Toward Verifiable Instruction-Following Alignment for Retrieval-Augmented Generation

Guanting Dong[1], **Xiaoshuai Song**[2], **Yutao Zhu**[1], **Runqi Qiao**[2], **Zhicheng Dou**[1*], **Ji-Rong Wen**[1]

[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]School of Artificial Intelligence, Beijing University of Posts and Telecommunications
{dongguanting, dou}@ruc.edu.cn

Abstract

Following natural instructions is crucial for the effective application of Retrieval-Augmented Generation (RAG) systems. Despite recent advancements in Large Language Models (LLMs), research on assessing and improving instruction-following (IF) alignment within the RAG domain remains limited. To address this issue, we propose VIF-RAG, an automated, scalable, and verifiable synthetic pipeline for instruction-following alignment in RAG systems. We start by manually crafting a minimal set of atomic instructions (<100) and developing combination rules to synthesize and verify complex instructions for a seed set. We then use supervised models for instruction rewriting while simultaneously generating code to automate the verification of instruction quality via a Python executor. Finally, we integrate these instructions with extensive RAG and general samples, scaling up to a high-quality VIF-RAG-QA dataset (>100k) through automated processes. To further bridge the gap in instruction-following auto-evaluation for RAG systems, we introduce FollowRAG Benchmark, which includes approximately 3K test samples, covering 22 categories of general instruction constraints and four knowledge-intensive QA datasets. Due to its robust pipeline design, FollowRAG can seamlessly integrate with different RAG benchmarks. Using FollowRAG and eight widely-used IF and foundational abilities benchmarks for LLMs, we demonstrate that VIF-RAG markedly enhances LLM performance across a broad range of general instruction constraints while effectively leveraging its capabilities in RAG scenarios. Further analysis offers practical insights for achieving IF alignment in RAG systems.

Code — https://github.com/dongguanting/FollowRAG
Extended version — https://arxiv.org/pdf/2410.09584

Introduction

The advancement of Large Language Models (LLMs) (OpenAI 2023; Yang et al. 2024) has profoundly revolutionized a variety of real-world tasks expressed in natural language (Wei et al. 2022). However, they still suffer from hallucinations and factual inconsistencies (Bang et al. 2023), impacting the authenticity of generated answers. Retrieval-Augmented Generation (RAG) has gained recognition as a promising solution, empowering LLMs to leverage reliable information from retrieved documents, thereby returning high-quality responses (Lewis et al. 2020).

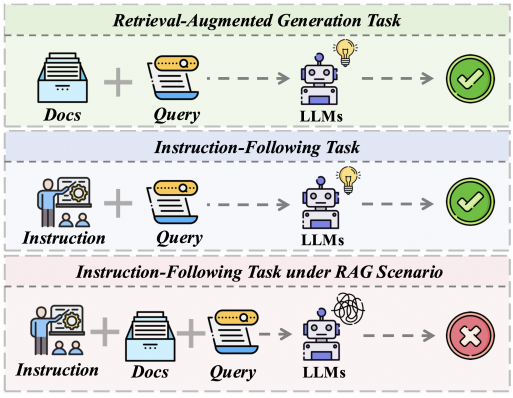

Figure 1: The instruction-following tasks for LLMs in RAG.

In real-world interaction scenarios, users often deviate from standard templates when posing questions, instead of imposing diverse instructions on model outputs to meet specific task requirements (Chung et al. 2024; Dong et al. 2022). Consequently, improving instruction-following (IF) capabilities is foundational to the effective application of LLM and RAG systems. The core goal of IF is to enable models to adapt to the diverse intents of users, which has garnered widespread attention in the LLM community.

Existing efforts on instruction-following alignment primarily focus on multi-grained evaluation and high-quality instruction data synthesis to enhance LLMs' natural instruction-following capabilities (Zhou et al. 2023; Wen et al. 2024). However, in complex RAG scenarios, the diverse knowledge introduced by retrieval-augmented techniques presents significant challenges for LLMs in effectively handling complex instructions (Figure 1). After supervised fine-tuning on high-quality general and knowledge-intensive QA datasets, LLMs demonstrate robust performance in both IF and RAG tasks. However, these capabilities do not always generalize well to instruction-following tasks under RAG scenarios and may even conflict with the performance of other fundamental abilities (Dong et al. 2024b). Unfortunately, research on instruction-following in

*Corresponding author

RAG systems remains limited, significantly hindering their application in real-world interactions. To tackle these challenges, our aim is to address following critical research questions:

- ***RQ1.*** *How can we comprehensively evaluate the complex instruction-following capabilities in the RAG scenario?*
- ***RQ2.*** *How can we achieve scalable and reliable instruction-following alignment in RAG systems while preserving the it's foundational abilities from conflict?*

In this paper, we propose VIF-RAG, the first automated, scalable, and reliable data synthesis pipeline for achieving complex instruction-following alignment in RAG scenarios. The core insight of VIF-RAG is to ensure every step of data augmentation and combination includes a proper verification process. Specifically, we start by manually crafting a minimal set of atomic instructions ($<$100) and developing combination rules to synthesize and verify complex instructions for a seed set. We then use supervised models for instruction rewriting. Motivated by tool execution studies (Le et al. 2022), we employ the same supervised model to generate verification code and automatically verify the quality of augmented instructions through the Python compiler's outputs. Finally, we combine these high-quality instructions with RAG datasets from various domains (each containing retrieved documents per query), performing the augmentation and dual validation process to synthesize a high-quality instruction-based RAG dataset, named VIF-RAG-QA ($>$100K samples).

To further bridge the gap in automatic instruction-following evaluation for RAG systems, we introduce FollowRAG, the first benchmark dedicated to comprehensively assessing the complex IF capabilities of RAG systems. FollowRAG aggregates constraints from real-world scenarios. It includes approximately 3K test samples, spanning 4 knowledge-intensive QA benchmarks and 22 types of constraints. Due to its robust pipeline design, FollowRAG can seamlessly integrate with different RAG benchmarks.

To summarize, our contributions are as follows:

- To first achieve instruction-following alignment in the RAG system, we propose VIF-RAG, an automated, scalable, and verifiable data synthetic framework. VIF-RAG uniquely combines augmented rewriting with diverse validation processes to synthesize high-quality instruction-following alignment data from almost scratch ($<$100), scaling up to over 100K samples.
- We introduce FollowRAG, the first benchmark designed to comprehensively evaluate LLM's complex instruction-following abilities in RAG tasks. FollowRAG includes nearly 3K test samples, spanning four knowledge-intensive QA benchmarks and 22 types of constraints. Its design ensures seamless integration with various RAG benchmarks, providing strong scalability.
- With FollowRAG and 8 widely-used IF and 3 foundational benchmarks, we demonstrate that different LLMs with VIF-RAG achieve extraordinary alignment on instruction following in both RAG and standard scenarios while effectively preserving foundational capabilities.

