Information-Seeking Agents in the Wild: Approaches to Evaluation in Dynamic and Open Environments

Preetam Prabhu Srikar Dammu

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2026

Reading Committee:
Chirag Shah, Chair
Lucy Lu Wang
Aylin Caliskan

Program Authorized to Offer Degree:
Information School

University of Washington

Abstract

Information-Seeking Agents in the Wild:
Approaches to Evaluation in Dynamic and Open Environments

Preetam Prabhu Srikar Dammu

Chair of the Supervisory Committee:
Chirag Shah
Information School

Information-Seeking Agents (ISAs) stand to reshape longstanding information access practices, ranging from question answering and fact verification to search, synthesis, and decision support. Traditional information access systems have typically left users responsible for formulating queries, inspecting sources, revising assumptions, and assembling evidence into usable knowledge. ISAs shift more of this work to agentic systems designed to fulfill users' informational needs through retrieval, reasoning, summarization, and tool use. Yet the evaluation methods used to measure these systems have not kept pace with this shift. Static QA benchmarks, fixed test items, and task-specific metrics can no longer fully capture systems that operate over changing information environments, use external tools at inference time, and combine parametric memory with retrieved evidence. Without new evaluation methods, measured performance may obscure the difference between reasoning and recall, evidence access and evidence use, or genuine information seeking and retrieval of leaked benchmark artifacts.

This dissertation develops approaches for evaluating ISAs in dynamic and open information environments. Concretely, it makes three contributions to the methodological foundations of ISA evaluation. First, it characterizes benchmark leakage as a multi-channel validity threat, showing that static QA evaluation can be compromised both by pretraining contamination and by retrieval-time access to benchmark artifacts. Second, it introduces dynamic benchmark construction methods that generate grounded question-answer instances from explicit evidence structures, reducing memorization risk while preserving comparability across evaluation runs. In structured knowledge settings, this is achieved through regeneration from semantically coherent knowledge-graph subgraphs; in open-web settings, it is achieved through traffic-driven topics, query-conditioned evidence, and auditable intermediate artifacts. Third, it develops an evaluation framework for cross-source sensemaking, where success requires not only retrieving relevant evidence but integrating information across sources, themes, and relationships. Taken together, these contributions argue for a shift from static answer matching toward contamination-aware, evidence-grounded, and dynamically adaptable evaluation. They point toward a future in which ISAs are measured not only by whether they produce correct answers, but by whether they reliably use the right evidence, under the changing conditions in which real information seeking takes place.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I am fortunate to be among those few who have been given the time and freedom to pursue questions they believe to be meaningful. I am glad to have the opportunity to express my gratitude to the many people and institutions whose guidance, support, and generosity made this work possible.

I want to begin with my advisor and mentor, Dr. Chirag Shah. I am deeply grateful for the time, energy, and care he has invested in my growth, not only as a researcher, but also as a person. He has created opportunities, advocated on my behalf, challenged me to grow, and given me the freedom to try new directions. His belief in me has pushed me to think bigger than I would have on my own, and it will continue to shape the kind of scholar I hope to become. I consider myself lucky to have him as a mentor, and I look forward to continuing to learn from him for many years to come.

I am also grateful to my dissertation committee: Dr. Lucy Lu Wang, Dr. Aylin Caliskan, and Dr. Carole Lee. I am thankful for the time they invested in reading my work, asking thoughtful questions, and helping me sharpen the ideas in this dissertation.

As part of the InfoSeeking Lab, I had the opportunity to meet several remarkable researchers and students whom I am now glad to call friends. I am especially grateful to Himanshu Naidu, who spent many hours helping us push through ambitious experiments and deadlines. I would also like to thank Dr. Shawon Sarkar and Dr. Yunhe Feng for making the lab a warm and welcoming place, and for offering guidance during my early projects. I also cherish the discussions and collaborations I shared with Maryam Amirizaniani, Kirandeep Kaur, Mouly Dewan, Harshita Chopra, and Michael Theologitis.

I was also fortunate to work with Dr. Tanushree Mitra, who generously gave her time,

guidance, and multiple quarters of funding. That support led to one of the most meaningful projects of my Ph.D. I am also grateful for the friendships and collaborations that grew out of this work. I have especially fond memories of working with Hayoung Jung, and I also enjoyed collaborating with Anjali Singh and Dr. Monojit Choudhury.

Beyond research, I also learned a great deal from the faculty at the University of Washington. I am especially thankful to Dr. Aylin Caliskan and Dr. Carson Miller-Rigoli, who taught me important lessons about teaching through my practicums. They generously gave me opportunities to lead class sessions and develop as an instructor, experiences that I will remember fondly.

I have been immensely blessed by my family. I am deeply grateful to my parents, whose decades of hard work created the experiences and opportunities that shaped my journey. My father, Dr. Srikar Dammu, taught me lessons early in life that continue to guide my decisions. My mother, Dr. Glory Nirmala, has always been a source of strength, and her prayers have been a constant source of comfort and hope. I am also grateful to my brother, Dr. Evan Dammu. Growing up with him by my side taught me things I could not have learned anywhere else, and I am thankful for the many ways he has always had my back. I am also grateful to my sister-in-law, Priscilla Shiny, for the care and joy she brings to our family. My grandmother, Jayamma Dammu, continues to inspire me with her strength and resolve, and her love and support helped me feel steady and cared for during my undergraduate years. I am also grateful to the many loving aunts, uncles, cousins, and friends who have supported me at different points in this journey.

I have also been blessed with a loving extended family. Sheba and I are especially lucky to have Shirley Baichan, my sister-in-law, in our lives. She made our move to Seattle feel far less overwhelming than it could have been, and in many ways, more memorable. Even before we arrived, she had already taken care of so many things so that we could begin our stay here comfortably. Since then, she has continued to look after us with the same care,

generosity, and love. I am also thankful to my parents-in-law, Alexander Emmanuel and Elizabeth Alexander, whose love and prayers have been a constant blessing in our lives.

Finally, I am grateful to my wife, Sheba Baichan. I would not have reached this point without her support, patience, and love. She has been my greatest source of strength through this journey, making sacrifices along the way, adjusting her own plans around my milestones, and caring for me through the chaos of deadlines and uncertainty. Her contribution to this moment began long before I entered the Ph.D. program. For more than a decade, she has encouraged me and stood by me through different stages of life and through my changing interests, always believing in what I was trying to build. Her love and support have remained constant, and I look forward to the many chapters of life I will share with her.

Chapter 1

INTRODUCTION

Recent advances in generative models have fundamentally transformed how information is accessed, processed, and used [13]. Traditional information access systems were largely user-driven, placing the user firmly in control of tasks such as generating ideas, making plans, adjusting to failed assumptions, and forming opinions. In contrast, we are now witnessing the emergence of agentic systems capable of autonomously handling many of these higher-order cognitive tasks [158, 186, 137, 61, 104, 101, 43, 112, 2, 178, 102].

This work refers to agentic systems that are specifically designed to fulfill a user's informational needs as *Information-Seeking Agents (ISAs)*. These agents focus on tasks such as question answering, multi-hop retrieval, fact verification, and knowledge aggregation. While the term agent encompasses a broad range of applications – from coding assistants to game-playing bots – ISAs are distinguished by their emphasis on information-centric reasoning and synthesis. They represent a natural progression from traditional information retrieval systems, enhanced by the planning and adaptivity afforded by modern generative architectures.

Over the past decades, various dedicated tools were designed to address users' informational needs. For instance, search engines help users find relevant documents, automated fact-checkers assist in verifying information, and summarization tools enable users to digest large amounts of information more efficiently. Each of these is a research problem in its own right, with multiple solutions explored for each. To measure progress, task-specific evaluation metrics were developed based on the requirements of each domain. For example, mean average precision (MAP) is used for document ranking, accuracy for fact-checking, and scores like

BLEU [119] and ROUGE [96] for summarization. However, agents now present themselves as unified solutions, capable of handling all the aforementioned tasks – and more. Notably, they can even chain multiple tasks together to achieve higher-level goals. Evaluating such systems is non-trivial, and no clear solution currently exists, underscoring the need for further research into suitable benchmarks and metrics.

Despite the lack of robust evaluation frameworks, such agentic systems are already being deployed in real-world applications and are experiencing growing adoption among users [115, 7, 57, 3]. This creates a pressing measurement problem. As ISAs increasingly mediate how users search, interpret, compare, and act on information, evaluation cannot be limited to whether a system produces a fluent or seemingly correct response. It must also ask whether the agent accessed appropriate evidence, used that evidence faithfully, handled temporal and contextual constraints, and fulfilled the user's informational need in a reliable way. This dissertation addresses that problem by developing evaluation approaches for ISAs operating in dynamic and open information environments, where answers may depend on changing sources, multi-step evidence use, and reasoning processes that are difficult to capture with static task-specific benchmarks.

1.1 Challenges with Evaluating ISAs

Unlike traditional machine learning systems – where problems are often well-defined and supported by stable, task-specific benchmarks – agentic systems are inherently more open-ended, customizable, and multi-purpose. This flexibility makes it significantly more challenging to design evaluations that are both comprehensive and reliable. Consider, by contrast, the case of image classification: the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [42] played a pivotal role in advancing deep learning by offering a clear, standardized benchmark. Image classification witnessed sustained progress over several years, with successive improvements from models like AlexNet [84] to ResNet [65], all while the underlying benchmark remained stable and widely accepted. There are a few reasons why this worked. First, the task was fixed: predicting image labels from a well-defined, static

dataset. Second, all methods were trained, validated, and tested on the same dataset splits, allowing for consistent comparisons. Despite significant research interest in the agentic space, no comparable benchmark has emerged for this class of systems.

Notably, this lack of a unified benchmark is evident in how widely-used agentic frameworks are currently evaluated. Take ReAct [175], for instance – a popular framework that not only serves as a standard baseline but also provides the foundational structure for a range of subsequent works in this space [27, 169, 138, 100, 165]. It is evaluated across a heterogeneous set of tasks, including question answering (HotPotQA [171]), fact verification (FEVER [148]), text-based games (ALFWorld [139]), and web navigation (WebShop [174]). Each of these represents a fundamentally different problem space with its own task-specific benchmark and evaluation criteria. Moreover, this list is far from exhaustive. ReAct and its variants have been extended to handle multimodal inputs and interact with hundreds of external tools and APIs [169, 27, 124]. Further complicating matters, the underlying generative models powering these agents are often interchangeable and pretrained on differing or undisclosed corpora, undermining the assumption of shared training and test conditions. Substantial evidence shows that training data influences model behavior at deployment [22, 66, 182, 37], further emphasizing the need for robust evaluation frameworks that account for such variation.

A common theme among the datasets used to evaluate ISAs is that they are predominantly static and publicly accessible. A growing body of evidence suggests that several widely used benchmarks have already been contaminated, rendering them unreliable [185, 168, 189, 190, 54, 93]. Zhou et al. [185] refer to this issue as *benchmark leakage*, a growing concern as large foundation models are trained on web-scale corpora encompassing vast portions of publicly available internet content [168, 189, 156]. Furthermore, even simple paraphrasing has been found to degrade performance, underscoring the brittle nature of these systems and their reliance on memorization [189].

To overcome these limitations, dynamic benchmarks have emerged as a promising alternative [40, 190, 189, 106, 181]. Unlike their static counterparts, dynamic benchmarks are designed to resist memorization and provide a more accurate assessment of adaptive, context-

aware performance in evolving scenarios [40, 176, 190, 189, 131]. This shift is particularly important to ensure that reported evaluations reflect genuine advancements in capabilities rather than superficial performance gains.

However, existing dynamic benchmarks do not yet support truly live information. Most of them focus on resampling existing data points through perturbations or controlled variations. This presents a significant limitation. Performance associated with evolving, real-time information remains untested, even though ISAs are increasingly built to interact with APIs and search engines that provide up-to-date content.

While most agents share common components – such as perception, planning, memory, and action modules [164] – their implementations are often tailored to specific use cases. This leads to architectural differences and affects transferability across tasks, underscoring the need for evaluation methods that reflect an agent's intended functionality. However, many studies continue to rely on established benchmarks that may not completely align with an agent's actual capabilities. For example, ReAct [175] is capable of accessing APIs to retrieve real-time information, yet it was evaluated on HotPotQA [171], a static dataset published in 2018. Although ReAct can answer questions involving current events – or respond to entirely different information needs, such as querying yesterday's stock market performance – these strengths remain unexamined due to the limitations of the benchmark.

What makes agentic systems powerful is precisely what makes them difficult to evaluate: their ability to operate across tasks, adapt to user goals, and interact with ever-changing environments. Yet current evaluation practices often overlook these dimensions, favoring static, task-specific benchmarks that miss the full scope of agent behavior. Without evaluations that reflect this fluidity, we risk mischaracterizing their capabilities and limitations. Recognizing these gaps sets the stage for a more targeted inquiry into how agentic systems should be assessed, grounded in the functions they are intended to serve.

1.2 Question Answering as an Evaluation Interface

This dissertation uses question answering as the primary evaluation interface for studying ISAs. Natural-language questions act as a compact way to express informational needs. A user can ask for a fact, a comparison, a grounded explanation, a verification judgment, or a synthesis across sources. In each case, the question provides a common input format. The agent's response can then be evaluated for correctness, evidence support, adequacy, and faithfulness.

The QA framing also makes the evaluation system-agnostic. Different ISAs may rely on different internal pipelines, including retrieval, web search, knowledge graphs, APIs, tool use, planning, or multi-step synthesis. A QA-style protocol does not prescribe how the agent should operate internally; it specifies the observable task outcome that should be assessed. This is why the dissertation studies QA datasets and benchmarks while maintaining a broader information-seeking perspective. The question is the interface, but the object of evaluation is the agent's ability to satisfy an informational need using appropriate evidence and reasoning.

1.3 Research Questions

Here, we outline the key research questions that guide this dissertation.

RQ1: How can we assess whether an ISA is effectively fulfilling a user's informational need? Users often pose complex queries that require agents to coordinate multiple capabilities – retrieving relevant documents, summarizing them, reasoning across sources, and presenting answers in a coherent and digestible form. Existing benchmarks typically evaluate each of these subskills in isolation using task-specific metrics such as MAP for ranking or BLEU for summarization. However, users care primarily about the overall quality and accuracy of the final answer. This motivates the need for integrated evaluation methods that reflect end-to-end task success, rather than decomposed component performance.

RQ2: How can we design benchmarks for ISAs that prevent task completion through memorized content or unintended tool use? ISAs often rely on tools such as search engines and APIs to retrieve information from the web. At the same time, the generative models that power these agents are trained on massive web-scale corpora and periodically updated, which increases the risk that evaluation datasets – especially if publicly released – may be included in their training data. This creates the problem of benchmark leakage, where agents can complete tasks by locating ground-truth answers rather than engaging in intended processes such as reasoning, planning, or synthesis. To ensure a meaningful assessment of agentic behavior, benchmarks must be designed to resist data contamination or unintended tool use, and structured in ways that require problem-solving rather than surface-level retrieval.

RQ3: How can we evaluate an agent's ability to reason over live, evolving information? The use of generative agents for accessing real-time information is becoming increasingly common, particularly with the integration of web search and API-based retrieval. This marks a shift from previous models with static knowledge cutoffs. However, evaluating real-time capabilities poses a significant challenge – content is dynamic, which introduces variability in both the input space and expected responses. A key question is how to develop evaluation protocols that fairly and reliably test an agent's handling of current or ephemeral information.

1.4 Path Toward ISA Evaluation

The research questions above are connected by a common measurement problem: an ISA should be evaluated by whether it satisfies an informational need through appropriate evidence use, not simply by whether it produces a plausible final answer. This requires a progression from diagnosing failures in existing evaluation, to developing controlled alternatives, to testing agents in more realistic information environments.

The first step is to ask whether widely used static QA benchmarks still provide a reliable

measurement target for web-enabled systems. If a model has seen benchmark content during pretraining, or if a web-enabled agent can retrieve answer-bearing benchmark pages during evaluation, then high performance may reflect exposure rather than the intended information-seeking behavior. Before proposing new benchmarks, the dissertation therefore begins by auditing this validity threat and separating pretraining contamination from retrieval-time leakage.

The second step is to develop a more controlled evaluation mechanism once the limitations of static public benchmarks are established. A dynamic benchmark should produce fresh items, but freshness alone is not enough. The generated items must remain grounded, auditable, and comparable across runs. Structured knowledge provides a useful intermediate setting for this purpose because entities, relations, answer paths, and generation decisions can be inspected directly. This motivates the DYNAMIC-KGQA [39] study, which tests whether dynamic benchmark generation can reduce exposure while preserving evaluation consistency.

The third step is to move beyond the clean structured setting toward the open-web environments in which ISAs are increasingly used. In deployment, relevant evidence is often distributed across sources, changes over time, and requires synthesis rather than extraction from a single passage or traversal of a clean graph. This motivates iAGENTBENCH [41], which evaluates whether agents can organize and integrate query-conditioned web evidence across themes. Together, the three studies form a path from measurement validity, to controlled dynamic generation, to open-web sensemaking.

1.5 Dissertation Claims and Contributions

This dissertation makes four claims about the evaluation of ISAs:

1. Static QA benchmarks can mismeasure modern information-seeking systems because contamination and retrieval-time leakage can make performance appear stronger than the system's actual reasoning ability.

2. Dynamic benchmark generation can mitigate these failures while preserving evaluative

consistency, as long as the generation process remains controlled, auditable, and grounded in verifiable evidence.

3. Structured settings such as knowledge graphs provide an important intermediate step for building dynamic evaluation pipelines, because they make entities, relations, answer paths, and generation procedures inspectable.

4. Realistic ISA evaluation must move beyond passage retrieval to cross-source sensemaking, where the central challenge is not simply finding evidence, but integrating evidence across multiple sources, themes, and temporal contexts.

The contributions of this dissertation follow the pathway described above. Chapter 3 studies contamination as a measurement validity problem for static and web-enabled QA evaluation. It separates pretraining contamination from retrieval-time contamination and shows why static public test sets are not enough for evaluating systems with web access. Chapter 4 develops Dynamic-KGQA [39] as a controlled dynamic response: a benchmark generation framework over structured knowledge that supports consistency, auditability, and contamination resistance. Chapter 5 then moves to the open web through iAgentBench [41], a benchmark designed to evaluate sensemaking over query-conditioned evidence rather than single-passage extraction. Chapter 6 synthesizes these studies into design principles for future ISA benchmarks.

The intended contribution is not a single benchmark that solves evaluation in all settings. Rather, this dissertation provides a way to reason about evaluation under different information regimes: static versus dynamic, structured versus open-web, single-source retrieval versus cross-source synthesis, and answer access versus evidence integration.

1.6 Dissertation Overview

Chapter 2 provides the conceptual background for the dissertation. It connects foundational models of information seeking, including anomalous states of knowledge and berrypicking,

to modern ISA capabilities such as memory, query reformulation, clarification, tool use, and multi-step evidence gathering. It then develops the information-level and system-level considerations that shape ISA design: whether the relevant information is static or changing, structured or unstructured, directly accessible or mediated through tools and interfaces. The chapter concludes by reviewing related work on ISAs and their evaluation, setting up the progression from static benchmark validity to dynamic generation and open-web sensemaking.

Chapter 3 establishes the central evaluation problem. It situates contamination within the longer history of machine-learning evaluation practice and shows why static public QA benchmarks become fragile for web-enabled ISAs. The chapter separates two contamination channels: pretraining exposure, where benchmark content may appear in model training data, and retrieval-time exposure, where answer-bearing benchmark content can be surfaced through web search during evaluation. This diagnosis provides the first step in the dissertation pathway: before ISAs can be evaluated reliably, the measurement instrument itself must be audited.

Chapter 4 presents Dynamic-KGQA [39] as the first constructive response to that diagnosis. A knowledge graph setting is deliberately controlled: entities, relations, subgraphs, and answer paths can be inspected. This makes it a useful intermediate step for studying dynamic benchmark generation, consistency across runs, and contamination resistance before moving to less structured information environments.

Chapter 5 then moves from controlled structured evaluation to open-web sensemaking. iAgentBench [41] targets questions whose answers require integrating evidence across multiple sources and themes. This step extends the evaluation pathway toward more realistic ISA settings by testing the idea that evidence access alone is not enough: agents must also organize and synthesize evidence reliably.

Chapter 6 returns to the research questions and develops a design framework for ISA benchmark design and interpretation. Chapter 7 summarizes the dissertation's findings, situates the three studies within the broader ISA evaluation agenda, and discusses implications for IR, RAG, and agent evaluation.

Chapter 2
BACKGROUND AND RELATED WORK

This chapter introduces core concepts relevant to ISA design and evaluation. Section 2.1 discusses theoretical foundations of information seeking and explains how newer generative and agentic capabilities relate to long-standing problems identified in foundational IR work. Section 2.2 then develops the information-level and system-level considerations that shape ISA design: what kind of information is being sought, how that information is represented, and how it can be accessed. The chapter concludes by reviewing prior work on agents for information retrieval and their evaluation.

2.1 Information-Seeking Foundations

Information seeking begins from a gap between what a person knows and what a person needs to know. This gap is often difficult to express directly. A user may recognize uncertainty, feel that something is missing, or need information to complete a task, but still lack the vocabulary or structure needed to formulate a precise query. Classic information-seeking theories are useful here because they explain why information access is not simply a matter of matching a query to a document.

Belkin's Anomalous State of Knowledge (ASK) framework provides one of the clearest accounts of this problem [9]. In this view, an information need arises when a person's state of knowledge is anomalous with respect to a goal or problem. The user's request is therefore not a complete representation of the need, but an imperfect expression of a partial and unstable cognitive state. Belkin noted that this framing introduced both "conceptual and practical difficulties," yet it also pointed toward a more realistic model of information retrieval: systems should not only match query terms, but help transform an uncertain state of knowledge into

a more complete one.

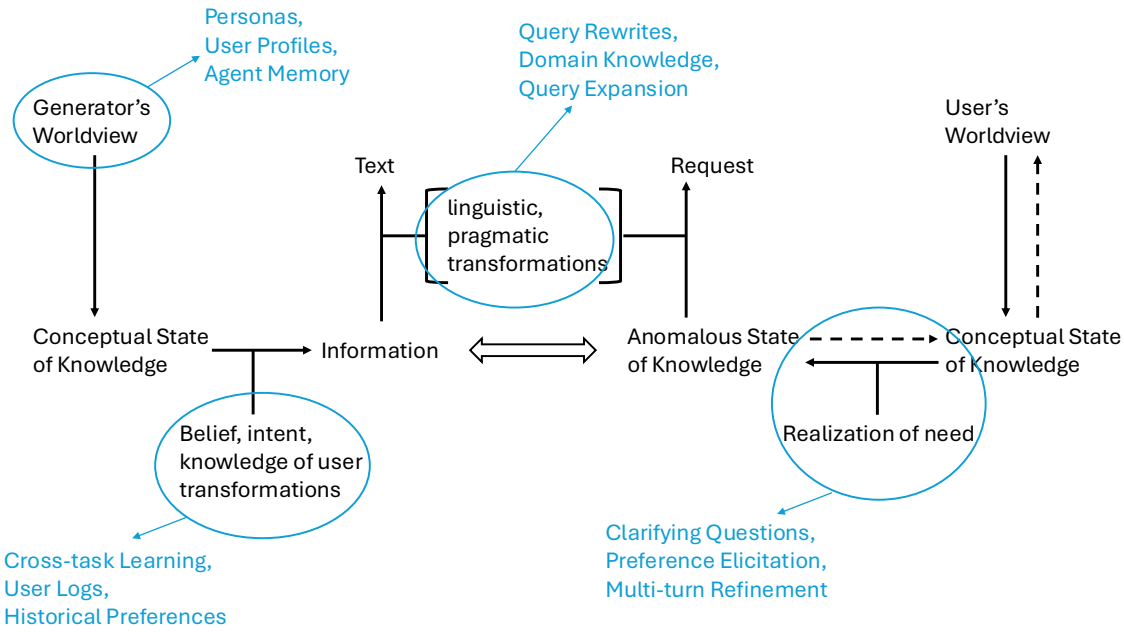


Figure 2.1: Mapping Belkin's ASK framework [9] to modern ISA capabilities. The annotations highlight how memory, user modeling, query rewriting, clarification, and preference elicitation can support different stages of information-need articulation and resolution.

This observation is especially relevant for modern information-seeking agents. Earlier retrieval systems had limited ability to model the user, revise the request, or interact over multiple turns. Generative and agentic systems now make some of these operations more feasible. They can maintain user profiles, personas, and memory representations [21, 120, 49, 136]; learn from logs, prior tasks, and historical preferences [80, 136, 180]; rewrite queries and expand them with domain knowledge [73, 55, 105]; and ask clarifying questions or elicit preferences over multiple turns [122, 38, 83]. Figure 2.1 illustrates this connection by overlaying these capabilities onto Belkin's model.

The ASK framework also clarifies why evaluating ISAs is difficult. If the system's role

is to help resolve an anomalous state of knowledge, then evaluation cannot be limited to whether the final response matches a reference answer. It must also consider whether the agent understood the need, gathered appropriate evidence, refined the request when necessary, and produced an answer that actually reduced the user's uncertainty.

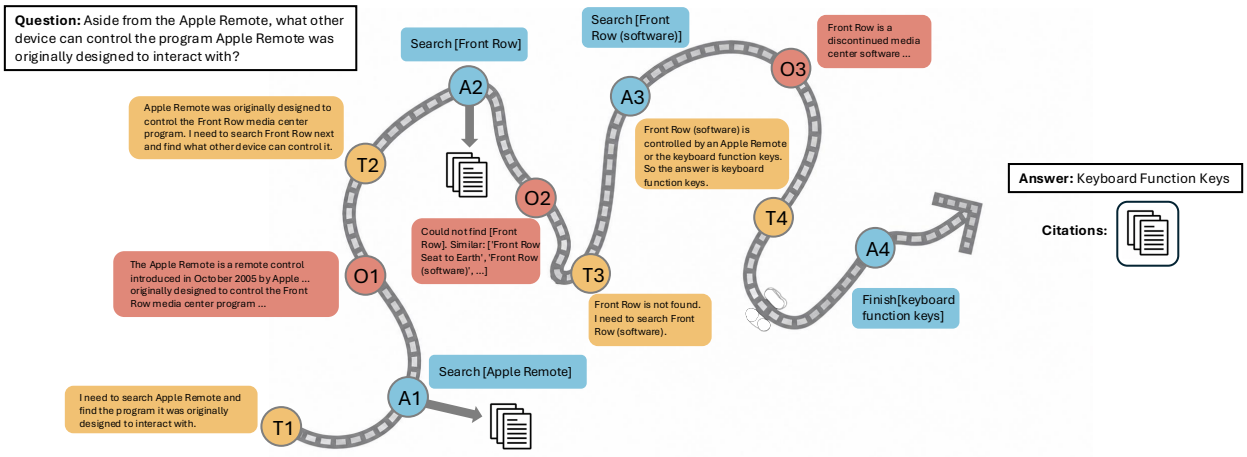


Figure 2.2: Connecting Bates' berrypicking model to a ReAct-style [175] QA trace. The example shows how agentic systems can iteratively gather, follow, and synthesize evidence across multiple steps on a HotPotQA [170] instance.

Bates' berrypicking model offers a complementary view [8]. Rather than treating search as a single query followed by a final answer set, Bates describes information seeking as an evolving process in which users collect useful pieces of information, follow new leads, reformulate their queries, and gradually move toward a satisfactory understanding. This model is useful for ISAs because many agentic systems now perform a computational analogue of this process: they search, inspect partial evidence, update their working state, issue follow-up queries, and synthesize information across steps.

ReAct-style agents make this connection visible because they interleave reasoning and action [175]. In a multi-hop QA setting, an agent may search for one entity, observe partial evidence, identify a missing connection, search again, and continue until the answer is

supported. Figure 2.2 shows this process using a HotPotQA [170] instance. However, it is important to note that current agents do not reproduce human information seeking, even when both involve iterative movement through an information space. In human-controlled search, the intermediate steps are not incidental. As users inspect results, compare sources, reformulate queries, and follow unexpected leads, their own state of knowledge changes. They may learn what terminology to use, which sources to trust, what assumptions were wrong, or which adjacent questions matter. When this process is delegated to an agent, the agent may update its working context, but the user's learning can be reduced to the final response. This changes the role of search from an interactive process of exploration and sensemaking into a compressed act of answer delivery.

This matters because search is not only a mechanism for answer retrieval. Intermediate steps performed during a search session also support verification, information literacy, and serendipity, which can be weakened when systems collapse the search process into a generated response [135]. From this perspective, the agent's action trace is a computational analogue of berrypicking, not a replacement for it. It captures iterative movement through an information space, but not the user's learning through that movement. This distinction matters for ISA evaluation: agents should be assessed not only by whether they reach a correct answer, but also by whether their evidence-gathering process remains visible, inspectable, and useful to the user whose informational need they serve.

Furthermore, it is important not to assume that all requests to ISAs can be addressed through direct plain-text answers. Web search has long involved different kinds of intent. Broder [17] distinguishes between *informational* queries, where the user seeks to learn or acquire information; *navigational* queries, where the goal is to reach a particular site or page; and *transactional* queries, where the goal is to carry out a web-mediated activity such as shopping, downloading, booking, or accessing a service. This distinction matters because a user's need is not always satisfied by generating an answer string. Sometimes the appropriate response is an explanation, sometimes it is a source or destination, and sometimes it requires interaction with an external system.

