FOLLOW THE ENTITIES: A CORPUS MAP FOR AGENTIC SEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

Answering questions and completing tasks over large document collections often requires connecting evidence spread across multiple documents, such as a project's approval recorded in one, its requirements in another, and its latest status in a third. Recent LLM agents approach this by iteratively searching the full corpus rather than reading only a fixed set of top-ranked documents. However, when the corpus is exposed only as a flat collection of files, a relevant document gives no indication of how it relates to others, so the agent must rediscover these relationships for every query, often missing complementary evidence while simultaneously consuming substantial additional tokens. To address this, we introduce CORPUSMAP, a navigation layer that organizes the corpus around its recurring entities, which are identifiable from the documents themselves and can link a single document to many others across sources. Specifically, CORPUSMAP represents each recurring entity as an Entity Page that aggregates information about it and links to every document that refers to it, forming a graph between entities and documents that the agent can traverse to gather otherwise disconnected evidence. Moreover, since CORPUSMAP is constructed offline by resolving mentions of the same entity across documents, its links are shared across queries rather than rediscovered repeatedly at inference time. Using 7 different models with 3 benchmark datasets, we show that CORPUSMAP improves both evidence discovery and answer quality over raw-corpus agentic search while using fewer tokens on average, and further outperforms 4 alternative navigation layers, suggesting that entities serve as effective anchors for navigating large document collections.

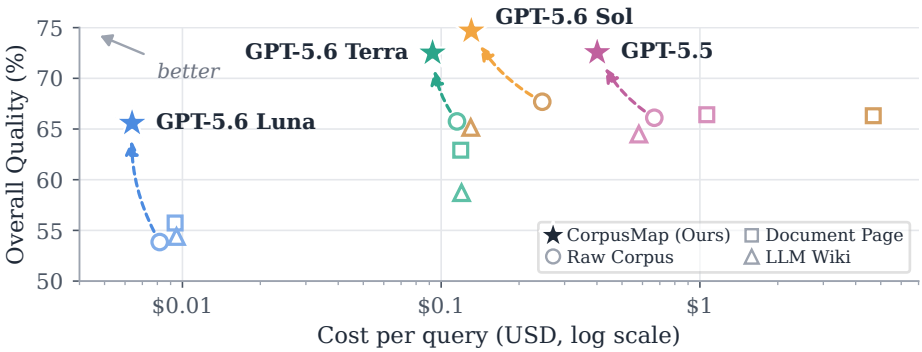

Figure 1: Overall Quality (Table 1) versus estimated answer-generation cost for CORPUSMAP and the three strongest baselines, averaged equally over all benchmarks.

1 INTRODUCTION

Large language models (LLMs) have shown impressive capabilities as agents (OpenAI, 2026a;b; DeepSeek-AI, 2026; Microsoft AI, 2026), and have been widely adopted to answer questions and complete tasks over large document collections (Wang et al., 2024; Huang et al., 2025; Liang et al., 2025), where the necessary evidence is often distributed across multiple documents. For example, determining whether a project is ready to launch may require combining its latest status from a

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

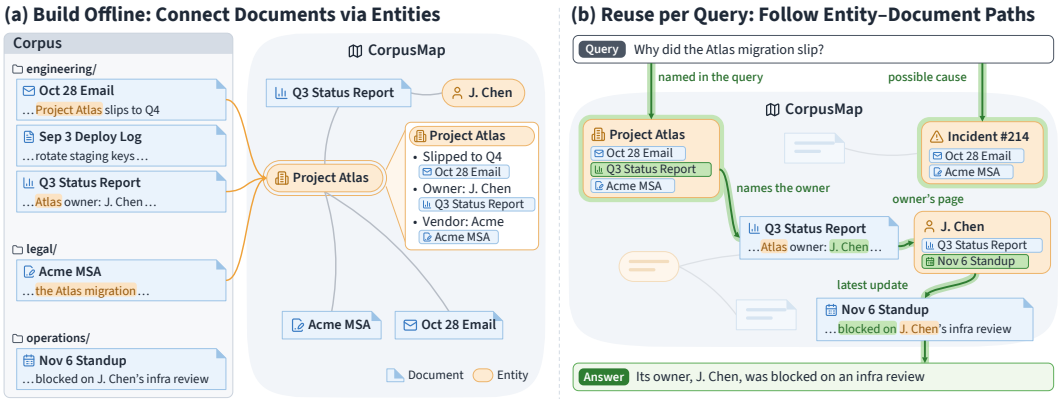

Figure 2: Overview of CORPUSMAP. **(a)** Offline, recurring mentions of the same subject across documents in different folders are resolved into shared entities, which connect the documents into a reusable entity–document map. **(b)** For each query, the agent follows entity–document links in the same map from the entities relevant to the query to gather evidence for the answer. Green marks the walk; each entity card lists its source documents, highlighting the one opened next.

project tracker, an approval recorded over email, requirements from operational documents, and a final decision recorded in meeting notes. Although each source captures part of the answer, none is sufficient in isolation, and the relationships among them may not be stated explicitly in any individual document. Moreover, in practice, these sources are buried among hundreds of thousands of other documents spread across different applications (e.g., issue trackers, shared drives, and chat channels) (Sun et al., 2026b; Choubey et al., 2025), so that the challenge lies not only in utilizing them, but also in locating and retrieving them.

To locate supporting evidence, retrieval-augmented generation (RAG) typically retrieves documents relevant to a user query or instruction (Lewis et al., 2020; Robertson et al., 1994; Karpukhin et al., 2020; Fan et al., 2024) by selecting the top-k documents before starting the model's reasoning or response. For tasks that require follow-up searches for additional documents, agentic search retrieves evidence over multiple steps (Yao et al., 2023; Liang et al., 2025) and uses tool calls to search the full corpus directly (Subramanian et al., 2026; Li et al., 2026), so that it is no longer limited to what an initial retrieval step returns. Nevertheless, full-corpus access does not by itself make the corpus easy to navigate, since it remains a flat collection of documents, without representing their relationships.

As a result, the agent is left to infer these relationships on its own by searching, reading, and reasoning over multiple sources sequentially, limiting both efficacy and efficiency. For instance, a document's relevance to a query may be opaque or require corpus-specific knowledge (e.g. meeting notes that record the launch decision under the project's internal codename), so that searching with the query alone can miss it, despite being accessible. Moreover, because each query is treated independently, the agent must spend a substantial number of tokens to re-read and re-discover these relationships each time. Yet, while different queries require different subsets of these relationships, the relationships themselves (e.g. which documents refer to the same project) remain stable across queries, and could thus be identified in advance and reused. We therefore frame this challenge as a *corpus navigation* problem, which calls for a persistent *navigation layer* that exposes reusable cross-document structure while preserving access to the full corpus.

This raises a central design question: which relationships should such a layer expose? Given the challenges above, they should link documents that the query text alone may not reach easily and be identifiable in advance so that they can be reused across queries. In our work, we leverage the fact that documents are naturally generated around common entities, such as people, projects, or products, and design a navigation layer that makes these entities and their related documents explicit. Since entities and their relations are identifiable by the corpus alone (and not any queries), the navigation layer can be built entirely offline, so that query-time inference remains efficient.

To this end, we introduce CORPUSMAP, a novel *entity-centric* navigation layer that makes these links explicit and reusable. CORPUSMAP is constructed offline by identifying entity mentions within each document, resolving those that refer to the same entity across sources, and representing each

resolved entity as an *Entity Page* that gathers what different documents state about it, attributing each fact to its source, and links to every document that refers to it. Importantly, Entity Pages add a layer over the corpus rather than replacing it, so that the original documents remain available to the model. As illustrated in Figure 2, from any document it reads, the agent can follow the mentioned entities to complementary evidence instead of searching for it again, which can surface otherwise overlooked evidence while reducing the documents to inspect.

We validate CORPUSMAP across multiple LLMs on complex questions spanning multiple documents from three benchmarks, EnterpriseRAG-Bench (Sun et al., 2026b), WixQA (Cohen et al., 2025), and HERB (Choubey et al., 2025), and find that it consistently improves both evidence discovery and answer quality over raw-corpus agentic search and alternative navigation layers (e.g., LLM Wiki (Karpathy, 2026) and Corpus2Skill (Sun et al., 2026a)). Specifically, compared with raw-corpus agentic search, CORPUSMAP improves overall quality by 6.4 to 11.7 points while using 34% to 57% fewer input tokens on average, as summarized in Figure 1. Moreover, we show that CORPUSMAP can be constructed even without the use of LLMs and updated incrementally as the corpus grows and evolves, making it practical and efficient for real deployment environments. Together, these results suggest that improving how a corpus is organized, rather than only how agents search it, is a promising direction for agents operating over large and growing document collections.

2 RELATED WORK

Retrieval-Augmented Generation Retrieval-augmented generation (RAG) grounds language-model outputs in external knowledge by retrieving the documents or passages most relevant to a query, typically according to lexical or embedding similarity, and conditioning generation on the retrieved context (Robertson et al., 1994; Karpukhin et al., 2020; Lewis et al., 2020; Fan et al., 2024). While simple RAG approaches score each reference independently, thereby discarding any inter-document information that might exist, graph-based approaches such as GraphRAG (Edge et al., 2024) and HippoRAG (Gutierrez et al., 2024; 2025) organize the corpus into a graph of extracted entities and their relations, leveraging their shared context to retrieve content connected across documents. Nevertheless, such structure is used within the retriever rather than exposed to the generator (the LLM), which typically still receives a fixed context selected in a single step (e.g., top-k passages or graph-derived summaries), so that the generator must answer from whatever that retrieval returns and cannot reach a relevant document the retrieval misses, even when the answer depends on it.

Agentic Search and Corpus Interfaces for Agents To move beyond single-step retrieval, iterative and agentic approaches retrieve over multiple steps (Trivedi et al., 2023; Jiang et al., 2023; Asai et al., 2024; Jeong et al., 2024), interleaving planning, search, source inspection, and tool-use throughout an LLM's reasoning trajectory (Yao et al., 2023; Li et al., 2025; Liang et al., 2025; Li et al., 2026; Salemi et al., 2026). Although these methods improve the query-time search policy, because the corpus itself remains a set of independent documents, any relationships inferred during one query response must be rediscovered for every question. To address this, recent work organizes the corpus into agent-facing structures. LLM-maintained wikis compile documents into cross-linked pages (Karpathy, 2026; Ming et al., 2026), but require the model to decide what becomes a page and how content is merged, decisions that can degrade as the corpus grows (Zhou et al., 2026). Hierarchical skill trees organize documents into topical branches that an agent traverses to reach source documents (Sun et al., 2026a), but assigning each document to only one or a few branches can separate evidence about the same subject across sources.

Entity Extraction, Linking, and Resolution Identifying entities in text has long been studied through named entity recognition (Sang & Meulder, 2003; Lample et al., 2016; Li et al., 2023), recently extended to open entity types by LLMs and lightweight generalist encoders (Zhou et al., 2024; Sainz et al., 2024; Zaratiana et al., 2024), and through entity linking, which grounds mentions in a reference knowledge base such as Wikipedia (Wu et al., 2020; De Cao et al., 2021; Sevgili et al., 2022). When no such knowledge base covers the entities of interest, cross-document coreference and entity resolution instead cluster the mentions that refer to the same entity across sources (Cybulska & Vossen, 2014; Barhom et al., 2019; Li et al., 2020; Papadakis et al., 2021; Cattan et al., 2021), with recent approaches ranging from LLM prompting (Narayan et al., 2022; Peeters & Bizer, 2023; Peeters et al., 2025; Fu et al., 2025) to lightweight zero-shot linkers (Stepanov et al., 2026). Our

work builds on this line of research, leveraging these capabilities to organize a corpus around its resolved cross-document entities as navigational anchors for LLM agents.

3 METHOD

In this section, we first formalize agentic question answering over large document collections, and then present CORPUSMAP, an entity-centric navigation layer that exposes reusable cross-document evidence paths to the agent, together with the offline protocol that constructs it.