Related Work

Instruction-Following Alignment for LLMs. Instruction-following ability is a core capability of large language models. Existing works fall into two main categories. The first includes efforts (Hendrycks et al. 2021; Zheng et al. 2024), which rigorously evaluate models' adherence to general instructions. Moreover, previous works focus on fine-grained assessment under specific constraints, using stricter criteria such as instruction difficulty, domain, and task formats (Qin et al. 2024). The other category focuses on improving IF alignment. Manual design of instructions and responses by human annotators (Wei et al. 2021) is challenging and costly. To address this, methods are developed to synthesize diverse instructions, allowing weaker models to mimic the responses of advanced models (Dong et al. 2024a), achieving strong-to-weak alignment (Cao et al. 2024).

Alignment for Retrieval-Augmented Generation. Retrieval-Augmented Generation (RAG) addresses the issue of knowledge hallucination in LLMs by retrieving relevant factual information, offering a promising solution (Guu et al. 2020; Lewis et al. 2020). However, efficiently aligning retrieved knowledge with LLMs' preferences remains a challenge. Researchers have developed robust reranker-based methods (Sun et al. 2023) and data filtering approaches (Wang et al. 2023) to reduce noisy information and bridge this gap. Additionally, approaches like RePLUG (Shi et al. 2023) integrate LLMs' preferences into training objectives to improve alignment. Query rewriting methods (Ma et al. 2023; Dong et al. 2023b) attempt to adjust inputs based on these preferences. Furthermore, SelfRAG (Asai et al. 2024) use multi-round retrieval and generation to refine outputs and achieve better alignment. Despite these advancements, the diverse knowledge introduced by retrieval-augmented techniques poses significant challenges for LLMs in handling complex instructions.

Preliminaries

Retrieval-Augmented Generation (RAG). Retrieval-Augmented Generation systems usually operates under a *retrieve-then-read* framework (Lewis et al. 2020). The external retriever is integrated to gather supporting knowledge and improve the generation process. Given a query q, a retriever R recalls k relevant documents $D_q = \{d_i\}_{i=1}^{k}$ from an external corpus comprised of N documents. We employ the DPR (Karpukhin et al. 2020) to obtain hidden vectors for queries and documents. The relevance score is determined by measuring the dot-product similarity between the query and document representations, allowing the retrieval of the top-k documents D_q:

$$D_q = \text{argtop-}k \left[E_{\text{d}}(d_i)^\top \cdot E_{\text{q}}(q) \mid i = \{1 \dots N\} \right]. \quad (1)$$

Then, the retrieved documents are concatenated with the query into an LLM reader R to generate the target text:

$$y = R(q, D_q) = \log P_\theta \left(q, D_q\right), \quad (2)$$

where P_θ is the output probability distribution.

Instruction-following Alignment for RAG. Following instructions is one of the most foundational ability for LLMs

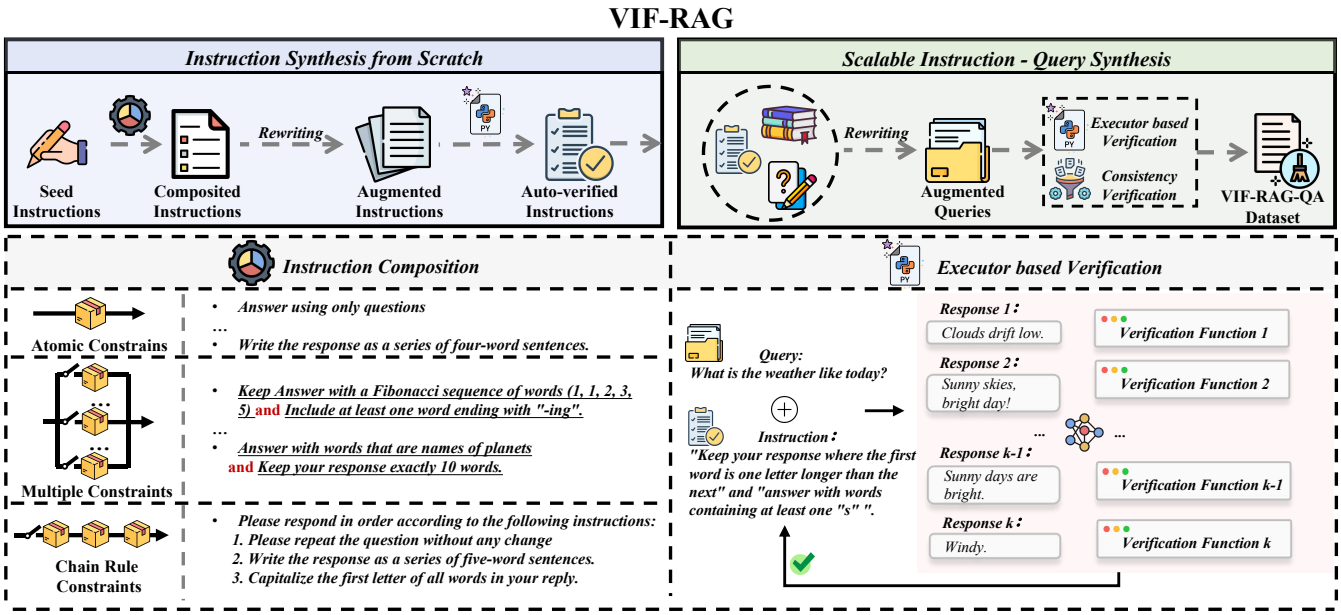

Figure 2: The overall framework of VIF-RAG. The top section presents the pipeline for automated IF data synthesis in RAG scenario, while the bottom section shows examples of 'Instruction Composition' and 'Executor-based Verification,' respectively.

in RAG systems. Given an instruction $I = \{I_j\}_{j=1}^{M}$ with M specific constraints and a specific query q with corresponding relevant k retrieved documents D_q, The LLM π_θ in the RAG system is expected to produce an accurate response $y \sim \pi_\theta(y \mid q, D_q, I)$ while obeying with constraints.

VIF-RAG Framework

In this section, we propose VIF-RAG, which can be broadly split into two sections: (1) the instruction synthesis stage and (2) instruction-query synthesis, scaling from almost scratch (<100) to over 100K high-quality instruction-query samples. Below, we will delve into the specifics.

