Lindberg Simpson – Personal Research Portal

1. Introduction

The Personal Research Portal is a local-first RAG system designed to answer research questions about remote work and productivity using a curated corpus of 15 academic and industry sources. The system makes the following contributions:

- A modular, end-to-end RAG pipeline (ingest → chunk → embed → retrieve → generate → cite → log) that runs entirely on a local machine with no external API dependencies.
- A dual-layer citation system combining inline (source_id, chunk_id) references for claim-level verification with auto-generated bibliographic entries for academic compliance.
- An automated evaluation framework with 20 hand-crafted test queries across three difficulty tiers, computing groundedness, answer relevance, and citation correctness metrics.
- A production web interface (Streamlit) supporting interactive research with source filtering, thread management, evidence table generation, and multi-format export.
- Trust behavior checks that verify the system cites sources when evidence is present and suggests next retrieval steps when evidence is absent.

The corpus covers remote work and productivity research published between 2021 and 2025, a period of rapid change driven by the COVID-19 pandemic. Sources include randomized controlled trials (Bloom et al., 2024), natural experiments (Aksoy et al., 2025; Gibbs et al., 2023), large-scale surveys (Buffer, 2023; Cisco, 2025), government data analysis (BLS, 2024), and systematic reviews (Ferrara et al., 2022; Prasad et al., 2024). This diversity of methods and source types creates a realistic test bed for RAG systems that must synthesize evidence across study designs, geographies, and measurement approaches.

2. System Architecture

2.1 Pipeline Overview

The system follows a five-stage pipeline architecture with clear separation of concerns. Each stage is implemented as an independent Python module, enabling component-level testing and backend substitution. The end-to-end flow is summarized below:

```
                      PDF Sources (15)
                             ↓
         [Ingest] → parse_pdf.py → raw text + metadata
                             ↓
    [Chunk] → chunk.py → 530 passages (2048 char, 256 overlap)
                             ↓
       [Embed] → embed_index.py → FAISS IndexFlatIP (384-dim)
                             ↓
    [Retrieve] → retrieve.py → top-k chunks + similarity scores
                             ↓
[Generate] → generate.py + citations.py → cited answer + references
```

Document Ingestion: PyPDF extracts text from 15 PDF sources. A data manifest (CSV with 17 metadata columns) maps each source to its raw file, processed JSON, and bibliographic details. Extracted text undergoes whitespace normalization and paragraph-boundary preservation.

Semantic Chunking: Documents are split into approximately 2,048-character segments with 256-character overlap. The chunker respects paragraph boundaries to avoid mid-sentence cuts, and applies a sliding window for paragraphs exceeding the chunk size. This produces 530 chunks across the corpus, each assigned a traceable chunk_id (e.g., SRC001_chunk_0).

Embedding and Indexing: Chunk text is embedded using FastEmbed (BAAI/bge-small-en-v1.5, ONNX-based, 384 dimensions) as the primary backend, with sentence-transformers (all-MiniLM-L6-v2) as a fallback. Embeddings are L2-normalized and stored in a FAISS IndexFlatIP index for cosine similarity search.

Retrieval: User queries are embedded with the same model and searched against the FAISS index, returning the top-k (default 5) chunks with similarity scores. When the index is unavailable or embedding fails, the system falls back to keyword-based retrieval using word overlap.

Generation and Post-Processing: Retrieved chunks and the user query are passed to Ollama (Gemma 3 4B) with a constrained system prompt requiring inline citations, explicit acknowledgment of evidence gaps, and suggested next retrieval steps. Post-processing fixes malformed citations, appends a formatted bibliographic reference section from the data manifest, and adds evidence strength labels (High/Medium/Low) based on retrieval similarity scores.

2.2 Technology Stack

Category	Technology	Purpose
LLM	Ollama (Gemma 3 4B)	Local answer generation
Embeddings (primary)	FastEmbed / bge-small-en-v1.5	ONNX-based document and query embedding
Embeddings (fallback)	sentence-transformers / all-MiniLM-L6-v2	PyTorch-based embedding fallback
Vector Index	FAISS IndexFlatIP	Dense vector cosine similarity search
PDF Parsing	PyPDF 4.x	Source document text extraction
Web Interface	Streamlit 1.28+	Five-page interactive research portal
Data Processing	pandas 2.x	Manifest loading, filtering, metadata
Export	fpdf2 2.7+	PDF export for evidence tables

2.3 Web Interface

The Streamlit-based web interface consists of five pages that support the complete research workflow. A sidebar navigation panel lets the user switch between pages, each of which addresses a distinct stage of the research process:

Ask: The primary query interface. Users enter a research question, optionally filter by source IDs, and adjust the top-k retrieval parameter. The page displays the generated answer with inline citations, a collapsible list of retrieved chunks with similarity scores and evidence strength labels (High/Medium/Low), and a formatted bibliographic reference section. Users can save any result as a research thread with one click.

