

Retrieval-Augmented Generation for Human Resource Decision Support: A Critical Review and Research Gap Analysis

Ashwini Bharat Phalke¹, Dr. Vinodkumar Bhutnal²

^{1,2}Computer Science Engineering(AI),
Pimpri Chinchwad University, Sate Maval.

Abstract

Human resource (HR) departments hold much of their most valuable knowledge in unstructured documents such as resumes, skill matrices, appraisal notes and project briefs. Locating the right employee for a given project from this material is slow, inconsistent and increasingly impractical as repositories grow. Large language models (llms) offer a natural-language interface to such data, but used alone they cannot see private organizational records and are prone to fabricating plausible but unsupported statements. Retrieval-Augmented Generation (RAG) addresses both problems by grounding a generative model in passages retrieved from a trusted corpus at query time. This paper reviews the literature that underpins an HR chatbot built on RAG and the Gemini model family. It covers the evolution of person–job fit research, the foundations of dense retrieval and RAG, document chunking and vector indexing, known failure modes of RAG pipelines, evaluation methodology, and the fairness and privacy obligations that apply to hiring-related systems. The review identifies a gap: published RAG work concentrates on open-domain and scientific question answering, while few studies examine retrieval over heterogeneous HR records for candidate recommendation. A set of design implications and open research questions is derived for medium and large organizations, and a staged system design for HR project-assignment decision support is proposed together with an evaluation plan.

Keywords— Retrieval-Augmented Generation, Large Language Models, Human Resource Analytics, Person–Job Fit, Vector Databases, Semantic Search, Gemini, Conversational Agents.

1. Introduction

Talent allocation is a recurring and consequential task in any organization. When a new project is approved, a manager must decide who among hundreds or thousands of employees has the right mix of technical skills, domain experience, certifications and availability. The evidence needed for that decision is scattered across resumes in several formats, internal profile pages, training records and earlier project reports. Searching it by hand, or with keyword filters over a database, tends to miss candidates whose experience is described in different words from those used in the query.

Recent progress in language modelling suggests a different approach. The Transformer architecture [1] made it practical to train very large models, and the few-shot behavior of models such as GPT-3 [2] showed that a single model could follow instructions across many tasks. Yet these models carry two

limitations that matter for enterprise use. First, their knowledge is frozen at training time and excludes confidential company data. Second, they can generate fluent text that is not supported by any source, a behavior broadly termed hallucination [3]. For an HR assistant that recommends people for roles, an unsupported claim about someone's certification is not a harmless error.

Retrieval-Augmented Generation, introduced by Lewis et al. [4], combines a retriever that selects relevant passages from an external corpus with a generator that conditions its answer on those passages. In an HR setting, the corpus would consist of resumes, profiles and project documents, and the generator could be a modern LLM such as Gemini [5]. The generator is expected to answer from the retrieved evidence rather than from its parametric memory alone.

This paper surveys the research relevant to building such a system. Section II reviews computational approaches to person–job matching. Section III covers the technical foundations of RAG. Section IV discusses document processing, embeddings and vector indexing. Section V examines reported weaknesses of RAG pipelines. Section VI considers evaluation. Section VII addresses fairness, privacy and governance. Section VIII synthesizes the findings, states the research gap and draws design implications. Section IX turns those implications into a proposed system for project-assignment decision support, and Section X concludes.

2. Person–Job Fit and Automated Talent Matching

Person–job fit, the problem of measuring how well a candidate's qualifications correspond to the requirements of a position, has been studied for years within recruitment analytics. Early computational treatments relied on keyword overlap or hand-built rules, which are brittle when the same skill is phrased in several ways. Qin et al. [6] proposed an ability-aware neural network that represents job requirements and candidate experiences at the word level using recurrent networks, then applies hierarchical attention to determine which requirement each past experience satisfies. The authors argue that this design reduces dependence on manual inspection by HR experts and produces more interpretable matching results. A later journal extension [7] introduced a topic-based variant and a refinement step that uses historical recruitment records, and showed how the same framework could support talent sourcing and job recommendation.

