Data-Centric Perspectives on Agentic Retrieval-Augmented Generation: A Survey

Anonymous ACL submission

Abstract

Large Language Models (LLMs) excel at natural language understanding and generation, yet their reliance on static pre-training corpora may lead to outdated knowledge, hallucinations, and limited adaptability. Retrieval-Augmented Generation (RAG) mitigates these issues by grounding model outputs with external retrieval, but conventional RAG remains constrained by a fixed retrieve-then-generate routine and struggles with multi-step reasoning and tool calls. **Agentic RAG** addresses these limitations by enabling LLM agents to actively decompose tasks, issue exploratory queries, and refine evidence through iterative retrieval. Despite growing interest, the development of Agentic RAG is impeded by data scarcity: unlike traditional RAG, it requires challenging tasks that require planning, retrieval, and multiple reasoning decisions, and corresponding rich, interactive agent trajectories. This survey presents the first data-centric overview of Agentic RAG, framing its data lifecycle—data collecting, data preprocessing and task formulation, task construction, data for evaluation, and data enhancement for training—and cataloging representative training datasets and benchmarks in different domains (e.g. question answering, web, software engineering). From data perspectives, we aim to guide the creation of scalable, high-quality datasets for the next generation of adaptive, knowledge-seeking LLM agents.

1 Introduction

Large Language Models (LLMs) (Minaee et al., 2024) have greatly advanced AI with strong natural language understanding and generation. Yet their dependence on static pre-training data leads to outdated facts, hallucinations (Huang et al., 2025), and limited adaptability to fast-changing information. **Retrieval-Augmented Generation (RAG)** (Zhao et al., 2024) mitigates these issues by augmenting LLMs with retrieving real-time knowledge from external databases, APIs, or the web to ground generation. Nevertheless, traditional RAG follows a fixed retrieve–then-generate routine and struggles with multi-step reasoning or iterative retrieval.

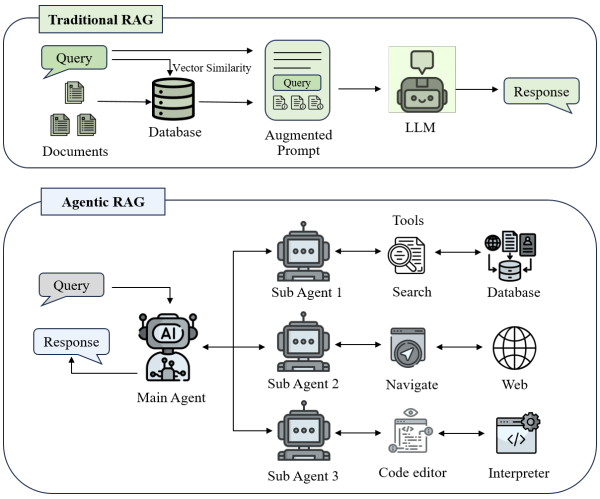

Figure 1: Comparison of traditional and Agentic RAG.

Recent developments in *agentic AI* (Wang et al., 2024a) introduce autonomous LLM-based agents that can plan, reflect, and coordinate tool use. Combining this paradigm with RAG yields **Agentic RAG** (Singh et al., 2025), where agents actively drive retrieval, assess evidence, and refine outputs through iterative interaction. Unlike traditional RAG, these RAG-reasoning agents (Li et al., 2025e) perform *active knowledge seeking*: decomposing tasks, issuing exploratory queries to multiple sub-agents, and looping retrieval until sufficient information is obtained (Figure 1).

Despite growing interest, Agentic RAG development is hindered by *data scarcity*. Unlike traditional RAG—where static corpora suffice—Agentic RAG requires challenging tasks that require planning, retrieval, and multiple reasoning decisions, and corresponding rich, interactive agent trajectories, as shown in Table 1. Such data are costly to annotate, difficult to scale, and

prone to quality issues when automatically synthesized. Therefore, curating scalable and high-quality datasets and benchmarks has been a central problem in the development of Agentic RAG systems.

Stage	Traditional RAG	Agentic RAG
Data Collecting	Static data (e.g. Wikipedia, ArXiv)	Interactive data (e.g. tool/API usage, web navigation)
Task Construction	Basic tasks (single-step, solvable with direct retrieval)	Hard tasks (requiring decomposing, different tools and reasoning)
Evaluation Metrics	Correctness	Multiple axes (e.g. correctness, efficiency, safety)
Data for Training	Chain-of-Thought	Thought-action trajectories, preference pairs, process rewards, new data generated during training for self-improvement

Table 1: Comparison of traditional RAG and Agentic RAG in data lifecycle.

The data curation process in Agentic RAG has two distinctive aspects: (1) **Traditional RAG vs. Agentic RAG:** traditional RAG relies on query–document pairs, whereas Agentic RAG demands rich *agent–environment interaction traces* encoding planning and retrieval actions. (2) **Agentic RAG vs. general agents:** general agents often use tools such as calculators or code interpreters for problem solving, whereas Agentic RAG uses search engines and knowledge bases for *knowledge seeking*. In the former cases, tools provide clear solutions, while in Agentic RAG, tools may actually bring more information for the agent to process.

This survey frames Agentic RAG through a *data lifecycle* (Figure 2) that spans data collecting, data preprocessing and task formulation, task construction, data for evaluation, and data enhancement for training. Specifically, we adopt a *generate-verify-filter/refine pipeline* to analyze the curation process of tasks and trajectories. Our survey makes three primary contributions:

1. We present the first *data-centric* survey of Agentic RAG, systematically reviewing the data lifecycle in Agentic RAG.

2. Following the *generate–verify–filter/refine* pipeline, we summarize representative methods of data curation for both evaluation and training, including task difficulty enhancement and trajectory generation.

3. We catalog Agentic RAG training datasets and benchmarks by domains to facilitate future research.

By foregrounding data as the critical resource, this survey aims to guide researchers and practitioners in building scalable, high-quality datasets that enable the next generation of adaptive, knowledge-seeking LLM agents.

The survey is structured as follows: Section 2 will talk about the whole lifecycle of data in Agentic RAG. Section 3 will talk about representative domain-specific training datasets and benchmarks of Agentic RAG. Section 4 will talk about current challenges and open problems in Agentic RAG.

2 Data Lifecycle in Agentic RAG

2.1 Data Collecting

Data collecting is the first step in the data lifecycle of Agentic RAG. From the perspective of data sources, we distinguish two paradigms: (1) static data and (2) interactive data.

Static Data. Traditional RAG systems rely on well-established corpora where knowledge is either manually curated or automatically extracted from large-scale sources such as Wikipedia, Arxiv, or PubMed. These text-based corpora provide a controlled environment for benchmarking and ensure reproducibility, but they remain static and often fail to capture the evolving nature of real-world knowledge. Over time, these sources have been organized into benchmark datasets: for instance, Natural Questions (NQ) (Kwiatkowski et al., 2019), HotpotQA (Yang et al., 2018), and 2WikiMultiHopQA (Ho et al., 2020) draw evidence from Wikipedia for QA tasks. Beyond QA, domain-specific static benchmarks have been curated from professional environments, such as SWE-bench (Jimenez et al., 2024) (from GitHub repositories and issue–patch pairs) and MLE-bench (Chan et al., 2024) (from Kaggle machine learning competitions). Although these datasets originate from interactive platforms, they are released as fixed benchmarks and thus categorized here as static sources.

