Retrieval-Augmented Generation for Large Language Models on Understanding and Reasoning

CPSC 532V 2023W2 Project Report

Juntai Cao[♠] Student #: 50171404 jtcao7@cs.ubc.ca

Yilin Yang[♠] Student #: 24754350 yangyl17@cs.ubc.ca

Yuwei Yin[♠] Student #: 36211928 yuweiyin@cs.ubc.ca

Department of Computer Science, University of British Columbia
2366 Main Mall, Vancouver, BC V6T 1Z4, Canada

Abstract

Large language models (LLMs) have been a transformative technique reshaping the natural language generation (NLG) field. Retrieval-augmented generation (RAG) is proposed to supplement the parametric knowledge in LLMs with external factual knowledge and achieve promising results on knowledge-intensive tasks like open-domain question answering. However, the effectiveness of RAG on natural language understanding (NLU) and inference (NLI) tasks lacks exploration. In this work, we comprehensively review various RAG methods and systematically implement the RAG framework. Then, we conduct extensive experiments to evaluate different RAG components and variants on multiple natural language understanding and reasoning benchmarks. The experimental results demonstrate that RAG methods are not always helpful to reasoning-intensive problems, which brings insights into the feasibility of RAG methods on such tasks. The findings and discussions shed light on future RAG research, especially for improving the reasoning ability of LLMs. Our code is available.[1]

1 Introduction

Natural language generation has improved considerably with the rapid development of large language models (LLMs) (Touvron et al., 2023b; OpenAI, 2023; Zhao et al., 2023b). Although LLMs achieve state-of-the-art results on many NLP tasks, their performance lags behind task-specific architectures on knowledge-intensive tasks (Lewis et al., 2020). LLMs store factual knowledge in their parameters, known as "parametric knowledge", during training on large corpora. Retrieval-augmented generation (RAG) was proposed to combine pre-trained parametric and non-parametric memory (from external sources) for language generation.

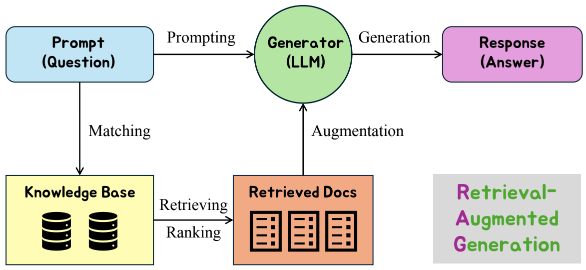

Figure 1: The overview of retrieval-augmented generation (RAG) for large language models (LLMs).

In addition, RAG is a promising approach to alleviate the hallucination problem (Ji et al., 2023), where LLMs make up plausible but irrelevant or factually incorrect answers, by providing LLMs with more knowledge context before generation.

Extensive research works show RAG improves LLMs on various knowledge-intensive tasks (Gao et al., 2023), e.g., open-domain QA (Rajpurkar et al., 2016, 2018; Yang et al., 2018). However, only a few of them focus on reasoning tasks (Levesque et al., 2012; Sakaguchi et al., 2021). Regarding reasoning, we focus on commonsense reasoning (Sap et al., 2020) instead of mathematical/formal reasoning. Although LLMs (Radford et al., 2018; Zhao et al., 2023b) are mainly used and evaluated for natural language generation (NLG) or code generation tasks (Chen et al., 2021; Zheng et al., 2023; Austin et al., 2021), they can also be applied in natural language understanding (NLU) (Winograd, 1972; Wang et al., 2019b,a) and inference (NLI) tasks (Bowman et al., 2015; Nie et al., 2020; Bhagavatula et al., 2020), where reasoning ability plays a vital role in solving these problems. However, to the best of our knowledge, no previous research applies RAG to enhance LLMs on these tasks. Hence, the research question of this project raises: ***Can external knowledge augmented by RAG improve the reasoning ability of LLMs, especially on NLU tasks?***

In this work, we comprehensively review dif-

[♠] Authors contributed equally and listed alphabetically.
[1] https://github.com/YuweiYin/UBC_CPSC_532V

ferent RAG methods, systematically implement the entire framework (including dataset processing, LLM generation, and the RAG procedure), and conduct extensive experiments to examine the performance of various RAG methods on multiple natural language understanding and reasoning benchmarks. Specifically, we implement the RAG system in the following steps: (1) Obtain the queries; (1.5) Query pre-processing, e.g.,rewriting (Ma et al., 2023); (2) Keywords extraction; (3) Search for relevant documents from multiple knowledge sources; (3.5) Document post-processing, e.g., ranking, filtering, and summarization; (4) Query augmentation; (5) LLM generation; (6) Result evaluation. Each of these components can be implemented with different designs, where Step 1.5 and 3.5 are optional.

After building the whole RAG pipeline, we design a series of experiments to test the performance of each RAG component on a wide variety of natural language understanding and reasoning tasks, including WSC (Levesque et al., 2012), WinoGrande (Sakaguchi et al., 2021), ANLI (Nie et al., 2020), ARC (Clark et al., 2018), PIQA (Bisk et al., 2020), SWAG (Zellers et al., 2018), HellaSwag (Zellers et al., 2019b), GLUE (Wang et al., 2019b), and SuperGLUE (Wang et al., 2019a). These benchmarks are all formed as classification tasks, so we adopt accuracy and f1 score for evaluation. Specifically, we conduct the following experiments: (1) **LLM Baselines**: Test the task-solving performance of different LLMs without RAG methods; (2) **RAG Knowledge Sources**: Test the effectiveness of RAG method with different knowledge sources; (3) **Pre- and Post-processing**: Apply multiple pre-processing and post-processing approaches; (4) **Augmentation Methods**: Employ different augmentation prompts, prompting strategies, and fine-tuning methods to combine the original query with retrieved documents.

Experimental results and analysis deepen the understanding of the RAG approach for LLMs on reasoning-intensive NLU tasks. Particularly, we find that RAG does not always help in such tasks, possibly because solving these tasks relies more on reasoning within the given context than external factual knowledge. Our further discussions on the future directions bring insights to the follow-up RAG research. The contributions of this project are summarized as follows:

- We comprehensively review different retrieval-augmented generation methods for large language models, and then systematically implement the RAG framework. The code is made publicly available.
- We conduct extensive experiments to evaluate the effectiveness of different RAG components and various advanced RAG methods on multiple natural language understanding and reasoning benchmarks.
- The experimental results and corresponding analysis bring insights into the effectiveness and feasibility of RAG methods on reasoning-intensive tasks instead of knowledge-intensive ones. The findings and discussions shed light on future RAG research.

2 Background: RAG for LLMs

In this section, we introduce the research background by comprehensively reviewing various RAG methods for LLMs.

2.1 Large Language Models

The development of large language models (LLMs) (Zhao et al., 2023b) like GPT (OpenAI, 2023) and LLaMA (Touvron et al., 2023b) has significantly advanced natural language generation (NLG). After training on massive corpora and tuning in an instruction-following way (Ouyang et al., 2022; Bai et al., 2022), LLMs can generate fluent and coherent responses in a human-like fashion (OpenAI, 2022). However, the generation process suffers from the hallucination problem (Ji et al., 2023) because LLMs tend to make up plausible answers regardless of whether they understand the question and context.

Moreover, LLMs frequently encounter challenges in producing satisfactory answers when confronted with tasks that demand commonsense reasoning (Sap et al., 2020), which makes the hallucination problem especially severe. This is rooted in the language modeling training paradigm (Vaswani et al., 2017; Radford et al., 2018), in which LLM models predict the next token based on the previously generated ones. Thus, the models are supposed to produce better output if they are conditioned on more relevant context for solving the question. Retrieval-augmented generation is a promising approach to alleviate the hallucination problem by regulating the generation with retrieved factual knowledge.

2.2 Retrieval-Augmented Generation

As illustrated in Figure 1, the traditional RAG system is constructed through the following steps: (1) **Building KB**: to build the knowledge base (KB) from encyclopedic and commonsense knowledge sources; (2) **Indexing**: to develop the document indexing module for handling information in the KB; (3) **Retrieval**: to develop the information extraction module for searching, matching, and ranking the most relevant documents; (4) **Augmentation**: to combine the extracted knowledge with the initial context, question, and options to form the final prompt; (5) **Generation**: to feed LLMs with the final prompt to generate answers; (6) **Advanced RAG**: to incorporate other advanced RAG methods as auxiliary modules. In this work, our implementation has some major differences from the above practice, as elaborated on in § 3.

2.3 Advanced RAG Methods

There has been a wide range of retrieval-augmented generation (RAG) research (Gao et al., 2023; Mialon et al., 2023) proposed in recent years. Karpukhin et al. (2020) propose to use dense representations from a dual-encoder framework to replace traditional sparse vector methods, such as TF-IDF or BM25 (Robertson et al., 2009), for open-domain QA tasks. Ma et al. (2023) propose to ask the model to rewrite the original query and then perform web searching to obtain the relevant documents. He et al. (2024) apply RAG for textual graph understanding and question answering using LLMs and graph neural networks (GNNs).

Multi-modal RAG. Zhao et al. (2023a) examine research involving the augmentation of generative models through the retrieval of multi-modal information, including image, code, structured knowledge, speech, and video. Hu et al. (2023) propose to augment a visual-language model by enabling it to retrieve multiple knowledge entries from diverse sources, thus aiding generation.

Multi-source RAG. Yu (2022) highlights the limitations associated with relying solely on single-source homogeneous knowledge, such as Wikipedia, and offers various solutions for implementing RAG using heterogeneous knowledge sources. Wang et al. (2024) propose a unified multi-source RAG method including three sub-tasks, i.e., knowledge source selection, knowledge retrieval, and response generation, with a self-refinement mechanism for iteratively refining the generated response.

RAG + GAR. Shao et al. (2023) introduce a framework that combines Retrieval-Augmented Generation and Generation-Augmented Retrieval (GAR) (Mao et al., 2021), which utilizes the model output from the previous iteration as the context to enhance the RAG process. Similarly, Feng et al. (2023) propose to iteratively use language models to refine the documents retrieved in the RAG step.

