Synthèse d'article : LLaMA – Open and Efficient Foundation Language Models

Analyse d'article dans le cadre du cours validation Numérique , encadré par Mme. Mouakher Amira

Salma BENSMAIL
Université de Perpignan Via Domitia, France
salma.bensmail@etudiant.univ-perp.fr

***Abstract*—L'article présente LLaMA, une famille de modèles de langage (7B, 13B, 33B et 65B paramètres) entraînés sur de très grandes quantités de texte, avec une contrainte importante, c'est d'utiliser uniquement des données publiques. [1]**

L'objectif est de concevoir des modèles offrant de bonnes performances tout en restant plus faciles à utiliser en pratique qu'un modèle unique de très grande taille. Donc cet article montre qu'un modèle de taille raisonnable, lorsqu'il est entraîné sur un volume important de données peut obtenir de bons résultats sur de nombreux benchmarks.

Mais les auteurs soulignent aussi l'importance de considérer le coût d'utilisation du modèle lors de son exploitation, et pas uniquement le coût lié à son entraînement. [1], [4]

***Index Terms*—LLM, pré-entraînement, données publiques, Transformer, évaluation zero-shot, biais, toxicité, empreinte carbone.**

I. Introduction

Les grands modèles de langage (LLM) savent bien générer du texte et apprendre à faire une tâche à partir de seulement quelques exemples, sans entraînement supplémentaire. [2]

Une manière simple d'augmenter les performances est d'augmenter le nombre de paramètres, mais cela rend l'inférence plus coûteuse et peut limiter le déploiement. [1]

Des études récentes indiquent que les performances dépendent d'un compromis entre la taille du modèle, la quantité de données d'entraînement et le budget de calcul. [4]

LLaMA suit cette idée : proposer plusieurs tailles de modèles pour différents budgets d'inférence, tout en restant compétitif. [1] Un autre point clé est la transparence : l'entraînement est fait avec des données publiques, ce qui facilite l'étude et la reproductibilité. [1]

À mon sens, c'est une approche pertinente car elle met l'accent sur l'usage réel (coût d'inférence) et pas uniquement sur le coût d'entraînement. [1]

II. État de l'art (contexte)

Avant LLaMA, plusieurs modèles de référence ont marqué l'évolution du domaine, comme GPT-3, Gopher, Chinchilla ou PaLM. [2]–[5] En parallèle, des initiatives ouvertes comme OPT ou BLOOM ont montré l'intérêt de publier des modèles et des détails d'entraînement. [6], [7] Cependant, il reste difficile de comparer correctement les modèles quand les données sont propriétaires ou peu documentées. [1]

L'article LLaMA se positionne donc comme une famille de modèles compétitifs, entraînés longtemps sur beaucoup de tokens, avec un mélange de données détaillé et public. [1]

Je trouve que ce positionnement est important car il facilite la comparaison scientifique (données décrites) et la reproductibilité des résultats. [1]

III. Problématique et objectifs

La problématique abordée dans l'article peut être résumée par les points suivants :

- Atteindre de bonnes performances sur un large éventail de tâches (bon sens, question-réponse, compréhension, mathématiques et code). [1]
- Limiter le coût d'utilisation du modèle en termes de latence, de mémoire et de budget d'inférence. [1]
- S'appuyer exclusivement sur des données publiques et documenter précisément le corpus d'entraînement. [1]

LLaMA vise donc une approche "utile en pratique", pas seulement un score sur un benchmark. [1]

Je comprends cette problématique comme un compromis assumé : viser une bonne performance tout en gardant des modèles exploitables avec des budgets d'inférence différents. [1]

IV. Contributions

Les contributions principales rapportées sont :

- **Une famille de modèles** (7B, 13B, 33B, 65B) entraînés à grande échelle. [1]
- **Un corpus public et documenté**, avec un mélange de sources variées (web filtré, Wikipédia, code, livres, ArXiv, StackExchange). [1]
- **Des choix d'architecture** et d'optimisation inspirés des LLM récents (stabilité, efficacité). [1]
- **Une évaluation large** en zero-shot et few-shot sur de nombreux benchmarks. [1]
- **Une discussion des risques** : toxicité, biais, vérité/hallucinations et empreinte carbone. [1]