For ISAs, these boundaries become even less stable. A single request may require explanation, retrieval, comparison, tool use, and action depending on what kind of information is involved. A question about a policy may require source-grounded explanation, a request about flight prices may require live structured data, a shopping task may require comparison and transaction support, and a developing news event may require temporally grounded evidence across multiple sources. Thus, the evaluation of ISAs depends not only on user intent, but also on the information landscape in which the agent operates: whether the relevant information is static or changing, structured or unstructured, authoritative or uncertain, directly accessible or mediated through tools and interfaces. The next section develops these information-level and system-level considerations.

2.2 Information Landscape and Design Considerations

The previous section focused on information seeking as a user- and process-level activity. For ISAs, this process must also be understood through the information environment in which the agent operates. The design of an ISA is not determined only by the user's request, but also by the properties of the information needed to satisfy that request and by the mechanisms through which that information can be accessed.

These two factors are related but distinct. Intrinsic information properties describe the nature of the information itself: whether it is static or changing, structured or unstructured, verifiable or uncertain, objective or subjective, and so on. Access and storage conditions describe how that information is made available to the system: whether it can be queried through an API, retrieved from a closed document collection, searched on the live web, extracted from a database, or obtained only through interactive navigation. Two information needs may share the same intrinsic property but require very different system behavior. For example, both stock prices and dynamic ticket prices are volatile. A stock quote may be available as a single structured value through a finance API, while a ticket price for a specific Lakers game may require navigating a website, selecting a date, choosing seats, and reading prices that are only revealed through interface interaction. In the first case, the core challenge

is reliable structured lookup. In the second, the challenge includes perception, navigation, state tracking, and interaction with an external system.

This distinction is central for ISA evaluation. Benchmarks that treat all information needs as answer-generation problems risk hiding the actual capability required: retrieving from a fixed corpus, querying live sources, navigating an interface, or synthesizing across many documents. The following subsections separate these concerns. Section 2.2.1 discusses intrinsic properties of information, while Section 2.2.2 presents a complexity ladder that connects information properties and access conditions to concrete ISA task types.

2.2.1 Intrinsic Properties of Information

Intrinsic properties are domain-agnostic characteristics that describe how information is formed, organized, and evolves over time. They dictate what an ISA must do and how it does it. For instance, serving video requires episodic memory to track temporal events, while image-centric tasks need a vision encoder for captioning or object recognition.

Because these properties shape both the tasks an ISA tackles and the modules it needs, they are a natural lens for judging robustness. Benchmarks that vary each property across its spectrum – from static encyclopedic facts to rapidly changing news streams – force agents to adapt to realistic informational scenarios.

Table 2.1 summarizes the ten crucial information properties that significantly affect system considerations. Each row lists a definition, an example spectrum, and the system-level challenges the property introduces.

Temporality. Temporality denotes the extent to which the relevance or truth value of a piece of information changes over time. At one extreme are essentially immutable facts – for example, mathematical constants – while at the other lie rapidly evolving streams such as breaking news or real-time sensor feeds. A system that ignores this axis risks returning stale, misleading, or incomplete evidence.

System Considerations: Retrieval and generation systems must integrate temporal signals

Table 2.1: Intrinsic Properties of Information

Property	Definition	Example Spectrum	Key Challenges
Temporality	Time-dependence of information's relevance or validity.	Static (theorem) ↔ Dynamic (news)	Time-sensitive indexing and ranking, query intent modeling, dynamic evaluation.
Structure	Degree of predefined organization or schema.	Structured (DB) ↔ Semi-structured (JSON) ↔ Unstructured (text)	Storage, indexing strategy, query formulation, hybrid retrieval.
Verifiability	Extent to which accuracy can be independently confirmed.	High (encyclopedia) ↔ Low (rumor)	Fact-checking, evidence retrieval, NLI, credibility modeling.
Source Authority	Perceived trustworthiness of the origin.	High (journal) ↔ Low (anon. blog)	Authority modeling, user trust, ranking scores.
Granularity	Atomicity or unit size of information.	Atomic (fact) ↔ Sentence/Chunk ↔ Document	Chunking, retrieval unit selection, context vs. noise.
Access Modality	Primary format or access protocol.	PDF ↔ API/JSON ↔ RDF	Multimodal IR, API access, vector search, reasoning.
Volatility	Rate of content change or update.	Low (history book) ↔ High (stock price)	Freshness, versioning, real-time handling.
Context Dependence	External context required for correct interpretation.	Low (definition) ↔ High (irony, jargon)	Disambiguation, personalization, pragmatics.
Subjectivity	Degree of opinion/emotion vs. factuality.	Objective (fact) ↔ Subjective (opinion)	Sentiment/stance detection, bias modeling.
Completeness	How fully the info meets needs or expectations.	Incomplete ↔ Fully Complete	Quality assessment, missing data handling.

at three levels. First, indexing infrastructures should retain fine-grained timestamps and support decay functions so that ranking models can balance topical similarity with freshness [47]. Second, query analysis must detect temporal intent, since user logs reveal distinct classes of atemporal, time-specific, and multi-episodic queries that each demand different treatment [76]. Third, evaluation protocols must simulate evolving collections; static test sets overestimate performance when recency matters. The TREC Temporal Summarization track introduced latency-sensitive metrics that reward early, non-redundant updates in a live

stream [153], while foundational work in temporal IR demonstrated measurable gains when temporal expressions in documents guide ranking [5]. Together, these findings show that temporality is not an optional feature but an important constraint that shapes architecture, ranking strategy, and benchmark design for ISAs.

Structure. Structure refers to the extent to which an information source conforms to an explicit schema or organisational framework. Highly structured data – such as relational tables or Resource Description Framework (RDF) triples – bind facts to fixed attributes and typed relations; semi-structured representations (XML, JSON) embed partial markup within otherwise flexible text; unstructured content (free-form prose, images, audio) lacks overt syntactic scaffolding. The greater the structure, the more a system can rely on symbolic access paths; the looser the structure, the more it must infer semantics from raw signals.
System Considerations: Structural variation drives fundamental engineering choices. Relational databases and knowledge graph engines exploit keyed joins and exhaustive triple indexes to support declarative languages such as SQL and SPARQL, yielding millisecond latency for precise field constraints [34, 113]. In contrast, unstructured corpora demand representation learning: dense-vector encoders approximate semantic similarity in high-dimensional space over large embedding stores [78]. Semi-structured formats sit between these poles: XML retrieval research shows how joint use of content and hierarchical tags improves element-level relevance estimates [50]. An ISA must map each incoming resource to the appropriate storage layer, choose a query execution strategy consistent with its structure, and fuse heterogeneous results into a coherent answer.

Verifiability. Verifiability measures the extent to which a claim can be confirmed – or refuted – by consulting independent evidence. Information of high verifiability supplies explicit provenance or can be cross-checked against well-curated references, whereas low-verifiability items (rumours, unsupported opinions) provide no reliable trail for corroboration. Crucially, verifiability is orthogonal to truth: a statement may be false yet readily verifiable if its cited

sources are accessible (and provably wrong), or true but unverifiable when no evidence is available.

System Considerations: ISAs that prioritize trustworthy outputs must ensure verifiability at every stage of the pipeline. In retrieval-augmented generation, non-parametric memory supplies source passages whose inclusion demonstrably improves factual specificity and enables post-hoc citation [92]. Verification datasets such as FEVER operationalise this requirement by forcing models to retrieve and explicitly present supporting or refuting evidence before classifying a claim [148]. Once candidate evidence is gathered, systems perform textual entailment or stance detection – tasks that rely on large-scale natural-language-inference resources like SNLI to train robust comparators [16]. Finally, evaluation must move beyond relevance alone: Rashkin et al. [129] introduce the AIS framework, which scores generated content only when it is attributable to an identified source. Together, these threads underscore that verifiability is an important design constraint: architectures require evidence-retrieval modules, inference engines to compare claims with evidence, and metrics that penalise unsupported assertions.

Source Authority. Source authority denotes the perceived expertise, reputation, and trustworthiness of an information provider. Early Web-scale retrieval systems operationalised this notion through link-analysis algorithms such as PageRank [117]. Even in social settings, people enlist their networks to help find, evaluate, and verify information on the Internet, using endorsements as heuristics for credibility [109]. The problem intensifies on open platforms like Twitter, where low-barrier publishing accelerates the spread of misinformation and false rumors [26].

System Considerations: Because authority shapes user trust, it must be encoded throughout the retrieval pipeline. Ranking models weight high-authority domains via link, citation, or reputation graphs, but these signals must adapt to adversarial environments in which malicious actors manufacture artificial prestige. Provenance identification – by URL patterns, organisational metadata, or user-level trust scores – feeds authority signals to rerankers and

result aggregators. Social-media credibility detectors combine content features (linguistic style, stance) with network features (retweet graphs, follower ratios) to discount low-authority cascades. Finally, evaluation protocols should report authority-conditioned precision, mirroring the trust dynamics documented by Metzger and colleagues. Without an explicit authority model, an ISA risks surfacing persuasive yet unreliable content, eroding user confidence and undermining downstream decisions.

Granularity. Granularity expresses the unit size at which information is stored, retrieved, and presented. At the fine end of the spectrum lie individual triples, sentences, or image patches; at the coarse end, entire documents or datasets. An ISA must choose a working granularity that preserves sufficient context for interpretation while avoiding the dilution and computational overhead that accompany very large units.

System Considerations: Granularity shapes every stage of the pipeline. Early passage-retrieval research showed that retrieving whole documents often dilutes topical focus, whereas passage indexing yields higher precision for fact-centric queries [23]. Subsequent evaluation work confirmed that window size is a critical hyper-parameter: too narrow and essential context is lost; too wide and noise erodes ranking quality [4]. Modern retrieval-augmented generation systems inherit the same trade-off. Lewis et al. [91] split Wikipedia into 100-token chunks because smaller passages reduce irrelevant content while still providing enough evidence for generation. More recent mixtures-of-granularity models dynamically blend document- and passage-level retrieval, learning to route queries to the unit size that maximises answer likelihood [184]. These findings imply that chunking, indexing, and reranking must be co-designed: aggressive decomposition can boost precision but increases latency and memory, while coarse units simplify infrastructure at the cost of answer exactness.

Access Modality. Access modality describes the native format or protocol through which information is encoded and queried. Text articles arrive as plain UTF-8 strings, encyclopedic triples reside behind SPARQL endpoints, Web services expose JSON via REST, images are

pixel arrays, while dense-vector stores index neural embeddings. Because each modality affords different operations – string matching, logical inference, nearest-neighbour search – an ISA must route requests to modality-appropriate parsers, indexes, and query languages.
System Considerations: Modality governs all layers of an ISA. Multimedia repositories rely on content-based retrieval: Smeulders et al. [140] observe that successful image search relies on visual content rather than textual annotations. When raw signals are projected into dense-vector space, scalable similarity search engines such as FAISS support billion-scale approximate nearest neighbour queries over embeddings [75]. Hybrid systems increasingly fuse these channels; long-form video is first transcribed to captions, enabling text-based reasoning over an essentially visual source [159]. Without modality-aware ingestion, indexing, and ranking, an ISA risks failing to exploit the expressive power of structured sources.

Volatility. Volatility captures the rate at which a source's content changes or becomes obsolete. Highly volatile streams – live stock quotes, microblog posts, breaking-news wires – may mutate minute-to-minute, whereas a digitised manuscript exhibits negligible change across decades. Because volatility determines how long a retrieved item remains valid, it becomes a core axis along which an ISA must schedule crawling, index maintenance, and ranking.
System Considerations: Rapidly changing collections impose architectural demands absent from static corpora. Cho and Garcia-Molina [32] formalize "freshness" as the fraction of local replicas identical to their on-line originals and show that incremental crawling policies can trade bandwidth for timeliness. Once new documents arrive, inverted indexes must be updated without full reconstruction; dynamic-indexing schemes achieve this by allowing an inverted index to be updated incrementally while maintaining query performance [88]. Ranking models must then blend topical relevance with up-to-date signals: Dong et al. [47] add freshness features and observe significant improvements for time-sensitive queries in web search. Finally, evaluation protocols must mirror the stream's pace. The TREC Microblog track requires systems to return the most recent and relevant tweets quickly after posting,

thereby rewarding latency-aware retrieval [116, 97].

Context Dependence. Context dependence denotes the extent to which an utterance or document requires extra-linguistic, situational, or discourse knowledge for correct interpretation. Sarcasm, domain-specific jargon, and elliptical follow-ups in dialogue all exhibit high context dependence, whereas a standalone mathematical formula typically does not.
System Considerations: Retrieval and question-answering pipelines that ignore context risk dramatic precision loss. Conversational search studies show that the interpretation of each query depends on the dialogue history and therefore must be modelled across turns [126]. Within documents, resolving coreference is impossible without document-level context. Lee et al. [87] demonstrate significant accuracy gains when mention links can leverage broad context. In agentic systems, explicit persona prompts help LLMs maintain discourse coherence over multiple turns [28], while memory streams enable agents to accumulate episodic context that guides future actions [120].

Subjectivity. Subjectivity gauges how strongly an information item is coloured by personal attitudes, emotions, or beliefs rather than by verifiable fact. Product reviews, political commentary, and social-media posts are inherently subjective, while an artifact such as a product specification sheet is inherently objective.
System Considerations: Correctly separating subjective from objective content is pivotal for modern retrieval and question-answering pipelines. Wiebe et al. [162] demonstrate that subjective sentences express opinions and other private states, whereas objective sentences present factual information. Pang and Lee [118] exploit this distinction by filtering out objective sentences, yielding higher sentiment-classification accuracy. From a user-centric angle, [68] show that shoppers value subjective reviews because they convey experiential judgments unavailable in objective specs. Consequently, an ISA must (i) detect sentiment, stance, and bias, (ii) tailor ranking to the user's preference for subjective or objective evidence, and (iii) ensure viewpoint diversity where appropriate.

Completeness. Completeness measures the extent to which a retrieved or stored item supplies all information required to satisfy an information need. A document that omits critical metadata or elides key argumentative steps is incomplete, forcing the user (or downstream model) to seek supplemental evidence elsewhere.

System Considerations: Completeness is a key dimension of data quality. Incomplete information, such as missing metadata or content fields, can substantially impair retrieval effectiveness by reducing indexing precision, disabling filtering mechanisms, or weakening ranking models [152, 20]. In downstream applications, missing or partial data can lead to biased outcomes, inaccurate inferences, or model failures. IR systems must therefore develop strategies to detect and manage incompleteness, whether through imputation, probabilistic modeling, or external data augmentation. Additionally, users often expect comprehensive, well-rounded results; failure to meet these expectations can negatively impact user satisfaction and trust in the system . Finally, evaluation in the presence of incomplete information poses significant challenges, most standard IR metrics assume full relevance judgments and may not yield reliable results when data is sparse or partially annotated [152, 20].

Interplay and Overlap of Properties It is crucial to recognize that intrinsic properties of information are not entirely independent or orthogonal. Instead, they often interact and correlate in meaningful ways that influence how information is retrieved, processed, and evaluated. For instance, volatility and temporality are closely linked. Information that changes frequently exhibits high volatility also tends to have time-sensitive relevance. This temporal aspect affects both how current the information remains and how it should be ranked or filtered in retrieval scenarios [24]. There is also a strong relationship between structure and context dependence. Highly structured data, such as relational tables or annotated datasets, generally has lower context dependence due to the presence of explicit semantics and schemas. In contrast, unstructured text often relies heavily on surrounding context for interpretation, requiring more sophisticated natural language understanding for accurate retrieval [110]. Further, the properties of verifiability, subjectivity, and source authority

are deeply intertwined. Information from high-authority sources, especially when presented objectively, tends to be more verifiable. On the other hand, subjective or anonymous content often lacks the evidence or transparency required for verification. The interaction between granularity and context also poses important trade-offs. Finer-grained retrieval units, such as individual sentences or facts, may lack sufficient context. In contrast, coarser-grained units, like full documents, risk including extraneous or irrelevant information. Choosing the appropriate level of granularity is therefore critical to balancing precision and comprehensiveness in retrieval systems [30].

2.2.2 A Complexity Ladder for Information-Seeking Tasks

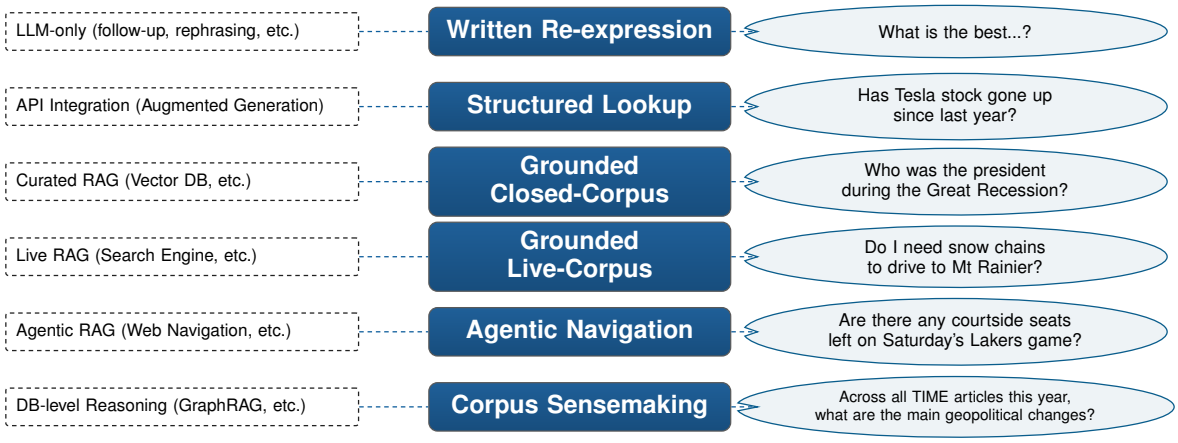


Figure 2.3: A complexity ladder for information-seeking tasks. Each level reflects a different combination of information properties, access conditions, and ISA capabilities required to satisfy the user's need.

The intrinsic properties above describe what kind of information is involved. A second question is how difficult it is for an ISA to access and use that information in practice. I use a complexity ladder to distinguish common information-seeking task regimes. The ladder is

not a ranking of importance; rather, it separates tasks by the combination of information properties, access conditions, and system capabilities required to answer them.

Table 2.2: Representative Examples of Each Level of the Complexity Ladder.

Level	Representative examples	Information setting	Primary ISA capability tested
Structured Lookup	Current UV index in Seattle; current USD–EUR exchange rate; elevation of Mount Rainier	Single structured value, often from a database, knowledge base, or API	Reliable lookup, freshness handling, and minimal inference
Grounded Closed-Corpus	Coffee break time in a provided tutorial schedule PDF; GPU request instructions in a provided FAQ; deductible in a provided insurance summary	Bounded source collection supplied to the agent	Retrieval within a closed corpus, evidence grounding, and faithful extraction
Grounded Live-Corpus	Current UHC claim processing time; whether tire chains are needed for Mt. Rainier tomorrow; latest stable Python release; current WA PFML eligibility threshold	Live web sources with changing or time-sensitive information	Source selection, recency handling, temporal grounding, and evidence citation
Agentic Navigation	Reservable campsite availability at Cougar Rock Campground for specific dates; dynamic ticket pricing revealed through a booking or seat-selection interface	Information is accessible only after interaction with a website, form, or interface state	Planning, tool use, interface navigation, state tracking, and action grounding
Corpus Sensemaking	Top recurring complaints across 200 product reviews by month; main policy themes and disagreements across 50 articles on AI regulation	Many documents or records whose answer emerges from aggregation and synthesis	Corpus coverage, theme extraction, comparison, aggregation, and cross-source synthesis

At the lower end are tasks where the relevant information is already structured and can be returned as a single value. For example, the current USD–EUR exchange rate or the current UV index in Seattle can typically be answered through an API. These tasks may be time-sensitive, but they do not require broad synthesis or navigation. Next are grounded closed-corpus tasks, where the answer must be extracted from a provided source such as a PDF, FAQ, insurance plan summary, or data dictionary. Here the core requirement is faithful grounding in a bounded evidence set. Grounded live-corpus tasks add volatility and open-web retrieval: the system must identify current authoritative sources, handle temporal constraints, and cite evidence that may change over time. Agentic navigation tasks require interaction

with an external interface, such as checking campsite availability or selecting seats on a ticketing site. Finally, corpus sensemaking tasks require broad coverage and synthesis across many items, such as identifying recurring complaints across product reviews or summarizing disagreements across a folder of news articles.

This ladder helps explain why ISA evaluation cannot be reduced to a single benchmark format. A short question may still require live retrieval, while a seemingly factual request may require interaction with an external system. Conversely, a volatile question can be simple if the answer is exposed through a reliable structured API. What matters is the joint configuration of the information need, the information properties, and the access path. A benchmark for structured lookup should test freshness and correctness of a returned value; a closed-corpus benchmark should test grounding and attribution; a live-corpus benchmark should test source selection and temporal validity; an agentic-navigation benchmark should test whether the system can interact with the environment; and a sensemaking benchmark should test whether the agent can synthesize across a sufficiently broad evidence set.

Furthermore, selecting the appropriate level of ISA capability is not only a matter of performance or cost. It also affects reliability and safety. When the needed information is available through a structured API or curated feed, that access path can reduce ambiguity about provenance, freshness, and data format. Even if an agent could obtain the same value through open-web search or navigation, doing so may introduce unnecessary failure modes, including stale pages, misleading sources, adversarial content, source ambiguity, and misinformation or disinformation. Thus, the ladder should be read not only as a measure of task complexity, but also as a design principle: use richer agentic behavior when the information need requires it, but prefer structured, auditable, and lower-risk access paths when they are sufficient.

2.3 Related Work

This section reviews prior work on ISAs and their evaluation. It first organizes agentic systems by the information environments they operate in, then discusses how existing benchmarks

evaluate these systems and where they fall short for dynamic, evidence-grounded information seeking.

2.3.1 Information-Seeking Agents and Information Environments

ISAs can be differentiated by the information environments in which they operate. Some agents retrieve relatively stable facts from curated knowledge graphs, tables, or document collections. Others search the live web, navigate websites, maintain memory over time, reason over socially contested information, or perceive situated visual environments. These distinctions are not only architectural; they reflect differences in the underlying information properties discussed above, including temporality, structure, volatility, verifiability, source authority, subjectivity, and access modality.

Table 2.3 summarizes five recurring ISA types in the literature: *Static*, *Dynamic*, *Persistent*, *Social*, and *Embodied*. These categories are not intended as a strict partition. Many deployed systems combine multiple types, and newer agents increasingly mix web search, tools, memory, multimodal perception, and long-horizon planning. The taxonomy is used here to organize the design and evaluation challenges that arise when agents operate over different information environments.

Static ISAs operate over relatively stable information sources, such as curated knowledge graphs, tabular data, legal corpora, encyclopedic documents, or fixed multimedia collections. Their core challenge is to retrieve and reason over information that is expected to remain valid within the evaluation setting. Systems such as ChatLaw, Think-on-Graph, Generate-on-Graph, and AutoTQA illustrate this regime, where the agent often uses retrieval, semantic parsing, graph traversal, or table reasoning to assemble an answer from structured or semi-structured evidence [36, 142, 166, 188]. Multimodal systems such as ViperGPT and ChatVideo broaden this setting to images and videos, but still often rely on fixed datasets or bounded inputs [144, 157]. Static ISAs are important because they make grounding and reasoning easier to inspect, but they also inherit the limitations of static benchmarks: fixed corpora can become stale, memorized, or misaligned with real-world information needs.

Table 2.3: Taxonomy of ISAs organized by information environment, core function, key information properties, and common architectural affordances.

Agent Type	Core Function	Key Information Properties	Architectural Affordances
Static ISAs	Retrieve and ground factual or encyclopedic information from stable sources	Temporality: Static; Structure: Structured / semi-structured; Verifiability: High; Authority: High	Dense/sparse retrieval; Reranking; Citation mapping; Semantic parsing
Dynamic ISAs	Access and filter real-time or frequently updating information to generate current responses	Temporality: Dynamic; Volatility: High; Completeness: Evolving; Authority: Mixed	Web-browsing controller; Recency biasing; Temporal retrievers
Persistent ISAs	Support long-term goals by tracking evolving information needs over time	Temporality: Episodic; Completeness: Evolving; Structure: Mixed	Episodic memory; Goal tracking; Task graphs
Social ISAs	Operate in socially constructed knowledge spaces where information is subjective, emergent, or conflicting	Structure: Unstructured; Verifiability: Low; Subjectivity: High; Authority: Contested	Stance detection; Contradiction modeling; Trust estimation
Embodied ISAs	Physically or virtually navigate and perceive environments to collect situated information for task completion	Modality: Multi-sensory; Spatial grounding: Required; Temporality: Real-time; Verifiability: Context-dependent	Active visual SLAM; Scene understanding; Spatial memory; Embodied QA

Dynamic ISAs operate over changing information environments, most commonly the live web. These systems must decide what to search, which sources to trust, how to handle recency, and when the gathered evidence is sufficient. ReAct, WebGPT, WebGLM, GopherCite, WebCPM, WebVoyager, and related systems illustrate this shift from retrieval over fixed corpora to iterative search, browsing, and evidence synthesis [175, 112, 101, 108, 123, 64]. Web interaction benchmarks such as WebShop, WebArena, Mind2Web, and VisualWebArena further extend this setting by requiring agents to click, scroll, fill forms, and reason over interface state [174, 187, 43, 82]. In these settings, the challenge is not only answer generation but also source selection, temporal validity, navigation, and action grounding.

Persistent ISAs add continuity across sessions, tasks, or user goals. Rather than treating

each query as independent, these systems maintain memory, task state, or interaction history that can shape future retrieval and reasoning. Frameworks such as TaskMatrix.AI, AutoGen, Magentic-One, AssistGPT, and CoSearchAgent reflect this direction by combining tool use, task decomposition, memory, and multi-agent coordination [95, 163, 49, 51, 55]. This persistence is useful when information needs unfold over time, but it also raises evaluation questions: a system may appear more capable because it has accumulated useful context, or less reliable if memory introduces outdated assumptions, privacy risks, or spurious personalization.

Social ISAs operate in information environments where evidence is subjective, conflicting, socially produced, or difficult to verify. Examples include social media, online communities, reviews, political claims, and conversational knowledge spaces. In these settings, the agent cannot simply assume that the most relevant or most frequent content is the most reliable. It may need to identify stance, distinguish opinion from evidence, model source authority, surface disagreement, and avoid collapsing contested information into a single overconfident answer. This category is especially important for evaluating ISAs because the answer may depend on how the system represents uncertainty, disagreement, credibility, and user values, not only whether it retrieves a relevant source.

Embodied and multimodal ISAs gather information from situated environments rather than only from text or web documents. They may answer questions about images, videos, screens, physical spaces, or interactive interfaces. Systems such as ViperGPT, ChatVideo, MM-ReAct, Chameleon, DoraemonGPT, WebVoyager, and VisualWebArena show how agents increasingly combine language reasoning with visual perception, tool use, and interface control [144, 157, 169, 103, 173, 64, 82]. These agents make the access path itself part of the task: the system must perceive what is visible, decide what action to take, track the resulting state, and ground the final response in what was observed.

Across these categories, a common pattern emerges. ISA design is shaped by the information environment, and evaluation must reflect that environment. A benchmark for static KGQA should test structured reasoning and answer-path grounding; a benchmark for dynamic web agents should test freshness, source selection, and evidence use; a benchmark

for persistent agents should test memory quality and task continuity; a benchmark for social agents should test uncertainty, disagreement, and credibility; and a benchmark for embodied agents should test perception, navigation, and situated grounding. This motivates the next subsection, which reviews how existing benchmarks and evaluation practices capture, or fail to capture, these requirements.

2.3.2 Benchmarking and Evaluation of Information-Seeking Agents

Benchmarks for ISAs inherit assumptions from several related evaluation traditions: question answering, retrieval-augmented generation, web-agent evaluation, and dynamic benchmarking. Each tradition captures part of the problem, but none fully addresses the evaluation setting developed in this dissertation: agents that must satisfy information needs by accessing, selecting, and synthesizing evidence in dynamic and open environments.

From QA benchmarks to ISA evaluation. Question answering has long served as a convenient proxy for evaluating information access systems. Reading-comprehension datasets such as SQuAD standardized span extraction over provided passages [128], while open-domain QA benchmarks moved closer to search by requiring systems to retrieve evidence before answering. Natural Questions is grounded in real search queries [85], TriviaQA emphasizes compositional questions with substantial lexical and syntactic variation [77], and HotPotQA formalized multi-hop reasoning across multiple documents with supporting facts [170]. These benchmarks were crucial for measuring retrieval and reading capabilities, but they also shaped a relatively narrow view of information seeking: success is usually defined as producing a short answer or matching a reference string.

