Journal Homepage: www.ijarpr.com ISSN: 3049-0103 (Online)

International Journal of Advance Research Publication and Reviews

Vol 03, Issue 9, pp 749-762, September, 2026

Retrieval Methods for Persistent AI Agent Memory in Small Collections: A Review of File Hierarchies, Sparse Search, and Vector Search

Mahipal J. Suchar[1], Ami M. Mehta[2]

[1] *Research Scholar, Computer Engineering Department, Dr. Subhash University, Junagadh, India*

[2] *Assistant Professor, Computer Engineering Department, Dr. Subhash University, Junagadh, India*

ABSTRACT

Persistent AI agents need to retrieve past information from an external memory store. For a small, human-maintained collection, the choice among folders, keyword search and dense embeddings is not settled by results from large retrieval benchmarks. This structured narrative review synthesizes 61 research and survey works on retrieval-augmented generation, indexing, agent memory, hierarchical and graph retrieval, evaluation, and local deployment. It explains where each method helps, identifies differences between organizing files and ranking them, and examines how the studies were evaluated. Dense retrieval can bridge different wording; explicit file paths support inspection; keyword search offers a strong baseline; and hybrid systems combine these properties. However, published outcomes use different data, models, and hardware, so their scores cannot be directly compared for a small memory store. The reviewed work supports a test of source accuracy, unsupported-answer rejection, updates, latency, memory use, and human maintenance under shared conditions. This article presents a literature synthesis and a reproducible comparison design; it reports no new experiment.

Keywords: retrieval-augmented generation; agent memory; file hierarchy; vector retrieval; sparse retrieval; low-resource systems

1. Introduction

A language model cannot automatically carry a project decision, a user preference, or a correction from one conversation into another. An agent can persist that information externally and bring back the relevant record when needed. Retrieval-augmented generation (RAG) uses this pattern: a retrieval component selects text, and a generator uses that text to answer [1]. Agent memory extends the pattern because stored facts may change, conflict, become stale, or need to be forgotten.

The common implementation stores text chunks as dense numeric embeddings and searches a vector index. The route is attractive because a query and an answer can have similar meaning even when they share few words [2], [3]. For a personal or small-team store with perhaps tens or hundreds of carefully named documents, other choices deserve attention. Files can be organized by topic and navigated through a table of contents; BM25 can search the same files by exact terms; a vector index can rank their content semantically. These designs differ in indexing, updating, inspection, and failure behavior.

The central question is not whether all vector systems are complex or all files are simple. It is when the added embedding and indexing work yields enough value over clear file organization and a strong lexical baseline. We synthesize work from six related areas, define a fair small-collection comparison, and identify what can and cannot be concluded from current studies. A file system can contain a vector index, and a vector store can retain visible source paths; organization and matching should be analyzed as separate design choices [4], [5]. Figure 1 shows these stages and where a folder route can be used with different ranking methods.

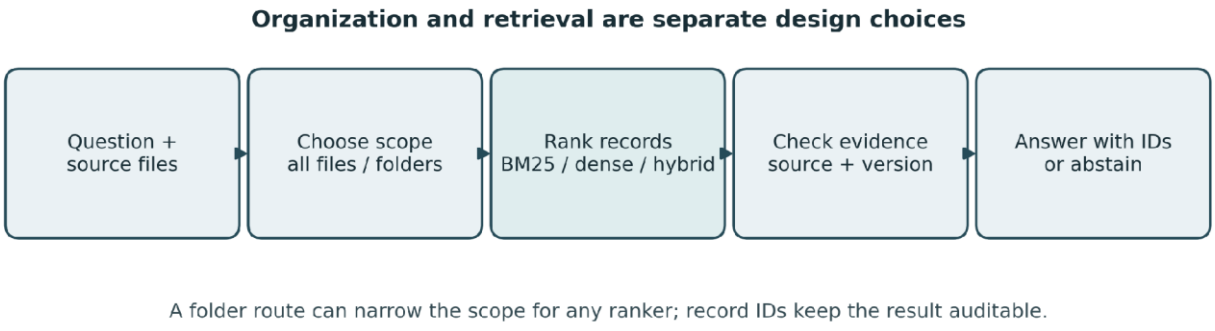


Fig. 1 - A small-memory retrieval workflow. Folder routing sets the scope; lexical, dense, or hybrid matching ranks the records.

2. Review Method

2.1 Review question and source selection

This is a structured narrative review, not a systematic review or statistical meta-analysis. We used targeted searches of author or publisher records in arXiv, ACL Anthology, NeurIPS, ICLR/OpenReview, ACM and relevant journals, followed by citation tracing from RAG, retrieval, and memory surveys. Search concepts included "retrieval-augmented generation", "agent memory", "dense passage retrieval", "vector database", "hierarchical retrieval", "semantic file system", "abstention", "RAG evaluation", and "local RAG". The review was updated through September 2026. A source was retained when it introduced a relevant method, benchmark, evaluated system, or focused synthesis. We excluded vendor posts and textbooks from the research-paper evidence set. This targeted selection may miss relevant studies and does not support claims about exhaustive coverage.

The resulting set contains 61 distinct research or survey papers. Eight are surveys; the other 53 are foundational, conceptual, methodological, system, or benchmark papers. Some are peer-reviewed conference or journal publications, while others are publicly posted preprints. Their peer-review status and measured claims must not be conflated. The search was purposeful, and there is no stored database export, duplicate-removal log, or pre-registered protocol. Thus, the collection is broad but cannot support a claim that every relevant study has been found.

2.2 Extraction and comparison approach

For each paper, we recorded the problem addressed, retrieval or memory mechanism, main reported contribution, and the limit of transfer to a small, curated collection. We grouped studies by the role they play in a RAG system, rather than treating every paper as a direct head-to-head competitor. Tables 2 through 7 give a compact evidence map; the discussion interprets their shared patterns and conflicts. Reported scores from different papers are not pooled, since they use different corpora, query sets, generators, budgets, and hardware. The numerical comparison in this paper is the count of reviewed publications, not a new performance experiment.

