

Agentic Search Workflows for Efficient Large-Scale Document Retrieval

Remigio Sanz

Department of Computer Science, Colorado State University, Fort Collins, CO, USA.
remigosanz@colostate.edu

Chuy Bayon

Department of Computer Science and Engineering, University of Nevada, Reno, Reno, NV, USA.
chuybayon370@unr.edu

Brigid Megaw

Department of Computer Science, Binghamton University, Binghamton, NY, USA.
megaw1981@binghamton.edu

Ermes Giusti

Department of Computer Science, University of Central Florida, Orlando, FL, USA.
ermeswork@ucf.edu

Abstract

Large-scale document retrieval is no longer merely a problem of matching queries to indexes; it is increasingly a socio-technical service in which autonomous or semi-autonomous software agents plan searches, select tools, inspect intermediate evidence, and coordinate with other agents or human operators. This paper examines agentic search workflows as a system-level architecture for efficient retrieval across very large document collections. The central argument is that agentic retrieval should not be understood as a single model or ranking algorithm, but as a layered infrastructure combining index design, query transformation, routing, evidence synthesis, provenance control, and governance. The paper analyzes structural trade-offs among retrieval quality, latency, cost, energy consumption, robustness, fairness, and accountability. It considers dense, sparse, hybrid, late-interaction, and reranking methods, and then situates them within agentic control loops that may invoke multiple retrieval strategies, critique intermediate results, and adapt search plans at runtime. The discussion emphasizes that efficiency in such workflows is multidimensional: it includes compute efficiency, data movement, storage amplification, orchestration overhead, failure recovery, and human oversight. The paper also explores deployment concerns, including multi-tenant infrastructure, observability, policy enforcement, and sustainability. It concludes that agentic search can improve retrieval effectiveness in complex domains, but only if architectural choices are governed by explicit service-level objectives, provenance requirements, and fairness constraints. Without such governance, agentic retrieval risks converting retrieval efficiency into hidden operational costs, opaque decision-making, and uneven access to high-quality information.

Keywords

agentic retrieval, large-scale search, document retrieval, information systems, workflow orchestration, governance, sustainability, fairness.

1. Introduction

Large-scale document retrieval has traditionally been organized around a relatively stable pipeline: documents are ingested, analyzed, indexed, and then matched against user queries using lexical or semantic signals. Foundational information retrieval research established the importance of inverted indexes, term weighting, probabilistic relevance models, and evaluation methodologies that made search systems comparable and improvable [1]. The probabilistic relevance framework associated with BM25 remains a durable baseline because it balances term frequency, document length, and collection statistics in a manner that is computationally tractable at very large scale [2]. However, modern retrieval environments have expanded beyond the classical paradigm. Queries are often underspecified, collections are heterogeneous, and user needs may require synthesis across multiple documents, formats, and jurisdictions.

Neural retrieval methods have changed the technical landscape by allowing queries and documents to be represented in dense vector spaces. Dense passage retrieval demonstrated that learned embeddings can outperform sparse retrieval in open-domain question answering when training data and negative sampling are carefully managed [3]. Retrieval-augmented generation further showed that a language model can condition its output on retrieved evidence, thereby linking search infrastructure directly to generative reasoning [4]. These developments made retrieval not merely a precursor to downstream analysis but an active component of knowledge production. At the same time, they introduced new cost structures, because dense indexes, cross-encoders, and generative models consume substantial memory, compute, and energy.

Agentic search workflows extend this trajectory by placing one or more autonomous or semi-autonomous agents inside the retrieval loop. Instead of issuing a single query and accepting a fixed ranked list, an agent may decompose a task, generate multiple queries, choose among indexes, inspect evidence, request clarification, or delegate subproblems to specialized agents. Such workflows promise greater flexibility for complex tasks, including legal discovery, scientific literature review, enterprise compliance, and emergency response. Yet they also complicate efficiency. A system that can invoke many retrieval tools may achieve better recall, but it may also multiply latency, cost, and failure modes. The central research problem is therefore not whether agentic search can work in principle, but how to design it as an efficient, robust, fair, and sustainable infrastructure.