History: Displays all saved research threads in reverse chronological order. Each thread shows the original query, the generated answer, retrieved sources, and computed scores (groundedness, answer relevance). Users can revisit any thread to review evidence or continue research.

Artifacts: Generates structured research artifacts from saved threads. Users select a thread and produce an evidence table (Claim | Evidence snippet | Citation | Confidence | Notes). The table can be exported in four formats: Markdown, CSV, PDF, and BibTeX. Generated artifacts are stored in outputs/artifacts/.

Build & Run: System administration page for index management and batch evaluation. Users can trigger PDF ingestion, rebuild the FAISS index, run the full 20-query evaluation suite, or execute trust behavior checks. Progress is displayed in real time with a per-query progress bar.

Evaluation: Displays evaluation results from completed runs. Shows per-query scores (groundedness, citation correctness, usefulness) in a sortable table, plus aggregate summary metrics across the full query set. Users select from available evaluation run files (JSONL) to compare performance across iterations.

The interface caches the FAISS index and chunk list in Streamlit session state for fast repeated queries. When Ollama is unavailable, the system displays an informative error, and the retrieval pipeline continues to function independently, returning matched chunks without generation. A keyword-based fallback retriever activates automatically when the embedding model fails to load, ensuring the system degrades gracefully rather than failing completely.

2.4 Research Threads

Research threads persist the full context of each query interaction for later review, artifact generation, and export. Each thread is stored as a JSON line in threads/threads.jsonl with the following schema:

Field	Type	Description
thread_id	string	8-character hex UUID for the thread
timestamp	ISO 8601	UTC timestamp of the query
query	string	The user's research question
retrieved	array	Top-k chunks with source_id, chunk_id, score, text
answer	string	Generated answer with inline citations
groundedness	float	Groundedness score (0–1)
answer_relevance	float	Answer relevance score (0–1)

This file-based approach avoids database dependencies while preserving full provenance. Each thread contains all information needed to regenerate artifacts, verify citations, or audit the retrieval pipeline. Threads are created from the Ask page and browsed from the History page.

2.5 Research Artifacts

The system generates structured research artifacts from saved threads, satisfying the rubric requirement for exportable research outputs. The primary artifact type is the evidence table, which follows this schema:

Column	Content	Source
Claim / Query	The research question or extracted claim	Thread query
Evidence Snippet	Relevant text passage (truncated to key content)	Retrieved chunk text
Citation	(source_id, chunk_id) inline reference	Retrieval metadata
Confidence	High/Medium/Low with numeric similarity score	FAISS cosine similarity
Notes	Source relevance note from the data manifest	data_manifest.csv

Citation traceability is maintained end-to-end: each evidence table row cites a specific (source_id, chunk_id) that resolves to a passage in data/processed/ and a bibliographic entry in data/data_manifest.csv. The data manifest provides 17 metadata columns including source_id, title, authors, year, source_type, venue, url_or_doi, raw_path, processed_path, methodology, measurement_type, geography, sample_size, and key_finding.

Artifacts are exported in four formats: Markdown (.md), CSV, PDF (via fpdf2), and BibTeX (.bib). Generated outputs are stored in outputs/artifacts/ and include evidence tables (e.g., evidence_table-2.md) and a references.bib file for integration with LaTeX workflows. See outputs/artifacts/ in the repository for example generated artifacts.

3. Design Choices

3.1 Local-First Deployment

The decision to deploy entirely on-device using Ollama (rather than cloud APIs such as OpenAI or Anthropic) was driven by three factors: (1) no API keys or rate limits are needed, enabling unrestricted iterative development; (2) full data privacy, since no research content leaves the local machine; and (3) zero marginal cost per query. The trade-off is reduced model capability: Gemma 3 4B has approximately 4 billion parameters compared to GPT-4's estimated 175B+, resulting in higher hallucination rates and weaker instruction-following for complex multi-part queries. This trade-off is acceptable for a research support tool with human-in-the-loop verification.

3.2 Chunking Parameters

Chunk size was set to 2,048 characters (approximately 512 tokens) with 256 characters of overlap. This balances retrieval granularity against semantic completeness: smaller chunks improve precision for factual queries but fragment multi-sentence findings, while larger chunks preserve context but reduce retrieval specificity. The overlap of approximately 12.5% mitigates boundary effects—critical findings that span a chunk boundary appear in both adjacent chunks, reducing information loss at the cost of a modest increase in index size. Paragraph-boundary awareness ensures that chunks begin and end at natural text breaks.

