



HAL
open science

Derby LLM : Évaluation comparative des approches RAG et fine-tuning

Christophe Bouvard, Mathieu Ciancone, Antoine Gourru, Marion Schaeffer

► To cite this version:

Christophe Bouvard, Mathieu Ciancone, Antoine Gourru, Marion Schaeffer. Derby LLM : Évaluation comparative des approches RAG et fine-tuning. 10^{ème} Conférence Nationale sur les Applications Pratiques de l'Intelligence Artificielle, AFIA-Association Française pour l'Intelligence Artificielle, Jul 2024, La Rochelle, France. pp.38-47. <hal-04638460>

HAL Id: hal-04638460

<https://hal.science/hal-04638460v1>

Submitted on 10 Jul 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY-NC 4.0 - Attribution - Non-commercial use - International License

Derby LLM : Évaluation comparative des approches RAG et fine-tuning

Christophe Bouvard¹, Mathieu Ciancone¹, Antoine Gourru², Marion Schaeffer¹

¹ Wikit, Lyon, France

² Laboratoire Hubert Curien, UMR CNRS 5516, Saint-Etienne, France

{christophe, mathieu, marion}@wikit.ai, antoine.gourru@univ-st-etienne.fr

Résumé

Les grands modèles de langage ont récemment été largement exploités dans les agents conversationnels, où l'injection de connaissances pour des domaines d'applications spécifiques est un enjeu crucial. Nous comparons deux approches : le fine-tuning et la génération augmentée de récupération. Nous évaluons ces techniques pour deux cas d'usage différents avec des métriques automatiques et la préférence humaine. Bien que la pertinence des réponses soit proche, la fidélité et la préférence humaine avantagent la génération augmentée de récupération.

Mots-clés

LLM, RAG, fine-tuning, benchmark, évaluation.

Abstract

Large language models have recently been widely exploited in conversational agents, where knowledge injection for specific application domains is a crucial issue. We compare two approaches : fine-tuning and retrieval augmented generation. We evaluate these techniques in two different use cases leveraging automatic metrics and human preference. Although the answer relevance is close, faithfulness and human preference favour retrieval augmented generation.

Keywords

LLM, RAG, fine-tuning, benchmark, evaluation.

1 Introduction

Les grands modèles de langage (*Large Language Models*, LLM) sont aujourd'hui utilisés dans diverses applications et projets de recherche. Leur pré-entraînement leur permet d'acquérir les règles morpho-syntaxiques d'une ou plusieurs langues et une grande quantité d'informations factuelles, leur donnant les connaissances nécessaires à la réalisation de tâches dans des domaines variés où le traitement automatique du langage naturel en est le cœur. Cependant, les connaissances du modèle sont statiques et évoluent seulement lorsqu'il est ré-entraîné. Elles sont également limitées à l'étendue du contenu des données d'entraînement, généralement non spécialisées, et peuvent ainsi manquer d'expertise sur des domaines spécifiques.

Ce manque de connaissances conduit à des hallucinations, c'est-à-dire un contenu généré qui n'a pas de sens ou qui n'est pas conforme au contenu source fourni [8]. Les hallucinations compliquent l'exploitation des LLM dans des cas où des compétences spécifiques sur des domaines fermés sont nécessaires, comme par exemple pour les agents conversationnels. Pourtant, les LLM restent largement utilisés pour leur capacité de génération de texte qui est un véritable atout. C'est pourquoi les travaux récents [3, 5, 12] cherchent à spécialiser les modèles grâce à des connaissances spécifiques au cas d'usage.

Différentes techniques ont été proposées pour injecter de la connaissance dans les LLM. La première est le fine-tuning [13, 20, 22], qui consiste à poursuivre l'entraînement du modèle sur une tâche et/ou un domaine spécifique pour le spécialiser. La seconde est la génération augmentée de récupération (*Retrieval Augmented Generation*, RAG) [10] où une étape de recherche d'information permet d'extraire du contenu pertinent pour répondre à la question. Ce contenu est ensuite associé à la question pour générer la réponse à l'aide du LLM. Ces techniques possédant chacune leurs avantages et inconvénients, le choix de l'une plutôt que l'autre s'avère complexe.

Nous proposons une comparaison des performances des techniques de RAG et de fine-tuning en français avec le modèle open-source *Mistral 7B* [9]. Nous utilisons des corpus spécifiques accessibles publiquement, parfaitement représentatifs d'un cas d'application d'agent conversationnel. Les métriques étant un réel enjeu pour l'évaluation, nous utilisons deux approches complémentaires : une arène de modèles permettant à des utilisateur-riche-s de voter en aveugle pour leur réponse préférée, ainsi que des métriques automatiques basées sur la pertinence des réponses et leur fidélité aux informations contenues dans les données. L'ensemble des données et du code produit est accessible sur demande auprès des auteur-riche-s.

2 Travaux antérieurs

De récents travaux ont comparé la RAG au fine-tuning d'un LLM [3, 5, 12]. D'après les analyses, la RAG obtient de meilleurs résultats que le fine-tuning. Par exemple, le mo-

dèle *gpt-3.5-turbo-0613* d'OpenAI¹ est comparé lorsqu'il est interrogé seul, fine-tuné ou avec de la RAG [5]. Les données utilisées portent sur un domaine spécifique, avec 100 questions dont les réponses ne sont pas disponibles dans la connaissance interne du modèle pour l'évaluation. Le fine-tuning est effectué avec le format de données rôle-contenu d'OpenAI et la quantité n'est pas précisée, tout comme pour la RAG. Les résultats sont obtenus en comptant les réponses correctes, les faux négatifs et les faux positifs.

