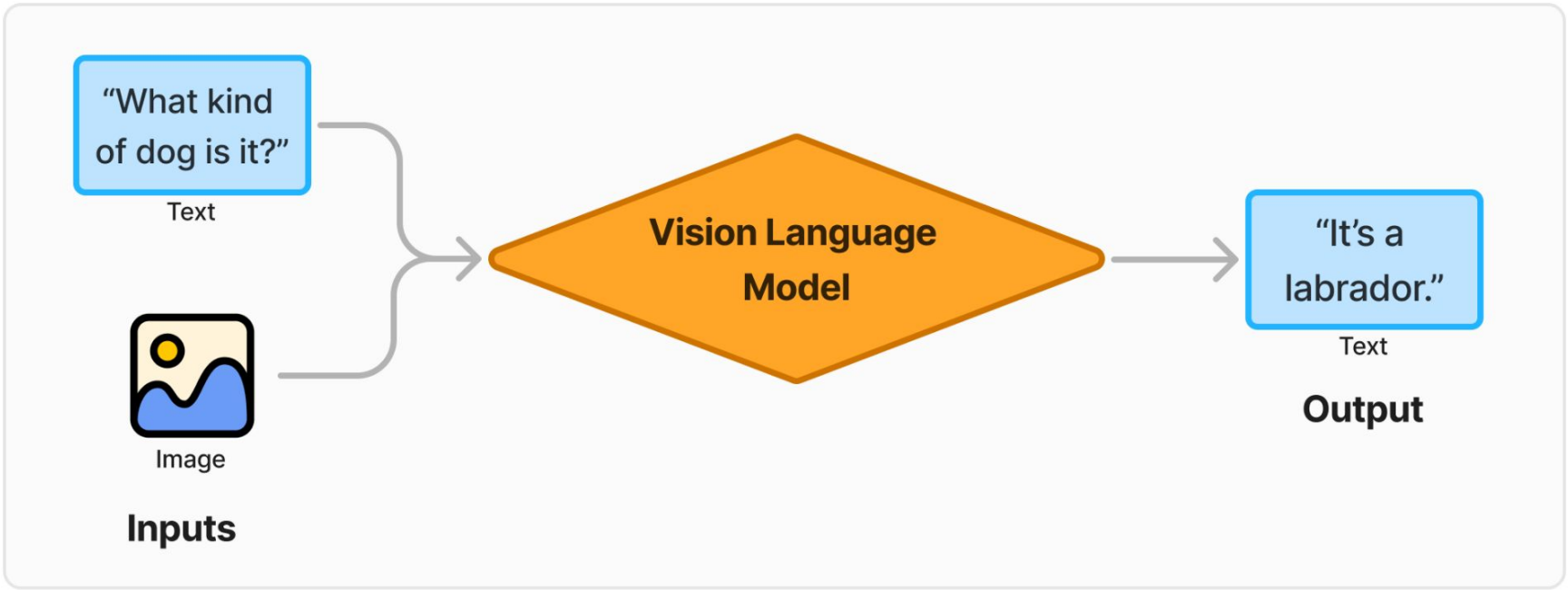
```
Initial Response:
Stop Reason: tool_use
Content:
[ContentBlock(text='<thinking>\nThe calculator function is the relevant tool
to answer this request, since it involves evaluating a mathematical expression.\n\nThe
required parameter for the calculator function is:\nexpression: The mathematical
expression to evaluate.\n\nThe human has directly provided the full expression to evaluate in
their request: "1,984,135 * 9,343,116". This contains all the information needed for the
required expression parameter.\n\nSo I have the necessary information to invoke the
calculator tool.\n</thinking>', type='text'),

ContentBlockToolUse(id='toolu_01V2mzqp5qkB5QucRFjJUJLD',
input={'expression': '1984135 * 9343116'},
name='calculator', type='tool_use')]

Tool Used: calculator
Tool Input: {'expression': '1984135 * 9343116'}
Tool Result: 18538003464660
[ContentBlock(text='Therefore, the result of 1,984,135 * 9,343,116 is 18,538,003,464,660.',
type='text')]

Final Response: Therefore, the result of 1,984,135 * 9,343,116 is 18,538,003,464,660.
```

Sampling strategies

Entropix Sampling: https://github.com/xjdr-alt/entropix

Vision Language Models: A Primer

Inputs/outputs of a Vision Language Model



Vision Language Models: A Primer

Prompt: Caption this image.

"In this image we can see birds on the wall. In the background, we can see a building with windows and plants in the pots."