This assumption is increasingly mismatched to ISAs. A user may need a grounded explanation, a comparison, a current answer, an action over an external interface, or a synthesis of partially conflicting evidence. Even when the task is phrased as a question, the system may need to search, filter, decide which sources are authoritative, reason over temporal constraints, and present uncertainty. Thus, QA benchmarks remain important

foundations, but they typically evaluate a small slice of ISA behavior: whether a system can locate and use a limited amount of answer-bearing evidence.

Retrieval-augmented and agentic QA. Retrieval-augmented generation (RAG) expands this setting by conditioning generation on retrieved evidence. Foundational approaches such as RAG, REALM, RETRO, and ATLAS show that query-time retrieval can improve knowledge-intensive tasks while allowing the external memory to be updated without retraining the parametric model [92, 62, 15, 72]. In this paradigm, the central question is not only whether a model knows the answer, but whether it can retrieve and use relevant evidence.

Agentic QA systems place retrieval inside a broader plan–act–observe loop. WebGPT augments a language model with browsing and requires references during answering [112]; ReAct interleaves reasoning traces with tool calls, enabling iterative search and multi-step evidence gathering [175]; and web-enabled systems such as WebGLM, GopherCite, and WebCPM further explore citation, browsing, and evidence-grounded generation [101, 108, 123]. These systems move closer to the ISA setting because they evaluate evidence access and use as part of the answer-generation process. However, many evaluations still rely on static QA datasets, making it difficult to tell whether success comes from retrieval, memorization, or genuine evidence integration.

Structured reasoning and KGQA. Knowledge-graph QA occupies a complementary part of the design space. Knowledge graphs provide explicit entities and relations, making reasoning paths more inspectable than in raw text. Modern KGQA methods increasingly combine retrieval, graph traversal, semantic parsing, and LLM-based reasoning [142, 166, 74]. This makes KGQA useful for studying controlled reasoning and evidence grounding.

However, KGQA benchmarks also reveal the limitations of static evaluation. Many influential datasets, including WebQuestions, WebQSP, GrailQA, and ComplexWebQuestions, derive from Freebase [10, 177, 58, 145, 12], a knowledge graph discontinued in 2015 [121]. This creates a mismatch between benchmark knowledge and current real-world knowledge.

Newer resources based on DBpedia or Wikidata are more current, but their collaborative construction can introduce noisy schemas, redundant relations, and taxonomic inconsistencies that complicate automated reasoning. These issues motivate the use of cleaner and more controllable structured sources when the goal is to evaluate reasoning rather than artifacts of a particular graph. They also motivate dynamic KGQA generation, where new QA instances can be generated while preserving control over graph structure and answer paths.

Web-agent and interface-based benchmarks. A separate line of work evaluates agents that interact with web pages and digital environments. WebShop evaluates shopping agents that search, compare, and select products [174]. WebArena, Mind2Web, and VisualWebArena extend this idea to broader web tasks that require clicking, scrolling, form filling, visual understanding, and state tracking [187, 43, 82]. These benchmarks are important because they test agentic navigation rather than only answer generation.

At the same time, navigation success is not equivalent to information-need fulfillment. Many web-agent benchmarks evaluate whether an agent completes a specified action in an environment, such as selecting an item or submitting a form. ISA evaluation often requires a different emphasis: whether the agent gathered appropriate evidence, interpreted it correctly, and produced a response or action that is justified by that evidence. This distinction matters for tasks where the environment is only a means of accessing information, such as checking appointment availability, reading dynamic prices, or locating a policy detail behind an interactive interface.

Contamination and retrieval-time leakage. The shift toward LLM-based and web-enabled systems introduces a central validity threat: benchmark items or answer-bearing evidence may already be available to the system. Pretraining contamination occurs when evaluation items, or close paraphrases, appear in the corpora used to train the model [130, 94]. In such cases, measured performance may reflect memorization rather than task-solving. This risk is amplified by public static benchmarks, which can be scraped, mirrored, quoted, or

reused in training data.

RAG and agentic systems introduce a second channel: retrieval-time contamination. Even if the model did not memorize the item, a web-enabled agent may retrieve a page that directly contains the benchmark question, answer, or annotation artifact, allowing the system to copy the target rather than reason from evidence. Recent work shows that search-time leakage can alter leaderboard conclusions once systems are allowed online access [63]. For ISAs, this is especially problematic because retrieval is part of the intended capability. Evaluation must therefore distinguish legitimate evidence gathering from unintended access to benchmark artifacts.

Dynamic and time-sensitive benchmarking. Time-sensitive benchmarks partially address this issue by evaluating whether systems can answer what is true at a particular time. RealTime QA, FreshQA, and StreamingQA test recency, changing answers, and failures caused by stale evidence [79, 155, 98]. These benchmarks are valuable because they move beyond static world knowledge and require systems to account for temporal validity.

Dynamic benchmarking goes further by changing or generating evaluation instances over time. Dynabench and Dynaboard introduced dynamic evaluation as a platform idea with human- and model-in-the-loop data collection [81, 106]. More recent approaches use environment-based generation for interactive agents or graph-based generation for controllable reasoning tasks [132, 191, 181, 39]. Dynamic generation is particularly attractive for ISA evaluation because it can reduce memorization risk, support refreshed evidence, and enable controlled variation in task complexity.

However, dynamic evaluation also introduces new requirements. A benchmark must remain comparable across rounds, otherwise performance changes may reflect arbitrary changes in the generated instances rather than differences in system capability. It must preserve evidence traceability, because answers over live or changing information are only meaningful relative to what was available at evaluation time. It must also support failure decomposition, since an agent may fail because it searched poorly, selected weak sources, misunderstood evidence,

or synthesized evidence incorrectly. These requirements connect directly to this dissertation's progression: identifying contamination as a validity threat, developing controlled dynamic generation over structured knowledge, and then evaluating open-web sensemaking where evidence is dynamic, dispersed, and query-conditioned.

Toward evaluation of evidence use and sensemaking. The common limitation across much of the literature is that evaluation often measures either final-answer correctness, environment-task completion, or retrieval success in isolation. ISA evaluation requires a more integrated view. A system may retrieve relevant evidence but fail to synthesize it; it may produce a plausible answer without adequate support; or it may complete a web task while leaving the user unable to verify the result. The central evaluation object is therefore not only the final answer, but the relationship between the user's need, the evidence accessed, the reasoning or synthesis performed, and the response delivered.

This dissertation builds on the above literature while addressing three gaps. First, it studies contamination in both pretraining and retrieval-time settings, showing why static evaluation becomes unreliable for web-enabled QA. Second, it develops dynamic benchmark generation methods that preserve control and auditability while reducing memorization risk. Third, it moves from structured reasoning to open-web sensemaking, where questions require integrating evidence across multiple sources rather than extracting a single answer-bearing passage. Together, these steps frame ISA evaluation as the measurement of evidence-grounded information-need fulfillment in dynamic and open environments.

Chapter 3

CONTAMINATION AND THE LIMITS OF STATIC EVALUATION

Static benchmarks have been central to measuring progress in question answering and information retrieval. They provide fixed test items, shared scoring rules, and a common basis for comparing systems. This chapter argues that this evaluation setup becomes fragile when applied to modern information-seeking agents (ISAs), because these systems are built from models trained on large public corpora and often use web retrieval at inference time. In such settings, benchmark content can enter the evaluation pipeline before or during the test, making it difficult to determine whether a score reflects information-seeking capability or prior exposure to the benchmark itself.

The chapter first situates this problem in the longer history of machine-learning evaluation practice. It then formalizes two contamination channels, audits both channels across widely used QA datasets, and discusses the implications for benchmark design.

3.1 Shifts in Machine-Learning Evaluation Practice

Modern contamination problems are easier to understand against the longer history of machine-learning evaluation. For much of this history, the basic evaluation practice was simple: train on one split, tune on another, and report performance on a held-out test set. This practice supported a wide range of traditional machine-learning methods, including decision trees, support vector machines, random forests, logistic regression, and other models trained from task-specific data. The crucial point is not that these systems were immune to poor evaluation practice. Train–test leakage, duplicated records, label artifacts, and distribution shift have always been possible. Rather, the boundary between training and evaluation was

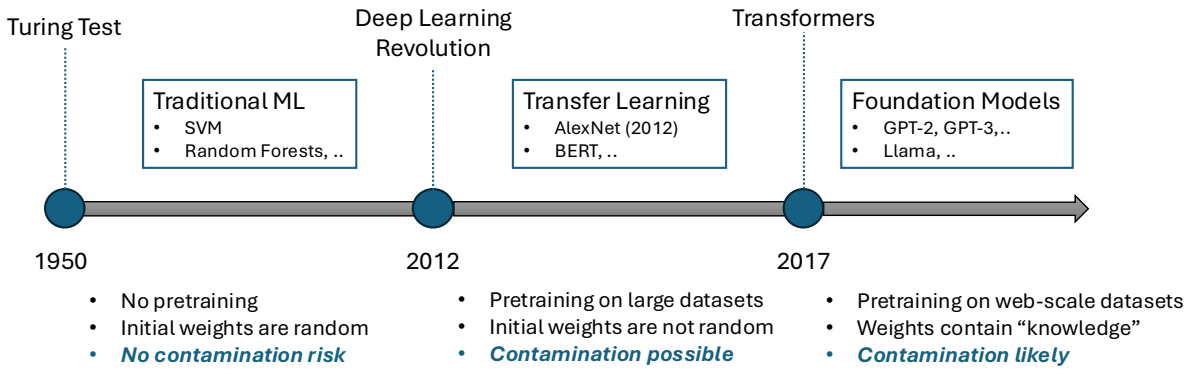


Figure 3.1: Evolution of machine-learning evaluation practice. Traditional machine-learning evaluation relied on task-specific training and held-out test splits, making contamination primarily a split-construction problem. Transfer learning introduced pretrained representations whose source data were harder to audit. Foundation models trained on web-scale corpora made benchmark exposure more likely, and web-enabled retrieval-augmented systems add a separate inference-time channel through which benchmark content can leak.

usually created by the benchmark designer and was often inspectable. The model typically starts with random weights initialization, and its task knowledge came primarily from the training split supplied for that task.

The deep learning period retained this split-based evaluation logic while scaling models, datasets, and benchmark culture. ImageNet and the ILSVRC challenge are emblematic of this period because they provided a fixed label space, carefully managed data splits, and a common leaderboard that allowed methods such as AlexNet and ResNet to be compared under a shared protocol [42, 84, 65]. Under this regime, contamination was still possible, but it was primarily treated as a dataset construction or experimental hygiene problem: if the splits were clean, the reported test score could be interpreted as performance on unseen examples from the benchmark distribution.

Transfer learning weakened this evaluation setup. Once pretrained encoders such as ImageNet backbones, BERT, and T5 became standard, downstream models no longer began

from random initialization [44, 127]. They inherited representations learned from larger and less task-specific corpora. The downstream train, validation, and test splits could still be clean, but the pretraining data were often external, much larger, and harder to audit against the downstream test set. This made contamination possible at a new layer: a test item could be unseen in the downstream training split while still being represented in the model through pretraining.

Foundation models trained on web-scale corpora changed the evaluation setting more substantially [13]. These models are not only initialized with general representations; they encode large amounts of world knowledge from broad, weakly documented data sources. As benchmarks, tutorials, leaderboards, GitHub repositories, documentation pages, and discussion threads become part of the public web, the distinction between the evaluation set and the training environment becomes porous. Memorization studies show that large models can reproduce training examples and store datapoint-specific information, especially when content is duplicated or highly repeated [25, 149, 111]. At this point, contamination is no longer only a mistake in splitting a dataset. It becomes a systemic risk of evaluating models trained on the same public information ecosystem that hosts the benchmarks.

Web-enabled retrieval introduces an additional layer of contamination risk. Retrieval-augmented systems and information-seeking agents use external search at inference time, so the model is not the only place where benchmark content can leak. A benchmark item may be absent from the model's pretraining data but still reachable through a search engine during evaluation. In that case, a system can appear to answer by reasoning over evidence while actually copying or lightly rephrasing an answer-bearing benchmark page. Figure 3.1 summarizes this progression: contamination moves from an avoidable split-level error, to possible pretraining-level exposure, to a likely risk in foundation models, and finally to a dual-channel problem for web-enabled systems.

The last two stages are the focus of this chapter. For ISAs, contamination can enter through the parametric model before evaluation or through retrieval during evaluation. These channels have different mechanisms and require separate audits, but both undermine the

same assumption: that a score on a static public benchmark measures the agent's ability to seek, select, and use evidence rather than its exposure to the benchmark itself.

3.2 Problem Setting

Information-seeking agents are typically built from two coupled components: a parametric language model trained on large web corpora, and a retriever that searches an external index at inference time. Both components draw from the public web. The same web also hosts the questions, answers, explanations, and dataset pages that compose widely used QA benchmarks. When a benchmark item, its answer, or an answer-bearing paraphrase is reachable through either component, a measured score no longer cleanly reflects what an agent has learned to do. It can instead reflect what an agent has previously seen, or what a search engine happens to surface during evaluation. This is the contamination problem, and it conditions every empirical claim that follows in this dissertation.

Two channels are at issue, and they should be tracked separately. *Pretraining contamination* arises when an evaluation item, or content that closely resembles it, appears in the corpora used to train the underlying model. *Retrieval-time contamination* arises when the retriever surfaces benchmark content during evaluation, even if the model itself never memorized the item. Both channels can inflate reported accuracy, and a benchmark that ignores either is liable to overstate capability for web-enabled question answering and retrieval-augmented generation [130, 25, 94, 63].

This chapter analyzes both channels using span-based, auditable detectors against two operationalized source sets: a large indexed pretraining proxy and a live retrieval pool obtained from a metasearch engine. The analysis covers five widely used QA datasets and reports dataset-level rates and rank-of-evidence distributions. The findings show measurable leakage through both channels, with answer-bearing matches frequently surfacing on the first page of results. The reported numbers are conservative lower bounds. Together, these results establish that static public QA evaluation is increasingly unreliable for information-seeking systems, and they motivate the controlled, dynamic structured setting developed in Chapter 4.

3.3 Limits of Static QA Evaluation in Web-Enabled Settings

Static QA benchmarks have anchored progress in open-domain question answering. SQuAD, Natural Questions, TriviaQA, HotpotQA, WebQuestions, SimpleQA, and FreshQA each fix a question pool, reference answers, and scoring rules so that systems can be compared on a common axis [128, 85, 77, 172, 10, 161, 155]. The convenience of a fixed test set rests on an assumption that has weakened over time: that the items used to evaluate a system are not also part of the environment from which the system was built or in which it now operates.

Retrieval-augmented generation breaks that assumption explicitly. A modern web-RAG system pairs a language model with a retriever that searches large external corpora at inference time [90, 71, 52]. Evaluation under this architecture is no longer a closed-book test against a static set of weights. It is a test against a model whose pretraining corpus may have included benchmark items, run on top of a retriever whose index can return pages that mirror or discuss those same items. Either path is sufficient to influence the score.

The consequence is a validity problem, not a small experimental artifact. A static accuracy number is normally interpreted as a proxy for reasoning over evidence. If the model has memorized a question-answer pair, the proxy collapses to recall. If the retriever surfaces an answer-bearing benchmark page, the proxy collapses to lookup. In either case, the system can score well without exhibiting the information-seeking behavior the benchmark was meant to measure. For agents that are explicitly designed to use search as a tool, this is a structural concern: tool access can simultaneously model the deployment setting and short-circuit the test. Contamination must therefore be modeled as both a training-time and an inference-time phenomenon, and any claim about web-enabled QA capability needs to be qualified by an audit of both.

3.4 Measuring Pretraining and Retrieval-Time Contamination

The remainder of this chapter presents an empirical audit of contamination in popular static QA benchmarks. The goal is to measure two concrete forms of exposure: whether benchmark

content appears in a large pretraining proxy, and whether the same content is surfaced by web retrieval during evaluation. This section defines the source sets, matching criteria, and detector choices used for both measurements.

Let an evaluation item be $x = (q, a)$, with question q and reference answer a. Let M be a language model pretrained on corpus D_M, and let $\mathcal{R}(q)$ denote the set of documents returned by a retriever when queried with q. The two contamination channels correspond to two source sets:

$$\mathcal{S}_{\text{pre}}(x) = D_M, \qquad \mathcal{S}_{\text{ret}}(x) = \mathcal{R}(q).$$

$\mathcal{S}_{\text{pre}}(x)$ captures what the model could have seen before deployment. $\mathcal{S}_{\text{ret}}(x)$ captures what the agent can fetch during deployment.

A consistent query policy is required to keep the audit transparent and comparable. Retrieval is performed using the question string only; the answer a is never used to construct $\mathcal{R}(q)$. This mirrors deployment, where a system is given q and must locate or generate a. Concatenating a into the search query would inflate measured exposure with information unavailable at test time and conflate audit with oracle access.

Two views of an item are then audited. The *question-only* view, $\pi_{\text{in}}(x) = q$, treats the input as the unit of leakage. The *question+answer* view, $\pi_{\text{in+lab}}(x) = q \oplus a$, treats the input together with its label as the unit of leakage. Both views matter for different reasons. Question-only matches indicate that the input itself, or a near-duplicate, is present in the source set; even input-only leakage can shift measured behavior, because models exposed to repeated or paraphrased prompts can produce different completions than they would for unseen prompts [94]. Question+answer matches are stricter and more directly diagnostic of an answer-lookup risk.

Following the standard contamination criterion of Ravaut et al. [130], let f be a similarity function, τ a threshold (with τ=1 corresponding to verbatim overlap), and $g(a, d) \in \{0, 1\}$ a predicate that holds when document d contains a. For channel $C \in \{\text{pre}, \text{ret}\}$ and view

$v \in \{\text{in}, \text{in+lab}\}$, an evaluation item is flagged when

$$\exists\, d \in \mathcal{S}_C(x) \;\text{ s.t. }\; f(\pi_v(x), d) \geq \tau, \tag{3.1}$$

with the additional answer-bearing variant

$$\exists\, d \in \mathcal{S}_C(x) \;\text{ s.t. }\; f(q, d) \geq \tau_q \;\wedge\; g(a, d) = 1. \tag{3.2}$$

With a common τ_q, positives under Equation (3.2) are a subset of positives under Equation (3.1).

Span-based detectors. The similarity function f is instantiated using span-based detectors that have become the practical standard in pretraining data audits. The detectors used here include exact match [46], 8-gram word overlap from the GPT-2 audit [125], 13-gram word overlap from the GPT-3 audit [19], a 50-character substring match from the GPT-4 technical report [114], and 8-gram overlap with a 70% threshold from the PaLM audit [33]. An item is flagged when any detector fires.

Span-based detectors are used for two practical reasons. First, they are auditable: a flagged item carries a concrete evidence span that a human can verify against the source document. This is essential when contamination claims must be defended against alternative explanations such as topical similarity. Second, they are tractable at web scale. Auditing a single question against a billion-document index is already a needle-in-a-haystack problem; sweeping entire benchmarks against such indices is feasible only with detectors that admit indexed lookup, and span methods do. More expressive paraphrase or embedding detectors can raise recall, but they are harder to audit, more sensitive to threshold and model choices, and less stable across the length mismatch between short questions and long web pages [99, 86]. For these reasons, span-based detection is consistent with prior practice in pretraining data audits and is adopted here for both channels [46, 125, 19, 114, 33, 86, 130].

Pretraining channel. The complete pretraining corpus D_M is generally unavailable for deployed models. Following candidate-set approximations used in open-data audits [94], $\mathcal{S}_{\text{pre}}(x)$

is approximated by a candidate pool drawn from a large indexed web corpus. Specifically, the Common Crawl C4 corpus [127] is indexed in an Elasticsearch-compatible service hosted on AWS OpenSearch. For each item $x = (q, a)$, the top-k candidates are retrieved by BM25 with a default $k = 1000$, and the contamination criterion is evaluated over this pool under both views and the answer-bearing variant. C4 acts as an auditable proxy for the type of large-scale web text that dominates modern pretraining mixtures; it does not reproduce D_M, but it does make the audit reproducible and inspectable. Any positive flag against the C4 candidate pool is therefore evidence of plausible pretraining exposure rather than a guarantee of memorization in M.

Retrieval-time channel. The retrieval-time source set $\mathcal{S}_{\text{ret}}(x) = \mathcal{R}(q)$ is constructed by issuing the raw question string to SearxNG, an open metasearch engine that aggregates results from multiple underlying engines [1, 134]. For each q, results from the first three result pages are pooled across underlying engines to form $\mathcal{R}(q)$. The same pool is then used both as the audit target and, where relevant, as the retrieval context for downstream answer generation. This design isolates retrieval-time leakage from memorization in M: any positive against $\mathcal{R}(q)$ reflects content that is reachable by a routine web query at evaluation time, regardless of what the model has internalized.

Conservative reading. Both channels use finite candidate pools and span-based detectors and inherit the limits of those choices. The C4 index does not include all pretraining data used by closed-source models. The first three pages of metasearch results do not include every page reachable through alternative queries, paid retrieval APIs, or more aggressive crawl strategies. Span detectors miss paraphrases that fall below their thresholds. The reported rates are therefore lower bounds. They likely under-count rather than over-count contamination, so the observed rates should be read as a conservative signal of a broader problem rather than as its full extent.

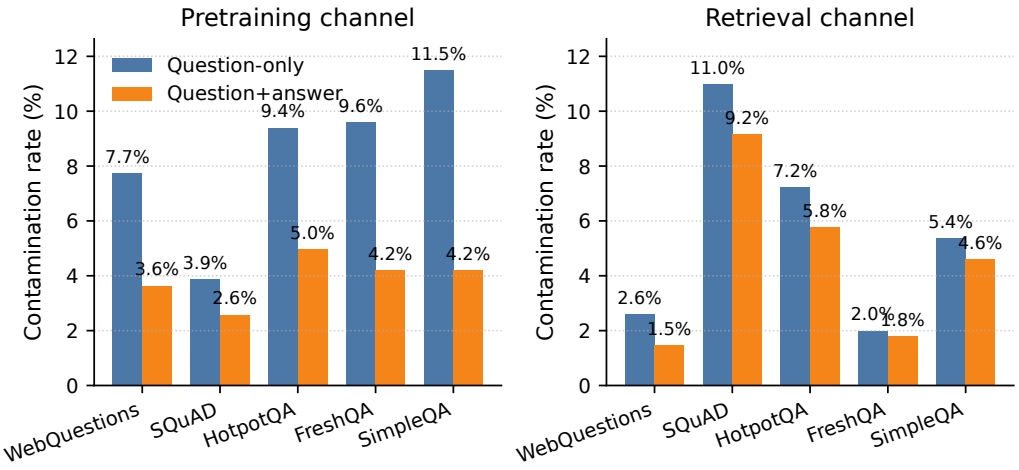


Figure 3.2: Contamination rates by dataset across the two channels: pretraining (against the C4 BM25 candidate pool) and retrieval-time (against the SearxNG pool). Rates are reported separately for the question-only and question+answer views.

3.5 Audit Results Across QA Benchmark Datasets

The audit covers five QA datasets spanning reading comprehension, multi-hop reasoning, concise factoid retrieval, and freshness-sensitive questions: SQuAD, HotpotQA, WebQuestions, SimpleQA, and FreshQA [128, 172, 10, 161, 155]. Each item is evaluated under both contamination channels using the question-only and question+answer views described in Section 3.4.

Dataset-level rates. Figure 3.2 reports contamination rates for both channels and both views. A consistent pattern holds across datasets: question-only rates exceed question+answer rates, reflecting that near-duplicate or paraphrased questions are more common on the web than full question-answer co-occurrences.

In the pretraining channel, question-only rates are highest for SimpleQA at 11.5% and FreshQA at 9.6%, followed by HotpotQA at 9.4%, WebQuestions at 7.7%, and SQuAD at 3.9%. The corresponding question+answer rates are 4.2%, 4.2%, 5.0%, 3.6%, and 2.6%. The

dataset ordering reflects how strongly each set's surface form overlaps with public web text: short, factoid-style items share more n-grams with encyclopedic and trivia-style content than do longer span-based items.

The retrieval-time channel produces a different ordering. SQuAD shows the largest exposure, with 11.0% question-only and 9.2% question+answer matches. HotpotQA follows at 7.2% and 5.8%, SimpleQA at 5.4% and 4.6%, WebQuestions at 2.6% and 1.5%, and FreshQA at 2.0% and 1.8%. The contrast between SQuAD's low pretraining rate and high retrieval-time rate is informative: SQuAD questions inherit a strong textual link to specific Wikipedia passages, and a question-only metasearch query frequently returns the original passage near the top of the results. Retrieval-time leakage is therefore not a function of how memorable a benchmark is in the abstract, but of how cleanly a question routes to its source page on a live web index.

These rates should be read as conservative lower bounds, in line with the limitations described in Section 3.4. They use finite candidate pools and span-based detectors, and they will under-count contamination that takes paraphrastic, multilingual, or syndicated form. Even under this conservative setup, both channels exhibit non-trivial leakage on every dataset audited. The observed rates are large enough to matter for comparisons among systems that report differences of only a few accuracy points.

Where contaminated evidence appears in the result list. Retrieval-time contamination is not only a question of presence; it is also a question of position. Figure 3.3 shows the rank at which contaminated documents appear in the pooled SearxNG list, separately for the two views. Question+answer matches concentrate at low ranks, frequently within the first page of results, while question-only matches are more spread out. Across datasets, the question+answer mass is consistently shifted toward the top of the ranking compared to the question-only mass.

This rank pattern is what makes retrieval-time exposure operationally consequential. A web-RAG system typically passes the top retrieved documents to its generator. If answer-

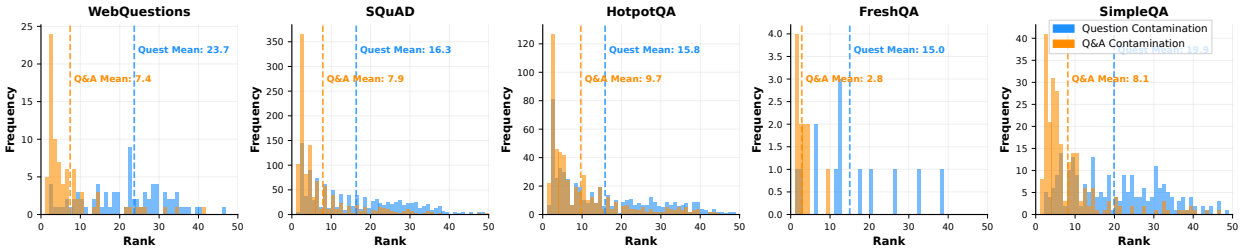


Figure 3.3: Rank distribution of contaminated evidence surfaced by the retriever, shown per dataset. Histograms separately show question-only and question+answer matches, where the rank refers to the position of the contaminated document within the pooled SearxNG result list.

bearing benchmark pages appear near the top of the result list, those pages are exactly the evidence that gets read. The system can then produce the correct answer by quoting or lightly rephrasing a single retrieved page rather than by aggregating across multiple pieces of evidence. From the outside, the system looks like it is reasoning about the question. Internally, it is performing a lookup. Static accuracy on a contaminated benchmark cannot distinguish these cases.

Illustrative contamination examples. The aggregate rates above are easier to interpret through concrete cases. In both examples below, the contaminated content comes from sources that are not obviously low quality. This is part of the difficulty: leakage can arise from authoritative reference pages, academic papers, dataset mirrors, and community repositories, not only from spam or low-quality pages.

Pretraining Contamination Example

Dataset: FreshQA

Question: "What was the shortest war in history?"

Answer: Anglo–Zanzibar War

Contamination source: Webpage indexed in the Common Crawl C4 corpus; Encyclopaedia Britannica page: `https://www.britannica.com/event/Anglo-Zanzibar-War`

The first example illustrates pretraining exposure. The answer is not obscure in the abstract, but its presence in an indexed reference page means that a model trained on web-scale data may answer from prior exposure rather than from evidence gathered during evaluation. The audit does not claim that a particular closed model memorized this item. It shows that the item is present in a plausible pretraining source, which weakens the interpretation of a static benchmark score as evidence of fresh reasoning.

Retrieval-time Contamination Example

Dataset: HotPotQA

Question: "Which Hells Angel member stabbed and beat an attendant of the 1969 counterculture-era rock concert in the United States?"

Answer: Alan Passaro

Contamination source: Search-indexed mirrored content; arXiv paper containing the question and answer verbatim: `https://arxiv.org/pdf/2505.15117`

The second example illustrates retrieval-time exposure. Here, the concern is not what the model saw during pretraining, but what a web-enabled system can retrieve at test time. If a search query returns an indexed page that reproduces the benchmark item and answer, the agent can obtain the target directly. This short-circuits the intended multi-hop reasoning behavior: the system appears to solve the HotPotQA instance, but the retrieval channel has made the benchmark answer available as a lookup target.