Table 1. Scope and boundaries of the review

Item	Definition used here	Why it matters
Persistent memory	Information saved outside the current model context and reused after an interaction.	Includes writing, updating, retrieval and forgetting.
Small collection	A deliberately bounded and human-maintained set of approximately 50–200 documents for the proposed experiment.	It is a study setting, not a universal cutoff for RAG.
File hierarchy	Human-readable records grouped by paths and possibly routed by an index.	The paths organize information; the routing algorithm must be specified separately.

Item	Definition used here	Why it matters
Vector retrieval	Similarity search over learned dense embeddings.	A numeric TF–IDF vector is a sparse lexical representation, not a dense embedding baseline.
Evidence level	Peer-reviewed papers and identifiable preprints, summarized separately.	A preprint claim is a reported result, not independent validation.

Figure 2 describes the selected literature by publication year. The 2024 group contains 26 of the 61 works. These counts describe this targeted set and should not be read as a complete publication trend or a measure of method quality.

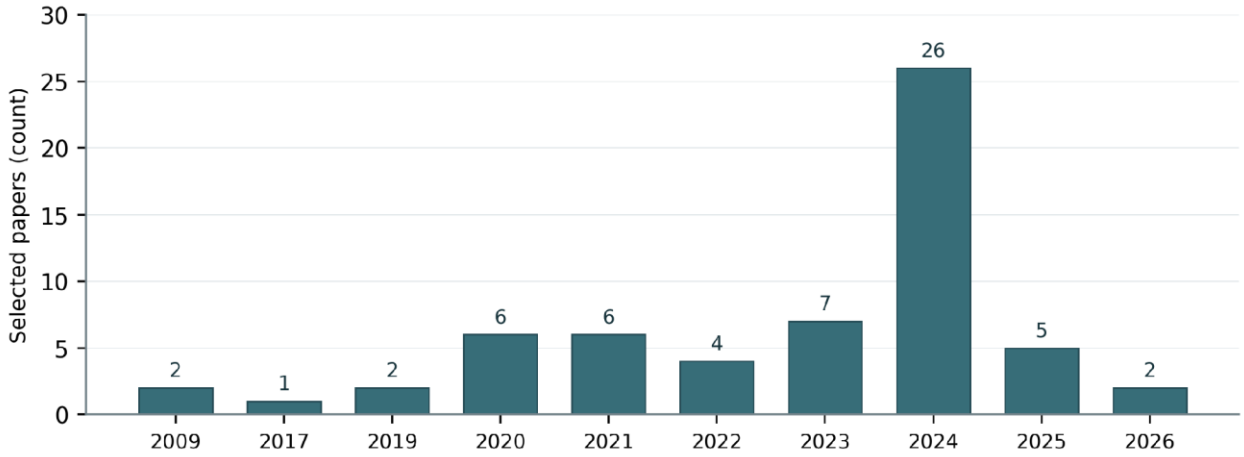


Fig. 2 - Publication years of the 61 selected research and survey works. Preprints are included; bars count papers, not performance outcomes.

3. Foundations of Retrieval Augmented Generation

3.1 Why retrieval helps and where it can fail

The RAG model of Lewis et al. [1] combined external passage retrieval with generation. REALM put retrieval into language-model pretraining [6], RETRO retrieved from a very large text collection [7], and kNN-LM used neighbor lookups as a nonparametric supplement to a language model [8]. These works establish the usefulness of external information, but their training and corpus scales are different from a few dozen personal files. Fusion-in-Decoder demonstrated that a generator can combine evidence from multiple retrieved passages [9]. It also shows that the number of returned passages affects downstream work, so ranking only the first item is not enough.

A fixed retrieve-then-generate pass can miss information needed later in a long answer. FLARE retrieves as generation proceeds [10], Self-RAG learns when to retrieve and critique evidence [11], and Corrective RAG assesses retrieved material and changes course when it appears weak [12]. These designs indicate that retrieval timing and evidence checks can matter independently of where records are stored. A fair small-memory experiment should therefore fix the generation and retrieval policy across baselines or report when it differs.

3.2 Surveys as context rather than experimental proof

Surveys by Y. Gao et al. [13], Fan et al. [14], S. Wu et al. [15], and Gupta et al. [16] organize RAG architectures and applications. Their taxonomies help describe naive, modular, and iterative pipelines but do not answer whether a specific folder tree beats dense retrieval on the same personal memory. Similarly, the broader surveys by Zhao et al. [17] and Pan et al. [18] describe dense retrieval and vector database infrastructure. The survey evidence is useful for selecting baselines, not for transferring a benchmark score to this study setting.

Table 2. RAG foundations and broader research context

Work	Contribution for this review
Lewis et al. [1]	Couples retrieval with generation; establishes why source selection affects answers.
Y. Gao et al. [13]	Organizes naive, advanced and modular RAG; synthesis rather than a small-memory experiment.
Fan et al. [14]	Reviews architectures, training and applications of retrieval-augmented models.
S. Wu et al. [15]	Maps NLP applications and architectures of RAG.
Gupta et al. [16]	Discusses RAG evolution and open design problems.
Jiang et al. [10]	Retrieves again during generation when more evidence is needed.
Borgeaud et al. [7]	Retrieves from a very large external text store during modeling.
Guu et al. [6]	Integrates retrieval during language-model pretraining.
Izacard and Grave [9]	Shows how a generator can combine many retrieved passages.
Khandelwal et al. [8]	Uses nearest-neighbor datastore lookups during language modeling.
Vaswani et al. [19]	Introduces transformer attention that underpins many encoders and LLMs.
Devlin et al. [20]	Provides bidirectional encoder representations used by dense retrievers.
Asai et al. [11]	Lets the generator decide when to retrieve and critique its evidence.
Yan et al. [12]	Evaluates retrieval confidence and changes action when evidence is poor.

