RAG Approach Enhanced by Category Classification with BERT

Yuki Taya
NEC Corporation
Tokyo Japan
taya-yuki@nec.com

Daiki Ito
NEC Corporation
Tokyo Japan
ito-daiki@nec.com

Shingo Maeda
NEC Corporation
Tokyo Japan
shingomaeda@nec.com

Yusuke Hamano
NEC Corporation
Tokyo Japan
yusuke-hamano@nec.com

ABSTRACT

We are honored to announce that our team has secured first place in the False-Premise category of Task 3 in the Meta Comprehensive Retrieval-Augmented Generation (CRAG) Challenge, part of the KDD Cup 2024 [1]. This competition addresses the critical issue of hallucination in Large Language Models (LLMs) by leveraging Retrieval-Augmented Generation (RAG) systems [2]. Despite the advancements in LLMs, their accuracy in answering questions about both slow-changing and fast-changing facts remains below 15%, and even for stable facts, the accuracy is below 35% for less popular entities [1]. The CRAG Benchmark evaluates RAG systems across five domains and eight question types, providing a rigorous framework for assessing their performance. The challenge comprises three tasks: Web-Based Retrieval Summarization, Knowledge Graph and Web Augmentation, and End-to-End RAG, each designed to progressively enhance the complexity and capability of RAG systems. Evaluation metrics include both automated and human assessments, with a focus on response quality and conciseness. Participants are required to use Llama models [3] and adhere to specific hardware and resource constraints.

Our approach consists of three major components. First, we classified the attributes of questions using BERT. This allowed us to handle relatively difficult questions by responding with "IDK" (I don't know), successfully navigating through them. Second, we implemented filtering techniques to use the same architecture across all tasks. This enabled us to conduct experiments efficiently across all tasks. Finally, after generating answers with the LLM, we adopted an architecture that refines the responses. This mechanism significantly reduced hallucinations in the LLM's answers. As a result, although our overall ranking across all tasks was not outstanding, we were able to secure first place in the False-Premise category of Task 3.

CCS CONCEPTS

• Artificial intelligence • Machine learning

KEYWORDS

LLM, RAG, BERT

1 Introduction

1.1 Background and Objectives

In recent years, significant advancements in large language models (LLMs) have led to numerous achievements in the field of natural language processing. However, LLMs still face the persistent issue of "hallucination," where the models generate responses that are not based on factual information. Even advanced models like GPT-4 have been reported to have low accuracy in answering fact-based questions. For instance, the accuracy of GPT-4 in answering questions about both slow-changing and fast-changing facts is less than 15%, and for stable facts, the accuracy for less popular entities is below 35% [1].

In this context, Retrieval-Augmented Generation (RAG) has gained attention as a promising approach to mitigate the knowledge deficiencies of LLMs. RAG systems aim to retrieve relevant information from external sources and to generate answers based on this information. However, RAG systems still face several challenges, such as selecting the most relevant information, reducing response latency, and synthesizing information to answer complex questions.

The Meta Comprehensive RAG Challenge (CRAG) aims to provide a robust benchmark for evaluating RAG systems, driving innovation, and advancing solutions in this domain.

Previous effort [1] has also revealed that certain questions are difficult to answer even when referencing external information, depending on the domain or question type, and that LLMs are prone to making mistakes on these questions. Therefore, we focused on developing a low-cost method for identifying difficult questions and implemented a step-by-step approach to answer generation to suppress hallucinations.