Robust RAG. RAG can harm performance when irrelevant retrieval is used. Recent research proposes methods to improve the robustness of generation (Yoran et al., 2023; Yan et al., 2024). Wang et al. (2023) propose to filter out irrelevant retrievals by training a context filtering model with different measures. (Berchansky et al., 2023) propose to eliminate non-essential retrieved information at the token level to streamline the answer generation process. In addition, RAG exhibits certain limitations, such as the attribution-fluency tradeoff (Aksitov et al., 2023), wherein the quality of output may be influenced by the constraints introduced by the retrieved knowledge.

LLMs as Knowledge Source. While the majority of RAG methods retrieve information from external knowledge bases, recent research suggests utilizing LLMs to generate documents or processing the retrieved ones. Petroni et al. (2019) systematically analyze the factual and commonsense knowledge present in publicly available pretrained language models.

2.4 Commonsense RAG

Commonsense knowledge constitutes a fundamental aspect of artificial intelligence (Gunning, 2018; Razniewski et al., 2021) and commonsense reasoning is a significant task in natural language processing (NLP) (Sap et al., 2020). Bosselut et al. (2019) propose the COMET model combining the power of the Transformer model (Vaswani et al., 2017) and commonsense knowledge graphs Atomic (Hwang et al., 2021) and ConceptNet (Speer et al., 2017). Lal et al. (2022) propose to use COMET as the commonsense knowledge source to augment different LLMs for answering why-questions.

Liu et al. (2022) propose to generate knowledge from a language model, and then perform RAG to answer questions. Li et al. (2021) propose a BERT-

Source	URL / API
Wikipedia	Wiki API
ConceptNet	ConceptNet API
arXiv	arXiv API
Google Search	Search API
Large Language Models	Google Gemini

Table 1: The knowledge sources for retrieval.

based filter model to filter low-quality candidates and implement contrastive learning (Chen et al., 2020) in both the encoder and decoder. Yu et al. (2022) propose a unified framework of retrieval-augmented commonsense reasoning, a commonsense corpus with over 20 million documents, and strategies for training a commonsense retriever. Ghosal et al. (2023) propose to train a sequence-to-sequence next-step prediction model by incorporating external commonsense knowledge and employing search techniques to generate intermediate steps for natural language inference (NLI) tasks. Seo et al. (2022) propose to retrieve scene knowledge to enhance compositional generalization and relational knowledge to improve commonsense reasoning. Cui et al. (2024) propose a multi-modal retrieval augmentation framework leveraging both text and images to enhance the commonsense capabilities of language models.

3 Implementation

In this section, we introduce the implementation steps of the project. Our project aims to develop an RAG system to examine and compare different RAG methods of integrating external knowledge into LLMs for solving NLU tasks emphasizing commonsense reasoning.

3.1 Step 1: Obtain the Queries

First, we implement the dataset processing module to obtain the queries from the original information in the datasets, where different tasks have different input-output formats. For Question-answering tasks (multi-choice QA), we use the questions as queries for LLMs. For NLI tasks, where we need to predict the sentence-level relations (i.e., entailment, contradiction, or neutral) based on the given premise and a hypothesis, we use the premises as queries. In most cases, we do not use the provided context for retrieval as it is usually too lengthy.

We do not adopt the traditional RAG method that downloads large dumps of knowledge bases like Wikipedia and then uses an encoder model to obtain embeddings of knowledge trunks for semantic matching. Instead, we leverage the off-the-shelf searching API's integrated matching and ranking abilities. The advantage is that searching APIs are powerful and the pipeline is relatively easy to implement. However, the API calling phase may result in a longer latency, which we discuss in the experiment section.

3.2 Step 1.5: Query Pre-processing (Optional)

As the queries are not always suitable for knowledge searching, we adopt the Query Pre-processing module to tailor the original queries to the searching API. Specifically, we employ LLMs, such as Google Gemini[2], OpenAI GPT [3], and Anthropic Claude[4] as the agent to perform pre-processing by feeding them with specified prompts. In practice, we only use Gemini as it is free. We designed six different pre-processing methods including, keyword extraction, contextual clarification, relevance filtering, query expansion, information structuring, and intent clarification. The prompt templates are shown in Appendix § C.1.

3.3 Step 2: Keywords Extraction

After obtaining the queries, we propose to extract keywords from them. These keywords are especially suitable to searching APIs like ConceptNet and Wikipedia. KeyBERT [5] is used to perform extraction. The extracted keywords are a list of strings, each of them has one word or two words, without duplication. Alternatively, we can adopt the `keyword_extraction` method in the pre-processing stage.

3.4 Step 3: Documents Retrieval

For retrieving documents from multiple knowledge sources, we utilize the searching APIs (Python interface) in Table 1, including Wikipedia, ConceptNet, arXiv, Google Search, and LLMs (Gemini). The retriever API is responsible for searching, semantic matching, and result ranking. The related documents are retrieved from the KB and sorted based on their relevance to the query. Specifically, we obtain Wikipedia page (concept) summaries from the Wikipedia API, descriptions of concept nodes and links from the ConceptNet API, relevant

[2] https://gemini.google.com/app
[3] https://chat.openai.com/
[4] https://www.anthropic.com/claude
[5] https://maartengr.github.io/KeyBERT/

paper Abstract from the arXiv API, first-page results (summaries) from the Google Search API, and LLM outputs from the LLM API. For efficiency, we limit the number of retrievals from each knowledge source to 10.

3.5 Step 3.5: Docs Post-processing (Optional)

As the raw retrievals can be lengthy, messy, or conflict with each other, we adopt the Documents Post-processing module to refine the retrieved documents. Similar to the pre-processing practice, we prompt LLMs to conduct post-processing using various prompt templates, as shown in Appendix § C.2. We devise six post-processing approaches: ranking documents, summarizing documents, extracting key information, refining documents, evaluating documents, and identifying conflict.

3.6 Step 4: Query Augmentation

After retrieval, we combine the extracted documents with the original context, question, and options to construct the final prompt as the input of LLMs. The implementation of augmentation is flexible. We experiment using LLM agents with different augmentation prompts (i.e., short, medium, and long prompt instructions), similar to the approach in the pre- and post-processing modules.

3.7 Step 5: LLM Generation

In the LLM generation phase, we try different prompting methods, including zero-shot generation (default), in-context learning (ICL) (Brown et al., 2020; Dong et al., 2023) (providing QA examples), chain-of-thought prompting (CoT) (Wei et al., 2022; Kojima et al., 2022) (ICL with reasoning). In addition to pure prompting, we also implement the training of language models and experiment supervised fine-tuning (SFT) with instruction tuning (Wei et al., 2021; Sanh et al., 2021; Longpre et al., 2023; Zhang et al., 2023; Jiang et al., 2024). The training and generation details are in Appendix C.4.

3.8 Step 6: Result Evaluation

For evaluation, we adopt the language model evaluation scripts [6] on selected NLU benchmarks. Instead of directly inputting the query to LLMs and expecting the LLMs will generate correct answer, we use a perplexity method to evaluate the model performance on these multi-choice tasks. Suppose the question Q has two choices: C_1 and C_2. We first concatenate Q with each choice and obtain instances I_1 and I_2, where $I_1 = Q; C_1$ and $I_2 = Q; C_2$. Then, we feed LLMs with I_1 and I_2 separately and obtain model logits representing the language model's confidence in the input text I_1 and I_2. At last, we pick the choice with the highest LLM confidence (lowest perplexity).

[6] `https://github.com/EleutherAI/lm-evaluation-harness`

4 Experimental Setup

In this section, we introduce the experimental setup of the project in detail.

4.1 Tasks and Datasets

We use a series of natural language understanding and reasoning tasks and datasets for evaluation, including the benchmarks in Table 13. All of them are classification tasks, where the training and validation set have labels, while the test set does not. We adopt the evaluation method described in § 3.8. For the evaluation metrics, we use accuracy for balanced test sets and F1 score for unbalanced ones. We briefly introduce each task as follows.

Reasoning Tasks. **WSC** (Levesque et al., 2012) requires the model to pick an option that the pronoun in the text refers to. **WinoGrande** (Sakaguchi et al., 2021) requires the model to choose either option1 or option2 to replace the `"_"` symbol in the sentence. **ANLI** (Nie et al., 2020) requires the model to predict the sentence-level relations (entailment, contradiction, or neutral) based on the given the premise and hypothesis. It has three subsets: "r1", "r2", and "r3". **ARC** (Clark et al., 2018) requires the model to pick a choice to answer the question. ARC includes **ARC_Easy** and **ARC_Challenge** subsets. **PIQA** (Bisk et al., 2020) is also a QA task involving physical interactions. **SWAG** (Zellers et al., 2018) and **HellaSwag** (Zellers et al., 2019b) requires the model to complete the sentence by picking a choice from a `ending` list.

Understanding Tasks. **GLUE** (Wang et al., 2019b) [7] and **SuperGLUE** (Wang et al., 2019a) [8] are well-known natural language understanding benchmarks. GLUE includes `rte`, `qnli`, `mnli`, `mnli_mismatch` `mrpc`, `qqp`, `wnli`, `sst2` sub-tasks.

[7] `https://gluebenchmark.com/`
[8] `https://super.gluebenchmark.com/`

SuperGLUE has `cb`, `wic`, `sglue_rte`, `boolq`, `copa`, `multirc`, `record`, and `wsc` sub-tasks.

4.2 Baseline LLMs

We use LLMs of different sizes and series as the backbone models, including GPT (Radford et al., 2018, 2019), GPT-NEO (Black et al., 2022), OPT (Zhang et al., 2022), OLMo (Groeneveld et al., 2024; Soldaini et al., 2024), LLaMA (Touvron et al., 2023a,b), and Mistral (Jiang et al., 2023).

4.3 Experiment 1: LLM Baselines

In this part, our objective is to evaluate the performance of LLMs of varying sizes on selected benchmarks under zero-shot conditions excluding the use of RAG. We have categorized the LLMs into four types based on sizes: smaller than 300M, between 300M and 1B, between 1B and 3B, and greater than 3B. We will identify the most suitable models to be used as backbone for future experiments based on the evaluation results.