These models share two characteristics. They are trained on large volumes of historical application outcomes, and they output a match score rather than a natural-language explanation. Both properties limit their direct use inside a company that wants to query its own employees conversationally. Supervised matching models need labelled outcomes, which many organizations do not possess for internal staffing. They also tend to be opaque to the end user, who may need to know why a person was suggested. A retrieval-based assistant that cites the specific resume passages supporting a recommendation addresses the second concern directly, and it needs no task-specific training data for the first.

3. Foundations of Retrieval-Augmented Generation

A. From Sparse Matching to Dense Retrieval

Classical retrieval ranks documents by lexical overlap. The probabilistic relevance framework and its BM25 scoring function [9] remain strong baselines because they are fast and need no training. Their

weakness is vocabulary mismatch: a query for "cloud infrastructure engineer" may not match a resume that says "AWS solutions architect." Contextual language models such as BERT [8] made it possible to encode text into dense vectors in which semantic similarity corresponds to geometric closeness. Karpukhin et al. [10] trained a dual-encoder retriever for open-domain question answering and reported that learned dense representations outperformed a strong BM25 baseline in top-passage accuracy on several benchmarks. Sentence-BERT [11] adapted BERT with a Siamese training scheme so that sentence embeddings could be compared efficiently by cosine similarity, which is the practical basis of many present-day semantic search tools.

B. Retrieval-Augmented Language Models

The idea of coupling a language model with a retriever developed along several lines. REALM [12] integrated a latent retriever into language model pre-training so that the model learns to fetch documents that help it predict masked text. Fusion-in-Decoder [13] encoded each retrieved passage independently and let the decoder attend across all of them, improving open-domain question answering. RETRO [14] showed that conditioning on chunks retrieved from a very large text database could let a comparatively small model approach the performance of much larger ones. The RAG framework of Lewis et al. [4] paired a dense retriever with a sequence-to-sequence generator and demonstrated gains on knowledge-intensive tasks, while also noting that the knowledge source can be updated without retraining the generator. This last property is central for HR use, because employee records change continually.

Gao et al. [15] survey the field and organize it into three paradigms: a naive pipeline of indexing, retrieval and generation; an advanced pipeline that adds pre-retrieval and post-retrieval steps such as query rewriting and re-ranking; and a modular paradigm in which components can be recombined. Self-RAG [16] goes further by training a model to decide when retrieval is needed and to critique its own output using special reflection tokens. These developments suggest that the simple retrieve-then-read structure proposed for the HR chatbot is a reasonable baseline but leaves room for refinement.

C. Role of the Generator

In the proposed work the generator is Gemini, a family of multimodal models designed for reasoning over long inputs [5]. A large context window is useful in RAG because more retrieved material can be supplied, but, as discussed in Section V, simply supplying more text does not guarantee that the model will use it well.

4. Document Processing, Embeddings and Vector Indexing

A. Ingestion and Chunking

HR documents vary widely in structure. Resumes may be single-column or multi-column PDFs, profiles may be tables, and project briefs may be free prose. Text extraction quality therefore sets an upper bound on everything downstream. After extraction, documents are divided into chunks that fit an embedding model's input and that ideally correspond to a coherent unit of meaning. Barnett et al. [20] report that retrieval problems often originate in this stage and in the choice of what to index, and they list chunking and metadata strategy among the design decisions that evolve during operation. For resumes, a practical consequence is that splitting a document mid-section can separate a job title from the skills listed beneath it; section-aware splitting and attaching an employee identifier to each chunk are common mitigations.

B. Embedding Models

Embedding models determine how faithfully a skill phrase, a certification name or a project description is placed in vector space. General-purpose sentence encoders derived from the approach in [11] work reasonably across domains, but terminology specific to a company, such as internal tool names or role codes, may be poorly represented. Hybrid retrieval, which combines dense similarity with lexical scoring [9], [10], is a widely discussed remedy because exact matches on certification identifiers or product names are cheap to capture with sparse methods.