Source diversity and the limits of domain filtering. The domain-level view in Figure 3.4 explains why blocklists are not a sufficient remedy. The retrieved pages span encyclopedic sources such as Wikipedia, dataset-hosting and replication pages such as the Hugging Face hub and GitHub, academic aggregators such as arXiv and the ACL Anthology, and a long tail of mirrors, blog posts, and discussion threads. Many of the implicated domains

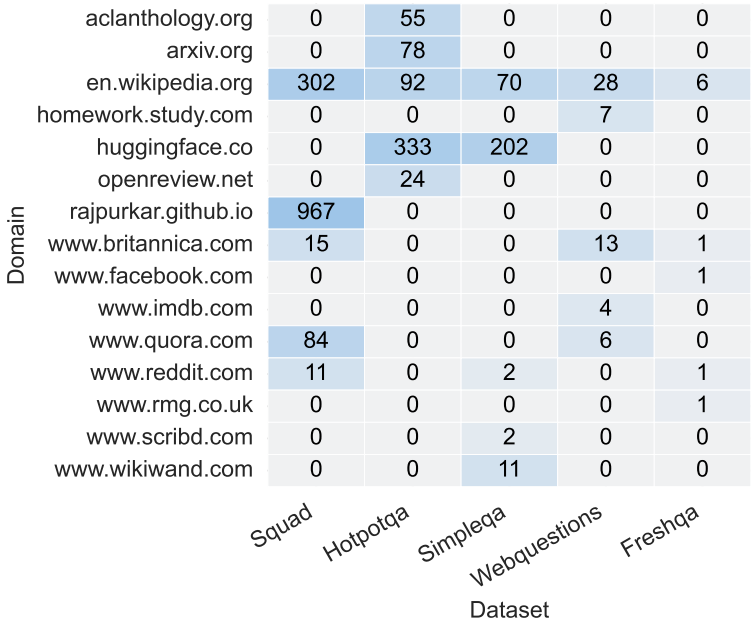


Figure 3.4: Dominant retrieval-time contamination sources across datasets. Counts indicate contaminated documents surfaced from each domain under the retrieval-time audit.

are precisely the high-quality, authoritative sources that retrieval-augmented systems are designed to use. Removing them from a retrieval index protects the audit at the cost of removing useful evidence, and it does not stop benchmark material from re-entering through community sites, mirrors, or paraphrased discussion. Domain filtering therefore offers a partial, fragile defense rather than a structural fix.

3.6 Implications for Benchmark Design

The audit yields three implications that condition the rest of this dissertation.

First, contamination is a property of the evaluation system as a whole, not only of a model. Pretraining contamination concerns what the model could have seen before inference. Retrieval-time contamination concerns what the agent can reach during inference. A benchmark that controls only one channel can still inflate scores through the other. For information-seeking agents, both channels must be tracked, and both must be reported with

explicit query and detector choices so that audits remain reproducible.

Second, static public benchmarks are increasingly fragile as proxies for realistic information-seeking behavior. Once a question, a reference answer, an explanation, or a leaderboard page is published on the web, it becomes part of the same information environment that web-RAG systems search. The conservative lower bounds reported above show that this is not a hypothetical risk for the most widely cited QA datasets. A score on these benchmarks no longer cleanly separates capability from exposure, and the size of the gap depends on the specific channel that dominates each dataset.

Third, the obvious mitigations are insufficient. Removing known benchmark domains from a retrieval index helps only narrowly: mirrors, paraphrased discussions, and answer-bearing pages on otherwise authoritative sites continue to surface during evaluation. Paraphrasing benchmark items reduces verbatim overlap but leaves topic, entity, and answer pathways largely intact, so retrieval can still route queries to the original or near-original sources. A response that addresses the underlying cause must change the structure of the evaluation, not only its surface. Specifically, evaluation items need to be generated or refreshed under controlled rules, with recorded evidence and auditable correctness criteria, so that the audit conditions are part of the protocol rather than an afterthought applied to a fixed test set.

These implications do not abandon static benchmarks. Static datasets remain useful for controlled comparisons, regression testing, and the study of particular sub-skills. The point is narrower: a static accuracy number, on its own, is no longer adequate evidence that a web-enabled information-seeking agent can retrieve, reason over, and synthesize information. Reports of such agents need to include contamination audits along the lines of those above, and where the goal is to measure reasoning over current information, evaluation should prefer protocols that can produce fresh, evidence-grounded items.

3.7 Conclusion

Contamination enters QA evaluation through two channels. Pretraining contamination introduces benchmark material into the parametric component before deployment. Retrieval-

time contamination introduces benchmark material through the retrieval component during deployment. The empirical audit in this chapter, conducted with span-based detectors against a C4 BM25 candidate pool and a pooled SearxNG result set, reports measurable leakage in every dataset audited. Question-only matches exceed question+answer matches, but answer-bearing matches concentrate at the top of the retrieval ranking, which is exactly where their effect on a web-RAG generator is largest. The reported rates are conservative lower bounds, and the source distribution of retrieval-time contamination shows that simple domain filtering cannot resolve the problem.

The methodological consequence for the rest of the dissertation is direct. If a static public test set can be exposed through both pretraining and retrieval, then a credible measure of information-seeking capability requires an evaluation regime in which item generation, evidence selection, and correctness checking are themselves auditable and controllable. Chapter 4 takes the first step in that direction, in the structured setting of knowledge graph question answering, where the underlying graph constrains what an item can ask and what counts as a correct answer.

Chapter 4

CONTROLLED DYNAMIC EVALUATION OVER STRUCTURED KNOWLEDGE

Static benchmark contamination motivates a different evaluation strategy: instead of treating a benchmark as a fixed public artifact, the benchmark can be defined as a controlled generation process. This chapter develops that idea in a structured knowledge setting, where evaluation items can be generated dynamically while retaining explicit grounding, verification, and distributional consistency.

4.1 Introduction

Chapter 3 established that static public QA evaluation no longer cleanly separates capability from exposure. Widely used QA benchmarks show evidence of both pretraining contamination and retrieval-time contamination, and surface-level defenses such as paraphrasing or domain blocking do not address the deeper evaluation problem. This chapter develops the first constructive response in the dissertation: a dynamic evaluation framework over structured knowledge, named Dynamic-KGQA [39]. The framework generates evaluation items on demand from a curated knowledge graph, pairs each item with an explicit grounding subgraph, and verifies generated questions against the same structured representation used to create them.

The structured setting is important because it makes dynamic evaluation auditable. Generation, grounding, and verification can be inspected directly; new items can be produced across runs without reusing the same surface forms; and distributional consistency can be tested rather than assumed. This chapter therefore establishes a controlled setting for studying contamination-resistant benchmark generation before moving to the open-web

setting in Chapter 5. The main claim is not that KGQA captures the full complexity of information-seeking agents in deployment. Rather, structured knowledge provides a tractable and technically precise environment for developing the core ideas that later chapters extend to noisier, more dynamic, and less controllable information environments.

4.2 Structured Knowledge as an Intermediate Evaluation Setting

A knowledge graph offers a representation that is rare in evaluation environments: every fact has a provenance, every entity is named, every relation is typed, and the boundary of what is and is not in the graph is enumerable. For an evaluation framework that needs to generate fresh items on demand, this matters in three ways.

First, items can be tied to explicit subgraphs. Question generation can be conditioned on a small, connected portion of the graph, and the answer can be required to lie within that portion. The evaluation item is therefore not only a (q, a) pair but a triple $(q, a, \mathcal{G}')$ in which $\mathcal{G}'$ is the structural grounding. Auditing whether an answer is supported reduces to membership and connectivity tests on $\mathcal{G}'$, not to free-text inference about a long retrieved passage.

Second, controllable generation becomes more feasible. The graph fixes what entities and relations exist, and the generator can be parameterized to produce questions of varying hop count or topical focus by adjusting how the seed subgraph is selected. The same parameters can be exercised across runs to produce variants whose difficulty distribution is comparable, supporting longitudinal reporting without reusing the exact items the model may already have seen.

Third, the contamination surface shrinks substantially. Items are not drawn from a fixed public test set. Their wording is generated rather than archived, and the grounding evidence is the local subgraph, not the public web. The pretraining and retrieval channels analyzed in Chapter 3 therefore have far fewer routes through which they can affect a measurement: there is no static benchmark page to memorize, and there is no live search step whose top results can carry the answer.

The structured setting also clarifies the boundary of what this chapter evaluates. A knowledge graph is not the open web. Its triples are curated, its coverage may lag real-world events, and its representation is cleaner than the heterogeneous documents that an ISA must read in deployment. Live retrieval, query reformulation, evidence aggregation across noisy sources, and sensemaking under uncertainty are not fully exercised when the evidence is already organized as a local subgraph. A KGQA benchmark therefore targets a specific part of ISA evaluation: structured retrieval and reasoning over named entities with inspectable evidence. This scope makes it possible to study dynamic generation and evidence grounding under controlled conditions and provides a baseline for the open-web evaluation regime developed in Chapter 5.

4.3 Framework Design

Dynamic-KGQA is designed around two requirements. First, evaluation items should be generated at evaluation time so that no fixed public test set becomes the object of memorization. Second, the generated items should remain comparable across runs so that scores reported at different times measure the same underlying capability. The framework satisfies these requirements through structural anchors rather than surface-level paraphrasing: items are generated from seed subgraphs of a curated knowledge graph, and the generation policy is held fixed while controlled randomness changes the resulting surface forms and answer paths.

Figure 4.1 sketches the end-to-end pipeline. The remainder of this section describes each stage: subgraph construction (Section 4.3.1), QA generation (Section 4.3.2), randomized variation (Section 4.3.3), structural verification (Section 4.3.4), and quality assessment by a panel of LLM judges (Section 4.3.5). The underlying knowledge graph and seed corpus are described in Section 4.3.6.

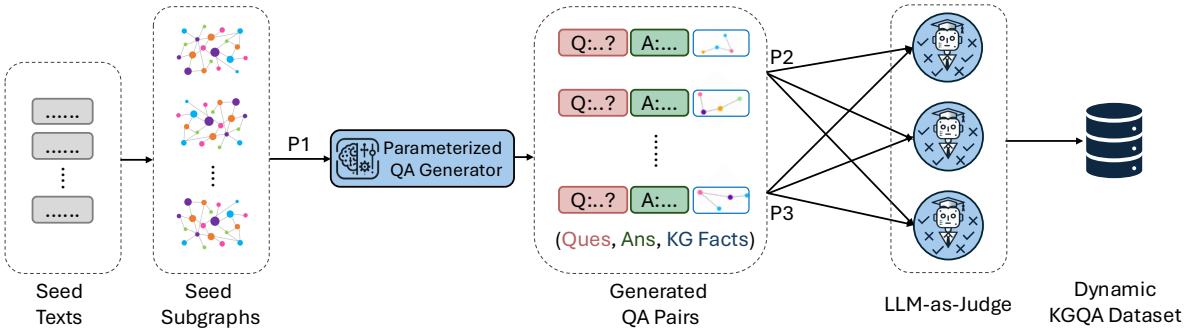


Figure 4.1: Overview of DYNAMIC-KGQA. Seed texts identify entities in a curated knowledge graph; a subgraph is extracted for each seed; an LLM is prompted to generate question-answer pairs grounded in that subgraph; and a panel of independent LLM-as-a-Judge models verifies grammaticality, non-redundancy, and answer support before the item is admitted to the dataset.

4.3.1 Constructing Seed Subgraphs

Let $\mathcal{G} = (\mathcal{V}, \mathcal{P}, \mathcal{T})$ be a knowledge graph with entity set $\mathcal{V}$, predicate set $\mathcal{P}$, and triple set $\mathcal{T} \subseteq \mathcal{V} \times \mathcal{P} \times \mathcal{V}$. Each triple has the form (h, p, t), where h and t are entities and p is a predicate. The objective is to extract, for each evaluation item, a subgraph $\mathcal{G}' = (\mathcal{V}', \mathcal{P}', \mathcal{T}')$ that is thematically coherent, retains only entities and relations relevant to a chosen topic, and is compact enough to ground a small number of high-quality QA pairs without dragging in unrelated structure.

Seed texts. Topic selection is delegated to a corpus of seed texts. A seed text is a short passage about a single concept, person, place, or event, and its role is to specify what the resulting subgraph should be about. Selecting seed texts from a particular domain biases the dataset toward that domain. A diverse seed corpus produces a general-purpose dataset; a specialized corpus produces a focused one. Subgraph size is also controlled here, because longer seed texts mention more entities and therefore yield larger subgraphs.

Seed entities. Given a seed text, the framework extracts the set of entities mentioned in it and retains those that exist in $\mathcal{V}$. Formally, with $\mathcal{E}$ the set of entities detected in the seed text,

$$\mathcal{S} = \{v \in \mathcal{E} \cap \mathcal{V}\}. \tag{4.1}$$

$\mathcal{S}$ is the set of seed entities and serves as the structural anchor for everything downstream. A seed text about *Harry Potter*, for example, may produce $\mathcal{S} = \{\text{Harry Potter}, \text{Hogwarts}, \text{Albus Dumbledore}\}$, with each entity bound to a node in the underlying graph rather than to a free-text mention.

Neighborhood expansion. Seed entities alone are typically too sparsely connected to support multi-hop questions, so the neighborhood around each seed entity is expanded by one hop. For each $v \in \mathcal{S}$, all triples in which v appears as head or tail are collected:

$$\mathcal{T}_{\text{one-hop}}(v) = \{(h, p, t) \in \mathcal{T} \mid h = v \text{ or } t = v\}. \tag{4.2}$$

Taking the union across seed entities yields an expanded subgraph

$$\mathcal{G}_{\text{expanded}} = (\mathcal{V}_{\text{expanded}}, \mathcal{P}_{\text{expanded}}, \mathcal{T}_{\text{expanded}}),$$

where $\mathcal{V}_{\text{expanded}}$, $\mathcal{P}_{\text{expanded}}$, and $\mathcal{T}_{\text{expanded}}$ are obtained by aggregating endpoints, predicates, and triples of $\mathcal{T}_{\text{one-hop}}(v)$ over all $v \in \mathcal{S}$. The expanded subgraph is dense but noisy: it includes off-topic neighbors that share a single predicate with a seed entity but contribute little to the topic that $\mathcal{S}$ is meant to express.

Steiner tree extraction. The next step trims $\mathcal{G}_{\text{expanded}}$ to a minimal connected substructure that still spans $\mathcal{S}$. This is the standard formulation of a Steiner tree. Concretely, the goal is to find the smallest set of triples that connects all seed entities,

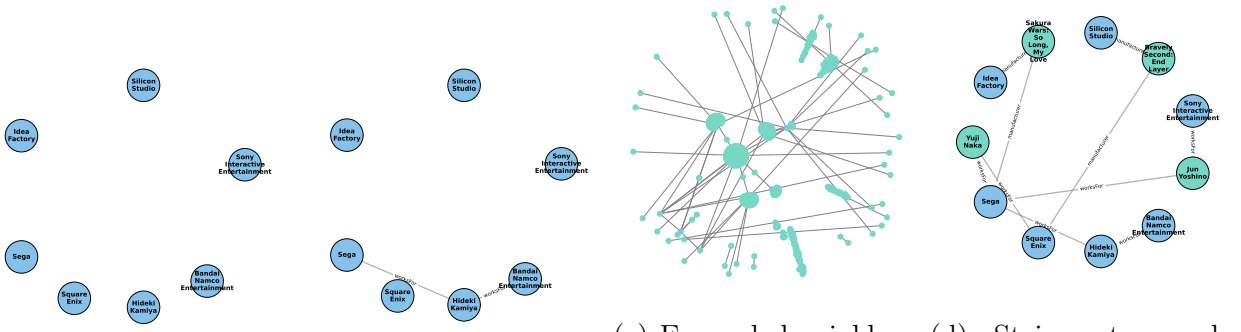
$$\mathcal{T}_{\mathcal{S}} = \arg\min_{\mathcal{T} \subseteq \mathcal{T}_{\text{expanded}}} |\mathcal{T}|, \qquad \text{subject to } \mathcal{S} \subseteq \mathcal{V}(\mathcal{T}), \tag{4.3}$$

where $\mathcal{V}(\mathcal{T})$ denotes the set of entities appearing in the selected triples. Solving the Steiner tree problem exactly is NP-hard, so an approximation algorithm is used in practice [107]. The result is a compact subgraph that retains structural coherence among the seed entities while removing isolated or weakly connected neighbors that the one-hop expansion introduced.

Final subgraph selection. Even after Steiner tree approximation, the resulting structure can split into multiple connected components when $\mathcal{G}_{\text{expanded}}$ itself is too sparse to admit a single spanning tree. Disconnected components do not support coherent multi-hop reasoning, so a final filtering step retains only the largest connected component:

$$\mathcal{G}' = \arg\max_{\mathcal{C}_i \subseteq \mathcal{T}_\mathcal{S}} |\mathcal{V}(\mathcal{C}_i)|. \tag{4.4}$$

The output is the seed subgraph $\mathcal{G}' = (\mathcal{V}', \mathcal{P}', \mathcal{T}')$ that grounds the QA generation step.



(a) Seed entities $\mathcal{S}$. (b) Direct seed links. (c) Expanded neighborhood. (d) Steiner tree subgraph.

Figure 4.2: Construction of a seed subgraph from a seed text. (a) Seed entities $\mathcal{S}$ extracted from the text. (b) The induced subgraph on $\mathcal{S}$ alone is sparsely connected. (c) One-hop expansion produces a denser but noisier graph $\mathcal{G}_{\text{expanded}}$. (d) Steiner tree extraction prunes $\mathcal{G}_{\text{expanded}}$ to a minimal connected structure spanning $\mathcal{S}$.

Figure 4.2 illustrates these steps on a single seed. The four panels show the sparsity of seed-only edges, the density introduced by one-hop expansion, and the structural pruning produced by the Steiner step.

4.3.2 Generating QA Pairs

Given a seed subgraph $\mathcal{G}'$, an LLM is asked to produce question-answer pairs that are grounded in the triples of $\mathcal{G}'$. The subgraph is serialized as a list of triples in textual form,

$$\mathcal{G}' = \{(h, p, t) \mid h, t \in \mathcal{V}',\ p \in \mathcal{P}'\},$$

and provided to the generator as input. The model is instructed to produce, for each item, a question q, an answer a, and an explicit answer path π that is a subset of $\mathcal{T}'$ identifying which triples justify the answer:

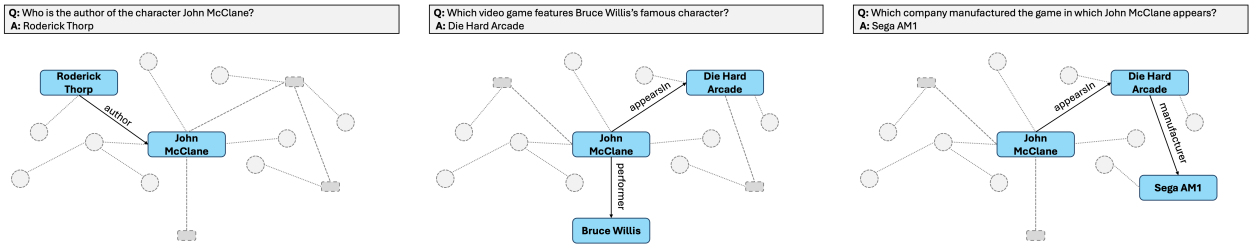
$$\text{GenerateQA}(\mathcal{G}') \rightarrow \{(q_i, a_i, \pi_i)\}_{i=1}^{N}, \qquad \pi_i \subseteq \mathcal{T}'. \tag{4.5}$$

Requiring an explicit π is the structural counterpart to evidence attribution. It forces the generator to commit to which triples were used, and it is what makes the verification step in Section 4.3.4 mechanical.

A single subgraph typically yields multiple QA pairs. The interconnected nature of a KG admits several reasoning paths through the same set of entities, and the generator can exploit this to produce questions that differ in directionality, hop count, and which subset of $\mathcal{T}'$ they engage. Figure 4.3 illustrates this on a small subgraph centered on a single seed entity. The first item asks for an entity reachable through a single incoming edge. The second requires reasoning over two distinct outgoing edges that connect the central entity to different relevant nodes. The third demands multi-hop traversal, in which an intermediate entity must be identified before the final answer is reached. The shared subgraph thus produces a small family of questions whose retrieval and reasoning demands differ in well-defined ways.

4.3.3 Randomized QA Generation

Two parameters introduce controlled variability into the generation step. The first is the decoding temperature T, which governs how stochastic the LLM's outputs are. Higher values of T promote diverse outputs at the cost of consistency; lower values produce more deterministic outputs. The second is a triple reordering seed R. Prior work has shown that



(a) Answer Path 1. (b) Answer Path 2. (c) Answer Path 3.

Figure 4.3: A single seed subgraph can support multiple QA pairs by exercising different answer paths $\pi_i \subseteq \mathcal{T}'$. The three panels show distinct reasoning paths over the same subgraph, ranging from a single-hop relation lookup to a multi-hop traversal through an intermediate entity.

the order in which inputs appear can materially affect generated outputs, even when the underlying content is unchanged [190, 192]. Dynamic-KGQA leverages this property by randomizing the serialization order of $\mathcal{G}'$ before passing it to the generator. Combining the two:

$$\text{DynamicQA}(\mathcal{G}', T, R) = \text{GenerateQA}(\text{Reorder}(\mathcal{G}', R), T), \tag{4.6}$$

where $\text{Reorder}(\mathcal{G}', R)$ produces a triple permutation determined by R. Holding $\mathcal{G}'$ and the prompt fixed, varying (T, R) across runs produces distinct surface realizations of items grounded in the same structural anchor. This is the mechanism by which fresh items are produced from run to run without reusing identical surface forms.

4.3.4 Verifying QA Pairs

Generation by an LLM is not a guarantee of grounding. Models can introduce entities or relations that do not exist in the subgraph, or claim support from triples that the subgraph does not contain. A verification step rules these cases out before items are admitted to the dataset. For each generated triple $(q_{ij}, a_{ij}, \pi_{ij})$, the framework checks whether the claimed

answer path π_{ij} is a subset of $\mathcal{T}'$:

$$\text{VerifyPath}(\pi_{ij}, \mathcal{G}') = \begin{cases} \text{True}, & \text{if } \pi_{ij} \subseteq \mathcal{T}', \\ \text{False}, & \text{otherwise.} \end{cases} \tag{4.7}$$

An item is retained only when every triple in π_{ij} belongs to $\mathcal{T}'$ and every entity referenced in π_{ij} belongs to $\mathcal{V}'$. This is a structural test on the same representation that produced the item, and it filters hallucinated grounding without requiring any further LLM call.

4.3.5 LLM-as-a-Judge for Quality Annotation

Structural verification ensures that an answer is supported by the subgraph but does not by itself ensure that the question is well-formed, non-redundant, fully answered, and adequate as an evaluation item. These quality dimensions are checked by a panel of LLM judges drawn from different research labs, following the rationale that a diverse panel reduces the influence of any single model's biases [133, 183, 53]. Each item is scored on four binary flags: *logical structure* (the question is grammatically and syntactically well-formed), *redundancy* (the question does not contain its own answer), *answer support* (the supporting triples justify the answer), and *answer adequacy* (the response fully addresses the question). Only items for which every judge marks logical structure, answer support, and answer adequacy as true and redundancy as false are retained. Unanimous agreement is required across the panel, and any disagreement causes the item to be discarded.

The four-flag schema is intentionally narrow. It does not attempt to evaluate question difficulty or interestingness, both of which are subjective and would invite judge-specific biases. It focuses instead on the validity and grounding properties that an evaluation item must satisfy before it can be used at all.

4.3.6 Dataset Construction and Artifacts

The full dataset is built by instantiating the pipeline above with concrete choices for the underlying knowledge graph, the seed corpus, and the LLM panel. The graph is YAGO

4.5 [141], which provides a contradiction-free ontology with reduced taxonomic loops and fewer redundancy issues than collaboratively maintained alternatives. Seed texts are drawn from Wiki-40B [60], where each entry corresponds to a Wikidata entity and provides a short text focused on that entity. Because YAGO 4.5 is itself derived from Wikidata, the entity space of the seed corpus aligns naturally with $\mathcal{V}$. Wiki-40B's train, validation, and test splits are inherited so that the resulting dataset has non-overlapping splits and inherits the freedom from leakage between phases of model development.

For QA generation and randomized variation, DYNAMIC-KGQA uses Claude 3.5 Sonnet v2 [6]. The LLM-as-a-Judge panel comprises Cohere's Command-R [35], Mistral Small 24.02 [146], and Amazon Nova Lite [70]. All judges run with default decoding parameters and temperature zero, while the generator uses the temperature schedule described in Section 4.3.3 to control variability. The choice of generator and judge models is treated as a design parameter rather than a fixed property of the framework: any sufficiently capable model can be substituted, with the same pipeline contract.

4.4 Dataset Properties and Evaluation

This section evaluates whether DYNAMIC-KGQA satisfies the requirements introduced above. The analysis focuses on five questions: how the dataset compares to prior KGQA benchmarks, what artifacts are available for auditing each item, how current LLMs perform on the benchmark, whether the generated items remain topically balanced, and whether dynamically generated variants remain distributionally stable across runs. These checks matter because a dynamic benchmark is only useful if it produces fresh items without changing the evaluation target.

Comparability with prior KGQA datasets. Table 4.1 situates DYNAMIC-KGQA among widely used KGQA benchmarks. The comparison reports the size of each split, the underlying knowledge base, whether each item is paired with a localized subgraph, and whether the dataset supports the generation of distributionally consistent dynamic variants.

Table 4.1: Comparison of KGQA datasets. Size (n) denotes QA pairs per split. *Base KG* is the underlying knowledge base. *QA Subgraph* indicates whether each item is paired with a localized subgraph. *Dynamic* indicates whether the dataset supports the generation of distributionally consistent dynamic variants.

Dataset	**Size (n)**			**Base KG**	**QA Subgraph**	**Dynamic**
	Train	**Test**	**Val**			
CWQ [145]	27,734	3,531	–	Freebase	×	×
WebQSP [177]	3,098	1,639	–	Freebase	×	×
GrailQA [58]	44,337	6,763	13,231	Freebase	×	×
QALD10-en [151]	–	–	394	Wikidata	×	×
SimpleQuestions [14]	75,910	21,687	10,845	Freebase	×	×
WebQuestions [10]	3,000	2,032	778	Freebase	×	×
DYNAMIC-KGQA	**200,000**	**40,000**	**40,000**	YAGO 4.5	✓	✓

The comparison can be read along four dimensions. The first concerns the choice of base knowledge graph. Every benchmark in the table other than QALD10-en and DYNAMIC-KGQA is built on Freebase [12], a knowledge graph that has been discontinued since 2015 [121]. A model evaluated on a Freebase-derived benchmark is therefore being scored against a snapshot of the world from over a decade ago, with schema and entity drift between Freebase and currently maintained graphs compounding the problem. Improvements reported on these benchmarks should be read narrowly: they speak to a system's ability to reproduce a stale view of the world, not to its ability to operate over current structured knowledge. The only Wikidata-based benchmark in the table, QALD10-en, illustrates the alternative failure mode: Wikidata is current and well populated, but its open collaborative maintenance produces taxonomic loops, redundant predicates, and contradictions that complicate automated reasoning over the graph [154]. YAGO 4.5 was constructed to remove these issues by enforcing a contradiction-

free ontology and reducing redundancy in the inheritance structure [141], which is why it is the natural base for an evaluation framework that requires programmatically grounded and verifiable items. The point is not that Wikidata is unsuitable in general but that, for evaluation, the cleanliness of the underlying graph directly bounds how reliable any structural verification step can be.

The second dimension concerns whether items are paired with localized subgraphs. None of the prior benchmarks in the table provide them, which is a consequential omission. Without a subgraph, retrieval and reasoning bottlenecks cannot be separated: a system that fails an item could be failing because it cannot find the relevant evidence in the full graph, or because it can find the evidence but cannot reason over it. With a subgraph attached to each item, an evaluator can isolate retrieval by providing $\mathcal{G}'$ directly and leave only reasoning under test, or conversely fix the reasoning prompt and vary the subgraph extraction step. Prior work has noted that subgraph extraction is itself an under-studied bottleneck for KGQA pipelines [45, 179]; a benchmark that ships with reference subgraphs makes such ablations possible without requiring each downstream user to recompute their own grounding.

The third dimension concerns size and split structure. Train/test/validation splits of 200,000/40,000/40,000 items provide enough capacity for fine-tuning, cross-validation, and final benchmarking on the same dataset, with each split inherited from a non-overlapping partition of Wiki-40B so that the splits are leakage-free with respect to one another. Several prior benchmarks in the comparison report only train and test partitions; the absence of a validation split is no longer adequate practice for modern model selection, because hyperparameter tuning on the test set inflates reported performance and obscures the very generalization the benchmark is meant to measure [56, 11].

The fourth dimension concerns dynamism. None of the prior datasets support distributionally consistent regeneration of evaluation items. Static benchmarks must be redrawn or reconstructed when contamination concerns arise, and any redraw breaks longitudinal comparability with prior reports. A benchmark that supports controlled regeneration from the same generation policy preserves longitudinal comparability without leaving items fixed

in a public test set. Both the localized-subgraph property and the dynamism property are first-class artifacts of DYNAMIC-KGQA and are direct consequences of the framework design described in Section 4.3.

Table 4.2: Per-item attributes of DYNAMIC-KGQA. The schema records the question and answer in both readable and URI form, the structural grounding (supporting facts and the local subgraph), and the full annotation trail produced by the panel of LLM-as-a-Judge models.