V. Méthodologie

A. Pipeline global

La démarche proposée dans l'article repose sur un pipeline bien défini, comprenant la construction et le filtrage des données, le pré-entraînement de modèles de tailles différentes,

puis leur évaluation sur divers benchmarks, suivie d'une analyse des limites. Ce découpage en étapes me paraît faciliter la compréhension de la méthodologie et l'interprétation des résultats présentés. [1]

La Fig. 1 résume la chaîne complète décrite dans l'article : constitution d'un corpus public, pré-traitement, pré-entraînement de plusieurs tailles, puis évaluation et analyse des risques. [1]

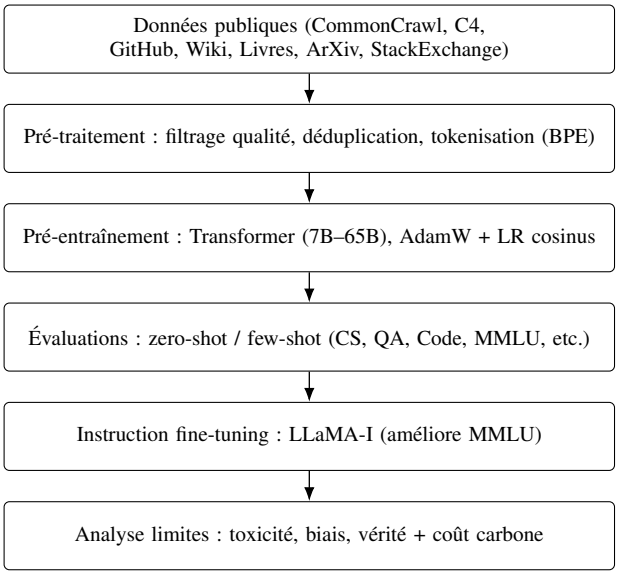

Fig. 1: Pipeline résumant l'approche LLaMA : corpus public, pré-entraînement, évaluation, instruction fine-tuning et analyse des risques.

B. Données de pré-entraînement

Le corpus est un mélange de plusieurs sources, dominé par CommonCrawl filtré, complété par C4, GitHub, Wikipédia, livres, ArXiv et StackExchange. [1]. L'article précise aussi des étapes de filtrage et de déduplication du web (pipeline CCNet) pour augmenter la qualité. [1], [14] Le tokeniseur repose sur une approche BPE via SentencePiece. [1], [15]

Le Tableau I synthétise la proportion de chaque source dans le corpus de pré-entraînement, dominé par CommonCrawl, ce qui reflète l'importance du web filtré dans la recette de LLaMA. [1]

TABLE I: Mélange de données de pré-entraînement (Table 1 de l'article).

Source	Proportion
CommonCrawl	67.0%
C4	15.0%
GitHub	4.5%
Wikipédia	4.5%
Livres (Gutenberg + Books3)	4.5%
ArXiv	2.5%
StackExchange	2.0%

C. Tailles des modèles et hyperparamètres

La famille LLaMA comprend quatre tailles, et les modèles les plus grands sont entraînés sur 1.4T tokens (contre 1.0T tokens pour 7B/13B). [1] Le papier détaille aussi l'architecture (dimension, nombre de têtes, nombre de couches) et certains hyperparamètres. [1]

Le Tableau II montre que LLaMA augmente à la fois en profondeur (nombre de couches) et en largeur (dimension et têtes), et que les plus grands modèles sont entraînés sur davantage de tokens, donc "plus longtemps". [1]

TABLE II: Résumé des tailles et architecture (Table 2 de l'article).