C. Vector Databases and Approximate Search

Exact nearest-neighbor search becomes costly as collections grow. Johnson et al. [17] describe GPU-based similarity search libraries capable of handling billion-scale collections, and graph-based indexes such as Hierarchical Navigable Small World (HNSW) [18] provide logarithmic-scaling approximate search with high recall. For an HR repository containing thousands of employees and tens of thousands of chunks, either approach is more than adequate; the more pressing engineering concern is metadata filtering, for example restricting retrieval to a department or availability window, which pure similarity search does not provide.

5. Known Limitations and Failure Modes

A RAG system reduces but does not eliminate error. Barnett et al. [20] analysed three case studies in research, education and biomedical settings and catalogued seven failure points, including missing content, relevant documents that fail to rank highly enough, and answers that are produced but not in the form requested. They conclude that robustness is not designed in at the start but develops through operation, and that validation is realistically possible only on a running system.

A second line of work concerns how models use the context they receive. Liu et al. [19] found that performance on multi-document question answering and key-value retrieval degrades when relevant information sits in the middle of a long input, with the best results when it appears at the beginning or end. For a recommender that lists several candidate employees, this suggests that retrieval should keep the context short and well ranked instead of filling the window.

Chen et al. [21] evaluated several LLMs on four abilities required of RAG: noise robustness, negative rejection, information integration and counterfactual robustness. They observed that models can be distracted by irrelevant passages and may fail to decline when the corpus lacks an answer. In the HR scenario, negative rejection matters: when no employee satisfies a requirement, the assistant should say so rather than propose the nearest plausible profile. The broader hallucination literature [3] provides a taxonomy that is useful for categorizing such errors as intrinsic (contradicting the source) or extrinsic (unsupported by it).

6. Evaluating Rag Systems

Evaluating a RAG system requires separating retrieval quality from generation quality. Es et al. [22] proposed RAGAs, a framework offering reference-free metrics that assess whether retrieved context is relevant and focused, whether the answer is faithful to that context, and whether the answer addresses the question. Because HR teams rarely have gold-standard answer sets, such reference-free measures are

attractive for early-stage assessment, although they depend on an LLM judge and should be calibrated against human review.

For retrieval, standard information-retrieval measures such as recall at k and mean reciprocal rank apply directly if a set of queries with known suitable employees is constructed. For the end-to-end task, a staffing-specific measure is needed: agreement between the chatbot's shortlist and a panel of HR experts, together with time taken to produce a shortlist. This mirrors the stated objectives of reducing manual effort and improving selection accuracy, and it is consistent with the person–job fit literature, which also evaluates against recruiter judgments [6], [7].

7. Fairness, Privacy and Governance

Systems that influence hiring or allocation decisions attract regulatory and ethical scrutiny. Bogen and Rieke [23] examined hiring algorithms across the recruitment pipeline and concluded that automated tools can reproduce or amplify existing inequities unless they are designed and audited with that risk in mind. Raghavan et al. [24] analysed claims made by vendors of algorithmic hiring tools about bias mitigation and highlighted how little is publicly verifiable about validation practices. A RAG-based assistant inherits any bias present in resumes and in the language model, and it can also introduce bias through retrieval, for example by favouring candidates whose documents are longer or written in a particular style.

Privacy is a second concern. Resumes and appraisal records are personal data. In India, the Digital Personal Data Protection Act, 2023 [25] sets obligations on those who process digital personal data, including purpose limitation and security safeguards. Deployments that send retrieved passages to an externally hosted model must therefore consider data minimization, access control and contractual terms with the model provider. Retaining a human decision-maker, and displaying the retrieved evidence alongside each recommendation, are consistent with the transparency goals raised in [23] and [24].

8. Synthesis, Research Gap and Design Implications

TABLE I summarizes the main strands of work reviewed and their relevance to an HR chatbot.