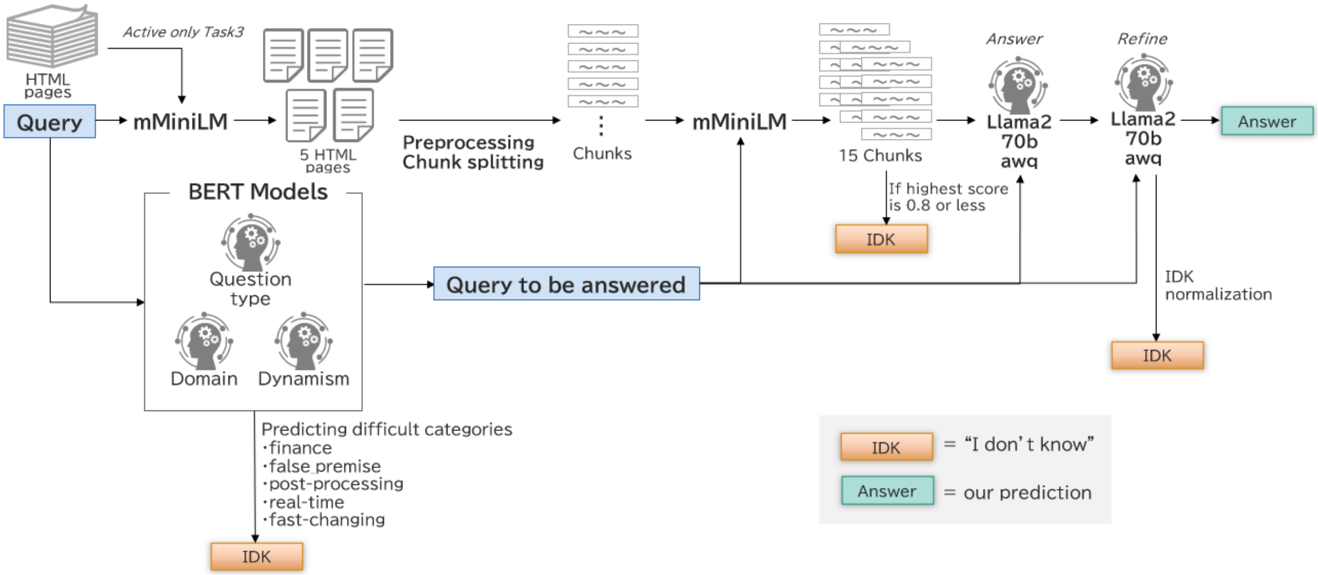

Figure 1: Overview of Our Method

1.2 Problem Statement

The CRAG competition challenges participants to develop RAG systems across three tasks:

1. Task1. Web-Based Retrieval Summarization: Participants receive five web pages per question and are evaluated on their ability to accurately summarize this information into a coherent answer.
2. Task2. Knowledge Graph and Web Augmentation: This task involves using mock API of knowledge graph to retrieve structured data related to the questions and integrating this data to formulate answers.
3. Task3. End-to-End RAG: Participants are provided with 50 web pages and mock API of knowledge graph access per question, and they must select and integrate the most relevant information to generate accurate answers.

The evaluation of RAG systems is based on a scoring method that measures the quality of responses:

- **Perfect**: The response correctly answers the question without any hallucinated or harmful content, earning 1 point.

- **Acceptable**: The response provides useful information but may contain minor errors, earning 0.5 points.

- **Missing**: The response fails to provide the requested information, earning 0 points (e.g., "I don't know," "I couldn't find...").
- **Incorrect**: The response provides wrong or irrelevant information, resulting in -1 point.

These evaluation criteria are used to comprehensively assess the performance of RAG systems.

To emphasize an important point, a correct answer is awarded +1, while an incorrect answer is penalized with -1. Therefore, in this competition, the ratio of correct to incorrect answers is a more critical evaluation metric than simply the number of correct answers.

it is effective to avoid answering questions that have even a slight possibility of being incorrect and instead mark them as IDK (I Don't Know).

2 Our Method

2.1 Overview of our RAG System Architecture

Figure 1 illustrates the overall architecture of our Retrieval-Augmented Generation (RAG) system, which is common to all tasks and does not utilize the mock API for the knowledge graph. We have not adopted an architecture that uses the mock API for the knowledge graph, as we believe that focusing on improving RAG will enhance accuracy more effectively. The architecture is designed to handle various types of queries and search results efficiently.

1. **Query Categorization:** The system begins by receiving a query, which is then processed by a BERT-based model that classifies the query according to various factors such as question type, domain, and dynamism. This classification step is crucial for identifying difficult categories, such as finance, false premises, post-processing, real-time, and fast-changing scenarios. If the query falls into one of these

challenging categories, the system outputs an "IDK" (I Don't Know) response, indicating that the query may not be effectively answered by the system.