Name	**Definition**
`id`	Unique identifier for the QA pair.
`question`	Input question text.
`answer`	YAGO entity name.
`answer_readable`	Human-readable version of the answer.
`answer_uri`	YAGO entity URI.
`supporting_facts`	Triples supporting the answer.
`supporting_facts_uri`	YAGO URIs for the supporting triples.
`subgraph`	Local subgraph $\mathcal{G}'$ for the QA pair.
`subgraph_size`	Number of triples in $\mathcal{G}'$.
`logical_structure_flag_n`	Logical-structure flag from judge n.
`logical_structure_reasoning_n`	Rationale for the logical-structure flag from judge n.
`redundancy_flag_n`	Redundancy flag from judge n.
`redundancy_reasoning_n`	Rationale for the redundancy flag from judge n.
`answer_support_flag_n`	Answer-support flag from judge n.
`answer_support_reasoning_n`	Rationale for the answer-support flag from judge n.
`answer_adequacy_flag_n`	Answer-adequacy flag from judge n.
`answer_adequacy_reasoning_n`	Rationale for the answer-adequacy flag from judge n.

Per-item attributes. Table 4.2 lists the attributes carried by each DYNAMIC-KGQA item. The schema is organized in three groups. The first group identifies the question and

its answer in both human-readable and URI form, allowing the dataset to be used either for natural-language QA evaluation or for entity-resolution tasks where canonical references are required. The second group captures structural grounding through the supporting facts, their URIs, the local subgraph $\mathcal{G}'$, and its size. These fields make the answer path π inspectable and recover the explicit connection back to the structural anchor used during generation. The third group captures the LLM-as-a-Judge annotations: every quality flag is recorded for every judge, together with the rationale produced alongside it.

The choice to retain rationales alongside flags is a deliberate commitment to auditability. A flag alone records a binary decision; a rationale records the reasoning that produced the decision. Downstream users can re-aggregate items under different acceptance rules, for example by requiring two-of-three rather than unanimous agreement, or by relaxing the answer-adequacy threshold for short factoid questions, and they can inspect specific judge disagreements to understand where the panel was uncertain. The audit trail is also what makes the dataset re-analyzable as judging models improve: a future re-run with a stronger panel can be compared against the original annotations item-by-item rather than reduced to opaque score deltas.

Baseline performance and the contamination signal. A natural concern with any new dataset is whether it is harder than existing benchmarks for reasons unrelated to capability. Table 4.3 reports baseline scores for a panel of recent LLMs on Dynamic-KGQA together with widely used KGQA benchmarks, using standard prompting (IO) [18], Chain-of-Thought prompting (CoT) [160], and the Think-on-Graph (ToG) agent baseline [143], which iteratively traverses the underlying knowledge graph to assemble reasoning paths.

Three patterns are visible in the table. The first is the expected ordering by capability and prompting strategy: larger models score higher than smaller ones, CoT prompting outperforms IO across the board, and the ToG agent baseline outperforms CoT prompting alone in most columns. Llama 3.2 3B Instruct, the smallest model in the panel, scores lowest on every benchmark, consistent with prior reports of strong correlation between model scale

Table 4.3: LLM baselines on multiple QA datasets, including DYNAMIC-KGQA. Models are evaluated using standard prompting (IO), Chain-of-Thought prompting (CoT), and the Think-on-Graph (ToG) agent baseline. Bold indicates the best score in each column.

Method	Multi-Hop KBQA				Single-Hop KBQA	Open-Domain QA	Multi-Hop Dynamic KGQA
	CWQ	WebQSP	GrailQA	QALD10-en	SimpleQuestions	WebQuestions	DYNAMIC-KGQA
Llama 3.2 3B Inst. – IO	17.98	31.97	13.10	36.34	12.40	27.12	12.03
Llama 3.2 3B Inst. – CoT	25.43	46.92	21.20	44.44	16.80	41.39	16.07
Command-R – IO	24.36	48.38	19.80	36.04	18.90	41.73	16.88
Command-R – CoT	28.72	58.99	28.20	42.94	22.60	51.97	18.45
Mistral Small (24.02) – IO	29.09	51.86	25.60	38.74	19.70	44.59	19.49
Mistral Small (24.02) – CoT	37.10	60.65	32.70	46.85	24.10	53.44	22.18
Claude 3.5 Sonnet v2 – IO	30.16	51.68	26.00	36.64	25.30	44.78	19.78
Claude 3.5 Sonnet v2 – CoT	40.02	65.53	36.70	52.55	33.50	57.23	25.22
GPT-4o-mini – IO	28.18	49.36	25.90	36.64	22.20	43.26	18.50
GPT-4o-mini – CoT	35.06	61.62	32.90	44.14	25.80	54.13	21.14
GPT-4o – IO	33.81	53.57	27.00	39.64	26.00	45.57	24.44
GPT-4o – CoT	42.11	62.17	36.30	46.85	31.70	**54.53**	27.43
GPT-4o – ToG	**44.22**	**67.42**	**39.80**	**54.65**	**34.40**	53.67	**27.80**

and performance on knowledge-intensive tasks [67, 31]. These trends are not surprising in themselves; they confirm that DYNAMIC-KGQA is responsive to the same axes of capability that drive scores on older benchmarks, which is necessary for the dataset to function as a useful comparator.

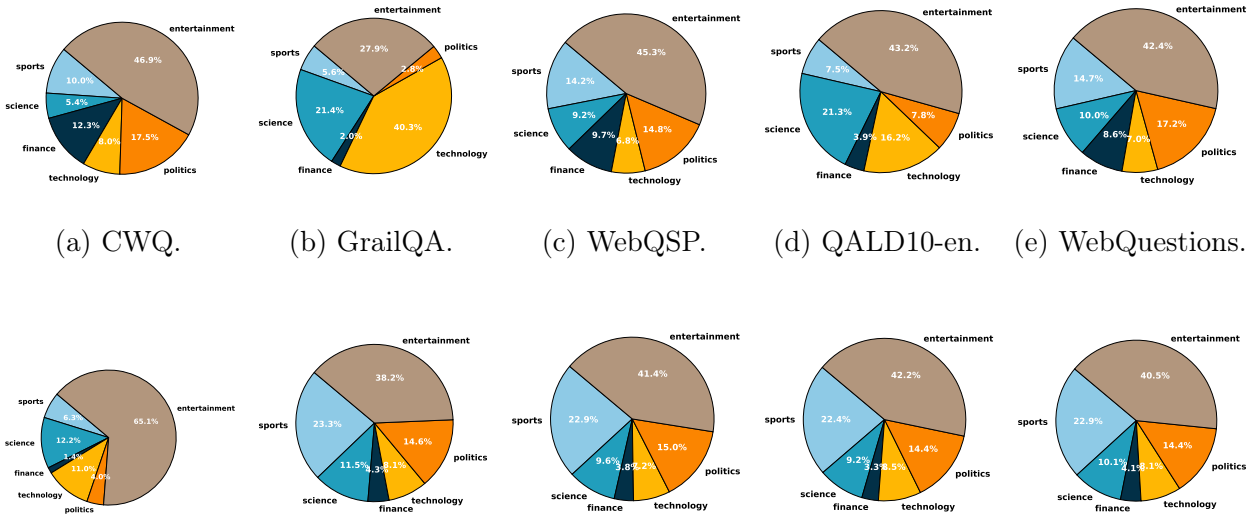
The second pattern is the diagnostically interesting one. Scores on DYNAMIC-KGQA are systematically lower than on older KGQA benchmarks at comparable hop counts. Models that score above 60% on WebQSP score in the high twenties on DYNAMIC-KGQA in the strongest agent configuration, and the gap is consistent across model families. The most natural explanation is the one that Chapter 3 makes precise: at the time of evaluation, DYNAMIC-KGQA items had not appeared in the public web in a fixed, indexable form and so could not have entered any of these models' pretraining mixtures or live retrieval pools. Older benchmarks have had a decade or more to be replicated, paraphrased, indexed, and discussed

in the public web. The score gap therefore admits a simpler reading than "Dynamic-KGQA is harder": it admits the reading that older benchmark scores blend capability with exposure, and that a freshly generated benchmark exposes the underlying capability more directly. The conservative interpretation is that some unknown share of the older-benchmark advantage is attributable to memorization rather than to reasoning; the contamination audit in Chapter 3 bounds that share from below.

The third pattern concerns transferability of agent-style methods. The reported gap between GPT-4o-ToG and GPT-4o-CoT is substantially smaller on Dynamic-KGQA than on the Freebase-based benchmarks where ToG was originally evaluated. ToG's published gains rely on Freebase-specific SPARQL templates, predicate selections, and prompt formulations that were tuned in conjunction with the underlying graph. Transferring the method to YAGO 4.5, with its different ontology and predicate vocabulary, does not preserve those gains intact. Read narrowly, this is a methodological warning rather than a critique of any specific agent system: improvements reported on a particular base KG do not generalize automatically to a different base KG, and a benchmark that supports evaluation under multiple KGs is necessary if reported gains are to be interpreted as gains in capability rather than as gains in fit to a particular grounding source.

Topic balance across datasets and runs. A KGQA benchmark whose items concentrate in one or two topical domains can produce inflated headline scores on those domains while masking weakness on others. Topic balance is therefore worth checking explicitly. Figure 4.4 shows the topic distribution of each KGQA benchmark in the comparison, with topics assigned by an off-the-shelf BART-MNLI zero-shot classifier [89] over six broad domains: sports, science, finance, technology, politics, and entertainment.

Two observations follow. First, several widely used benchmarks show substantial topical skew. SimpleQuestions is heavily weighted toward entertainment, GrailQA is concentrated in technology, and CWQ and WebQSP inherit Freebase's bias toward film, music, and biographical entries. These skews are not problems by themselves, but they should temper



(a) CWQ. (b) GrailQA. (c) WebQSP. (d) QALD10-en. (e) WebQuestions.

(f) SimpleQuestions. (g) Dynamic-KGQA (full). (h) Dynamic-KGQA Run 1. (i) Dynamic-KGQA Run 2. (j) Dynamic-KGQA Run 3.

Figure 4.4: Topic distribution of KGQA benchmarks. Topics are assigned by an off-the-shelf BART-MNLI zero-shot classifier [89] over six broad domains: sports, science, finance, technology, politics, and entertainment. Panels (a)–(f) show widely used KGQA benchmarks. Panel (g) shows the full Dynamic-KGQA dataset. Panels (h)–(j) show three dynamic runs of Dynamic-KGQA on the same 2,000-item slice with distinct triple reordering seeds at generation temperature $T = 0.8$.

how aggregate accuracy on these benchmarks is interpreted: a system that performs well on the dominant topic can dominate the headline number while remaining weak on the less represented domains. Second, Dynamic-KGQA shows a more even distribution across the six domains than any of its predecessors. The seed corpus drawn from Wiki-40B is itself broad, and the topical balance carries through to the generated items, which keeps aggregate accuracy from being driven by a single dominant domain.

The same figure, panels (h)–(j), shows three dynamic runs of Dynamic-KGQA on the

same 2,000-item slice with distinct triple reordering seeds and a generation temperature of $T = 0.8$. The pie charts are nearly indistinguishable across the three runs. This is the visual counterpart to the statistical analysis that follows, and it provides an independent qualitative check: if the surface form of items varies but the topical composition of the dataset does not, a model evaluated on Run 1 and another model evaluated on Run 3 are being asked questions drawn from the same topical mix, even though the questions themselves differ.

Dynamic consistency across runs. The framework's claim to dynamism rests on a precise empirical condition: successive runs must produce items that are genuinely fresh in surface form while remaining drawn from the same underlying distribution. "Fresh" means that a model cannot trivially recognize an item from a previous run by surface match alone. "Same distribution" means that scores from different runs are statistically comparable, so longitudinal evaluation does not silently confound benchmark drift with capability change. A benchmark that satisfies only the first condition produces items that look new but cannot be aggregated across runs; one that satisfies only the second is simply a static benchmark with paraphrases. Both conditions are required.

To test these conditions, three dynamic variants are generated for each item in a 2,000-item evaluation slice, using distinct triple reordering seeds R_1, R_2, and R_3 at a generation temperature of $T = 0.8$. The slice size is chosen as a compromise between coverage and the LLM-judging cost incurred per run; a large enough slice that the relevant tests have power, and a small enough slice that all three runs can be judged by the full panel within a reasonable budget. Each variant is then matched against the others and labeled along three categories: *identical*, when surface form matches exactly; *paraphrased*, when surface form differs but topic and answer are preserved; and *unique*, when surface form, reasoning path, or both differ substantively. Table 4.4 reports the resulting counts together with a χ^2 test of independence on per-run topic-label distributions and the corresponding Cramer's V effect size.

The freshness check follows directly from the Identical/Paraphrased/Unique counts. Across

Table 4.4: Pairwise comparison of DYNAMIC-KGQA variants across three runs (Run 1, Run 2, Run 3) on a 2,000-item slice. *Identical* counts items whose surface form matches exactly across runs, *Paraphrased* counts items whose surface form differs but whose topic and answer are preserved, and *Unique* counts items that differ in both surface form and reasoning path. The χ^2 statistic, p-value, and Cramer's V (ϕ_c) are computed on topic-label distributions across runs.

Comparison	**Identical**	**Paraphrased**	**Unique**	$\boldsymbol{\chi^2}$	p**-value**	$\boldsymbol{\phi_c}$
Run 1 vs. Run 2	8	308	1,684	3.33	0.6489	0.0288
Run 1 vs. Run 3	12	320	1,668	1.86	0.8679	0.0216
Run 2 vs. Run 3	15	313	1,672	3.45	0.6311	0.0293

all three pairwise comparisons, fewer than 0.01% of items are identical between any two runs (8, 12, and 15 of 2,000). Roughly 16% are paraphrases of items in the other run, and the remaining 84% are unique items differing in surface form, reasoning path, or both. A model encountering Run 3 after Run 1 therefore sees almost entirely new surface forms and a non-trivial fraction of items whose underlying reasoning path also differs. This is the operational meaning of "regenerated without overlap".

The distributional check is more delicate, and it is where the dissertation argument benefits from spelling out the statistical logic in full. The relevant comparison is between the topic-label distributions of two runs, where labels are assigned by the BART-MNLI zero-shot classifier into the six domains used in Figure 4.4. The null hypothesis H_0 is that the two distributions are independent samples from a common underlying topic distribution; the alternative H_1 is that they differ. The χ^2 statistic measures the discrepancy between observed pairwise counts and the counts that would be expected under H_0. With a significance threshold of $\alpha = 0.05$, a p-value below 0.05 would constitute sufficient evidence to reject H_0

and conclude that the distributions differ.

For all three pairwise comparisons, the observed χ^2 values are small (3.33, 1.86, 3.45) and the corresponding p-values are well above 0.05 (0.6489, 0.8679, 0.6311). There is therefore no statistical evidence to reject H_0. A non-rejection of H_0 is, strictly speaking, weaker than a positive demonstration of distributional equality: classical hypothesis testing provides evidence against H_0 when it can, and absence of evidence is not the same as evidence of absence. Reporting the p-value alone would be an incomplete description of the test outcome, especially with sample sizes large enough that even small distributional differences could in principle become detectable.

Cramer's V (ϕ_c) is reported alongside the p-value to address this gap. Cramer's V is an effect-size measure for χ^2 on contingency tables: it is bounded between 0 and 1, scales with the magnitude of the deviation between observed and expected counts, and is not driven by sample size in the way that χ^2 and p are. Conventional rules of thumb treat $\phi_c < 0.10$ as a negligible effect, $0.10 \leq \phi_c < 0.30$ as small, $0.30 \leq \phi_c < 0.50$ as moderate, and $\phi_c \geq 0.50$ as large. For the three pairwise comparisons, the observed ϕ_c values are 0.0288, 0.0216, and 0.0293, all an order of magnitude below the negligible-effect threshold. The interpretation is therefore not only that there is no statistical evidence of a distributional difference, but that any deviation that the test could not detect is bounded above by an effect size too small to be operationally relevant.

Taken together, the two signals are stronger than either alone. The non-significant p-values rule out any obvious distributional gap that the test would be expected to detect at $n = 2{,}000$. The small ϕ_c values bound the size of any deviation that the test could not detect. The pie charts in Figure 4.4(h)–(j) provide an independent qualitative cross-check: the visual stability across runs is consistent with the numerical results. The collective conclusion is that the runs draw from the same topic distribution to within an effect size that is operationally negligible, even as their surface form varies as documented in the freshness check.

Operational consequences for evaluation. The freshness, balance, and distributional-stability findings together support a specific operational claim about Dynamic-KGQA. Successive runs draw items from the same topical distribution while producing different surface forms, so a model that reports an accuracy on Run 1 and a different accuracy on Run 3 is being measured against statistically comparable test conditions. Aggregation, longitudinal comparison, and tracking-over-time become possible without reusing the items themselves. This is the property that distinguishes a contamination-resistant dynamic benchmark from a paraphrase or augmentation pipeline applied to a fixed test set: paraphrasing reduces verbatim overlap but inherits the seed items' difficulty distribution and any contamination already present in those seeds, while a structurally anchored framework such as Dynamic-KGQA decouples the items from the benchmark itself. The benchmark is no longer a fixed artifact; it is the policy that produces the artifact, and that policy can be re-run as evaluation conditions evolve.

4.5 Scope and Limitations

Dynamic-KGQA addresses a specific part of the evaluation problem identified in Chapter 3. It shows how dynamic generation, explicit grounding, and distributional consistency can be achieved in a structured setting. At the same time, its scope is bounded by the properties of knowledge graphs. Making this boundary explicit is important because the next chapter moves to the open web, where evidence is noisier, less structured, and more difficult to audit.

What the framework addresses. First, items are not drawn from a fixed public test set, and they can be regenerated indefinitely. The pretraining channel of Chapter 3 is therefore narrowed: there is no canonical benchmark page that a model can have memorized at training time, and there is no static surface form whose presence in C4 or a mirror site can be inflating performance. Second, every item carries an explicit answer path π within a local subgraph $\mathcal{G}'$. Correctness is checked structurally rather than by free-text matching against a long retrieved passage, which removes a major source of evaluator ambiguity. Third, generation

is parameterized by structural anchors and a fixed policy, so successive runs produce items whose distributional properties remain comparable. Longitudinal evaluation is supported without reusing the same items. Fourth, the LLM-as-a-Judge layer enforces well-formedness and grounding agreement across multiple independent models, so admitted items are not artifacts of a single generator's idiosyncrasies.

What remains outside its scope. The structured setting also leaves out several properties that a faithful information-seeking benchmark must eventually exhibit. The retrieval-time channel from Chapter 3 is sidestepped rather than exercised: items are grounded in subgraph triples, not in live web pages, so the framework does not measure how a model copes when retrieval is the actual bottleneck and when retrieved evidence is noisy, partial, or contradictory. Live web volatility is similarly absent. A KG entry for a public figure or an ongoing event does not change as quickly as a news page about the same entity, and Dynamic-KGQA does not test how well an agent handles freshness or sources whose content is updating in real time. Open-web sensemaking is also out of scope. Real information needs are seldom answerable from a single connected subgraph; they require reading across heterogeneous documents, weighing source quality, and aggregating partial evidence. The structured setting renders these problems trivial by construction, because the evidence has already been curated into a clean local graph. Finally, the user-facing dimension of an information-seeking agent, where the system must interpret intent, ask clarifying questions, and produce calibrated, attributed responses to a human, is not exercised here at all.

Framework-internal limitations. A second category of limitations is internal to the Dynamic-KGQA framework itself rather than to the structured paradigm. Three of these are worth stating explicitly. First, the framework inherits the limitations of its underlying knowledge graph: YAGO 4.5 is current and clean relative to alternatives, but every KG has gaps in coverage, lags real-world events, and reflects the editorial choices of its source. Items generated from such a graph cannot exceed the graph's own scope. This is shared by every

benchmark in Table 4.1; what distinguishes DYNAMIC-KGQA is that it inherits a maintained KG rather than a discontinued one, so the framework can be re-run with future versions of YAGO without redesigning its construction pipeline. Second, both the QA generator and the LLM-as-a-Judge panel are subject to the well-known failure modes of large language models, including hallucination, inconsistency under input perturbation, and biases inherited from pretraining [69, 167]. The structural verification step in Section 4.3.4 catches generator hallucinations that produce ungrounded answer paths, and the multi-lab LLM-as-a-Judge panel reduces the influence of any single model's idiosyncratic biases [133], but the framework cannot, on its own, certify that no shared bias remains across panel members. Third, the baseline coverage in Table 4.3 is representative rather than exhaustive: fine-tuning on KGQA-specific data, retrieval-augmented variants of CoT, and the broader space of agent-style traversal methods are not all covered, and several alternative configurations would likely move scores in either direction [29]. The reported numbers should be read as initial baselines for DYNAMIC-KGQA rather than as a comprehensive survey of the state of the art on this benchmark. None of these limitations alters the chapter's substantive claims; they simply mark where the empirical evidence in this chapter ends and where future work would extend it.

These boundaries define the role of DYNAMIC-KGQA in the dissertation. The framework shows what an auditable, contamination-resistant, and longitudinally comparable evaluation can look like when the structural side of the problem is held fixed. With that baseline in place, the next chapter extends the evaluation regime to the open web, where retrieval is the bottleneck, evidence is noisy, and sensemaking across sources becomes a first-class capability rather than a property granted by a clean graph.

4.6 Conclusion

DYNAMIC-KGQA shows that the validity threat identified in Chapter 3 can be addressed in a structured setting through dynamic generation, explicit grounding, and distributional checks. The framework anchors evaluation items in seed subgraphs of a curated knowledge graph,

generates fresh items at evaluation time using a fixed policy with controlled randomness, and verifies grounding both structurally and through a panel of independent LLM judges. The resulting dataset is large, paired with localized subgraphs, dynamically refreshable without distributional drift, and substantially harder for current LLMs than older Freebase-based benchmarks at comparable hop counts. Read alongside Chapter 3, the lower scores on DYNAMIC-KGQA are consistent with the interpretation that older benchmark scores partly reflect exposure, while freshly generated items provide a cleaner measurement target.

The chapter also makes clear where the structured setting stops. Subgraph grounding does not exercise live retrieval, web volatility, or cross-source sensemaking, and a KG-anchored benchmark cannot, on its own, test the open-environment behavior that an information-seeking agent must exhibit in deployment. Chapter 5 takes that next step. It moves from clean structured evidence to noisy open-web evidence, from KG-mediated grounding to live retrieval over heterogeneous sources, and from single-question correctness to multi-source synthesis under realistic information conditions.

Chapter 5

SENSEMAKING OVER DYNAMIC OPEN-WEB EVIDENCE

Structured dynamic evaluation addresses contamination and auditability, but it does not capture the full open-web setting in which information-seeking agents operate. This chapter moves from clean graph-grounded evidence to dynamic, query-conditioned web evidence, where the central challenge is not only finding relevant information, but organizing and synthesizing it across sources.

5.1 Introduction

Chapter 4 showed that contamination-resistant, distributionally consistent, and structurally auditable evaluation is possible when evidence is modeled as a structured graph representation. It also clarified the boundary of that setting. Items grounded in $\mathcal{G}'$ do not exercise live retrieval; the evidence has been pre-curated, the noise has been removed, and the query-conditioned slice of the open web that an ISA must read in deployment is replaced by a clean local subgraph. The capabilities most strongly associated with deployed information-seeking behavior, including searching across heterogeneous sources, weighing partial or conflicting evidence, and producing a coherent synthesis for the user, are therefore outside the scope of KGQA.

This chapter develops iAgentBench [41], an open-web evaluation framework for sense-making over dynamic evidence. The central design problem is not whether a model can find an answer in a clean evidence graph, but whether it can gather, organize, and integrate evidence across multiple sources whose content overlaps imperfectly. Instances are seeded from real-world attention signals, grounded in query-conditioned web corpora, and structured so that answering requires cross-theme synthesis rather than passage-level extraction or

shallow multi-hop stitching. The chapter describes the construction pipeline, the released artifacts that make the benchmark auditable, and an empirical study that separates evidence access from evidence integration.

The role of this chapter in the dissertation is to move evaluation closer to the open information environments that ISAs face in practice while preserving the contamination-awareness and auditability developed in earlier chapters.

5.2 From Structured Reasoning to Open-Web Sensemaking

A KG-grounded benchmark evaluates a specific slice of information-seeking behavior. The agent receives, or can construct, a small connected subgraph that is already known to support the answer. The retrieval problem reduces to subgraph traversal, and the reasoning problem reduces to following an explicit path of typed relations. These are important capabilities, and Chapter 4 showed how to evaluate them under a contamination-controlled and distributionally consistent regime. However, structured QA does not test the behaviors that distinguish open-web information seeking from graph reasoning.

Five properties of the open-web setting matter for this chapter. The first is that evidence is *query-conditioned* rather than fixed. There is no global, persistent corpus that an agent can pre-index and reuse across queries; the relevant evidence is the slice of the open web that the system retrieves at test time, given a particular query and a particular point in time. Evaluating an agent under this condition requires holding the retrieval slice fixed across systems so that comparisons are meaningful, which in turn requires recording the retrieved slice as part of the benchmark instance.

The second property is that the evidence is *dynamic and time-indexed*. For an ongoing event, the available evidence can change substantially from one day to the next. A benchmark built around a fixed retrieval snapshot quickly becomes outdated as the web changes. By contrast, a benchmark defined by a generation policy and a time window can be rerun, allowing the evaluated slice to move forward with the evidence. This is the open-web counterpart of the dynamic-consistency property in Chapter 4.

The third property is that answering requires *cross-source synthesis*. Real informational needs are rarely answerable from a single passage or even from a single document. Evidence is distributed across multiple sources whose content partially overlaps, sometimes contradicts, and rarely aligns at the surface level with the query. The bottleneck is not finding one matching span but integrating information across themes whose relations the system has to recover.

The fourth property is that retrieved evidence is *heterogeneous and partially overlapping*. Different sources give different framings of the same event, attribute different actors to the same outcome, and emphasize different downstream effects. Sensemaking under this condition is qualitatively different from multi-hop traversal of a clean graph. It requires the agent to recognize when two sources are talking about the same theme, when they are talking about adjacent themes, and when the link between two themes is itself the thing being asked about.

The fifth property is that the questions an information-seeking system must answer reflect *realistic user informational needs*. Users do not, in deployment, ask the kinds of questions that knowledge-base benchmarks are populated with. They ask why something happened, what its consequences were, how two events are connected, what the stakes are, and what triggered a particular response. These intent patterns are common across topics and have a structure that can be captured explicitly, and a benchmark that ignores them ends up testing a distribution of questions that no actual user is asking.

Multi-hop QA might appear to capture the same requirement, but the overlap is limited. Many widely used multi-hop datasets reward locating a small number of lexical matches or short supporting statements [172]. The task often becomes path-following and snippet stitching: identify a bridge entity, retrieve a downstream fact, and combine the result. This is mechanically multi-document, but it is not the same as sensemaking. Sensemaking is closer to what an analyst does when reading several articles about an unfolding event and producing a concise answer that depends on the relationship between themes, not on any single article in isolation. A benchmark that targets this operation must enforce, at construction time, that no single passage in the retrieval slice contains the full answer and that the answer follows

from the relation between two or more themes. The pipeline in Section 5.3 is built around this constraint.

The dataset-level summarization framing of Edge et al. [48] is complementary to this goal. Their setting tests whether a system can produce query-focused summaries of a fixed corpus, which is appropriate for a particular class of analytical workloads. The open-web ISA setting is different: the corpus is the small slice retrieved for a query at a particular time, not a persistent global repository. iAGENTBENCH therefore draws on structural ideas from GraphRAG, such as story graph representations and theme-level community detection, but redesigns them around the ISA setting: query-conditioned, time-indexed slices of the live web whose value depends on source coverage, cross-source synthesis, evidential grounding, and the evolving informational properties discussed earlier.

5.3 Benchmark Construction Pipeline

The construction pipeline starts from a traffic-driven topic, retrieves a query-conditioned web corpus, builds a structured representation of the themes in that corpus, and generates questions whose correctness depends on multiple themes and the links between them. The pipeline has five stages: interest-driven seed selection, story-graph construction, community detection and role assignment, packet construction, and QA generation with judge filtering. Figure 5.1 summarizes the pipeline, and the remainder of this section describes each stage.

5.3.1 Interest-Driven Seed Selection

The first design choice concerns where seed topics come from. The DYNAMIC-KGQA [39] pipeline can select seeds from any concept in its underlying graph, since every concept is a valid grounding anchor. However, this does not necessarily reflect the queries users would typically issue. Selecting seeds from a curated knowledge base can produce topics that no user is actively searching for, while selecting seeds from a fixed editorial source can skew the benchmark toward whatever that source happens to cover. Both choices fall short of the realism requirement in Section 5.2.