Ce type de comparaison est étendu à d'autres modèles [12], par exemple *Mistral 7B* [9], *Llama2 7B* [17] et *Orca2 7B* [11]. Les modèles obtiennent également de meilleures performances en utilisant seulement la RAG plutôt qu'en combinant fine-tuning et RAG sur divers domaines plus génériques du dataset Massive Multilingual Language Understanding Evaluation (MMLU) [7]. Les auteur·rice·s de [12] justifient cela par la difficulté d'apprentissage de nouvelles informations factuelles des LLM lors d'un fine-tuning non-supervisé. Cependant, la répétition de nombreuses variations d'un même fait pendant cette étape peut atténuer cette difficulté. Les données utilisées pour la RAG et le fine-tuning sont extraites de Wikipedia, mais la quantité n'est pas précisée. Cette fois, la métrique utilisée est la précision de la log-vraisemblance des scores attribués par le modèle. Les modèles *Vicuna* [23] et *GPT-4*² ont également été testés sur des domaines spécifiques [3]. La quantité de données utilisées pour la RAG et le fine-tuning est à nouveau non précisée dans l'article. De nombreuses métriques y sont présentées comme la cohérence, la pertinence, la fidélité (*groundedness*) et la complétude, toutes évaluées par un LLM. La mise en pratique et le coût de chacune des techniques est aussi discutée.

Les travaux précédemment évoqués sont effectués sur des textes en anglais, nous proposons donc une comparaison des performances des techniques de RAG et de fine-tuning sur des textes en français. Nous discutons plus particulièrement des problèmes dus, pour le fine-tuning, aux spécificités des textes contenus dans les deux datasets utilisés, et pour la RAG des problèmes de récupération des fragments réellement utiles.

3 Méthodologie

Dans cette section, nous décrivons le processus expérimental mis en place pour comparer la RAG et le fine-tuning d'un LLM.

3.1 La compétition

Nous avons créé deux jeux de données différents qui illustrent le cas d'application d'un agent conversationnel de support pour des usager·ère·s.

Département de la Drôme Le premier jeu de données est issu du site internet du département de la Drôme³. 196 contenus de pages ont été collectés via l'API ouverte sur ledit site. Chaque contenu HTML a été converti en Mark-

down avec *markdownify*⁴ puis découpé en 357 fragments (*chunks*) de façon automatique en s'appuyant sur les titres. Les pages initiales ont une taille moyenne de 1973 caractères avec un écart type de 2898 caractères alors que les fragments obtenus ont une taille moyenne de 1109 caractères avec un écart type de 1483 caractères. Le processus de découpage du contenu en fragments réduit presque de moitié la taille des documents considérés bien que l'écart type nous indique une grande variabilité dans la taille des fragments. 80 questions ont été créées manuellement en sélectionnant de façon aléatoire des fragments de texte et en rédigeant une question dont la réponse est contenue dans le fragment considéré.

Télécom Saint-Etienne Le second jeu de données a été obtenu grâce au site internet de Télécom Saint-Etienne⁵. 502 pages Web ont été collectées par *web scraping*. Le contenu HTML a de nouveau été converti en Markdown avec *markdownify* pour obtenir 1114 fragments. Les pages initiales ont une taille moyenne de 5166 caractères avec un écart type de 38100 caractères alors que les fragments obtenus ont une taille moyenne de 2336 caractères avec un écart type de 4401 caractères. Cette fois, le processus de découpage réduit de plus de moitié la taille des documents considérés et diminue l'écart-type de façon drastique, bien qu'il reste assez important ce qui dénote ici aussi une grande variabilité dans la taille des fragments. 100 questions ont été annotées manuellement en sélectionnant de façon aléatoire des fragments et en rédigeant une question dont la réponse est contenue dans le fragment considéré.

A noter que le découpage des fragments est effectué en s'appuyant sur l'arborescence du plan. Les sous-parties sont découpées individuellement et les titres des parties et sous-parties sont ajoutés au début du fragment. L'ensemble des fragments a été utilisé pour le fine-tuning du modèle et pour la recherche d'information de la RAG.

3.2 Les joueurs

Pour comparer les différentes techniques d'injection de connaissances, nous utilisons le modèle *Mistral 7B* (plus précisément *Mistral-7B-Instruct-v0.2*) avec différentes configurations.

Mistral 7B *Mistral 7B* [9] est un LLM à 7 milliards de paramètres développé par l'entreprise Mistral AI. Comme l'ensemble des modèles de langage, il est entraîné sur un très grand corpus de documents (principalement constitué de données ouvertes du Web) en anglais, sur une tâche de prédiction du mot suivant. Son architecture repose sur un Transformer [18] de type *decoder*. Il dépasse les performances de modèles plus grands comme *Llama 13B* [17] sur de nombreux benchmarks d'évaluation [9] et intègre plusieurs innovations permettant notamment d'accélérer le calcul de l'attention. Le modèle que nous utilisons est instruit, c'est-à-dire qu'il est ensuite entraîné à suivre des instructions. Nous avons fait le choix de fine-tuner ce modèle plutôt que la version non instruite de manière à pouvoir com-

1. <https://platform.openai.com/docs/models/gpt-3-5>

2. <https://platform.openai.com/docs/models/gpt-4-and-gpt-4-turbo>

3. <https://www.ladrome.fr/>

4. <https://github.com/matthewwithanm/python-markdownify>

5. <https://www.telecom-st-etienne.fr/>

parer des modèles équivalents, or le modèle instruit possède de meilleures compétences en RAG. La poursuite du pré-entraînement (tâche de prédiction du prochain token) pourrait dégrader les compétences acquises grâce à la phase d'alignement, mais il est difficile dans ce contexte de modéliser la tâche d'injection de connaissance comme un objectif de préférence compatible avec les optimisations RLHF ou DPO.

Procédure de fine-tuning Pour injecter de nouvelles informations dans le modèle de langage, nous adoptons l'approche classique de continuation du pré-entraînement sur la tâche de génération conditionnelle (*Mistral* étant un modèle auto régressif). Cette même approche a été utilisée par les auteur-riche-s de [3]. Elle consiste à entraîner le modèle avec un faible pas d'apprentissage sur la tâche de prédiction du mot suivant sur l'ensemble du corpus contenant les informations que l'on souhaite injecter dans le modèle.

Pour ce faire, nous utilisons le protocole de parallélisation DeepSpeed⁶ développé par Microsoft pour affiner le modèle sur 3 *Titan RTX* de 24 Gb de VRAM avec l'approche Zero Stage 3 [14]. Les jeux de données étant relativement petits, l'entraînement dure seulement deux heures pour les données du département de la Drôme et quatre heures pour les données de Télécom Saint-Etienne. Nous utilisons les hyperparamètres classiques [3] : un pas d'apprentissage de $2e-5$ sur deux itérations d'apprentissage (*epochs*) en lots de 16 fragments de texte et un planificateur cosinus (*cosine scheduler*), avec un ratio d'échauffement (*warmup*) de 10% des pas d'entraînement.