4. Retrieval and Indexing Methods

4.1 Sparse lexical search

BM25 ranks documents using query terms and document statistics [21]. It is inexpensive to explain and can be effective for exact names, identifiers, and technical phrases. Its weakness is vocabulary mismatch: a question about “where the invoice went” may not share terms with a note titled “reimbursement receipt”. Learned sparse retrieval, illustrated by SPLADE, expands or reweights terms and can close part of that gap, but it requires model work and should not be called a simple no-model keyword baseline [22]. TF–IDF is also a sparse lexical representation. Representing it as an array does not turn a comparison into a test of dense neural embeddings.

4.2 Learned dense search

Sentence-BERT produces sentence representations suited to similarity comparison [3]. DPR trained question and passage encoders for open-domain

answering and reported gains over a strong BM25 baseline in its task setting [2]. ANCE showed how selecting hard training negatives changes dense retriever quality [23]. Contriever examined unsupervised transfer when labeled examples are unavailable [24]; HyDE uses a generated hypothetical document to improve zero-shot dense retrieval [25]. These results show that "vector retrieval" is not one fixed system. Encoder selection and training can alter results even before the index is chosen.

ColBERT uses token-level late interaction instead of a single vector per passage, improving fine-grained matching at higher indexing cost; ColBERTv2 reduces part of that footprint [26], [27]. A small-store comparison should use a sensible embedding baseline, state its model and version, and avoid generalizing from one weak encoder to all dense retrieval methods. If the store is small, exact cosine comparison is a useful control beside an approximate nearest-neighbor index.

4.3 Indexes and hybrid matching

HNSW and FAISS address efficient nearest-neighbor search, often on collections much larger than the target memory [28], [29], [30]. The vector-database survey covers management concerns beyond ranking, including updates and storage [18]. At a 100-record scale, index construction and service operation may matter more than search speed, but that is a hypothesis to measure. Wang et al. [31] found that combining BM25 and dense scores helped the dense systems and tasks they studied. Reciprocal Rank Fusion offers a simple way to merge ranked lists without calibrating raw scores [32]. A hybrid baseline is therefore important, while its extra components must be counted in resource and maintenance totals.

4.4 How these methods may behave on a small corpus

A small collection changes the relative importance of work done before and after a query. On a large web corpus, an approximate index saves computation on each search. On 100 records, exact comparison against all embeddings may be cheap enough that an approximate index has little opportunity to help, while keeping the index current still takes work. This inference follows from how nearest-neighbor methods operate; it is not a performance result established by this review. Testing exact dense search alongside an indexed version can isolate whether the database layer adds measurable value at this scale [28], [30].

An exact-name question is another useful boundary case. A dense model may place a short identifier near a semantically related but incorrect passage, while BM25 may match the identifier precisely. Conversely, a note about "time off" can answer a question about "leave balance" despite little word overlap; a semantic model may have an advantage there. A hybrid system can keep both signals, but the combination rule and threshold matter. Wang et al. [31] tested score interpolation, while Cormack et al. [32] proposed rank fusion. A future comparison should report these query types separately, retain the retrieved record IDs for error review, and avoid selecting a favorable method using the held-out test questions.

Table 3. Search methods and infrastructure

Work	Contribution for this review
Karpukhin et al. [2]	Demonstrates learned dual-encoder passage search against lexical baselines.
Reimers et al. [3]	Produces useful sentence embeddings through a Siamese encoder.
Malkov et al. [28]	Introduces HNSW approximate-neighbor search; benefits depend on scale and settings.
Johnson et al. [29]	Describes FAISS similarity search, including GPU-oriented large-scale methods.
Thakur et al. [33]	Tests retrieval across heterogeneous datasets; transfer to personal memory needs testing.
Pan et al. [18]	Surveys vector database operations, indexing, management and systems tradeoffs.
Douze et al. [30]	Documents FAISS library designs and operational tradeoffs.
Xiong et al. [23]	Uses hard negatives selected with an ANN index.

Work	Contribution for this review
Khattab and Zaharia [26]	Compares token-level representations rather than one vector per passage.
Santhanam et al. [27]	Reduces storage cost of token-level passage search.
Wang et al. [31]	Finds value from interpolating BM25 and dense scores in studied settings.
Zhao et al. [17]	Synthesizes pretrained-model dense retrieval methods and benchmarks.
Robertson and Zaragoza [21]	Explains BM25, a defensible non-neural lexical baseline.
Cormack et al. [32]	Combines ranked lists without assuming comparable raw scores.
L. Gao et al. [25]	Uses hypothetical document embeddings to improve zero-shot search.
Izacard et al. [24]	Shows a trained unsupervised dense baseline can transfer without labels.
Formal et al. [22]	Learns expanded sparse term weights, complicating a simple sparse/dense dichotomy.

5. Agent Memory and Structure

5.1 What a memory system stores

Memory is more than a pile of passages. Generative Agents stores observations and reflections and retrieves them for planning [34]. Reflexion writes verbal feedback for later attempts [35], while Voyager stores reusable skills as inspectable code [36]. The CoALA framework distinguishes memory and action components [37], and a memory survey classifies records by object, form, and time [38]. A personal file tree may hold facts, decisions, instructions, and experiences, which have different update and retrieval rules. Tests limited to simple factual questions underrepresent an agent's real memory workload.

MemGPT manages limited context using an external memory tier [39]. MemoryBank continually updates records from interactions [40]; Mem0 extracts and updates selective memories [41]; Zep uses temporal relationships in a knowledge graph [42]. A-MEM builds structured, linked notes [43]. These systems differ in generation, extraction, consolidation, and retrieval, so reported overall answer performance cannot isolate whether folders or vectors caused an improvement. Comparing their storage architecture without controlling the rest of the pipeline would be misleading.