2. **Document Filtering for Task3:** In Task 3, the system filters HTML documents using the mMiniLM model (https://huggingface.co/nreimers/mmarco-mMiniLMv2-L12-H384-v1). The mMiniLM model, functioning as a cross-encoder, evaluates and scores the relevance between the query and each page snippet. Based on these scores, the top 5 snippets are selected for further processing. This approach ensures that the system focuses on the most relevant content, optimizing the efficiency and accuracy of the subsequent analysis and answer generation processes.
3. **Preprocessing**: This step involves preparing the HTML documents for further analysis. The preprocessing may include tasks like cleaning the text, removing unnecessary HTML tags, and normalizing the content. Additionally, non-English content and other irrelevant data are removed to ensure that only pertinent information is retained. On web pages, parsing errors can sometimes result in the loss of these spaces, so any word longer than 30 characters was excluded.
4. **Chunk Splitting**: After preprocessing, the filtered documents are split into smaller, manageable pieces called "chunks." Specifically, the text is divided into chunks of 150 words each, with each chunk having a 25-word overlap with the preceding one. This overlap helps maintain context between chunks. If the chunking process resulted in more than 2000 chunks, a further filtering step is applied using BM25. This is done to reduce the number of chunks to 2000, optimizing the processing time and making the subsequent analysis more efficient.
5. **Reranking**: After the documents are split into chunks, mMiniLM is used to rank the chunks based on their relevance to the query. Specifically, the top 15 most relevant chunks are selected for further processing. During this ranking process, if the relevance score (referred to as the TOP1 score) for the highest-ranked chunk is 0.8 or less, the system defaults to an "I don't know" (IDK) response, indicating a lack of confidence in providing an answer.
6. **Answer Generation**: For queries with a Top1 score above 0.8, the system advances to the next step, where the chunks are passed to a llama2-70b-awq. This model generates an initial answer based on the evaluated chunks.
7. **Refinement:** The initial answer generated by the first Llama2 model is then refined by a second pass through llama2-70b-awq. After the refinement process, the system provides the final answer to the user unless the system determines an "IDK" outcome after normalization. This refinement step is crucial for improving the accuracy and reliability of the final output.

This architecture is designed to maximize the accuracy and efficiency of the RAG system while minimizing the computational resources required.

2.2 BERT Models

For the task of categorizing text, we employed the BERT-base-uncased model. We developed three separate models to classify the categories of domain, question type, and dynamism. The models were trained on validation data and subsequently evaluated on test data.

The training parameters were set as follows: the learning rate was fixed at 2e-5, and a weight decay of 0.01 was applied. The final model for the domain category was selected after the 8th epoch, while the models for question type and dynamism were selected after the 18th epoch.

2.3 RAG Hyperparameters

We have been focusing on the hyperparameters of RAG since we started working on the competition. While strong models like GPT-4 tend to improve accuracy by adding more information to the model's input context, open-source LLMs, including Llama2, tend to increase hallucinations when more context is included. Therefore, we conducted comprehensive experiments on chunk size and chunk number.

2.4 Generation and Refinement

Our team anticipated that quantizing a model with more parameters would yield higher accuracy than a 7B LLM. Therefore, from the outset, we adopted TheBloke/Llama-2-70B-AWQ. (https://huggingface.co/TheBloke/Llama-2-70B-AWQ)

In our implementation, we utilized the vLLM inference engine, which is faster than other transformer-based inference engines. The parameters used in our system configuration are as follows:

Model Initialization:
Max Model Length: 4096 tokens
Enforce Eager: True
GPU Memory Utilization: 80%

Sampling Parameters:
Temperature: 0.0
Top-p (Nucleus Sampling): 0.95
Max Tokens: 50

We are using an environment with four Nvidia T4 GPUs, as specified by the KDD Cup competition.
By leveraging these parameters and the vLLM inference engine, we achieved efficient and effective model performance.