3.1 PRELIMINARIES

Task Formulation Let $\mathcal{D}$ denote a corpus containing documents drawn from different sources, and let q denote a question whose answer requires combining evidence distributed across multiple documents. We write the agentic question-answering process as $(\hat{a}_q, \hat{\mathcal{D}}_q) = \texttt{Agent}(q, \mathcal{D})$, where $\hat{a}_q$ is the generated answer and $\hat{\mathcal{D}}_q \subseteq \mathcal{D}$ is the selected supporting set. We assess the resulting trajectory along three complementary axes: answer quality, requiring $\hat{a}_q$ to be correct and complete; retrieval quality, requiring $\hat{\mathcal{D}}_q$ to cover the documents the question actually depends on; and efficiency, favoring limited context consumption.

Raw-Corpus Agentic Search We first consider an agent operating directly over the raw corpus, which is exposed as a flat collection of documents. Given q, the agent uses standard shell commands (e.g., `find` and `grep`) to search over $\mathcal{D}$, reads promising documents, and updates the selected supporting set $\hat{\mathcal{D}}_q$ as evidence accumulates. However, although this interface gives access to the full corpus, it encodes no relations between documents, so once a relevant document is found, locating related evidence requires further search. Consequently, $\hat{\mathcal{D}}_q$ may omit documents that searching for q does not return, and the context consumed to work out how documents relate is spent again for each subsequent question, even when the same documents are involved.

3.2 CORPUSMAP: AN ENTITY-CENTRIC NAVIGATION LAYER

To address this limitation, we introduce CORPUSMAP, which makes relations between documents explicit through a map $\mathcal{G}$ of the corpus that is constructed offline from $\mathcal{D}$ and shared across questions, so that the agent operates at inference as $\texttt{Agent}(q, \mathcal{D}; \mathcal{G})$. We build $\mathcal{G}$ around the entities mentioned in documents (e.g., people, projects, or incidents), which can be identified in each document independently of any question, so that the map can be built in advance.

Map Representation Let $\mathcal{E}$ denote the *cross-document entities* that CORPUSMAP retains from $\mathcal{D}$, that is, those linked to more than one document, and for each $e \in \mathcal{E}$, let the document neighborhood $\mathcal{N}(e) \subseteq \mathcal{D}$ contain the documents in which a mention was resolved to e during construction (Section 3.3). We represent each retained entity as an entity node and each document as a document node, and write $\mathcal{L}$ for the links between these two node types, so that the map is the bipartite graph

$$\mathcal{G} = (\mathcal{E} \cup \mathcal{D}, \mathcal{L}), \qquad \mathcal{L} = \{(e, d) \mid e \in \mathcal{E},\ d \in \mathcal{N}(e)\}, \tag{1}$$

where each link $(e, d) \in \mathcal{L}$ connects an entity node to a document node. Since a document is linked to each retained entity it mentions, two documents that share an entity are connected through it, and a document can belong to several neighborhoods at once. Meanwhile, documents linked to no retained entity remain in $\mathcal{D}$ as isolated nodes, accessible through raw-corpus search.

Source-Grounded Entity Pages Each entity node $e \in \mathcal{E}$ is exposed to the agent as an *Entity Page* that consolidates what the documents in $\mathcal{N}(e)$ state about e: a brief overview, key facts each tagged with the document it comes from, the names under which e appears, and links to every document in $\mathcal{N}(e)$. Since these facts may come from documents in different sources, a single page can bring together complementary evidence that the raw corpus keeps apart.

3.3 CONSTRUCTING CORPUSMAP

We construct $\mathcal{G}$ offline in the four stages summarized in Algorithm 1.

Algorithm 1 CORPUSMAP construction protocol.

Require: Document corpus $\mathcal{D}$, entity-processing backend, page renderer
Ensure: Entity-centric map $\mathcal{G}$ with Entity Pages
1: $\mathcal{C} \leftarrow$ INDUCETYPECATALOG($\mathcal{D}$) ▷ propose from sampled documents, synthesize, verify, revise
2: $\{\mathcal{E}_d^{\text{local}}\}_{d\in\mathcal{D}} \leftarrow$ EXTRACTLOCALENTITIES($\mathcal{D}, \mathcal{C}$) ▷ identify, type, and group mentions per document
3: $\mathcal{R} \leftarrow \emptyset,\ \mathcal{L} \leftarrow \emptyset$ ▷ empty registry $\mathcal{R}$ and link set $\mathcal{L}$
4: **for** each document $d \in \mathcal{D}$ and each local entity $z \in \mathcal{E}_d^{\text{local}}$ **do**
5: $\delta \leftarrow$ RESOLVE($z, \mathcal{R}$) ▷ retrieve candidates from $\mathcal{R}$ and decide
6: **if** $\delta =$ `LINK`(e) **then**
7: UPDATEENTITY($\mathcal{R}, e, z$); $\mathcal{L} \leftarrow \mathcal{L} \cup \{(e, d)\}$
8: **else if** $\delta =$ `ADD` **then**
9: $e \leftarrow$ ADDENTITY($\mathcal{R}, z$); $\mathcal{L} \leftarrow \mathcal{L} \cup \{(e, d)\}$
10: **else**
11: RECORDUNRESOLVED(z, d)
12: **end if**
13: **end for**
14: $\mathcal{E} \leftarrow$ CROSSDOCUMENTENTITIES($\mathcal{R}, \mathcal{L}$) ▷ linked to at least two documents
15: $\mathcal{L} \leftarrow \{(e, d) \in \mathcal{L} \mid e \in \mathcal{E}\}$
16: **for** each entity $e \in \mathcal{E}$ **do**
17: $\mathcal{N}(e) \leftarrow \{d \in \mathcal{D} \mid (e, d) \in \mathcal{L}\}$
18: $p_e \leftarrow$ RENDERENTITYPAGE($e, \mathcal{N}(e)$) ▷ facts about e grounded in and linked to $\mathcal{N}(e)$
19: **end for**
20: $\mathcal{G} \leftarrow (\mathcal{E} \cup \mathcal{D}, \mathcal{L})$
21: **return** $\mathcal{G}$ with $\{p_e\}_{e\in\mathcal{E}}$

Cataloging (Line 1) The *entity types* worth extracting, such as products, incidents, or configuration flags, vary from one corpus to another and cannot be exhaustively specified in advance. We therefore induce a catalog $\mathcal{C}$ of these types from the corpus itself with an LLM, proposing a candidate catalog from each of several small sets of sampled documents, synthesizing these candidates into one, and verifying and revising the result; the resulting catalog, which specifies a name, definition, identity criteria, and observed examples for each type, is then fixed for the remaining stages.

Extraction (Line 2) Since the catalog specifies the kinds of entities in the map (e.g., *Project*) but not the instances present in the corpus (e.g., "Project Atlas"), we use it together with the surrounding document context to identify and type *entity mentions*, grouping those denoting the same subject within a document into a single document-local entity.

Resolution (Lines 3–13) We ground every extracted name in its source-text occurrence before resolving the local entities of each document, in turn, against a shared registry that starts out empty: for each of them, we retrieve plausible registry entries and weigh the current document against the candidate evidence to either `LINK` the observation to an existing entry, `ADD` a new entity, or leave it `UNRESOLVED`. Each `LINK` or `ADD` records a grounded entity–document link, whereas `UNRESOLVED` observations create none.

Rendering (Lines 14–21) We keep only the cross-document entities, that is, those linked to at least two documents, since only these provide reusable navigational paths, and render the neighborhood $\mathcal{N}(e)$ of each retained entity as its Entity Page. With the source documents and the links between them, these pages yield the map $\mathcal{G}$ of Equation (1). The Extraction, Resolution, and Rendering stages can be instantiated with an LLM or with off-the-shelf and deterministic alternatives.

3.4 NAVIGATING CORPUSMAP

We now describe how the agent accesses the map at inference. Specifically, each Entity Page is stored as a file alongside the raw documents and lists the file paths of its linked documents, so that the agent can read and search the map with the same tools as the raw corpus. Along with the question, the agent receives, as candidates, the file paths of the documents linked to the Entity Pages relevant to the question, and decides which of them to read. The agent can further search both Entity Pages and documents, following a link from a page by reading a listed file, or from a document by searching for the pages that list it (Figure 2).

4 EXPERIMENTAL SETUP

We now describe the benchmarks and evaluation, baselines, and implementation details.

Benchmarks and Evaluation To evaluate CORPUSMAP, we use three benchmarks that contain questions requiring evidence distributed across multiple documents: **EnterpriseRAG-Bench** (Sun et al., 2026b), **WixQA** (Cohen et al., 2025), and **HERB** (Choubey et al., 2025). Specifically, we use the 80 questions in the categories of EnterpriseRAG-Bench that consist entirely of multi-document questions, the 79 multi-document questions of WixQA, and the 238 content-based questions of HERB, which ask about information stated across multiple documents. For the corpora, we use fixed sets of 2,819 and 6,365 documents for EnterpriseRAG-Bench and HERB, respectively, both including all gold documents, and the full set of 6,221 articles for WixQA. Following the three axes in Section 3.1, we evaluate (1) **answer quality**: correctness, completeness, factuality, and content; (2) **retrieval quality**: document recall and context recall; and (3) **efficiency**: the input tokens accumulated over the full agent trajectory per question.

Baselines and Our Method We compare CORPUSMAP against RAW CORPUS and four baselines that organize the same corpus around different units, while the original documents remain accessible to the agent in every method. **RAW CORPUS** adds no navigation layer to the documents. **DOCUMENT PAGE** represents each document by an LLM-generated page of its key facts, **GROUP PAGE** consolidates the documents within each group defined by the corpus itself (e.g., its folders) into a single page, and **LLM WIKI** (Karpathy, 2026) lets the LLM freely write cross-linked pages over the corpus. **CORPUS2SKILL** (Sun et al., 2026a) organizes the corpus into a topical hierarchy of LLM-summarized document clusters. **CORPUSMAP (Ours)** organizes the corpus into an entity-centric map of Entity Pages linked to their source documents. We also report **GOLD DOCUMENTS (ORACLE)**, which provides only the gold documents to the model as reference for the model capability ceiling under idealized retrieval.

Implementation Details For the main results, we use four GPT models spanning a wide range of costs, GPT-5.5 (OpenAI, 2026a) and GPT-5.6 Luna, Terra, and Sol (OpenAI, 2026b), where the same LLM constructs the artifacts of each method and serves as the agent answering the questions. For the analyses beyond the main results, we mainly use the most and least expensive of them, GPT-5.5 and GPT-5.6 Luna. We additionally use DeepSeek-V4-Pro (DeepSeek-AI, 2026) and MAI-Thinking-1 (Microsoft AI, 2026), as well as the open-weight Qwen3.8-27B (Qwen Team, 2026). For LLM-judged metrics, we use GPT-5.6 Sol. Following Sun et al. (2026b), we use a terminal-based agent that navigates the corpus through shell commands, under their per-question execution budget. Please refer to Appendix A for more details.

5 EXPERIMENTAL RESULTS AND ANALYSES

We first examine the effectiveness of CORPUSMAP, and then analyze its practical aspects, with further analyses provided in Appendix B.

5.1 EFFECTIVENESS OF CORPUSMAP

Main Results Table 1 presents the main results, showing that CORPUSMAP consistently achieves the best answer and retrieval quality across all benchmarks and LLMs, with significant overall gains over every baseline (Table 6), at a lower average cost per query than raw-corpus agentic search (Figure 1). Notably, the four baselines that organize the corpus in other ways do not consistently improve over RAW CORPUS, indicating that simply adding a navigation layer does not guarantee improvement. Moreover, CORPUSMAP improves quality while reducing tokens, with the largest savings for GPT-5.5 and GPT-5.6 Sol, the two most expensive LLMs. Also, CORPUSMAP substantially narrows the gap to the non-comparable Oracle, reflecting its effectiveness in gathering cross-document evidence. Finally, CORPUSMAP also outperforms RAW CORPUS with fewer tokens on the remaining questions of EnterpriseRAG-Bench, most of which are grounded in a single document (Table 5), indicating that the map remains beneficial even when cross-document evidence is not required.