Modèle	Dim.	Têtes	Couches	Tokens
LLaMA-7B	4096	32	32	1.0T
LLaMA-13B	5120	40	40	1.0T
LLaMA-33B	6656	52	60	1.4T
LLaMA-65B	8192	64	80	1.4T

À mon sens, cette lecture "largeur + profondeur + plus de tokens" aide à comprendre pourquoi le coût (temps/mémoire) augmente vite avec la taille. [1]

D. Architecture et entraînement

LLaMA repose sur Transformer, mais avec des choix modernes pour la stabilité et la performance. [1], [9] L'article mentionne notamment la pré-normalisation avec RMSNorm, l'activation SwiGLU, et des embeddings de position de type RoPE. [1], [10]–[12] L'optimisation est faite avec AdamW et un planning de taux d'apprentissage cosinus. [1], [13]

Enfin, un aspect important est l'implémentation efficace : attention optimisée, checkpointing et parallélisme pour réduire mémoire/temps. [1]

Un point important du papier est l'optimisation de l'entraînement : les auteurs utilisent une implémentation d'attention plus efficace (xformers) pour réduire mémoire et temps, puis du checkpointing pour limiter le coût des activations pendant le backward. [1] Ils mentionnent aussi l'usage de parallélisme (modèle / séquence) et le recouvrement calcul-communication pour mieux exploiter un grand nombre de GPU. [1] Je trouve que cette partie est très "HPC", car elle montre que les performances viennent aussi de l'ingénierie (mémoire/communications) et pas seulement des choix théoriques. [1]

VI. Résultats et interprétation

A. Protocole d'évaluation

Les performances sont évaluées en conditions *zero-shot* et *few-shot*, c'est-à-dire sans apprentissage supervisé spécifique pour chaque tâche. [1] Les benchmarks utilisés couvrent plusieurs catégories, notamment le bon sens (questions à choix multiple), la question-réponse dite "closed-book" , la compréhension, les mathématiques et la génération de code. [1]

Pour l'évaluation en question-réponse "closed-book", l'article présente des résultats basés sur la métrique *exact match* obtenus sur des jeux de données tels que NaturalQuestions et TriviaQA, le modèle n'ayant pas accès à des sources externes lors de la génération des réponses. [1]

B. Exemples de résultats (lecture par familles)

Le papier montre une amélioration régulière avec la taille sur beaucoup de tâches. Par exemple, sur des tâches de bon sens (BoolQ, PIQA, HellaSwag, etc.), LLaMA-65B obtient des scores élevés et la famille est globalement compétitive. [1] Sur la question-réponse "closed-book" (NaturalQuestions, TriviaQA), les modèles plus grands obtiennent des gains clairs en few-shot. [1]

Pour le code, les auteurs évaluent HumanEval et MBPP : les scores pass@1 augmentent avec la taille, et LLaMA-65B est compétitif face à des modèles généraux de taille similaire. [1], [19], [20]

Cela suggère que le mélange de données (incluant GitHub) aide aussi ce type de tâches. [1]

À mon sens, cet exemple illustre bien l'effet du corpus (présence de code) sur des capacités spécifiques comme la génération de programmes. [1]

Le Tableau III résume cette tendance, plus le modèle est grand plus les scores de génération de code augmentent, et LLaMA-65B est le meilleur de la famille sur ces deux tests. [1]

TABLE III: Extrait de résultats "Code" (Table 8 de l'article, pass@1).

Modèle	HumanEval	MBPP
LLaMA-7B	10.5	17.7
LLaMA-13B	15.8	22.0
LLaMA-33B	21.7	30.2
LLaMA-65B	23.7	37.7

C. Comparaison avec d'autres modèles (repères)

Pour situer LLaMA par rapport à des modèles précédents, l'article compare les performances sur plusieurs benchmarks représentatifs (bon sens, QA et MMLU). [1] Le Tableau IV donne des repères simples : LLaMA-65B est globalement compétitif face à Chinchilla-70B et PaLM-540B selon la tâche, tandis que LLaMA-13B est souvent proche de GPT-3 175B malgré une taille plus faible. [1]

TABLE IV: Repères de performance vs autres modèles (extraits reformulés).