4.4 Experiment 2: RAG Knowledge Sources

We assess five knowledge sources, Wikipedia, Google search, LLM (Gemini), ConceptNet, and arXiv Abstract to determine their utility in augmenting the selected models. The results enable us to find the most effective knowledge sources for enhancing commonsense understanding. These identified sources will be incorporated into our subsequent experiments via RAG to enhance the LLMs' understanding and reasoning capabilities.

4.5 Experiment 3: Pre- and Post-processing

we evaluate a variety of pre-processing and post-processing techniques to refine the input and output of LLMs within our chosen RAG framework. The pre-processing approaches includes query rewriting, keyword extraction, and contextual clarification, etc. For post-processing, we implement methods such as ranking, filtering, and summarizing, etc. Based on the outcomes of these experiments, we will select the most effective pre- and post-processing methods to employ in our pipeline.

4.6 Experiment 4: Augmentation Methods

We investigate a range of augmentation techniques within our selected RAG framework. We evaluate various prompt templates alongside two distinct prompting methods: zero-shot generation and in-context learning (ICL). Our goal is to identify the most effective augmentation techniques to develop a robust and efficient RAG system.

5 Results and Analysis

In this section, we report the experimental results and corresponding analysis.

5.1 Experiment 1: LLM Baselines

Table 2 shows the experimental results of various baseline LLMs of different model sizes on GLUE (Wang et al., 2019b), SuperGLUE (Wang et al., 2019a), and other selected commonsense reasoning datasets. The full results are shown in Table 31, Table 32, and Table 33 at Appendix C.

The result indicates that among the models smaller than 300M, GPT-1 has the best average scores, notably achieving better results than its successor GPT-2-Small, despite the latter having more parameters. Among models between 300M and 1B, GPT-2-Large outperforms other models, which can be attributed to its larger size. Yet despite having twice as many parameters as GPT-2-Medium, GPT-2-Large only achieves an average score improvement of 0.01. For models between 1B and 3B, OpenLLaMA-3B achieves the highest average score. However, OLMo-1B, the model with the fewest parameters in this range, demonstrates performance comparable to OpenLLaMA-3B across most benchmarks except for WSC273. This indicates that using OLMo-1B as a baseline model achieves greater efficiency without sacrificing performance. For large models with over 5B parameters, Mistral-7B consistently outperforms all competitors across all benchmarks by a large margin.

Therefore, we choose Mistral-7B and OLMo-1B for further experiments as representatives for models above and below 3 billion parameters, respectively, due to their superior performance.

5.2 Experiment 2: RAG Knowledge Sources

We first conduct evaluation only on two small test set WSC273 and WinoGrande, because the time consumption of ConceptNet (3.5-4.5s) and arXiv (5.0-8.0s) is very high when the test sets are large (10k instances $\times$ 5s $\rightarrow$ 13.9h). The approximated running times of each knowledge source per instance are shown in Table 3.

In the following experiments, we only keep `rte`, `mrpc`, `wnli`, and `sst2` in the GLUE dataset (named "GLUE[4]") because other test sets are too large, making the RAG process too slow. For

	WSC273	WinoGrande	ANLI	ARC_E	ARC_C	PIQA	SWAG	HellaSwag	GLUE	SuperGLUE	**AVG**
GPT-1 (117M)	0.6154	0.5272	0.3307	0.3670	0.3527	0.5881	0.4583	0.2497	0.4794	0.4488	<u>0.4417</u>
GPT-2-Small (124M)	0.5641	0.5185	0.3425	0.4360	0.1911	0.6295	0.4057	0.2895	0.4607	0.4519	0.429
GPT-NEO-125M	0.5531	0.5051	0.3374	0.4377	0.1903	0.6300	0.4051	0.2866	0.5013	0.4607	0.4307
OPT-125M	0.5568	0.5043	0.3635	0.4352	0.1903	0.6284	0.4109	0.2919	0.4829	0.4544	0.4319
GPT-2-Medium (355M)	0.6081	0.5257	0.3373	0.4924	0.2167	0.6752	0.4547	0.3332	0.4975	0.4742	0.4615
OPT-350M	0.6447	0.5257	0.3302	0.4411	0.2082	0.6464	0.4424	0.3201	0.4852	0.4651	0.4509
GPT-2-Large (774M)	0.6300	0.5517	0.3291	0.5316	0.2176	0.7040	0.4721	0.3641	0.4770	0.4755	<u>0.4753</u>
GPT-2-XL (1.6B)	0.6593	0.5833	0.3493	0.5825	0.2500	0.7078	0.4930	0.4002	0.4795	0.4854	0.4990
GPT-NEO-1.3B	0.7179	0.5533	0.3317	0.5623	0.2304	0.7116	0.4953	0.3865	0.5092	0.4702	0.4968
OPT-1.3B	0.7326	0.5959	0.3392	0.5711	0.2346	0.7165	0.5052	0.4152	0.5085	0.4674	0.5086
OLMo-1B	0.7363	0.6014	0.3347	0.6334	0.2867	0.7503	0.5111	0.4694	0.4963	0.5227	<u>0.5342</u>
GPT-NEO-2.7B	0.7326	0.5746	0.3411	0.6107	0.2765	0.7214	0.5177	0.4272	0.5207	0.4816	0.5204
OPT-2.7B	0.7802	0.6109	0.3392	0.6077	0.2679	0.7383	0.5241	0.4586	0.4778	0.5232	0.5328
OpenLLaMA-3B	0.8315	0.6188	0.3212	0.6928	0.3404	0.7503	0.5367	0.4884	0.5193	0.5228	<u>0.5622</u>
OPT-6.7B	0.8168	0.6527	0.3318	0.6561	0.3063	0.7628	0.5446	0.5052	0.5102	0.4905	0.5577
OLMo-7B	0.8462	0.6630	0.3503	0.7340	0.3686	0.7884	0.5508	0.5563	0.5119	0.4993	0.5869
OpenLLaMA-7B	0.8242	0.6661	0.3442	0.7117	0.3754	0.7568	0.5498	0.5256	0.5242	0.5351	0.5813
Mistral-7B	**0.8791**	**0.7403**	**0.4720**	**0.8140**	**0.5410**	**0.8020**	**0.5973**	**0.6601**	**0.6430**	**0.6251**	<u>**0.6774**</u>

Table 2: The experimental results of various baseline LLMs of different model sizes on GLUE, SuperGLUE, and other selected commonsense reasoning datasets. Here, no RAG methods are applied. The scores of ANLI, GLUE, and SuperGLUE in this table are the average scores of their subtasks. "ARC_E" and "ARC_C" represent "ARC_Easy" and "ARC_Challenge" respectively. The evaluation metrics are either accuracy or f1 score, as described in Table 13. The models are divided into several groups according to the number of parameters. The highest score on each task is in **bold** and that of each group is <u>underlined</u>.

	Wikipedia	Google Search	LLM (Gemini)
s/i	~2.0	0.5-1.0	1.0-2.0
	ConceptNet	arXiv Abstract	Atomic-COMET
s/i	3.5-4.5	5.0-8.0	30-60

Table 3: The running time (s/i: second per instance) of the RAG procedure.

	KB	WSC273	WinoGrande	**AVG**
OLMo-1B	*w/o RAG*	0.7363	0.6014	0.6688
	Wikipedia	0.7509	0.5880	0.6694
	Google Search	0.6960	0.5991	0.6475
	LLM (Gemini)	0.6777	0.5596	0.6186
	ConceptNet	0.7473	0.5841	0.6657
	arXiv Abstract	0.7399	0.6014	0.6706
Mistral-7B	*w/o RAG*	0.8791	0.7403	0.8097
	Wikipedia	0.7253	0.6898	0.7075
	LLM (Gemini)	0.7253	0.719	0.7221

Table 4: The experimental results (with RAG) of using different knowledge sources. The green color means the improvement of RAG over baselines and the red color means the performance degradation.

the same reason, we only evaluate the cb, wic, sglue_rte, boolq, copa, and wsc test sets in the SuperGLUE dataset (named "SuperGLUE[6]"). The scores of ANLI, GLUE, and SuperGLUE in this table are the average scores of their subtasks. "ARC_E" and "ARC_C" means "ARC_Easy" and "ARC_Challenge". The evaluation metrics are either accuracy or f1 score, as described in Table 13. The green color means the improvement of RAG over baselines and the red color means the performance degradation.

The evaluation result of the vanilla RAG on the WSC273 and WinoGrande benchmarks using each knowledge source is presented in Table 4. For the Mistral-7B model, challenges arise with memory constraints due to the extensive data retrieved from Google Search, ConceptNet, and arXiv, even when processing at a batch size of 1. As a result, these knowledge sources are not evaluated on the Mistral-7B model, underscoring the necessity for post-processing after document retrieval.

The result shows that larger models like Mistral-7B consistently experience performance degradation. Conversely, smaller models like OLMo-1B demonstrate that RAG, when powered by reliable knowledge sources, can sometimes enhance commonsense understanding.

Given these observations, we decide to focus further analysis on OLMo-1B, where RAG shows potential benefits. For upcoming experiments, we will limit the knowledge sources to Wikipedia, Google Search, and the LLM (Gemini) to mitigate runtime issues. The results of evaluating all benchmarks on OLMo-1B with different knowledge sources is shown in Table 5, which indicates RAG may improve OLMo-1B in commonsense tasks.