Our Generate Prompt and Refine Prompt are as follows:
GENERATE. prompt = """You are given a quesition and references which may or may not help answer the question.
You are to respond with just the answer and no surrounding sentences.
If you are unsure about the answer, respond with "I don't know".
Question
{query}

References
{references}

Question
{query}

Answer"""

Table 1: Number of data entries

	Number of samples
Validation data	1371
Test data	1335

Table 2: Definition of question types.

question_type	Definition
Simple	Questions asking for simple facts that are unlikely to change overtime, such as the birth date of a person and the authors of a book.
Simple w. Condition	Questions asking for simple facts with some given conditions, such as stock prices on a certain date and a director's recent movies in a certain genre.
Set	Questions that expect a set of entities or objects as the answer (e.g., "what are the continents in the southern hemisphere?").
Comparison	Questions that compare two entities (e.g., "who started performing earlier, Adele or Ed Sheeran?").
Aggregation	Questions that require aggregation of retrieval results to answer (e.g., "how many Oscar awards did Meryl Streep win?").
Multi-hop	Questions that require chaining multiple pieces of information to compose the answer (e.g., "who acted in Ang Lee's latest movie?").
Post-processing heavy	Questions that need reasoning or processing of the retrieved information to obtain the answer (e.g., "how many days did Thurgood Marshall serve as a Supreme Court justice?"). One of difficult question type.
False Premise	Questions that have a false preposition or assumption (e.g., "What's the name of Taylor Swift's rap album before she transitioned to pop?" (Taylor Swift has not yet released any rap album)). One of difficult question type.

Table 3: Number of samples by question type

question_type	Validation data	Test data
Simple	395	359
Simple w. Condition	201	206
Set	124	125
Comparison	170	163
Aggregation	154	161
Multi-hop	107	124
Post-processing heavy	64	44
False Premise	156	153

REFINE. prompt= """# Instructions
You are a professional in document comprehension.
Read the document and if the question cannot be answered from the document, respond succinctly with 'I don't know' only.
If the answer to the question is correct, respond succinctly with 'Correct' only.

If the answer is incorrect, respond succinctly with the correct answer only. Do not output unnecessary sentences such as "the given answer is incorrect," just answer the question.

# Document
{references}
# Question
{question}
# Answer given by someone who is not good at document comprehension
{answer}
# Your judgment (Correct/I don't know) or appropriate answer
"""

3 Experiments and Results

3.1 CRAG Data

In this study, we use validation data and test data. Each dataset contains 1371 and 1335 entries, respectively. (Table 1)

The data used is categorized into 8 question types and covers 5 domains: finance, sports, music, movies, and the open domain of encyclopaedias. These domains represent a range of information change rates, classified into rapid (finance and sports), gradual (music and movies), and stable (open domain).

The question types are categorized into eight groups: Simple, Simple w. Condition, Set, Comparison, Aggregation, Multi-hop, Post-processing heavy, and False Premise. These include types that require calculations and those that contain incorrect information. Details of each question type can be found in

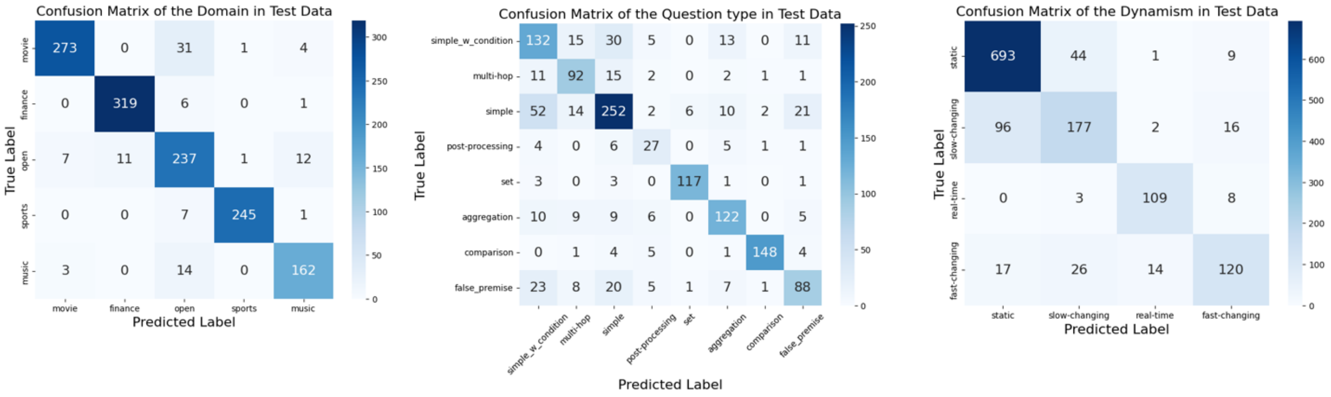

Figure 2: Confusion Matrices of the Test Data: Domain (left), Question Type (center), and Dynamism (right).