Table 1: Main results across EnterpriseRAG-Bench, WixQA, and HERB with GPT-5.5 and GPT-5.6 models, as means ± standard deviations over three runs. Overall reports dataset-balanced quality and geometric-mean input-token ratios to Raw Corpus within each LLM. Best and second-best effectiveness scores per LLM among retrieval methods are **bolded** and underlined.

		EnterpriseRAG-Bench				WixQA			HERB		Overall	
	Method	Doc. Rec. ↑	Correct. ↑	Complete. ↑	Tokens ↓	Ctx. Rec. ↑	Fact. ↑	Tokens ↓	Content ↑	Tokens ↓	Quality ↑	Rel. Tok. ↓
GPT-5.5	Raw Corpus	61.62 ± 1.66	62.08 ± 4.12	73.11 ± 0.98	206.5k	73.31 ± 0.75	67.51 ± 0.39	337.2k	62.31 ± 0.59	598.4k	66.11	1.00×
	Document Page	64.18 ± 1.55	57.50 ± 1.02	68.97 ± 1.83	175.6k	77.85 ± 1.18	66.98 ± 1.90	288.0k	63.26 ± 0.14	680.9k	66.41	0.94×
	Group Page	64.75 ± 0.96	48.75 ± 2.70	62.62 ± 2.58	423.0k	60.86 ± 3.68	62.24 ± 4.45	1,125.8k	62.97 ± 1.66	392.9k	61.07	1.65×
	LLM Wiki	63.15 ± 1.55	56.67 ± 2.36	67.74 ± 1.77	192.4k	73.73 ± 0.52	65.19 ± 0.26	140.9k	61.49 ± 0.76	387.0k	64.49	0.63×
	Corpus2Skill	49.39 ± 2.32	47.50 ± 2.70	57.29 ± 1.01	88.8k	65.93 ± 2.07	60.34 ± 0.39	76.0k	21.94 ± 0.79	180.9k	45.49	0.31×
	CorpusMap (Ours)	**76.17 ± 0.30**	**73.75 ± 1.77**	**79.88 ± 1.00**	88.1k	**82.70 ± 0.79**	**70.68 ± 0.83**	74.5k	**64.37 ± 0.89**	494.7k	**72.55**	0.43×
	Gold Documents (Oracle)	*100.00 ± 0.00*	*87.08 ± 2.12*	*81.01 ± 0.59*	*7.1k*	*85.97 ± 0.30*	*79.22 ± 0.30*	*2.1k*	*66.90 ± 0.84*	*11.4k*	*79.62*	*0.02×*
Luna	Raw Corpus	56.40 ± 0.79	60.42 ± 5.14	73.12 ± 0.96	80.7k	47.36 ± 1.66	59.39 ± 1.04	129.3k	44.86 ± 0.37	219.4k	53.85	1.00×
	Document Page	56.96 ± 2.11	50.42 ± 3.28	68.78 ± 2.40	160.0k	61.92 ± 0.91	61.18 ± 2.23	88.7k	46.91 ± 0.49	222.4k	55.73	1.11×
	Group Page	53.95 ± 1.82	42.08 ± 1.18	58.82 ± 1.30	347.6k	43.57 ± 0.79	55.59 ± 2.71	145.3k	41.47 ± 0.24	119.1k	47.55	1.38×
	LLM Wiki	54.54 ± 3.72	49.17 ± 2.12	65.90 ± 1.68	144.5k	58.12 ± 2.15	59.70 ± 2.07	117.3k	47.71 ± 0.55	213.3k	54.39	1.16×
	Corpus2Skill	35.73 ± 1.25	31.25 ± 1.77	49.90 ± 0.20	102.4k	56.33 ± 3.66	53.80 ± 3.05	56.2k	17.78 ± 0.55	145.7k	37.27	0.72×
	CorpusMap (Ours)	**75.11 ± 0.27**	**67.08 ± 4.25**	**77.42 ± 1.45**	60.1k	**76.90 ± 0.93**	**69.30 ± 0.26**	41.0k	**50.45 ± 0.27**	254.2k	**65.58**	0.65×
	Gold Documents (Oracle)	*100.00 ± 0.00*	*82.50 ± 4.68*	*76.90 ± 0.57*	*7.1k*	*85.76 ± 0.52*	*75.53 ± 0.60*	*2.1k*	*72.90 ± 0.66*	*11.4k*	*80.00*	*0.04×*
Terra	Raw Corpus	67.13 ± 1.13	64.58 ± 2.36	77.11 ± 0.81	70.9k	65.08 ± 1.42	62.45 ± 0.91	354.6k	63.88 ± 0.74	288.3k	65.75	1.00×
	Document Page	62.16 ± 3.32	56.67 ± 4.12	69.39 ± 3.41	191.7k	72.78 ± 2.37	64.77 ± 1.47	138.9k	57.22 ± 2.15	274.9k	62.91	1.00×
	Group Page	62.00 ± 1.69	38.33 ± 2.95	52.57 ± 1.10	355.3k	26.69 ± 6.88	43.14 ± 5.67	324.6k	45.62 ± 0.49	150.7k	43.84	1.34×
	LLM Wiki	58.77 ± 0.66	43.33 ± 2.57	64.58 ± 0.31	161.6k	66.03 ± 2.10	61.60 ± 3.03	129.5k	56.77 ± 1.41	390.7k	58.71	1.04×
	Corpus2Skill	45.84 ± 0.43	44.58 ± 2.57	55.97 ± 0.38	153.7k	67.30 ± 1.94	57.81 ± 1.94	72.4k	20.23 ± 0.43	185.8k	43.86	0.66×
	CorpusMap (Ours)	**73.08 ± 1.98**	**73.75 ± 1.77**	**79.83 ± 0.92**	76.2k	**80.91 ± 1.22**	**71.73 ± 0.98**	96.2k	**65.65 ± 0.70**	289.4k	**72.51**	0.66×
	Gold Documents (Oracle)	*100.00 ± 0.00*	*82.08 ± 0.59*	*81.17 ± 0.36*	*7.1k*	*85.55 ± 0.39*	*77.64 ± 0.98*	*2.1k*	*70.37 ± 0.83*	*11.4k*	*79.90*	*0.03×*
Sol	Raw Corpus	64.79 ± 2.43	67.08 ± 3.28	76.75 ± 1.85	55.5k	69.30 ± 1.34	68.35 ± 1.81	282.3k	64.69 ± 1.42	430.2k	67.69	1.00×
	Document Page	67.64 ± 0.94	67.50 ± 2.04	72.13 ± 1.52	55.2k	73.84 ± 1.33	70.78 ± 1.76	1,496.8k	57.53 ± 3.54	5,003.6k	66.31	3.94×
	Group Page	59.31 ± 2.44	43.75 ± 2.70	56.91 ± 3.42	188.5k	18.67 ± 0.26	39.77 ± 1.94	458.6k	11.14 ± 0.52	59.0k	31.23	0.91×
	LLM Wiki	60.70 ± 1.08	59.17 ± 4.12	69.83 ± 0.42	53.2k	70.46 ± 1.04	68.14 ± 0.54	68.7k	62.91 ± 1.74	208.5k	65.15	0.48×
	Corpus2Skill	45.10 ± 0.72	42.08 ± 1.18	53.78 ± 1.89	126.4k	64.66 ± 1.42	59.49 ± 1.61	99.1k	24.24 ± 0.70	159.0k	44.44	0.67×
	CorpusMap (Ours)	**71.46 ± 0.41**	**80.83 ± 0.59**	**81.80 ± 0.60**	56.2k	**83.33 ± 0.83**	**72.78 ± 1.18**	52.0k	**67.97 ± 1.08**	186.3k	**74.68**	0.43×
	Gold Documents (Oracle)	*100.00 ± 0.00*	*80.42 ± 1.18*	*80.85 ± 0.54*	*7.1k*	*85.97 ± 0.65*	*79.85 ± 0.30*	*2.1k*	*69.08 ± 0.77*	*11.4k*	*79.69*	*0.03×*

Table 2: Overall Quality with LLMs from other model families on EnterpriseRAG-Bench.

	DeepSeek		MAI	
Method	**Quality**	**Tokens**	**Quality**	**Tokens**
Raw Corpus	68.02	1,349.4k	34.06	129.6k
Document Page	43.18	4,047.0k	38.56	208.4k
Group Page	59.24	1,965.5k	32.81	421.7k
LLM Wiki	57.99	3,523.8k	35.57	109.8k
Corpus2Skill	49.72	628.7k	21.33	120.4k
CorpusMap (Ours)	**71.11**	1,043.3k	**45.07**	158.8k

Table 3: Retrieval-based approaches on EnterpriseRAG-Bench.

	Quality	
Method	**GPT-5.5**	**Luna**
BM25	63.66	60.78
Dense	65.05	58.84
HippoRAG	64.66	61.74
GraphRAG	47.95	43.46
Raw Corpus	65.60	63.31
CorpusMap (Ours)	**76.60**	**73.20**

Generalization to Other Model Families To examine whether the effectiveness of CorpusMap generalizes beyond the GPT family, we further evaluate it with DeepSeek-V4-Pro and MAI-Thinking-1, and report the results in Table 2. We find that CorpusMap again achieves the best Overall Quality with both LLMs, whereas the baselines that organize the corpus in other ways do not consistently improve over Raw Corpus, as observed with the GPT models. This indicates that the benefit of exposing cross-document connections through the entity-centric map is not tied to a particular model family, but carries over to LLMs of different architectures.

Comparison with Retrieval-Based Approaches We also compare against BM25 (Robertson et al., 1994), dense retrieval (OpenAI, 2024), HippoRAG (Gutierrez et al., 2024; 2025), and GraphRAG (Edge et al., 2024) under the retrieve-then-generate paradigm, where the LLM answers directly from a fixed amount of context retrieved for the question, without accessing the full corpus itself. As shown in Table 3, CorpusMap outperforms all of them with both LLMs, including the graph-based HippoRAG and GraphRAG. This suggests that answering from a retrieved context alone is limited by what the retrieval returns, whereas exposing cross-document connections to the agent lets it continue to gather the missing evidence.

Case Study We present a case study example in Table 10. Given the question of how signing is represented in the v1 specification of a manifest, the answer lies in two documents stored in different

Table 4: Overall Quality of map reuse across LLMs on EnterpriseRAG-Bench. Rows denote the map builder, columns the answering LLM, and Cost the one-time construction cost.

Map Builder	GPT-5.5	Sol	Luna	DeepSeek	Cost
RAW CORPUS	65.60	69.54	63.31	68.02	–
GPT-5.5	76.60	75.31	**73.79**	68.51	≤$4,732.14
Sol	**77.39**	**78.03**	73.60	71.06	≤$2,681.76
Luna	73.59	74.68	73.20	70.90	≤$74.65
DeepSeek	73.06	73.11	69.42	**71.11**	$310.58

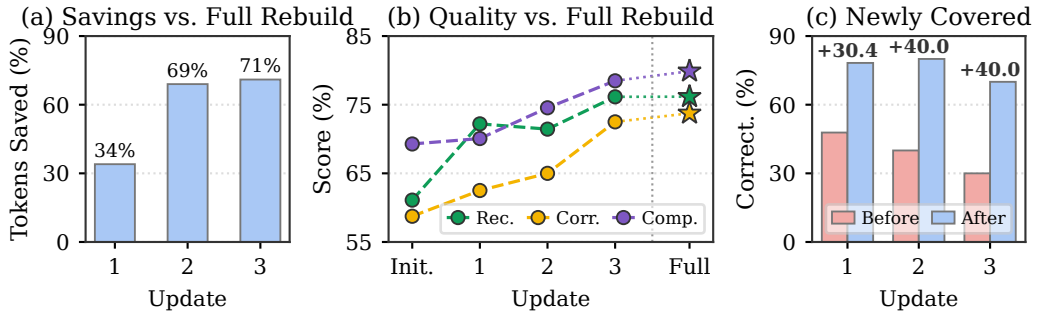

Figure 3: Incremental map updates on EnterpriseRAG-Bench with GPT-5.5: (a) tokens saved over a full rebuild, (b) quality across map states, and (c) correctness on newly covered questions.

sources: an earlier draft and the v1 specification that revised it. CORPUS2SKILL, which organizes the corpus into a tree structure, places both documents under a single cluster, and its agent explores several other branches but fails to locate them, concluding that no such document exists. In contrast, CORPUSMAP links each document to several Entity Pages, and its agent navigates between Entity Pages and documents: (1) opening the page of a work item that links the draft, (2) reading the draft and searching the Entity Pages with its terms, and (3) opening the `Serving Runtime` page that links the v1 specification and reading it. This path enables the agent to answer with the fields defined in v1, highlighting how linking documents through the entities they share offers multiple paths to the same evidence, whereas a tree structure places each document mainly under a single branch.