5.3 Experiment 3: Pre- and Post-processing

The experimental results in Table 6 and Table 7 demonstrate that introducing pre/post-processing methods, when utilizing all knowledge sources,

	KB	WSC273	WinoGrande	ANLI	ARC_E	ARC_C	PIQA	GLUE[4]	SuperGLUE[6]	AVG
OLMo-1B	*w/o RAG*	0.7363	0.6014	0.3347	0.6334	0.2867	0.7503	0.5806	0.5648	0.5610
	Wikipedia	0.7509	0.5880	0.3295	0.6301	0.2952	0.7416	0.5821	0.5452	0.5578
	Google Search	0.6960	0.5991	0.3341	0.6334	0.2858	0.7508	0.5809	0.5646	0.5556
	LLM (Gemini)	0.6777	0.5596	0.3360	0.6835	0.3643	0.7492	0.5757	0.5751	0.5651
	All	0.7070	0.5809	0.3362	0.6987	0.3635	0.7356	0.5988	0.5695	0.5738

Table 5: The experimental results of the selected baseline LLMs on GLUE, SuperGLUE, and other commonsense reasoning datasets. Here, we apply the basic RAG method by supplying the original query with external knowledge as context. "*All*" means we use all three knowledge sources.

tend to result in poorer performance compared to scenarios where these methods are not employed. This suggests that pre/post-processing may lead to a significant loss of necessary information in retrieved documents. Consequently, it is advisable to avoid using these methods unless essential to keep the length of retrieved documents within limit.

5.4 Experiment 4: Augmentation Methods

The evaluation results for using various augmentation prompts are displayed in Table 8. These results indicate that the basic prompt aligns best with OLMo-1B's generative capabilities. Consequently, we choose to use basic augmentation prompts to integrate queries and RAG documents.

The performance of the model on generation benchmarks using ICL without RAG is shown in Table 9. The result indicates that ICL significantly improves the model's generative capabilities compared to the baseline across most benchmarks. Notably, while adding more examples to ICL typically results in marginal performance gains, it occasionally leads to a decrease in performance.

Conversely, when incorporating RAG, as is shown in Table 10, the benefits of ICL are limited to specific benchmarks, such as WinoGrande and PIQA. Furthermore, variations in the number of ICL examples do not significantly affect performance outcomes in these benchmarks.

6 Future Work

More Experiments. In the future, we can experiment with more variants of our RAG components and extra advanced RAG methods. For example, Chain-of-thought (CoT) prompting (Wei et al., 2022; Kojima et al., 2022) is potentially helpful for these reasoning tasks, although the reasoning path is not easy to obtain. Also, supervised fine-tuning (SFT) via instruction tuning (Wei et al., 2021; Sanh et al., 2021; Longpre et al., 2023) is worth exploring because LLMs might use the augmented documents better after fine-tuning. Besides, preference alignment using RLHF (Ouyang et al., 2022) or DPO (Rafailov et al., 2023) could also help LLMs utilize key information of the RAG retrievals. In addition, we can consider using GPT-3.5 (Brown et al., 2020; OpenAI, 2022) or GPT-4 (OpenAI, 2023) to serve as the baseline or play the agent roles for pre-processing, post-processing, and augmentation. In this project, we only use Google Gemini since it is free.

Efficient RAG. As mentioned in § 5.2, the online searching RAG method we adopt is not efficient. To improve the efficiency, we can apply offline RAG and reuse the cached retrievals to speed up the experiments of multiple runs. Another plan is to adopt multiprocessing to perform RAG retrieving of different instances simultaneously.

Traditional RAG. The performance of RAG methods in our experiments is less than ideal, potentially because we use the searching API for retrieval. We consider implementing the traditional RAG pipeline: (1) Download and parse a large knowledge base[9] and cut it into chunks; (2) Use or train an embedding model to obtain dense representations of chunks; (3) Encode the query and perform semantic matching and ranking to retrieve relevant documents.

Deep into RAG. Recently, RAG has been a trendy and promising topic in academia and industry, and many research ideas and engineering tricks have been proposed to improve its effectiveness. Although the performance of RAG in our experiments is not satisfactory, we believe this is mainly because simply adding extra factual knowledge can hardly assist LLMs in solving reasoning-intensive tasks. To fully unleash the potential of RAG, we should dig into the following questions: What knowledge is needed to solve the problem? What do LLMs know and do not know?

[9]For example, Wikipedia, WikiText, and RedPajama.

	KB	WSC273	WinoGrande	ARC_E	ARC_C	PIQA	AVG
	RAG w/o Pre-processing	0.7070	0.5809	0.6987	0.3635	0.7356	0.6171
OLMo-1B	keyword_extraction	0.6557	0.5714	0.6271	0.2969	0.7171	0.5736
	contextual_clarification	0.6996	0.5714	0.6595	0.3072	0.7329	0.5941
	relevance_filtering	0.6117	0.5596	0.6221	0.3012	0.7225	0.5634
	query_expansion	0.6740	0.5809	0.6448	0.3166	0.7318	0.5896
	information_structuring	0.6630	0.5675	0.6322	0.3276	0.7263	0.5833
	intent_clarification	0.6996	0.5651	0.6170	0.2927	0.7296	0.5808

Table 6: he results of using different RAG pre-processing methods, compared with not using any pre-processing.

	KB	WSC273	WinoGrande	ARC_E	ARC_C	PIQA	AVG
	RAG w/o Post-processing	0.7070	0.5809	0.6987	0.3635	0.7356	0.6171
OLMo-1B	ranking_documents	0.6154	0.5888	0.6136	0.2910	0.7182	0.5654
	summarizing_documents	0.6227	0.5454	0.6423	0.3200	0.7280	0.5717
	extracting_key_info	0.5788	0.5627	0.6372	0.3234	0.7214	0.5647
	refining_documents	0.6081	0.5501	0.6397	0.3225	0.7171	0.5675
	evaluating_documents	0.6520	0.5493	0.6183	0.2952	0.7176	0.5665
	identifying_conflict	0.6044	0.5470	0.6107	0.2961	0.7285	0.5573

Table 7: The results of using different RAG post-processing methods, compared with not using any post-processing.

	KB	WSC273	WinoGrande	ARC_E	ARC_C	PIQA	AVG
	RAG w/ Basic Prompt	0.7070	0.5809	0.6987	0.3635	0.7356	0.6171
OLMo-1B	Short Prompt	0.5311	0.4870	0.6486	0.3618	0.7443	0.5546
	Medium Prompt	0.5018	0.4949	0.6477	0.3592	0.7345	0.5476
	Long Prompt	0.5421	0.4988	0.6494	0.3575	0.7405	0.5577

Table 8: LLM generation performance using different prompt templates to combine the original query with retrievals.

	KB	WSC273	WinoGrande	ARC_E	ARC_C	PIQA	AVG
	LLM w/o RAG; w/o ICL	0.7363	0.6014	0.6334	0.2867	0.7503	0.6016
OLMo-1B	w/ ICL 1 Example	0.7802	0.6030	0.6578	0.3131	0.7535	0.6215
	w/ ICL 3 Examples	0.7839	0.5927	0.6604	0.3166	0.7492	0.6206
	w/ ICL 5 Examples	0.7582	0.6069	0.6662	0.3294	0.7617	0.6245
	w/ ICL 8 Examples	0.7692	0.6093	0.6629	0.3251	0.7590	0.6251
	w/ ICL 10 Examples	0.8059	0.5927	0.6705	0.3200	0.7644	**0.6307**

Table 9: LLM generation performance (without RAG) using different number of examples for in-context learning.

	KB	WSC273	WinoGrande	ARC_E	ARC_C	PIQA	AVG
	RAG w/o ICL	0.7070	0.5809	0.6987	0.3635	0.7356	0.6171
OLMo-1B	w/ ICL 1 Example	0.6484	0.5904	0.6864	0.3541	0.7514	0.6061
	w/ ICL 3 Examples	0.6630	0.6022	0.6932	0.3507	0.7557	0.6130
	w/ ICL 5 Examples	0.6337	0.5864	0.6860	0.3592	0.7557	0.6042
	w/ ICL 8 Examples	0.6667	0.6054	0.6860	0.3532	0.7606	0.6144
	w/ ICL 10 Examples	0.6484	0.5975	0.6869	0.3532	0.7628	0.6098

Table 10: LLM generation performance (with RAG) using different number of examples for in-context learning.

7 Conclusion

In this work, we explored retrieval-augmented generation (RAG) for large language models (LLMs) focusing on natural language understanding and reasoning by comprehensively reviewing the related literature, systematically implementing the entire RAG pipeline, and conducting extensive experiments to evaluate the effectiveness of different RAG components.

Our RAG framework includes query obtaining and pre-processing, keyword extraction, document retrieval and post-processing, query augmentation, LLM generation, and result evaluation. The implementation is systematic, flexible, and scalable, and the code is publicly available.

As the experimental results demonstrate, we compared the performance of different LLM baselines, knowledge sources, pre- and post-processing approaches, augmentation prompts, and in-context learning methods. Our analysis and discussions about the findings provide insights into RAG's effectiveness and feasibility on reasoning-intensive tasks, which sheds light on future RAG research.

References

Renat Aksitov, Chung-Ching Chang, David Reitter, Siamak Shakeri, and Yunhsuan Sung. 2023. Characterizing attribution and fluency tradeoffs for retrieval-augmented large language models. *arXiv preprint arXiv:2302.05578*.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: visual question answering. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 2425–2433. IEEE Computer Society.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosiute, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemí Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, abs/2212.08073.

Moshe Berchansky, Peter Izsak, Avi Caciularu, Ido Dagan, and Moshe Wasserblat. 2023. Optimizing retrieval-augmented reader models via token elimination. *arXiv preprint arXiv:2310.13682*.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen-tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7432–7439. AAAI Press.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*, abs/2204.06745.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, volume 33, pages 1877–1901, Virtual Event. NeurIPS.

Yingshan Chang, Guihong Cao, Mridu Narang, Jianfeng Gao, Hisami Suzuki, and Yonatan Bisk. 2022. Webqa: Multihop and multimodal QA. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 16474–16483. IEEE.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harrison Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan

Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, abs/2107.03374.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Wanqing Cui, Keping Bi, Jiafeng Guo, and Xueqi Cheng. 2024. More: Multi-modal retrieval augmented generative commonsense reasoning. *arXiv preprint arXiv:2402.13625*, abs/2402.13625.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*, abs/2301.00234.