This paper argues that agentic search workflows should be studied as layered socio-technical systems rather than as isolated model improvements. The analysis proceeds through several dimensions: background in retrieval and agentic architectures, system architecture, efficiency and scalability trade-offs, robustness and evaluation, governance and fairness, and deployment and sustainability. The paper's contribution is a system-level synthesis that connects retrieval algorithms to the institutional conditions under which they operate. In doing so, it treats efficiency as a composite property that includes computational economy, operational reliability, human oversight burden, and environmental impact.

2. Background and Related Work

Classical retrieval systems are optimized for low-latency matching over large collections. Inverted indexes support rapid term lookup, while ranking functions combine lexical overlap with collection statistics [1]. BM25 remains widely deployed because it is interpretable, efficient, and competitive in many domains [2]. Sparse retrieval has also evolved through

learned term weighting and expansion methods, which retain inverted-index compatibility while improving semantic matching. For example, sparse lexical and expansion models can produce enriched representations that support effective retrieval without abandoning the efficiency of sparse indexes [11]. This hybrid trajectory is important for agentic workflows because agents often need fast, cheap first-stage retrieval before expensive reasoning or reranking.

Dense retrieval introduced a different set of trade-offs. Dual-encoder architectures map queries and documents into a shared vector space, enabling approximate nearest neighbor search over embeddings [3]. This approach supports semantic generalization and can handle paraphrases that lexical methods miss. However, dense indexes consume significant memory and require specialized hardware or quantization strategies for large-scale deployment. Hierarchical navigable small world graphs provide one widely used method for approximate nearest neighbor search with favorable recall-latency characteristics [7]. GPU-accelerated similarity search has also made billion-scale retrieval more practical, although it concentrates computational resources and raises infrastructure dependencies [8]. Vector quantization and anisotropic compression further reduce memory pressure, but they introduce approximation errors that must be managed in the overall retrieval workflow [9].

Hybrid and late-interaction models attempt to combine the strengths of lexical and dense retrieval. ColBERT uses contextualized late interaction to preserve token-level matching while enabling efficient retrieval through precomputed representations [10]. Such methods can improve effectiveness in zero-shot and domain-specific settings, but they increase index size and query processing complexity. Reranking models, including BERT-based passage rankers and sequence-to-sequence rankers, can substantially improve precision when applied to a smaller candidate set [12], [13]. The cost of reranking depends on candidate depth, model size, and batching strategy, which means that agentic workflows must decide when additional ranking stages are worth their marginal benefit.

Evaluation infrastructure has also matured. Heterogeneous benchmarks such as BEIR enable zero-shot evaluation across diverse retrieval tasks and domains, exposing the fragility of models that perform well only on a single distribution [6]. Retrieval-augmented generation systems show that retrieval quality directly affects downstream generation, including factuality and attribution [4]. Fusion-in-decoder methods similarly demonstrate that generative models can aggregate evidence from multiple passages, but they also increase inference cost and complicate provenance [5]. These developments motivate agentic control because a single retrieval strategy may not be sufficient for all tasks, yet naively invoking many strategies can be inefficient.

Agentic architectures draw from research on reasoning, tool use, and multi-agent collaboration. Chain-of-thought prompting showed that language models can produce intermediate reasoning steps that improve performance on some tasks [14]. ReAct combined reasoning traces with actions, allowing models to interact with external tools in an interleaved manner [15]. Toolformer explored self-supervised learning of tool invocation, suggesting that models can learn when to call external resources [17]. Multi-agent frameworks such as AutoGen enable conversations among specialized agents that can divide labor and critique one another [18]. Generative agents research illustrates how memory, planning, and reflection can produce complex behavior over time [19]. Within retrieval, multi-role collaborative reranking workflows have been proposed to coordinate several language-model roles for ranking, demonstrating both the promise and the orchestration burden of agentic relevance

judgment [16]. Together, these lines of work indicate that agentic search is a systems problem: it requires coordination, memory, tool interfaces, and accountability mechanisms.