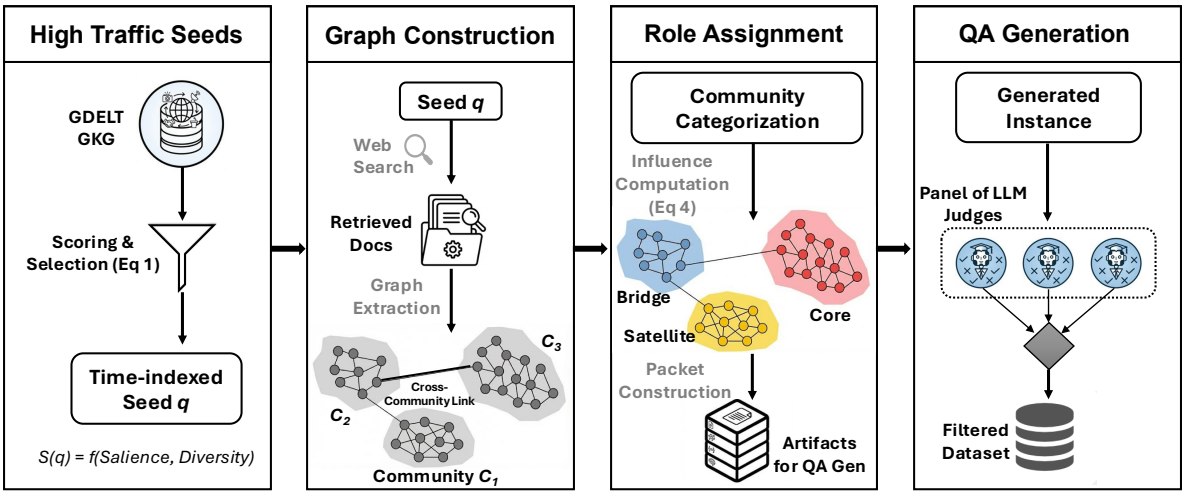


Figure 5.1: Overview of the iAGENTBENCH construction pipeline. Time-indexed seed queries are sampled from public attention signals, query-conditioned web corpora are retrieved, story graphs and thematic communities are extracted, compact cross-theme packets are constructed, and QA pairs are generated and filtered using a panel of LLM judges.

iAGENTBENCH draws seeds from the GDELT 1.0 Global Knowledge Graph (GKG) [147], which provides daily snapshots of entities and event descriptors derived from worldwide news streams. Each day yields a candidate set of event-style seeds by combining salient entities with event descriptors mined from GKG records. Aggregating across a time window produces a pool of candidates whose composition reflects shifting public attention. The pipeline does not assume that every candidate is a good seed; it scores them and selects only those that are salient, temporally specific, and topically diverse.

Concretely, given a candidate seed pool $\mathcal{Q}$ aggregated over a time window, each candidate $q \in \mathcal{Q}$ receives a score

$$\begin{aligned} S(q) \;=\; & \log\bigl(1 + A(q)\bigr) + \alpha\,\mathrm{Geo}(q) + \beta\,\mathrm{Freq}(q) \\ & + \gamma\,\Delta t(q) + \delta\,\mathrm{Spec}(q) - \eta\,\mathrm{Cov}(q), \end{aligned} \tag{5.1}$$

where $A(q)$ is an attention proxy when available, $\mathrm{Geo}(q)$ is geographic breadth, $\mathrm{Freq}(q)$ is

observation count, $\Delta t(q)$ is the date span covered by the signal, $\mathrm{Spec}(q)$ is a specificity bonus that downweights generic seeds, and $\mathrm{Cov}(q)$ penalizes candidates that occur on nearly every day in the window because they typically index generic terms rather than specific events. The weights $\alpha, \beta, \gamma, \delta, \eta$ are tunable, but the structure of the score reflects a small number of design commitments: prefer events with real attention, prefer events with geographic and temporal specificity, and avoid both topical novelty seekers and broad noise terms. The output of this stage is a ranked list of time-indexed seeds whose composition tracks live public interest rather than the editorial choices of a fixed source.

5.3.2 Story-Graph Construction

Given a selected seed query q, the pipeline retrieves a query-conditioned web corpus $\mathcal{D}(q) = \{d_i\}_{i=1}^{N}$ via a standard search API. This is the operational evaluation setting that iAGENTBENCH targets: an agent that can search, read, and synthesize from the small set of relevant sources returned at test time. The framework explicitly does not assume a persistent global representation of an entire corpus such as the open web. The retrieved slice is the corpus the agent must work with, and the benchmark records that slice so that downstream comparisons hold the evidence fixed across systems.

To support cross-document sensemaking, the pipeline constructs a query-specific structured representation that aggregates information across $\mathcal{D}(q)$. Following GraphRAG-style pipelines [48], an LLM-assisted extractor identifies entities and relational assertions from text. The resulting structure is not a schema-fixed knowledge graph: the relations are short natural-language claims grounded in evidence pointers, which makes the representation closer to a hypergraph of grounded assertions than to a predicate-limited KG. The pipeline calls this structure a *story graph*.

The graph is

$$G(q) = (V, E), \tag{5.2}$$

where each node $v \in V$ corresponds to an entity and each edge $e \in E$ is a tuple (s, r, o, ev)

with subject $s \in V$, object $o \in V$, relation text r as a claim-like statement, and evidence pointers ev $\subseteq \mathcal{U}$ to the underlying text units (sentences or chunks). Each edge also carries a support signal such as edge weight, frequency, or evidence mass that records how strongly the relation is grounded in $\mathcal{D}(q)$. Recording ev at edge level is what makes downstream auditing possible: every claim that the pipeline uses for question generation is traceable back to a specific span in a specific retrieved document.

The story graph is partitioned into communities $\mathcal{C}(q) = \{c_1, \ldots, c_m\}$ using Leiden clustering [150]. Each community is interpreted as a *theme*, a coherent sub-story within the retrieved corpus. For each community c, the pipeline produces a short natural-language summary and a set of grounded findings, which are salient statements tied to evidence pointers. These artifacts give the benchmark a compact, human-readable interface to the thematic structure of the retrieved corpus, which is the level of representation at which sensemaking questions naturally live.

5.3.3 Community Roles and Influence

A query-conditioned story graph is typically large and noisy. It contains many peripheral entities, redundant edges, and themes that are not equally important for understanding the query. To focus question construction on the most consequential parts of the corpus, the pipeline computes a theme-level view of the graph and assigns each community a structural role.

The theme-level view is a community meta-graph

$$H(q) = (\mathcal{C}(q), E_H), \tag{5.3}$$

in which each node is a community and each edge aggregates the cross-community relations induced by $G(q)$. The weight of an edge $(c_i, c_j) \in E_H$ reflects both the strength and the quantity of cross-community evidence, so a heavier meta-edge corresponds to a thematic relationship that is repeatedly grounded in the retrieved corpus rather than to a single incidental mention.

Each community c receives an influence score

$$\begin{aligned} I(c) \;&= \tfrac{1}{4}\, z\big(\log(1+|c|)\big) + \tfrac{1}{4}\, z\big(\mathrm{PR}(c)\big) \\ &\quad + \tfrac{1}{4}\, z\big(\mathrm{BC}(c)\big) + \tfrac{1}{4}\, z\big(\mathrm{Ev}(c)\big), \end{aligned} \tag{5.4}$$

where $|c|$ is community size, $\mathrm{PR}(c)$ and $\mathrm{BC}(c)$ are PageRank and betweenness centrality on $H(q)$, $\mathrm{Ev}(c)$ counts the number of unique evidence units supporting c, and $z(\cdot)$ denotes within-query standardization. The four components of $I(c)$ correspond to four distinct ways in which a theme can matter inside a query-conditioned corpus: it can be large in the number of entities it contains, well connected to other themes through PageRank, structurally pivotal as measured by betweenness, or heavily grounded in retrieved evidence.

Three roles are assigned on the basis of $I(c)$ and $\mathrm{BC}(c)$. The CORE set comprises the highest-influence themes,

$$\mathcal{C}_{\mathrm{core}}(q) = \mathrm{TopK}\big(\mathcal{C}(q),\, I,\, k_{\mathrm{core}}\big), \tag{5.5}$$

with $k_{\mathrm{core}} = \max\{5,\, 0.3\,|\mathcal{C}(q)|\}$. The BRIDGE set comprises the highest-betweenness themes,

$$\mathcal{C}_{\mathrm{bridge}}(q) = \mathrm{TopK}\big(\mathcal{C}(q),\, \mathrm{BC},\, k_{\mathrm{bridge}}\big), \tag{5.6}$$

with $k_{\mathrm{bridge}} = \max\{3,\, 0.2\,|\mathcal{C}(q)|\}$. SATELLITE themes are the remaining communities that are strongly linked to a CORE or BRIDGE theme in $H(q)$: c is marked SATELLITE if there exists $c' \in \mathcal{C}_{\mathrm{core}}(q) \cup \mathcal{C}_{\mathrm{bridge}}(q)$ such that $\max\{w(c,c'),\, w(c',c)\} \geq \tau_q$, where $w(\cdot,\cdot)$ is the meta-edge weight and τ_q is set to the median meta-edge weight for query q.

These roles do real work later in the pipeline. They bias question construction toward themes that are structurally important to the query rather than toward whatever happens to appear first in the retrieved corpus, and they distinguish themes that mediate between sub-stories from themes that merely accompany them. The same role assignments are also released with each instance, which means that downstream users can re-run the same role-driven selection under different acceptance rules without having to recompute the meta-graph from scratch.

5.3.4 Connector Relations and Packet Construction

Sensemaking questions depend on multiple themes and on the way those themes are connected. The pipeline therefore extracts *connector relations* that tie communities together. Let $\text{comm} : V \to \mathcal{C}(q)$ map each entity to its community. An edge $e = (s, r, o, \text{ev}) \in E$ is a connector if it crosses community boundaries, that is, $\text{conn}(e) = \mathbb{I}\big[\text{comm}(s) \neq \text{comm}(o)\big]$. Not all cross-theme assertions are equally informative, so a small top-K set of connector edges is retained, ranked by their support signal. This favors links that are repeatedly grounded in $\mathcal{D}(q)$ while keeping packets compact.

Generating questions directly from $G(q)$ would expose the LLM generator to the full document set and the full graph, producing context that is too long to be reliably attended to and too rich to enforce cross-theme dependence. The pipeline therefore constructs *packets*, compact bundles that preserve only the information required for cross-theme reasoning. A packet is built around a small set of themes that exposes a meaningful cross-theme linkage while remaining token-efficient. Construction begins with a high-influence CORE theme and pairs it with one or more BRIDGE themes that connect it to other regions of the meta-graph. Most bundles contain two themes (a salient CORE and a connecting BRIDGE); three-theme bundles such as CORE–BRIDGE–CORE are used when needed to preserve a connector chain. A bundle is retained only if at least one selected connector relation links its themes, which is the structural guarantee that the resulting QA candidates depend on cross-theme connections rather than isolated fact lookups.

Given a selected bundle B, the packet $P(B)$ contains each theme's community card, that is, the summary and top findings with evidence IDs, and the connector relations linking themes in the bundle. The representation is intentionally compact yet preserves the ingredients required for cross-document sensemaking: who the themes are, what the salient findings are inside each theme, and how the themes connect at the level of grounded claims.

5.3.5 QA Generation and Judge Filtering

Conditioned on a packet $P(B)$, an LLM generator produces one-sentence, user-like questions with short, objective answers grounded in the provided findings and connector relations. The generator is prompted to bias toward realistic sensemaking intents (trigger, consequence, cross-theme connection) and explicitly to discourage trivia-like entity lookups. Decoding is deterministic at temperature zero, and the generator is required to output structured bookkeeping fields alongside each candidate: the required communities, the supporting findings (at least one from each required community), and the supporting connector IDs. The bookkeeping fields are not shown to models under evaluation, but they are what enable downstream filtering and analysis to ensure that questions require cross-theme integration rather than a single isolated fact.

Candidates pass through an LLM-as-a-Judge filter [183, 53]. The pipeline acknowledges that LLM judgments have documented reliability issues for subjective tasks, but the checks here are intentionally narrow and evidence-only. The judge verifies (a) that a candidate is supported by the provided artifacts and (b) that it necessarily depends on multiple themes and at least one connector. Both of these reduce to fact verification and natural language inference, which are settings where LLM judgments have demonstrated strong performance [59]. The framework employs a panel of three judge LLMs from independent providers; each judge independently evaluates a candidate and returns a structured decision with binary flags and short, artifact-grounded reasons. A candidate is retained only if it satisfies all hard criteria and passes the necessity tests. Aggregation is by majority vote, with an additional veto rule: a candidate is rejected if any judge flags missing evidence, non-unique answers, or trivia drift. The panel configuration prioritizes precision over recall, on the assumption that a smaller benchmark of well-grounded sensemaking questions is more useful than a larger benchmark with mixed quality.

5.4 Released Artifacts and Auditability

Open-web evaluation introduces failure modes that do not appear in a clean structured benchmark. An agent may fail to retrieve the relevant evidence, retrieve it but miss the relevant theme, recognize the themes but fail to integrate them, or integrate them and still produce an unsupported answer. Reporting only end-to-end accuracy collapses these cases into a single number. iAGENTBENCH therefore releases intermediate artifacts with each instance so that failures can be analyzed at specific stages of the pipeline.

The release schema is organized around four design goals. The first is realism (G1): seed queries are sourced from real-world attention signals via Section 5.3.1 rather than from a curated knowledge base or a quiz-style collection, and they are paired with query-conditioned web corpora that reflect the bounded evidence slice an information-seeking agent would actually read at test time. The second is cross-theme dependence (G2): each retained question is explicitly tied to at least two themes and at least one connector relation, and the supporting findings and connectors are released with their IDs so that an evaluator can diagnose whether a model failed at retrieval or at synthesis. The third is information-seeking fidelity (G3): questions are generated under five recurring intent patterns that capture how users typically express informational needs, namely *explainer*, *connection*, *trigger*, *consequence*, and *stake*, and the intended pattern is recorded per instance to support stratified analysis. The fourth is auditability (G4): the released artifacts are sufficient to trace each question back through the construction pipeline and to regenerate compatible instances over new time windows.

Table 5.1 summarizes the per-instance schema. Three properties of the schema are worth stating explicitly because they support the dissertation argument. First, recording `retrieved_sources` and `text_units` together with the question makes the benchmark amenable to contamination checks of the kind described in Chapter 3. The retrieved slice is recorded, so an evaluator can audit whether any answer-bearing passage in the slice originates from a benchmark mirror rather than from genuine web evidence, and the audit can be

performed on the same artifact that the model under test consumed. Second, releasing the story-graph artifacts (`communities`, `community_cards`, `connectors`, and `packet`) allows failures to be localized to specific stages of the pipeline rather than collapsed into an end-to-end accuracy number. A model that retrieves the right documents but produces a wrong answer is in a different operating regime from a model that fails to retrieve the relevant theme at all, and the released artifacts make this distinction recoverable from any evaluation run. Third, the `intent_pattern` field supports stratified analysis across the five user-intent categories. A system that performs well on *explainer* intents but degrades on *connection* or *trigger* intents is exhibiting a specific weakness, and a benchmark that treats all intents as a single bucket cannot reveal it.

The release also supports controlled regeneration. Because the construction rules are deterministic given a seed and a time window, and because the retrieved corpora are recorded, an instance can be regenerated for a future time window using the same pipeline, yielding a fresh evaluation slice without altering the construction policy. This is the open-web counterpart of the dynamic-consistency property described for DYNAMIC-KGQA [39] in Chapter 4: the framework is the artifact, and the released instances are a snapshot that can be refreshed under unchanged generation rules. The combination of recorded slices, intermediate artifacts, intent labels, and judge decisions is what distinguishes iAGENTBENCH from a static QA benchmark released as a CSV. It is intended as a methodological contribution rather than a one-time leaderboard target.

5.5 Experimental Results

The experimental study evaluates how evidence access and evidence integration affect end-to-end QA performance. Four widely used LLMs (Claude Sonnet 4.5, LLaMA 4 Maverick 17B, Mistral Large 3, and Gemma 3 27B) are evaluated under three inference settings: *Base*, where the model answers directly without external evidence; *RAG*, where the model is given the first page of retrieved documents from SearxNG [134]; and *Reflexion*, where an agentic self-reflection loop is applied over retrieved evidence [138]. Performance is reported on

SimpleQA [161], HotpotQA [172], and iAgentBench. Because end-to-end agent evaluation is expensive, 500 questions are sampled from each benchmark, following the protocol used in prior work [142]. Accuracy is computed using the SimpleQA evaluation procedure [161].

Model	SimpleQA					HotpotQA					iAgentBench				
	Base	RAG	Refl.	Δ_{RAG}	Δ_{Refl}	Base	RAG	Refl.	Δ_{RAG}	Δ_{Refl}	Base	RAG	Refl.	Δ_{RAG}	Δ_{Refl}
Claude Sonnet 4.5	0.240	**0.740**	0.716	0.500	−0.024	0.574	0.700	**0.790**	0.126	0.090	0.584	0.648	**0.682**	0.064	0.034
LLaMA 4 Maverick 17B	0.174	0.657	**0.826**	0.483	0.169	0.456	0.537	**0.758**	0.081	0.221	0.356	0.532	**0.628**	0.176	0.096
Mistral Large 3 (675B)	0.196	0.730	**0.820**	0.534	0.090	0.500	0.654	**0.720**	0.154	0.066	0.486	**0.638**	0.564	0.152	−0.074
Gemma 3 27B	0.092	0.700	**0.764**	0.608	0.064	0.420	0.548	**0.694**	0.128	0.146	0.432	**0.592**	0.570	0.160	−0.022

Table 5.2: Accuracy of Base, RAG, and Reflexion across SimpleQA, HotpotQA, and iAgentBench. Δ_{RAG} = RAG − Base measures the gain from evidence access; Δ_{Refl} = Refl − RAG measures the gain from agentic self-reflection beyond RAG.

Three patterns are visible in Table 5.2 and the two scatter plots that decompose it (Figures 5.2 and 5.3). They are the basis of the dissertation argument that retrieval helps, but evidence access alone does not solve sensemaking-oriented questions.

The first pattern is that retrieval substantially improves performance across all datasets and models. In Figure 5.2, every point lies above the $y = x$ line, which means that access to external evidence is beneficial even for strong frontier models. The magnitude of this effect, however, differs systematically across benchmarks. SimpleQA exhibits the largest Base→RAG jumps. SimpleQA was constructed to be adversarially difficult for base LLMs and is constrained to short questions with a single, indisputable answer [161], which produces low Base accuracy across models. Once retrieval is enabled, SimpleQA becomes the easiest benchmark for most models, suggesting that its remaining difficulty is dominated by evidence access and precise extraction rather than by reasoning over the retrieved evidence. HotpotQA also benefits from retrieval, but the lifts are more modest, consistent with a setting where evidence access and multi-hop composition jointly matter. iAgentBench likewise improves under RAG, but the gap between Base and RAG is smaller in absolute terms and the residual

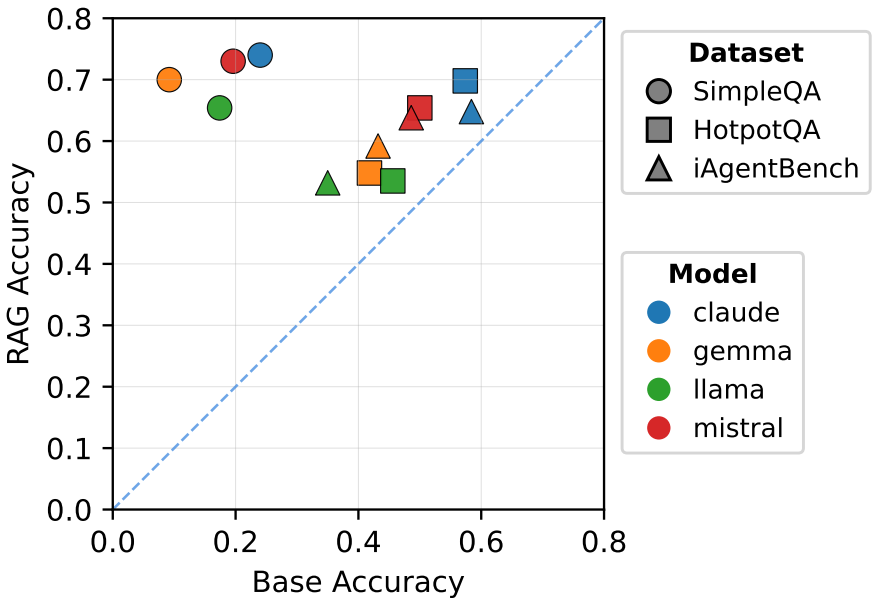


Figure 5.2: Base vs. RAG accuracy across datasets and models. Points above the diagonal indicate gains from evidence access. The magnitude of the lift differs systematically across benchmarks: SimpleQA shows the largest jumps, HotpotQA is intermediate, and iAgentBench shows smaller per-model lifts and a residual gap that remains under RAG.

gap to perfect performance remains substantial. The interpretation that this section advances is the one that motivates the chapter: on iAgentBench, supplying evidence is only part of the challenge, because the questions require cross-theme integration that retrieval alone does not deliver.

The second pattern concerns the additional effect of multi-step self-reflection beyond RAG. While Δ_{RAG} is uniformly positive, Δ_{Refl} is mixed. Reflexion improves over RAG when extra steps help connect dispersed evidence, but it also regresses when multi-step reasoning introduces drift or over-correction. This behavior is visible across all three datasets and is especially salient for iAgentBench, where models split on whether iteration helps or hurts. Claude Sonnet 4.5 and LLaMA 4 Maverick benefit from Reflexion on iAgentBench, while Mistral Large 3 and Gemma 3 regress relative to RAG. The same Reflexion loop, applied to

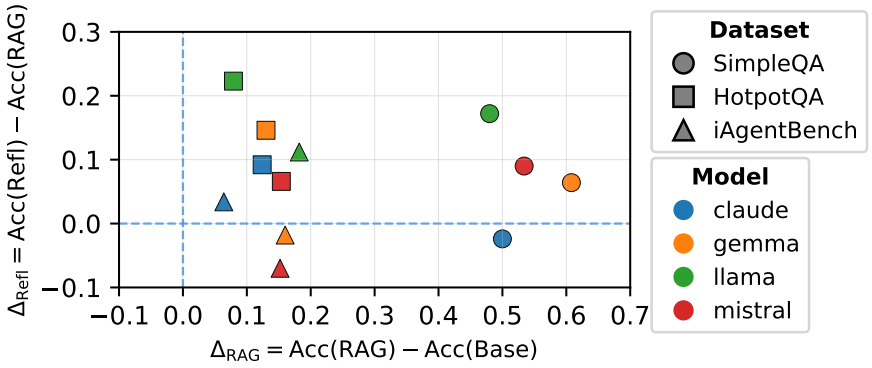


Figure 5.3: Retrieval gains versus agentic gains, decomposing improvements into $\Delta_{\text{RAG}} = \text{Acc(RAG)} - \text{Acc(Base)}$ and $\Delta_{\text{Refl}} = \text{Acc(Refl)} - \text{Acc(RAG)}$. Positive Δ_{Refl} means iteration helps beyond RAG; negative values indicate regressions.

the same retrieval slice, produces different signs of effect depending on the model. This is the kind of finding that headline accuracy does not surface; reading the per-row Δ_{Refl} column is what reveals it.

The third pattern concerns the structural difference between SimpleQA, HotpotQA, and iAgentBench when retrieval is held constant. On SimpleQA, several models show large gains from Reflexion beyond RAG, with LLaMA and Mistral both gaining double-digit accuracy. The interpretation is straightforward: SimpleQA's residual difficulty is concentrated in extraction, and an iterative loop that lets the model re-read evidence and revise its answer reduces extraction noise. On HotpotQA, Reflexion yields a sizable boost for LLaMA and smaller improvements for the other models, consistent with benefits for multi-hop composition that depend on which model can leverage the iteration step. On iAgentBench, the gains are smaller and the regressions more visible, which is the empirical signature of a benchmark whose remaining difficulty is concentrated in cross-theme synthesis rather than in extraction or path-following. Iteration over the same retrieval slice does not help when the bottleneck is integration; it can hurt when an additional step gives the model another opportunity to mis-aggregate evidence.

The combined picture is the one the chapter set out to expose. Evidence access is the dominant lever on benchmarks whose difficulty is concentrated in finding the right passage, and retrieval-augmented generation closes most of that gap. Iteration is a supplemental lever that helps when the residual difficulty is extraction or path-following, but its benefit is model-specific and not monotone. On benchmarks where the residual difficulty is cross-theme integration, neither lever is sufficient by itself: retrieval narrows the gap but does not close it, and iteration over the same slice is unstable. The empirical signature of iAgentBench relative to SimpleQA and HotpotQA is consistent with the framing of Section 5.2, where the benchmark was constructed precisely to make cross-theme integration the binding constraint.

5.6 Discussion

This chapter connects the dissertation's contamination-aware evaluation agenda to the open-web setting. Chapter 3 showed why static QA benchmarks can no longer be treated as clean measurements of information-seeking capability under realistic exposure. Chapter 4 developed a controlled dynamic regime in which generation, grounding, and verification are auditable. The present chapter extends that response to dynamic web evidence, where the central capability under test is not correctness over a clean evidence graph, but cross-source synthesis under realistic information conditions.

Three points are worth stating explicitly about what iAgentBench adds beyond Dynamic-KGQA. The first concerns the evidence model. Dynamic-KGQA grounds items in a curated subgraph $\mathcal{G}'$, which guarantees that the evidence the system needs is structurally present and well-formed. iAgentBench grounds items in a query-conditioned slice $\mathcal{D}(q)$ of the open web, which guarantees only that the evidence is in scope; whether the system can find, organize, and integrate it is precisely what is under test. The two evidence models support distinct claims: Dynamic-KGQA isolates reasoning over clean structure; iAgentBench probes retrieval and synthesis over realistic noise. A capable ISA must perform well in both regimes, and benchmarks for the two regimes are not interchangeable.

The second point concerns the question distribution. Dynamic-KGQA's questions are

anchored to seed subgraphs and exercise multi-hop reasoning over typed relations. The intent distribution is implicitly determined by what can be asked of a structured graph: identification, attribute lookup, and relational chaining. iAgentBench explicitly enforces a different distribution by generating questions under five recurring user-intent patterns (*explainer*, *connection*, *trigger*, *consequence*, *stake*). These patterns reflect the kinds of informational needs that users actually express when they are trying to make sense of an evolving topic, rather than the kinds of questions that a knowledge graph happens to support cheaply. Coupling the question distribution to user intent rather than to KG affordances is what moves the evaluation regime closer to the fulfillment of an actual information need.

The third point concerns failure decomposition. Headline accuracy on a structured benchmark conflates retrieval and reasoning at the level of subgraph traversal; on an open-web benchmark, headline accuracy conflates retrieval, theme recognition, cross-theme integration, and answer formation. The released artifacts in Section 5.4 let an evaluator decompose these stages, and the mixed Reflexion results in Section 5.5 show why such a decomposition matters. Iteration on the same retrieval slice produces different signs of effect across models. A single accuracy number cannot tell the difference between a model whose retrieval is the bottleneck, a model whose theme recognition is the bottleneck, and a model whose synthesis step is unstable. A benchmark that supports decomposition can, and an ISA evaluation regime that does not support decomposition is not really evaluating information-seeking behavior.

The chapter does not claim to have solved ISA evaluation. Two limitations are worth restating. First, iAgentBench grounds an information need in a query-conditioned slice of the open web, which is a faithful proxy for the evidence an agent reads at test time but is not a faithful proxy for the user. The benchmark does not exercise clarification dialogue, intent disambiguation, or multi-turn refinement. Second, the framework leaves the live retrieval mechanism as a fixed component. The retrieval slice is recorded so that comparisons across systems are well-defined, which is what makes the empirical study possible, but it also means that the benchmark does not measure how a model would behave under a different retrieval policy. Both of these are appropriate scoping choices for a benchmark that targets

sensemaking over retrieved evidence, and both are flagged here so that subsequent work can extend the regime in either direction without misreading the present chapter's claims.

5.7 Conclusion

iAgentBench extends the dissertation's evaluation regime from clean structured grounding to the open-web setting that ISAs face in deployment. It anchors items in time-indexed, traffic-driven topics; grounds them in query-conditioned web corpora recorded as part of the benchmark instance; represents the corpus as a story graph with explicit communities and connectors; and generates questions whose correctness depends on cross-theme integration rather than passage-level extraction or shallow multi-hop stitching. The released artifacts allow evaluators to audit the construction pipeline, decompose end-to-end failures into retrieval, theme recognition, integration, and answer formation, and regenerate compatible instances over future time windows.

The empirical study makes the chapter's central point concrete. Retrieval reliably improves performance across SimpleQA, HotpotQA, and iAgentBench, but the remaining gap is largest on the benchmark whose questions were constructed to require cross-theme synthesis. Multi-step self-reflection helps when the remaining difficulty is extraction or path-following, but it can regress when the remaining difficulty is integration; the same iteration loop produces opposite effects across models on the same retrieval slice. These findings are visible because the benchmark records retrieval slices and supports failure decomposition.

Taken together, Chapters 3, 4, and 5 show that evaluating ISAs requires more than static accuracy on public QA benchmarks. It requires contamination-aware protocols, auditable evidence structures, and benchmark designs that distinguish evidence access from evidence integration. Chapter 6 synthesizes what the three chapters collectively show about ISA evaluation: which capabilities have now been put under test, which remain out of reach, and what a more comprehensive evaluation framework would need to add to the regime developed here.