Zhangyin Feng, Xiaocheng Feng, Dezhi Zhao, Maojin Yang, and Bing Qin. 2023. Retrieval-generation synergy augmented large language models. *arXiv preprint arXiv:2310.05149*, abs/2310.05149.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, abs/2312.10997.

Deepanway Ghosal, Somak Aditya, and Monojit Choudhury. 2023. Prover: Generating intermediate steps for NLI with commonsense knowledge retrieval and next-step prediction. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 872–884, Nusa Dua, Bali. Association for Computational Linguistics.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Raghavi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, Will Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah A. Smith, and Hannaneh Hajishirzi. 2024. Olmo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838*, abs/2402.00838.

David Gunning. 2018. Machine common sense concept paper. *arXiv preprint arXiv:1810.07528*.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *arXiv preprint arXiv:2402.07630*, abs/2402.07630.

Ziniu Hu, Ahmet Iscen, Chen Sun, Zirui Wang, Kai-Wei Chang, Yizhou Sun, Cordelia Schmid, David A. Ross, and Alireza Fathi. 2023. Reveal: Retrieval-augmented visual-language pre-training with multi-source multimodal knowledge memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23369–23379.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2391–2401, Hong Kong, China. Association for Computational Linguistics.

Jena D. Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. (comet-) atomic 2020: On symbolic and neural commonsense knowledge graphs. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 6384–6392. AAAI Press.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):248:1–248:38.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Zhengbao Jiang, Zhiqing Sun, Weijia Shi, Pedro Rodriguez, Chunting Zhou, Graham Neubig, Xi Victoria Lin, Wen tau Yih, and Srinivasan Iyer. 2024. Instruction-tuned language models are better knowledge learners. *arXiv preprint arXiv:2402.12847*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262, New Orleans, Louisiana. Association for Computational Linguistics.

Hyunwoo Kim, Jack Hessel, Liwei Jiang, Peter West, Ximing Lu, Youngjae Yu, Pei Zhou, Ronan Bras, Malihe Alikhani, Gunhee Kim, Maarten Sap, and Yejin Choi. 2023. SODA: Million-scale dialogue distillation with social commonsense contextualization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12930–12949, Singapore. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Yash Kumar Lal, Niket Tandon, Tanvi Aggarwal, Horace Liu, Nathanael Chambers, Raymond Mooney, and Niranjan Balasubramanian. 2022. Using commonsense knowledge to answer why-questions. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1204–1219, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June 10-14, 2012*. AAAI Press.

Patrick S. H. 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. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Haonan Li, Yeyun Gong, Jian Jiao, Ruofei Zhang, Timothy Baldwin, and Nan Duan. 2021. Kfcnet: Knowledge filtering and contrastive learning for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2918–2928, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. Commongen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online. Association for Computational Linguistics.

Fangyu Liu, Emanuele Bugliarello, Edoardo Maria Ponti, Siva Reddy, Nigel Collier, and Desmond Elliott. 2021. Visually grounded reasoning across languages and cultures. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10467–10485, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022. Generated knowledge prompting for commonsense reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3154–3169, Dublin, Ireland. Association for Computational Linguistics.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315, Singapore. Association for Computational Linguistics.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Generation-augmented retrieval for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4089–4100, Online. Association for Computational Linguistics.

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, et al. 2023. Augmented language models: a survey. *arXiv preprint arXiv:2302.07842*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James F. Allen. 2016. A corpus and evaluation framework for deeper understanding of commonsense stories. *arXiv preprint arXiv:1604.01696*, abs/1604.01696.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

OpenAI. 2022. Chatgpt. *OpenAI Research*.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Sarah M. Pratt, Mark Yatskar, Luca Weihs, Ali Farhadi, and Aniruddha Kembhavi. 2020. Grounded situation recognition. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part IV*, volume 12349 of *Lecture Notes in Computer Science*, pages 314–332. Springer.

Lianhui Qin, Aditya Gupta, Shyam Upadhyay, Luheng He, Yejin Choi, and Manaal Faruqui. 2021. TIMEDIAL: Temporal commonsense reasoning in dialog. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7066–7076, Online. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. *OpenAI blog*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, abs/2305.18290.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 784–789. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Simon Razniewski, Niket Tandon, and Aparna S. Varde. 2021. Information to wisdom: Commonsense knowledge extraction and compilation. In *WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8-12, 2021*, pages 1143–1146. ACM.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *Logical Formalizations of Commonsense Reasoning, Papers from the 2011 AAAI Spring Symposium, Technical Report SS-11-06, Stanford, California, USA, March 21-23, 2011*. AAAI.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473, Hong Kong, China. Association for Computational Linguistics.

Maarten Sap, Vered Shwartz, Antoine Bosselut, Yejin Choi, and Dan Roth. 2020. Commonsense reasoning for natural language processing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 27–33, Online. Association for Computational Linguistics.

Jaehyung Seo, Dongsuk Oh, Sugyeong Eo, Chanjun Park, Kisu Yang, Hyeonseok Moon, Kinam Park, and Heuiseok Lim. 2022. PU-GEN: enhancing generative commonsense reasoning for language models with human-centered knowledge. *Knowledge-Based Systems*, 256:109861.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9248–9274, Singapore. Association for Computational Linguistics.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. ALFRED: A benchmark for interpreting grounded instructions for everyday tasks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10737–10746. Computer Vision Foundation / IEEE.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. *arXiv preprint arXiv:2402.00159*, abs/2402.00159.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4444–4451. AAAI Press.

Alane Suhr, Mike Lewis, James Yeh, and Yoav Artzi. 2017. A corpus of natural language for visual reasoning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 217–223, Vancouver, Canada. Association for Computational Linguistics.

Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. 2019. A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6418–6428, Florence, Italy. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4149–4158. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem 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, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, abs/2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3261–3275.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Hongru Wang, Wenyu Huang, Yang Deng, Rui Wang, Zezhong Wang, Yufei Wang, Fei Mi, Jeff Z. Pan, and Kam-Fai Wong. 2024. Unims-rag: A unified multi-source retrieval-augmented generation for personalized dialogue systems. *arXiv preprint arXiv:2401.13256*, abs/2401.13256.

Zhiruo Wang, Jun Araki, Zhengbao Jiang, Md Rizwan Parvez, and Graham Neubig. 2023. Learning to filter context for retrieval-augmented generation. *arXiv preprint arXiv:2311.08377*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. 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 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Terry Winograd. 1972. Understanding natural language. *Cognitive Psychology*, 3(1):1–191.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*, abs/2401.15884.

Yue Yang, Artemis Panagopoulou, Qing Lyu, Li Zhang, Mark Yatskar, and Chris Callison-Burch. 2021. Visual goal-step inference using wikiHow. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2167–2179, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Da Yin, Feng Gao, Govind Thattai, Michael Johnston, and Kai-Wei Chang. 2023. GIVL: improving geographical inclusivity of vision-language models with pre-training methods. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 10951–10961. IEEE.

Da Yin, Liunian Harold Li, Ziniu Hu, Nanyun Peng, and Kai-Wei Chang. 2021. Broaden the vision: Geo-diverse visual commonsense reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2115–2129, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2023. Making retrieval-augmented language models robust to irrelevant context. *arXiv preprint arXiv:2310.01558*, abs/2310.01558.

Wenhao Yu. 2022. Retrieval-augmented generation across heterogeneous knowledge. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop*, pages 52–58, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.

Wenhao Yu, Chenguang Zhu, Zhihan Zhang, Shuohang Wang, Zhuosheng Zhang, Yuwei Fang, and Meng Jiang. 2022. Retrieval augmentation for commonsense reasoning: A unified approach. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4364–4377, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019a. From recognition to cognition: Visual commonsense reasoning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6720–6731. Computer Vision Foundation / IEEE.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019b. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension. *arXiv preprint arXiv:1810.12885*, abs/1810.12885.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2023. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*, abs/2308.10792.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, abs/2205.01068.

Ruochen Zhao, Hailin Chen, Weishi Wang, Fangkai Jiao, Xuan Long Do, Chengwei Qin, Bosheng Ding, Xiaobao Guo, Minzhi Li, Xingxuan Li, and Shafiq Joty. 2023a. Retrieving multimodal information for augmented generation: A survey. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4736–4756, Singapore. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023b. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Qinkai Zheng, Xiao Xia, Xu Zou, Yuxiao Dong, Shan Wang, Yufei Xue, Zihan Wang, Lei Shen, Andi Wang, Yang Li, Teng Su, Zhilin Yang, and Jie Tang. 2023. Codegeex: A pre-trained model for code generation with multilingual evaluations on humaneval-x. *arXiv preprint arXiv:2303.17568*, abs/2303.17568.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. "going on a vacation" takes longer than "going for a walk": A study of temporal commonsense understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3363–3369, Hong Kong, China. Association for Computational Linguistics.

A Relevant Knowledge Sources

For building the retrieval-augmented generation system, we list commonsense-related datasets that may serve as knowledge sources in Table 12.

B Tasks & Datasets Details

Table 13 shows the statistics of datasets for evaluation, including GLUE (Wang et al., 2019b) and SuperGLUE (Wang et al., 2019a) benchmarks and multiple commonsense reasoning tasks.

C Detailed Experimental Settings

C.1 Pre-processing Prompts

Table 14 to Table 19 separately presents each prompt template for the LLM agent to perform query pre-processing, including keyword extraction, contextual clarification, relevance filtering, query expansion, information structuring, and intent clarification.

C.2 Post-processing Prompts

Table 20 to Table 25 separately introduces each prompt template for the LLM agent to perform document post-processing, including document ranking, document summarization, key information extraction, refine documents, document evaluation, and conflict identification.

C.3 Augmentation Prompts

Table 28, Table 29, and Table 30 illustrate short, medium, and long prompt template for the LLM agent to perform query augmentation, respectively.

C.4 LLM Training and Generation

We implement the LLM training and generation using PyTorch (Paszke et al., 2019) and Hugging Face `transformers` (Wolf et al., 2020) toolkits. The loss curves of fine-tuning LLMs (GPT-2) is shown in Figure 2, which demonstrates that we can successfully perform instruction tuning.