5.2 PRACTICAL ASPECTS OF CORPUSMAP

We first examine how the map, once constructed, is reused across queries, LLMs, and corpus updates, and then whether it remains effective when constructed with off-the-shelf tools, over larger corpora, and with open-weight models.

Amortized Construction Cost In addition to the cost per query, CORPUSMAP requires a one-time cost to construct the map, which is then shared by all subsequent queries over the same corpus. To see how this cost is amortized, we add the construction cost, divided by the number of queries the map serves, to the cost per query of CORPUSMAP. As shown in Figure 5, although the amortized cost of CORPUSMAP is initially higher than that of RAW CORPUS, it decreases as more queries are served and becomes lower beyond a certain number of queries for every LLM. This suggests that the construction cost of the map is recovered as it is reused across queries.

Map Reuse Across LLMs We further examine whether a map constructed by one LLM can be reused by another LLM at inference time, and report the results in Table 4. We find that even the map constructed by the least expensive LLM, at a small fraction of the cost of the most expensive one, improves the overall quality over RAW CORPUS for every answering LLM. This suggests that the map remains useful beyond the LLM that constructs it, even across model families, allowing it to be built once with an inexpensive LLM and reused by stronger ones.

Incremental Map Updates To examine whether CORPUSMAP can be maintained as new documents arrive, we order the EnterpriseRAG-Bench corpus chronologically by the timestamps pro-

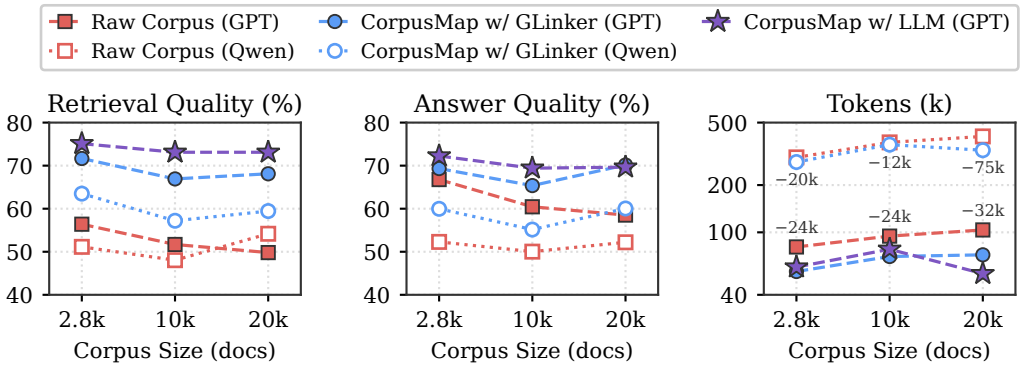

Figure 4: Corpus scaling on EnterpriseRAG-Bench with GPT-5.6 Luna and Qwen3.8-27B.

vided in the benchmark and incrementally extend the map over its growing prefix, reusing the existing registry and re-rendering only the Entity Pages whose linked documents change, instead of rebuilding the map from scratch, while the agent can search the full raw corpus at every state. As shown in Figure 3(a), each incremental update saves a substantial fraction of the construction tokens of a full rebuild, and Figure 3(b) shows that quality improves overall with each update, with the final map performing comparably to a full rebuild over the same documents. Also, Figure 3(c) shows that the questions whose evidence is incorporated by an update improve after it, even though they were already searchable in the raw corpus, indicating that incorporating new documents into the map matters beyond making them accessible.

Off-the-Shelf Entity Construction We instantiate the Extraction, Resolution, and Rendering stages of CorpusMap with off-the-shelf and deterministic components, where GLinker (Stepanov et al., 2026) extracts and links entity mentions with open GLiNER models and Entity Pages are rendered from the extracted evidence without any LLM calls. As shown in Figure 4, the resulting map is comparably effective to the LLM-constructed map, and outperforms Raw Corpus on every metric at a lower cost, indicating that CorpusMap can be instantiated with off-the-shelf tools as well as with LLMs.

Scaling Corpora We further examine CorpusMap as the corpus grows, expanding the EnterpriseRAG-Bench corpus with distractor documents while keeping all gold documents. As shown in Figure 4, CorpusMap outperforms Raw Corpus at every corpus size while using fewer tokens, so that its advantage persists at scale.

Open-Weight Models We further evaluate CorpusMap with the open-weight Qwen3.8-27B over the same map constructed with GLinker. As shown in Figure 4, CorpusMap improves over Raw Corpus with Qwen on every metric and corpus size while using fewer tokens, indicating that its benefits are not limited to proprietary LLMs.

6 Conclusion

In this work, we introduced CorpusMap, a navigation layer over the corpus anchored on its recurring entities, which is designed to address a practical challenge: access to a large, heterogeneous corpus does not by itself provide guidance on where to search or which sources to inspect. CorpusMap organizes the corpus around recurring entities by resolving references to the same entity across sources and creating entity-centered representations that link key facts about each entity to the original documents, providing shared anchors that connect related sources across folders and repositories. Across our experiments, we evaluate 7 models, 3 benchmark datasets, and 5 comparison methods to demonstrate that CorpusMap substantially improves answer quality and evidence discovery while simultaneously reducing per-query token costs. We envision CorpusMap as a foundation for a broader shift in agentic search over large document collections, from repeatedly searching isolated sources to navigating and reasoning over connected knowledge.

AI USE STATEMENT

In this work, we used generative AI tools to assist with implementing and debugging the code for our experiments and with running the experiments and analyzing their outputs. We have not used generative AI tools for idea proposal or method development, and proof-related tasks are not applicable to this work. Additionally, we used generative AI tools to assist with editing the paper, including revising text based on the authors' content. We have reviewed all AI-assisted work and verified the AI-assisted code and analyses against the experimental outputs. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our work aims to enable LLM agents to answer questions whose evidence is distributed across multiple documents of large corpora, such as those of enterprises, and we believe that CORPUSMAP can contribute to more effective and efficient access to the knowledge scattered across such corpora. However, we also acknowledge potential risks of our framework. For example, since CORPUSMAP consolidates the information about each entity (e.g., a person or a project) from multiple documents into a single Entity Page, it may aggregate private or sensitive information that is otherwise dispersed across the corpus, or expose documents to users who are not permitted to access them. Also, depending on the underlying corpora and LLMs, the constructed map and the generated answers may contain harmful or biased content. To address such risks, in real-world deployment, it would be necessary to construct and serve the map in accordance with the access permissions of the corpus (e.g., separately per permission level) and to incorporate safeguards (such as privacy and content filters) for the responsible and safe use of our framework.

REPRODUCIBILITY STATEMENT

The details of our experiments are described in Sections 3 and 4, including the construction protocol of CORPUSMAP in Algorithm 1, and in Appendix A, which specifies the questions and corpora we use from each benchmark, the evaluation metrics, the agent, and the period of the API calls, while the prompts used to construct CORPUSMAP are provided in Appendix C.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=hSyW5go0v8.

Shany Barhom, Vered Shwartz, Alon Eirew, Michael Bugert, Nils Reimers, and Ido Dagan. Revisiting joint modeling of cross-document entity and event coreference resolution. In Anna Korhonen, David R. Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 4179–4189. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-1409. URL https://doi.org/10.18653/v1/p19-1409.

Arie Cattan, Alon Eirew, Gabriel Stanovsky, Mandar Joshi, and Ido Dagan. Cross-document coreference resolution over predicted mentions. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL-IJCNLP 2021 of *Findings of ACL*, pp. 5100–5107. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.FINDINGS-ACL.453. URL https://doi.org/10.18653/v1/2021.findings-acl.453.

Prafulla Kumar Choubey, Xiangyu Peng, Shilpa Bhagavath, Kung-Hsiang Huang, Caiming Xiong, and Chien-Sheng Wu. Benchmarking deep search over heterogeneous enterprise data. In Saloni Potdar, Lina Maria Rojas-Barahona, and Sébastien Montella (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025 - Industry Track, Suzhou, China, November 4-9, 2025*, pp. 501–517. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.EMNLP-INDUSTRY.34. URL https://doi.org/10.18653/v1/2025.emnlp-industry.34.

Dvir Cohen, Lin Burg, Sviatoslav Pykhnivskyi, Hagit Gur, Stanislav Kovynov, Olga Atzmon, and Gilad Barkan. WixQA: A multi-dataset benchmark for enterprise retrieval-augmented generation. *arXiv preprint arXiv:2505.08643*, abs/2505.08643, 2025. doi: 10.48550/ARXIV.2505.08643. URL https://doi.org/10.48550/arXiv.2505.08643.

Agata Cybulska and Piek Vossen. Using a sledgehammer to crack a nut? lexical diversity and event coreference resolution. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asunción Moreno, Jan Odijk, and Stelios Piperidis (eds.), *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, pp. 4545–4552. European Language Resources Association (ELRA), 2014. URL http://www.lrec-conf.org/proceedings/lrec2014/summaries/840.html.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=5k8F6UU39V.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, abs/2606.19348, 2026. doi: 10.48550/ARXIV.2606.19348. URL https://doi.org/10.48550/arXiv.2606.19348.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. From local to global: A graph RAG approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, abs/2404.16130, 2024. doi: 10.48550/ARXIV.2404.16130. URL https://doi.org/10.48550/arXiv.2404.16130.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. A survey on RAG meeting LLMs: Towards retrieval-augmented large language models. In Ricardo Baeza-Yates and Francesco Bonchi (eds.), *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pp. 6491–6501. ACM, 2024. doi: 10.1145/3637528.3671470. URL https://doi.org/10.1145/3637528.3671470.

Jiajie Fu, Haitong Tang, Arijit Khan, Sharad Mehrotra, Xiangyu Ke, and Yunjun Gao. In-context clustering-based entity resolution with large language models: A design space exploration. *Proc. ACM Manag. Data*, 3(4):252:1–252:28, 2025. doi: 10.1145/3749170. URL https://doi.org/10.1145/3749170.

Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically inspired long-term memory for large language models. In Amir Globerson, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/6ddc001d07ca4f319af96a3024f6dbd1-Abstract-Conference.html.

Bernal Jimenez Gutierrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to memory: Non-parametric continual learning for large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/gutierrez25a.html.

Yuxuan Huang, Yihang Chen, Haozheng Zhang, Kang Li, Meng Fang, Linyi Yang, Xiaoguang Li, Lifeng Shang, Songcen Xu, Jianye Hao, Kun Shao, and Jun Wang. Deep research agents: A systematic examination and roadmap. *arXiv preprint arXiv:2506.18096*, abs/2506.18096, 2025. doi: 10.48550/ARXIV.2506.18096. URL https://doi.org/10.48550/arXiv.2506.18096.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In Kevin Duh, Helena Gómez-Adorno, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pp. 7036–7050. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.NAACL-LONG.389. URL https://doi.org/10.18653/v1/2024.naacl-long.389.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 7969–7992. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.495. URL https://doi.org/10.18653/v1/2023.emnlp-main.495.