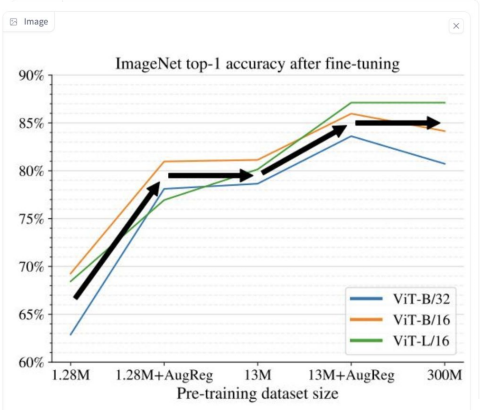
Captioning

Prompt: Parse this receipt as JSON.


```
{
  "items": [
    {
      "name": "Bintang Bremer",
      "Quantity": 1,
      "Price": 59000
    },
    {
      "name": "Chicken H-H",
      "Quantity": 1,
      "Price": 190000
    },
    {
      "name": "Ades",
      "Quantity": 1,
      "Price": 10000
    }
  ],
  "Subtotal": 259000,
  "Service": 9600,
  "Tax": 52416,
  "Discount": 19000,
  "Total": 302016
}
```


OCR + Layout Understanding

Prompt: What does this image show?



"vit-b/32, vit-1/16, and vit-b/16+augreg are trained on different sizes of training data."

Vision Question Answering & Chart Understanding

Vision Language Models Explained, https://huggingface.co/blog/vlms
Accelerating Document AI, https://huggingface.co/blog/document-ai

Vision Language Models: A Primer

The standard VLM architecture

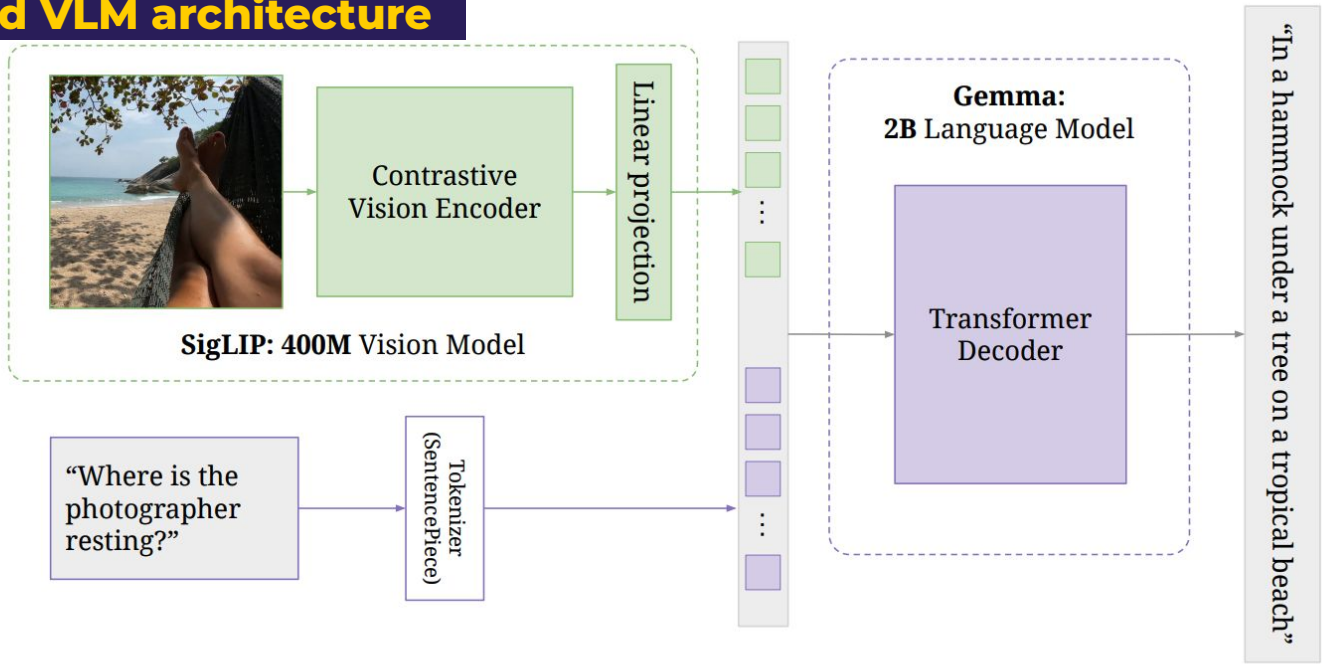


Figure 1 | PaliGemma's architecture: a SigLIP image encoder feeds into a Gemma decoder LM.

Beyer et al. (2024). PaliGemma: A versatile 3B VLM for transfer.