TABLE I. Summary of Reviewed Work and Relevance to an HR RAG Chatbot

Theme	Representative works	Contribution	Relevance to HR chatbot
Person–job fit	[6], [7]	Neural matching of resumes to job requirements	Establishes task; needs labelled data and gives scores without explanations
Dense retrieval	[10], [11]	Learned semantic embeddings	Handles vocabulary mismatch in skill descriptions
RAG paradigms	[4], [12]–[16]	Grounding generation in retrieved text	Core architecture; supports updating without retraining

Theme	Representative works	Contribution	Relevance to HR chatbot
Indexing	[17], [18]	Scalable approximate search	Supports growth in HR repository
Failure analysis	[3], [19]–[21]	Catalogues of error sources	Guides chunking, ranking and refusal behaviour
Evaluation	[22]	Reference-free RAG metrics	Practical when gold labels are scarce
Fairness and privacy	[23]–[25]	Bias, transparency, data protection	Governs deployment in organizations

Research gap. Three observations emerge. First, person–job fit models [6], [7] are trained for score prediction and do not offer a conversational, evidence-citing interface over an organization's own records. Second, the RAG literature [4], [12]–[16], [20], [21] is dominated by open-domain, scientific or biomedical question answering, and the reviewed sources provide little direct evidence on retrieval over resumes, skill matrices and project briefs together. Third, the evaluation and governance work [22]–[25] has not been combined into a protocol for internal staffing recommendations. These are the areas where an HR RAG chatbot can contribute.

Design implications. The literature supports the following choices for the proposed system. Documents should be chunked along section boundaries and tagged with an employee identifier and document type, in line with the failure analysis in [20]. Retrieval should combine dense and lexical signals [9], [10], with metadata filters and a small number of ranked chunks passed to the generator in light of [19]. The generator prompt should require answers drawn only from supplied context and permit an explicit "no suitable candidate found" response, addressing the negative-rejection weakness reported in [21]. Each recommendation should cite its source passages so that an HR professional can verify it. Evaluation should combine retrieval metrics, reference-free faithfulness measures [22] and expert agreement. Governance should include access control, data minimization and human sign-off [23]–[25].

Open questions. (i) How much does hybrid retrieval improve recall of skills expressed in non-standard wording, compared with dense retrieval alone? (ii) Which chunking strategy best preserves the link between role, skill and duration in resumes? (iii) How should multiple partially matching candidates be ranked when requirements conflict? (iv) Can retrieval-stage bias be measured and reduced without lowering relevance?

9. Proposed System for Hr Project-Assignment Decision Support

This section converts the design implications of Section VIII into a concrete system. It is a proposal derived from the reviewed literature, not a reported implementation, and its choices should be validated experimentally.

A. Design Principles

The system should act as decision support, not as an automatic assigner. Four principles follow from the reviewed work. Recommendations must be grounded in retrieved evidence so that an HR professional can

verify them [4], [20]. The model should be allowed to answer that no suitable candidate exists [21]. Numerical ranking should be computed by transparent rules from extracted facts, with the language model used to interpret and explain, because free-text generation is a weak basis for consistent scoring [3]. A human makes the final assignment, with audit records kept [23]–[25].