Instruction Synthesis from Scratch

Handwritten Seed Instructions. We initially develop a minimal seed instruction set $D_{\text{seed}}^{\text{atom}}$ manually, using four foundational categories of constraints: *format constraints, semantic constraints, knowledge constraints, and lexical constraints*, as themes for instruction writing.

We hire only one well-educated human annotator to manually create 15 single-atomic instructions for each type of constraint. Notably, this is the only process in our data synthesis process that includes human supervision.

Instruction Composition & Verification. Real-world instructions often involve multiple constraints in a user query. To address this, we design rules to automatically combine atomic instructions into diverse, complex instructions:

- ***Multiple Constraints***: As illustrated in Figure 2, we randomly sample pairs of instructions from $D_{\text{seed}}^{\text{atom}}$ and insert them into a constraint template. By directly concatenating these instruction pairs, we create complex instructions that contain dual and triple constraints. This type of instruction requires the model to generate results that satisfy multiple constraints simultaneously.
- ***Chain Rule Constraints***: We design sequential conditional constraint templates and selected atomic instructions from $D_{\text{seed}}^{\text{atom}}$ to form chain constraints. Formally, the chain consists of n tasks $\{T_1, T_2, ..., T_n\}$, requiring the model's output to complete these n tasks sequentially.

Verification. Randomly combining atomic instructions can easily lead to conflicts between them (e.g., don't use words that end with '-ing'). These semantic conflicts can be challenging to detect using a simple Natural Language Inference model. To detect potential conflicts between these instructions, we use a robust supervised model that rates their consistency from 1 to 10. Samples scoring below 8 are excluded to refine our high-quality complex instruction set $D_{\text{seed}}^{\text{complex}}$. Ultimately, we arrive at the initial seed instruction set $D_{\text{seed}} = \{D_{\text{seed}}^{\text{atom}} \cup D_{\text{seed}}^{\text{complex}}\}$.

Instruction Rewriting & Quality Verification. To automate the scaling up of instructions, the instruction rewriting strategy is considered the most natural augmentation method, and has received significant attention in the RAG and reasoning fields (Li et al. 2024b,a; Dong et al. 2024c). We use a supervised model[1] to iteratively rewrite instructions from the D_{seed} set in batches of 50 for K rounds, generating an augmented set D_{aug}. Subsequently, we merge the seed and augmented samples to form the combined instruction set $D_{\text{ins}} = D_{\text{seed}} \cup D_{\text{aug}}$, removing duplicates.

Inspired by tool execution works (Le et al. 2022), we aim to leverage the powerful coding abilities of LLMs to assist in verifying the quality of auto-generated instructions. As

[1]For the supervised model, we use GPT-4-turbo-2024-04-09.

shown in Figure 2, for each instruction $I \in D_{\text{ins}}$, we use the supervision model to generate K verification function codes and corresponding test cases $\{func_j^I, c_j^I\}_{j=1}^K \in D_{\text{verify}}$, and assess the instruction's quality by analyzing the output of the executor $\mathcal{E}$. For any function and test case $\{func_j^I, c_j^I\} \in D_{\text{verify}}$, its execution output is:

$$\mathcal{E}(func_j^I, c_j^I) = \begin{cases} 1 & \text{If output is "True".} \\ 0 & \text{If output is "False" or "Error".} \end{cases} \quad (3)$$

Therefore, we can calculate the accuracy Acc_{func} of each verification function based on K test samples, as well as the accuracy Acc_{case} of each case evaluated using K verification functions. These can be formulated as:

$$\begin{cases} Acc_{\text{func}} = \frac{1}{K} \sum_{j=1}^{K} \mathcal{E}(func^I, c_j^I)_{j=1}^K \\ Acc_{\text{case}} = \frac{1}{K} \sum_{j=1}^{K} \mathcal{E}(func_j^I, c^I)_{j=1}^K. \end{cases} \quad (4)$$

Based on the above cross metrics, we require that at least one Acc_{func} and Acc_{case} of the each instruction must exceed 0.5, Ultimately, we obtain the auto-verified instruction set as

$$D_{\text{ins}}^{\text{verify}} = \{d \in D_{\text{ins}} \mid Acc_{\text{func}}(d) > 0.5 \ \& \ Acc_{\text{case}}(d) > 0.5\}. \quad (5)$$

The samples that do not meet the cross metrics are discarded.

Scalable Instruction-Query Synthesis

Random Instruction-Query Combination. In real-world interactions with RAG systems, achieving IF alignment depends on effectively integrating the synthesized instructions with the queries used by the RAG system. To meet this goal, as depicted in Figure 2, we first extract high-quality queries from two different data sources.

1) RAG Domain: To build an effective RAG system, We need to prepare sufficient amounts of QA-format data with relevant knowledge to enhance its knowledge-based interaction capabilities. Consequently, we randomly select a query set Q from mixed QA data sources, including multi-hop and knowledge base QA [2]. Following Lewis et al.; Dong et al., We employ the dense retriever R to fetch the top-K relevant documents D_i for each query $q \in Q$ from an external knowledge base, resulting in the dataset $D_{\text{RAG}} = \{q_i, D_i\}_{i=1}^K$. Furthermore, we randomly select K queries along with their corresponding retrieved documents from D_{RAG} for each instruction I and combine them to create the RAG query set with IF constraints $D_{\text{IF-RAG}} = \{I_j, q_j, D_j\}_{j=1}^K$.

2) General Domain: In addition to incorporating RAG-specific abilities, the RAG system has to possess basic human-aligned abilities to meet users' daily interaction needs. Therefore, ShareGPT (Chiang et al. 2023), which provides authentic multi-turn human dialogue data, is our natural choice. Similar to how we handle the RAG domain, for each instruction $I \in D_{\text{ins}}$, we randomly select K queries from the ShareGPT to combine with the instruction and construct the general dataset $D_{\text{IF-General}}$ for each instruction.

[2]We use the training sets from Natural Questions, TriviaQA, HotpotQA, and WebQuestionsSP as mixed QA sources.

Ultimately, we merge the instruction-constrained query sets from these two domains into the final query set of VIF-RAG-QA, formulated as $D_{\text{VIF-RAG}}^q$.