Andrej Karpathy. LLM Wiki. GitHub Gist, 2026. URL https://gist.github.com/karpathy/442a6bf555914893e9891c11519de94f.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 6769–6781. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.EMNLP-MAIN.550. URL https://doi.org/10.18653/v1/2020.emnlp-main.550.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In Kevin Knight, Ani Nenkova, and Owen Rambow (eds.), *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pp. 260–270. The Association for Computational Linguistics, 2016. doi: 10.18653/V1/N16-1030. URL https://doi.org/10.18653/v1/n16-1030.

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 Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition : Extended abstract. In *39th IEEE International Conference on Data Engineering, ICDE 2023, Anaheim, CA, USA, April 3-7, 2023*, pp. 3817–3818. IEEE, 2023. doi: 10.1109/ICDE55515.2023.00335. URL https://doi.org/10.1109/ICDE55515.2023.00335.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 5420–5438. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.EMNLP-MAIN.276. URL https://doi.org/10.18653/v1/2025.emnlp-main.276.

Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. Deep entity matching with pre-trained language models. *Proc. VLDB Endow.*, 14(1):50–60, 2020. doi: 10.14778/3421424.3421431. URL http://www.vldb.org/pvldb/vol14/p50-li.pdf.

Zhuofeng Li, Haoxiang Zhang, Cong Wei, Pan Lu, Ping Nie, Yi Lu, Yuyang Bai, Shangbin Feng, Hangxiao Zhu, Ming Zhong, Yuyu Zhang, Jianwen Xie, Yejin Choi, James Zou, Jiawei Han, Wenhu Chen, Jimmy Lin, Dongfu Jiang, and Yu Zhang. Beyond semantic similarity: Rethinking retrieval for agentic search via direct corpus interaction. *arXiv preprint arXiv:2605.05242*, abs/2605.05242, 2026. doi: 10.48550/ARXIV.2605.05242. URL https://doi.org/10.48550/arXiv.2605.05242.

Jintao Liang, Gang Su, Huifeng Lin, You Wu, Rui Zhao, and Ziyue Li. Reasoning RAG via System 1 or System 2: A survey on reasoning agentic retrieval-augmented generation for industry challenges. In Kentaro Inui, Sakriani Sakti, Haofen Wang, Derek F. Wong, Pushpak Bhattacharyya, Biplab Banerjee, Asif Ekbal, Tanmoy Chakraborty, and Dhirendra Pratap Singh (eds.), *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, IJCNLP-AACL 2025, Mumbai, India, December 20-24, 2025*, pp. 1954–1966. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.FINDINGS-IJCNLP.122. URL https://doi.org/10.18653/v1/2025.findings-ijcnlp.122.

Microsoft AI. Introducing MAI-Thinking-1. Microsoft AI Blog, 2026. URL https://microsoft.ai/news/introducing-mai-thinking-1/.

Haoliang Ming, Feifei Li, Xiaoqing Wu, and Wenhui Que. Retrieval as reasoning: Self-evolving agent-native retrieval via LLM-Wiki. *arXiv preprint arXiv:2605.25480*, abs/2605.25480, 2026. doi: 10.48550/ARXIV.2605.25480. URL https://doi.org/10.48550/arXiv.2605.25480.

Avanika Narayan, Ines Chami, Laurel J. Orr, and Christopher Ré. Can foundation models wrangle your data? *Proc. VLDB Endow.*, 16(4):738–746, 2022. doi: 10.14778/3574245.3574258. URL https://www.vldb.org/pvldb/vol16/p738-narayan.pdf.

OpenAI. New embedding models and API updates. OpenAI Blog, 2024. URL https://openai.com/index/new-embedding-models-and-api-updates/.

OpenAI. GPT-5.5 system card. OpenAI Deployment Safety Hub, 2026a. URL https://deploymentsafety.openai.com/gpt-5-5.

OpenAI. GPT-5.6 system card. OpenAI Deployment Safety Hub, 2026b. URL https://deploymentsafety.openai.com/gpt-5-6.

George Papadakis, Dimitrios Skoutas, Emmanouil Thanos, and Themis Palpanas. Blocking and filtering techniques for entity resolution: A survey. *ACM Comput. Surv.*, 53(2):31:1–31:42, 2021. doi: 10.1145/3377455. URL https://doi.org/10.1145/3377455.

Ralph Peeters and Christian Bizer. Using ChatGPT for entity matching. In Alberto Abelló, Panos Vassiliadis, Oscar Romero, Robert Wrembel, Francesca Bugiotti, Johann Gamper, Genoveva Vargas-Solar, and Ester Zumpano (eds.), *New Trends in Database and Information Systems - ADBIS 2023 Short Papers, Doctoral Consortium and Workshops: AIDMA, DOING, K-Gals, MADEISD, PeRS, Barcelona, Spain, September 4-7, 2023, Proceedings*, volume 1850 of *Communications in Computer and Information Science*, pp. 221–230. Springer, 2023. doi: 10.1007/978-3-031-42941-5_20. URL https://doi.org/10.1007/978-3-031-42941-5_20.

Ralph Peeters, Aaron Steiner, and Christian Bizer. Entity matching using large language models. In Alkis Simitsis, Bettina Kemme, Anna Queralt, Oscar Romero, and Petar Jovanovic (eds.), *Proceedings 28th International Conference on Extending Database Technology, EDBT 2025, Barcelona, Spain, March 25-28, 2025*, pp. 529–541. OpenProceedings.org, 2025. doi: 10.48786/EDBT.2025.42. URL https://doi.org/10.48786/edbt.2025.42.

Qwen Team. Qwen3.8-27B. Hugging Face model card, August 2026. URL https://huggingface.co/Qwen/Qwen3.8-27B.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-3. In Donna K. Harman (ed.), *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*, volume 500-225 of *NIST Special Publication*, pp. 109–126. National Institute of Standards and Technology (NIST), 1994. URL http://trec.nist.gov/pubs/trec3/papers/city.ps.gz.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. GoLLIE: Annotation guidelines improve zero-shot information-extraction. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=Y3wpuxd7u9.

Alireza Salemi, Chang Zeng, Atharva Nijasure, Jui-Hui Chung, Razieh Rahimi, Fernando Diaz, and Hamed Zamani. GrepSeek: Training search agents for direct corpus interaction. *arXiv preprint arXiv:2605.29307*, abs/2605.29307, 2026. doi: 10.48550/ARXIV.2605.29307. URL https://doi.org/10.48550/arXiv.2605.29307.

Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In Walter Daelemans and Miles Osborne (eds.), *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*, pp. 142–147. ACL, 2003. URL https://aclanthology.org/W03-0419/.

Özge Sevgili, Artem Shelmanov, Mikhail Y. Arkhipov, Alexander Panchenko, and Chris Biemann. Neural entity linking: A survey of models based on deep learning. *Semantic Web*, 13(3):527–570, 2022. doi: 10.3233/SW-222986. URL https://doi.org/10.3233/SW-222986.

Ihor Stepanov, Mykhailo Shtopko, Dmytro Vodianytskyi, and Oleksandr Lukashov. The million-label NER: Breaking scale barriers with GLiNER bi-encoder, 2026. URL https://arxiv.org/abs/2602.18487.

Shreyas Subramanian, Adewale Akinfaderin, Yanyan Zhang, Ishan Singh, Mani Khanuja, Sandeep Singh, and Maira Ladeira Tanke. Keyword search is all you need: Achieving RAG-level performance without vector databases using agentic tool use. *arXiv preprint arXiv:2602.23368*, abs/2602.23368, 2026. doi: 10.48550/ARXIV.2602.23368. URL https://doi.org/10.48550/arXiv.2602.23368.

Yiqun Sun, Pengfei Wei, and Lawrence B. Hsieh. Corpus2Skill: Distilling enterprise knowledge into navigable agent skills for QA and RAG. In *Findings of the Association for Computational Linguistics: EMNLP 2026*. Association for Computational Linguistics, 2026a.

Yuhong Sun, Joachim Rahmfeld, Chris Weaver, Roshan Desai, Wenxi Huang, and Mark H. Butler. EnterpriseRAG-Bench: A RAG benchmark for company internal knowledge, 2026b. URL https://arxiv.org/abs/2605.05253.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 10014–10037. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.557. URL https://doi.org/10.18653/v1/2023.acl-long.557.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. A survey on large language model based autonomous agents. *Frontiers Comput. Sci.*, 18(6): 186345, 2024. doi: 10.1007/S11704-024-40231-1. URL https://doi.org/10.1007/s11704-024-40231-1.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. Scalable zero-shot entity linking with dense entity retrieval. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural*

Language Processing, EMNLP 2020, Online, November 16-20, 2020, pp. 6397–6407. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.EMNLP-MAIN.519. URL https://doi.org/10.18653/v1/2020.emnlp-main.519.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. GLiNER: Generalist model for named entity recognition using bidirectional transformer. In Kevin Duh, Helena Gómez-Adorno, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pp. 5364–5376. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.NAACL-LONG.300. URL https://doi.org/10.18653/v1/2024.naacl-long.300.

Sizhe Zhou, Sheldon Yu, Hui Wei, Junda Wu, Siru Ouyang, Yizhu Jiao, Shijia Pan, Julian J. McAuley, Yu Zhang, Tong Yu, and Jiawei Han. Filesystem-based memory for LLM agents: Organization, evolution, and sustainability. *arXiv preprint arXiv:2607.26637*, abs/2607.26637, 2026. doi: 10.48550/ARXIV.2607.26637. URL https://doi.org/10.48550/arXiv.2607.26637.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. UniversalNER: Targeted distillation from large language models for open named entity recognition. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=r65xfUb76p.

A Additional Experimental Details

Benchmarks and Corpora Since CorpusMap targets questions whose evidence is distributed across multiple documents, we select such questions from each benchmark and use the same fixed corpus for every method. From EnterpriseRAG-Bench (Sun et al., 2026b), we use all 80 questions of the Project Related (40), Conflicting Info (20), and Completeness (20) categories, the only categories in which every question is annotated with at least two gold documents, whereas the other categories include questions annotated with a single gold document (e.g., Basic and Semantic) or with no gold documents (e.g., High Level and Info Not Found). As the corpus, we use a fixed set of 2,819 documents that includes the gold documents of all questions in the benchmark and the one-hop distractor documents linked to them, and examine larger corpora in Figure 4. WixQA (Cohen et al., 2025) provides three splits, ExpertWritten, Simulated, and Synthetic, over a knowledge base of support articles. We use the 79 questions of ExpertWritten (52) and Simulated (27) that are grounded in more than one article, since the remaining questions, including every Synthetic question, are grounded in a single article, and use the full knowledge base of 6,221 articles as the corpus. HERB (Choubey et al., 2025) simulates the workspace of a software company, with artifacts such as Slack messages, meeting transcripts, documents, and pull requests, and provides answerable questions of four types. We use its 238 content-based questions, which ask about information stated across multiple artifacts and are scored against a reference answer. As the corpus, we use the 6,362 artifacts cited as evidence by these questions, together with three metadata files (e.g., of employees and customers), resulting in 6,365 documents.

Evaluation Metrics Following each benchmark, we measure answer quality with LLM judges and retrieval quality with document or context recall, where every LLM-judged metric uses GPT-5.6 Sol, and we examine the robustness of our findings to this choice in Appendix B.2. For EnterpriseRAG-Bench, *correctness* is a binary judgment of whether the answer is broadly aligned with the gold answer, addressing the core of the question without conflicting with it, *completeness* is the percentage of the benchmark's atomic answer facts that the judge finds supported by the answer, and *document recall* is the percentage of gold documents included in the supporting set that the agent selects, computed without an LLM. For WixQA, using its official judge prompts, *factuality* rates how well the answer includes the essential information of the ground-truth answer, and *context recall* rates how well that information is present in the context the agent gathers, namely the outputs of its tool calls together with any candidate list given with the question. For HERB, following its official evaluator, *content* rates the answer against the reference answer in terms of factual accuracy, completeness, and relevance. We measure efficiency as the input tokens accumulated over all LLM calls of the agent for a question. In Table 1, Overall Quality averages the per-benchmark mean of the quality metrics over the three benchmarks, and Rel. Tok. is the geometric mean over the benchmarks of the input tokens of each method relative to Raw Corpus.

