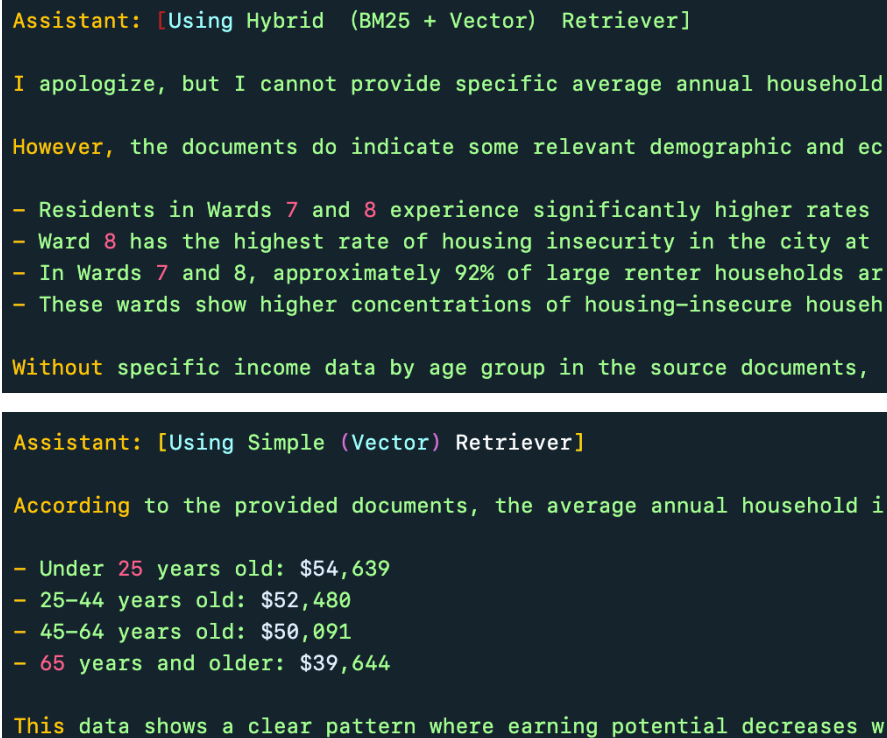


Reflection Note: DC Housing Policy Chatbot

Strengths and Limitations of the Approach

My approach to building a RAG-based chatbot for DC housing policy demonstrated several notable strengths while also revealing important limitations. The focused scope of my document collection, covering DC housing policy from 2019-2024, proved particularly effective in simulating a realistic policy research assistant scenario. This domain specificity allowed for deeper insight into how such a system might perform in actual policy research settings. The implementation of metadata storage for source traceability and real-time document ingestion capabilities provided a solid foundation for practical research applications, enabling users to verify information sources and continuously update the knowledge base as new research emerges.

The hybrid retrieval system, combining semantic search through vector embeddings with BM25 keyword-based search, represented an attempt to leverage complementary search methodologies. BM25 (Best Matching 25, also known as Okapi BM25) is a ranking algorithm for information retrieval and search engines that determines a document's relevance to a given query and ranks documents based on their relevance scores. In RAG systems, BM25 can complement vector retrieval by handling keyword matching while vector search captures semantic relationships, theoretically offering improved retrieval performance while maintaining computational efficiency. However, this implementation revealed unexpected complexities in retrieval behavior.



The image shows two screenshots of a chatbot interface. The top screenshot shows the 'Assistant: [Using Hybrid (BM25 + Vector) Retriever]' response, which is incomplete. The bottom screenshot shows the 'Assistant: [Using Simple (Vector) Retriever]' response, which is complete and includes a list of average annual household income data by age group.

Assistant: [Using Hybrid (BM25 + Vector) Retriever]

I apologize, but I cannot provide specific average annual household

However, the documents do indicate some relevant demographic and ec

- Residents in Wards 7 and 8 experience significantly higher rates
- Ward 8 has the highest rate of housing insecurity in the city at
- In Wards 7 and 8, approximately 92% of large renter households ar
- These wards show higher concentrations of housing-insecure househ

Without specific income data by age group in the source documents,

Assistant: [Using Simple (Vector) Retriever]

According to the provided documents, the average annual household i

- Under 25 years old: \$54,639
- 25-44 years old: \$52,480
- 45-64 years old: \$50,091
- 65 years and older: \$39,644

This data shows a clear pattern where earning potential decreases w

Figure 1 The Hybrid Retriever was unable to answer question 4, while the Simple Retriever successfully provided a response.

Perhaps the most significant finding of my experiment was the critical importance of chunking parameters. The system's performance proved remarkably sensitive to chunk size and overlap settings, with seemingly minor adjustments sometimes determining whether specific information could be

retrieved at all. This sensitivity emerged as a fundamental limitation, overshadowing even the sophisticated hybrid retrieval mechanism I implemented. In several cases, changes to chunking parameters had more impact on answer quality than the choice between vector-only or hybrid retrieval methods.

Notably, with optimized chunking parameters, I made another surprising discovery regarding model selection: the open-source Llama 3.2-90B model achieved results comparable to more expensive models like Claude 3.5 Sonnet and GPT o1-preview. This finding suggests that proper document processing might be more crucial than using the most advanced language models for achieving high-quality results. The implications for scaling are significant: organizations might be able to deploy effective RAG systems using more affordable open-source models, substantially reducing operational costs while maintaining performance quality.

The small size of my document corpus (eight documents) presented both advantages and limitations. While it allowed for detailed analysis of retrieval behavior, it made it difficult to properly evaluate certain enhancement techniques like reranking, which typically show their value in larger collections. Additionally, metadata tracking, particularly for page numbers, showed inconsistencies that would need addressing in a production environment.

Potential Improvements for Production

Looking toward production deployment, several key areas for improvement emerge. Given my findings about the critical impact of document processing on retrieval quality, developing a sophisticated chunking strategy stands as the highest priority. Rather than relying on fixed chunk sizes, a production system should implement dynamic chunking that adapts to document structure and content type, supported by automated testing to optimize parameters for different document categories.

The retrieval pipeline requires careful refinement, particularly in fine-tuning the hybrid approach of BM25 and vector search. While my experiments showed mixed results with BM25, a more nuanced implementation with proper parameter tuning could prove valuable for larger document sets. Essential features for a production environment should include conversation history management, robust source verification, and efficient document update mechanisms.

For larger-scale deployments, implementing GraphRAG architectures and hierarchical retrieval systems could help maintain performance as the knowledge base grows. Importantly, my findings suggest that sophisticated document processing pipelines - rather than more powerful language models - might be the key to improving performance in production environments. This insight, combined with the promising performance of more affordable language models, offers a practical path forward for developing cost-effective policy research assistance systems.