Instruction-Query Rejection Sampling. It is worth noting that under diverse instruction-following constraints, the original grounding truth answers for queries in both the RAG and general datasets become unreliable. To address this issue and improve synthetic data diversity, we adopt a rejection sampling strategy (Yuan et al. 2023). Specifically, we use the supervision model to generate K responses $y_x = \{y_i\}_{i=1}^K$ for each instruction-query pair $x \in D_{\text{VIF-RAG}}^q$, resulting in $\{x, y_x\} \in D_{\text{VIF-RAG}}$.

Dual Stage Verification. To further ensure comprehensive quality control of the synthetic dataset, we employ a dual stage verification process for the instruction-query data:

- ***Executor-based Verification:*** We leverage pre-existing verification functions to evaluate adherence in the augmented outputs. As in the "Instruction Rewriting & Quality Verification" section, at least one response in $D_{\text{VIF-RAG}}$ must achieve an accuracy rate Acc_{case} above 0.5 across all verification functions; otherwise, the sample is discarded.
- ***Consistency Verification:*** We notice that combined instructions and queries often conflict. A simple example is when the query "Please write a brief biography of Barack Obama." does not meet the instruction "Strictly limit your answer to less than 10 tokens." , Therefore, we employ a supervision model to evaluate the alignment between queries and instructions on a scale of 1 to 10, discarding samples that receive a score below 8.

After dual stage verification, we have automatically obtained a large-scale, high-quality VIF-RAG-QA dataset.

FollowRAG Benchmark

To bridge the gap in automatic instruction-following evaluation for RAG systems, we introduce FollowRAG in this section. We provide a detailed introduction from two aspects: "Data Construction" and "Evaluation and Statistics".

Dataset Construction

Instruction Collection & Extraction. FollowRAG aims to assess the model's ability to follow user instructions in complex multi-document contexts. Drawing from general IF datasets like FollowBench (Jiang et al. 2024), we collect and verify definitions and examples of atomic instructions using rules, excluding those irrelevant to RAG scenarios. Ultimately, we identify 22 types of instruction constraints, encompassing language, length, structure, and keywords.

Instruction Reforming. We use widely-used question-answering (QA) datasets, such as Natural Questions (Kwiatkowski et al. 2019), as the foundation for constructing FollowRAG samples. For a query sampled from the QA datasets, we need to generate a complex instruction containing n atomic instruction constraints (with n ranging from 1 to 4). To enhance the diversity of atomic instruction representations, we employ GPT-4o as the instruction generator. Specifically, given a query, we first sample n instructions from the atomic instruction set and perform conflict

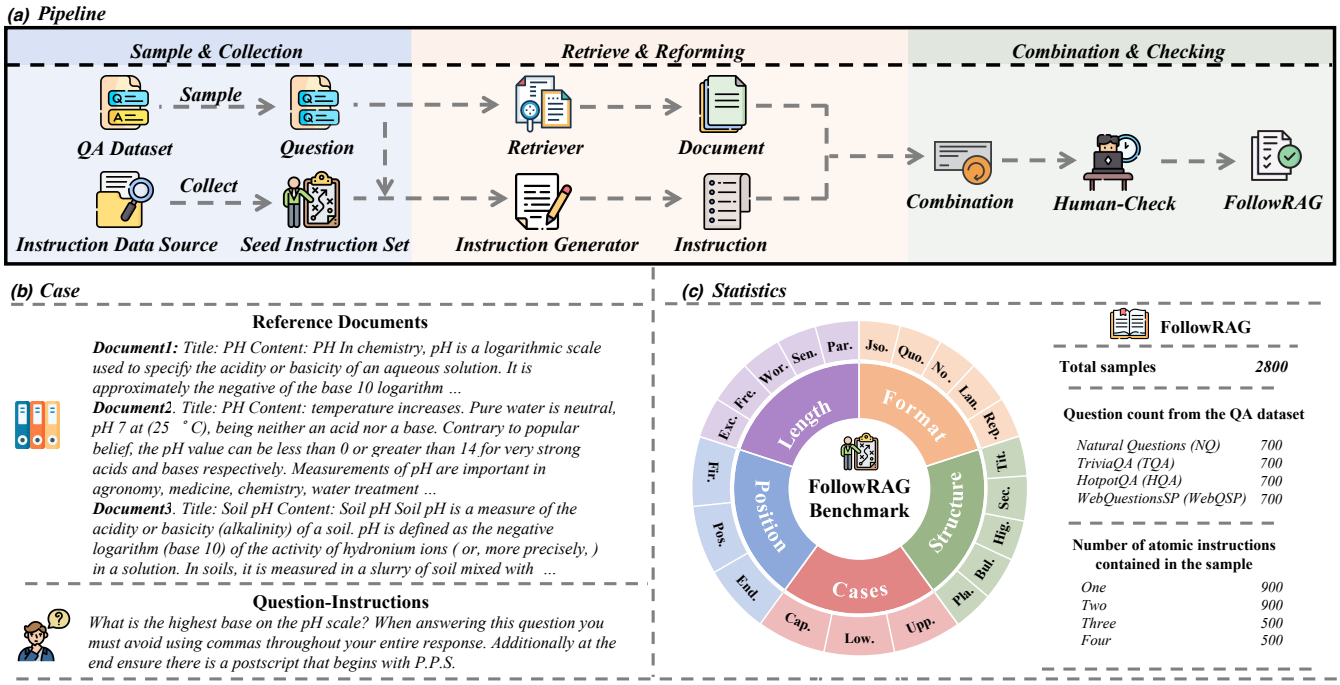

Figure 3: The construction pipeline, diagram and statistics of FollowRAG.

detection. Subsequently, with examples as demonstrations, we prompt the LLM to generate a new varied instruction text for each type of atomic instruction, along with parameters for instruction-following evaluation.

Combination. Finally, we integrate the retrieved passages, query and atomic instructions to construct the sample input for FollowRAG. To avoid mechanically concatenating the query and instructions in a template-based manner, we prompt supervised model to naturally blend the multiple atomic instructions and the query into a coherent instruction-query paragraph. We then add the top-K document passages retrieved based on the query to the instruction-query paragraph, forming the complete sample input.

Evaluation and Statistics

After obtaining the model's output, we evaluate it from two perspectives: instruction following and question answering (QA) under the RAG paradigm:

Instruction Following: Using the verifiable nature of our atomic instructions and following the IFEval approach, we automate the verification of the model's compliance with each instruction by code validation. We then calculate the average pass rate for each atomic instruction across all samples to determine the IF score in FollowRAG.