3.3 Embedding Model Selection

FastEmbed (BAAI/bge-small-en-v1.5) was chosen as the primary embedding backend because it uses ONNX runtime rather than PyTorch, which avoids segmentation faults observed on Apple Silicon with sentence-transformers and reduces memory footprint. The bge-small model produces 384-dimensional embeddings optimized for retrieval tasks. Sentence-transformers (all-MiniLM-L6-v2) serves as an automatic fallback, ensuring the system operates on any platform regardless of ONNX availability.

3.4 Citation and Grounding System

The citation system operates at two levels. Inline citations use the format (source_id, chunk_id)—for example, (SRC004, SRC004_chunk_0)—enabling claim-level verification against specific text passages. A post-processing step appends a formatted bibliographic reference section drawn from the data manifest (Authors, Year, Title, URL/DOI) and an evidence strength section that categorizes each retrieved chunk as High (similarity ≥ 0.45), Medium (≥ 0.30), or Low (< 0.30). This dual-layer approach balances granular traceability with academic citation standards.

3.5 Prompt Engineering

The system prompt enforces six constraints: (1) answer only from provided chunks; (2) cite every claim with (source_id, chunk_id); (3) state explicitly when evidence is missing; (4) suggest a next retrieval step for gaps; (5) do not infer causality unless the source claims it; and (6) present both sides of conflicting evidence. These constraints, informed by Phase 1 prompt comparison experiments, reduce fabrication and build trust but produce more verbose, citation-heavy responses that can obscure the direct answer. A post-generation citation fixer corrects common malformation patterns (e.g., numeric-only or format-swapped citations) by mapping them back to retrieved chunk IDs.

4. Evaluation

4.1 Test Set Design

The evaluation test set consists of 20 hand-crafted queries with documented expected behaviors, organized into three difficulty tiers. Ten direct queries focused on single-fact extraction from a specific source, five synthesis queries that ask for a multi-source comparison and integration, and five edge case queries that call for uncertainty handling and absence detection. Each query specifies the target information, expected source(s), and evaluation focus. This three-tier structure covers realistic research use cases and tests distinct system capabilities.

4.2 Metrics

Three automated metrics are computed for every query during evaluation:

Groundedness (0–1): The fraction of answer sentences whose words overlap at least 40% with the concatenated text of retrieved chunks. This measures how much of the answer is directly supported by retrieved evidence. It is a conservative lower bound that captures direct quotes and near-verbatim restatements but assigns zero credit to paraphrased content.

Answer Relevance (0–1): The fraction of query keywords present in the answer. This measures whether the answer addresses the question's topic and terminology, penalizing off-topic content.

Citation Correctness (1–4): A score based on citation pattern validity: 4 = all citations match retrieved chunks; 3 = some match; 2 = citations present but none match; 1 = no citations at all.

4.3 Results

Results are reported from evaluation run 14, the final and most complete run with 20/20 successful query-answer pairs. A total of 14 evaluation runs were conducted during iterative development, enabling comparison across system configurations.

Metric	Value	Interpretation
Retrieval Precision (top-5)	90%	18/20 queries contain relevant info in the top-5 chunks
Mean Groundedness	0.50	50% of answer sentences directly supported by chunks
Mean Answer Relevance	0.76	76% of query keywords appear in answer
Citation Format Compliance	100%	20/20 answers use correct (source_id, chunk_id) format

The system’s 0.50 groundedness with a 4B-parameter model is at the upper end of expectations for a small parameter model. Citation compliance of 100% indicates a robust citation formatting and post-processing pipeline. Answer relevance of 0.76 reflects the model’s tendency toward verbose preambles that dilute keyword density.

Performance varied by query type. Direct factual queries achieved strong groundedness (0.50) and relevance (0.79), benefiting from clear semantic overlap between query and corpus. Synthesis queries showed the lowest relevance (0.64) due to source imbalance in the top-5 retrieval set as the retriever often returned 3–4 chunks from a single dominant source, and the model’s comparative language diverged from query keywords. Edge cases achieved the highest groundedness (0.53) because the model’s hedging language closely mirrored retrieved chunk wording, though this surface-level grounding often masked substantive reasoning failures.

