

# 🔍 Hybrid RAG System with Automated Evaluation# Hybrid RAG System with Automated Evaluation# Hybrid RAG System with Automated Evaluation# Hybrid RAG System with Automated Evaluation

---

GitHub Repository

Python 3.10+

Streamlit Dashboard GitHub Repository

A comprehensive **Hybrid RAG System** combining Dense Vector Retrieval, Sparse Keyword Retrieval, and Reciprocal Rank Fusion for Wikipedia-based Question Answering.

📦 **Repository:** [github.com/vishalvishal099/Hybrid\\_RAG\\_System\\_with\\_Automated\\_Evaluation](https://github.com/vishalvishal099/Hybrid_RAG_System_with_Automated_Evaluation)  
A comprehensive implementation of a **Hybrid RAG System** combining **Dense Vector Retrieval (ChromaDB)**, **Sparse Keyword Retrieval (BM25)**, and **Reciprocal Rank Fusion (RRF)** to answer questions from Wikipedia articles.

GitHub Repository GitHub Repository

---

## 📄 Table of Contents GitHub Repository:

[https://github.com/vishalvishal099/Hybrid\\_RAG\\_System\\_with\\_Automated\\_Evaluation](https://github.com/vishalvishal099/Hybrid_RAG_System_with_Automated_Evaluation)

- [Quick Start](#)
  - [Project Overview](#)
  - [System Architecture](#)  
**Last Updated:** February 8, 2026  
A comprehensive implementation of a **Hybrid RAG System** combining **Dense Vector Retrieval (ChromaDB)**, **Sparse Keyword Retrieval (BM25)**, and **Reciprocal Rank Fusion (RRF)** to answer questions from Wikipedia articles.  
A comprehensive implementation of a **Hybrid RAG System** combining **Dense Vector Retrieval (ChromaDB)**, **Sparse Keyword Retrieval (BM25)**, and **Reciprocal Rank Fusion (RRF)** to answer questions from Wikipedia articles.
  - [Project Structure](#)
  - [Evaluation Results](#)
  - [Documentation](#)
  - [Screenshots](#)---
  - [Contributors](#)
-

## 🚀 Quick Start GitHub Repository:

[https://github.com/vishalvishal099/Hybrid\\_RAG\\_System\\_with\\_Automated\\_Evaluation](https://github.com/vishalvishal099/Hybrid_RAG_System_with_Automated_Evaluation) GitHub Repository:

[https://github.com/vishalvishal099/Hybrid\\_RAG\\_System\\_with\\_Automated\\_Evaluation](https://github.com/vishalvishal099/Hybrid_RAG_System_with_Automated_Evaluation)

## 🚀 Quick Start

### Prerequisites

#### Prerequisites

| Requirement | Version |

|-----|-----|- Python 3.10+

| Python | 3.10+ |

| RAM | 4GB+ |- 4GB+ RAM **Last Updated:** February 8, 2026---

| Disk Space | 2GB+ |

### Installation

#### Installation

```
# 1. Clone repository

git clone
https://github.com/vishalvishal099/Hybrid_RAG_System_with_Automated_Evaluation.git

cd Hybrid_RAG_System_with_Automated_Evaluation``bash---## 🚀 Quick Start


# 2. Create virtual environment# Clone repository

python -m venv venv

source venv/bin/activate # Windows: venv\Scripts\activate
git clone
https://github.com/vishalvishal099/Hybrid_RAG_System_with_Automated_Evaluation.git


# 3. Install dependenciescd Hybrid_RAG_System_with_Automated_Evaluation

pip install -r requirements.txt

```## 🚀 Quick Start## Prerequisites
```

```
### Run Application# Create virtual environment

```bash
python -m venv venv - Python 3.10+
# Start Streamlit Dashboard

streamlit run app_chromadb.py --server.port 8502 source venv/bin/activate
# Windows: venv\Scripts\activate
```

```

Prerequisites- 4GB+ RAM

🌐 **Access URL:** <http://localhost:8502>

## Install dependencies

---

Run Evaluation