RAG: Under new instruction constraints, the model's target output differs from the gold answers in the original QA dataset, rendering traditional metrics like Exact-Match ineffective. To address this, we use the original gold answers as a reference and utilize GPT-4o to evaluate whether the model's outputs correctly address the questions. The scoring criteria are as follows: Completely correct (1), Partially correct (0.5), Completely incorrect (0). The average score of all samples is taken as the RAG score for FollowRAG.

For detailed statisticsin in Figure 3, FollowRAG is the first instruction-following evaluation dataset under RAG scenario comprising 2.8K samples, covering 22 fine-grained atomic instructions across 6 categories. The queries in FollowRAG are sourced from 4 QA datasets across 3 types: 1) Open-Domain QA: **Natural Questions (NQ)** (Kwiatkowski et al. 2019) and **TriviaQA (TQA)** (Joshi et al. 2017); 2) Multi-Hop QA: **HotpotQA (HQA)** (Yang et al. 2018); and 3) Knowledge Base QA: **WebQuestionsSP (WebQSP)** (Yih et al. 2016). To further construct varying levels of instruction-following difficulty, FollowRAG includes 0.9K samples of single and dual atomic instructions, as well as 0.5K complex multi-instruction samples containing 3 and 4 atomic instructions, respectively.

Experiment

Experimental Setup

Datasets. We evaluate over 10+ benchmarks to comprehensively evaluate the VIF-RAG. For the instruction-following tasks in RAG scenarios, we use the **FollowRAG** benchmark as mentioned in Section 5, which covering 4 question-answering (QA) datasets. For general IF evaluation, we selected two commonly used complex IF datasets, **IFEval** and **FollowBench**, along with the natural instruction dataset **MT-Bench** (Zheng et al. 2024) and the challenging ChatBot IF bench, **Arena-Hard** (Li et al. 2024c). Additionally, to measure that the foundational abilities of LLMs, we further evaluate two widely used LLM's general abilties evaluation sets, **C-Eval** (Huang et al. 2023) and **MMLU** (Hendrycks

Model	NQ			TQ			HQ			WebQSP			ALL		
	IF	RAG	AVG	IF	RAG	AVG	IF	RAG	AVG	IF	RAG	AVG	IF	RAG	AVG
Llama3-8B-base	3.2	5.7	4.4	4.1	15.9	10.0	3.6	7.3	5.5	10.0	23.1	16.5	5.2	13.0	9.1
Llama3-8B-SFT	15.7	59.5	37.6	15.0	76.5	45.7	15.0	**52.5**	33.8	14.4	70.0	42.2	15.0	64.6	39.8
Llama3-8B-SFT-VIF-RAG	**43.9**	**65.0**	**54.5**	**42.7**	**78.0**	**60.4**	**39.6**	46.0	**42.8**	**42.5**	**70.5**	**56.5**	**42.2**	**64.9**	**53.5**
Mistral-7B-base	25.7	31.1	28.4	25.9	44.4	35.2	26.9	19.9	23.4	24.7	20.4	22.6	25.8	29.0	27.4
Mistral-7B-SFT	21.0	48.5	34.7	17.2	**71.5**	44.3	17.6	**46.5**	32.1	21.7	**66.5**	44.1	19.3	**58.3**	38.8
Mistral-7B-SFT Conifer	29.9	49.5	39.7	30.5	67.0	48.7	26.5	40.0	33.2	31.1	63.0	47.1	29.5	54.9	42.2
Mistral-7B-SFT Evol-Instruct	41.7	41.5	41.6	37.0	63.5	50.4	35.4	35.0	35.2	39.4	54.0	46.7	38.4	48.5	43.5
Mistral-7B-SFT-VIF-RAG	**51.2**	**56.5**	**53.8**	**45.9**	70.5	**58.2**	**44.9**	43.0	**44.0**	**47.8**	58.0	**52.9**	**47.4**	57.0	**52.2**
Deita-7B-V1.0-SFT	31.4	31.5	31.4	29.0	42.5	35.8	26.5	30.5	28.5	26.3	40.0	33.2	28.3	36.1	32.2
Qwen1.5-7B-base	27.7	34.4	31.0	27.7	45.9	36.8	27.5	19.8	23.6	29.9	45.8	37.9	28.2	36.5	32.3
Qwen1.5-7B-SFT	16.1	**50.5**	33.3	14.3	70.0	42.2	14.8	40.0	27.4	13.7	59.0	36.3	14.7	54.9	34.8
Qwen1.5-7B-SFT-VIF-RAG	**38.9**	41.5	**40.2**	**35.8**	**78.0**	**56.9**	**38.1**	**45.0**	**41.6**	**31.9**	**60.0**	**45.9**	**36.2**	**56.1**	**46.2**
Qwen1.5-14B-base	33.7	38.1	35.9	32.5	54.7	43.6	32.4	26.5	29.5	33.0	48.3	40.7	32.0	41.9	36.9
Qwen1.5-14B-SFT	22.0	**54.5**	38.3	18.7	66.0	42.3	18.8	**41.0**	29.9	19.9	63.0	41.4	19.8	56.1	38.0
Qwen1.5-14B-SFT-VIF-RAG	**42.1**	53.0	**47.6**	**40.1**	**71.0**	**55.5**	**38.8**	39.5	**39.2**	**35.7**	**69.0**	**52.3**	**39.2**	**58.1**	**48.6**

Table 1: The main results on FollowRAG."AVG" represents the weighted average of the corresponding IF and RAG scores. The top two results in each column are highlighted in **bold** and underlined.

Model	IFEval				FollowBench (SSR Avg.)	MT-Bench	Arena-Hard	C-Eval	MMLU	GSM8k	HumanEval (Pass@1)
	Pr (S)	Pr. (L)	Ins. (S)	Ins. (L)							
Llama3-8B-base	24.6	26.1	38.1	39.7	11.6	4.0	0.5	24.2	38.8	0.5	0.6
Llama3-8B-SFT	32.5	34.3	43.3	45.4	33.6	5.6	2.2	35.6	45.2	12.6	3.6
Llama3-8B-SFT-VIF-RAG	**37.0**	**42.7**	**48.8**	**54.2**	**49.2**	**6.2**	**3.2**	**39.6**	**49.6**	**22.9**	**8.0**
Mistral-7B-base	14.6	15.3	25.8	27.0	38.0	3.5	0.6	31.8	44.5	**16.0**	25.6
Mistral-7B-SFT	23.3	24.6	38.4	45.7	42.9	6.2	3.1	26.2	32.1	7.3	13.9
Mistral-7B-SFT-VIF-RAG	**34.6**	**41.0**	**46.3**	**52.0**	**53.4**	**6.5**	**3.6**	**33.0**	**49.6**	**16.0**	**32.9**
Qwen1.5-7B-base	25.1	27.9	37.8	40.6	38.7	5.4	3.2	72.8	58.3	50.6	36.0
Qwen1.5-7B-SFT	36.4	39.3	46.4	49.4	46.3	5.7	2.1	69.1	55.5	48.6	39.0
Qwen1.5-7B-SFT-VIF-RAG	**42.3**	**46.0**	**53.5**	**57.1**	**51.1**	**6.1**	**3.9**	**75.6**	**61.2**	**61.4**	**44.5**
Qwen1.5-14B-base	35.5	39.0	46.7	50.2	45.5	5.8	6.4	77.8	64.7	71.8	**59.1**
Qwen1.5-14B-SFT	38.4	41.7	49.4	52.6	49.8	6.0	6.5	76.2	62.0	71.5	58.5
Qwen1.5-14B-SFT-VIF-RAG	**46.3**	**49.9**	**60.0**	**62.2**	**56.3**	**7.3**	**7.0**	**79.5**	**66.5**	**73.8**	**59.1**