Table 4: The results of RAG inference on the entire test data.

		Ratio		
score	exact accuracy	accuracy	hallucination	missing
0.087	0.038	0.177	0.090	0.733
		Count		
total	exact correct	correct	hallucination	missing
1335	51	236	120	979

Table 2. Additionally, the number of each question type included in the validation and test data is shown in Table 3.

3.2 Accuracy of the RAG system on the overall test data

Table 4 shows the accuracy of our RAG system.

Our RAG system avoids unnecessary deductions by marking many responses as 'missing.' Additionally, the number of correct responses is higher than the number of hallucinations, which leads to better results.

3.3 BERT Models

The evaluation of the BERT-based models across domain, question type, and dynamism categories provided insights into the strengths and weaknesses of the models, as illustrated by the confusion matrices in Figure 2.

3.3.1 Accuracy of BERT Model Classification

Domain classification. The domain classification model exhibited high accuracy, particularly in the 'finance' and 'sports' categories, with minimal misclassifications. However, there were some confusions between the 'music' and 'movie' categories, indicating that the model may occasionally struggle with distinguishing between these closely related domains.

Table 5: Effect of BERT on Question Type Performance.

question type	w/o BERT	w/ BERT	effect of BERT
aggregation	0.075	0.068	-0.006
comparison	0.160	0.141	-0.018
false premise	-0.144	-0.059	+0.085
multi-hop	0.185	0.185	0
post-processing	-0.045	0.068	+0.114
set	0.088	0.064	-0.024
simple	0.111	0.106	-0.006
simple w condition	0.083	0.097	+0.015

Question type Classification. In the question type classification task, the model generally performed well across most categories but faced significant challenges with the 'false premise' category. The confusion matrix shows that 'false premise' questions were often misclassified as 'simple' or 'simple w condition' questions. This misclassification suggests that the model struggles to detect the underlying incorrect assumptions or logical inconsistencies that characterize 'false premise' questions.

Instead, it appears to treat these questions as straightforward or comparative, indicating a need for more sophisticated reasoning capabilities to accurately identify the nuanced errors present in 'false premise' questions. This misclassification of the false premise leads our results in a positive direction.

Dynamism Classification. The dynamism classification model demonstrated strong performance in identifying 'static' and 'real-time' data, correctly classifying many instances in these categories.

Table 6: Effect of BERT on Domain Performance.

domain	w/o BERT	w/ BERT	effect of BERT
finance	-0.058	-0.003	+0.055
movie	0.071	0.065	-0.006
music	0.140	0.145	+0.006
open	0.213	0.179	-0.034
sports	0.079	0.095	+0.016

Table 7: Effect of BERT on Dynamism Performance.

static or dynamic	w/o BERT	w/ BERT	effect of BERT
fast changing	-0.062	-0.339	-0.108
real time	-0.075	0	-0.160
slow changing	0.065	0.079	+0.223
static	0.142	0.134	-0.052

However, the model exhibited some confusion between 'slow-changing' and 'fast-changing' data. Specifically, 'slow-changing' data was often misclassified as 'static,' and 'fast-changing' data was frequently confused with both 'slow-changing' and 'real-time' categories.

3.3.2 Effect of BERT on CRAG Score

Based on the evaluation metrics of this CRAG, answering IDK (=0) rather than making a mistake (=-1) improves the overall score. Therefore, we analyzed the high-difficulty categories where it is better to answer IDK. The results are shown in the tables 5, 6, and 7 under the "w/o BERT" column. Among these, the categories with negative scores—false premise, post-processing, finance, fast-changing, and real-time—often result in incorrect answers. For these categories, we used BERT to detect them and respond with IDK.