B. Processing Pipeline

1. **Ingestion.** Collect resumes, employee profiles, skill records, certification lists and past project reports. Convert each file to clean text and record its employee identifier, document type and date.
2. **Chunking with metadata.** Split along section boundaries (summary, experience, skills, certifications, projects). Attach metadata such as employee ID, department, role and availability, so that retrieval can be filtered as well as ranked [20].
3. **Embedding and indexing.** Encode chunks with a sentence-embedding model [11] and store them in a vector index supported by approximate nearest-neighbour search [17], [18]. Keep a parallel lexical index for exact terms [9].
4. **Requirement extraction.** Parse the project brief into a structured list of required skills, minimum experience, mandatory certifications, domain and duration. HR reviews and edits this list before search begins.
5. **Hybrid retrieval.** Retrieve with both dense similarity [10] and BM25 [9], merge the two ranked lists with reciprocal rank fusion [26], and apply metadata filters. A cross-encoder re-ranker [27] can be added if precision at the top ranks is insufficient.
6. **Context assembly.** Pass only a small number of top-ranked chunks per candidate to the generator, ordered by relevance, in line with the position sensitivity reported in [19].
7. **Grounded generation.** Prompt Gemini [5] to extract, for each candidate, the evidence matching each requirement with a reference to the source chunk, and to mark any requirement as "not evidenced" when support is absent.
8. **Scoring and ranking.** Compute a suitability score from the extracted evidence using the rule in Eq. (1), with weights set by HR for the project.
9. **Presentation and feedback.** Show a ranked shortlist with the matched evidence, gaps and score breakdown. Record HR's accept or reject decision and reasons as feedback for later tuning.

C. Suitability Score

For employee e and project p , the proposed score is a weighted sum of criterion-level match values:

$$S(e,p) = \sum_{i=1}^5 w_i s_i(e,p), \quad \sum_{i=1}^5 w_i = 1, \quad 0 \leq s_i \leq 1 \quad \text{tag}\{1\}$$

Here the five criteria s_1 to s_5 are required-skill coverage, relevant experience, mandatory certifications, domain familiarity from earlier projects, and availability. A mandatory requirement that is not evidenced should act as a hard filter, not merely lower the score. Keeping S outside the language model makes rankings repeatable and lets HR see exactly how weights affect the outcome.

D. Evaluation Plan

TABLE II lists the measures that follow from the literature reviewed in Sections V–VII.

TABLE II. Proposed Evaluation Plan

Aspect	Measure	Basis
Retrieval quality	Recall@k, mean reciprocal rank against HR-labelled relevant chunks	Standard IR practice; [10]
Faithfulness	Share of recommendation statements supported by retrieved text; reference-free RAG metrics	[22]
Unanswerable queries	Rate of correct "no suitable candidate" responses	[21]
Context-position effect	Accuracy when the key chunk is moved within the context	[19]
Decision quality	Overlap between system shortlist and an HR expert panel's shortlist	[6], [7]
Efficiency	Time to produce a shortlist, with and without the system	Project objectives
Fairness	Change in shortlist after masking names and other protected attributes	[23], [24]
Failure analysis	Manual review of errors classified by failure point	[20]

E. Deployment and Governance

The system should run with role-based access so that managers see only records they are entitled to view. Retrieved text sent to an external model should be minimized to what the query needs, and the arrangement with the model provider should be checked against the obligations of the Digital Personal Data Protection Act, 2023 [25]. Every recommendation, the evidence shown and the final human decision should be logged for audit. A pilot on one department with a bounded set of past projects is a reasonable first stage, as the evidence in [20] indicates that robustness grows through operation.

10. Conclusion

The reviewed literature indicates that Retrieval-Augmented Generation is a technically sound basis for an HR assistant. It offers a way to use a powerful language model with confidential records, to keep knowledge current without retraining, and to present evidence for each recommendation. At the same time, the evidence on failure modes, long-context behavior and bias shows that a naive retrieve-and-generate pipeline is insufficient on its own. A responsible design needs careful chunking, hybrid retrieval, constrained prompting, explicit handling of unanswerable queries, systematic evaluation and strong

governance. The remaining gap is empirical: controlled studies of RAG over real HR corpora, measured against expert shortlists, are needed to quantify the benefit in accuracy and time saved.