Benchmark (métrique)	GPT-3 175B	Gopher 280B	Chinchilla 70B	PaLM 540B	LLaMA 65B
BoolQ (0-shot, acc.)	60.5	79.3	83.7	88.0	85.3
NaturalQuestions (64-shot, EM)	29.9	28.2	35.5	39.6	39.9
MMLU (5-shot, avg.)	43.9	60.0	67.5	69.3	63.4

D. MMLU et instruction fine-tuning

Sur MMLU (5-shot), LLaMA-65B est performant mais reste derrière certains modèles concurrents dans l'article. [1], [16] Les auteurs suggèrent que cela peut venir de la proportion limitée de livres et de texte académique dans leur corpus par rapport à d'autres modèles. [1]

Le papier montre aussi qu'un instruction fine-tuning (LLaMA-I) améliore MMLU et la capacité à suivre des consignes. [1], [25] Cela confirme que le pré-entraînement seul ne suffit pas pour obtenir un comportement "assistant" robuste. [1]

VII. Limites et discussion

A. Toxicité

La toxicité est mesurée avec RealToxicityPrompts. Les scores moyens reportés peuvent augmenter avec la taille, y compris sur des consignes "respectful". [1], [21] Cela montre qu'augmenter la taille n'améliore pas automatiquement la sûreté. [1]

B. Biais

Les biais sociaux sont évalués via CrowS-Pairs et d'autres tests. Les résultats indiquent des biais présents dans plusieurs catégories (genre, religion, etc.). [1], [22], [23] Même avec filtrage, le web contient des stéréotypes, et le modèle peut les apprendre. [1] À mon avis, cela justifie d'ajouter des méthodes d'alignement et/ou de contrôle des données, en plus du simple pré-entraînement. [1]

C. Vérité et hallucinations

TruthfulQA évalue la tendance du modèle à produire des réponses vraies (et informatives). Les scores progressent avec la taille, mais restent loin d'un comportement parfaitement fiable. [1], [24] Donc, pour des usages sensibles, il faut des stratégies de contrôle (vérification, accès à des sources, etc.). [1]

D. Discussion (lecture critique)

Le papier défend l'idée qu'à budget donné, entraîner plus longtemps un modèle plus petit peut être plus intéressant pour l'inférence qu'un modèle énorme entraîné moins longtemps. [1] En revanche, certaines limites (biais, toxicité, hallucinations) restent présentes même quand la taille augmente, donc "plus grand" ne veut pas forcément dire "plus sûr". [1]

Enfin, les auteurs notent que la part relativement faible de livres et de texte académique peut expliquer une partie des écarts sur MMLU par rapport à d'autres modèles. [1]

VIII. Coût énergétique et impact

L'article discute l'énergie consommée et l'empreinte carbone de l'entraînement. Les auteurs donnent des estimations (MWh et tCO_2eq) et comparent différentes tailles, ainsi que d'autres entraînements, sous des hypothèses communes. [1], [26] Cela rappelle que publier des modèles peut éviter de ré-entraîner plusieurs fois des systèmes similaires, mais que le coût initial reste important. [1]

Le Tableau V met en évidence un compromis clair, quand on augmente la taille, le coût énergétique et carbone de l'entraînement augmente fortement. [1]

TABLE V: Empreinte estimée (Table 15 de l'article).

Modèle	Énergie (MWh)	Carbone (tCO_2eq)
LLaMA-7B	36	14
LLaMA-13B	59	23
LLaMA-33B	233	90
LLaMA-65B	449	173

IX. Perspectives

Plusieurs pistes d'amélioration, en cohérence avec les conclusions de l'article, peuvent être envisagées :

- Renforcer la fiabilité des réponses et limiter les hallucinations, en particulier lors de la génération libre. [1]
- Atténuer les biais et la toxicité à travers un meilleur choix des données, des méthodes de filtrage et des techniques d'alignement. [1]
- Diminuer le coût de l'inférence, notamment via des approches comme la quantification ou la distillation, afin de faciliter le déploiement. [1]
- Analyser l'influence de la composition du corpus d'entraînement, par exemple en augmentant la part de livres et de textes académiques, sur des benchmarks tels que MMLU. [1]

X. Conclusion

LLaMA propose une famille de modèles performants, avec un compromis clair entre taille, coût d'inférence et qualité. [1]

L'approche est convaincante car elle combine un corpus public documenté, des choix d'architecture modernes, et une évaluation large sur des tâches variées. [1] Cependant, l'article met aussi en évidence des limites importantes (biais, toxicité, hallucinations) et un coût énergétique non négligeable, ce qui justifie des travaux futurs sur l'alignement et l'efficacité. [1] Selon moi, le message principal est qu'il faut évaluer un LLM à la fois sur la performance et sur la faisabilité pratique (inférence, coût, risques). [1]