Table 2: The cross validation on 4 general instruction-following (Left 4) and 4 foundational abilities (Right 4). Pr. and Ins. refer to the prompt level and instruction level metric, respectively. S or L denote the strict or loose metrics used in IFEval.

et al. 2021), as well as the mathematical reasoning dataset **GSM8K** (Cobbe et al. 2021) and the code evaluation bench **HumanEval** (Chen et al. 2021).

Baselines. We select Mistral-7B (Jiang et al. 2023), Llama3-8B (Meta 2024), Qwen1.5-7B, and Qwen1.5-14B (Yang et al. 2024) as our backbone models, fine-tuning ShareGPT and four QA training sets as SFT version. Besides, we introduce several strong IF baselines, including Conifer (Sun et al. 2024), Evol-Instruct (Xu et al. 2023), and Deita (Liu et al. 2024). To ensure fairness, we add an equal-sized RAG training set to the original synthetic data.

Main Result

Our primary findings are presented in Table 1. Overall, VIF-RAG consistently surpasses all baselines in FollowRAG across multiple setups, highlighting the advantages of our method. Additionally, we have several key insights:

1) Existing IF baselines struggle in complex RAG scenarios. Comparisons between different base models and SFT versions in Tables 1 & 2 show that while SFT general data like ShareGPT improves performance on IFEval, it actually shows a performance decline in the IF aspect of FollowRAG (e.g., NQ-IF: 25.7→21.0 in Mistral). Moreover, several strong IF baselines, such as Conifer, also perform poorly in FollowRAG's IF aspect (HQ-IF: 26.9→26.45). This corroborates the issue highlighted in the introduction: traditional synthetic data may improve LLMs' vanilla IF ability but often fails to generalize in RAG scenarios, sometimes even leading to decreased performance.

2) VIF-RAG shows exceptional IF alignment capability across various datasets, models, and parameter sizes. It consistently outperforms all baselines by over 10% on average accuracy, including a 44% improvement over Llama3-base, showcasing the significant performance advantage of our method. On four detailed QA benchmarks, VIF-RAG achieves the best results across all tested backbones. Moreover, whether using Qwen1.5-7B or Qwen1.5-14B, our method maintains a stable and significant performance increase of over 10%. These results highlight that VIF-RAG is not only plug-and-play but also exhibits strong generalization capabilities.

3) The RAG capability is effectively preserved. Protecting RAG capability is a core focus of RAG systems. Compared to various SFT version baselines, our VIF-RAG sig-

Model	FollowRAG (NQ)		IFEval	
	IF	RAG	Ins(L)	Prompt(L)
Mistral-7B-SFT-VIF-RAG	51.6	56.5	41.0	52.0
w/o Multiple Constraints	46.5 (-5.1)	52.3 (-4.2)	37.9 (-3.1)	48.6 (-3.4)
w/o Chain rule Constraints	49.2 (-2.4)	53.3 (-3.2)	39.2 (-1.8)	49.9 (-2.1)
w/o Executor based Verification	43.5 (-8.1)	56.1 (-0.4)	33.2 (-7.8)	47.6 (-4.4)
w/o Consistency Verification	47.6 (-4.0)	46.2 (-10.3)	38.4 (-2.6)	46.5 (-5.5)

Table 3: Ablation study on different designs of VIF-RAG.

nificantly enhances IF capability while maintaining more stable RAG performance. This allows us to be optimistic about its potential in real-world RAG system applications.

Cross-Domain Validation

To explore the transferability of VIF-RAG, we conduct cross-domain validation on four natural instruction-following datasets and four foundational abilities benchmarks for LLMs in Tabel 2. Our findings are as follows:

1) Consistent IF alignment in both standard and RAG scenarios. Table 1 shows that VIF-RAG achieves remarkable IF alignment in RAG scenarios. In Table 2, comparing Llama3-8B SFT version, VIF-RAG demonstrates strong gains on two widely-used IF benchmarks, IFEval and FollowBench, with improvements of 8.8% (Ins.L) and 15.5% respectively. It also maintains stable improvement across different parameter sizes (7B & 14B). These results confirm that VIF-RAG consistently enhances IF alignment in both RAG and standard scenarios.

2) Robust General IF Transferability. To assess general IF alignment, we test VIF-RAG on challenging benchmarks Arena-Hard and MT-Bench. The results demonstrate that VIF-RAG maintains consistent alignment across various backbones (1.3% improvement on MT-Bench for the 14B model). This reveals significant potential for larger models to achieve better alignment of natural instruction.

3) Great Preservation of foundational Abilities. Previous research highlights that enhancing specific capabilities often compromises others (Dong et al. 2024b). As indicated in Table 2, VIF-RAG effectively preserves general capabilities (MMLU, C-Eval), math reasoning (GSM8K), and coding skills (HumanEval) across different setups, with some slight improvements. This preservation is largely attributed to the integration of ShareGPT data in the synthesis process, demonstrating VIF-RAG's ability to balance diverse capabilities while maintaining broad applicability.