5.2 File hierarchy is an organization choice

A file-based memory can use category folders, a top-level index, metadata, short summaries, and explicit source links. This organization helps a human inspect and correct records. Yet incorrect routing can exclude an otherwise relevant file; duplicated facts can diverge; broad queries may require several branches. The LLM-based Semantic File System provides a natural-language file interface but builds semantic indexes with a vector database [4]. It is evidence that a file interface and an embedding-free method are not synonyms. Human time spent curating the tree is a real cost; any comparison should record it.

5.3 Learned hierarchies and graphs

RAPTOR recursively clusters and summarizes passages into a tree, allowing retrieval at different levels of detail [44]. GraphRAG builds entity communities and summaries for broader questions over a corpus [45]. LightRAG combines graph structure and vector representations [5]; HippoRAG uses a graph and propagation for long-term, multi-hop retrieval [46]. MiniRAG focuses on small generators using lightweight graph structure [47]. These results motivate multi-record questions and structural baselines, but none is equivalent to a manually curated Markdown folder tree. Their graph

construction, summarization and embedding costs belong in any practical comparison.

5.4 Memory updates and responsibility for organization

New records are not the only update operation. An agent may learn that a preference changed, receive a correction to an earlier fact, merge duplicate notes, or delete information that should no longer be used. A folder-based system can show the edited source directly, but a summary index that mentions the old value must also be changed. A vector system can store source IDs and timestamps, yet its index must reflect the latest content. MemoryBank, Mem0, Zep, and A-MEM make memory evolution part of their design [40], [41], [42], [43]. Their methods differ, so none proves that a particular storage medium solves stale facts on its own.

The cost of organizing a file tree is easy to overlook. A person may need to choose a category, write an informative title, maintain a top-level index, and resolve records that fit two topics. A model can help with this work but then its routing decisions and token use become part of the system. Graph and summarization methods likewise spend computation to create useful structure. For a fair comparison, the same amount of human description should be available to each baseline as record text or metadata, and any extra editorial work should be reported. Auditability should be assessed with a concrete task, such as how quickly another person can find and correct the source of a wrong answer, instead of asserted from the presence of readable files.

Table 4. Agent memory and structured retrieval systems

Work	Contribution for this review
Fan et al. [47]	Adapts graph retrieval to limited generator capability, a different resource constraint.
Sarthi et al. [44]	Clusters and summarizes passages as a retrieval tree, with indexing overhead.
Guo et al. [5]	Combines graph structure and vector representations; not an embedding-free folder method.
Edge et al. [45]	Uses communities to support broad corpus-level questions.
Shi et al. [4]	Provides prompt-driven file operations while retaining a vector-backed semantic index.
Park et al. [34]	Combines experience storage, reflection and retrieval for social agents.
Shinn et al. [35]	Stores verbal feedback for later attempts rather than only factual passages.
Packer et al. [39]	Treats finite context and external storage as managed memory tiers.
Wang et al. [36]	Keeps reusable skills as an inspectable code library.
Chhikara et al. [41]	Builds selective long-term memories with extraction and update steps.
Rasmussen et al. [42]	Adds temporal relationships to an agent memory graph.
Gutiérrez et al. [46]	Uses knowledge graphs and propagation for multi-hop recall.
Sumers et al. [37]	Separates agent memory and action components conceptually.

Work	Contribution for this review
Wu et al. [38]	Classifies AI memory across objects, forms and time.
Zhong et al. [40]	MemoryBank stores and updates conversational information.
Xu et al. [43]	Creates structured notes and links them dynamically.

6. Evaluation Evidence

6.1 Retrieval and answer quality are different

A retrieved record can be relevant without containing enough evidence to answer, and a generator can misuse a correct record. BEIR tests search over varied tasks [33]; LoCoMo tests long conversational recall [48]. MultiHop-RAG asks for evidence distributed across documents [49]. LongMemEval adds updating, temporal reasoning and abstention [50]. These benchmarks suggest query categories but do not directly represent a small collection of human-maintained Markdown files. A local test set needs relevance judgments at the record level and independently checked answers.

RAGAS and ARES assess context relevance or answer faithfulness with automated judges [51], [52]. RGB tests robustness to noise, conflicting evidence, information integration and negative cases [53]; RAGChecker separates retrieval and generation diagnostics [54]. Yu et al. [55] surveys evaluation choices. Salemi and Zamani [56] show that ordinary relevance labels may correlate weakly with downstream answer utility. These findings favor a set of metrics over one headline score, plus human review of a sample of generated answers and judge disagreements.

6.2 Unanswerable questions and changing facts

A search method can rank the least irrelevant passage highest even when none supports the user's question. NoMIRACL explicitly assesses whether a model answers only when the retrieved passages contain relevant material [57]. RGB and LongMemEval likewise contain rejection or abstention challenges [53], [50]. A matched comparison of small-file methods should include these questions, decide thresholds using a development set, and report both correct abstention and false answers on held-out questions.

A memory test should also ask which of two conflicting versions is current, whether an obsolete preference is ignored, and whether a deleted record disappears from retrieval. These cases require record identifiers, update timestamps, and an explicit policy for stale information. The policy can be applied to folders, keyword search, and vector retrieval alike. A result without its source path is harder to audit regardless of the ranking method.

6.3 Costs, reproducibility, and long context

Cost should be decomposed into document preparation, indexing, each retrieval query, generation, updates, and human maintenance. Reporting only search latency excludes the embedding step and the agent's choice of route. Tokens should be counted for retrieval prompts, indexing summaries, and generated answers rather than estimated from file sizes. Hardware, local versus hosted models, corpus length, chunking rules, cache state, and repeated timing runs should be disclosed. Lost in the Middle shows that including a passage somewhere in a large context does not guarantee effective use of it [58]. This is a reason to evaluate answers as well as source recall.

6.4 Interpreting a balanced set of measures

Hit@1 asks whether the first result is one of the known relevant records. Recall@k asks what share of all relevant records appears among the first k; it is especially useful when two separate notes are required. Mean reciprocal rank rewards placing the first relevant record near the top but can look strong even when a necessary second record is missing. For a multi-record question, an additional "all evidence retrieved" rate closes that gap. On questions with no supporting record, these answerable-query measures are undefined or unhelpful; report correct abstention and false positive rates on that group separately.