```
pip install -r requirements.txt - Python 3.10+
```

```
# Run full evaluation (100 questions × 3 methods)```
python evaluate_chromadb_fast.py
```
-- 4GB+ RAM### Installation
```

```
---### Run the Application
```

```
## 🎓 Project Overview
```

This project implements a \*\*state-of-the-art Hybrid RAG system\*\* that:

```
```bash
```

Feature	Description	# Start Streamlit UI### Installation	```bash
			----- -----

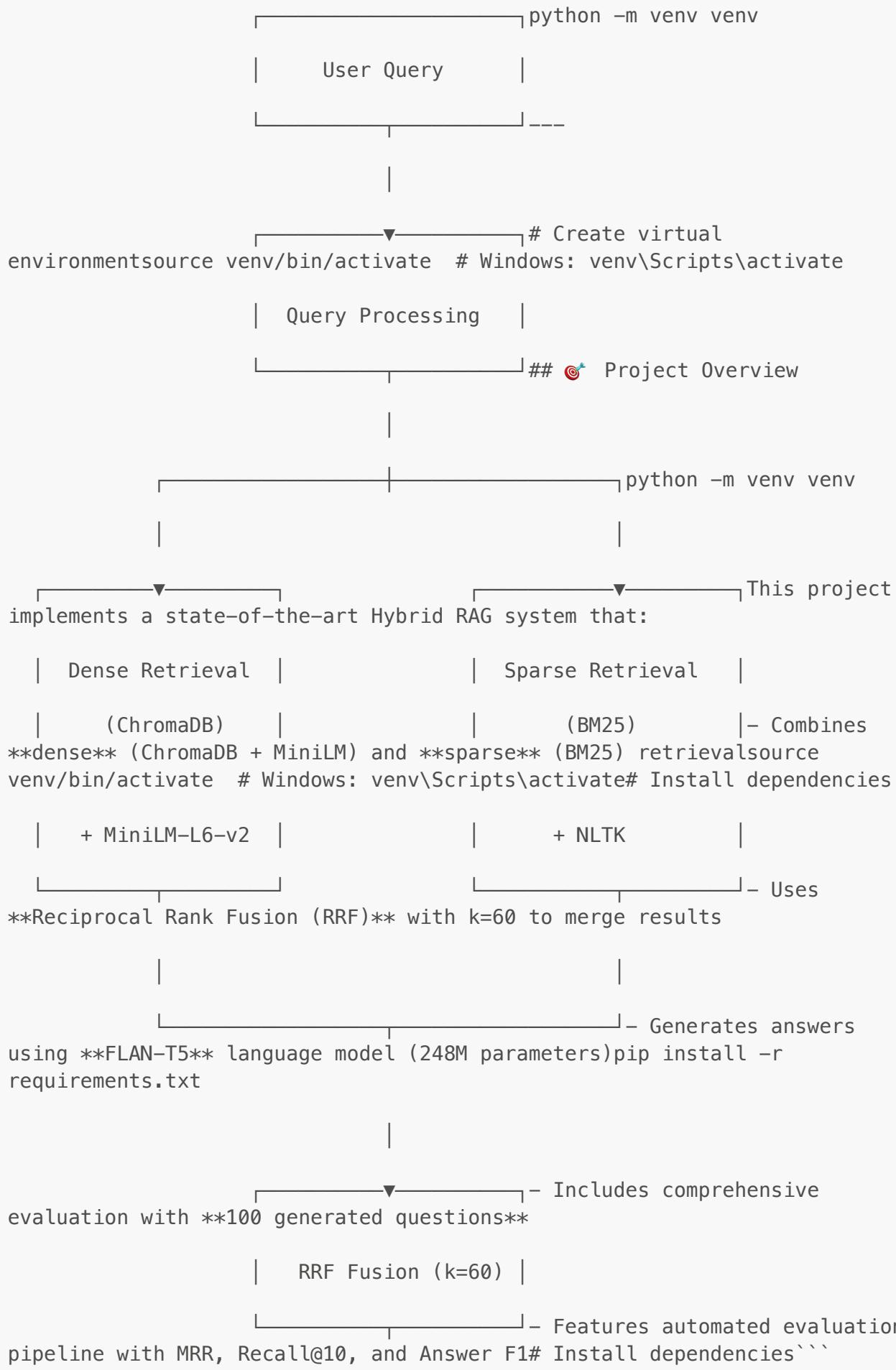
```
| ♦ **Dense Retrieval** | ChromaDB + MiniLM embeddings (384 dimensions)  
| streamlit run app_chromadb.py --server.port 8502  
  