Nous utilisons les outils mis à disposition par Hugging Face, notamment le modèle *mistralai/Mistral-7B-Instruct-v0.2*, et la fonction *Trainer* de la librairie Transformers⁷.

Nous rappelons que le but est d'injecter la connaissance dans le modèle. Nous ne disposons donc pas de jeu de données de validation : le contenu informationnel des textes qui n'a pas été vu dans l'entraînement ne peut pas être intégré dans la connaissance du LLM. Pour cette raison, nous avons adopté les hyperparamètres de bases cités plus haut pour le fine-tuning du modèle. Néanmoins, une inspection qualitative semble montrer qu'un pas d'apprentissage ou un nombre de passes d'apprentissage trop élevés mènent à une dégradation de la qualité du contenu généré par le modèle (contenu non structuré, hallucination) alors qu'à l'inverse les réduire empêche le modèle d'apprendre le contenu spécifique. Un exemple typique de sur-apprentissage observé dans nos expériences pour le jeu de données de La Drôme (pas d'apprentissage trop élevé) : *Question* : "Présente-moi le projet pour le Pont du Tram" *Réponse* : "La Drôme est le Département de la Drôme. Le Département de la Drôme est la Drôme du Département de la Drôme.", ou encore *Question* : "Quelle est la plus haute chaîne de montagne de France?" *Réponse* : "Le massif du Vercors.". Ce point central est encore une question ouverte et a été peu exploré par les travaux précédents. Ce problème s'écarte du champ d'étude du présent article mais fera l'objet de futures contri-

butions. Lorsque le modèle est fine-tuné, il est ensuite interrogé pour répondre aux questions avec une température à 0 et le prompt suivant :

Département de la Drôme <s> [INST] Ceci est une conversation entre un humain et un assistant du département de la Drôme. L'assistant donne uniquement sa réponse. Sa réponse doit être courte et en français. La réponse est toujours en lien avec le département de la Drôme.
{question} [/INST]

Télécom Saint-Etienne <s> [INST] Ceci est une conversation entre un humain et un assistant de l'école d'ingénieur Télécom Saint-Etienne. L'assistant donne uniquement sa réponse. Sa réponse doit être courte et en français. La réponse est toujours en lien avec Télécom Saint-Etienne.
{question} [/INST]

Génération augmentée de récupération (RAG) La génération augmentée de récupération est fondée sur trois étapes principales :

1. Préparation de la base de données à exploiter pour la recherche d'information (sous-étapes *1a* et *1b* de la figure 2) ;
2. Récupération des éléments de contexte proches de la requête de l'utilisateur-riche (sous-étapes *2a*, *2b* et *2c*) ;
3. Génération de la réponse (sous-étapes *2d* et *2e*).

Le pré-requis est la préparation des données spécifiques au cas d'application. Il s'agit de documents dont les données textuelles sont extraites, nettoyées, découpées en fragments (*chunking*), vectorisées (création d'*embeddings*, le cas échéant) puis indexées dans une base de données.

À l'arrivée d'une requête formulée par l'utilisateur-riche, la tâche de récupération (*retrieval*) consiste en la recherche d'information liée à la phrase exprimée. Elle est opérée par exemple via une méthode de pondération (e.g. BM25 [15] ou TF-IDF [16]), ou via une méthode de calcul vectoriel [1] (e.g. similarité cosinus entre *embeddings*), voire une hybridation de ces méthodes.

Ensuite, à partir du contexte constitué par la tâche de récupération, la tâche de génération implique un LLM pour synthétiser la réponse.

Dans ces travaux, nous implémentons le processus de RAG avec la solution commerciale Wikit Semantics⁸. Le modèle d'*embeddings text-embedding-ada-002* a été utilisé pour la partie récupération car il affiche de très bonnes performances pour cette tâche en français d'après le leaderboard MTEB⁹. Le modèle *Mistral-7B-Instruct-v0.2* (ou *mistral-tiny*) décrit précédemment a également été utilisé pour l'étape de génération. Pour l'étape de récupération, les 10 fragments ayant la similarité cosinus la plus élevée avec la question sont conservés pour former le contexte lors de la génération. La génération est faite avec une température à 0 et le prompt suivant :

6. <https://github.com/microsoft/DeepSpeed>

7. <https://huggingface.co/docs/transformers/index>

8. <https://www.wikit.ai/fonctionnalites/wikit-semantics>

9. <https://huggingface.co/spaces/mteb/leaderboard>

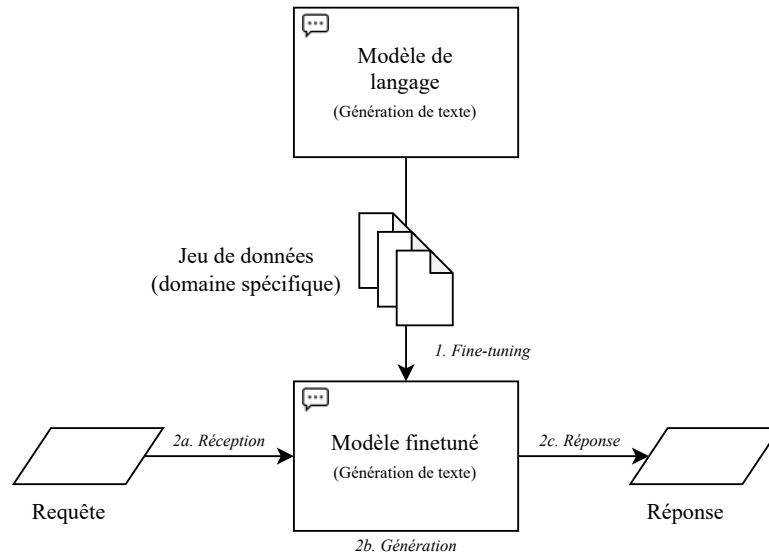


FIGURE 1 – Fine-tuning (1) et génération via le LLM (2)

Système Vous êtes un assistant utile qui répond aux questions à propos du département {de la Drôme | de l'école d'ingénieur Télécom Saint-Etienne}.

Humain Répondre à la question en utilisant uniquement le contexte ci-après. Si la question ne peut pas être répondue avec le contexte, répondre "Je ne sais pas".