Interactive Data. In contrast, Agentic RAG emphasizes data that agents acquire dynamically through interaction with external environments. One approach is API-based retrieval, where models query structured resources such as PubMed, Wikidata (Jin et al., 2025b; Chen et al., 2025a), or live GitHub repositories. Another direction is web navigation, in which autonomous agents browse, click, and parse webpages to gather relevant context (Nakano et al., 2021; Wu et al., 2025a). To approximate complex environments, multi-source or multi-modal evidence may also be collected (Wu

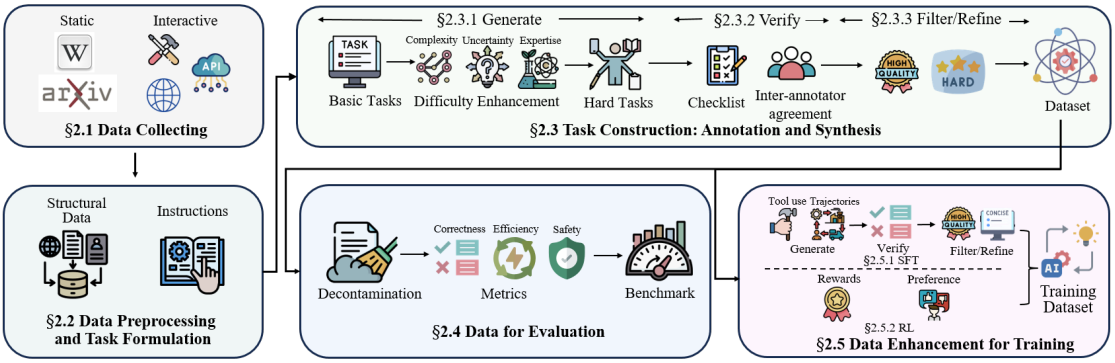

Figure 2: Data lifecycle in Agentic RAG.

et al., 2025c). In addition, tool-augmented retrieval allows agents to execute SQL queries or code to collect task-specific evidence (Zhang et al., 2024a; Li et al., 2025b). Such interactive data is inherently heterogeneous, evolving, and incomplete, thus offering a stronger testbed for long-term retrieval strategies.

In summary, static data provide large-scale credible knowledge sources, whereas interactive data shapes the design of real-world tasks and evaluation settings, as discussed in the following sections.

2.2 Data Preprocessing and Task Formulation

Once data are collected, preprocessing and task formulation transform raw corpora into structured formats, ready to compose usable training datasets or benchmarks.

Preprocessing. Collected data are often noisy, redundant, or unstructured. Preprocessing typically involves filtering invalid, duplicated, or overly difficult samples. Documents are segmented into chunks or represented as graphs with specific ordering, such as relation schemas (Guo et al., 2025) or chronological structures (Li et al., 2025a). In some cases, domain-specific knowledge bases are created to support efficient retrieval. This step ensures quality and prepares the data for downstream task construction.

Task formulation. After preprocessing, tasks are defined to align with model training and evaluation goals. Traditional QA benchmarks formulate closed-ended tasks with answers confined to paragraphs (Kwiatkowski et al., 2019; Yang et al., 2018), resulting in static corpora. Agentic RAG benchmarks, however, increasingly feature complex tasks that require decomposing and reasoning. Task definitions are often tied to real-world workflows, such as real-world software issues in SWE-bench (Jimenez et al., 2024) or competition-style modeling problems in MLE-bench (Chan et al., 2024). There are also creative tasks such as survey writing (Wang et al., 2024c; Yan et al., 2025; Liang et al., 2025b) or research assistant (Schmidgall et al., 2025). These tasks are more dynamic and open-ended compared to traditional QA. As a result, in Agentic RAG systems, different retrieving tools replace static supporting documents.

Together, data preprocessing and task formulation bridge the gap between raw sources and structured problems, paving the way for Section 2.3, where annotation and synthesis methods for task construction are discussed.

2.3 Task Construction: Annotation and Synthesis

Once structural data are obtained, they can be repurposed to construct diverse tasks for training and evaluating Agentic RAG systems. Such tasks may be generated either through manual annotation or automatic synthesis methods, typically with the assistance of LLMs. Most current approaches adopt a *generate–verify–filter/refine* pipeline.

2.3.1 Generate

In the generation stage, structural data and instructions are provided to human annotators or LLMs to create tasks of various types. While data are often randomly sampled from the entire corpus, recent work (Tao et al., 2025) has explored more principled formalizations (e.g., set theory, Knowledge Projection) to actively search for informative samples. Many widely used datasets use crowdsourced tasks (Kwiatkowski et al., 2019; Yang et al., 2018; Wei et al., 2024; Mialon et al., 2024) or modify ready tasks available on the Internet (Joshi et al., 2017; Jimenez et al., 2024). However, for scalability and contamination reasons, synthetic

datasets generated with the assistance of LLMs are also commonly seen (Zhu et al., 2024; Patil et al., 2024; Ho et al., 2020; Wu et al., 2025a), employing prompting strategies such as self-instruct (Wang et al., 2023a), in-context learning (Brown et al., 2020), and template-based methods.

A central challenge in developing complex agentic Retrieval-Augmented Generation (RAG) systems is how to generate tasks that remain sufficiently difficult as models improve. Many widely used benchmarks have quickly reached saturation, raising concerns that high scores reflect superficial pattern matching or test-set leakage rather than genuine reasoning. For instance, recent analyses (Alberti et al., 2019; Wei et al., 2024) show that large language models achieve near-ceiling performance on classic single-hop QA datasets such as NQ (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017), and evaluations reveal large generalization gaps between training and unseen test levels (Sen and Saffari, 2020). **Difficulty enhancement** thus serves to push models beyond shallow reasoning, direct retrieval, or memorization. Existing methods can be grouped into three categories, each targeting a distinct limitation (Table 2):

Limitation	Reason	Solution
Poor planning	Shallow, single-step tasks	Complexity
Weak reasoning	Tasks solvable by direct retrieval	Uncertainty
Bad generalization	Memorizing the pretraining corpora	Expertise

Table 2: Limitations of classic datasets and corresponding solutions via difficulty enhancement.

- **Complexity**: By increasing structural complexity or tool requirements, tasks force agents to perform multi-step reasoning and long-horizon planning. Techniques include expanding single-hop questions into multi-hop ones (Yang et al., 2018; Ho et al., 2020; Trivedi et al., 2022; Shi et al., 2025; Wu et al., 2025a), extending single-turn interactions into multi-turn conversations (Deng et al., 2024), integrating information from multiple webpages (Wu et al., 2025b), scaling coding tasks from single files to full repositories (Jimenez et al., 2024; Liu et al., 2024a), and designing tasks with multi-tool usage (Mialon et al., 2024). Datasets constructed in this way can naturally supply decomposed sub-tasks to facilitate training of the planning abilities of agents.

- **Uncertainty**: Introducing ambiguity or hidden reasoning steps prevents agents from simply locating answers and encourages deeper inference. Methods include obfuscating key information (Li et al., 2025c), generating inverted problems (Wei et al., 2025), constructing implicit reasoning tasks (Geva et al., 2021), adding distractors to reference documents (Ho et al., 2020; Trivedi et al., 2022) or including unanswerable questions (Trivedi et al., 2022).

- **Expertise**: Incorporating tasks requiring specialized human expertise challenges models to operate beyond their pretraining corpora. Benchmarks such as GPQA (Rein et al., 2024) and Humanity's Last Exam (Phan et al., 2025) leverage domain experts to craft questions that remain unsolved by current frontier models.

2.3.2 Verify

Verification is essential to ensure dataset validity, especially for LLM-generated tasks that are prone to hallucination (Huang et al., 2025). Verification can be conducted by human annotators or automated LLM-based verifiers. For human-based verification, some benchmarks (Wei et al., 2024, 2025; Mialon et al., 2024) adopt inter-annotator agreement to improve reliability. For LLM-based verification, Dhuliawala et al. (2024) proposes the Chain-of-Verification framework, where models are prompted to generate a checklist of verification questions and independently answer them, reducing hallucinations.