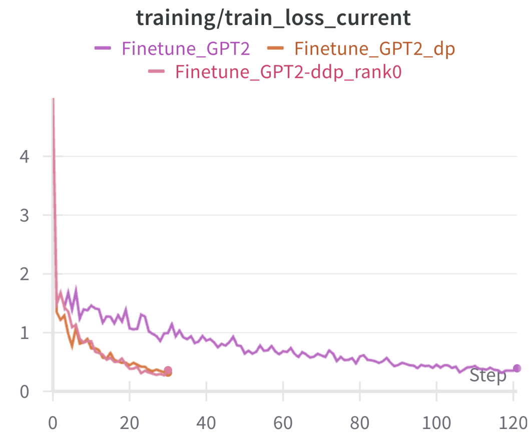

Figure 2: The losses of fine-tuning LLMs (GPT-2) using a single GPU, 4 GPUs with data parallel (dp), and 4 GPUs with distributed data parallel (ddp).

D Detailed Experimental Results

Due to the page limit (9 pages for the main body), we present the detailed results of Experiment 1 (Table 2) in Table 31, Table 32, and Table 33.

E Task Division and Timeline

E.1 Task Division

We work together on research idea discussion and method implementation. The main focus of each member is shown in Table 11. We all learned a lot in completing this project.

Task	**Juntai**	**Yilin**	**Yuwei**
Step 1: Obtain the Queries			✔
Step 1.5: Query Pre-processing	✔	✔	
Step 2: Keywords Extraction	✔	✔	
Step 3: Documents Retrieval	✔	✔	
Step 3.5: Docs Post-processing	✔	✔	
Step 4: Query Augmentation			✔
Step 5: LLM Generation & Tuning			✔
Step 6: Result Evaluation			✔
Run Experiments	✔	✔	✔
Results analysis	✔	✔	✔
Paper writing	✔	✔	✔

Table 11: Task division of group members.

Dataset	Knowledge Type	Website	API
GLUE (Wang et al., 2019b)	NLU & Commonsense	gluebenchmark.com	huggingface.co
SuperGLUE (Wang et al., 2019a)	NLU & Commonsense	super.gluebenchmark.com	huggingface.co
SNLI (Bowman et al., 2015)	NLI	nlp.stanford.edu	huggingface.co
Adversarial NLI (Nie et al., 2020)	NLI	github.com	huggingface.co
OpenBookQA (Mihaylov et al., 2018)	Subject & Commonsense	allenai.org	huggingface.co
ARC (Clark et al., 2018)	Science QA	allenai.org	huggingface.co
CommonGen (Lin et al., 2020)	Daily-life Commonsense	inklab.usc.edu	huggingface.co
Cosmos QA (Huang et al., 2019)	Commonsense Reading Comprehension	github.io	huggingface.co
MultiRC (Khashabi et al., 2018) (in SuperGLUE)	Commonsense Reading Comprehension	cogcomp.seas.upenn.edu	huggingface.co
ReCORD (Zhang et al., 2018) (in SuperGLUE)	Commonsense Reading Comprehension	github.io	huggingface.co
Social IQA (Sap et al., 2019)	Social Commonsense	allenai.org	huggingface.co
COPA (Roemmele et al., 2011) (in GLUE)	Social Commonsense	ict.usc.edu	huggingface.co
WSC (Levesque et al., 2012) (in GLUE)	Social Commonsense	cs.nyu.edu	huggingface.co
RocStories (Mostafazadeh et al., 2016)	Social Commonsense	cs.rochester.edu	huggingface.co
SODA (Kim et al., 2023)	Social Commonsense	github.com	huggingface.co
PIQA (Bisk et al., 2020)	Physical Commonsense	allenai.org	huggingface.co
SWAG (Zellers et al., 2018)	Physical Commonsense	rowanzellers.com	huggingface.co
WinoGrande (Sakaguchi et al., 2021)	Social+Physical Commonsense	allenai.org	huggingface.co
Commonsense QA (Talmor et al., 2019)	Social+Physical Commonsense	tau-nlp.sites.tau.ac.il	huggingface.co
Abductive NLI (Bhagavatula et al., 2020)	Social+Physical Commonsense	allenai.org	-
HellaSwag (Zellers et al., 2019b)	Physical+Temporal Commonsense	rowanzellers.com	huggingface.co
MCTaco (Zhou et al., 2019)	Temporal Commonsense	allenai.org	huggingface.co
TimeDial (Qin et al., 2021)	Temporal Commonsense	github.com	huggingface.co
VQA (Antol et al., 2015)	Multimodal Commonsense	visualqa.org	visualqa.org
VCR (Zellers et al., 2019a)	Multimodal Commonsense	visualcommonsense.com	github.com
NLVR (Suhr et al., 2017, 2019)	Multimodal Commonsense	nlp.cornell.edu	github.com
WebQA (Chang et al., 2022)	Multimodal Commonsense	github.io	github.com
GSR (Pratt et al., 2020)	Multimodal Commonsense	allenai.org	github.com
VGSI (Yang et al., 2021)	Multimodal Commonsense	github.com	-
ALFRED (Shridhar et al., 2020)	Multimodal Commonsense	askforalfred.com	github.com
MaRVL (Liu et al., 2021)	Cultural Commonsense	github.io	github.com
GD-VCR (Yin et al., 2021)	Cultural Commonsense	github.com	-
GIVL (Yin et al., 2023)	Cultural Commonsense	github.com	-

Table 12: Relevant knowledge sources.

Dataset	Training	Validation	Test	# Class	Metric
GLUE (Wang et al., 2019b)					
`rte` (Recognizing Textual Entailment)	2.49k	277	3k	2	Acc
`qnli` (Question NLI)	105k	5.46k	5.46k	2	Acc
`mnli` (MultiNLI Matched)	393k	9.82k	9.8k	3	Acc
`mnli` (MultiNLI Mismatched)	393k	9.83k	9.85k	3	Acc
`mrpc` (Microsoft Research Paraphrase Corpus)	3.67k	408	1.73k	2	Acc, **F1**
`qqp` (Quora Question Pairs)	364k	40.4k	391k	2	Acc, **F1**
`wnli` (Winograd NLI)	635	71	146	2	Acc
`sst2` (The Stanford Sentiment Treebank)	67.3k	872	1.82k	2	Acc
SuperGLUE (Wang et al., 2019a)					
`cb` (CommitmentBank)	250	56	250	3	Acc, **F1**
`wic` (Words in Context)	5.43k	638	1.4k	2	Acc
`sglue_rte` (Recognizing Textual Entailment)	2.49k	277	3k	2	Acc
`boolq` (BoolQ)	9.43k	3.27k	3.25k	2	Acc
`copa` (Choice of Plausible Alternatives)	500	100	400	2	Acc
`multirc` (Multi-Sentence Reading Comprehension)	27.2k	4.85k	9.69k	2	Acc
`record` (Reading Comprehension with Commonsense Reasoning)	101k	10k	10k	N	**F1**, EM
`wsc` (The Winograd Schema Challenge)	554	104	146	2	Acc
WSC273 (Levesque et al., 2012)	-	-	273	2	Acc
WinoGrande (Sakaguchi et al., 2021)	9.25k	1.27k	1.77k	2	Acc
ANLI r1 (Nie et al., 2020)	16.9k	1k	1k	3	Acc
ANLI r2	45.5k	1k	1k	3	Acc
ANLI r3	100k	1.2k	1.2k	3	Acc
ARC Easy (Clark et al., 2018)	2.25k	570	2.38k	4	Acc
ARC Challenge	1.12k	299	1.17k	4	Acc
PIQA (Bisk et al., 2020)	16.1k	1.84k	3.08k	2	Acc
SWAG (Zellers et al., 2018)	73.5k	20k	20k	4	Acc
HellaSwag (Zellers et al., 2019b)	39.9k	10k	10k	4	Acc

Table 13: The statistics of datasets for evaluation. All of them are classification tasks, where the training and validation set have labels, while the test set does not. For the evaluation metrics, “Acc”, “F1”, and “EM” mean accuracy, f1 score, and exact match score, respectively. In the experiments, we only report the f1 scores for tasks with multiple evaluation metrics.

Given the query: "{query}"
Extract the main keywords from the above query. Simplify the query to its most essential components or keywords to aid in efficient information retrieval.

Table 14: Prompt template for pre-processing: keyword extraction.

Given the query: "{query}"
Clarify the above query by rephrasing it into a more specific question or statement. Ensure the revised context is concise and directly related to the core topic for effective external information retrieval.

Table 15: Prompt template for pre-processing: contextual clarification.

Given the query: "{query}"
Identify and remove any irrelevant details from the above query that may hinder the retrieval of focused information. Summarize the refined query to emphasize the most relevant aspects.

Table 16: Prompt template for pre-processing: relevance filtering.

Given the query: "{query}"
Expand the above query by adding related terms or questions that might help in retrieving more comprehensive and relevant information from external sources.

Table 17: Prompt template for pre-processing: query expansion.

Given the query: "{query}"
Structure the information within the above query into a clear and organized format. Categorize the details into themes or topics to facilitate targeted information retrieval.

Table 18: Prompt template for pre-processing: information structuring.

Given the query: "{query}"
Clarify the intent behind the above query by rephrasing it into a more direct query. Highlight the main goal or the type of information sought to guide the retrieval process effectively.

Table 19: Prompt template for pre-processing: intent clarification.

Given the original query: "{query}"
Give a list of retrieved documents, where each document separated by "\n":
{docs}
Rank these documents in order of their relevance to the original query. Provide the ranked list.

Table 20: Prompt template for post-processing: document ranking.

Given the original query: "{query}"
Give a list of retrieved documents, where each document separated by "\n":
{docs}
Summarize the above documents by extracting its core message or information. Ensure the summary is concise and captures the essence of the document related to the original query.

Table 21: Prompt template for post-processing: document summarization.

Given the original query: "{query}"
Give a list of retrieved documents, where each document separated by "\n":
{docs}
From the above documents, extract the most critical pieces of information related to the original query. Organize the information by relevance and clarity.