Humain Contexte : {contexte}

Humain {question}

3.3 L'arbitrage

Deux stratégies différentes ont été utilisées de façon complémentaire pour évaluer la RAG et le fine-tuning : le vote en aveugle par des annotateur-ric-e-s pour la technique de génération favorite et des métriques calculées automatiquement.

3.3.1 L'arène de modèles

L'évaluation de la préférence des réponses est réalisée via une interface utilisateur-ric-e dédiée présentée figure 3, inspirée de la Chatbot Arena¹⁰. L'écran principal présente le contexte de la question, la question, et les deux réponses générées respectivement par le LLM fine-tuné et par la RAG. L'ordre d'affichage des réponses est aléatoire, sans indication de l'approche de génération. Les boutons mis à disposition des évaluateur-ric-e-s pour l'évaluation des réponses sont :

- "A est meilleure" pour indiquer que la réponse présentée à gauche est préférée ;
- "B est meilleure" pour celle de droite ;
- "Match nul" lorsque l'évaluateur-ric-e considère que les deux réponses se valent ;
- "Aucune" si aucune des réponses n'est considérée comme pertinente.

Les votes ont été réalisés par trois personnes expertes du domaine des agents conversationnels mais non expertes des domaines d'application utilisés pour les expérimentations. A titre d'illustration, nous présentons en Figures 4, 5, et 6 quelques cas annotés.

3.3.2 Métriques automatiques

De nombreux outils sont disponibles pour évaluer les réponses générées par un LLM, en particulier pour la RAG, comme RAGAS [6], Athina Evals¹¹ ou encore UpTrain¹². Nous nous intéressons aux métriques pouvant s'appliquer aussi bien au fine-tuning qu'à la RAG, qui ne nécessitent pas d'autres annotations que celles dont nous disposons déjà, et qui rendent compte de la qualité de réponse des LLM dans le cas d'application à des agents conversationnels.

Pertinence La première métrique que nous sélectionnons est la pertinence, qui traduit le terme *answer relevance* largement répandu dans la littérature [6]. Avec cette métrique, nous cherchons à mesurer à quel point la réponse générée par le LLM est pertinente au vu de la question posée. La véracité des informations apportées n'est pas vérifiée, cette métrique se concentre sur le périmètre de la question, en vérifiant que la réponse le couvre entièrement. Pour calculer la pertinence d'une réponse, nous nous inspirons de l'implémentation d'Athina Evals qui utilise un prompt contenant une question, une réponse et des instructions pour qu'un LLM évalue avec une valeur booléenne si la réponse donnée répond à la question posée, en justifiant l'évaluation avec des explications. Nous modifions le prompt pour obtenir un score plutôt qu'une valeur booléenne indiquant si oui ou non la réponse est pertinente par rapport à la question. Un LLM de plus grande taille génère donc un score et une explication qui représente la pertinence de la réponse

10. <https://chat.lmsys.org/>

11. <https://github.com/athina-ai/athina-evals>

12. <https://docs.uptrain.ai/predefined-evaluations/overview>

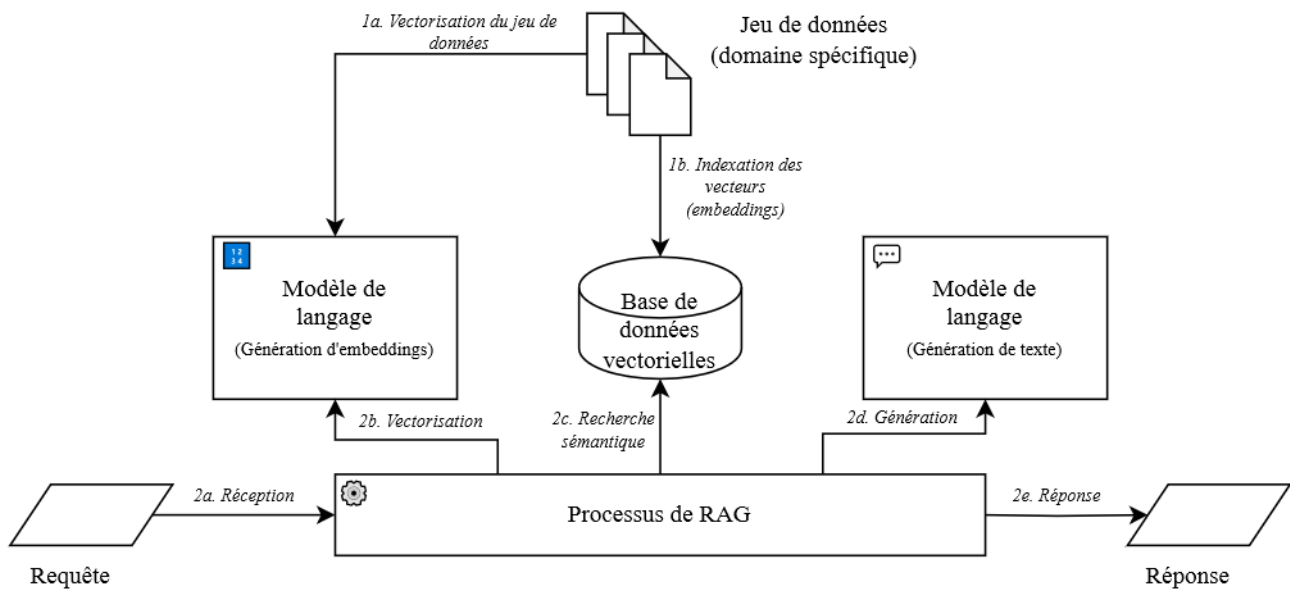


FIGURE 2 – Indexation des données (1) et génération via le processus de RAG (2)

Contexte

Séances publiques

Pour vous rendre et assister aux séances publiques

Salle Marius Moutet Conseil départemental de la Drôme Hôtel du Département 26 avenue du président Herriot 26000 VALENCE

Réponse A

🔴 Où se déroulent les scéances publiques ?

🟡 Les séances publiques du Conseil départemental de la Drôme se déroulent à la Salle Marius Moutet, Conseil départemental de la Drôme, Hôtel du Département, 26 avenue du président Herriot, 26000 VALENCE.

Réponse B

🔴 Où se déroulent les scéances publiques ?