Field	Description
`id`	Unique identifier for the instance.
`query`	Seed query (interest-driven topic string).
`time_window`	Time interval used for topic sourcing and retrieval.
`question`	Final one-sentence question.
`answer`	Short objective answer (entity, date, or short phrase).
`intent_pattern`	Information-seeking intent label: `explainer`, `connection`, `trigger`, `consequence`, `stake`.
`retrieved_sources`	URLs and metadata for retrieved documents in $\mathcal{D}(q)$.
`text_units`	IDs of chunks/sentences used as evidence units.
`story_graph_stats`	Graph statistics for $G(q)$ (number of nodes, number of edges).
`communities`	Community IDs with role labels (CORE, BRIDGE, SATELLITE) and influence scores.
`community_cards`	Per-community summary and grounded findings (with IDs).
`connectors`	Connector relations (with IDs and evidence pointers).
`packet`	Selected bundle B and the packet $P(B)$ used for generation.
`supporting_findings`	Finding IDs supporting the QA (≥ 1 per required community).
`supporting_connectors`	Connector IDs required for the QA (≥ 1).
`judge_decisions`	Outputs from the three-judge panel, including per-criterion flags and rationales.

Table 5.1: Core fields released per iAGENTBENCH instance. The schema captures the seed and the retrieved evidence, the intermediate story-graph artifacts, the packet used for generation, and the full annotation trail produced by the LLM-as-a-Judge panel.

Chapter 6

A DESIGN FRAMEWORK FOR EVALUATING INFORMATION-SEEKING AGENTS

Chapters 3–5 developed the dissertation's empirical arc: static public QA benchmarks are vulnerable to contamination, structured dynamic generation can make evaluation more auditable, and open-web sensemaking requires benchmarks that separate evidence access from evidence integration. This chapter synthesizes those results into a design framework for ISA evaluation. The goal is not to introduce another benchmark, but to make explicit what the preceding studies collectively show about how ISA benchmarks should be designed and interpreted.

6.1 Research Questions Revisited

The research questions stated in Section 1.3 frame the dissertation's evaluation problem from three complementary angles. RQ1 asks whether an ISA fulfills a user's informational need. RQ2 asks whether benchmarks can prevent task completion through memorized content or unintended tool use. RQ3 asks how evaluation should handle live, evolving information. Chapters 3–5 address these questions together by showing where static evaluation fails, how dynamic generation can improve measurement validity, and what additional challenges arise when evidence is drawn from the open web.

RQ1: assessing fulfillment of a user's informational need. A user-centered assessment cannot rely on a single end-to-end accuracy number. Three requirements follow from the preceding studies. First, the benchmark must encode the structure of the information need. Questions whose answers depend on integrating multiple themes, with explicit grounding in connector relations and supporting findings, expose synthesis behavior that single-passage

benchmarks suppress (Chapter 5). Second, the question distribution should reflect recurring information-seeking intents, such as *explainer*, *connection*, *trigger*, *consequence*, and *stake*, rather than only the affordances of an underlying knowledge source (Chapter 5). Third, the benchmark should release intermediate artifacts, including retrieved sources, story-graph statistics, community cards, connectors, supporting evidence, and judge decisions, so that aggregate accuracy can be decomposed into retrieval, theme recognition, integration, and answer formation.

The answer to RQ1 is therefore not that any single benchmark can certify fulfillment of a user's information need. A credible assessment requires three properties together: question construction tied to user intents, evidence requirements that force synthesis, and artifacts that support failure decomposition. iAgentBench instantiates these properties for the open-web setting, while Dynamic-KGQA provides the controlled counterpart in which structural grounding can be inspected directly (Chapter 4). The two settings exercise complementary slices of user-need fulfillment: structured reasoning over clean evidence on one side, and open-web sensemaking under realistic noise on the other.

RQ2: preventing task completion through memorized content or unintended tool use. Pretraining and retrieval-time contamination occur at meaningful rates on widely cited public QA benchmarks, and the rates reported in Chapter 3 should be read as conservative lower bounds. The two channels are distinct. Pretraining contamination can involve paraphrase, mirrors, and partial overlap that span-based detectors may miss. Retrieval-time contamination operates during evaluation and depends on what the retriever surfaces in response to the benchmark question. A defense that addresses only one channel leaves the other open.

The constructive response is twofold. Structural anchoring of items to a generation policy, rather than to a fixed public test set, narrows the pretraining channel because there is no canonical surface form to memorize (Chapter 4). Recording the retrieval slice as part of the benchmark instance narrows the retrieval-time channel because contamination checks can

be applied to the same evidence that the system consumed (Chapter 5). The combination of structural anchoring, dynamic regeneration, and recorded retrieval slices addresses both channels more directly than single-channel defenses such as paraphrasing or domain blocking.

The answer to RQ2 is that contamination resistance is possible, but it should be understood as a design goal rather than a single-property guarantee of a benchmark. It requires accounting for both pretraining exposure and retrieval-time leakage, because each channel can make benchmark content available to the system in a different way. Static public test sets, even when refreshed periodically, remain fragile because they re-enter the public web after release and can become part of the same exposure surface that motivated the refresh. This does not make static evaluation unusable, but it limits the strength of the claims that can be drawn from it without additional contamination audits or dynamic construction mechanisms.

RQ3: reasoning over live, evolving information. Live information access addresses two requirements that fixed test sets cannot satisfy: temporal grounding and longitudinal comparability. Temporal grounding means that an answer must be interpreted relative to what was available at a particular time. Longitudinal comparability means that refreshed evaluations should remain comparable across reporting periods. Chapter 4 shows that distributional consistency across regenerated runs can be tested in a structured setting using both significance and effect-size measures. Chapter 5 extends the same discipline to open-web evidence by anchoring seed selection to time-indexed attention signals and recording the retrieved evidence slice as part of each benchmark instance.

The answer to RQ3 is that freshness and comparability can be compatible under a generative benchmark design. The benchmark should be defined not only by a fixed set of items, but by the policy that produces those items. When the construction policy is stable and the temporal anchor is recorded, the benchmark can be re-instantiated over new time windows while still supporting comparison across runs. This dissertation does not claim to test every dimension of live information, but it shows how to avoid the false choice between a stale fixed benchmark and an incomparable stream of new items.

What the three answers add up to. Together, the three studies provide a progression from diagnosis to benchmark design. The contamination audit identifies a measurement validity threat in static public QA evaluation. Dynamic-KGQA develops a controlled dynamic response in which generation, grounding, and distributional consistency can be audited. iAgentBench extends this response to the open-web regime where ISAs are increasingly used and where evidence access must be separated from evidence integration. Across these settings, the dissertation identifies a set of requirements for contemporary ISA benchmarks: intent-anchored question distributions, evidence requirements that force synthesis, structural anchoring of items, time-indexed seed selection, recorded retrieval slices, distributional checks across runs, and intermediate artifacts that support audit and failure decomposition.

6.2 Core Evaluation Considerations

The empirical chapters point to four recurring considerations in ISA evaluation. These are not independent axes; each shapes the others, and overlooking any one of them can lead to misleading benchmark claims.

Static accuracy versus evaluation validity. A static accuracy number is a measurement, but it does not explain what was measured. The contamination audit makes this distinction visible. When pretraining and retrieval-time channels expose benchmark content, the reported score may reflect exposure as much as capability (Chapter 3). The validity question therefore comes before the accuracy question: a benchmark that does not account for contamination produces scores whose evidentiary value is uncertain. Chapter 4 provides an empirical signature of this issue. The same models that perform strongly on older Freebase-based QA benchmarks score substantially lower on Dynamic-KGQA, which is consistent with the interpretation that older benchmark scores partly reflect exposure while freshly generated items provide a cleaner measurement target.

Contamination resistance versus benchmark realism. Contamination resistance and realism are both necessary, but they pull on different parts of the design problem. A KG-grounded benchmark can be made highly contamination-resistant by regenerating items from a controlled graph, but its realism is bounded by the fact that the evidence has already been curated. An open-web benchmark can be more realistic by using traffic-driven topics and query-conditioned retrieval slices, but its contamination surface is larger because the evidence comes from the same web that supplies pretraining data and live retrieval pools. The two empirical benchmarks occupy different points in this trade-off and lead to the same design principle: contamination resistance must be built into both the construction policy and the audit infrastructure.

Structured grounding versus open-web evidence. Reasoning over a curated subgraph and reasoning over a query-conditioned web slice are different evaluation problems. In a structured setting, retrieval can be reduced to graph traversal and grounding can be checked by membership in $\mathcal{G}'$. In an open-web setting, retrieval, theme recognition, cross-source integration, and answer formation are all part of the task. Chapter 5 makes this difference visible: retrieval closes much of the gap on benchmarks dominated by access and extraction, but leaves a larger residual gap when the remaining difficulty is cross-theme synthesis. A complete evaluation portfolio for ISAs therefore needs both structured and open-web regimes, since performance in one cannot stand in for performance in the other.

Evidence access versus evidence integration. Retrieval-augmented generation often improves performance when the main bottleneck is access to evidence. However, once relevant evidence is available, the remaining difficulty may shift to integration. Chapter 5 shows that multi-step self-reflection can help when the task is extraction or path-following, but can regress when the task requires integrating themes across sources. This distinction motivates a central claim of the dissertation: benchmarks that report only end-to-end accuracy collapse retrieval, integration, and answer formation into a single signal. Releasing intermediate

artifacts and reporting per-stage effects are therefore not optional additions; they are what allow a benchmark to explain the score it reports.

Taken together, these distinctions define the shape of ISA evaluation. Validity precedes accuracy. Contamination resistance and realism are co-required rather than interchangeable. Structured and open-web settings test complementary capabilities. Evidence access and evidence integration are separable failure modes that benchmarks should allow evaluators to diagnose.

6.3 A Complexity-Aware View of Benchmark Design

The preceding distinctions can be organized as a design space. The relevant axes come from the information environment in which an ISA operates, not only from the model architecture being evaluated. Four axes recur across the empirical chapters and the broader information-seeking literature.

Temporality: static versus dynamic. Information needs vary in how strongly their answers depend on the moment of retrieval. A query about a foundational scientific result is largely time-invariant; a query about an unfolding event is time-dependent. Static benchmarks treat all items as if they were time-invariant. Dynamic benchmarks, in the sense developed in Chapters 4 and 5, treat the construction policy as part of the benchmark and allow items to be regenerated for a new time window. Both regimes are useful, but only the dynamic regime can measure behavior under evolving information conditions.

Evidence representation: structured versus open-web. Evidence varies in how curated it is when the system uses it. A clean knowledge graph is highly structured: relations are typed, entities are named, facts have provenance, and the graph boundary is enumerable. A query-conditioned web corpus is less controlled: relations are implicit, entities appear as mentions rather than canonical nodes, sources overlap imperfectly, and the evidence boundary is determined by the retrieved slice. These representations exercise different capabilities, and

a benchmark that mixes them without distinction risks attributing success to the wrong capability.

Task structure: single-source retrieval versus cross-source synthesis. Question difficulty is not only about the number of documents involved. One dimension is whether the answer can be found in a single source or requires multiple sources. A second, more demanding dimension is whether the answer depends on a relation between themes that the system must recover from the evidence. Multi-hop QA often addresses the first dimension. Cross-theme sensemaking addresses the second. The empirical patterns in Chapter 5 show that these regimes respond differently to retrieval and iteration, which makes the distinction important for benchmark design.

Information state: stable versus evolving. The underlying information may itself be stable or changing during the period of evaluation. Stable information can support fixed benchmark releases that age slowly. Evolving information requires either periodic re-collection or a generative construction policy that produces fresh items under unchanged rules. Distributional consistency across runs makes the latter viable in structured settings (Chapter 4), while time-indexed seed selection and recorded retrieval slices extend the same idea to open-web settings (Chapter 5).

These axes interact. A static, structured, single-source, stable regime corresponds to many classical QA benchmarks and is precisely the regime in which public benchmark exposure becomes a dominant validity threat. A dynamic, structured regime corresponds to Dynamic-KGQA and isolates reasoning over clean evidence under contamination-resistant conditions. A dynamic, open-web, cross-source regime corresponds to iAgentBench and tests capabilities associated with deployed information-seeking behavior. Other positions in the space are also useful: fixed open-web snapshots can support controlled retrieval studies, and dynamic structured benchmarks can isolate reasoning under temporal drift.

This design-space view is not just a taxonomy. It makes clear that benchmark claims are

conditional on the information regime being tested. A system's performance in one regime should not be treated as evidence of capability in another without additional measurement. The dissertation's three empirical studies occupy complementary positions in this space, and a broader benchmark portfolio can extend this coverage.

6.4 Design Principles for Future ISA Benchmarks

The empirical studies suggest a set of design principles that recur across regimes. Stating them as principles makes them transferable beyond the particular benchmarks introduced in this dissertation.

User-need fidelity. The question distribution should be anchored to informational needs that users actually express, not only to the affordances of an evidence source. Questions generated from a KG schema, a trivia list, or conveniently extractable spans may be useful for narrow capability studies, but they can drift away from realistic information-seeking behavior. A more defensible approach is to enumerate recurring intent patterns, generate questions under those patterns, and record the intended pattern per instance so that performance can be analyzed by need type.

Contamination awareness. Contamination must be modeled along both pretraining and retrieval-time channels. A construction that only paraphrases static items may reduce some pretraining matches but does not prevent answer-bearing pages from being retrieved at test time. A construction that only restricts retrieval does not address what the model may have already absorbed during training. The benchmark should narrow the contamination surface through its construction policy and release the artifacts needed to audit that surface during use.

Evidence traceability. A benchmark instance should include the artifacts needed to inspect the answer. For a KG-grounded benchmark, this includes the local subgraph $\mathcal{G}'$,

the answer path π, and the judge decisions that admitted the item. For an open-web benchmark, this includes retrieved sources, story-graph statistics, community cards, connector relations, generation packets, and judge decisions. The principle is the same across regimes: a benchmark instance is not only a question and an answer; it is a question, an answer, and the evidence trail that makes evaluation auditable.

Dynamic freshness. Benchmarks that target evolving information needs should refresh their items rather than freeze them indefinitely. Refreshing is not enough on its own. The construction policy must be reproducible, regenerated items must remain comparable to prior runs, and the temporal anchor of each run must be recorded. Freshness without comparability prevents longitudinal study; comparability without freshness reintroduces staleness and exposure.

Comparability across runs. Longitudinal comparability requires explicit evidence. In structured settings, distributional consistency can be reported with significance tests and effect-size measures over relevant axes such as topic distribution. In open-web settings, comparability depends on holding the construction policy fixed and recording the retrieval slice so that the evidence consumed by the model is reproducible. Aggregate accuracy alone is not sufficient; the conditions under which that accuracy was produced must also be documented.

Failure decomposition. Headline accuracy collapses retrieval, theme recognition, integration, and answer formation into a single signal. ISA benchmarks should support the decomposition of these stages so that failures can be attributed to specific operating regimes. Releasing intermediate artifacts makes this possible, and reporting per-stage effects such as Δ_{RAG} and Δ_{Refl} makes the decomposition visible to readers. A benchmark that cannot support this analysis behaves like a scoring instrument, not a diagnostic evaluation framework.

Realistic information environments. The evaluation environment should resemble the environment in which the system will be used. For open-web ISAs, this includes traffic-driven topic selection, query-conditioned evidence slices, time-indexed retrieval, and questions that reflect realistic user-intent patterns rather than benchmark-authoring artifacts. Static, curated, or trivia-style benchmarks remain useful for narrow capability studies, but they should not be treated as proxies for deployed information-seeking behavior without additional evidence.

The principles are connected. User-need fidelity constrains the question distribution; contamination awareness constrains the construction policy; traceability constrains the release schema; freshness and comparability constrain run-to-run discipline; failure decomposition constrains the reporting protocol; and realism constrains the information environment. Applied together, the principles close failure modes that any one principle alone would leave open.

6.5 Scope of the Design Framework

The design framework developed in this chapter is meant to organize the measurement problem, not to close every open question in ISA evaluation. It focuses on single-turn information needs, textual evidence, benchmark construction, contamination-aware evaluation, and the separation of evidence access from evidence integration. These are central dimensions of ISA evaluation, but they do not exhaust the space.

Several capabilities remain outside the empirical scope of this dissertation. The benchmarks developed in Chapters 4 and 5 do not test multi-turn clarification, persistent personalization, multimodal evidence use, embodied interaction, adversarial robustness, calibration under ambiguity, or cost-aware long-horizon planning. Chapter 7 returns to these directions as branches of the broader ISA evaluation agenda.

This boundary does not change the central claim of the design framework. The same evaluation discipline should carry over to those settings: benchmark instances should expose the evidence used by the system, record the temporal and environmental conditions of

evaluation, distinguish access from integration, and support failure decomposition. Extending the framework therefore means adding new artifacts and task regimes, not abandoning the principles developed here.

Chapter 7

CONCLUSION

This dissertation began from a broad shift in information access: generative and agentic systems are no longer limited to returning ranked lists or isolated answers. They increasingly search, retrieve, reason, summarize, compare, and act on behalf of users. This shift changes what evaluation must measure. For ISAs, the central question is not only whether the system produced a correct-looking answer, but whether it fulfilled an informational need through evidence that is appropriate, inspectable, timely, and sufficient.

The dissertation develops one path through this broader evaluation problem. It first diagnoses why static public QA benchmarks become fragile for web-enabled systems, then develops a controlled dynamic alternative over structured knowledge, and finally extends the evaluation setting to open-web sensemaking over query-conditioned evidence. These steps do not exhaust ISA evaluation. Rather, they establish a measurement foundation for one important branch of the larger agenda: evidence-grounded evaluation under contamination risk, dynamic information, and distributed web evidence.

7.1 Summary of Findings

The dissertation makes three main findings. First, static public QA benchmarks are fragile for evaluating modern web-enabled systems. Chapter 3 showed that contamination can enter through both pretraining exposure and retrieval-time leakage. These channels are distinct: pretraining contamination reflects what may have been absorbed before evaluation, while retrieval-time contamination reflects what a web-enabled system can surface during evaluation. The observed rates should be read as conservative lower bounds, but they are large enough to affect the interpretation of benchmark scores.

Second, dynamic benchmark generation can improve measurement validity when it is grounded in a stable construction policy. Chapter 4 showed this in a structured KGQA setting. DYNAMIC-KGQA generates fresh items from curated graph structure, pairs each item with an explicit grounding subgraph, verifies answer paths structurally, and tests whether generated variants remain distributionally comparable across runs. The key lesson is that the benchmark need not be a fixed public list of items; it can be a controlled procedure for producing auditable, refreshable, and comparable evaluation instances.

Third, open-web ISA evaluation requires measuring synthesis over retrieved evidence, not only access to that evidence. Chapter 5 showed this through iAGENTBENCH, where questions are grounded in time-indexed, query-conditioned web corpora and require cross-theme integration. The empirical results show that retrieval improves performance, but does not remove the difficulty of sensemaking. In particular, multi-step self-reflection helps when the remaining difficulty is extraction or path-following, but can be unstable when the task requires integrating themes across sources.

Together, these findings support the design framework developed in Chapter 6. Reliable ISA evaluation requires attention to validity before accuracy, contamination resistance alongside realism, structured and open-web evidence regimes, and the distinction between evidence access and evidence integration. The broader result is a way to reason about benchmark design under different information conditions, rather than a single benchmark intended to cover all ISA behavior.

The three empirical studies in this dissertation are best understood as a coherent branch of a larger ISA evaluation agenda. They address the branch concerned with evidence-grounded question answering and sensemaking under dynamic information conditions. This branch is important because QA-style interaction is a common interface through which users express informational needs, and because many deployed ISAs combine parametric models with retrieval, web search, and tool use. If evaluation in this setting is contaminated, stale, or unable to distinguish access from integration, then the field can easily overestimate the capabilities of systems that appear strong on public benchmarks.

At the same time, ISA evaluation is broader than the branch studied here. Other branches involve multi-turn interaction, ranging from bounded clarification exchanges to long-running learning-oriented conversations; personalization and memory, where the agent may use prior context, preferences, or conversation history; navigational and transactional information seeking, where the system must interact with websites, forms, and interfaces; multimodal information seeking, where users may provide images, screenshots, tables, or documents as part of the information need; and socially contested information, where the agent must represent uncertainty, disagreement, credibility, and possible harm.

These branches are natural extensions of the agenda developed here. The contribution of this dissertation is to establish measurement principles that future work can carry into those settings. Future benchmarks should make the evidence environment explicit, record the conditions under which evidence was accessed, separate access from integration, and release artifacts that allow failures to be traced. In a multi-turn setting, those artifacts may include dialogue state and clarification decisions. In a navigational setting, they may include interface states and action traces. In a multimodal setting, they may include modality-specific evidence pointers. In a socially contested setting, they may include source-credibility judgments, stance distributions, or disagreement structures. The specific artifacts will differ, but the evaluation discipline remains the same.

This is how the dissertation contributes to the broader goal of evaluating and improving ISAs. Rather than treating ISA evaluation as a single solved problem, it develops a central measurement thread: auditing whether existing measurements are valid, replacing fragile static items with controlled dynamic construction where appropriate, and evaluating open-web systems on their ability to synthesize evidence rather than merely retrieve it.

7.2 Implications for Evaluation Practice

For information retrieval, the unit of evaluation should expand beyond a query and a ranked list. In ISA settings, the relevant object is closer to a tuple of user need, retrieved evidence, system reasoning or synthesis, and final response. Recording the retrieved slice as part

of the benchmark instance makes retrieval auditable rather than treating it as an opaque preprocessing step.

For RAG evaluation, headline accuracy is insufficient. A system may fail because it did not retrieve the right evidence, because it retrieved the evidence but failed to use it, or because it produced an unsupported synthesis. These are different failures and should not be collapsed into one score. Benchmarks should therefore report the conditions under which evidence was supplied, record intermediate artifacts, and support analysis of access, integration, and answer formation separately.

Benchmark leakage also has a user-facing implication, even though users usually cannot identify it directly. A user cannot typically tell whether a strong answer reflects robust reasoning, appropriate retrieval, or prior exposure to the evaluation instance or closely related content. This opacity can distort trust. If a system performs well on contaminated instances, that success may create the appearance of a capability that does not generalize to other non-contaminated cases. Users may then carry confidence from the system's earlier fluent and correct-looking responses into later information needs, where the same system may still answer fluently despite weaker grounding. The concern is therefore not only that leakage inflates benchmark scores, but that it can obscure brittle behavior that becomes consequential when users rely on the system outside the contaminated setting.

For benchmark design, the dissertation supports a shift from fixed datasets to construction policies. A fixed public dataset is easy to distribute and compare against, but it becomes increasingly exposed over time. A construction policy can generate fresh instances while preserving comparability, provided that the policy is stable, the temporal anchor is recorded, and distributional checks are reported. This does not make static benchmarks obsolete, but it changes the strength of the claims they can support on their own.

For agent evaluation, more agentic behavior is not automatically better. Search, tool use, reflection, and multi-step planning are useful when the information need requires them, but each also introduces opportunities for retrieval errors, source ambiguity, and unsupported synthesis. Evaluation should therefore ask not only whether an agent used tools or took more

steps, but whether those steps improved grounded information-need fulfillment.

7.3 Open Directions

Several directions follow directly from the broader agenda. The first is interactive evaluation. Many real information needs are not fully specified at the first turn. Some multi-turn cases can be treated as bounded extensions of the framework developed here: a benchmark instance could include an initial user need, expected clarifying questions, user responses, evidence retrieved at each turn, and a final answer judged against the resolved need. This would allow the framework to evaluate follow-up questions and short dialogue trajectories while preserving the same emphasis on evidence-grounded information-need fulfillment. However, not all multi-turn interaction fits this model. Conversations aimed at learning a new subject, developing understanding over days, or iteratively refining an open-ended goal require evaluation frameworks that track user state, pedagogical progression, and long-horizon outcomes. These settings are less amenable to on-demand benchmark generation and may require longitudinal protocols, simulated users, or user studies.

The second is evaluation of persistent and personalized agents. The framework in this dissertation largely assumes a cold-start setting: the agent is evaluated on an information need without relying on prior knowledge of the user's preferences, history, or long-term goals. This assumption supports scalable and comparable evaluation, but it differs from increasingly common deployed systems that learn from previous conversations or maintain persistent user preferences. Once personalization becomes part of the evaluated capability, the benchmark must specify not only the task and evidence environment, but also the user model available to the agent. This makes evaluation more complex: the same response may be appropriate for one user and inappropriate for another, and failures may arise from stale memory, over-personalization, privacy risks, or incorrect assumptions. Evaluating these systems may therefore require synthetic user profiles, longitudinal interaction traces, or user studies, which complicates scalable on-demand evaluation.

The third is evaluation over richer information environments. The framework developed

in this dissertation is primarily text-based: instances are generated from textual sources, and evidence artifacts are represented as text, graph structure, or web documents. This is an important limitation because contemporary users increasingly provide images and screenshots as part of the information need itself. A user may ask an ISA to interpret a chart, explain a screenshot, compare a product photo with a specification, or combine visual context with web evidence. Extending the framework to such cases would require multimodal evidence artifacts, including image regions, screen states, OCR outputs when applicable, and links between visual observations and generated claims. The evaluation question remains aligned with this dissertation's framework, but the evidence object becomes richer than text alone. More broadly, deployed ISAs increasingly operate over tables, dashboards, videos, webpages, and interactive systems, where evaluation also requires records of environment states, actions, and consequences.

The fourth is robustness, calibration, and abstention. Open-web evidence can be misleading, stale, adversarial, or incomplete. A reliable ISA should recognize when the available evidence does not support a confident answer, when sources disagree, and when the appropriate response is to abstain or ask for clarification. Future benchmarks should therefore include controlled adversarial, ambiguous, and unanswerable cases.

Finally, ISA evaluation should become more cost-aware. Search depth, tool calls, agent steps, latency, and token budget all shape whether an approach is usable in practice. Reporting accuracy without these costs can make systems appear comparable when they are not. A more complete evaluation would report performance as a function of budget, not only as a single end-to-end score.

7.4 Final Remarks

Information-seeking agents are increasingly evaluated in the same public information environment that trains them, supplies their retrieval results, and changes the answers they are expected to provide. This breaks the assumption that a static benchmark score cleanly reflects capability. The dissertation shows that this problem is not solved by surface-level fixes such

as paraphrasing benchmark items or blocking a few domains. It requires evaluation designs that are contamination-aware, dynamically refreshable, evidence-grounded, and auditable.

The broader lesson is that ISA benchmarks should be designed around the information conditions in which the systems operate. When the task requires structured reasoning, the benchmark should expose the structure. When the task requires open-web sensemaking, the benchmark should record the retrieved evidence and test integration across sources. When the information changes over time, the benchmark should record the temporal context and support regeneration under a stable policy. Evaluating ISAs well therefore means matching benchmark design to the properties of the information need, the evidence environment, and the system capabilities being claimed.

The studies in this dissertation provide one path toward that goal. They diagnose why static QA evaluation becomes fragile, develop a controlled dynamic alternative over structured knowledge, extend the approach to open-web sensemaking, and synthesize these results into a design framework for future ISA evaluation. The resulting message is simple but important: benchmarks for information-seeking agents should not only ask whether a system produced the right answer, but whether it reached that answer through evidence that is appropriate, inspectable, timely, and sufficient for the user's informational need.

BIBLIOGRAPHY

[1] Searxng documentation. https://docs.searxng.org/, 2025. Accessed: 2025-10-07.

[2] Tamer Abuelsaad, Deepak Akkil, Prasenjit Dey, Ashish Jagmohan, Aditya Vempaty, and Ravi Kokku. Agent-e: From autonomous web navigation to foundational design principles in agentic systems. *arXiv preprint arXiv:2407.13032*, 2024.

[3] Perplexity AI. Getting started with perplexity api, 2025. URL https://www.perplexity.ai/.

[4] James Allan, Courtney Wade, and Alvaro Bolivar. Retrieval and novelty detection at the sentence level. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 314–321, Toronto Canada, July 2003. ACM. ISBN 978-1-58113-646-3. doi: 10.1145/860435.860493. URL https://dl.acm.org/doi/10.1145/860435.860493.

[5] Omar Alonso, Michael Gertz, and Ricardo Baeza-Yates. On the value of temporal information in information retrieval. *ACM SIGIR Forum*, 41(2):35–41, December 2007. ISSN 0163-5840. doi: 10.1145/1328964.1328968. URL https://dl.acm.org/doi/10.1145/1328964.1328968.

[6] Anthropic. *The Claude 3 Model Family: Opus, Sonnet, Haiku*, 2024. URL https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf. Accessed: 2025-02-16.

[7] Anthropic. Introducing web search on the anthropic api, May 2025. URL https://www.anthropic.com/news/web-search-api. Accessed: 2025-05-20.

[8] Marcia J Bates. The design of browsing and berrypicking techniques for the online search interface. *Online review*, 13(5):407–424, 1989.

[9] Nicholas J Belkin et al. Anomalous states of knowledge as a basis for information retrieval. *Canadian journal of information science*, 5(1):133–143, 1980.

[10] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544, 2013.

[11] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[12] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008.

[13] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[14] Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*, 2015.

[15] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving

from trillions of tokens, February 2022. Comment: Fix incorrect reported numbers in Table 14.

[16] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

[17] Andrei Broder. A taxonomy of web search. In *ACM Sigir forum*, volume 36, pages 3–10. ACM New York, NY, USA, 2002.