3. Architecture of Agentic Search Workflows

An agentic search workflow can be understood as a control layer that sits above one or more retrieval services. The control layer receives a task, constructs a search plan, selects tools, issues queries, evaluates intermediate evidence, and decides whether to continue, revise, or stop. This architecture differs from a conventional pipeline because control is adaptive rather than fixed. A legal research agent, for example, may begin with a broad lexical query, identify jurisdictional terms in the results, issue narrower queries against a case-law index, and then request a summary from a synthesis module. The same task could be handled by a single dense retrieval call if the query is simple, but the agentic workflow is justified only when task complexity or risk justifies the additional coordination cost.

The first architectural layer is query understanding and decomposition. Agents may rewrite ambiguous queries, expand acronyms, translate natural-language requests into structured filters, or split a compound question into subquestions. This layer benefits from both sparse and dense representations, because lexical matching remains valuable for names, identifiers, and rare terms, while dense matching helps with paraphrases and conceptual similarity [2], [3]. The agent must decide which retrieval modality to use, how many results to request, and whether to invoke reranking. These decisions are not merely model choices; they are infrastructure choices that affect latency, index load, and cost allocation.

The second layer is tool routing and orchestration. In a large enterprise, retrieval tools may include an inverted index, a vector database, a graph store, a document summarization service, and a permission-aware search API. The agentic controller must route requests according to task requirements, access controls, and service-level objectives. Multi-agent systems can assign specialized roles, such as planner, retriever, critic, and synthesizer, but role specialization introduces coordination overhead and potential deadlock [18]. Multi-role collaborative reranking illustrates how multiple language-model roles can improve ranking quality, yet it also shows that the quality gain depends on disciplined aggregation and stopping rules [16]. The orchestration layer must therefore balance expressive control against predictable execution.

The third layer is memory and state management. Agentic search is not stateless; it accumulates queries, retrieved documents, intermediate judgments, and user feedback. Memory can improve coherence across turns, but it can also amplify stale or incorrect evidence. Generative agents research highlights the importance of memory retrieval, reflection, and planning in sustaining long-horizon behavior [19]. In retrieval, memory must be governed by provenance, expiration, and permission constraints. A document retrieved under one user's authorization should not automatically become available to another agent session. This requirement makes identity, access control, and audit logging core architectural concerns rather than peripheral security features.

The fourth layer is evidence synthesis and provenance. Agents may summarize, compare, or rank retrieved documents, but downstream users need to know which sources support which claims. Retrieval-augmented generation can improve factual grounding when retrieval is accurate, but it can also produce fluent errors when retrieval is incomplete or noisy [4]. Tool use and reasoning traces can make the process more transparent, yet they can also create lengthy logs that are difficult to audit [15], [17]. A robust architecture should therefore

separate raw evidence, agent-generated interpretations, and final outputs, preserving links between them. This separation supports reproducibility, error analysis, and regulatory compliance.

4. Efficiency and Scalability Trade-offs

Efficiency in agentic search is not reducible to model latency. It includes index construction, storage, memory bandwidth, network transfer, orchestration overhead, retry behavior, and human review. Dense retrieval may offer strong semantic matching, but it can require large vector indexes and specialized similarity search infrastructure [3], [7], [8]. Compression and quantization reduce memory cost, but they may lower recall and force the agent to retrieve more candidates to compensate [9]. Sparse retrieval is often cheaper at query time, but learned expansion can increase index size and scoring complexity [11]. Late-interaction models may improve effectiveness while increasing storage and computation compared with simpler dual encoders [10].

The agentic control loop adds another layer of cost. Each reasoning step, tool call, and critique consumes tokens or compute, and multi-agent workflows can multiply calls through conversation and delegation [18]. A system that invokes five retrieval tools for every query may improve recall, but it may also create redundant work and overload shared infrastructure. Efficiency therefore depends on routing policies that estimate the marginal value of additional actions. For simple factoid queries, a lexical or dense first-stage retrieval followed by a lightweight reranker may be sufficient [2], [13]. For complex investigations, multi-query decomposition and multi-role reranking may be justified [16]. The system should learn or enforce thresholds based on task type, confidence, and service-level objectives.