Quantitative Analysis

Ablation Study. To examine the effects of various components in VIF-RAG, we conduct an ablation study in Table 3. The term "w/o" indicates versions where specific components are removed. Our key observations are: (1) Removing any component from VIF-RAG results in decreased performance, indicating that all components, such as the complex instruction composition strategy and quality verification design, are crucial to its effectiveness. (2) The largest

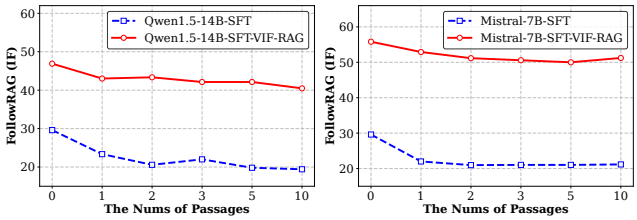

Figure 4: The scaling analysis of retrieved document count.

performance decline in FollowRAG is observed when executor verification is removed. This underscores the critical role of automated instruction-response validation in improving synthetic data quality and confirms the advantage of using LLMs to oversee instruction-following abilities through other core skills like coding. (3) The consistency verification proves beneficial in preserving RAG capabilities. It effectively filters out samples with high-level semantic conflicts between instructions and queries, reducing noise in IF tasks and maintaining RAG performance integrity.

Scaling Analysis. To explore the impact of retrieved document quantity on IF performance in RAG scenarios, we refer to Table 4. For the baseline models (SFT versions), IF capability declines as the number of passages increases. Specifically, performance drops sharply by over 6% when the document quantity in FollowRAG increases from 0 to 1. Further increasing the number to 10 leads to a significant performance decline, with Qwen-14B-SFT experiencing a drop of over 10%. This indicates that integrating knowledge through retrieval-augmented techniques challenges the IF abilities of existing models. In contrast, VIF-RAG shows a minor performance drop ($<$3%) when encountering the first document. As the number of documents increases to 10, VIF-RAG's performance remains relatively stable.

Conclusion

In this paper, we propose VIF-RAG, the first automated, scalable, and verifiable data synthesis pipeline for aligning complex instruction-following in RAG scenarios. VIF-RAG integrates a verification process at each step of data augmentation and combination. We begin by manually creating a minimal instruction set ($<$100) and then apply steps including instruction composition, quality verification, instruction-query combination, and dual-stage verification to generate a large-scale, high-quality VIF-RAG-QA dataset ($>$100K). To address gaps in IF evaluation for RAG systems, we present FollowRAG, featuring around 3K samples with 22 types of complex instruction constraints. Experiments show that VIF-RAG offers insights for optimizing IF alignment.

Acknowledgments

This work was supported by Beijing Natural Science Foundation L233008, and Beijing Municipal Science and Technology Project No. Z231100010323009, National Natural Science Foundation of China No. 62272467. The work was partially done at the Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE.

References

Asai, A.; Wu, Z.; Wang, Y.; Sil, A.; and Hajishirzi, H. 2024. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In *The Twelfth International Conference on Learning Representations*.

Bang, Y.; Cahyawijaya, S.; Lee, N.; Dai, W.; Su, D.; Wilie, B.; Lovenia, H.; Ji, Z.; Yu, T.; Chung, W.; et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Cao, B.; Lu, K.; Lu, X.; Chen, J.; Ren, M.; Xiang, H.; Liu, P.; Lu, Y.; He, B.; Han, X.; et al. 2024. Towards Scalable Automated Alignment of LLMs: A Survey. *arXiv preprint arXiv:2406.01252*.

Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; Pinto, H. P. D. O.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; Stoica, I.; and Xing, E. P. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality.

Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70): 1–53.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. *CoRR*, abs/2110.14168.

Dong, G.; Guo, D.; Wang, L.; Li, X.; Wang, Z.; Zeng, C.; He, K.; Zhao, J.; Lei, H.; Cui, X.; Huang, Y.; Feng, J.; and Xu, W. 2022. PSSAT: A Perturbed Semantic Structure Awareness Transferring Method for Perturbation-Robust Slot Filling. In Calzolari, N.; Huang, C.; Kim, H.; Pustejovsky, J.; Wanner, L.; Choi, K.; Ryu, P.; Chen, H.; Donatelli, L.; Ji, H.; Kurohashi, S.; Paggio, P.; Xue, N.; Kim, S.; Hahm, Y.; He, Z.; Lee, T. K.; Santus, E.; Bond, F.; and Na, S., eds., *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, 5327–5334. International Committee on Computational Linguistics.

Dong, G.; Li, R.; Wang, S.; Zhang, Y.; Xian, Y.; and Xu, W. 2023a. Bridging the KB-Text Gap: Leveraging Structured Knowledge-aware Pre-training for KBQA. In Frommholz, I.; Hopfgartner, F.; Lee, M.; Oakes, M.; Lalmas, M.; Zhang, M.; and Santos, R. L. T., eds., *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*, 3854–3859. ACM.

Dong, G.; Lu, K.; Li, C.; Xia, T.; Yu, B.; Zhou, C.; and Zhou, J. 2024a. Self-play with Execution Feedback: Improving Instruction-following Capabilities of Large Language Models. *CoRR*, abs/2406.13542.

Dong, G.; Wang, Z.; Wang, L.; Guo, D.; Fu, D.; Wu, Y.; Zeng, C.; Li, X.; Hui, T.; He, K.; Cui, X.; Gao, Q.; and Xu, W. 2023b. A Prototypical Semantic Decoupling Method via Joint Contrastive Learning for Few-Shot Named Entity Recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2023, Rhodes Island, Greece, June 4-10, 2023*, 1–5. IEEE.

Dong, G.; Yuan, H.; Lu, K.; Li, C.; Xue, M.; Liu, D.; Wang, W.; Yuan, Z.; Zhou, C.; and Zhou, J. 2024b. How Abilities in Large Language Models are Affected by Supervised Fine-tuning Data Composition. In Ku, L.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, 177–198. Association for Computational Linguistics.

Dong, G.; Zhu, Y.; Zhang, C.; Wang, Z.; Dou, Z.; and Wen, J. 2024c. Understand What LLM Needs: Dual Preference Alignment for Retrieval-Augmented Generation. *CoRR*, abs/2406.18676.

Guu, K.; Lee, K.; Tung, Z.; Pasupat, P.; and Chang, M. 2020. REALM: Retrieval-Augmented Language Model Pre-Training. *CoRR*, abs/2002.08909.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2021. Measuring Massive Multitask Language Understanding. arXiv:2009.03300.

Huang, Y.; Bai, Y.; Zhu, Z.; Zhang, J.; Zhang, J.; Su, T.; Liu, J.; Lv, C.; Zhang, Y.; Lei, J.; Fu, Y.; Sun, M.; and He, J. 2023. C-Eval: A Multi-Level Multi-Discipline Chinese Evaluation Suite for Foundation Models. arXiv:2305.08322.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; de Las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; Lavaud, L. R.; Lachaux, M.; Stock, P.; Scao, T. L.; Lavril, T.; Wang, T.; Lacroix, T.; and Sayed, W. E. 2023. Mistral 7B. *CoRR*, abs/2310.06825.