Table 5 compares the scores when queries classified as false premise and post-processing are identified using BERT and answered with IDK. For false premise and post-processing, we observed a significant improvement. Although there is a slight decrease in scores for other question types, the overall contribution is positive.

Table 6 compares the scores when queries in the finance domain are classified using BERT and answered with IDK. The finance category showed a significant improvement. Although the accuracy for the open category slightly decreased, the increase in the finance category is much larger.

Table 7 shows the results for the dynamic category. Contrary to expectations, the scores for fast-changing and real-time did not improve, while the score for slow-changing did improve. This is likely due to the influence of the classification results from other question types and domains.

Table 8: Relationship Between Chunk Size, Chunk Number, and CRAG Score for Llama2-70B-AWQ

words per chunk	top N chunks	score	accuracy	hallucination
30	**15**	0.16	0.35	0.19
50	**10**	0.13	0.33	0.2
50	**15**	0.16	0.36	0.2
200	**5**	0.14	0.34	0.2
200	**10**	0.13	0.36	0.23

Table 9: Effect of Refine Prompt Performance

number of test data=400	w/o Refine	w/ Refine	effect of Refine
score	-0.070	0.098	+0.168
accuracy	0.303	0.205	-0.098
hallucination	0.373	0.108	-0.265

3.4 RAG Hyperparameters

Table 8 illustrates the relationship between chunk size, chunk number, and accuracy for Llama2-70B-AWQ. The best score was achieved with 30 words per chunk and 15 chunks. In these experiments, while accuracy remained almost unchanged, there was a tendency for hallucinations to increase as the number of words per chunk increased. This indicates that simply increasing the chunk size is not necessarily beneficial.

3.5 Generation and Refinement

We had the LLM respond using relatively standard prompts. When using an LLM of this parameter size, we focused on creating simple prompts rather than complex instructions.

Furthermore, after generating the responses, we had the same LLM refine them. Typically, such mechanisms do not significantly contribute to accuracy improvement, but in this task, reducing hallucinations is very important. As shown in Table 9, the refine prompts significantly reduced hallucinations and contributed to an improvement in the score.

3.6 “false premise” Prediction Result

Questions with false premises are correctly classified with an accuracy of over 50%. However, there are cases where questions with false premises, such as ‘which political office does Ben Affleck currently hold?’ or ‘which team went head-to-head with the Denver Nuggets on 2023-01-25?’, are predicted as simple_w_condition.

We will highlight a characteristic case of these incorrect predictions. The question “what type of dog does Taylor Swift have?” was predicted as simple despite being a false premise question. In reality, Taylor Swift has been a dog owner since 2018, which suggests that the question type prediction using BERT is correct. Therefore, while BERT's predictions for false premises are generally accurate, it is likely that our accuracy in the false premise category was high because some questions could be answered rather than being false premises.

4 Conclusion

Our approach to the CRAG competition involved three key components. First, we used BERT to classify question attributes, allowing us to handle difficult questions by responding with "IDK" (I don't know). Second, we implemented filtering techniques to maintain a consistent architecture across all tasks, enabling efficient experimentation. Finally, we refined the LLM-generated answers to significantly reduce hallucinations.

While our overall ranking was not exceptional, these strategies led us to secure first place in the False-Premise category of Task 3. This success demonstrates the effectiveness of our approach in managing specific question types and suggests potential for further improvements.

In the future, we aim to explore the utilization of knowledge graphs and work on improving accuracy and inference efficiency when leveraging knowledge graphs.

REFERENCES

[1] Yang, X., Sun, K., Xin, H., Sun, Y., Bhalla, N., Chen, X., Choudhary, S., Gui, R. D., Jiang, Z. W., Jiang, Z., Kong, L., Moran, B., Wang, J., Xu, Y. E., Yan, A., Yang, C., Yuan, E., Zha, H., Tang, N., Chen, L., Scheffer, N., Liu, Y., Shah, N., Wanga, R., Kumar, A., Yih, W., & Dong, X. L. CRAG - Comprehensive RAG Benchmark. 2023

[2] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, Douwe Kiela. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. 2020

[3] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.