Scalability also depends on the division of labor between online and offline computation. Precomputed embeddings, compressed indexes, and cached query expansions can reduce online latency, but they require periodic refresh and storage management. Reranking models can be batched or distilled to control cost [12], [13]. Approximate nearest neighbor methods trade recall for speed, and the acceptable trade-off depends on the risk of missing relevant documents [7], [9]. In high-stakes domains such as medicine or law, lower recall may be unacceptable, which means that agentic workflows may need to escalate to exhaustive search or human review. In low-stakes exploratory search, approximate retrieval may be sufficient. Efficiency is therefore not a universal constant but a policy decision about acceptable risk.

Another scalability challenge is data movement. Large-scale retrieval often distributes indexes across nodes, and agentic workflows may issue queries to multiple shards or services. Network latency and cross-service serialization can dominate computation, especially when agents iteratively refine queries. Caching intermediate results can help, but cache invalidation becomes complex when documents are updated frequently or permissions change. Observability tools must track not only latency and error rates but also retrieval depth, reranking depth, agent step count, and token consumption. Without such telemetry, operators cannot distinguish between necessary complexity and accidental inefficiency.

Finally, efficiency must account for the human in the loop. Agents may reduce cognitive load by filtering and summarizing evidence, but they may also increase verification burden if provenance is weak. A system that returns a concise answer without traceable sources may save time initially but create costly rework later. Conversely, a system that returns too many agent traces may overwhelm users. The design goal is not maximal automation but calibrated automation, where the agent handles routine retrieval and synthesis while preserving human

judgment for ambiguous, consequential, or policy-sensitive decisions. This calibration links efficiency to governance, because the acceptable level of automation depends on accountability requirements.

5. Robustness and Evaluation

Robustness in agentic search refers to the system's ability to maintain acceptable performance under distribution shift, adversarial inputs, tool failures, and uncertain permissions. Retrieval models often degrade when evaluated outside their training domain, as shown by heterogeneous benchmarks such as BEIR [6]. Agentic workflows can mitigate this by selecting different tools or query formulations, but they can also propagate errors if the controller trusts flawed intermediate evidence. A robust system needs fallback strategies, confidence estimation, and graceful degradation. If a vector database is unavailable, the workflow might fall back to a lexical index [2]. If a reranker times out, the system might return first-stage results with a clear indication of reduced confidence [12].

Evaluation must therefore move beyond single-turn relevance metrics. Agentic retrieval should be assessed on task success, evidence coverage, provenance quality, latency, cost, and failure recovery. Multi-role reranking and multi-agent collaboration require evaluation of coordination quality, not just final ranking accuracy [16], [18]. Tool-use frameworks introduce questions about when tools are helpful and when they introduce noise [17]. Reasoning traces can improve interpretability, but they may also be post hoc rationalizations rather than faithful explanations [14]. Evaluation should include adversarial cases, such as queries designed to trigger excessive tool calls, poisoned documents, or prompt injection through retrieved content.

Robustness also depends on data quality and index hygiene. Duplicate documents, outdated versions, and conflicting sources can confuse both retrieval models and agents. In enterprise settings, permission-aware retrieval is essential because an agent that retrieves unauthorized content may create legal and ethical exposure. Provenance metadata, versioning, and access control must be maintained throughout the workflow. Retrieval-augmented generation can improve factual grounding, but only if the retrieved evidence is trustworthy and complete [4]. Agentic synthesis should not obscure source limitations; it should expose uncertainty and point users to the underlying documents.

Another dimension of robustness is operational resilience. Large-scale search systems must handle traffic spikes, node failures, and partial index outages. Agentic workflows may be more fragile than monolithic pipelines because they depend on multiple services and model endpoints. Circuit breakers, timeouts, retries with backoff, and idempotent tool interfaces are necessary but insufficient if the agent's plan is not aware of degraded service states. The controller should receive health signals from retrieval services and adjust its plan accordingly. This requirement suggests that agentic search architectures should integrate with standard observability and service mesh infrastructure rather than operate as isolated applications.