🟡 Les séances publiques se déroulent au Conseil Départemental, 26 avenue du Président Herriot 26026 Valence Cedex 9.

Evaluation

A est meilleure
 B est meilleure
 Match nul
 Aucune

Evaluateur·rice : Christophe

Code : D26

FIGURE 3 – Interface de l'arène d'évaluation.

Question : *Qu'est ce que la cartoucherie ?*

Contexte (extrait) : *[...] l'ancienne cartoucherie nationale de Bourg-lès-Valence est devenue pôle d'excellence régional de l'image animée. En plus de nombreux studios, la Cartoucherie accueille l'école de La Poudrière (reconnue par le Ministère de la Culture) et l'ECAS (école du studio TeamTo, spécialisée dans la formation à l'animation 3D).*

Réponse Fine-tuning : *Une cartoucherie est un site d'exploitation de la tourbe, utilisé pour la fabrication des cartouches de fusil. (pour plus d'informations : <https://www.ladrome.fr/mon-quotidien/economie/les-entreprises/les-entreprises-de-la-filiere-de-la-tourbe/>)*

Réponse RAG : *La Cartoucherie est un pôle d'excellence régional de l'image animée situé à Bourg-lès-Valence en Drôme. Il abrite de nombreux studios de production de films d'animation, l'école de La Poudrière, et l'ECAS (école du studio TeamTo), spécialisée dans la formation à l'animation 3D.*

FIGURE 4 – Exemple de question-réponses dont la réponse préférée est celle de la RAG. Ce premier exemple montre une bonne formulation de la réponse par la RAG, alors que celle issue du LLM fine-tuné comporte une hallucination montrant que l'étape de fine-tuning n'a pas réussi à spécialiser suffisamment le LLM, ainsi qu'un lien hyper-texte invalide, ce qui est un problème assez récurrent avec le fine-tuning.

générée par un LLM de plus petite taille. Nous utilisons le modèle d'OpenAI *gpt-3.5-turbo* avec le prompt suivant :

Système *You are an expert at evaluating whether the response answers specifically what the user is asking about, and covers all aspects of the user's query. You are not checking for correctness, or factual accuracy. You are only checking if the response answers the user's query.*

Humain *Consider the following :*
user's query : {question}
response : {response}

Humain *Determine if the response answers specifically what the user is asking about, and covers all aspects of the user's query. Provide a brief explanation of why the response does or does not answer the user's query sufficiently, labeled as 'explanation', leading up to a score between 0 and 5 labeled as 'result'.*

Humain *Return a JSON object in the following format :*
'result' : result, 'explanation' : explanation.

Comme le score obtenu est compris entre 0 et 5, il est ensuite normalisé. L'explication demandée au modèle lui permet de générer des scores plus proches du résultat attendu, probablement grâce au principe des *chain-of-thought* [21]. De même, le modèle génère des scores plus variés lorsqu'on impose que le score soit entre 0 et 5 plutôt qu'entre 0 et 1 (où le modèle a tendance à ne générer que des 0 ou des 1).

Fidélité La seconde métrique que nous considérons est la fidélité, qui traduit le terme *faithfulness* utilisé dans la littérature [6]. Cette mesure représente la véracité factuelle de la réponse générée par le LLM. La fidélité peut facilement être liée à la détection d'hallucinations. La plupart des outils disponibles utilisent également un LLM pour l'évaluer. Dans le but de limiter notre utilisation de très gros modèles de langues, nous avons cherché à effectuer cette tâche sans LLM. Nous utilisons donc spaCy¹³ pour extraire les passages d'intérêt de la réponse et des fragments identifiés comme pertinents pour répondre à la question. Les passages d'intérêts sont les entités identifiées par spaCy grâce à son module de reconnaissance d'entités pré-entraîné, ainsi que les adresses de sites internet, les adresses e-mail et nombres (tels que les codes postaux ou numéros de téléphones). Soit $PI_{reponse}$ l'ensemble des passages d'intérêts de la réponse et $PI_{fragment}$ l'ensemble des passages d'intérêts du fragment pertinent correspondant. Le score de fidélité est alors calculé d'après l'équation 1 comme la proportion des passages d'intérêts de la réponse présents dans le fragment pertinent. Les passages d'intérêts étant des chaînes de caractères, la correspondance entre les passages d'intérêts de la réponse et ceux du fragment pertinent est calculée par égalité des chaînes de caractères.

$$fidélité = \frac{card(PI_{reponse} \cap PI_{fragment})}{card(PI_{reponse})} \quad (1)$$

13. <https://spacy.io/>

Question : <i>Quelle est l'activité de la société Sick ?</i>
Contexte (extrait) : <i>Télécom Saint-Etienne signe une Chaire Innovation avec SICK France [...] En cohérence directe avec les enseignements du cycle ingénieur, SICK vient apporter dans le cadre de cet accord son expertise dans le domaine de la captation, de la donnée et de l'intelligence artificielle auprès de nos étudiants. [...]</i>
Réponse Fine-tuning : <i>La société SICK produit des capteurs pour la captation de données.</i>
Réponse RAG : <i>Sick is a leading global supplier of sensor solutions for industrial applications.</i>

FIGURE 5 – Exemple de question-réponses dont la réponse préférée est celle générée par le LLM fine-tuné. Cet exemple montre une génération en anglais de la RAG, et une formulation du LLM qui est préférée par les évaluateur-riche-s.

4 Résultats

Dans cette section, nous analysons les résultats des comparaisons de la RAG et du fine-tuning sur le jeu de données du département de la Drôme et celui de Télécom Saint-Etienne.

4.1 Résultats des annotations de l'arène

Nous analysons d'abord la cohérence des évaluations effectuées par les annotateur-riche-s.

Comme indiqué sur la figure 7, les 3 annotateur-riche-s sont en accord total sur la réponse préférée dans 68% des cas. Dans 29% des cas, deux des trois annotateur-riche-s sont en accord. Nous pouvons donc conclure à un bon accord global entre les 3 annotateur-riche-s. Ces chiffres sont étayés par les scores du Kappa de Cohen [2] calculés pour les 3 paires d'annotateur-riche-s. Les scores obtenus pour chacune des paires sont peu variables, avec une valeur moyenne de 0.55 indiquant un accord modérément élevé entre chaque annotateur-riche-s. Notons que ce score, bien qu'il représente un indicateur utile, souffre d'une fiabilité diminuée lorsque les classes étudiées sont déséquilibrées[19], comme explicité dans le prochain paragraphe.