Table 22: Prompt template for post-processing: key information extraction.

Given the original query: "{query}"
Give a list of retrieved documents, where each document separated by "\n":
{docs}
Refine and clarify the content of the above documents to make it more directly related to the original query. Remove any irrelevant details and enhance the clarity of the document's main points.

Table 23: Prompt template for post-processing: refine documents.

Given the original query: "{query}"
Give a list of retrieved documents, where each document separated by "\n":
{docs}
Evaluate the relevance and quality of the above documents in relation to the original query. Provide a brief assessment highlighting its relevance, accuracy, and any biases or inaccuracies detected.

Table 24: Prompt template for post-processing: document evaluation

Given the original query: "{query}"
Give a list of retrieved documents, where each document separated by "\n":
{docs}
Identify and highlight any agreements or contradictions among the above documents with respect to the original query. Summarize the points of agreement or conflict.

Table 25: Prompt template for post-processing: conflict identification

Given the original query: "{query}"
Give a list of retrieved documents, where each document separated by "\n":
{docs}
Identify and remove duplicate information found across the above documents. Provide a cleaned-up version of the content that retains unique information relevant to the original query.

Table 26: Prompt template for post-processing: filtering out duplication.

Given the original query: "{query}"
Give a list of retrieved documents, where each document separated by "\n":
{docs}
Transform the key information found in the above documents into a structured format (e.g., bullet points, tables) to make the information more accessible and understandable in relation to the original query.

Table 27: Prompt template for post-processing: convert documents to structural information.

Give a list of retrieved documents, where each document separated by "\n":
{docs}
Based on the above documents, generate an accurate, concise, and reasonable answer to the following query:
{query}

Table 28: Prompt template for RAG augmentation (short prompt).

Give the relevant information extracted from external documents as follows:
{docs}
Using the key information from the above documents to create an accurate, concise, and reasonable response. Aim for coherence and insight, addressing the query with depth and clarity. Highlight any significant agreements or contradictions from the external information, ensuring a balanced view. Answer the following query:
{query}

Table 29: Prompt template for augmentation (medium prompt).

Give the relevant information extracted from external documents as follows:
{docs}
Generate a comprehensive response that incorporates this information to provide an accurate, concise, and reasonable answer. The response should reflect an understanding of the query's intent and the knowledge contained within the processed documents. Ensure the generated content is coherent, logically structured, and seamlessly integrates the external information to enhance the quality and depth of the answer. If the processed information supports or contradicts the query, highlight these aspects appropriately, providing a balanced and informed perspective. Answer the following query:
{query}

Table 30: Prompt template for augmentation (long prompt).

Model	rte	qnli	mnli	mnli_mismatch
GPT-1 (117M)	0.5343 ±0.0300	0.5193 ±0.0068	0.3513 ±0.0048	0.3492 ±0.0048
GPT-2-Small (124M)	0.5199 ±0.0301	0.5016 ±0.0068	0.3376 ±0.0048	0.3325 ±0.0048
GPT-2-Medium (355M)	0.5271 ±0.0301	0.4946 ±0.0068	0.3517 ±0.0048	0.3510 ±0.0048
GPT-2-Large (774M)	0.5235 ±0.0301	0.4937 ±0.0068	0.3592 ±0.0048	0.3598 ±0.0048
GPT-2-XL (1.6B)	0.5235 ±0.0301	0.5135 ±0.0068	0.3650 ±0.0049	0.3697 ±0.0049
GPT-NEO-125M	0.5451 ±0.0300	0.4946 ±0.0068	0.3551 ±0.0048	0.3538 ±0.0048
GPT-NEO-1.3B	0.6029 ±0.0295	0.4984 ±0.0068	0.3577 ±0.0048	0.3626 ±0.0048
GPT-NEO-2.7B	0.5235 ±0.0301	0.5081 ±0.0068	0.3401 ±0.0048	0.3376 ±0.0048
OPT-125M	0.5018 ±0.0301	0.4944 ±0.0068	0.3446 ±0.0048	0.3492 ±0.0048
OPT-350M	0.5199 ±0.0301	0.4953 ±0.0068	0.3447 ±0.0048	0.3503 ±0.0048
OPT-1.3B	0.5235 ±0.0301	0.5140 ±0.0068	0.3573 ±0.0048	0.3525 ±0.0048
OPT-2.7B	0.5487 ±0.0300	0.5114 ±0.0068	0.3556 ±0.0048	0.3535 ±0.0048
OPT-6.7B	0.5523 ±0.0299	0.5081 ±0.0068	0.3282 ±0.0047	0.3334 ±0.0048
OLMo-1B	0.5560 ±0.0299	0.5067 ±0.0068	0.3610 ±0.0048	0.3592 ±0.0048
OLMo-7B	0.5271 ±0.0301	0.4973 ±0.0068	0.3295 ±0.0047	0.3350 ±0.0048
OpenLLaMA-3B	0.5451 ±0.0300	0.5114 ±0.0068	0.3747 ±0.0049	0.3784 ±0.0049
OpenLLaMA-7B	0.6101 ±0.0294	0.5059 ±0.0068	0.3953 ±0.0049	0.4032 ±0.0049
Mistral-7B	0.704 ±0.0275	0.5847 ±0.0067	0.5532 ±0.005	0.5561 ±0.0050

Model	mrpc	qqp	wnli	sst2
GPT-1 (117M)	0.7915 ±0.0175	0.2752 ±0.0037	0.5211 ±0.0597	0.4931 ±0.0169
GPT-2-Small (124M)	0.6594 ±0.0234	0.3648 ±0.0035	0.4225 ±0.0590	0.5470 ±0.0169
GPT-2-Medium (355M)	0.8122 ±0.0163	0.4106 ±0.0031	0.4225 ±0.059	0.6101 ±0.0165
GPT-2-Large (774M)	0.7988 ±0.0169	0.3441 ±0.0034	0.4366 ±0.0593	0.5000 ±0.0169
GPT-2-XL (1.6B)	0.7817 ±0.0178	0.2567 ±0.0036	0.5352 ±0.0596	0.4908 ±0.0169
GPT-NEO-125M	0.8122 ±0.0163	0.5017 ±0.0028	0.4225 ±0.0590	0.5252 ±0.0169
GPT-NEO-1.3B	0.8122 ±0.0163	0.2808 ±0.0036	0.5352 ±0.0596	0.6239 ±0.0164
GPT-NEO-2.7B	0.8122 ±0.0163	0.3332 ±0.0034	0.5493 ±0.0595	0.7615 ±0.0144
OPT-125M	0.8122 ±0.0163	0.3568 ±0.0034	0.4648 ±0.0596	0.5390 ±0.0169
OPT-350M	0.8122 ±0.0163	0.3826 ±0.0033	0.3803 ±0.0580	0.5963 ±0.0166
OPT-1.3B	0.7926 ±0.0175	0.2569 ±0.0036	0.4225 ±0.0590	0.8486 ±0.0121
OPT-2.7B	0.8006 ±0.0169	0.3280 ±0.0035	0.4085 ±0.0588	0.5161 ±0.0169
OPT-6.7B	0.7589 ±0.0189	0.3718 ±0.0034	0.4648 ±0.0596	0.7638 ±0.0144
OLMo-1B	0.7468 ±0.0194	0.4208 ±0.0033	0.5070 ±0.0598	0.5126 ±0.0169
OLMo-7B	0.8122 ±0.0163	0.4494 ±0.0033	0.5634 ±0.0593	0.5814 ±0.0167
OpenLLaMA-3B	0.7504 ±0.0198	0.3702 ±0.0034	0.5352 ±0.0596	0.6892 ±0.0157
OpenLLaMA-7B	0.7034 ±0.0227	0.4863 ±0.0029	0.5211 ±0.0597	0.6732 ±0.0159
Mistral-7B	0.8310 ±0.0160	0.4389 ±0.0036	0.6197 ±0.0580	0.8567 ±0.0119

Table 31: The experimental results (with standard deviation) of various baseline LLMs of different model sizes on the GLUE (Wang et al., 2019b) benchmark. The evaluation metrics of each task are described in Table 13. Here, no RAG methods are applied.

Model	cb (±N/A)	wic	sglue_rte	boolq
GPT-1 (117M)	0.1712	0.5188 ±0.0198	0.5343 ±0.0300	0.5098 ±0.0087
GPT-2-Small (124M)	0.2441	0.4969 ±0.0198	0.5199 ±0.0301	0.4832 ±0.0087
GPT-2-Medium (355M)	0.2360	0.5000 ±0.0198	0.5271 ±0.0301	0.5847 ±0.0086
GPT-2-Large (774M)	0.2296	0.4969 ±0.0198	0.5235 ±0.0301	0.6049 ±0.0086
GPT-2-XL (1.6B)	0.2170	0.4984 ±0.0198	0.5235 ±0.0301	0.6180 ±0.0085
GPT-NEO-125M	0.1941	0.5000 ±0.0198	0.5451 ±0.0300	0.6171 ±0.0085
GPT-NEO-1.3B	0.2631	0.5000 ±0.0198	0.5993 ±0.0295	0.6202 ±0.0085
GPT-NEO-2.7B	0.2904	0.5000 ±0.0198	0.5235 ±0.0301	0.6190 ±0.0085
OPT-125M	0.1450	0.5000 ±0.0198	0.5054 ±0.0301	0.5544 ±0.0087
OPT-350M	0.2401	0.5000 ±0.0198	0.5235 ±0.0301	0.5777 ±0.0086
OPT-1.3B	0.2057	0.5078 ±0.0198	0.5271 ±0.0301	0.5771 ±0.0086
OPT-2.7B	0.3017	0.5031 ±0.0198	0.5487 ±0.0300	0.6037 ±0.0086
OPT-6.7B	0.1833	0.4843 ±0.0198	0.5523 ±0.0299	0.6606 ±0.0083
OLMo-1B	0.2456	0.5235 ±0.0198	0.5560 ±0.0299	0.6190 ±0.0085
OLMo-7B	0.1418	0.5016 ±0.0198	0.5271 ±0.0301	0.7242 ±0.0078
OpenLLaMA-3B	0.2221	0.4702 ±0.0198	0.5451 ±0.0300	0.6694 ±0.0082
OpenLLaMA-7B	0.4017	0.4875 ±0.0198	0.6101 ±0.0294	0.7040 ±0.0080
Mistral-7B	0.6805	0.6034 ±0.0194	0.7040 ±0.0275	0.8532 ±0.0062