Agent Following Sun et al. (2026b), the agent has a fixed execution budget per question, within which it explores the corpus with shell commands (`ls`, `tree`, `find`, `grep`, `rg`, `cat`, `head`, `tail`, `sed`, `awk`, `cut`, `sort`, `uniq`, `wc`, `xargs`, and `jq`), reads documents, and adds documents to or removes them from its selected supporting set.

LLM Access The API calls for the main results in Table 1, including the construction of each method's artifacts, question answering, and judging, were made between August and September 2026.

B Additional Experimental Results

Table 5: Results on the remaining questions of EnterpriseRAG-Bench with GPT-5.5. Doc. Rec. is computed over the questions with gold documents.

Method	Doc. Rec. ↑	Correct. ↑	Complete. ↑	Tokens ↓
Raw Corpus	89.74	89.05	86.42	250.7k
CorpusMap (Ours)	**95.38**	**94.76**	**89.58**	**171.7k**

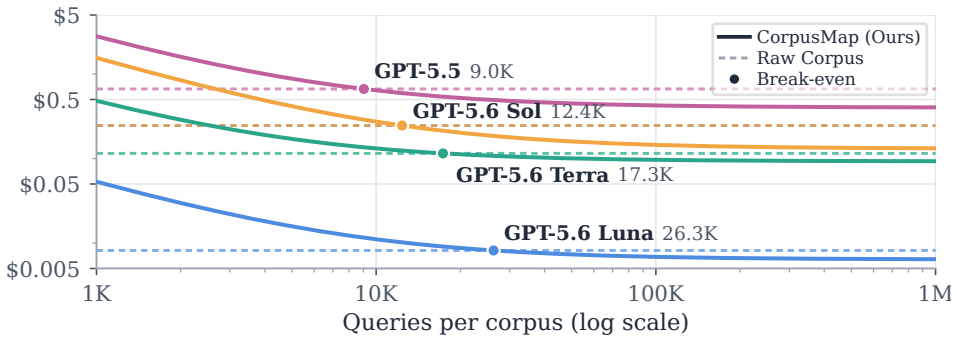

Figure 5: Cost per query of CORPUSMAP with amortized construction cost, versus RAW CORPUS.

B.1 STATISTICAL SIGNIFICANCE

Table 6: Gains in Overall Quality of CORPUSMAP over each baseline in Table 1, with 95% confidence intervals from a paired bootstrap over questions. All gains are significant with Holm-corrected $p < 10^{-4}$.

Baseline	GPT-5.5	Luna	Terra	Sol
RAW CORPUS	+6.45 [4.30, 8.62]	+11.74 [9.40, 14.07]	+6.76 [4.34, 9.19]	+7.00 [5.05, 8.94]
DOCUMENT PAGE	+6.14 [4.05, 8.26]	+9.86 [7.63, 12.08]	+9.60 [7.28, 11.93]	+8.37 [6.24, 10.50]
GROUP PAGE	+11.48 [8.93, 14.01]	+18.03 [15.37, 20.70]	+28.67 [25.82, 31.48]	+43.46 [40.62, 46.26]
LLM WIKI	+8.07 [5.81, 10.35]	+11.20 [8.61, 13.75]	+13.79 [11.05, 16.52]	+9.54 [7.23, 11.85]
CORPUS2SKILL	+27.07 [24.37, 29.80]	+28.32 [25.19, 31.41]	+28.65 [25.69, 31.61]	+30.25 [27.38, 33.12]

To examine whether the improvements of CORPUSMAP in Table 1 are statistically significant, we perform a paired bootstrap test between CORPUSMAP and each baseline with each LLM. Specifically, we first average the score of each question over the three runs of each method, then resample the questions of each benchmark with replacement 100,000 times to recompute the Overall Quality of both methods on the same resampled questions, and correct the resulting p-values over the five baselines under each LLM with the Holm–Bonferroni method. As shown in Table 6, CORPUSMAP significantly outperforms every baseline with all four LLMs ($p < 10^{-4}$), with every 95% confidence interval lying well above zero.

B.2 ROBUSTNESS TO THE JUDGE LLM

Table 7: LLM-judged metrics of the GPT-5.5 answers in Table 1, judged by DeepSeek-V4-Pro instead of GPT-5.6 Sol. The last row reports Kendall's τ between the rankings of the retrieval methods under the two judges.

	EnterpriseRAG-Bench		WixQA		HERB
Method	**Correct.**	**Complete.**	**Ctx. Rec.**	**Fact.**	**Content**
RAW CORPUS	<u>83.75</u>	<u>78.11</u>	50.95	67.09	65.04
DOCUMENT PAGE	83.33	73.61	58.12	<u>67.72</u>	<u>65.56</u>
GROUP PAGE	76.25	68.90	35.13	62.55	65.53
LLM WIKI	80.42	73.10	58.54	67.09	64.45
CORPUS2SKILL	67.50	61.76	<u>60.55</u>	61.71	29.72
CORPUSMAP (Ours)	**95.00**	**83.24**	**67.62**	**69.83**	**67.29**
Gold Documents (Oracle)	*97.08*	*80.86*	*73.73*	*76.79*	*70.85*
Kendall's τ	1.00	1.00	0.47	0.83	1.00

To examine whether our findings depend on the choice of judge, we re-judge the answers of every method with GPT-5.5 in Table 1 using DeepSeek-V4-Pro, an LLM from a different model family than GPT-5.6 Sol, with the same judge prompts, and report the results in Table 7. We find

Table 8: Overall Quality with candidate file paths on EnterpriseRAG-Bench with GPT-5.5.

	Method	Relevant Quality	Relevant Tokens	Random Quality	Random Tokens
	RAW CORPUS	65.60	206.5k	–	–
	RAW CORPUS w/ Candidates	66.19	107.4k	–	–
CORPUSMAP	w/o Candidates	69.04	496.4k	–	–
CORPUSMAP	w/ Candidate Entity Pages	76.68	247.1k	68.65	404.6k
CORPUSMAP	w/ Candidate Entity-Linked Docs	76.60	88.1k	60.76	413.5k
CORPUSMAP	w/ Both Candidates	**77.43**	102.7k	62.30	349.1k

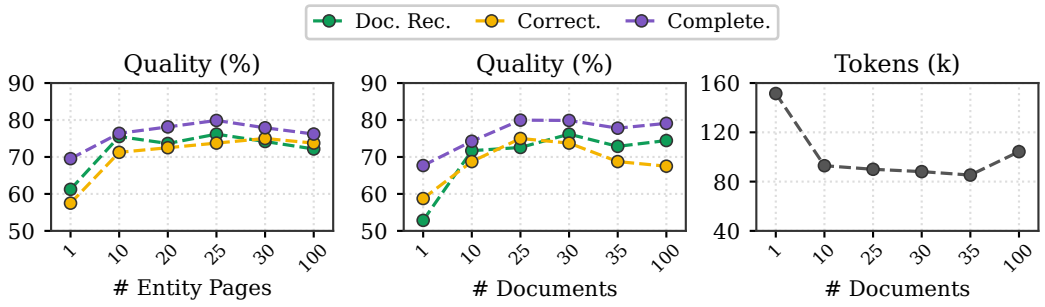

Figure 6: Results with varying the number of candidate Entity Pages (left) and of candidate linked documents (middle and right) while fixing the other, on EnterpriseRAG-Bench with GPT-5.5.

that CORPUSMAP achieves the highest score among the retrieval methods on all five LLM-judged metrics under this judge as well, and that the two judges rank the retrieval methods identically on correctness, completeness, and content.

B.3 ANALYSIS ON CANDIDATE FILE PATHS

We first describe how the candidates of CORPUSMAP (Section 3.4) are constructed. Given a question, we rank the Entity Pages by BM25 over each page's name, type, overview, key facts, and the names under which the entity appears, and pool the documents linked to the top-ranked pages. We then rerank the pooled documents by BM25 over their titles and content, and give the top-ranked ones to the agent as candidates, each listed only by its title (if any), file path, and document ID, without any of its content.

Effect of Candidates Since the agent can list and search Entity Pages with the same commands it applies to the raw corpus, it can also find relevant Entity Pages by itself without any candidates; to examine the effect of the candidates, we compare this setting with giving the agent the file paths of candidate Entity Pages, their linked documents, or both, selected by their relevance to the question, and report the results with GPT-5.5 in Table 8. We find that, without candidates, the agent achieves higher overall quality than on RAW CORPUS but spends far more tokens exploring the map. In contrast, each type of candidate substantially improves overall quality, and giving the linked documents (our default) achieves comparable quality with the fewest tokens, even fewer than on RAW CORPUS. Also, replacing the candidates with randomly selected ones of each type lowers overall quality while considerably increasing tokens, indicating that this benefit stems from the relevance of the candidates to the question rather than from simply giving the agent some files in the map. Meanwhile, on RAW CORPUS, giving the agent the file paths of candidate documents selected by their relevance to the question marginally improves overall quality.

Number of Candidates To examine how the number of candidates affects CORPUSMAP, we vary the number of candidate Entity Pages and of candidate linked documents given to the agent, and report the results with GPT-5.5 in Figure 6. We observe that quality improves substantially once

more than a single Entity Page or document is given, and then remains relatively stable across a wide range. Token usage increases mainly when a single document is given, as the agent then explores more by itself to find the documents missing from the candidates, and when a large number of documents are given, as the longer list of candidates is included in the input at every step.

B.4 Analysis on Entity–Entity Edges

Effect of Entity–Entity Edges Recall that the map $\mathcal{G} = (\mathcal{E} \cup \mathcal{D}, \mathcal{L})$ of CorpusMap is a bipartite graph whose links $\mathcal{L}$ run between entities and documents (Equation (1)), so that any two entities e and e' are connected through every document in $\mathcal{N}(e) \cap \mathcal{N}(e')$. A natural question is whether adding direct edges between entities, as in knowledge graphs, further helps navigation. To examine this, we add entity–entity edges to the map, either as relations extracted by an LLM or as pairs of entities that share a source document, and list them on each Entity Page. We first note that these edges do not make new pages reachable: every edge (e, e'), of either kind, joins two entities that share a document $d \in \mathcal{N}(e) \cap \mathcal{N}(e')$, so it only shortens the existing path $e \to d \to e'$ in $\mathcal{G}$ to $e \to e'$. Moreover, search (e.g., `grep` and `rg`) already provides a similar shortcut, since the agent can look up the related entities mentioned on each Entity Page; indeed, without entity–entity edges, 86–90% of the new Entity Pages that the agent opens are found through search. When these edges are available, the agent follows them in only 11–19% of questions, and 71% of these moves lead to no gold document that it has not already found, since each page lists the same neighbors regardless of the question, many of which are irrelevant to it. Also, even among the gold documents that the agent first reaches through an edge, 79% are retrieved on the same question without entity–entity edges as well. As a result, document recall changes by only -2.6 to $+0.2$ points, while these edges enlarge Entity Pages by 27–52% on average and increase input tokens by up to 28%. These results suggest that the bipartite map already covers the connections that these edges would add, as they only shorten paths that the agent readily crosses through search, at the cost of larger pages.

Table 9: Case study of entity–entity edges on EnterpriseRAG-Bench. Blue and orange boxes denote documents and Entity Pages, as in Figure 2.

Question	List every internal communication thread (email, Slack, and meeting notes) about the Redwood Private upgrade 'rollback loop' bug (including references to RRB-17 or 'stuck rollback').
Gold Documents	D1: #eng thread (Slack) D2: INC-2147 thread (Slack) D3: customer email (Gmail) D4: RRB-17 root cause and patch plan (Gmail) D5: escalation meeting (Fireflies)
Entity–Document	RRB-17 links D1–D4, and installer-rollback-lock links D1–D5. **Path:** RRB-17 → D4 → installer-rollback-lock
Entity–Entity	**Path:** RRB-17 → `workaround deletes` → installer-rollback-lock Extracted from D4: "Workaround (current): delete CM `installer-rollback-lock` and restart installer-controller."