Beyond correctness in a narrow sense, overlooked validity criteria should be explicitly considered. In QA tasks, questions should have a unique, time-invariant answer (Wei et al., 2024; Mialon et al., 2024). In coding tasks, the environment must be reproducible and the reference code passable under all unit tests (Jimenez et al., 2024).

2.3.3 Filter/Refine

After verification, datasets are further filtered or refined to enhance practical utility and better serve downstream training and evaluation. Such refinement typically proceeds along two major axes:

- **Quality:** Dataset quality is inherently multi-faceted and can be assessed from several complementary perspectives. One key dimension is *linguistic naturalness*: tasks should be phrased in fluent, human-like language that aligns with real-world usage (Wu et al., 2025b; Zhang et al., 2025b). Another is *robustness*, which requires

avoiding data leakage or exploitable shortcuts that could trivialize the task (Trivedi et al., 2020, 2022). In addition, external signals such as GitHub stars, StackOverflow likes, or paper citation counts can serve as *source credibility* indicators to guide data selection.

- **Difficulty:** Filtering out overly easy tasks ensures higher utility for training and evaluation. Moreover, difficulty assessment itself supports curriculum learning (Soviany et al., 2022) and fine-grained evaluation. Existing methods fall into two categories: rule-based and LLM-based. Rule-based criteria often reuse the difficulty signals from the generation stage, such as the number of hops (Shi et al., 2025), webpages (Wu et al., 2025b), or tools (Mialon et al., 2024), with or without unanswerable questions (Trivedi et al., 2022), and accuracy of experts and non-experts (Rein et al., 2024). LLM-based methods include prompting models to score difficulty, or measuring performance by letting an LLM agent attempt the task and using its success rate as a proxy (Ding et al., 2024; Tambon et al., 2024).

2.4 Data for Evaluation

A central challenge in building and benchmarking Agentic RAG systems lies in defining and operationalizing appropriate evaluation data. The first crucial step for a valid benchmark is decontamination. After this, reasonable evaluation metrics and approaches should be applied to evaluate models.

2.4.1 Decontamination

Standard decontamination techniques aim to ensure test sets are not leaked into training data. This not only includes identical tasks, but should also include tasks that are partially related (e.g., MuSiQue (Trivedi et al., 2022) filter out multi-hop questions with any identical single-hop component). More recent benchmarks, such as GAIA (Mialon et al., 2024), emphasize internet-era test splits (e.g., question does not exist on the internet in plain text) to prevent contamination from pretraining corpora. A related open question is whether evaluation data should remain static or be dynamically updated over time, reflecting shifts in world knowledge and system usage. Dynamic benchmarks could provide stronger guarantees against overfitting but introduce challenges in reproducibility and longitudinal comparison.

2.4.2 Evaluation Metrics and Approaches

Unlike conventional RAG, which often focuses on retrieval accuracy and answer correctness in relatively constrained QA settings, Agentic RAG systems involve multi-step reasoning, tool usage, and real-world settings. As a result, evaluation must move beyond correctness-only signals and account for metrics including efficiency and safety.

Correctness. Correctness remains the fundamental metric for evaluating Agentic RAG systems. Depending on the task, this can take multiple forms. In *QA tasks*, correctness is typically defined with respect to gold-standard answers, evaluated by string matching (exact, fuzzy, or F1 score). In *task-oriented domains* such as software engineering or machine learning, programmatic validators such as unit tests (Jimenez et al., 2024) or machine learning model test scores (Chan et al., 2024) serve as correctness signals. For *open-ended domains* such as academic writing, correctness may be approximated via LLM-as-a-judge (Gu et al., 2024) in peer-review style, evaluating citation quality and content quality (coverage, structure, and relevance) for surveys (Wang et al., 2024c), or quality of problem, method, and experiment for problem-solving papers (Baek et al., 2025).

Beyond correctness. Given the agentic nature of RAG, additional dimensions of evaluation are increasingly necessary.

- **Efficiency** captures the computational or interaction cost required to reach a solution, such as the number of tool calls, reasoning steps, or generated tokens. For example, WebWalkerQA (Wu et al., 2025b) uses the action count of successful agentic executions as the efficiency metric.

- **Safety** measures the potential risks brought by system failures. Recent work, such as Amazon-bench (Zhang et al., 2025b), has distinguished (via LLM-as-a-judge) between *benign failures* (where the agent fails to complete the task but does not cause any changes in user state) and *harmful failures* (where the agent performs actions that have negative impacts on users).

Incorporating these metrics ensures that evaluations align with real-world deployment concerns.

2.5 Data Enhancement for Training

The generated dataset often requires further enhancement depending on the finetuning paradigm:

supervised finetuning (SFT) or reinforcement learning (RL).

2.5.1 SFT

In the early stage of SFT, agents are typically exposed to relatively simple instruction-following corpora to acquire basic tool-usage skills. Datasets can be automatically constructed by modifying pretraining corpora (Schick et al., 2023), integrating multiple resources into meta-datasets (Zhu et al., 2024), or applying self-instruction and in-context learning techniques (Patil et al., 2024).

With the rise of chain-of-thought (CoT) prompting (Wei et al., 2022), most agentic systems are finetuned on extended thought–action trajectories, often based on the ReAct framework (Yao et al., 2023). These trajectories include plans for retrieval, tool calls, and reflections based on the retrieved information. Such trajectories are generally produced following the *generate–verify–filter/refine* pipeline described in section 2.3.

Generate. Trajectories may be obtained from scratch by human annotators, script-based templates, or in-context bootstrapping (e.g., STaR (Zelikman et al., 2022)). Trajectory distillation (Ho et al., 2023; Magister et al., 2023; Mukherjee et al., 2023; Hsieh et al., 2023) from stronger reasoning models is also a standard practice in Agentic RAG systems (Kang et al., 2025; Tao et al., 2025; Li et al., 2025c). In addition, new tasks executed by the agent can provide fresh successful trajectories, which may be reused for self-improvement and continual refinement of the model.

Verify. Verifying the correctness of generated trajectories remains a major challenge. Current practice often relies on rejection sampling, retaining only those trajectories with correct final answers. However, it does not guarantee the validity of intermediate reasoning steps, leaving the problem of trajectory-level correctness largely unresolved.

Filter/Refine. After verification, trajectories may be filtered or refined along two axes—*quality* and *conciseness*—to be more educationally useful and to facilitate learning for smaller models. *Quality* is influenced by factors such as trajectory granularity, formatting choices, and the teacher model used (Chen et al., 2025b), yet no universal solution has emerged. Regarding *conciseness*, overthinking (Chen et al., 2024) is a common issue in large reasoning models and may hinder smaller models during training. To mitigate this, Tao et al. (2025) filters out trajectories with severe repetition, Li et al. (2025c) reconstructs concise rationales from action–observation sequences, and Luo et al. (2025) removes redundant or incorrect reasoning paths.

2.5.2 RL

Reinforcement learning (RL) has become a powerful paradigm for improving the reasoning capabilities of LLM agents. In contrast to SFT, where data enhancement primarily focuses on generating and refining trajectories, RL emphasizes the design of reward signals. Most current work (Jin et al., 2025a; Song et al., 2025; Zheng et al., 2025; Chen et al., 2025a; Wu et al., 2025a; Li et al., 2025c; Tao et al., 2025) achieves competitive results with outcome-based rewards (sometimes with format rewards) without extensive data enhancement. Nevertheless, incorporating richer, data-aware reward functions represents a promising future direction. For instance, DeepRetrieval (Jiang et al., 2025) and ReZero (Dao and Le, 2025) employ retrieval-based rewards (i.e., whether the correct documents are retrieved) to improve query formulation. WebThinker (Li et al., 2025d) generated diverse reasoning trajectories and manually constructed a preference pair dataset based on quality, efficiency, and conciseness. Moreover, process reward models (PRMs), proved effective in mathematical reasoning (Lightman et al., 2024; Wang et al., 2024b; Zhang et al., 2025c), may be adapted to Agentic RAG systems to guide intermediate reasoning rather than relying solely on outcome correctness.