Answer quality needs another layer of judgment. A retrieved passage may discuss the right topic yet contradict the proposed answer, or an answer may mix one supported claim with one invented detail. RAGAS, ARES, RGB, and RAGChecker offer useful diagnostic approaches, but automatic judges can disagree with careful human review [51], [52], [53], [54]. A reproducible study can provide example questions, gold record IDs, a short answer rubric, blinded human ratings for a subset, and examples where systems failed. The unit of comparison should be the same question and same document version

for every method. Aggregate averages then become easier to interpret through per-category results and paired errors.

Table 5. Evaluation and benchmark studies

Work	Contribution for this review
Maharana et al. [48]	Tests long conversational memory and sythesis across sessions.
Es et al. [51]	Introduces reference-free RAG context and answer metrics.
Saad-Falcon et al. [52]	Uses lightweight judges and calibrated assessment of RAG outputs.
Chen et al. [53]	Tests noise, rejection, integration and counterfactual robustness.
Ru et al. [54]	Diagnoses retrieval and generation failures separately.
Yu et al. [55]	Catalogues retrieval and answer evaluation measures.
Salemi et al. [56]	Questions whether conventional retrieval labels predict downstream answer utility.
D. Wu et al. [50]	Tests extraction, multi-session reasoning, time, updates and abstention.
Liu et al. [58]	Shows that placement in a large prompt can affect use of evidence.
Tang and Yang [49]	Pairs multi-hop questions with supporting evidence.
Thakur et al. [57]	Includes cases where retrieved passages cannot answer the question.

7. Local and Low Resource Deployment

The phrase “low resource” covers different bottlenecks. MiniRAG focuses on the limited reasoning ability of a small generator [47]. EACO-RAG considers distributed routing among edge and cloud resources [59]. CUBO engineers a self-contained laptop pipeline under a memory ceiling [60]. vstash combines SQLite full-text search, local vectors, and adaptive fusion [61]. These works broaden the practical choices and show that local does not have to mean folders only. They also use distinct workloads and hardware, so their reported latencies or memory figures cannot be subtracted from one another to estimate the advantage of a file tree.

A small memory collection can avoid a continuously running service, but a dense index may also be local and compact. The resource question is therefore whether each component is necessary for the required accuracy and maintenance pattern. A clean comparison would record peak memory, on-disk size, index build time, query latency, update latency, token use, and manual categorization effort on the same computer.

Table 6. Resource constrained deployment

Work	Contribution for this review
Astrino [60]	Engineers local RAG within a laptop memory budget; larger workload than 100 files.
Steffens [61]	Combines SQLite text search with vector similarity and adaptive fusion.
Li et al. [59]	Routes RAG across edge and cloud tiers; network setting differs from local files.

8. Synthesis Across the Reviewed Literature

8.1 What the evidence supports

Three findings recur. Dense retrievers improve semantic matching in many published settings, but their performance depends strongly on the encoder, training, and test set [2], [24], [25]. Structure can help a system collect evidence spread across a document or corpus, but the supporting studies often use model-generated trees or graphs rather than hand-maintained folders [44], [45], [5]. Finally, current evaluation work shows that useful memory must retrieve the right records, support a faithful answer, handle uncertainty, and cope with changed facts [53], [54], [50]. None of these findings establishes a universal winner for a small agent memory.

8.2 Where the evidence is weak or incommensurable

The studies differ in corpus size, document structure, query source, generator, and whether the method creates summaries or labels. Some optimize retrieval on benchmark passages; others judge conversational responses or local software systems. A paper that improves Recall@10 over a large web corpus has not measured human editing time for 100 notes. A folder method that retrieves carefully titled records has not proved it can handle paraphrases or messy real histories. Results must be interpreted within their stated data and baselines. Preprints by Astrino [60] and Steffens [61] provide useful design evidence but deserve independent replication before broad operational claims.

8.3 A conditional reading for practitioners

For a collection dominated by exact project names, version numbers and well-maintained headings, a lexical baseline and clear file paths are plausible starting points. A collection dominated by natural paraphrases and inconsistent wording gives dense embeddings a stronger reason to exist. When questions routinely require facts in different records, hierarchical summaries, graph links, or an iterative retrieval policy may add value. When records change often, update correctness and provenance can matter more than the fastest isolated query. These are design expectations inferred from the cited mechanisms and benchmarks, not conclusions of a shared small-memory trial.

The most useful outcome of a controlled study would therefore be a decision boundary rather than a single winner: what query mix, corpus disorder, update frequency and resource budget justify each additional component? Hybrid retrieval may be a practical choice if both exact identifiers and paraphrases matter, but its extra model, index and tuning steps should be visible in the results. The evidence reviewed here supports that question and the measures needed to answer it; it does not supply a universal numerical threshold.

Table 7. Practical decision questions implied by the evidence

Question	What a comparison should observe	Relevant evidence
Do questions use exact names or paraphrases?	Lexical and semantic hit rates by query type.	Robertson and Zaragoza; Karpukhin; Izacard.
Is information spread across records?	Recall of every needed source and answer integration.	RAPTOR; GraphRAG; MultiHop-RAG.

Question	What a comparison should observe	Relevant evidence
How often do facts change?	Time and errors after insert, edit and delete.	MemoryBank; Mem0; LongMemEval.
Can the system say it does not know?	Correct abstention and false-answer rate.	RGB; NoMIRACL; LongMemEval.
What are the real operating costs?	Indexing, query, tokens, disk, RAM, and human effort.	FAISS; CUBO; vstash.

9. Research Gap and Testable Questions

The gap is a controlled, directly comparable study of a manually navigable file hierarchy, a flat lexical baseline, and a genuine dense embedding index over the same small, evolving memory. The reviewed set contains close relatives—RAPTOR's generated tree, a semantic file interface that uses vectors, graph-based memory, and a local hybrid index—but it does not yield an already controlled answer to this particular comparison. This is a bounded conclusion about the reviewed sources; it does not claim no related experiment exists anywhere.