TABLE 1 – Accord inter-annotateur-riche-s

	P1/P2	P2/P3	P1/P3
Kappa de Cohen	0.545	0.539	0.577

Les réponses générées par la RAG sont largement préférées par les annotateur-riche-s d'après la figure 8. En effet, sur les 180 questions du jeu de données, la réponse de la RAG est en moyenne préférée dans $67\% \pm 3\%$ des cas. En revanche, la réponse générée par le modèle fine-tuné n'est préférée en moyenne que dans $3\% \pm 0\%$ des cas. Ce résultat s'explique par le fait que la réponse du modèle fine-tuné souffre d'un manque de justesse notable. Souvent, le modèle génère une

<p>Question : <i>Quand auront lieu les projets d'ingénierie de 2023 ?</i></p>
<p>Contexte (extrait) : <i>Lancement des projets d'ingénierie (PING) pour nos étudiants ingénieurs en 2e année [...] 16 mars 2023 [...] Les projets d'ingénierie sont lancés! [...] Du 14 mars au 23 juin, les étudiants répartis en équipe de 4 à 8 personnes vont travailler à la réalisation de leur projet à travers 3 étapes-clés : 1 phase de cadrage et 2 "sprints". [...] Au total, les étudiants travailleront l'équivalent de 700 heures réparties en séances de travail de mars à juin. [...]</i></p>
<p>Réponse Fine-tuning : <i>Les projets d'ingénierie de 2023 auront lieu du 20 avril au 14 juin. Pour plus d'informations : <https://www.telecom-st-etienne.fr/entreprise/etre-accompagne/projet-dingenierie/> Partagez Précédent (pour plus d'informations : https://www.telecom-st-etienne.fr/telecom-saint-etienne-fete-erasmus-days-2023/) Suivant (pour plus d'informations : https://www.telecom-st-etienne.fr/telecom-saint-etienne-acteur-du-projet-europeen-digital-creativity-for-digital-maturity-dc4dm/)</i></p>
<p>Réponse RAG : <i>Les projets d'ingénierie de 2023 ont commencé le 16 mars et s'achèveront en juin. Les étudiants travailleront l'équivalent de 700 heures sur leur projet, réparties en trois étapes-clés : une phase de cadrage, un premier sprint et un deuxième sprint. Les projets seront clôturés par une journée de soutenances de projets le 25 juin. Selon la situation sanitaire, les Projets Ingénierie se tiendront en présentiel et/ou en distanciel.</i></p>

FIGURE 6 – Exemple de question-réponses dont aucune des réponses n'est préférée par les évaluateur-ric-e-s. Ce dernier exemple montre une question-réponse liée à une période de temps. Les deux approches proposent des dates incorrectes.

réponse qui semble vraisemblable car elle reprend des références et éléments de langage du jeu de données. Cependant, les informations données sont fausses et ne correspondent pas à l'information qui peut être trouvée dans les documents. Au contraire, dans le cas de la RAG, le modèle a tendance à paraphraser précisément le document pertinent pour répondre à la question, lorsque ce dernier a bien été extrait par l'étape de récupération. Dans $23\% \pm 5\%$ des cas, aucune des deux réponses n'a été jugée satisfaisante pour être utile à l'utilisateur, à cause d'un problème de fond ou de forme (réponses fausses, rédigées en anglais ou incomplètes par exemple).

4.2 Résultat de l'évaluation automatique

Pour l'évaluation grâce aux métriques calculées automatiquement, nous utilisons les réponses générées par la RAG et par le fine-tuning sur chacun des jeux de données. Les résultats sont présentés dans la table 2. Comme l'évalua-

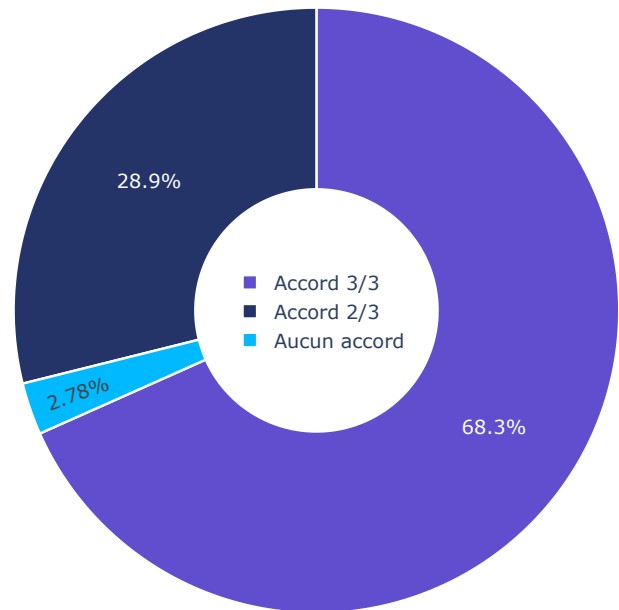


FIGURE 7 – Proportion d'accord entre les annotateur-ric-e-s

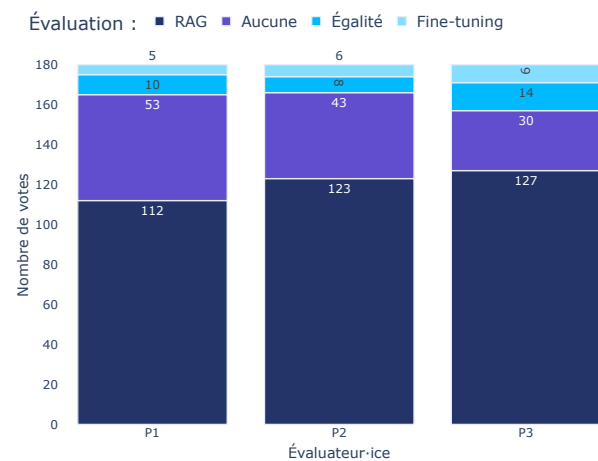


FIGURE 8 – Proportion de votes de préférence de réponse aux 180 questions du jeu de données selon la méthode utilisée pour la génération pour chacun-e des annotateur-ric-e-s.

tion de la pertinence (*answer relevance*) est faite avec un LLM, les résultats obtenus ne sont pas déterministes. Une même réponse peut donc se voir attribuer un score différent d'une génération à une autre. Pour obtenir un score robuste, la pertinence de chaque réponse a été évaluée 10 fois. Le score de pertinence attribué à chaque modèle dans la table 2 est donc la moyenne de la pertinence de chacune des réponses pour les 10 générations. Le calcul de la fidélité (*faithfulness*) est déterministe car la détection des passages d'intérêts avec spaCy est le même d'une exécution à l'autre pour une même phrase. La fidélité est donc calculée une seule fois pour chaque génération. Le score de fidélité présenté pour chaque technique dans la table 2 est la moyenne de la fidélité pour chaque réponse générée.