3 Domain-Specific Agentic RAG Training Datasets and Benchmarks

Agentic RAG is widely used in different domains such as Question Answering (QA), web, software engineering, research, medical, and legal fields. Table 3 summarizes the retrieval data sources, tools, and representative Agentic RAG training datasets and benchmarks in different domains. The main topics of these domains are as follows:

Question Answering (QA). QA tasks can be categorized along two primary dimensions: (i) *reasoning complexity*, ranging from single-hop QA that relies on direct evidence to multi-hop QA requiring compositional reasoning across multiple documents; and (ii) *knowledge source*, such as Wikipedia-based datasets (e.g., Natural Questions (Kwiatkowski et al., 2019), HotpotQA (Yang

Domain	Retrieval Data Source	Tools	Training Datasets	Benchmarks
QA	Wikipedia, Internet	Search engine	ELI5 (Fan et al., 2019), WebGLM-QA (Liu et al., 2023), Self-RAG (Asai et al., 2024), Auto-RAG (Yu et al., 2024) Chain of Agents (Zhang et al., 2024c)	**Single-hop:** NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), SimpleQA (Wei et al., 2024), **Multi-hop:** HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), Bamboogle (Press et al., 2023), Taskcraft (Shi et al., 2025)
Web	Internet	Web navigator, OCR, HTML parser, web APIs	**Text-based:** WebGPT (Nakano et al., 2021), AutoWebGLM (Lai et al., 2024), WebThinker (Li et al., 2025d), WebDancer (Wu et al., 2025a), SailorFog-QA (Li et al., 2025c), WebShaper (Tao et al., 2025) **MultiModal:** OpenWebVoyager (He et al., 2024), WebWatcher (Geng et al., 2025)	WebArena (Zhou et al., 2024), AgentBench (Liu et al., 2024b), GAIA (Mialon et al., 2024), BrowseComp (Wei et al., 2025), WebWalkerQA (Wu et al., 2025b), Amazon-bench (Zhang et al., 2025b)
Software Engineering	Github repositories, API documentations	Code navigator and interpreter, search engine	/	SWE-Bench (Jimenez et al., 2024), RepoBench (Liu et al., 2024a), DevEval (Li et al., 2024a)
Research	Arxiv, Google scholar, Huggingface datasets	Python, Latex	/	MLAgentBench (Huang et al., 2023), MLE-Bench (Chan et al., 2024), SurveyBench (Yan et al., 2025)
Medical	PubMed, UMLS	Search engine	Patho-AgenticRAG (Zhang et al., 2025a), DermaVQA-DAS (Yim et al., 2025)	Quilt-VQA (Seyfioglu et al., 2023), Path-VQA (He et al., 2020) MedQA (Jin et al., 2020), MedMCQA (Pal et al., 2022), PathMMU (Sun et al., 2024b)
Legal	Legal documents	Search engine, web navigator	LawLuo (Sun et al., 2024a)	LegalBench (Guha et al., 2023), LegalBench-RAG (Pipitone and Alami, 2024)

Table 3: Features and representative Agentic RAG training datasets and benchmarks in different domains. Training datasets without specific names are referred to by the titles of the papers that introduced them.

et al., 2018)) versus non-Wikipedia sources (e.g., quiz websites (Joshi et al., 2017)).

Web. Web-based tasks can appear as QA-like information-seeking problems (Mialon et al., 2024; Wei et al., 2025), or include goal-driven interactions such as online shopping, booking, or multi-step navigation (Liu et al., 2024b; Zhang et al., 2025b). Knowledge sources are more diverse and often *text-based* (OCR, HTML parser or web APIs) or *multimodal* (screenshots of webpages).

Software Engineering. Software engineering tasks differ from *function-level* tasks such as code completion or bug fixing, which can be completed by an LLM. They are usually *repository-level* tasks where an agent navigates and edits multi-file projects. Evaluation often involves *unit-test-based* correctness (Jimenez et al., 2024) or fine-grained *line-level* matching (Liu et al., 2024a). Knowledge sources, including API documentation, related repositories, or historical issues, can be retrieved to guide code generation and refactoring.

Research. Research-oriented tasks span a spectrum from *programmatic* tasks, such as participating in Kaggle competitions (Chan et al., 2024), to *creative* tasks like writing survey papers (Wang et al., 2024c) or generating novel research ideas (Li et al., 2024b). Knowledge sources include academic papers, datasets, and codebases.

Medical. Medical tasks often resemble QA-style problems, with a large proportion framed as text-based multiple-choice questions resembling real-world medical exam questions (Pal et al., 2022; Jin et al., 2020). More challenging datasets involve VQA (vision question answering) (Zhang et al., 2024b; He et al., 2020; Thakrar et al., 2025). In addition to broad diagnostic reasoning, tasks may focus on narrower domains such as vaccination (Zeggai et al., 2025), surgery (Low et al., 2025), etc.

Legal. Legal tasks are generally framed as QA problems that require evidence-retrieval, multi-step reasoning, classification, and arithmetic computation (Guha et al., 2023; Pipitone and Alami, 2024). The underlying corpus of statutes, regulations, and case law contains extensive definitions and highly contextual provisions, making retrieval and reasoning more challenging (Wahidur et al., 2025; Watson et al., 2024; Barron et al., 2025).

Table 4 summarizes the task definitions, knowledge sources, data scale, evaluation metrics, and data curation methods of different Agentic RAG benchmarks in these domains.

4 Challenges and Open Problems

Despite recent advances, several challenges remain open in developing robust Agentic RAG systems.

Quality of Synthetic Data. Synthetic data generation (Nadas et al., 2025) offers a clear path to scale: large language models can produce vast quantities of training datasets and trajectories that bootstrap both SFT and RL. However, scale amplifies an accuracy tradeoff — automatically generated trajectories often contain factual inaccuracies, spurious reasoning steps, or low-quality data that are difficult to detect at scale, and iterative reliance on such synthetic content risks degrading downstream behavior. Therefore, it is essential to include a strict verification and filtering/refinement stage in the data preparation pipeline.