The study should ask four questions. First, how do the methods differ on exact-term, paraphrase, multi-record and negative queries? Second, what changes when records are added, corrected, moved or deleted? Third, how do source correctness and answer faithfulness differ when the generator and allowed context are held fixed? Fourth, what quality is obtained per unit of setup effort, time, tokens, memory, disk and human maintenance? These questions can produce a conditional recommendation instead of an unsupported rule that one approach always wins.

10. Proposed Comparative Protocol

10.1 Corpus, baselines and controls

Create a versioned collection of roughly 100 short, human-readable records covering project notes, user preferences, decisions, instructions and obsolete versions. A synthetic pilot permits exact answer keys, but a later study should include consented, privacy-safe real records or a realistic anonymized set. Keep document contents identical across methods. Evaluate (A) explicit folder routing with a written route rule and bounded within-folder reading, (B) flat BM25, (C) dense embeddings searched exactly, and (D) the same embeddings in a specified local vector index. A fifth hybrid BM25+dense system can show complementarity. If an LLM chooses folders, count its calls, tokens and failures. Do not name a TF–IDF implementation "vector database RAG".

10.2 Question design and gold judgments

A small pilot can begin with 40 questions across direct lookup, paraphrase, multi-record reasoning, updates and unanswerable cases; a publishable comparative claim needs a larger, independently checked set. Each answerable question should identify all supporting record IDs, not just one convenient document. Construct queries without copying the exact folder names or document headings into every question. Place ambiguous and unsupported questions in both development and final test splits. Have at least two annotators adjudicate a sample of source labels and answers, and document how disagreements were resolved.

10.3 Metrics and analysis

Report Hit@1, Recall@k and mean reciprocal rank for answerable questions, along with success in returning all sources needed for multi-record questions. For negative questions report correct abstention, false positive retrieval and the rate of unsupported final answers. Human raters should check whether each answer statement follows from cited evidence; automated judges can supplement but should not replace this check. Record median and tail query time over repeated runs, peak RAM, index size, indexing time, update time, model calls and tokens. Include confidence intervals or per-question paired outcomes, failure cases, and an ablation that removes folder routing or hybrid fusion. Publish the corpus, query labels, code and configuration where privacy allows.

10.4 Threats to validity

The folder arrangement may encode human knowledge not present in a flat baseline; count the work used to build it. Synthetic documents may have unusually clear names and easy facts. Dense performance is sensitive to the embedding model; BM25 depends on tokenization and language; generation

depends on the chosen LLM and context length. Small query sets give unstable percentage estimates, and tuning a threshold on the test set leaks information. Results from one machine and one memory domain should be reported as local evidence, not a general law about RAG. The comparison is a proposed future experiment, and this review reports no new retrieval or answer scores.

11. Limitations of This Review

The corpus was assembled through targeted searching and citation tracing rather than an exhaustive, registered process. It may omit relevant publications, especially fast-moving preprints after September 2026. Several reviewed works are preprints, and different versions may change authorship or reported results. Source descriptions are deliberately concise and focus on relevance to small persistent memory; the review is not a full technical reproduction of each system. We did not pool benchmark statistics or conduct an independent performance experiment. These limits constrain claims of completeness and establish why the proposed controlled study is needed.

12. Conclusion

For a small persistent AI-agent memory, folders, sparse search, dense embeddings and structural methods answer different needs. Human-readable paths support inspection and editing; lexical methods provide a transparent baseline; learned embeddings handle semantic mismatch; and hybrid or graph methods can help when evidence is distributed. The 61 reviewed works explain these mechanisms and several evaluation pitfalls, but their varied datasets and systems do not determine a winner for a shared small-memory workload. The next defensible contribution is a controlled, reproducible comparison that measures source retrieval, faithful answers, abstention, updates, resource use and human maintenance together.

13. References

1) Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-T., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems, 33, 9459–9474.* https://arxiv.org/abs/2005.11401
2) Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W.-T. (2020). Dense passage retrieval for open-domain question answering. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing.* https://aclanthology.org/2020.emnlp-main.550/
3) Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using siamese BERT-Networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing.* https://aclanthology.org/2019.emnlp-main.398/
4) Shi, Z., Mei, K., Jin, M., Su, Y., Zuo, C., Hua, W., Xu, W., Ren, Y., Liu, Z., Du, M., Deng, D., & Zhang, Y. (2024). *From commands to prompts: LLM-Based semantic file system for AIOS* [Preprint]. arXiv. https://arxiv.org/abs/2410.11843
5) Guo, Z., Xia, L., Yu, Y., Ao, T., & Huang, C. (2024). *LightRAG: Simple and fast retrieval-augmented generation* [Preprint]. arXiv. https://arxiv.org/abs/2410.05779
6) Guu, K., Lee, K., Tung, Z., Pasupat, P., & Chang, M.-W. (2020). *REALM: Retrieval-augmented language model pre-training* [Preprint]. arXiv. https://arxiv.org/abs/2002.08909
7) Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., van den Driessche, G., Lespiau, J.-B., Damoc, B., Clark, A., Casas, D. D. L., Guy, A., Menick, J., Ring, R., Hennigan, T., Huang, S., Maggiore, L., Jones, C., Cassirer, A., . . . Sifre, L. (2022). Improving language models by retrieving from trillions of tokens. *Proceedings of the International Conference on Machine Learning.* https://arxiv.org/abs/2112.04426
8) Khandelwal, U., Levy, O., Jurafsky, D., Zettlemoyer, L., & Lewis, M. (2020). Generalization through memorization: Nearest neighbor language models. *International Conference on Learning Representations.* https://arxiv.org/abs/1911.00172
9) Izacard, G., & Grave, E. (2021). Leveraging passage retrieval with generative models for open domain question answering. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics.* https://aclanthology.org/2021.eacl-main.74/
10) Jiang, Z., Xu, F. F., Gao, L., Sun, Z., Liu, Q., Dwivedi-Yu, J., Yang, Y., Callan, J., & Neubig, G. (2023). Active retrieval augmented generation. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing.* https://aclanthology.org/2023.emnlp-main.495/
11) Asai, A., Wu, Z., Wang, Y., Sil, A., & Hajishirzi, H. (2024). Self-RAG: Learning to retrieve, generate, and critique through self-reflection. *International Conference on Learning Representations.* https://openreview.net/forum?id=hSyW5go0v8
12) Yan, S.-Q., Gu, J.-C., Zhu, Y., & Ling, Z.-H. (2024). *Corrective retrieval augmented generation* [Preprint]. arXiv. https://arxiv.org/abs/2401.15884
13) Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., & Wang, H. (2023). *Retrieval-augmented generation for large language models: A survey* [Preprint]. arXiv. https://arxiv.org/abs/2312.10997
14) Fan, W., Ding, Y., Ning, L., Wang, S., Li, H., Yin, D., Chua, T.-S., & Li, Q. (2024). A survey on RAG meeting LLMs: Towards retrieval-augmented large language models. *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining.* https://arxiv.org/abs/2405.06211
15) Wu, S., Xiong, Y., Cui, Y., Wu, H., Chen, C., Yuan, Y., Huang, L., Liu, X., Kuo, T.-W., Guan, N., & Xue, C. J. (2024). *Retrieval-*