| ♦ **Sparse Retrieval** | BM25 with NLTK tokenization |  
  
| ✎ **Fusion** | Reciprocal Rank Fusion (RRF) with k=60 |```# Clone  
repository  
  
| 🎨 **Generation** | FLAN-T5-Base (248M parameters) |  
  
| 📈 **Evaluation** | MRR, Recall@10, Answer F1 |  
  
| 💻 **Interface** | Interactive Streamlit Dashboard |  
  
**Access URL:** http://localhost:8502```bashgit clone  
https://github.com/vishalvishal099/Hybrid\_RAG\_System\_with\_Automated\_Evaluation.git
```

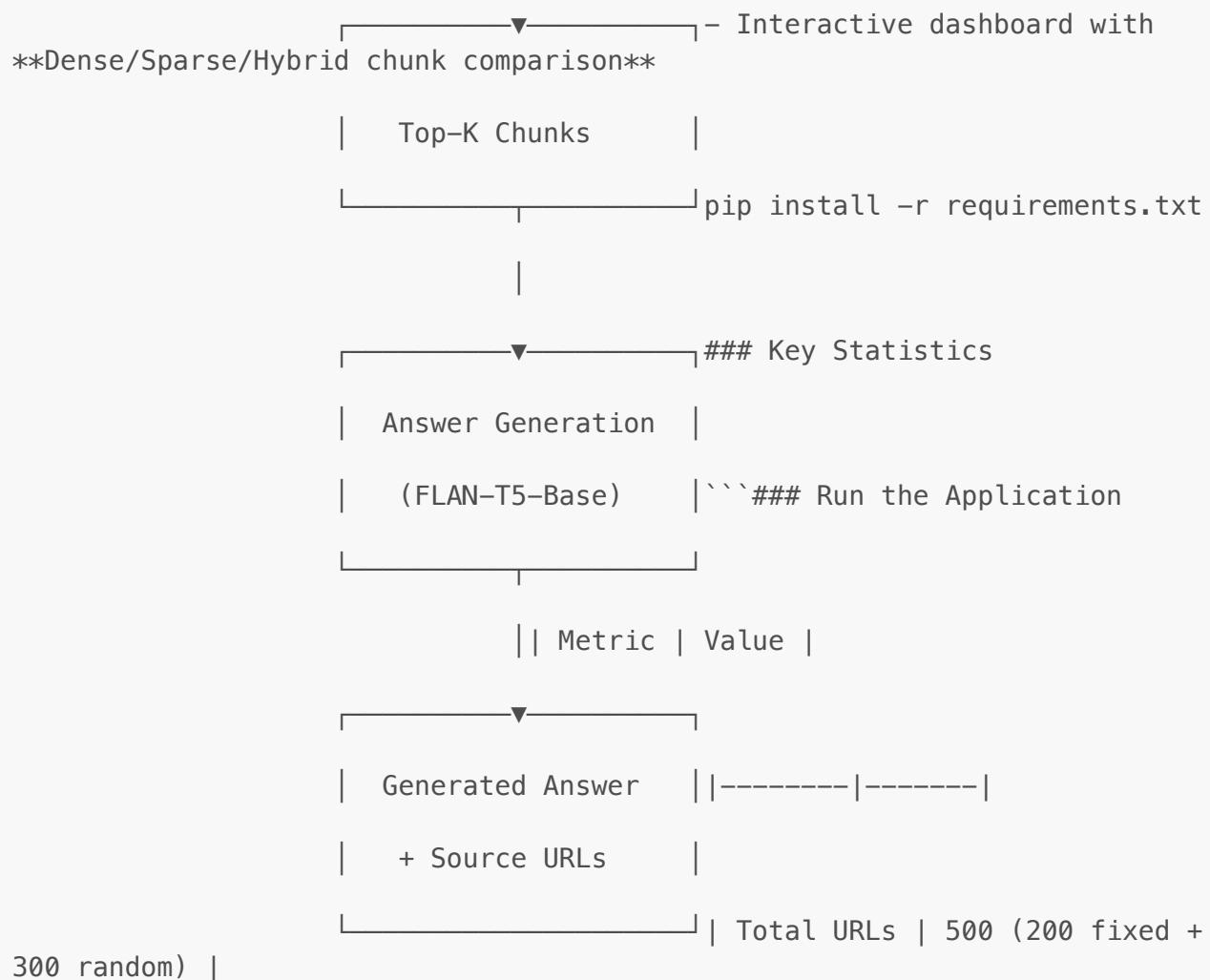
### ### 📈 Key Statistics

Metric	Value
Wikipedia URLs	500 (200 fixed + 300 random)
Total Chunks	7,519 segments
Embedding Dimensions	384
Chunk Size	500 characters
Evaluation Questions	100