Query Type	Groundedness	Answer Relevance	Key Observation
Direct (Q01–Q10)	0.50	0.79	Strong retrieval but occasional hallucinated specifics
Synthesis (Q11–Q15)	0.45	0.64	Source imbalance limits multi-perspective coverage
Edge Case (Q16–Q20)	0.53	0.84	Surface grounding masks hedging failures

4.4 Trust Behavior Checks

Two automated trust tests verify core system behaviors. The citation presence test confirms that the system includes at least one valid inline citation when relevant evidence is retrieved (pass rate: 100%). The suggested next step test verifies that, when chunks do not support the query, the answer explicitly states the evidence gap and suggests a concrete next retrieval action—e.g., 'Consider searching for career advancement remote work.' These checks run programmatically and are accessible from both the CLI and the web interface.

4.5 Error Analysis

Qualitative analysis of incorrect or suboptimal answers revealed four primary failure modes:

Hallucination of specifics: The model fabricated exact numbers (e.g., '60 minutes' not in the retrieved chunk for Q01) and structural references ('Table A3' for Q07). This was the most frequent error type, concentrated in direct factual queries.

Citation misattribution: Correct claims were sometimes paired with overly general chunk citations, or the same chunk was cited for multiple unrelated claims, reducing citation precision.

Source imbalance in synthesis: For multi-source queries, cosine similarity retrieved 3–4 chunks from a single detailed source, crowding out shorter but equally relevant alternative sources.

Inconsistent uncertainty calibration: Edge-case queries designed to elicit 'evidence not found' responses instead received hedged answers (e.g., 'might contain' rather than 'does not contain'), undermining the system's trust behavior.

Three representative failure cases from evaluation run 14 illustrate these patterns concretely:

Failure Case 1: Source Confusion (Q11, Synthesis)
Query: "Compare how Bloom et al. (SRC004) and Gibbs et al. (SRC010) measure productivity."

Issue: The retriever returned 0/5 chunks from the requested sources (SRC004 and SRC010). Instead, it retrieved chunks from SRC013 (Lim et al.) and SRC002 (Aksoy et al.). The model then incorrectly attributed Lim et al.'s survey methodology to Bloom et al., stating: "Bloom et al. (SRC013, SRC013_chunk_37) primarily uses survey responses from corporate Malaysia employees." This is factually wrong—Bloom et al. conducted a randomized controlled trial at a Chinese technology firm. The root cause is retrieval: cosine similarity matched the general topic (remote work productivity) rather than the specific source requested.

Failure Case 2: Hedging Instead of Absence (Q16, Edge Case)
Query: "Does this corpus contain credible evidence that fully remote work increases promotion rates?"

Issue: The corpus contains no direct evidence on promotion rates. The expected behavior is a clear absence statement with a suggested next retrieval step. Instead, the model responded: "The provided evidence suggests a complex relationship between remote work and promotion rates, but doesn't directly state that remote work *increases* promotion rates." This hedged framing implies the evidence is ambiguous rather than absent, undermining trust. The model then discussed siloing effects from Yang et al. (cited as SRC011_chunk_37), which concerns communication patterns, not promotions.

Failure Case 3: Missing Statistic (Q07, Direct)
Query: "What percentage of respondents in the Cisco Global Hybrid Work Study 2025 reported improved wellbeing due to hybrid work?"

Issue: The Cisco study (SRC007) does report wellbeing improvements, but the specific percentage was not in the retrieved chunks. The model correctly acknowledged: "SRC007 does not explicitly state the percentage of respondents reporting improved wellbeing," and pivoted to discussing trends from SRC007_chunk_0. While the absence acknowledgment is correct, the model failed to suggest a next retrieval step (e.g., "Try searching for 'Cisco wellbeing percentage hybrid work'"), which is required by the system prompt constraints.

5. Limitations

5.1 Generation Quality

The primary bottleneck is the local LLM's capability. Gemma 3 4B, while sufficient for demonstration and iterative development, exhibits higher hallucination rates than larger models. Specifically, it fabricates precise statistics, invents table/figure references, and produces verbose preambles that dilute answer relevance. Instruction-following is inconsistent: the model often acknowledges constraints in the system prompt but violates them within the same response. Upgrading to a larger local model (8B+ parameters) or a cloud API would likely reduce these errors by 30–50%.

5.2 Retrieval Diversity

Dense retrieval with a single embedding model favors sources with longer, more detailed discussions that match query embeddings across multiple chunks. This causes source imbalance for synthesis queries: one or two detailed sources dominate the top-5, excluding concise but relevant alternatives. Increasing k trades diversity for noise, since lower-ranked chunks are often off-topic. Hybrid retrieval (combining dense embeddings with BM25 keyword matching) and cross-encoder reranking would address this limitation.