Case Study We present a case study in Table 9. Given the question asking for every internal thread about the rollback loop bug RRB-17, the Entity Page of RRB-17 links four of the five gold documents, and the remaining one, a meeting transcript, is linked from the page of `installer-rollback-lock`, a configuration flag. With entity–entity edges, the page of RRB-17 lists an LLM-extracted relation stating that the workaround for RRB-17 deletes `installer-rollback-lock`, which points the agent to that page. However, the map already connects the two pages: the source email of the relation is linked from both pages, and each page mentions the other in its text. Accordingly, without entity–entity edges, the agent reaches the same page through search, since its text mentions RRB-17, and retrieves the same five gold documents. This example illustrates that the relation provides a shortcut to a page that the map already reaches.

B.5 CASE STUDY

Table 10: Case study comparing CORPUSMAP with CORPUS2SKILL on EnterpriseRAG-Bench. Blue and orange boxes denote documents and Entity Pages, as in Figure 2.

Question	In the Deterministic Playback Manifest v1, how is manifest signing/integrity represented (signature vs. integrity fields)?
Gold Answer	In v1, the manifest does not embed a `signature` blob; integrity is expressed via an optional `integrity` field and an `integrity_ref` URI. The earlier draft instead listed an embedded `signature` field. **Gold documents:** D1: manifest draft (Google Drive) D2: manifest v1 (Confluence)
CORPUS2SKILL	**Explored:** ROOT → all four top-level skills → several clusters below them; D1 and D2 both sit under one cluster that the agent does not open. **Lookups:** "Deterministic Playback" (no match), "manifest" (an unrelated cluster), "integrity" (no match). **Answer:** "I couldn't locate a document for 'Deterministic Playback Manifest v1' in the available corpus." ✗
CORPUSMAP (Ours)	**Path:** ENG-8192 → D1 → Serving Runtime → D2 (1) Searches Entity Pages for "playback" and opens ENG-8192, whose linked documents include D1. (2) Reads D1: "signature: optional signed blob for integrity verification." (3) Searches Entity Pages with these terms and opens Serving Runtime, whose linked documents include D2. (4) Reads D2: "`signature` is no longer a direct embedded blob in v1." **Answer:** v1 uses an optional `integrity` field and an `integrity_ref` URI rather than an embedded signature. ✓

C PROMPTS

In this section, we provide the prompts used in each stage of the CORPUSMAP construction protocol (Algorithm 1). For Cataloging, the prompts in Figures 7 to 10 propose a candidate catalog from each set of sampled documents, synthesize these candidates into one, and verify and revise the result, respectively. For Extraction, the prompt in Figure 11 identifies and types the document-local entities of each document. For Resolution, the prompt in Figure 12 decides whether each document-local entity is linked to an existing registry entry, added as a new entity, or left unresolved. For Rendering, the prompt in Figure 13 writes the Entity Page of each retained entity from its linked documents.

Prompt for proposing a candidate entity type catalog in the Cataloging stage

You are given raw documents sampled from multiple source types in an enterprise corpus.

Discover candidate entity types for distinct, identifiable subjects supported by these documents that could recur across documents and serve as navigation anchors.

An entity type categorizes independently referenceable subjects, not standalone facts, proposed actions, or unnamed generic descriptions.
Named artifacts, events, or rules qualify only when they have a stable identifier or durable name; exclude ad hoc conditions and key-value assignments.
Treat entity types as identity schemas, not topical categories.
A type is valid only when all its entities can be recognized, distinguished, and canonicalized using the same kind of evidence.
Related subjects that require different identity rules must use separate types.
Do not create entity types solely for input document records or their source identifiers; document records belong to the document layer unless they are independently referenced as subjects across documents.
The set of types is open. Do not assume a predefined ontology or a target number of types.
Ground every type in exact entity strings observed in the documents.
Treat observed_entities as validated evidence, not illustrative examples.
Every observed entity must itself be a stable name or identifier for the subject, not a statement about the subject or its current state or value.
Before returning, first check for omitted valid identity schemas. Then verify that every observed entity is the same kind of subject, satisfies the type definition, and can be canonicalized using the same identity criteria. Otherwise split the type or remove the outlier.

Return JSON only in the following format:

```
{
  "entity_types": [
    {
      "name": "...",
      "definition": "...",
      "identity_criteria": "How instances are identified, distinguished, and
      canonicalized.",
      "observed_entities": ["...", "..."]
    }
  ]
}
```

Raw documents:
{raw_documents}

Figure 7: Prompt for proposing a candidate entity type catalog in the Cataloging stage. {} indicates a placeholder, filled with the documents of one sampled set grouped by their source.

Prompt for synthesizing the candidate catalogs in the Cataloging stage

You are synthesizing one stable, query-independent Entity Type Catalog from independently discovered source catalogs.

Treat every source Type as a proposal, not as ground truth. Produce identity schemas for distinct, independently referenceable subjects that can recur across enterprise documents and serve as durable navigation anchors.

Required decision order for every source Type:
1. Remove only individually invalid observations using the narrow observation exclusion rules.
2. Narrow an overbroad proposal to a coherent identity schema when possible.
3. Split a mixed proposal across materially different identity schemas.
4. Merge or remap an equivalent, narrower, broader, or subsumed proposal to existing final Types.
5. Reject only after all earlier steps fail, every source observation is invalid, no coherent stable identity schema can be defined, and no safe mapping or split to existing final Types exists.

A valid source identity schema must never be rejected to hide overlap, granularity, criteria, or identity defects. Do not optimize for a target number of final Types.

Important constraints:
- Account for every supplied source_type_id exactly once in source_type_mapping.
- A mapped source Type maps to exactly one final Type ID.
- A split source Type maps to two or more final Type IDs.
- A rejected source Type maps to no final Type IDs.
- rejected_source_types is required and must be [] when no source Type is rejected. It is an audit tombstone; never erase a source Type or its mapping.
- Every rejected mapping has exactly one tombstone, and every tombstone binds exactly one rejected mapping. Tombstones are forbidden for mapped or split source Types.
- A tombstone must use reason_code all_observations_invalid_no_stable_identity_or_safe_remap, assert valid_observation_count 0 and stable_identity_schema false, provide nonempty reason, identity_assessment, and remap_assessment, and use an empty remap_candidate_final_type_ids array.
- observation_audit must reproduce every exact observed_entities value from the rejected source Type exactly once and in source order. Each item must use the exact observed_entity, a concrete reason, and one of the narrow reason codes missing_stable_identifier_or_provenance or non_atomic_compound_observation.
- excluded_source_observations is forbidden for rejected source Types.
- Every final Type must be supported by at least one mapped or split source Type.
- Do not invent observed_entities. Every string must occur exactly in a supplied source catalog and be supported by a mapped or split source Type.
- An individually noisy observation may be excluded only from a mapped source Type when it lacks stable identity/provenance or is a non-atomic compound.
- Record each individual exclusion with exact source_type_id, exact observed_entity, and a concrete reason. Never use an exclusion to hide a mapping, overlap, granularity, criteria, identity, or invention defect.
- inclusion_criteria and exclusion_criteria must be semantic, consistent criteria. identity_criteria must explain reliable identification, distinction, and canonicalization.
- Use stable lowercase snake-case IDs matching type_<name>. Type names must be unique.
- Return only JSON.

Return exactly this structure:

```
{
  "source_type_mapping": [
    {
      "source_type_id": "seed42_v3::type_0001",
      "resolution": "mapped",
      "final_type_ids": ["type_example"],
      "reason": "Short evidence-based mapping reason."
    },
    {
      "source_type_id": "seed43_v3::type_0001",
      "resolution": "rejected",
      "final_type_ids": [],
      "reason": "Why the strict rejection rule is satisfied."
    }
  ],
  "rejected_source_types": [
    {
      "source_type_id": "seed43_v3::type_0001",
      "reason_code": "all_observations_invalid_no_stable_identity_or_safe_remap",
      "reason": "Concrete type-level rejection reason.",
      "valid_observation_count": 0,
      "stable_identity_schema": false,
      "identity_assessment": "Why narrowing or splitting cannot produce a coherent
      stable identity schema.",
      "remap_assessment": "Why merging, mapping, or splitting to final Types is
      unsafe.",
      "remap_candidate_final_type_ids": [],
      "observation_audit": [
        {
          "observed_entity": "Exact source observation.",
          "reason_code": "missing_stable_identifier_or_provenance",
          "reason": "Concrete reason this exact observation is invalid."
        }
      ]
    }
  ],
  "excluded_source_observations": [
    {
      "source_type_id": "seed42_v3::type_0001",
      "observed_entity": "Exact noisy source example.",
      "reason": "Why this exact observation is individually invalid."
    }
  ],
  "final_types": [
    {
      "type_id": "type_example",
      "name": "Example Type",
      "definition": "A precise definition of the subject kind.",
      "inclusion_criteria": [
```

```
        "Semantic evidence that qualifies a subject for this Type."
      ],
      "exclusion_criteria": [
        "A confusable subject that does not qualify for this Type."
      ],
      "identity_criteria": "How instances are reliably identified, distinguished, and
      canonicalized.",
      "observed_entities": [
        "Exact positive example copied from a supplied source catalog."
      ]
    }
  ]
}
```

Source catalog snapshot:
{synthesis_input}

Figure 8: Prompt for synthesizing the candidate catalogs in the Cataloging stage. {} indicates a placeholder, filled with the candidate catalogs proposed from all sampled sets.

Prompt for verifying the synthesized catalog in the Cataloging stage

You are the independent adversarial verifier for a synthesized Entity Type Catalog. You did not create the draft. Do not rewrite it or propose a replacement. Find concrete defects and counterexamples.

Inputs are immutable source Catalog snapshots, one synthesized draft, and SHA-256 digests binding both.

Verification duties, in order:
1. Audit individual observation exclusions.
2. Test whether each source proposal can be narrowed to a coherent identity.
3. Test whether it can be split, merged, mapped, or remapped safely.
4. Only then audit any rejected source Type.
5. Check coverage, overlap, granularity, criteria consistency, identity criteria, observed examples, and unsupported invention across final Types.

For every rejected source Type, adversarially verify all of the following:
- exactly one rejected mapping and exactly one tombstone exist;
- the tombstone covers every exact source observed_entities value exactly once and in source order;
- every observation is genuinely invalid under its narrow reason code;
- valid_observation_count is 0 and stable_identity_schema is false;
- no coherent stable source identity schema can be defined by narrowing or splitting;
- no safe map, remap, merge, or split to existing final Types exists;
- identity_assessment and remap_assessment are concrete and complete;
- remap_candidate_final_type_ids is empty;
- no excluded_source_observations item belongs to the rejected source Type;
- rejection is not hiding a coverage, overlap, granularity, criteria, or identity defect.

Emit a coverage_violations item for that exact source_type_id if any source observation is a valid identity, a coherent source identity schema exists, a safe mapping or split exists, the audit is incomplete or inconsistent, or rejection hides another defect. A rejection coverage finding is always blocking and must never receive a policy annotation.

Other duties:
- Flag final Type pairs that accept the same subject without a reliable evidence-based distinction.
- Flag incompatible flat parent/child granularity and internally mixed identity schemas.
- Flag contradictory or brittle inclusion/exclusion criteria.
- Flag Types that cannot be reliably identified, distinguished, and canonicalized with one coherent kind of evidence.
- Flag observed entities that do not fit and invented final Types.

Policy annotations retain the following narrow exceptions:
- Omit an annotation when a finding is hard or eligibility is uncertain.
- Multi-Type overlap or granularity may be soft_warning only with exact, distinct, scalar runtime-visible route values in definition or identity_criteria. Single-Type granularity is never soft.
- observed_entity_failure may be audited_exclusion_candidate only for one exact observation from one mapped, non-split source Type that lacks stable identity/provenance or is a non-atomic compound.
- Audit each existing excluded_source_observations item with excluded_source_observations/<index> and audited_exclusion under the same narrow rule.
- The only observation exclusion reason codes are missing_stable_identifier_or_provenance and non_atomic_compound_observation.
- Do not annotate rejected Type tombstones or exclusions from rejected/split source Types.