augmented generation for natural language processing: A survey [Preprint]. arXiv. https://arxiv.org/abs/2407.13193

16) Gupta, S., Ranjan, R., & Singh, S. N. (2024). *A comprehensive survey of retrieval-augmented generation (RAG): Evolution, current landscape and future directions* [Preprint]. arXiv. https://arxiv.org/abs/2410.12837
17) Zhao, W. X., Liu, J., Ren, R., & Wen, J.-R. (2022). *Dense text retrieval based on pretrained language models: A survey* [Preprint]. arXiv. https://arxiv.org/abs/2211.14876
18) Pan, J. J., Wang, J., & Li, G. (2024). Survey of vector database management systems. *The VLDB Journal, 33(5), 1591–1615.* https://arxiv.org/abs/2310.14021
19) Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems.* https://arxiv.org/abs/1706.03762
20) Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics.* https://arxiv.org/abs/1810.04805
21) Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval, 3(4), 333–389.* https://doi.org/10.1561/1500000019
22) Formal, T., Piwowarski, B., & Clinchant, S. (2021). SPLADE: Sparse lexical and expansion model for first stage ranking. *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval.* https://doi.org/10.1145/3404835.3463098
23) Xiong, L., Xiong, C., Li, Y., Tang, K.-F., Liu, J., Bennett, P., Ahmed, J., & Overwijk, A. (2021). Approximate nearest neighbor negative contrastive learning for dense text retrieval. *International Conference on Learning Representations.* https://arxiv.org/abs/2007.00808
24) Izacard, G., Caron, M., Hosseini, L., Riedel, S., Bojanowski, P., Joulin, A., & Grave, E. (2022). *Unsupervised dense information retrieval with contrastive learning* [Preprint]. arXiv. https://arxiv.org/abs/2112.09118
25) Gao, L., Ma, X., Lin, J., & Callan, J. (2023). Precise zero-shot dense retrieval without relevance labels. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics.* https://aclanthology.org/2023.acl-long.99/
26) Khattab, O., & Zaharia, M. (2020). ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval.* https://arxiv.org/abs/2004.12832
27) Santhanam, K., Khattab, O., Saad-Falcon, J., Potts, C., & Zaharia, M. (2022). ColBERTv2: Effective and efficient retrieval via lightweight late interaction. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics.* https://arxiv.org/abs/2112.01488
28) Malkov, Y. A., & Yashunin, D. A. (2020). Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* https://arxiv.org/abs/1603.09320
29) Johnson, J., Douze, M., & Jégou, H. (2021). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data.* https://arxiv.org/abs/1702.08734
30) Douze, M., Guzhva, A., Deng, C., Johnson, J., Szilvasy, G., Mazaré, P.-E., Lomeli, M., Hosseini, L., & Jégou, H. (2024). *The Faiss library* [Preprint]. arXiv. https://arxiv.org/abs/2401.08281
31) Wang, S., Zhuang, S., & Zuccon, G. (2021). BERT-Based dense retrievers require interpolation with BM25 for effective passage retrieval. *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval.* https://doi.org/10.1145/3471158.3472233
32) Cormack, G. V., Clarke, C. L. A., & Buettcher, S. (2009). Reciprocal rank fusion outperforms condorcet and individual rank learning methods. *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval.* https://doi.org/10.1145/1571941.1572114
33) Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., & Gurevych, I. (2021). BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. *NeurIPS Datasets and Benchmarks Track.* https://arxiv.org/abs/2104.08663
34) Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior. *Proceedings of the ACM Symposium on User Interface Software and Technology.* https://doi.org/10.1145/3586183.3606763
35) Shinn, N., Cassano, F., Berman, E., Gopinath, A., Narasimhan, K., & Yao, S. (2023). Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems.* https://arxiv.org/abs/2303.11366
36) Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., & Anandkumar, A. (2023). *Voyager: An open-ended embodied agent with large language models* [Preprint]. arXiv. https://arxiv.org/abs/2305.16291
37) Sumers, T. R., Yao, S., Narasimhan, K., & Griffiths, T. L. (2024). Cognitive architectures for language agents. *Transactions on Machine Learning Research.* https://arxiv.org/abs/2309.02427
38) Wu, Y., Liang, S., Zhang, C., Wang, Y., Zhang, Y., Guo, H., Tang, R., & Liu, Y. (2025). *From human memory to AI memory: A survey on memory mechanisms in the era of LLMs* [Preprint]. arXiv. https://arxiv.org/abs/2504.15965
39) Packer, C., Wooders, S., Lin, K., Fang, V., Patil, S. G., Stoica, I., & Gonzalez, J. E. (2023). *MemGPT: Towards LLMs as operating systems* [Preprint]. arXiv. https://arxiv.org/abs/2310.08560
40) Zhong, W., Guo, L., Gao, Q., Ye, H., & Wang, Y. (2024). Enhancing large language models with long-term memory. *Proceedings of the AAAI Conference on Artificial Intelligence.* https://ojs.aaai.org/index.php/AAAI/article/view/29946
41) Chhikara, P., Khant, D., Aryan, S., Singh, T., & Yadav, D. (2025). *Mem0: Building production-ready AI agents with scalable long-term memory* [Preprint]. arXiv. https://arxiv.org/abs/2504.19413