6. Governance, Fairness, and Policy

Governance of agentic search involves decisions about what documents may be retrieved, how results may be used, who is accountable for errors, and how affected parties can contest outcomes. These questions become more salient as retrieval systems influence hiring, lending, healthcare, education, and public services. Foundation models and retrieval-augmented systems can reproduce and amplify biases present in data and ranking objectives [20], [21]. An agentic workflow may compound these risks by combining multiple models, each with its

own biases and failure modes. Fairness cannot be treated as a single metric; it requires attention to representation, access, error distribution, and procedural justice.

Policy implications include transparency, auditability, and data minimization. Users may need to know whether an answer was generated by an agent, which sources were consulted, and whether human review occurred. Regulators may require logs of automated decisions, especially in domains with legal or financial consequences. Provenance mechanisms can support accountability, but they must be designed to avoid exposing sensitive information. Access control must be enforced at retrieval time, not merely at presentation time, because agents may otherwise internalize unauthorized content into summaries or memory. Multi-agent systems add further complexity because responsibility may be distributed across planners, retrievers, critics, and synthesizers [18].

Fairness also concerns the distribution of efficiency gains. If agentic search is expensive to build and operate, it may concentrate in well-resourced organizations, leaving smaller institutions with less capable retrieval infrastructure. This disparity can affect scientific research, legal representation, and public access to information. Sustainable and equitable deployment may require shared infrastructure, open benchmarks, and procurement policies that prioritize interoperability and auditability. Evaluation datasets such as BEIR provide a foundation for comparison, but they do not resolve questions of local relevance, language coverage, or cultural context [6]. Governance should therefore include participatory mechanisms that allow affected communities to shape retrieval objectives and redress pathways.

Finally, governance must address misuse and adversarial behavior. Agentic systems can be used to automate surveillance, generate misleading citations, or overwhelm public information systems with synthetic content. Retrieval systems may be poisoned with documents designed to manipulate agent reasoning. Defenses include source verification, rate limiting, anomaly detection, and human review for high-stakes outputs. The stochastic parrots critique of large language models remains relevant because scale alone does not guarantee truthfulness or social responsibility [21]. Agentic search should be designed with contestability in mind, so that users can challenge results, request alternative sources, and understand the limits of automated judgment.

7. Deployment and Sustainability

Deployment of agentic search workflows requires infrastructure that can support both retrieval and orchestration at scale. This includes storage for documents and indexes, compute for embedding and reranking, network capacity for distributed search, and monitoring for agent behavior. Cloud-native deployment offers elasticity, but it can also create lock-in and unpredictable costs. On-premises deployment may provide greater control over data governance, but it requires specialized operations. Hybrid architectures may combine local indexes for sensitive data with cloud services for elastic computation. The choice depends on regulatory constraints, budget, latency requirements, and environmental commitments.

Sustainability is an increasingly important consideration. Large language models and neural retrieval systems consume substantial energy during training and inference [22]. While retrieval can be more efficient than relying solely on a model's parametric memory, agentic workflows can multiply inference calls if they are not carefully managed. Multi-agent collaboration, iterative critique, and reranking can improve quality, but they also increase token consumption and server utilization [16], [18]. Sustainable design should therefore

include energy-aware routing, caching, model distillation, and the use of smaller models for routine tasks. It should also consider the embodied carbon of hardware and the lifecycle costs of frequent index rebuilding.

Operational sustainability also includes maintainability. Agentic workflows that depend on many prompts, tools, and models can become brittle over time. Changes in one component may alter behavior elsewhere, making regression testing difficult. Documentation, versioning, and continuous evaluation are essential. Retrieval indexes must be updated as documents change, and embeddings may need regeneration when models are upgraded. These processes consume resources and require coordination across teams. A sustainable architecture should favor modular interfaces, reproducible configurations, and clear ownership boundaries.