Jiang, Y.; Wang, Y.; Zeng, X.; Zhong, W.; Li, L.; Mi, F.; Shang, L.; Jiang, X.; Liu, Q.; and Wang, W. 2024. FollowBench: A Multi-level Fine-grained Constraints Following Benchmark for Large Language Models. In Ku, L.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, 4667–4688. Association for Computational Linguistics.

Joshi, M.; Choi, E.; Weld, D. S.; and Zettlemoyer, L. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In Barzilay, R.; and Kan, M., eds., *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 1601–1611. Association for Computational Linguistics.

Karpukhin, V.; Oğuz, B.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; and tau Yih, W. 2020. Dense Passage Retrieval for Open-Domain Question Answering. arXiv:2004.04906.

Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A. P.; Alberti, C.; Epstein, D.; Polosukhin, I.; Devlin, J.; Lee, K.; Toutanova, K.; Jones, L.; Kelcey, M.; Chang, M.; Dai, A. M.; Uszkoreit, J.; Le, Q.; and Petrov, S. 2019. Natural Questions: a Benchmark for Question Answering Research. *Trans. Assoc. Comput. Linguistics*, 7: 452–466.

Le, H.; Wang, Y.; Gotmare, A. D.; Savarese, S.; and Hoi, S. C. H. 2022. CodeRL: Mastering Code Generation through Pretrained Models and Deep Reinforcement Learning. arXiv:2207.01780.

Lewis, P. S. H.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.; Rocktäschel, T.; Riedel, S.; and Kiela, D. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Li, C.; Dong, G.; Xue, M.; Peng, R.; Wang, X.; and Liu, D. 2024a. DotaMath: Decomposition of Thought with Code Assistance and Self-correction for Mathematical Reasoning. *CoRR*, abs/2407.04078.

Li, C.; Yuan, Z.; Yuan, H.; Dong, G.; Lu, K.; Wu, J.; Tan, C.; Wang, X.; and Zhou, C. 2024b. MuggleMath: Assessing the Impact of Query and Response Augmentation on Math Reasoning. In Ku, L.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, 10230–10258. Association for Computational Linguistics.

Li, T.; Chiang, W.-L.; Frick, E.; Dunlap, L.; Wu, T.; Zhu, B.; Gonzalez, J. E.; and Stoica, I. 2024c. From Crowdsourced Data to High-Quality Benchmarks: Arena-Hard and Benchbuilder Pipeline. *arXiv preprint arXiv:2406.11939*.

Liu, W.; Zeng, W.; He, K.; Jiang, Y.; and He, J. 2024. What Makes Good Data for Alignment? A Comprehensive Study of Automatic Data Selection in Instruction Tuning. In *The Twelfth International Conference on Learning Representations*.

Ma, X.; Gong, Y.; He, P.; Zhao, H.; and Duan, N. 2023. Query rewriting for retrieval-augmented large language models. *arXiv preprint arXiv:2305.14283*.

Meta. 2024. Introducing Meta Llama 3: The most capable openly available LLM to date.

OpenAI. 2023. GPT-4 Technical Report. *CoRR*, abs/2303.08774.

Qin, Y.; Song, K.; Hu, Y.; Yao, W.; Cho, S.; Wang, X.; Wu, X.; Liu, F.; Liu, P.; and Yu, D. 2024. Infobench: Evaluating instruction following ability in large language models. *arXiv preprint arXiv:2401.03601*.

Shi, W.; Min, S.; Yasunaga, M.; Seo, M.; James, R.; Lewis, M.; Zettlemoyer, L.; and Yih, W. 2023. REPLUG: Retrieval-Augmented Black-Box Language Models. *CoRR*, abs/2301.12652.

Sun, H.; Liu, L.; Li, J.; Wang, F.; Dong, B.; Lin, R.; and Huang, R. 2024. Conifer: Improving Complex Constrained Instruction-Following Ability of Large Language Models. *CoRR*, abs/2404.02823.

Sun, W.; Yan, L.; Ma, X.; Wang, S.; Ren, P.; Chen, Z.; Yin, D.; and Ren, Z. 2023. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agents. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, 14918–14937. Association for Computational Linguistics.

Wang, Z.; Araki, J.; Jiang, Z.; Parvez, M. R.; and Neubig, G. 2023. Learning to Filter Context for Retrieval-Augmented Generation. *CoRR*, abs/2311.08377.

Wei, J.; Bosma, M.; Zhao, V. Y.; Guu, K.; Yu, A. W.; Lester, B.; Du, N.; Dai, A. M.; and Le, Q. V. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E. H.; Le, Q. V.; and Zhou, D. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *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*.

Wen, B.; Ke, P.; Gu, X.; Wu, L.; Huang, H.; Zhou, J.; Li, W.; Hu, B.; Gao, W.; Xu, J.; et al. 2024. Benchmarking Complex Instruction-Following with Multiple Constraints Composition. *arXiv preprint arXiv:2407.03978*.

Xu, C.; Sun, Q.; Zheng, K.; Geng, X.; Zhao, P.; Feng, J.; Tao, C.; and Jiang, D. 2023. WizardLM: Empowering Large Language Models to Follow Complex Instructions. arXiv:2304.12244.

Yang, A.; Yang, B.; Hui, B.; Zheng, B.; Yu, B.; Zhou, C.; Li, C.; Li, C.; Liu, D.; Huang, F.; et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W. W.; Salakhutdinov, R.; and Manning, C. D. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In Riloff, E.; Chiang, D.; Hockenmaier, J.; and Tsujii, J., eds., *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, 2369–2380. Association for Computational Linguistics.

Yih, W.; Richardson, M.; Meek, C.; Chang, M.; and Suh, J. 2016. The Value of Semantic Parse Labeling for Knowledge Base Question Answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*. The Association for Computer Linguistics.

Yuan, Z.; Yuan, H.; Li, C.; Dong, G.; Tan, C.; and Zhou, C. 2023. Scaling Relationship on Learning Mathematical Reasoning with Large Language Models. *CoRR*, abs/2308.01825.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Zhou, J.; Lu, T.; Mishra, S.; Brahma, S.; Basu, S.; Luan, Y.; Zhou, D.; and Hou, L. 2023. Instruction-Following Evaluation for Large Language Models. arXiv:2311.07911.