XI. Lien avec mon stage (HPC & optimisation)

Même si l'article parle de modèles de langage, il est en lien avec mon stage orienté HPC/optimisation, parce que les auteurs insistent beaucoup sur l'efficacité et pas seulement sur la précision. [1]

Le papier montre qu'en pratique, il faut optimiser le temps de calcul, la mémoire et les communications entre GPU pour entraîner des modèles très grands. [1]

Concrètement, les auteurs décrivent des astuces d'implémentation (attention plus efficace, checkpointing, parallélisme, recouvrement calcul/communication) qui ressemblent aux problématiques qu'on rencontre en calcul haute performance. [1] Cela correspond bien à ce que je cherche dans mon stage : améliorer la performance d'un pipeline (accélérer, réduire la mémoire, mieux utiliser les ressources). [1] Je trouve que ce parallèle est intéressant, car il montre que les compétences HPC sont directement utiles dans les projets LLM à grande échelle. [1]

Enfin, l'article discute aussi du coût énergétique et de l'empreinte carbone de l'entraînement, ce qui montre que l'optimisation a aussi un impact "green computing", pas seulement un gain de vitesse. [1]

References

[1] H. Touvron et al., "LLaMA: Open and Efficient Foundation Language Models," arXiv:2302.13971, 2023.

[2] T. B. Brown et al., "Language Models are Few-Shot Learners," NeurIPS, 2020.

[3] J. W. Rae et al., "Scaling Language Models: Methods, Analysis & Insights from Training Gopher," arXiv:2112.11446, 2021.

[4] J. Hoffmann et al., "Training Compute-Optimal Large Language Models," arXiv:2203.15556, 2022.

[5] A. Chowdhery et al., "PaLM: Scaling Language Modeling with Pathways," arXiv:2204.02311, 2022.

[6] S. Zhang et al., "OPT: Open Pre-trained Transformer Language Models," arXiv:2205.01068, 2022.

[7] T. L. Scao et al., "BLOOM: A 176B-Parameter Open-Access Multilingual Language Model," arXiv:2211.05100, 2022.

[8] S. Black et al., "GPT-NeoX-20B: An Open-Source Autoregressive Language Model," arXiv:2204.06745, 2022.

[9] A. Vaswani et al., "Attention is All You Need," NeurIPS, 2017.

[10] B. Zhang and R. Sennrich, "Root Mean Square Layer Normalization," NeurIPS, 2019.

[11] N. Shazeer, "GLU Variants Improve Transformer," arXiv:2002.05202, 2020.

[12] J. Su et al., "RoFormer: Enhanced Transformer with Rotary Position Embedding," arXiv:2104.09864, 2021.

[13] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," arXiv:1711.05101, 2017.

[14] G. Wenzek et al., "CCNet: Extracting High Quality Monolingual Datasets from Web Crawl Data," LREC, 2020.

[15] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing," arXiv:1808.06226, 2018.

[16] D. Hendrycks et al., "Measuring Massive Multitask Language Understanding," arXiv:2009.03300, 2020.

[17] T. Kwiatkowski et al., "Natural Questions: A Benchmark for Question Answering Research," TACL, 2019.

[18] M. Joshi et al., "TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension," arXiv:1705.03551, 2017.

[19] M. Chen et al., "Evaluating Large Language Models Trained on Code," arXiv:2107.03374, 2021.

[20] J. Austin et al., "Program Synthesis with Large Language Models," arXiv:2108.07732, 2021.

[21] S. Gehman et al., "RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models," arXiv:2009.11462, 2020.

[22] N. Nangia et al., "CrowS-Pairs: A Challenge Dataset for Measuring Social Biases in Masked Language Models," EMNLP, 2020.

[23] R. Rudinger et al., "Gender Bias in Coreference Resolution," NAACL-HLT, 2018.

[24] S. Lin et al., "TruthfulQA: Measuring How Models Mimic Human Falsehoods," arXiv:2109.07958, 2021.

[25] H. W. Chung et al., "Scaling Instruction-Finetuned Language Models," arXiv:2210.11416, 2022.

[26] C.-J. Wu et al., "Sustainable AI: Environmental Implications, Challenges and Opportunities," MLSys, 2022.