Name	Task	Source	Scale	Metrics	Method
Question Answering (QA)					
NQ	Single-hop QA	Google queries, Wikipedia	train 307k, dev 7.8k, test 7.8k	-	Select queries from Google. Search for relevant documents in Wikipedia, and ask annotators to identify answers and filter low-quality questions.
SimpleQA	Single-hop QA	Crowdsourced	4326	-	Annotators create questions with unique time-invariant answer. All questions are verified by another person independently. Keep only those that are incorrectly answered at least once in 4 times by gpt-4.
HotpotQA	Multi-hop QA	Crowdsourced from Wikipedia	train 90.4k, val 7.4k, test 7.4k	-	Build a relation graph from the links in Wikipedia. Choose relevant paragraphs from it, and ask annotators to create multi-hop questions based on the paragraphs and identify supporting facts in them.
MuSiQue	Multi-hop QA	Synthesized and annotated from Wikipedia	train 39.9k, val 4.8k, test 4.9k	-	Collect Wikipedia-based single-hop questions. Compose 2-hop questions and filter out those with shortcuts. Build different multi-hop question structures and crowdsource questions. Add distractors in supporting documents. Add unanswerable questions.
Web					
WebArena	QA-like & task-oriented web interaction	Custom web environments (shopping, email, forum, map, social media)	7 environments, 812 tasks	Task success rate	Provide realistic multi-page websites. Annotators design diverse tasks requiring navigation, reasoning and interaction.
BrowseComp	Fact-seeking QA over web browsing	Internet (open web), human-crafted QA	1,266 questions	Exact match	Questions designed so answer is short and verifiable. Human annotators ensure difficulty (not solved by existing models, not in top search results), enforce time/effort thresholds.
Software Engineering					
SWE-bench	Generate a pull request (PR) to solve a given issue	GitHub issues from 12 Python repositories	train 19k, test 2294	Unit test pass rate	Select PRs that resolve an issue and contribute tests. Keep only those that install successfully and passes all tests.
RepoBench	Code retrieval, code completion	Github-code dataset, Github Python and Java repositories	Python 24k, Java 26k	Golden snippet matching, line matching	Random sample lines as completion goals (with a first-to-use subset). Extract candidate snippets based on import codes, and annotate golden snippets.
Machine Learning					
MLEbench	Achieve the best score on a metric pre-defined for each competition	Kaggle	75	Test score compared on leaderboard (e.g. medals)	Crawl task description, dataset, grading code and leaderboard from Kaggle website. Keep only those reproducible and up-to-date. Manually label the category and difficulty.
Medical					
MedQA	Four-option multiple-choice question	National Medical Board Examination	train 48.9k, dev 6.1k, test 8.1k	Exact match	Collect question-answer pairs from the National Medical Board Examination.
Quilt-VQA	VQA (Vision question answering)	Educational histopathology videos in Youtube	**Image-dependent:** 1055 **General-knowledge:** 255	LLM evaluation	Localize the "?"s in the video's transcript. Extract the relevant texts and images. Prompt GPT-4 to generate QA pairs. Perform a manual verification.
Legal					
LegalBench	Issue-spotting, rule-recall, rule-application and rule-conclusion, interpretation, rhetorical-understanding	Existing datasets, in-house datasets	9.1k	Accuracy, human evaluation	Filter and restructure the data from the data sources.

Table 4: Details of domain-Specific Agentic RAG benchmarks. The metrics of QA are generally string matching (exact or fuzzy) or F1 score, and are omitted in the table. References are in Table 3. This table only include part of the benchmarks due to page limit, see the full table in appendix B.

Trajectory Quality. Beyond raw volume, a central open problem is how to quantify and enforce trajectory quality — i.e., whether a recorded sequence of retrievals, tool uses, and reasoning steps is truthful, causally relevant, and pedagogically useful for SFT. Outcome-based rewards, though proven effective, risk conflating surface success with reasoning quality. Instead, we need formulations of process rewards that can provide finer-grained trajectory feedback to be applied in RL.

Feasibility: Easy Grading vs. Complex Environments Practical evaluation pipelines are tightly coupled to task design. Benchmarks with short verifiable answers (Kwiatkowski et al., 2019; Wei et al., 2024; Wu et al., 2025b; Wei et al., 2025; Mialon et al., 2024) provide examples where grading is relatively straightforward, making them useful for rapid evaluation and iteration of agents. In contrast, assessing performance in open-ended, multi-turn, or interactive environments (Liu et al., 2024b) is substantially more complex, limiting the ease of deployment and reproducibility of such benchmarks.

Robustness of Evaluation. (i) **Incorrect Reasoning Traces:** A harmful failure mode is when agents produce incorrect internal traces that nevertheless lead to correct final answers. Such behavior contaminates SFT corpora (poor-quality trajectories become imitated), biases RL signal (rewards tied to final correctness ignore process defects), and undermines evaluation (agents appear competent while lacking correct reasoning). (ii) **Bias of LLM-as-Judge:** Relying on LLMs as automatic judges introduces systematic biases: studies have shown verbosity/formatting biases (Zheng et al., 2023) and self-preference tendencies (Wataoka et al., 2024) where LLM judges favor certain styles even when those outputs are not objectively better. This affects both benchmark scores and model development cycles (developers optimize to please the judge rather than to improve true competence). Research is needed to produce fair, reliable assessments for Agentic RAG systems.

5 Conclusion

This paper conducts a comprehensive survey of Agentic RAG from data perspectives. We first summarize and analyze the data lifecycle in the Agentic RAG field. Next, we review dataset curation methods following the generate–verify–filter/refine pipeline. To facilitate researchers across diverse domains, we showcase representative Agentic RAG training datasets and benchmarks in different domains. Finally, we discuss current challenges and open problems of data in Agentic RAG, and shed light on promising future directions.

Limitations

This survey primarily focuses on text-based data in Agentic RAG and does not cover multimodal data—such as the figures and formulas commonly used in AI4Science—which may involve substantially different data pipelines. The survey emphasizes conceptual frameworks and data lifecycles but does not provide large-scale quantitative benchmarking or unified evaluation across datasets.

References

Chris Alberti, Kenton Lee, and Michael Collins. 2019. A bert baseline for the natural questions. *arXiv preprint arXiv:1901.08634*.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. 2025. Researchagent: Iterative research idea generation over scientific literature with large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6709–6738.

Tianyi Bai, Hao Liang, Binwang Wan, Yanran Xu, Xi Li, Shiyu Li, Ling Yang, Bozhou Li, Yifan Wang, Bin Cui, et al. 2024. A survey of multimodal large language model from a data-centric perspective. *arXiv preprint arXiv:2405.16640*.

Ryan C. Barron, Maksim E. Eren, Olga M. Serafimova, Cynthia Matuszek, and Boian S. Alexandrov. 2025. Bridging legal knowledge and ai: Retrieval-augmented generation with vector stores, knowledge graphs, and hierarchical non-negative matrix factorization. *Preprint*, arXiv:2502.20364.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, et al. 2024. Mle-bench: Evaluating machine learning agents on machine learning engineering. *arXiv preprint arXiv:2410.07095*.

Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z Pan, Wen Zhang, Huajun Chen, Fan Yang, et al. 2025a. Learning to reason with search for llms via reinforcement learning. *arXiv preprint arXiv:2503.19470*.

Xinghao Chen, Zhijing Sun, Wenjin Guo, Miaoran Zhang, Yanjun Chen, Yirong Sun, Hui Su, Yijie Pan, Dietrich Klakow, Wenjie Li, et al. 2025b. Unveiling the key factors for distilling chain-of-thought reasoning. *arXiv preprint arXiv:2502.18001*.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. 2024. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*.

Alan Dao and Thinh Le. 2025. Rezero: Enhancing llm search ability by trying one-more-time. *arXiv preprint arXiv:2504.11001*.

Yang Deng, Xuan Zhang, Wenxuan Zhang, Yifei Yuan, See Kiong Ng, and Tat-Seng Chua. 2024. On the multi-turn instruction following for conversational web agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8795–8812.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. Chain-of-verification reduces hallucination in large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 3563–3578.

Mucong Ding, Chenghao Deng, Jocelyn Choo, Zichu Wu, Aakriti Agrawal, Avi Schwarzschild, Tianyi Zhou, Tom Goldstein, John Langford, Animashree Anandkumar, et al. 2024. Easy2hard-bench: Standardized difficulty labels for profiling llm performance and generalization. *Advances in Neural Information Processing Systems*, 37:44323–44365.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567. Association for Computational Linguistics.

Xinyu Geng, Peng Xia, Zhen Zhang, Xinyu Wang, Qiuchen Wang, Ruixue Ding, Chenxi Wang, Jialong Wu, Yida Zhao, Kuan Li, et al. 2025. Webwatcher: Breaking new frontiers of vision-language deep research agent. *arXiv preprint arXiv:2508.05748*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon,

Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, Jessica Wu, Joe Nudell, Joel Niklaus, John Nay, Jonathan H. Choi, Kevin Tobia, Margaret Hagan, Megan Ma, Michael Livermore, Nikon Rasumov-Rahe, Nils Holzenberger, Noam Kolt, Peter Henderson, Sean Rehaag, Sharad Goel, Shang Gao, Spencer Williams, Sunny Gandhi, Tom Zur, Varun Iyer, and Zehua Li. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Preprint*, arXiv:2308.11462.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2025. Lightrag: Simple and fast retrieval-augmented generation. *Preprint*, arXiv:2410.05779.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Hongming Zhang, Tianqing Fang, Zhenzhong Lan, and Dong Yu. 2024. Openwebvoyager: Building multimodal web agents via iterative real-world exploration, feedback and optimization. *arXiv preprint arXiv:2410.19609*.