Model	copa	multirc	record	wsc
GPT-1 (117M)	0.7100 ±0.0456	0.4709 ±0.0072	0.2429 ±0.0043	0.4327 ±0.0488
GPT-2-Small (124M)	0.6200 ±0.0488	0.5344 ±0.0072	0.2646 ±0.0044	0.4519 ±0.0490
GPT-2-Medium (355M)	0.6900 ±0.0465	0.5287 ±0.0072	0.3038 ±0.0046	0.4231 ±0.0487
GPT-2-Large (774M)	0.7200 ±0.0451	0.4872 ±0.0072	0.2994 ±0.0045	0.4423 ±0.0489
GPT-2-XL (1.6B)	0.7600 ±0.0429	0.4645 ±0.0072	0.3021 ±0.0046	0.5000 ±0.0493
GPT-NEO-125M	0.6700 ±0.0473	0.5718 ±0.0071	0.2219 ±0.0041	0.3654 ±0.0474
GPT-NEO-1.3B	0.6700 ±0.0473	0.5223 ±0.0072	0.2212 ±0.0041	0.3654 ±0.0474
GPT-NEO-2.7B	0.7900 ±0.0409	0.5547 ±0.0071	0.2101 ±0.004	0.3654 ±.0474
OPT-125M	0.6900 ±0.0465	0.5615 ±0.0071	0.3132 ±0.0046	0.3654 ±0.0474
OPT-350M	0.6900 ±0.0465	0.5501 ±0.0071	0.2739 ±0.0044	0.3654 ±0.0474
OPT-1.3B	0.8100 ±0.0394	0.5386 ±0.0072	0.1975 ±0.0040	0.3750 ±0.0477
OPT-2.7B	0.7700 ±0.0423	0.5708 ±0.0071	0.2534 ±0.0043	0.6346 ±0.0474
OPT-6.7B	0.8100 ±0.0394	0.5714 ±0.0071	0.2388 ±0.0042	0.4231 ±0.0487
OLMo-1B	0.8200 ±0.0386	0.4973 ±0.0072	0.2949 ±0.0045	0.6250 ±0.0477
OLMo-7B	0.8500 ±0.0359	0.5693 ±0.0071	0.3054 ±0.0046	0.3750 ±0.0477
OpenLLaMA-3B	0.8500 ±0.0359	0.5070 ±0.0072	0.2934 ±0.0045	0.6250 ±0.0477
OpenLLaMA-7B	0.8500 ±0.0359	0.5472 ±0.0071	0.3051 ±0.0046	0.3750 ±0.0477
Mistral-7B	0.9200 ±0.0273	0.3375 ±0.0068	0.2771 ±0.0044	0.6250 ±0.0477

Table 32: The experimental results (with standard deviation) of various baseline LLMs of different model sizes on the SuperGLUE (Wang et al., 2019a) benchmark. The evaluation metrics of each task are described in Table 13. Here, no RAG methods are applied.

Model	WSC273	WinoGrande	ANLI r1	ANLI r2	ANLI r3
GPT-1 (117M)	0.6154 ±0.0295	0.5272 ±0.0140	0.3340 ±0.0149	0.3080 ±0.0146	0.3500 ±0.0138
GPT-2-Small (124M)	0.5641 ±0.0301	0.5185 ±0.0140	0.3400 ±0.0150	0.3400 ±0.0150	0.3475 ±0.0138
GPT-2-Medium (355M)	0.6081 ±0.0296	0.5257 ±0.0140	0.3320 ±0.0149	0.3300 ±0.0149	0.3500 ±0.0138
GPT-2-Large (774M)	0.6300 ±0.0293	0.5517 ±0.0140	0.3230 ±0.0148	0.3310 ±0.0149	0.3333 ±0.0136
GPT-2-XL (1.6B)	0.6593 ±0.0287	0.5833 ±0.0139	0.3370 ±0.0150	0.3510 ±0.0151	0.3600 ±0.0139
GPT-NEO-125M	0.5531 ±0.0301	0.5051 ±0.0141	0.3320 ±0.0149	0.3410 ±0.0150	0.3392 ±0.0137
GPT-NEO-1.3B	0.7179 ±0.0273	0.5533 ±0.0140	0.3250 ±0.0148	0.3300 ±0.0149	0.3400 ±0.0137
GPT-NEO-2.7B	0.7326 ±0.0268	0.5746 ±0.0139	0.3290 ±0.0149	0.3400 ±0.0150	0.3542 ±0.0138
OPT-125M	0.5568 ±0.0301	0.5043 ±0.0141	0.3620 ±0.0152	0.3710 ±0.0153	0.3575 ±0.0138
OPT-350M	0.6447 ±0.0290	0.5257 ±0.0140	0.3110 ±0.0146	0.3380 ±0.0150	0.3417 ±0.0137
OPT-1.3B	0.7326 ±0.0268	0.5959 ±0.0138	0.3410 ±0.0150	0.3390 ±0.0150	0.3375 ±0.0137
OPT-2.7B	0.7802 ±0.0251	0.6109 ±0.0137	0.3380 ±0.0150	0.3370 ±0.0150	0.3425 ±0.0137
OPT-6.7B	0.8168 ±0.0235	0.6527 ±0.0134	0.3090 ±0.0146	0.3440 ±0.0150	0.3425 ±0.0137
OLMo-1B	0.7363 ±0.0267	0.6014 ±0.0138	0.3050 ±0.0146	0.3500 ±0.0151	0.3492 ±0.0138
OLMo-7B	0.8462 ±0.0219	0.6630 ±0.0133	0.3320 ±0.0149	0.3590 ±0.0152	0.3600 ±0.0139
OpenLLaMA-3B	0.8315 ±0.0227	0.6188 ±0.0137	0.3230 ±0.0148	0.2980 ±0.0145	0.3425 ±0.0137
OpenLLaMA-7B	0.8242 ±0.0231	0.6661 ±0.0133	0.3130 ±0.0147	0.3520 ±0.0151	0.3675 ±0.0139
Mistral-7B	0.8791 ±0.0198	0.7403 ±0.0123	0.4820 ±0.0158	0.4640 ±0.0158	0.4700 ±0.0144

Model	ARC Easy	ARC Challenge	PIQA	SWAG	HellaSwag
GPT-1 (117M)	0.3670 ±0.0099	0.3527 ±0.0098	0.5881 ±0.0115	0.4583 ±0.0035	0.2497 ±0.0043
GPT-2-Small (124M)	0.4360 ±0.0102	0.1911 ±0.0115	0.6295 ±0.0113	0.4057 ±0.0035	0.2895 ±0.0045
GPT-2-Medium (355M)	0.4924 ±0.0103	0.2167 ±0.0120	0.6752 ±0.0109	0.4547 ±0.0035	0.3332 ±0.0047
GPT-2-Large (774M)	0.5316 ±0.0102	0.2176 ±0.0121	0.7040 ±0.0107	0.4721 ±0.0035	0.3641 ±0.0048
GPT-2-XL (1.6B)	0.5825 ±0.0101	0.2500 ±0.0127	0.7078 ±0.0106	0.4930 ±0.0035	0.4002 ±0.0049
GPT-NEO-125M	0.4377 ±0.0102	0.1903 ±0.0115	0.6300 ±0.0113	0.4051 ±0.0035	0.2866 ±0.0045
GPT-NEO-1.3B	0.5623 ±0.0102	0.2304 ±0.0123	0.7116 ±0.0106	0.4953 ±0.0035	0.3865 ±0.0049
GPT-NEO-2.7B	0.6107 ±0.0100	0.2765 ±0.0131	0.7214 ±0.0105	0.5177 ±0.0035	0.4272 ±0.0049
OPT-125M	0.4352 ±0.0102	0.1903 ±0.0115	0.6284 ±0.0113	0.4109 ±0.0035	0.2919 ±0.0045
OPT-350M	0.4411 ±0.0102	0.2082 ±0.0119	0.6464 ±0.0112	0.4424 ±0.0035	0.3201 ±0.0047
OPT-1.3B	0.5711 ±0.0102	0.2346 ±0.0124	0.7165 ±0.0105	0.5052 ±0.0035	0.4152 ±0.0049
OPT-2.7B	0.6077 ±0.0100	0.2679 ±0.0129	0.7383 ±0.0103	0.5241 ±0.0035	0.4586 ±0.0050
OPT-6.7B	0.6561 ±0.0097	0.3063 ±0.0135	0.7628 ±0.0099	0.5446 ±0.0035	0.5052 ±0.0050
OLMo-1B	0.6334 ±0.0099	0.2867 ±0.0132	0.7503 ±0.0101	0.5111 ±0.0035	0.4694 ±0.0050
OLMo-7B	0.7340 ±0.0091	0.3686 ±0.0141	0.7884 ±0.0095	0.5508 ±0.0035	0.5563 ±0.0050
OpenLLaMA-3B	0.6928 ±0.0095	0.3404 ±0.0138	0.7503 ±0.0101	0.5367 ±0.0035	0.4884 ±0.0050
OpenLLaMA-7B	0.7117 ±0.0093	0.3754 ±0.0142	0.7568 ±0.0100	0.5498 ±0.0035	0.5256 ±0.0050
Mistral-7B	0.8140 ±0.0080	0.5410 ±0.0146	0.8020 ±0.0093	0.5973 ±0.0035	0.6601 ±0.0047

Table 33: The experimental results (with standard deviation) of various baseline LLMs of different model sizes on other commonsense reasoning datasets. The evaluation metrics of each task are described in Table 13. Here, no RAG methods are applied.