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

[19] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners, July 2020. Comment: 40+32 pages.

[20] Chris Buckley and Ellen M Voorhees. Retrieval evaluation with incomplete information. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 25–32, 2004.

[21] Hongru Cai, Yongqi Li, Wenjie Wang, Fengbin Zhu, Xiaoyu Shen, Wenjie Li, and Tat-Seng Chua. Large language models empowered personalized web agents. In *Proceedings of the ACM on Web Conference 2025*, pages 198–215, 2025.

[22] Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017.

[23] James P. Callan. Passage-Level Evidence in Document Retrieval. In Bruce W. Croft and C. J. Van Rijsbergen, editors, *SIGIR '94*, pages 302–310. Springer London, London, 1994. ISBN 978-3-540-19889-5 978-1-4471-2099-5. doi: 10.1007/978-1-4471-2099-5_31. URL http://link.springer.com/10.1007/978-1-4471-2099-5_31.

[24] Ricardo Campos, Gaël Dias, Alípio M Jorge, and Adam Jatowt. Survey of temporal information retrieval and related applications. *ACM Computing Surveys (CSUR)*, 47 (2):1–41, 2014.

[25] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying Memorization Across Neural Language Models, March 2023.

[26] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 675–684, Hyderabad India, March 2011. ACM. ISBN 978-1-4503-0632-4. doi: 10.1145/1963405.1963500. URL https://dl.acm.org/doi/10.1145/1963405.1963500.

[27] Lluis Castrejon, Thomas Mensink, Howard Zhou, Vittorio Ferrari, Andre Araujo, and Jasper Uijlings. Hammr: Hierarchical multimodal react agents for generic vqa. *arXiv preprint arXiv:2404.05465*, 2024.

[28] Jiangjie Chen, Xintao Wang, Rui Xu, Siyu Yuan, Yikai Zhang, Wei Shi, Jian Xie, Shuang Li, Ruihan Yang, Tinghui Zhu, Aili Chen, Nianqi Li, Lida Chen, Caiyu Hu, Siye Wu, Scott Ren, Ziquan Fu, and Yanghua Xiao. From Persona to Personalization: A Survey on Role-Playing Language Agents, October 2024. URL http://arxiv.org/abs/2404.18231. arXiv:2404.18231 [cs].

[29] Lingjiao Chen, Matei Zaharia, and James Zou. How is chatgpt's behavior changing over time? *arXiv preprint arXiv:2307.09009*, 2023.

[30] Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. Dense x retrieval: What retrieval granularity should we use? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15159–15177, 2024.

[31] Yangyi Chen, Binxuan Huang, Yifan Gao, Zhengyang Wang, Jingfeng Yang, and Heng Ji. Scaling laws for predicting downstream performance in llms. *arXiv preprint arXiv:2410.08527*, 2024.

[32] Junghoo Cho and Hector Garcia-Molina. Synchronizing a database to improve freshness. *ACM SIGMOD Record*, 29(2):117–128, June 2000. ISSN 0163-5808. doi: 10.1145/335191.335391. URL `https://dl.acm.org/doi/10.1145/335191.335391`.

[33] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. PaLM: Scaling Language Modeling with Pathways, October 2022.

[34] E. F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, June 1970. ISSN 0001-0782, 1557-7317. doi: 10.1145/362384.362685. URL https://dl.acm.org/doi/10.1145/362384.362685.

[35] Cohere. *Command R: Details and Application*, 2024. URL https://docs.cohere.com/v2/docs/command-r. Accessed: 2025-02-16.

[36] Jiaxi Cui, Munan Ning, Zongjian Li, Bohua Chen, Yang Yan, Hao Li, Bin Ling, Yonghong Tian, and Li Yuan. Chatlaw: A multi-agent collaborative legal assistant with knowledge graph enhanced mixture-of-experts large language model. *arXiv preprint arXiv:2306.16092*, 2023.

[37] Preetam Prabhu Srikar Dammu, Hayoung Jung, Anjali Singh, Monojit Choudhury, and Tanushree Mitra. " they are uncultured": Unveiling covert harms and social threats in llm generated conversations. *arXiv preprint arXiv:2405.05378*, 2024.

[38] Preetam Prabhu Srikar Dammu, Omar Alonso, and Barbara Poblete. A Shopping Agent for Addressing Subjective Product Needs. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pages 1032–1035, Hannover Germany, March 2025. ACM. ISBN 9798400713293. doi: 10.1145/3701551.3704124. URL https://dl.acm.org/doi/10.1145/3701551.3704124.

[39] Preetam Prabhu Srikar Dammu, Himanshu Naidu, and Chirag Shah. Dynamic-kgqa: A scalable framework for generating adaptive question answering datasets. *arXiv preprint arXiv:2503.05049*, 2025.

[40] Preetam Prabhu Srikar Dammu, Himanshu Naidu, and Chirag Shah. Dynamic-KGQA: A Scalable Framework for Generating Adaptive Question Answering Datasets, March 2025. URL http://arxiv.org/abs/2503.05049. arXiv:2503.05049 [cs].

[41] Preetam Prabhu Srikar Dammu, Arnav Palkhiwala, Tanya Roosta, and Chirag Shah.

iagentbench: Benchmarking sensemaking capabilities of information-seeking agents on high-traffic topics. *arXiv preprint arXiv:2603.04656*, 2026.

[42] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[43] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36, 2024.

[44] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[45] Wentao Ding, Jinmao Li, Liangchuan Luo, and Yuzhong Qu. Enhancing complex question answering over knowledge graphs through evidence pattern retrieval. In *Proceedings of the ACM on Web Conference 2024*, pages 2106–2115, 2024.

[46] Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. Documenting Large Webtext Corpora: A Case Study on the Colossal Clean Crawled Corpus, September 2021. Comment: EMNLP 2021 accepted paper camera ready version.

[47] Anlei Dong, Yi Chang, Zhaohui Zheng, Gilad Mishne, Jing Bai, Ruiqiang Zhang, Karolina Buchner, Ciya Liao, and Fernando Diaz. Towards recency ranking in web search. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 11–20, New York, NY, USA, February 2010. Association for Computing Machinery. ISBN 978-1-60558-889-6. doi: 10.1145/1718487.1718490. URL `https://dl.acm.org/doi/10.1145/1718487.1718490`.

[48] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson.

From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

[49] Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, et al. Magentic-one: A generalist multi-agent system for solving complex tasks. *arXiv preprint arXiv:2411.04468*, 2024.

[50] Norbert Fuhr and Kai Großjohann. XIRQL: a query language for information retrieval in XML documents. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 172–180, New Orleans Louisiana USA, September 2001. ACM. ISBN 978-1-58113-331-8. doi: 10.1145/383952.383985. URL https://dl.acm.org/doi/10.1145/383952.383985.

[51] Difei Gao, Lei Ji, Luowei Zhou, Kevin Qinghong Lin, Joya Chen, Zihan Fan, and Mike Zheng Shou. Assistgpt: A general multi-modal assistant that can plan, execute, inspect, and learn. *arXiv preprint arXiv:2306.08640*, 2023.

[52] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-Augmented Generation for Large Language Models: A Survey, March 2024. Comment: Ongoing Work.

[53] Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120, 2023.

[54] Shahriar Golchin and Mihai Surdeanu. Data contamination quiz: A tool to detect and estimate contamination in large language models. *arXiv preprint arXiv:2311.06233*, 2023.

[55] Peiyuan Gong, Jiamian Li, and Jiaxin Mao. Cosearchagent: A lightweight collaborative search agent with large language models. In *Proceedings of the 47th International*

ACM SIGIR Conference on Research and Development in Information Retrieval, pages 2729–2733, 2024.

[56] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[57] Google. Try deep research and our new experimental model in gemini, your ai assistant, January 2025. URL `https://blog.google/products/gemini/google-gemini-deep-research/`. Accessed: 2025-05-20.

[58] Yu Gu, Sue Kase, Michelle Vanni, Brian Sadler, Percy Liang, Xifeng Yan, and Yu Su. Beyond iid: three levels of generalization for question answering on knowledge bases. In *Proceedings of the Web Conference 2021*, pages 3477–3488, 2021.

[59] Jian Guan, Jesse Dodge, David Wadden, Minlie Huang, and Hao Peng. Language models hallucinate, but may excel at fact verification. *arXiv preprint arXiv:2310.14564*, 2023.

[60] Mandy Guo, Zihang Dai, Denny Vrandečić, and Rami Al-Rfou. Wiki-40b: Multilingual language model dataset. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2440–2452, 2020.

[61] Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. A real-world webagent with planning, long context understanding, and program synthesis. *arXiv preprint arXiv:2307.12856*, 2023.

[62] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval Augmented Language Model Pre-Training. In *Proceedings of the 37th International Conference on Machine Learning*, pages 3929–3938. PMLR, November 2020.

[63] Ziwen Han, Meher Mankikar, Julian Michael, and Zifan Wang. Search-Time Data Contamination, August 2025.

[64] Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. Webvoyager: Building an end-to-end web agent with large multimodal models. *arXiv preprint arXiv:2401.13919*, 2024.

[65] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[66] Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. Women also snowboard: Overcoming bias in captioning models. In *Proceedings of the European conference on computer vision (ECCV)*, pages 771–787, 2018.

[67] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[68] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, Seattle WA USA, August 2004. ACM. ISBN 978-1-58113-888-7. doi: 10.1145/1014052.1014073. URL https://dl.acm.org/doi/10.1145/1014052.1014073.

[69] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*, 2023.

[70] Amazon Artificial General Intelligence. The amazon nova family of models: Technical report and model card. *Amazon Technical Reports*, 2024. URL https://www.amazon

.science/publications/the-amazon-nova-family-of-models-technical-report-and-model-card.

[71] Gautier Izacard and Edouard Grave. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering, February 2021.

[72] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: Few-shot Learning with Retrieval Augmented Language Models, November 2022.

[73] Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. Query expansion by prompting large language models. *arXiv preprint arXiv:2305.03653*, 2023.

[74] Jinhao Jiang, Kun Zhou, Wayne Xin Zhao, and Ji-Rong Wen. Unikgqa: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph. *arXiv preprint arXiv:2212.00959*, 2022.

[75] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-Scale Similarity Search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, July 2021. ISSN 2332-7790. doi: 10.1109/TBDATA.2019.2921572. URL https://ieeexplore.ieee.org/document/8733051.

[76] Rosie Jones and Fernando Diaz. Temporal profiles of queries. *ACM Transactions on Information Systems*, 25(3):14, July 2007. ISSN 1046-8188, 1558-2868. doi: 10.1145/1247715.1247720. URL https://dl.acm.org/doi/10.1145/1247715.1247720.

[77] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension, May 2017. Comment: Added references, fixed typos, minor baseline update.

[78] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781, 2020.

[79] Jungo Kasai, Keisuke Sakaguchi, Yoichi Takahashi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A. Smith, Yejin Choi, and Kentaro Inui. RealTime QA: What's the Answer Right Now?, February 2024. Comment: RealTime QA Website: https://realtimeqa.github.io/.

[80] Kirandeep Kaur, Preetam Prabhu Srikar Dammu, Hideo Joho, and Chirag Shah. Beyond static evaluation: Rethinking the assessment of personalized agent adaptability in information retrieval. In *Proceedings of the 2025 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pages 396–405, 2025.

[81] Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, et al. Dynabench: Rethinking benchmarking in nlp. *arXiv preprint arXiv:2104.14337*, 2021.

[82] Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. *arXiv preprint arXiv:2401.13649*, 2024.

[83] Ivica Kostric, Krisztian Balog, and Filip Radlinski. Generating usage-related questions for preference elicitation in conversational recommender systems. *ACM Transactions on Recommender Systems*, 2(2):1–24, 2024.

[84] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[85] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7: 452–466, 2019. doi: 10.1162/tacl_a_00276.

[86] Ariel N. Lee, Cole J. Hunter, and Nataniel Ruiz. Platypus: Quick, Cheap, and Powerful Refinement of LLMs, March 2024. Comment: Workshop on Instruction Tuning and Instruction Following at NeurIPS 2023.

[87] Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. End-to-end Neural Coreference Resolution, December 2017. URL `http://arxiv.org/abs/1707.07045`. arXiv:1707.07045 [cs].

[88] Nicholas Lester, Alistair Moffat, and Justin Zobel. Efficient online index construction for text databases. *ACM Transactions on Database Systems*, 33(3):1–33, August 2008. ISSN 0362-5915, 1557-4644. doi: 10.1145/1386118.1386125. URL `https://dl.acm.org/doi/10.1145/1386118.1386125`.

[89] Mike Lewis. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[90] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc., 2020.

[91] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al.

Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

[92] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html`.

[93] Yucheng Li. An open source data contamination report for llama series models. *arXiv preprint arXiv:2310.17589*, 2023.

[94] Yucheng Li, Frank Guerin, and Chenghua Lin. An Open Source Data Contamination Report for Large Language Models, January 2024.

[95] Yaobo Liang, Chenfei Wu, Ting Song, Wenshan Wu, Yan Xia, Yu Liu, Yang Ou, Shuai Lu, Lei Ji, Shaoguang Mao, et al. Taskmatrix. ai: Completing tasks by connecting foundation models with millions of apis. *Intelligent Computing*, 3:0063, 2024.

[96] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[97] Jimmy Lin, Miles Efron, Yulu Wang, and Garrick Sherman. Overview of the TREC-2014 Microblog Track.

[98] Adam Liška, Tomáš Kočiský, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, Cyprien de Masson d'Autume, Tim Scholtes, Manzil Zaheer, Susannah Young, Ellen Gilsenan-McMahon, Sophia Austin, Phil Blunsom, and Angeliki Lazaridou. StreamingQA: A Benchmark for Adaptation to New Knowledge over Time in Question Answering Models, May 2022.

[99] Ken Ziyu Liu, Christopher A. Choquette-Choo, Matthew Jagielski, Peter Kairouz, Sanmi Koyejo, Percy Liang, and Nicolas Papernot. Language Models May Verbatim Complete Text They Were Not Explicitly Trained On, March 2025. Comment: Main text: 9 pages, 7 figures, 1 table. Appendix: 29 pages, 20 tables, 15 figures.

[100] Na Liu, Liangyu Chen, Xiaoyu Tian, Wei Zou, Kaijiang Chen, and Ming Cui. From LLM to Conversational Agent: A Memory Enhanced Architecture with Fine-Tuning of Large Language Models, January 2024. URL `http://arxiv.org/abs/2401.02777`. arXiv:2401.02777 [cs].

[101] Xiao Liu, Hanyu Lai, Hao Yu, Yifan Xu, Aohan Zeng, Zhengxiao Du, Peng Zhang, Yuxiao Dong, and Jie Tang. 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, 2023.

[102] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688*, 2023.

[103] Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models. *Advances in Neural Information Processing Systems*, 36: 43447–43478, 2023.

[104] Xing Han Lù, Zdeněk Kasner, and Siva Reddy. Weblinx: Real-world website navigation with multi-turn dialogue. *arXiv preprint arXiv:2402.05930*, 2024.

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

[106] Zhiyi Ma, Kawin Ethayarajh, Tristan Thrush, Somya Jain, Ledell Wu, Robin Jia, Christopher Potts, Adina Williams, and Douwe Kiela. Dynaboard: An evaluation-as-a-service platform for holistic next-generation benchmarking. *Advances in Neural Information Processing Systems*, 34:10351–10367, 2021.

[107] Kurt Mehlhorn. A faster approximation algorithm for the steiner problem in graphs. *Information Processing Letters*, 27(3):125–128, 1988.

[108] Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, et al. Teaching language models to support answers with verified quotes. *arXiv preprint arXiv:2203.11147*, 2022.

[109] Miriam J Metzger, Andrew J Flanagin, and Ryan B Medders. Social and heuristic approaches to credibility evaluation online. *Journal of communication*, 60(3):413–439, 2010.

[110] Bhaskar Mitra, Nick Craswell, et al. An introduction to neural information retrieval. *Foundations and Trends® in Information Retrieval*, 13(1):1–126, 2018.

[111] John X. Morris, Chawin Sitawarin, Chuan Guo, Narine Kokhlikyan, G. Edward Suh, Alexander M. Rush, Kamalika Chaudhuri, and Saeed Mahloujifar. How much do language models memorize?, May 2025.

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

[113] Thomas Neumann and Gerhard Weikum. RDF-3X: a RISC-style engine for RDF. *Proceedings of the VLDB Endowment*, 1(1):647–659, August 2008. ISSN 2150-8097. doi:

10.14778/1453856.1453927. URL https://dl.acm.org/doi/10.14778/1453856.1453927.

[114] OpenAI. GPT-4 Technical Report, March 2024. Comment: 100 pages; updated authors list; fixed author names and added citation.

[115] OpenAI. Introducing operator, January 2025. URL https://openai.com/index/introducing-operator/. Accessed: 2025-05-20.

[116] Iadh Ounis, Craig Macdonald, Jimmy Lin, and Ian Soboroff. Overview of the TREC-2011 Microblog Track.

[117] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford infolab, 1999.

[118] Bo Pang and Lillian Lee. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts, September 2004. URL http://arxiv.org/abs/cs/0409058. arXiv:cs/0409058.

[119] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[120] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22, 2023.

[121] Thomas Pellissier Tanon, Denny Vrandečić, Sebastian Schaffert, Thomas Steiner, and Lydia Pintscher. From freebase to wikidata: The great migration. In *Proceedings of the 25th international conference on world wide web*, pages 1419–1428, 2016.

[122] Cheng Qian, Bingxiang He, Zhong Zhuang, Jia Deng, Yujia Qin, Xin Cong, Yankai Lin, Zhong Zhang, Zhiyuan Liu, and Maosong Sun. Tell me more! towards implicit user intention understanding of language model driven agents. *arXiv preprint arXiv:2402.09205*, 2024.

[123] Yujia Qin, Zihan Cai, Dian Jin, Lan Yan, Shihao Liang, Kunlun Zhu, Yankai Lin, Xu Han, Ning Ding, Huadong Wang, et al. Webcpm: Interactive web search for chinese long-form question answering. *arXiv preprint arXiv:2305.06849*, 2023.

[124] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs, October 2023. URL `http://arxiv.org/abs/2307.16789`. arXiv:2307.16789 [cs].

[125] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners.

[126] Filip Radlinski and Nick Craswell. A theoretical framework for conversational search. In *Proceedings of the 2017 conference on conference human information interaction and retrieval*, pages 117–126, 2017.

[127] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140): 1–67, 2020.

[128] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text, October 2016. Comment: To appear in Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP).

[129] Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. Measuring attribution in natural language generation models. *Computational Linguistics*, 49(4):777–840, 2023.

[130] Mathieu Ravaut, Bosheng Ding, Fangkai Jiao, Hailin Chen, Xingxuan Li, Ruochen Zhao, Chengwei Qin, Caiming Xiong, and Shafiq Joty. A Comprehensive Survey of Contamination Detection Methods in Large Language Models, July 2025. Comment: Accepted by TMLR in July 2025. 18 pages, 1 figure, 3 tables.

[131] Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, et al. Androidworld: A dynamic benchmarking environment for autonomous agents. *arXiv preprint arXiv:2405.14573*, 2024.

[132] Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, Daniel Toyama, Robert Berry, Divya Tyamagundlu, Timothy Lillicrap, and Oriana Riva. AndroidWorld: A Dynamic Benchmarking Environment for Autonomous Agents, April 2025.

[133] Philipp Schoenegger, Indre Tuminauskaite, Peter S Park, and Philip E Tetlock. Wisdom of the silicon crowd: Llm ensemble prediction capabilities match human crowd accuracy. *arXiv preprint arXiv:2402.19379*, 2024.

[134] SearXNG contributors. Searxng. URL `https://github.com/searxng/searxng`. Source code repository (AGPL-3.0).

[135] Chirag Shah and Emily M Bender. Situating search. In *Proceedings of the 2022 Conference on Human Information Interaction and Retrieval*, pages 221–232, 2022.

[136] Chirag Shah, Hideo Joho, Kirandeep Kaur, and Preetam Prabhu Srikar Dammu. Dynamic Evaluation Framework for Personalized and Trustworthy Agents: A Multi-Session Approach to Preference Adaptability, March 2025. URL `http://arxiv.org/abs/2504.06277`. arXiv:2504.06277 [cs].

[137] Xiang Shi, Jiawei Liu, Yinpeng Liu, Qikai Cheng, and Wei Lu. Know where to go: Make llm a relevant, responsible, and trustworthy searcher. *arXiv preprint arXiv:2310.12443*, 2023.

[138] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

[139] Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

[140] Arnold WM Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on pattern analysis and machine intelligence*, 22(12):1349–1380, 2000.

[141] Fabian M Suchanek, Mehwish Alam, Thomas Bonald, Lihu Chen, Pierre-Henri Paris, and Jules Soria. Yago 4.5: A large and clean knowledge base with a rich taxonomy. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 131–140, 2024.

[142] Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M Ni, Heung-Yeung Shum, and Jian Guo. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. *arXiv preprint arXiv:2307.07697*, 2023.

[143] Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=nnVO1PvbTv`.

[144] Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11888–11898, 2023.

[145] Alon Talmor and Jonathan Berant. The web as a knowledge-base for answering complex questions. *arXiv preprint arXiv:1803.06643*, 2018.

[146] Mistral AI Team. *Mistral Small 3*, 2025. URL `https://mistral.ai/en/news/mistral-small-3`. Accessed: 2025-02-16.

[147] The GDELT Project. The gdelt project. `https://www.gdeltproject.org/`, n.d. Accessed: 2026-02-02.

[148] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*, 2018.

[149] Kushal Tirumala, Aram H. Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. Memorization Without Overfitting: Analyzing the Training Dynamics of Large Language Models, November 2022.

[150] Vincent Traag, Ludo Waltman, and Nees Jan van Eck. From Louvain to Leiden: Guaranteeing well-connected communities. *Scientific Reports*, 9(1):5233, March 2019. ISSN 2045-2322. doi: 10.1038/s41598-019-41695-z.

[151] Ricardo Usbeck, Xi Yan, Aleksandr Perevalov, Longquan Jiang, Julius Schulz, Angelie Kraft, Cedric Möller, Junbo Huang, Jan Reineke, Axel-Cyrille Ngonga Ngomo, et al.

Qald-10–the 10th challenge on question answering over linked data: Shifting from dbpedia to wikidata as a kg for kgqa. *Semantic Web*, 15(6):2193–2207, 2024.

[152] Ellen M Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 315–323, 1998.

[153] Ellen M Voorhees et al. The trec-8 question answering track report. In *Trec*, volume 99, pages 77–82, 1999.

[154] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

[155] Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, and Thang Luong. FreshLLMs: Refreshing Large Language Models with Search Engine Augmentation, November 2023. Comment: Preprint, 26 pages, 10 figures, 5 tables; Added FreshEval.

[156] Joseph Michael Vukov, Tera Lynn Joseph, Gina Lebkuecher, Michelle Ramirez, and Michael B Burns. The ouroboros threat. *The American Journal of Bioethics*, 23(10): 58–60, 2023.

[157] Junke Wang, Dongdong Chen, Chong Luo, Xiyang Dai, Lu Yuan, Zuxuan Wu, and Yu-Gang Jiang. Chatvideo: A tracklet-centric multimodal and versatile video understanding system. *arXiv preprint arXiv:2304.14407*, 2023.

[158] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024.

[159] Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. In *European Conference on Computer Vision*, pages 58–76. Springer, 2024.

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

[161] Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models, November 2024. Comment: Blog post: https://openai.com/index/introducing-simpleqa/.

[162] Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating Expressions of Opinions and Emotions in Language. *Language Resources and Evaluation*, 39(2-3):165–210, May 2005. ISSN 1574-020X, 1572-8412. doi: 10.1007/s10579-005-7880-9. URL `http://link.springer.com/10.1007/s10579-005-7880-9`.

[163] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

[164] Junlin Xie, Zhihong Chen, Ruifei Zhang, Xiang Wan, and Guanbin Li. Large multimodal agents: A survey. *arXiv preprint arXiv:2402.15116*, 2024.

[165] Binfeng Xu, Zhiyuan Peng, Bowen Lei, Subhabrata Mukherjee, Yuchen Liu, and Dongkuan Xu. ReWOO: Decoupling Reasoning from Observations for Efficient Augmented Language Models, May 2023. URL `http://arxiv.org/abs/2305.18323`. arXiv:2305.18323 [cs].

[166] Yao Xu, Shizhu He, Jiabei Chen, Zihao Wang, Yangqiu Song, Hanghang Tong, Guang Liu, Kang Liu, and Jun Zhao. Generate-on-graph: Treat llm as both agent and kg in incomplete knowledge graph question answering. *arXiv preprint arXiv:2404.14741*, 2024.

[167] Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. Hallucination is inevitable: An innate limitation of large language models, 2024. URL https://arxiv.org/abs/2401.11817.

[168] Shuo Yang, Wei-Lin Chiang, Lianmin Zheng, Joseph E Gonzalez, and Ion Stoica. Rethinking benchmark and contamination for language models with rephrased samples. *arXiv preprint arXiv:2311.04850*, 2023.

[169] Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*, 2023.

[170] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

[171] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.

[172] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering, September 2018. Comment: EMNLP 2018 long paper. The first three authors contribute equally. Data, code, and blog posts available at https://hotpotqa.github.io/.

[173] Zongxin Yang, Guikun Chen, Xiaodi Li, Wenguan Wang, and Yi Yang. Doraemongpt:

Toward understanding dynamic scenes with large language models (exemplified as a video agent). *arXiv preprint arXiv:2401.08392*, 2024.

[174] Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757, 2022.

[175] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing Reasoning and Acting in Language Models, March 2023. URL `http://arxiv.org/abs/2210.03629`. arXiv:2210.03629 [cs].

[176] Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. *tau*-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.

[177] Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206, 2016.

[178] Ori Yoran, Samuel Joseph Amouyal, Chaitanya Malaviya, Ben Bogin, Ofir Press, and Jonathan Berant. Assistantbench: Can web agents solve realistic and time-consuming tasks? *arXiv preprint arXiv:2407.15711*, 2024.

[179] Jing Zhang, Xiaokang Zhang, Jifan Yu, Jian Tang, Jie Tang, Cuiping Li, and Hong Chen. Subgraph retrieval enhanced model for multi-hop knowledge base question answering. *arXiv preprint arXiv:2202.13296*, 2022.

[180] Yu Zhang, Shutong Qiao, Jiaqi Zhang, Tzu-Heng Lin, Chen Gao, and Yong Li. A survey of large language model empowered agents for recommendation and search: Towards next-generation information retrieval. *arXiv preprint arXiv:2503.05659*, 2025.

[181] Zhehao Zhang, Jiaao Chen, and Diyi Yang. Darg: Dynamic evaluation of large language models via adaptive reasoning graph. *arXiv preprint arXiv:2406.17271*, 2024.

[182] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. *arXiv preprint arXiv:1707.09457*, 2017.

[183] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 2024.

[184] Zijie Zhong, Hanwen Liu, Xiaoya Cui, Xiaofan Zhang, and Zengchang Qin. Mix-of-Granularity: Optimize the Chunking Granularity for Retrieval-Augmented Generation, January 2025. URL `http://arxiv.org/abs/2406.00456`. arXiv:2406.00456 [cs].

[185] Kun Zhou, Yutao Zhu, Zhipeng Chen, Wentong Chen, Wayne Xin Zhao, Xu Chen, Yankai Lin, Ji-Rong Wen, and Jiawei Han. Don't make your llm an evaluation benchmark cheater. *arXiv preprint arXiv:2311.01964*, 2023.

[186] Ruiwen Zhou, Yingxuan Yang, Muning Wen, Ying Wen, Wenhao Wang, Chunling Xi, Guoqiang Xu, Yong Yu, and Weinan Zhang. Trad: Enhancing llm agents with step-wise thought retrieval and aligned decision. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–13, 2024.

[187] Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023.

[188] Jun-Peng Zhu, Peng Cai, Kai Xu, Li Li, Yishen Sun, Shuai Zhou, Haihuang Su, Liu Tang, and Qi Liu. Autotqa: Towards autonomous tabular question answering through

multi-agent large language models. *Proceedings of the VLDB Endowment*, 17(12): 3920–3933, 2024.

[189] Kaijie Zhu, Jiaao Chen, Jindong Wang, Neil Zhenqiang Gong, Diyi Yang, and Xing Xie. Dyval: Dynamic evaluation of large language models for reasoning tasks. In *International Conference on Learning Representations*, 2023. URL https://api.semanticscholar.org/CorpusID:263310319.

[190] Kaijie Zhu, Jindong Wang, Qinlin Zhao, Ruochen Xu, and Xing Xie. Dyval 2: Dynamic evaluation of large language models by meta probing agents. *ArXiv*, abs/2402.14865, 2024. URL https://api.semanticscholar.org/CorpusID:267897463.

[191] Kaijie Zhu, Jindong Wang, Qinlin Zhao, Ruochen Xu, and Xing Xie. Dynamic Evaluation of Large Language Models by Meta Probing Agents, June 2024. Comment: International Conference on Machine Learning (ICML) 2024.

[192] Yongshuo Zong, Tingyang Yu, Ruchika Chavhan, Bingchen Zhao, and Timothy Hospedales. Fool your (vision and) language model with embarrassingly simple permutations. *arXiv preprint arXiv:2310.01651*, 2023.