Xuehai He, Yichen Zhang, Luntian Mou, Eric Xing, and Pengtao Xie. 2020. Pathvqa: 30000+ questions for medical visual question answering. *arXiv preprint arXiv:2003.10286*.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2023. Large language models are reasoning teachers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14852–14882.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. 2023. Mlagentbench: Evaluating language agents on machine learning experimentation. *arXiv preprint arXiv:2310.03302*.

Pengcheng Jiang, Jiacheng Lin, Lang Cao, Runchu Tian, SeongKu Kang, Zifeng Wang, Jimeng Sun, and Jiawei Han. 2025. Deepretrieval: Hacking real search engines and retrievers with large language models via reinforcement learning. *arXiv preprint arXiv:2503.00223*.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. 2024. Swe-bench: Can language models resolve real-world github issues? In *ICLR*.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025a. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2020. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Preprint*, arXiv:2009.13081.

Jiajie Jin, Yutao Zhu, Zhicheng Dou, Guanting Dong, Xinyu Yang, Chenghao Zhang, Tong Zhao, Zhao Yang, and Ji-Rong Wen. 2025b. Flashrag: A modular toolkit for efficient retrieval-augmented generation research. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 737–740.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Minki Kang, Jongwon Jeong, Seanie Lee, Jaewoong Cho, and Sung Ju Hwang. 2025. Distilling llm agent into small models with retrieval and code tools. *arXiv preprint arXiv:2505.17612*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Hanyu Lai, Xiao Liu, Iat Long Iong, Shuntian Yao, Yuxuan Chen, Pengbo Shen, Hao Yu, Hanchen Zhang, Xiaohan Zhang, Yuxiao Dong, et al. 2024. Autowebglm: A large language model-based web navigating agent. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5295–5306.

Dong Li, Yichen Niu, Ying Ai, Xiang Zou, Biqing Qi, and Jianxing Liu. 2025a. T-grag: A dynamic graphrag framework for resolving temporal conflicts and redundancy in knowledge retrieval. *Preprint*, arXiv:2508.01680.

Jia Li, Ge Li, Yunfei Zhao, Yongmin Li, Huanyu Liu, Hao Zhu, Lecheng Wang, Kaibo Liu, Zheng Fang, Lanshen Wang, et al. 2024a. Deveval: A manually-annotated code generation benchmark aligned with

real-world code repositories. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 3603–3614.

Jia Li, Xianjie Shi, Kechi Zhang, Lei Li, Ge Li, Zhengwei Tao, Fang Liu, Chongyang Tao, and Zhi Jin. 2025b. Coderag: Supportive code retrieval on bigraph for real-world code generation. *arXiv preprint arXiv:2504.10046*.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, et al. 2025c. Websailor: Navigating super-human reasoning for web agent. *arXiv preprint arXiv:2507.02592*.

Long Li, Weiwen Xu, Jiayan Guo, Ruochen Zhao, Xingxuan Li, Yuqian Yuan, Boqiang Zhang, Yuming Jiang, Yifei Xin, Ronghao Dang, et al. 2024b. Chain of ideas: Revolutionizing research via novel idea development with llm agents. *arXiv preprint arXiv:2410.13185*.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou. 2025d. Webthinker: Empowering large reasoning models with deep research capability. *arXiv preprint arXiv:2504.21776*.

Yangning Li, Weizhi Zhang, Yuyao Yang, Wei-Chieh Huang, Yaozu Wu, Junyu Luo, Yuanchen Bei, Henry Peng Zou, Xiao Luo, Yusheng Zhao, et al. 2025e. Towards agentic rag with deep reasoning: A survey of rag-reasoning systems in llms. *arXiv preprint arXiv:2507.09477*.

Jintao Liang, Gang Su, Huifeng Lin, You Wu, Rui Zhao, and Ziyue Li. 2025a. Reasoning rag via system 1 or system 2: A survey on reasoning agentic retrieval-augmented generation for industry challenges. *arXiv preprint arXiv:2506.10408*.

Xun Liang, Jiawei Yang, Yezhaohui Wang, Chen Tang, Zifan Zheng, Shichao Song, Zehao Lin, Yebin Yang, Simin Niu, Hanyu Wang, et al. 2025b. Surveyx: Academic survey automation via large language models. *arXiv preprint arXiv:2502.14776*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Tianyang Liu, Canwen Xu, and Julian McAuley. 2024a. Repobench: Benchmarking repository-level code auto-completion systems. In *The Twelfth International Conference on Learning Representations*.

Xiao Liu, Hanyu Lai, Hao Yu, Yifan Xu, Aohan Zeng, Zhengxiao Du, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. Webglm: towards an efficient web-enhanced question answering system with human preferences. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pages 4549–4560.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2024b. Agentbench: Evaluating LLMs as agents. In *The Twelfth International Conference on Learning Representations*.

Yang Liu, Jiahuan Cao, Chongyu Liu, Kai Ding, and Lianwen Jin. 2024c. Datasets for large language models: A comprehensive survey. *arXiv preprint arXiv:2402.18041*.

Chang Han Low, Ziyue Wang, Tianyi Zhang, Zhitao Zeng, Zhu Zhuo, Evangelos B. Mazomenos, and Yueming Jin. 2025. Surgraw: Multi-agent workflow with chain-of-thought reasoning for surgical intelligence. *Preprint*, arXiv:2503.10265.

Yijia Luo, Yulin Song, Xingyao Zhang, Jiaheng Liu, Weixun Wang, GengRu Chen, Wenbo Su, and Bo Zheng. 2025. Deconstructing long chain-of-thought: A structured reasoning optimization framework for long cot distillation. *arXiv preprint arXiv:2503.16385*.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2023. Teaching small language models to reason. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1773–1781.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2024. GAIA: a benchmark for general AI assistants. In *The Twelfth International Conference on Learning Representations*.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196*.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*.

Mihai Nadas, Laura Diosan, and Andreea Tomescu. 2025. Synthetic data generation using large language models: Advances in text and code. *arXiv preprint arXiv:2503.14023*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR.

Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. 2024. Gorilla: Large language model connected with massive apis. *Advances in Neural Information Processing Systems*, 37:126544–126565.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. 2025. Humanity's last exam. *arXiv preprint arXiv:2501.14249*.

Nicholas Pipitone and Ghita Houir Alami. 2024. Legalbench-rag: A benchmark for retrieval-augmented generation in the legal domain. *Preprint*, arXiv:2408.10343.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551.

Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Michael Moor, Zicheng Liu, and Emad Barsoum. 2025. Agent laboratory: Using llm agents as research assistants. *arXiv preprint arXiv:2501.04227*.

Priyanka Sen and Amir Saffari. 2020. What do models learn from question answering datasets? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2429–2438.

Mehmet Saygin Seyfioglu, Wisdom O. Ikezogwo, Fatemeh Ghezloo, Ranjay Krishna, and Linda Shapiro. 2023. Quilt-llava: Visual instruction tuning by extracting localized narratives from open-source histopathology videos. *Preprint*, arXiv:2312.04746.

Dingfeng Shi, Jingyi Cao, Qianben Chen, Weichen Sun, Weizhen Li, Hongxuan Lu, Fangchen Dong, Tianrui Qin, King Zhu, Minghao Liu, et al. 2025. Taskcraft: Automated generation of agentic tasks. *arXiv preprint arXiv:2506.10055*.

Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. 2025. Agentic retrieval-augmented generation: A survey on agentic rag. *arXiv preprint arXiv:2501.09136*.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. 2025. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2022. Curriculum learning: A survey. *International Journal of Computer Vision*, 130(6):1526–1565.

Jingyun Sun, Chengxiao Dai, Zhongze Luo, Yangbo Chang, and Yang Li. 2024a. Lawluo: A multi-agent collaborative framework for multi-round chinese legal consultation. *Preprint*, arXiv:2407.16252.

Yuxuan Sun, Hao Wu, Chenglu Zhu, Sunyi Zheng, Qizi Chen, Kai Zhang, Yunlong Zhang, Xiaoxiao Lan, Mengyue Zheng, Jingxiong Li, Xinheng Lyu, Tao Lin, and Lin Yang. 2024b. Pathmmu: A massive multimodal expert-level benchmark for understanding and reasoning in pathology. *Preprint*, arXiv:2401.16355.

Florian Tambon, Amin Nikanjam, Foutse Khomh, and Giuliano Antoniol. 2024. Assessing programming task difficulty for efficient evaluation of large language models. *arXiv e-prints*, pages arXiv–2407.

Zhengwei Tao, Jialong Wu, Wenbiao Yin, Junkai Zhang, Baixuan Li, Haiyang Shen, Kuan Li, Liwen Zhang, Xinyu Wang, Yong Jiang, et al. 2025. Webshaper: Agentically data synthesizing via information-seeking formalization. *arXiv preprint arXiv:2507.15061*.

Karishma Thakrar, Shreyas Basavatia, and Akshay Daftardar. 2025. Architecting clinical collaboration: Multi-agent reasoning systems for multimodal medical vqa. *Preprint*, arXiv:2507.05520.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2020. Is multihop qa in dire condition? measuring and reducing disconnected reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8846–8863.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Rahman S. M. Wahidur, Sumin Kim, Haeung Choi, David S. Bhatti, and Heung-No Lee. 2025. Legal query rag. *IEEE Access*, 13:36978–36994.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2024a. A survey on large

language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024b. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439.

Yidong Wang, Qi Guo, Wenjin Yao, Hongbo Zhang, Xin Zhang, Zhen Wu, Meishan Zhang, Xinyu Dai, Qingsong Wen, Wei Ye, et al. 2024c. Autosurvey: Large language models can automatically write surveys. *Advances in neural information processing systems*, 37:115119–115145.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023a. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508.

Zige Wang, Wanjun Zhong, Yufei Wang, Qi Zhu, Fei Mi, Baojun Wang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023b. Data management for training large language models: A survey. *arXiv preprint arXiv:2312.01700*.

Koki Wataoka, Tsubasa Takahashi, and Ryokan Ri. 2024. Self-preference bias in llm-as-a-judge. *arXiv preprint arXiv:2410.21819*.

William Watson, Nicole Cho, Nishan Srishankar, Zhen Zeng, Lucas Cecchi, Daniel Scott, Suchetha Siddagangappa, Rachneet Kaur, Tucker Balch, and Manuela Veloso. 2024. Law: Legal agentic workflows for custody and fund services contracts. *Preprint*, arXiv:2412.11063.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. 2025. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Gang Fu, Yong Jiang, et al. 2025a. Webdancer: Towards autonomous information seeking agency. *arXiv preprint arXiv:2505.22648*.

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, and Fei Huang. 2025b. WebWalker: Benchmarking LLMs in web traversal. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10290–10305.

Wenlong Wu, Haofen Wang, Bohan Li, Peixuan Huang, Xinzhe Zhao, and Lei Liang. 2025c. Multirag: A knowledge-guided framework for mitigating hallucination in multi-source retrieval augmented generation. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*, page 3070–3083. IEEE.

Xiangchao Yan, Shiyang Feng, Jiakang Yuan, Renqiu Xia, Bin Wang, Bo Zhang, and Lei Bai. 2025. Surveyforge: On the outline heuristics, memory-driven generation, and multi-dimensional evaluation for automated survey writing. *arXiv preprint arXiv:2503.04629*.

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.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

W Yim, Y Fu, A Ben Abacha, M Yetisgen, N Codella, RA Novoa, and J Malvehy. 2025. Dermavqa-das: Dermatology assessment schema (das) and datasets for closed-ended question answering and segmentation in patient-generated dermatology images.

Tian Yu, Shaolei Zhang, and Yang Feng. 2024. Auto-rag: Autonomous retrieval-augmented generation for large language models. *arXiv preprint arXiv:2411.19443*.

Abdellah Zeggai, Ilyes Traikia, Abdelhak Lakehal, and Abdennour Boulesnane. 2025. Ai-vaxguide: An agentic rag-based llm for vaccination decisions. *Preprint*, arXiv:2507.03493.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.

Kechi Zhang, Jia Li, Ge Li, Xianjie Shi, and Zhi Jin. 2024a. Codeagent: Enhancing code generation with tool-integrated agent systems for real-world repo-level coding challenges. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13643–13658.

Wenchuan Zhang, Jingru Guo, Hengzhe Zhang, Penghao Zhang, Jie Chen, Shuwan Zhang, Zhang Zhang, Yuhao Yi, and Hong Bu. 2025a. Patho-agenticrag: Towards multimodal agentic retrieval-augmented generation for pathology vlms via reinforcement learning. *Preprint*, arXiv:2508.02258.

Xianren Zhang, Shreyas Prasad, Di Wang, Qiuhai Zeng, Suhang Wang, Wenbo Yan, and Mat Hans. 2025b. A functionality-grounded benchmark for evaluating web agents in e-commerce domains. *Preprint*, arXiv:2508.15832.

Xiaoman Zhang, Chaoyi Wu, Ziheng Zhao, Weixiong Lin, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2024b. Pmc-vqa: Visual instruction tuning for medical visual question answering. *Preprint*, arXiv:2305.10415.

Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Arik. 2024c. Chain of agents: Large language models collaborating on long-context tasks. *Advances in Neural Information Processing Systems*, 37:132208–132237.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025c. The lessons of developing process reward models in mathematical reasoning. *arXiv preprint arXiv:2501.07301*.

Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, Jie Jiang, and Bin Cui. 2024. Retrieval-augmented generation for ai-generated content: A survey. *arXiv preprint arXiv:2402.19473*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. 2025. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. *arXiv preprint arXiv:2504.03160*.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2024. Webarena: A realistic web environment for building autonomous agents. In *The Twelfth International Conference on Learning Representations*.

Xuanhe Zhou, Junxuan He, Wei Zhou, Haodong Chen, Zirui Tang, Haoyu Zhao, Xin Tong, Guoliang Li, Youmin Chen, Jun Zhou, et al. 2025. A survey of llm $\times$ data. *arXiv preprint arXiv:2505.18458*.

Yutao Zhu, Peitian Zhang, Chenghao Zhang, Yifei Chen, Binyu Xie, Zheng Liu, Ji-Rong Wen, and Zhicheng Dou. 2024. Inters: Unlocking the power of large language models in search with instruction tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2782–2809.

A Related Work

There have been several surveys on Agentic RAG: Singh et al. (2025) conducts a general survey on Agentic RAG pipelines and frameworks. Li et al. (2025e) and Liang et al. (2025a) focus on the reasoning methods and frameworks in Agentic RAG. Our survey examines Agentic RAG from the data perspectives and summarizes the data lifecycle in Agentic RAG.

Data is also an important subject in LLM research. Several surveys (Bai et al., 2024; Zhou et al., 2025; Wang et al., 2023b; Liu et al., 2024c) have summarized the general data preparation approaches to build LLMs, and our survey also reflects similar preparation workflows. However, our survey focuses on Agentic RAG and provides deeper insight into special data curation methods in this field.