TABLE 2 – Pertinence et fidélité des réponses pour le fine-tuning et la RAG sur chaque dataset.

Dataset	Technique	Pertinence	Fidélité
Département de la Drôme	Fine-tuning	0.74	0.23
	RAG	0.81	0.56
Télécom St-Etienne	Fine-tuning	0.68	0.37
	RAG	0.79	0.66

Quel que soit le jeu de données, la RAG obtient de meilleurs résultats que le fine-tuning pour la pertinence et pour la fidélité. Les réponses des deux techniques semblent plus pertinentes sur les données du département de la Drôme mais plus fidèles sur celles de Télécom Saint-Etienne. Cela peut s'expliquer par le fait que le jeu de données de la Drôme contient plus d'informations complexes à restituer pour le LLM comme des numéros de téléphone, des adresses e-mail ou des adresses de sites internet. Les fragments de la Drôme contiennent donc plus de passages pertinents à retrouver dans la réponse générée, ce qui semble limiter les scores obtenus pour la fidélité.

Les scores de pertinence sont élevés pour les deux techniques, ce qui suggère que même si la réponse n'est pas exacte, le texte généré est cohérent avec la question posée et tend bien à y répondre.

Les scores de fidélité sont eux plus faibles, ce qui suggère que les informations factuelles transmises sont facilement erronées bien que la formulation de la réponse soit cohérente avec la question posée. Des résultats aussi bas que ceux du fine-tuning sont particulièrement problématiques pour le cas d'usage d'agents conversationnels car la réponse générée n'apporte pas d'informations fiables aux utilisateur.rice.s.

5 Discussion et conclusion

Les jeux de données utilisés dans cette étude ont plusieurs particularités notables. La première est qu'ils contiennent des informations spécifiques non vues par les modèles pré-entraînés couramment utilisés (comme *Mistral*). Ce type de donnée a fait l'objet d'études antérieures comparant RAG et fine-tuning, comme présenté précédemment. La deuxième spécificité n'a quant à elle pas encore été étudiée dans les approches existantes : les documents sont en français. Or,

même si *Mistral* peut générer du contenu dans cette langue, ce modèle est spécialisé pour la langue anglaise [9]. Enfin, les documents des corpus ont des structures particulièrement standardisées (par exemple titre + contenu + section "plus d'information <URL>" pour Télécom Saint-Etienne).

5.1 Injection de connaissances par fine-tuning

En raison des spécificités présentées plus haut, le fine-tuning n'a pas permis d'injecter convenablement les connaissances. En effet, de nombreux faits n'apparaissent qu'une seule fois dans le jeu de données (cf. figure 6), ce qui complique l'intégration de l'information. L'augmentation du nombre d'itérations d'apprentissage (*epochs*) permettrait de pallier à ce problème. Cependant, si le modèle est entraîné trop longtemps, il sur-apprend les données de spécialisation et perd ses capacités à générer du langage cohérent. Il est donc difficile de présenter les faits plusieurs fois au modèle sans lui faire perdre ses capacités de pré-entraînement. Pour résoudre ce problème, une piste intéressante consisterait à générer des paraphrases des documents du corpus spécifique, de manière à augmenter les données en réduisant le sur-apprentissage [12]. Néanmoins, une étude approfondie serait nécessaire : les paraphrases doivent être suffisamment diverses et ne pas altérer les informations essentielles contenues dans les documents. De plus, nous avons observé que la structure standardisée amplifie le phénomène de sur-apprentissage et d'hallucination. Sur le jeu de données Télécom Saint-Etienne, par exemple, le LLM termine quasiment toujours sa génération par "Pour plus d'information : <URL>". Ce phénomène indique clairement que le modèle a sur-appris. Il génère aussi systématiquement un lien, qui souvent semble lié à la réponse mais qui mène rarement à une page existante. Enfin, nous avons également évalué le fine-tuning au moyen de méthodes frugales, notamment la méthode QLora [4]. Cette approche consiste à quantifier le modèle, c'est-à-dire réduire la précision de stockage des paramètres du réseau sur 4 bits et ajouter des *adapters*. Ces couches supplémentaires sont entraînées sur la tâche pendant que les paramètres du réseau pré-entraîné sont gelés. Cette approche permet de réduire de façon drastique le nombre de paramètres entraînés. Néanmoins, nous avons observé une perte en génération : la log vraisemblance négative du contenu est 10 fois plus élevée en moyenne avec cette approche. L'utilisation de méthodes *parameter efficient* devra être plus longuement explorée dans le futur mais ne semble, pour l'instant, pas être une alternative satisfaisante sur ce type d'application.

5.2 Optimisation et enrichissement de la technique de RAG

La technique de RAG utilisée en comparaison au fine-tuning représente l'une des méthodes disponibles pour contextualiser la génération d'une réponse par un LLM. Cependant, elle repose sur de nombreux paramètres dont la modification pourrait modifier les résultats. La première étape de découpage du contenu en fragments est primordiale car elle permet d'obtenir du texte propre et cohérent,

porteur d'une information claire pour l'étape de recherche d'information. Optimiser cette étape permet donc d'améliorer la récupération qui améliore à son tour la génération. Lors de l'étape de recherche d'information, la technique de similarité et le modèle de représentation du langage choisis ont tous les deux une grande influence sur les résultats. Nous pouvons par exemple envisager d'utiliser un modèle de représentation du langage spécialisé sur la tâche de récupération d'informations ou sur les domaines spécifiques de nos données. Actuellement, la récupération extrait 10 fragments de texte qui sont ensuite utilisés comme éléments de contexte. Dans de futurs travaux, nous souhaitons réduire ce nombre de fragments pour conserver uniquement ceux contenant des informations relatives à la question posée et ainsi limiter la consommation de ressources en diminuant de façon drastique la taille du prompt. Des étapes supplémentaires pourront également être ajoutées comme le *re-ranking*, qui consiste à réordonner les fragments identifiés par ordre de pertinence pour être présentés au LLM. Enfin, une optimisation du prompt de génération de réponse est envisagée pour s'adapter au mieux au format attendu (langue, longueur de la réponse, précision, etc.).