42) Rasmussen, P., Paliychuk, P., Beauvais, T., Ryan, J., & Chalef, D. (2025). *Zep: A temporal knowledge graph architecture for agent memory* [Preprint]. arXiv. https://arxiv.org/abs/2501.13956

43) Xu, W., Liang, Z., Mei, K., Gao, H., Tan, J., & Zhang, Y. (2025). *A-MEM: Agentic memory for LLM agents* [Preprint]. arXiv. https://arxiv.org/abs/2502.12110

44) Sarthi, P., Abdullah, S., Tuli, A., Khanna, S., Goldie, A., & Manning, C. D. (2024). RAPTOR: Recursive abstractive processing for tree-organized retrieval. *International Conference on Learning Representations.* https://arxiv.org/abs/2401.18059

45) Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Metropolitansky, D., Ness, R. O., & Larson, J. (2024). *From local to global: A graph RAG approach to query-focused summarization* [Preprint]. arXiv. https://arxiv.org/abs/2404.16130

46) Gutiérrez, B. J., Shu, Y., Gu, Y., Yasunaga, M., & Su, Y. (2024). HippoRAG: Neurobiologically inspired long-term memory for large language models. *Advances in Neural Information Processing Systems.* https://arxiv.org/abs/2405.14831

47) Fan, T., Wang, J., Ren, X., & Huang, C. (2025). *MiniRAG: Towards extremely simple retrieval-augmented generation* [Preprint]. arXiv. https://arxiv.org/abs/2501.06713

48) Maharana, A., Lee, D.-H., Tulyakov, S., Bansal, M., Barbieri, F., & Fang, Y. (2024). Evaluating very long-term conversational memory of LLM agents. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics.* https://aclanthology.org/2024.acl-long.747/

49) Tang, Y., & Yang, Y. (2024). MultiHop-RAG: Benchmarking retrieval-augmented generation for multi-hop queries. *Conference on Language Modeling.* https://arxiv.org/abs/2401.15391

50) Wu, D., Wang, H., Yu, W., Zhang, Y., Chang, K.-W., & Yu, D. (2024). *LongMemEval: Benchmarking chat assistants on long-term interactive memory* [Preprint]. arXiv. https://arxiv.org/abs/2410.10813

51) Es, S., James, J., Espinosa-Anke, L., & Schockaert, S. (2024). RAGAS: Automated evaluation of retrieval augmented generation. *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations.* https://aclanthology.org/2024.eacl-demo.16/

52) Saad-Falcon, J., Khattab, O., Potts, C., & Zaharia, M. (2024). ARES: An automated evaluation framework for retrieval-augmented generation systems. *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics.* https://aclanthology.org/2024.naacl-long.20/

53) Chen, J., Lin, H., Han, X., & Sun, L. (2024). Benchmarking large language models in retrieval-augmented generation. *Proceedings of the AAAI Conference on Artificial Intelligence.* https://ojs.aaai.org/index.php/AAAI/article/view/29728

54) Ru, D., Qiu, L., Hu, X., Zhang, T., Shi, P., Chang, S., Jiayang, C., Wang, C., Sun, S., Li, H., Zhang, Z., Wang, B., Jiang, J., He, T., Wang, Z., Liu, P., Zhang, Y., & Zhang, Z. (2024). RAGChecker: A fine-grained framework for diagnosing retrieval-augmented generation. *Advances in Neural Information Processing Systems.* https://arxiv.org/abs/2408.08067

55) Yu, H., Gan, A., Zhang, K., Tong, S., Liu, Q., & Liu, Z. (2024). *Evaluation of retrieval-augmented generation: A survey* [Preprint]. arXiv. https://arxiv.org/abs/2405.07437

56) Salemi, A., & Zamani, H. (2024). *Evaluating retrieval quality in retrieval-augmented generation* [Preprint]. arXiv. https://arxiv.org/abs/2404.13781

57) Thakur, N., Bonifacio, L., Zhang, X., Ogundepo, O., Kamalloo, E., Alfonso-Hermelo, D., Li, X., Liu, Q., Chen, B., Rezagholizadeh, M., & Lin, J. (2024). "Knowing when you don't know": A multilingual relevance assessment dataset for robust retrieval-augmented generation. *Findings of the Association for Computational Linguistics: EMNLP 2024.* https://aclanthology.org/2024.findings-emnlp.730/

58) Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., & Liang, P. (2024). Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics.* https://aclanthology.org/2024.tacl-1.9/

59) Li, J., Xu, C., Jia, L., Wang, F., Zhang, C., & Liu, J. (2024). *EACO-RAG: Towards distributed tiered LLM deployment using edge-assisted and collaborative RAG with adaptive knowledge update* [Preprint]. arXiv. https://arxiv.org/abs/2410.20299

60) Astrino, P. (2026). *CUBO: Self-contained retrieval-augmented generation on consumer laptops 10 GB corpora, 16 GB RAM, single-device deployment* [Preprint]. arXiv. https://arxiv.org/abs/2602.03731

61) Steffens, J. (2026). *Vstash: Local-first hybrid retrieval with adaptive fusion for LLM agents* [Preprint]. arXiv. https://arxiv.org/abs/2604.15484