B Details of Domain-Specific Agentic RAG Benchmarks (Full Table)

Table 4 is the full table that summarizes the task definitions, knowledge sources, data scale, evaluation metrics, and data curation methods of different Agentic RAG benchmarks in question answering (QA), web, software engineering, research (machine learning specifically), medical, and legal domains, as described in section 3.

Name	Task	Source	Scale	Metrics	Method
			Question Answering (QA)		
NQ	Single-hop QA	Google queries, Wikipedia	train 307k, dev 7.8k, test 7.8k	-	Select queries from Google. Search for relevant documents in Wikipedia, and ask annotators to identify answers and filter low-quality questions.
TriviaQA	Single-hop QA	Quiz websites, Wikipedia and Internet	train 76.5k, val 10.0k, test 9.5k	-	Select questions from 14 quiz websites. Search for relevant documents in Wikipedia and Internet, and keep those with answers.
SimpleQA	Single-hop QA	Crowdsourced	4326	-	Annotators create questions with unique time-invariant answer. All questions are verified by another person independently. Keep only those that are incorrectly answered at least once in 4 times by gpt-4.
HotpotQA	Multi-hop QA	Crowdsourced from Wikipedia	train 90.4k, val 7.4k, test 7.4k	-	Build a relation graph from the links in Wikipedia. Choose relevant paragraphs from it, and ask annotators to create multi-hop questions based on the paragraphs and identify supporting facts in them.
2WikiMultihopQA	Multi-hop QA	Synthesized from Wikipedia	train medium 155k, train hard 12.6k, dev 12.6k, test 12.6k	-	Classify the entities in Wikidata. Manually write different question templates, and sample entities to create questions. Filter out questions with no answer or multiple answers. Add distractors in supporting documents.
MuSiQue	Multi-hop QA	Synthesized and annotated from Wikipedia	train 39.9k, val 4.8k, test 4.9k	-	Collect Wikipedia-based single-hop questions. Compose 2-hop questions and filter out those with shortcuts. Build different multi-hop question structures and crowdsource questions. Add distractors in supporting documents. Add unanswerable questions.
Bamboogle	Multi-hop QA	Manually created from Wikipedia	125	-	Create 2-hop questions based on Wikipedia. Keep only those that cannot be directly searched for the correct answer.
Taskcraft	Multi-hop QA	Synthesized from different corpus	36k	-	Generate single-hop questions based on different corpus by LLM. Extend to multi-hop questions via depth-based and width-based extension. Filter out those with shortcuts.
			Web		
WebArena	QA-like & task-oriented web interaction	Custom web environments (shopping, email, forum, map, social media)	7 environments, 812 tasks	Task success rate	Provide realistic multi-page websites. Annotators design diverse tasks requiring navigation, reasoning and interaction.
AgentBench	Open-ended web tasks with tool use	Real-world web APIs and websites	8 domains, 2000+ tasks	Success rate, human eval	Collect tasks from multiple domains (travel, shopping, QA, etc.). Provide tool APIs and human-verified success criteria.
GAIA	Complex open-domain information-seeking	Live web environment	466 tasks (300 retained answers)	F1 score, factual accuracy	Ask annotators to design multi-step questions requiring reasoning, planning and external search. Include hidden evaluation sets to test real-time retrieval.
BrowseComp	Fact-seeking QA over web browsing	Internet (open web), human-crafted QA	1,266 questions	Exact match	Questions designed so answer is short and verifiable. Human annotators ensure difficulty (not solved by existing models, not in top search results), enforce time/effort thresholds.
WebWalkerQA	Multi-hop QA via web navigation	Real Wikipedia + open web	680 questions	Exact match, F1 score	Generate multi-hop QA pairs requiring active web navigation. Filter with LLM-based difficulty control and human verification.
Amazon-Bench	E-commerce	Live Amazon.com webpages	400 user queries across 7 task types	Task success rate, harmful/benign failure rate, efficiency	Explore and categorize 60k+ Amazon pages. Sample diverse pages by functionality score, then prompt LLMs to generate realistic user queries and refine them to make them sound more natural and user-like.
			Software Engineering		
SWE-bench	Generate a pull request (PR) to solve a given issue	GitHub issues from 12 Python repositories	train 19k, test 2294	Unit test pass rate	Select PRs that resolve an issue and contribute tests. Keep only those that install successfully and passes all tests.
RepoBench	Code retrieval, code completion	Github-code dataset, Github Python and Java repositories	Python 24k, Java 26k	Golden snippet matching, line matching	Random sample lines as completion goals (with a first-to-use subset). Extract candidate snippets based on import codes, and annotate golden snippets.
DevEval	Repository-level function completion	Popular repositories from PyPI	1874	Unit test pass rate, recall of reference dependency	Select functions with test cases from repositories. Ask annotators to write requirements and reference dependencies. Filter out those with no cross-file dependency.
			Machine Learning		
MLAgentbench	Improve the performance metric by at least 10% over the baseline in the starter code	Kaggle	13	Success rate of 10% improvement, total time and tokens	Manually construct task description, starter code and evaluation code.
MLEbench	Achieve the best score on a metric pre-defined for each competition	Kaggle	75	Test score compared on leaderboard (e.g. medals)	Crawl task description, dataset, grading code and leaderboard from Kaggle website. Keep only those reproducible and up-to-date. Manually label the category and difficulty.
			Medical		
MedQA	Four-option multiple-choice question	National Medical Board Examination	train 48.9k, dev 6.1k, test 8.1k	Exact match	Collect question-answer pairs from the National Medical Board Examination.
MedMCQA	Four-option multiple-choice QA resembling medical exams	Open websites and books, All India Institute of Medical Sciences, National Eligibility cum Entrance Test	train 18.2k, dev 4.2k, test 6.2k	Exact Match	Collect question-answer pairs from medical examinations. Use rule-based method to preprocess the data. Spilt the dataset by exams (the training set consists of questions from mock and online exams, while the developing and test set consists of questions from formal exams.)
Quilt-VQA	VQA (Vision question answering)	Educational histopathology videos in Youtube	**Image-dependent:** 1055 **General-knowledge:** 255	LLM evaluation	Localize the "?"s in the video's transcript. Extract the relevant texts and images. Prompt GPT-4 to generate QA pairs. Perform a manual verification.
PathVQA	VQA	Electronic pathology textbooks and Pathology Education Informational Resource Digital Library website	**Images:** 4998 **QA pairs:** 32799	Accuracy(yes/no questions), exact match, Macro-averaged F1, BLEU	Extract images and their captions from the data sources. Perform natural language processing of the captions to break a long sentence into several short ones and get POS tagging. Generate open-ended questions based on POS tags and named entities.
PMC-VQA	VQA	PMC-OA	**Images:** 149k, **QA pairs:** 227k	BLEU, accuracy	Prompt ChatGPT with the images and captions to generate QA pairs. Perform LLM-based and manual data filtering.
PathMMU	VQA	PubMed, EduContent, Atlas, SocialPath, PathCLS	**Images:** train 16312, val 510, test 7213 **QA pairs:** train 23041, val 710, test 9677	-	Extract image-cpation pairs from the data source. Prompt GPT-4V to generate detailed description of images and then three questions per image. Perform expert validation.
			Legal		
LegalBench	Issue-spotting, rule-recall, rule-application and rule-conclusion, interpretation, rhetorical-understanding	Existing datasets, in-house datasets	9.1k	Accuracy, human evaluation	Filter and restructure the data from the data sources.
LegalBench-RAG	Retrieve snippets from legal corpora	LegalBench, PrivacyQA, CUAD, MAUD, ContractNLI	6889	Recall@k, precision@k	Start from LegalBench queries. Trace back each query's context to its original document span in the corpus. Final dataset pairs each query with its exact evidence.

Table 5: Details of domain-Specific Agentic RAG benchmarks. The metrics of QA are generally string matching (exact or fuzzy) or F1 score, and are omitted in the table. References are in Table 3.