5.3 Conclusion

Ces travaux nous ont permis de mettre en lumière les points suivants :

1. Bien que la pertinence des réponses soit proche pour les modèles RAG et fine-tunés, l'approche RAG est largement préférée dans une évaluation de type "Arena".
2. La RAG est peu dégradée d'un dataset à l'autre alors que l'impact sur le fine-tuning est plus important : la qualité des données semble donc avoir un impact plus important sur le fine-tuning.
3. Le fine-tuning est moins efficace que la RAG dans le cas de données spécialisées où les informations sont peu répétées.

Notre analyse possède également quelques limitations. Le modèle Mistral 7B n'est pas spécialisé sur la génération de contenu en français. Nous n'avons évalué qu'un seul modèle de langage, néanmoins, le coût d'entraînement peut s'avérer très élevé, notamment quand le nombre de paramètres du modèle croît fortement. Ces axes de recherche feront l'objet de travaux futurs.

Enfin, les observations résultant de ce travail permettent de donner de nouvelles directions de recherche dans le domaine de l'application des grands modèles de langage sur les données expertes. Nous envisageons d'étudier plus finement l'impact de la taille des jeux de données sur les résultats ou encore de l'approche de fine-tuning (par exemple par la mise en forme de la tâche de continuation comme une tâche d'instruction).

Remerciements

Ces travaux s'inscrivent dans le programme QABot, qui bénéficie du dispositif R&D Booster financé par la Région

Auvergne-Rhône-Alpes et BPIFrance que nous remercions ici.

Références

- [1] Tolgahan Cakaloglu, Christian Szegedy, and Xiaowei Xu. Text embeddings for retrieval from a large knowledge base. In *International Conference on Research Challenges in Information Science*, volume 385 of *Lecture Notes in Business Information Processing*, pages 338–351. Springer, 2020.
- [2] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1) :37–46, 1960.
- [3] Angels de Luis Balaguer, Vinamra Benara, Renato Luiz de Freitas Cunha, Roberto de M. Estevao Filho, Todd Hendry, Daniel Holstein, Jennifer Marsman, Nick Mecklenburg, Sara Malvar, Leonardo Nunes, Rafael Padilha, Morris Sharp, Bruno Leonardo Barros Silva, Swati Sharma, Vijay Aski, and Ranveer Chandra. Rag vs fine-tuning : Pipelines, tradeoffs, and a case study on agriculture. *ArXiv*, abs/2401.08406, 2024.
- [4] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora : Efficient finetuning of quantized LLMs. *Advances in Neural Information Processing Systems*, 36, 2024.
- [5] Jennifer Dodgson, Nanzheng Lin, Julian Peh, Akira Raphael Janson Pattirane, Alfath Daryl Alhajir, Eko Ridho Dinarto, Joseph Lim, and Syed Danyal Ahmad. Establishing performance baselines in fine-tuning, retrieval-augmented generation and soft-prompting for non-specialist llm users. *ArXiv*, abs/2311.05903, 2023.
- [6] Shahul ES, Jithin James, Luis Espinosa Anke, and Steven Schockaert. Ragas : Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024 - System Demonstrations*, pages 150–158, 2024.
- [7] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *Proceedings of the 9th International Conference on Learning Representations, ICLR, 2021*.
- [8] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models : Principles, taxonomy, challenges, and open questions. *ArXiv*, abs/2311.05232, 2023.
- [9] Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Re-

- nard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *ArXiv*, abs/2310.06825, 2023.
- [10] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NeurIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.
- [11] Arindam Mitra, Luciano Del Corro, Shweti Mahajan, Andres Coda, Clarisse Simoes, Sahaj Agrawal, Xuxi Chen, Anastasia Razdaibiedina, Erik Jones, Kriti Agarwal, Hamid Palangi, Guoqing Zheng, Corby Rosset, Hamed Khanpour, and Ahmed Awadallah. Orca 2 : Teaching small language models how to reason. *ArXiv*, abs/2311.11045, 2023.
- [12] Oded Ovadia, Menachem Brief, Moshik Mishaeli, and Oren Elisha. Fine-tuning or retrieval? comparing knowledge injection in LLMs. *ArXiv*, abs/2312.05934, 2023.
- [13] Evani Radiya-Dixit and Xin Wang. How fine can fine-tuning be? learning efficient language models. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020*, volume 108 of *Proceedings of Machine Learning Research*, pages 2435–2443, 2020.
- [14] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero : Memory optimizations toward training trillion parameter models. In *SC20 : International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE, 2020.
- [15] S. E. Robertson and Sparck K. Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3) :129–146, 1976.
- [16] Karen Sparck Jones. *A statistical interpretation of term specificity and its application in retrieval*, page 132–142. Taylor Graham Publishing, GBR, 1988.
- [17] Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar
- Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2 : Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288, 2023.
- [18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30 : Annual Conference on Neural Information Processing Systems*, pages 5998–6008, 2017.
- [19] Anthony J Viera, Joanne M Garrett, et al. Understanding interobserver agreement : the kappa statistic. *Family Medicine*, 37(5) :360–363, 2005.
- [20] Zhongwei Wan, Xin Wang, Che Liu, Samiul Alam, Yu Zheng, Jiachen Liu, Zhongnan Qu, Shen Yan, Yi Zhu, Quanlu Zhang, Mosharaf Chowdhury, and Mi Zhang. Efficient large language models : A survey. *arXiv preprint arXiv :2312.03863*, 2024.
- [21] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35 : Annual Conference on Neural Information Processing Systems, NeurIPS 2022*, 2022.
- [22] Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. Instruction tuning for large language models : A survey. *arXiv preprint arXiv :2308.10792*, 2023.
- [23] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36 : Annual Conference on Neural Information Processing Systems, NeurIPS*, 2023.