5.3 Corpus Coverage

The 15-source corpus, while carefully curated, provides limited coverage for edge-case topics such as promotion rates, gender equity, and sectoral breakdowns. Several edge-case queries (Q16, Q19) tested for information that is legitimately absent from the corpus, but the system could not reliably distinguish between 'the corpus lacks this information' and 'retrieval failed to find it.' Expanding to 50+ sources with broader topical coverage would reduce this ambiguity.

5.4 Evaluation Metric Constraints

The automated evaluation metrics have methodological limitations. Groundedness (sentence-level word overlap) penalizes paraphrased but accurate content and cannot detect fabricated numbers embedded in otherwise well-grounded sentences. Answer relevance (keyword overlap) captures topical coverage but not factual correctness. Citation correctness (pattern matching) verifies that cited IDs appear in the retrieved set but does not assess whether the cited chunk actually supports the specific claim. More sophisticated approaches—LLM-as-judge scoring, semantic entailment models, or human annotation—would provide richer quality assessment.

6. Next Steps

The highest-impact improvement is upgrading the generation model. Replacing Gemma 3 4B with Llama 3.1 8B, Instruct or Mistral 7B Instruct would significantly reduce hallucination and improve instruction-following while maintaining local-first deployment. Adding BM25 hybrid retrieval addresses the second major bottleneck, rare-term coverage, by combining sparse keyword matching with dense semantic search. Together, these two changes address the system's principal weaknesses with minimal architectural disruption.

7. Conclusion

This project demonstrates the feasibility of building a local, privacy-preserving RAG system for academic research synthesis. The Personal Research Portal processes 15 curated sources into 530 semantically chunked passages, retrieves relevant evidence with 90% top-5 precision, and generates citation-backed answers with 100% format compliance. The automated evaluation framework, 20 test queries across three difficulty tiers with reproducible metrics, provides a foundation for iterative system improvement.

The evaluation reveals that retrieval quality is strong, but generation quality remains the principal bottleneck. A 4B-parameter local LLM produces moderate groundedness (0.50) with hallucination concentrated in specific numbers and structural references. This gap between retrieval capability and generation reliability is the central challenge for domain-specific RAG systems and motivates the prioritized roadmap of model upgrades, hybrid retrieval, and post-generation safeguards.

The system's modular architecture, dual-layer citation design, and production web interface establish a strong technical baseline that supports further development. The evaluation test set and automated metrics enable systematic benchmarking across future iterations, transforming RAG development from ad hoc experimentation into a measurable engineering process.

References

[1] Adomako, K. et al. (2024). Impact of Working from Home on Employee Productivity in Post COVID-19 Era. World Journal of Advanced Research and Reviews. doi:10.30574/wjarr.2024.23.1.2022

[2] Aksoy, C. G. et al. (2023). Time Savings When Working from Home. NBER Working Paper w30866.

[3] Aksoy, C. G. et al. (2025). Remote Work Employee Mix and Performance. Becker Friedman Institute, UChicago.

[4] Barrero, J. M., Bloom, N. & Davis, S. J. (2023). The Evolution of Work from Home. Journal of Economic Perspectives, 37(4), 1–30.

[5] Bartik, A. et al. (2023). The Rise of Remote Work: Evidence on Productivity and Preferences. NBER Working Paper w20-138.

[6] Bloom, N., Han, R. & Liang, J. (2024). Hybrid Working from Home Improves Retention without Damaging Performance. Nature. doi:10.1038/s41586-024-07500-2

[7] Bureau of Labor Statistics (2024). Productivity and Remote Work. U.S. Department of Labor.

[8] Buffer (2023). State of Remote Work 2023. buffer.com/state-of-remote-work/2023.

[9] Cisco Systems (2025). Global Hybrid Work Study 2025.

[10] Ferrara, B. et al. (2022). Investigating the Role of Remote Working on Employees' Performance and Well-Being. IJERPH, 19(19), 12373.

[11] George, T. J. et al. (2022). Supporting the Productivity and Wellbeing of Remote Workers: Lessons from COVID-19. Organizational Dynamics. doi:10.1016/j.orgdyn.2021.100869

[12] Gibbs, M., Mengel, F. & Siemroth, C. (2023). Work from Home and Productivity: Evidence from Personnel and Analytics Data. JPEM. doi:10.1086/721803

[13] Lim, B. S. W. et al. (2025). Assessing the Causal Relationship of Remote Work and Employee Productivity. Cogent Social Sciences. doi:10.1080/23311886.2025.2481194

[14] Makridis, C. (2025). Remote Staff Hours Fall but Productivity Steady (For Now). Gallup/ATUS Analysis.

[15] Prasad, S. et al. (2024). Exploring the Impact of Remote Work on Productivity: A Systematic Review. Acta Universitatis Bohemiae Meridionalis. doi:10.32725/acta.2024.008