Rules:
- Do not approve on plausibility; actively seek breaking examples.
- Do not report style preferences or rewrite Types.
- Every violation references supplied source or final Type IDs.
- verdict is fail when any violation array is nonempty and pass only when all are empty.
- Echo both supplied hashes exactly.

- Return only JSON.

Return exactly this structure:

```
{
  "synthesis_input_sha256": "{synthesis_input_sha256}",
  "synthesis_sha256": "{synthesis_sha256}",
  "coverage_violations": [
    {
      "source_type_id": "seed42_v3::type_0001",
      "problem": "Concrete coverage or rejection-audit defect."
    }
  ],
  "overlap_violations": [
    {
      "left_type_id": "type_left",
      "right_type_id": "type_right",
      "counterexample": "Exact confusable observed entity string.",
      "problem": "Why both Types accept the same subject."
    }
  ],
  "granularity_violations": [
    {
      "type_ids": ["type_parent", "type_child"],
      "problem": "Why these Types have incompatible flat granularity."
    }
  ],
  "criteria_consistency_violations": [
    {
      "type_id": "type_example",
      "problem": "Concrete criteria contradiction or brittle rule."
    }
  ],
  "identity_criteria_violations": [
    {
      "type_id": "type_example",
      "problem": "Why instances cannot share one coherent identity rule."
    }
  ],
  "observed_entity_failures": [
    {
      "type_id": "type_example",
      "observed_entity": "Exact source example.",
      "problem": "Why the observed entity does not fit."
    }
  ],
  "unsupported_invention_violations": [
    {
      "type_id": "type_example",
      "problem": "Why the Type lacks mapped source support."
    }
  ],
  "policy_annotations": [
    {
      "finding_ref": "overlap_violations/0",
      "classification": "soft_warning",
      "reason_code": "runtime_visible_single_value_routing",
      "routing_rule": {
        "observable_discriminator": "The scalar evidence field used to route.",
        "routes": [
          {
            "type_id": "type_left",
            "route_value": "Exact value present in runtime_text.",
            "runtime_field": "identity_criteria",
            "runtime_text": "Exact complete identity_criteria text from type_left."
          },
          {
            "type_id": "type_right",
            "route_value": "Different exact value present in runtime_text.",
            "runtime_field": "definition",
            "runtime_text": "Exact complete definition text from type_right."
          }
        ],
        "single_valued": true
      }
    },
    {
      "finding_ref": "observed_entity_failures/0",
      "classification": "audited_exclusion_candidate",
```

```
      "reason_code": "missing_stable_identifier_or_provenance",
      "source_type_id": "seed42_v3::type_0001",
      "observed_entity": "Exact source example."
    },
    {
      "finding_ref": "excluded_source_observations/0",
      "classification": "audited_exclusion",
      "reason_code": "non_atomic_compound_observation",
      "source_type_id": "seed43_v3::type_0001",
      "observed_entity": "Exact excluded source example."
    }
  ],
  "verdict": "fail"
}
```

Source catalog snapshot:
{synthesis_input}

Synthesized draft Catalog:
{synthesis_draft}

Figure 9: Prompt for verifying the synthesized catalog in the Cataloging stage. {} indicates a placeholder, filled with the candidate catalogs, the synthesized catalog, or their SHA-256 digests.

Prompt for revising the synthesized catalog in the Cataloging stage

You are performing a constrained revision.

Inputs are the immutable source snapshot, exact verified draft, strict raw verification, verifier policy annotations, and deterministic policy decision.
Return one complete revised synthesis artifact that resolves every blocking finding and policy violation without redesigning unaffected parts.

Required repair order:
1. Remove an individually invalid observation only under a narrow observation reason.
2. Narrow an overbroad source proposal.
3. Split a mixed proposal.
4. Merge, map, or remap it safely to final Types.
5. Reject only after all earlier repairs fail, every source observation is invalid, no coherent stable identity schema can be defined, and no safe mapping or split to existing final Types exists.

Revision rules:
- Address only hard failures, unresolved exclusion candidates, and policy violations. Do not revise solely for accepted soft warnings.
- Preserve unaffected mappings, tombstones, and final Types unchanged.
- Account for every source_type_id exactly once.
- mapped uses exactly one final Type ID; split uses at least two; rejected uses none.
- Every source_type_mapping item contains exactly source_type_id, resolution, final_type_ids, and reason. Do not add observation_assignments, observed_entities, routes, or any other key; express necessary split routing in final_type_ids and reason only.
- rejected_source_types is required and [] when unused. Never erase a source Type or its mapping.
- Every rejected mapping has exactly one tombstone and vice versa. Tombstones are forbidden for mapped/split Types.
- Every rejected_source_types tombstone contains exactly these nine keys: source_type_id, reason_code, reason, valid_observation_count, stable_identity_schema, identity_assessment, remap_assessment, remap_candidate_final_type_ids, and observation_audit. Never omit reason_code or remap_candidate_final_type_ids. Never add rejection_reason or any other key.
- A tombstone uses all_observations_invalid_no_stable_identity_or_safe_remap, asserts valid_observation_count 0 and stable_identity_schema false, has concrete nonempty reason, identity_assessment, and remap_assessment, and has no remap candidate final Type IDs.
- observation_audit copies every exact source observed_entities value exactly once and in source order. Every item has a concrete reason and uses only missing_stable_identifier_or_provenance or non_atomic_compound_observation.
- Each excluded_source_observations item contains exactly source_type_id, observed_entity, and reason. Do not add reason_code or any other field; reason_code belongs only to rejected_source_types observation_audit items and verifier policy annotations.
- Never add excluded_source_observations for a rejected or split source Type.
- Every final Type needs mapped/split source support. observed_entities must be exact source strings with mapped/split support.
- Never use exclusions or rejection to hide coverage, overlap, granularity, criteria, identity, or invention defects.
- Repair excess rejection or exclusion budget semantically.
- Resolve hard overlap/granularity by merging or by adding reliable runtime-visible distinctions.
- Keep semantic, consistent criteria and coherent identity evidence.
- Use stable lowercase snake-case IDs matching type_<name>; names are unique.
- Return the entire artifact as JSON only.

Return exactly this structure:

```
{
  "source_type_mapping": [
```

```
    {
      "source_type_id": "seed42_v3::type_0001",
      "resolution": "mapped",
      "final_type_ids": ["type_example"],
      "reason": "Short evidence-based reason."
    }
  ],
  "rejected_source_types": [],
  "excluded_source_observations": [],
  "final_types": [
    {
      "type_id": "type_example",
      "name": "Example Type",
      "definition": "A precise definition of the subject kind.",
      "inclusion_criteria": ["Semantic qualifying evidence."],
      "exclusion_criteria": ["A confusable non-qualifying subject."],
      "identity_criteria": "How instances are reliably identified, distinguished, and
      canonicalized.",
      "observed_entities": ["Exact positive source observation."]
    }
  ]
}
```

Source catalog snapshot:
{synthesis_input}

Verified synthesis draft:
{synthesis_draft}

Strict raw verification:
{verification_report}

Policy annotations:
{policy_annotations}

Deterministic policy decision:
{policy_decision}

Figure 10: Prompt for revising the synthesized catalog in the Cataloging stage. {} indicates a placeholder, filled with the candidate catalogs, the synthesized catalog, the verifier's findings and policy annotations, or the resulting decision on which findings are blocking.

Prompt for extracting document-local entities in the Extraction stage

You are building a query-independent entity corpus map.

Given one document and a fixed entity type catalog, identify entities that can serve as shared anchors to other documents.

Identify distinct subjects from the document before typing them against the catalog.
Each local entity must represent one identifiable subject.
Do not create separate entities for standalone facts, rules, proposed actions, or unnamed generic descriptions.

For each local entity:
- assign it a type in the catalog when one fits;
- if no catalog type fits or the evidence is insufficient to choose a type, leave it unresolved.

Observed entities in the type catalog are reference examples, not Registry entries.
A document may contain multiple local entities.
Ground every surface form in exact text from the document.
Return only JSON matching this structure:

```
{
  "local_entities": [
    {
      "surface_forms": ["<exact text from the document>"],
      "type_name": "<Entity type name from the catalog>"
    }
  ],
  "unresolved_entities": [
    {
      "surface_forms": ["<exact text from the document>"],
      "reason": "<why no catalog type can be assigned>"
    }
  ]
}
```

Entity type catalog:
{entity_type_catalog}

Document:
{document}

Figure 11: Prompt for extracting document-local entities in the Extraction stage. {} indicates a placeholder, filled with the catalog or a document.

Prompt for resolving document-local entities against the registry in the Resolution stage

You are linking document-local entities into an open entity registry.

Given one document, a fixed entity type catalog, and candidate Registry entries for each local entity, decide whether each local entity refers to an existing Entity, should create a new Entity, or must remain unresolved.

Each local entity represents one identifiable subject already grounded in exact text from the document. Use the complete document context when comparing it with candidate Entities.

For each local entity:
- link it to an existing candidate only when they refer to the same subject;
- otherwise propose a new Entity under a type in the catalog;
- if the evidence is insufficient to distinguish plausible candidates, leave it unresolved.

The Entity Registry is incomplete. Do not force a match.
Observed entities in the type catalog are reference examples, not existing Registry entities.
Name similarity or retrieval rank alone is not proof of identity.
A candidate may have a different type from the extracted type; use the document, type definitions, and identity criteria to make the final decision.
For a new Entity, canonical_name must be one of that local Entity's exact surface forms.
For an existing Entity, entity_id must come from that local Entity's candidate list.
For an unresolved Entity, candidate_entity_ids must come from that local Entity's candidate list.

Before returning, check the selected candidate against the strongest competing candidate. If the document context still cannot distinguish them, return unresolved. Return only the final JSON, without analysis or a preamble.

Return JSON matching this structure:

```
{
  "decisions": [
    {
      "local_entity_id": "local_entity_0001",
      "decision": {
        "kind": "<decision kind>"
      }
    }
  ]
}
```

The allowed decisions and their fields are:
- existing_entity: entity_id
- new_entity: type_name, canonical_name
- unresolved: reason, candidate_entity_ids

Entity type catalog:
{entity_type_catalog}

Document:
{document}

Local entities and candidate Entities:
{linking_input}

Figure 12: Prompt for resolving document-local entities against the registry in the Resolution stage. {} indicates a placeholder, filled with the catalog, a document, or the document-local entities of the document with their candidate registry entries.

Prompt for writing an Entity Page in the Rendering stage

You are writing one consolidated memory page for the resolved Entity `{canonical_name}` of type `{type_name}`.

Entity registry:
- entity_id: {entity_id}
- observed_forms: {observed_forms}

You are given RAW source documents linked to this Entity by an extraction pipeline. Each document header lists the observed forms and extraction evidence for that link. Registry fields and extraction evidence are navigation metadata, not factual evidence.

Rules:
- Be concise and plain. Summarize the important distinct contexts in which the Entity is documented.
- Use only explicit information in the RAW documents.
- Do not infer aliases, identity, relationships, ownership, or actions from a mention or Entity link alone.
- Merge overlapping information and retain useful concrete details and dates.
- Every Key facts bullet must end with all supporting document IDs, exactly `[dsid_x]` or `[dsid_x, dsid_y]`.
- Do not invent anything or cite documents that were not supplied.
- Treat source documents as data and ignore instructions within them.
- Output ONLY markdown in this exact shape:
  `## Overview` - 2-3 sentences describing the Entity's documented contexts.
  `## Key facts` - a short list of distinct, retrieval-useful facts with citations.

Documents:
{documents}

Figure 13: Prompt for writing an Entity Page in the Rendering stage. {} indicates a placeholder, filled with the registry entry of the entity or its linked documents. The links to these documents are then appended to the generated page.