Finally, deployment must account for human factors. Operators need dashboards that show retrieval quality, latency, cost, fairness indicators, and error rates. Users need interfaces that reveal uncertainty and provenance without overwhelming them. Policymakers need mechanisms for oversight that do not require deep technical expertise. The success of agentic search will depend not only on algorithmic advances but also on institutional capacity to govern complex automated systems. As foundation models become more capable, the retrieval layer may become the primary site where accountability is enforced, because it determines which evidence enters the reasoning process [20]. Investing in robust, fair, and sustainable retrieval infrastructure is therefore a precondition for responsible agentic systems.

8. Conclusion

Agentic search workflows offer a promising architecture for large-scale document retrieval because they can adapt to task complexity, combine multiple retrieval modalities, and synthesize evidence across heterogeneous sources. Sparse, dense, hybrid, late-interaction, and reranking methods each provide distinct trade-offs in effectiveness, latency, storage, and cost [2], [3], [10], [11], [12]. Agentic control can improve task performance by decomposing queries, routing to appropriate tools, and coordinating multiple roles for critique and synthesis [15], [16], [18]. However, these benefits are not automatic. Without disciplined orchestration, agentic retrieval can multiply cost, obscure provenance, and create new failure modes.

The system-level conclusion is that efficiency must be defined broadly. It includes computational efficiency, operational resilience, human oversight burden, and environmental impact. Governance must be built into the architecture through access control, provenance, auditability, and contestability. Fairness must be treated as a design constraint rather than a post hoc audit. Sustainability requires energy-aware routing and maintainable infrastructure. The most effective agentic search systems will likely be those that combine adaptive reasoning with strong retrieval engineering, transparent evidence chains, and explicit service-level objectives. Such systems can support complex knowledge work while preserving accountability and equitable access to information.

References

1. Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.
2. Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4), 333–389.
3. Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W. (2020). Dense passage retrieval for open-domain question answering. In *Proceedings of*

the 2020 Conference on Empirical Methods in Natural Language Processing (pp. 6769–6781). Association for Computational Linguistics.

4. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems* 33 (pp. 9459–9474).
5. Izacard, G., & Grave, E. (2021). Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 874–880). Association for Computational Linguistics.
6. Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., & Gurevych, I. (2021). BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*.
7. 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*, 42(4), 824–836.
8. Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3), 535–547.
9. Guo, R., Sun, P., Lindgren, E., Geng, Q., Simcha, D., Chern, F., & Kumar, S. (2020). Accelerating large-scale inference with anisotropic vector quantization. In *Proceedings of the 37th International Conference on Machine Learning* (pp. 3887–3896). PMLR.
10. Khattab, O., & Zaharia, M. (2020). ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 39–48). Association for Computing Machinery.
11. Formal, T., Lassance, C., Piwowarski, B., & Clinchant, S. (2021). SPLADE v2: Sparse lexical and expansion model for information retrieval. *arXiv preprint arXiv:2109.10086*.
12. Nogueira, R., & Cho, K. (2019). Passage re-ranking with BERT. *arXiv preprint arXiv:1901.04085*.
13. Nogueira, R., Jiang, Z., Pradeep, R., & Lin, J. (2020). Document ranking with a pretrained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020* (pp. 708–718). Association for Computational Linguistics.
14. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems* 35 (pp. 24824–24837).
15. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*.
16. Jin, C., Peng, H., Zhang, A., Chen, N., Zhao, J., Xie, X., ... & Metaxas, D. N. (2025, May). Rankflow: A multi-role collaborative reranking workflow utilizing large language

models. In *Companion Proceedings of the ACM on Web Conference 2025* (pp. 2484-2493).

17. Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., & Scialom, T. (2023). Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems* 36.
18. Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., Jiang, L., Zhang, X., Zhang, S., Liu, J., Awadallah, A. H., White, R. W., Burger, D., & Wang, C. (2023). AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*.
19. 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. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. Association for Computing Machinery.
20. Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., ... & Liang, P. (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.
21. Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (pp. 610–623). Association for Computing Machinery.
22. Schwartz, R., Dodge, J., Smith, N. A., & Etzioni, O. (2020). Green AI. *Communications of the ACM*, 63(12), 54–63.