References

1. A. Vaswani et al., "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.
2. T. B. Brown et al., "Language models are few-shot learners," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 1877–1901.
3. Z. Ji et al., "Survey of hallucination in natural language generation," *ACM Comput. Surv.*, vol. 55, no. 12, Art. no. 248, pp. 1–38, 2023.
4. P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 9459–9474.
5. Gemini Team, Google, "Gemini: A family of highly capable multimodal models," *arXiv:2312.11805*, 2023.
6. C. Qin, H. Zhu, T. Xu, C. Zhu, L. Jiang, E. Chen, and H. Xiong, "Enhancing person-job fit for talent recruitment: An ability-aware neural network approach," in *Proc. 41st Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, Ann Arbor, MI, USA, 2018, doi: 10.1145/3209978.3210025.
7. C. Qin, H. Zhu, T. Xu, C. Zhu, C. Ma, E. Chen, and H. Xiong, "An enhanced neural network approach to person-job fit in talent recruitment," *ACM Trans. Inf. Syst.*, vol. 38, no. 2, Art. no. 15, pp. 1–33, Apr. 2020.
8. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.
9. S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Found. Trends Inf. Retr.*, vol. 3, no. 4, pp. 333–389, 2009.
10. V. Karpukhin et al., "Dense passage retrieval for open-domain question answering," in *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 6769–6781.
11. N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. EMNLP-IJCNLP*, 2019, pp. 3982–3992.
12. K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, "REALM: Retrieval-augmented language model pre-training," in *Proc. 37th Int. Conf. Machine Learning (ICML)*, PMLR vol. 119, 2020, pp. 3929–3938.
13. G. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open domain question answering," in *Proc. 16th Conf. European Chapter of the ACL (EACL)*, 2021, pp. 874–880.
14. S. Borgeaud et al., "Improving language models by retrieving from trillions of tokens," in *Proc. 39th Int. Conf. Machine Learning (ICML)*, PMLR vol. 162, 2022, pp. 2206–2240.
15. Y. Gao et al., "Retrieval-augmented generation for large language models: A survey," *arXiv:2312.10997*, 2023.
16. A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to retrieve, generate, and critique through self-reflection," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2024, *arXiv:2310.11511*.
17. J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 535–547, Jul. 2021.

18. Y. A. Malkov and D. A. Yashunin, "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 824–836, Apr. 2020.
19. N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, "Lost in the middle: How language models use long contexts," *Trans. Assoc. Comput. Linguistics*, vol. 12, pp. 157–173, 2024, doi: 10.1162/tacl_a_00638.
20. S. Barnett, S. Kurniawan, S. Thudumu, Z. Brannelly, and M. Abdelrazek, "Seven failure points when engineering a retrieval augmented generation system," in *Proc. IEEE/ACM 3rd Int. Conf. AI Engineering – Software Engineering for AI (CAIN)*, Lisbon, Portugal, Apr. 2024, arXiv:2401.05856.
21. J. Chen, H. Lin, X. Han, and L. Sun, "Benchmarking large language models in retrieval-augmented generation," in *Proc. AAAI Conf. Artificial Intelligence*, vol. 38, no. 16, 2024, pp. 17754–17762.
22. S. Es, J. James, L. Espinosa Anke, and S. Schockaert, "RAGAs: Automated evaluation of retrieval augmented generation," in *Proc. 18th Conf. European Chapter of the ACL: System Demonstrations (EACL)*, St. Julian's, Malta, Mar. 2024, pp. 150–158.
23. M. Bogen and A. Rieke, "Help wanted: An examination of hiring algorithms, equity, and bias," *Upturn*, Washington, DC, USA, Dec. 2018.
24. M. Raghavan, S. Barocas, J. Kleinberg, and K. Levy, "Mitigating bias in algorithmic hiring: Evaluating claims and practices," in *Proc. ACM Conf. Fairness, Accountability, and Transparency (FAT*)*, 2020, pp. 469–481.
25. Government of India, *The Digital Personal Data Protection Act, 2023*, Act No. 22 of 2023, Aug. 2023.
26. G. V. Cormack, C. L. A. Clarke, and S. Buettcher, "Reciprocal rank fusion outperforms Condorcet and individual rank learning methods," in *Proc. 32nd Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, Boston, MA, USA, 2009, pp. 758–759.
27. R. Nogueira and K. Cho, "Passage re-ranking with BERT," arXiv:1901.04085, 2019.