```
# Full evaluation (100 questions × 3 methods)  
  
---  
  
python evaluate_chromadb_fast.pycd  
Hybrid_RAG_System_with_Automated_Evaluation# Create virtual environment
```

### ## 📈 System Architecture





| Total Chunks | 7,519 segments |### Run the Application```bash

---

| Embedding Dimensions | 384 |

## 📁 Project Structure

| Chunk Size | 500 characters |# Start Streamlit UI

```
Hybrid_RAG_System_with_Automated_Evaluation/| Evaluation Questions | 100 |
|
|   Core Files``bash./start_ui.sh
|   |   chromadb_rag_system.py      # Core RAG implementation
|   |   app_chromadb.py            # Streamlit UI---
|   |   evaluate_chromadb_fast.py # Evaluation pipeline
```



```
|   └── architecture_diagram.png # System diagram  
|  
|   └── data_flow_diagram.png    # Data flow      |      (ChromaDB +  
|       (BM25 +           |### Run Evaluation  
|  
|  
|   └── screenshots/          # UI screenshots     |      MinILM-L6-v2)  
|       └── NLTK  
|  
|   └── submission/          # Submission package  
|  
|   └── chroma_db/           # Vector database  
|       └── bash  
|  
|  
|   └── requirements.txt      # Dependencies  
|  
|  
|   └── config.yaml          # Configuration  
|  
└── README.md                # This file  
    └── bash# Full evaluation (100 questions × 3  
methods)
```

```
# Full evaluation (100 questions × 3  
methods)python evaluate_chromadb_fast.py
```

## 📈 Evaluation Results

```
|      Reciprocal Rank      |
```

## 🏆 Performance Summary

```
|      Fusion (k=60)        |
```

```
|      python evaluate_chromadb_fast.py
```

| Method | MRR | Recall@10 | Avg Time |

----- ----- ----- -----	-----
-------------------------	-------

| Dense (ChromaDB) | 0.3025 | 0.33 | 5.86s |

| **Sparse (BM25)** | **0.4392** | **0.47** | **5.53s** | | ``# Generate reports

| Hybrid (RRF) | 0.3783 | 0.43 | 6.37s |



**Key Finding:** BM25 outperforms Dense by **45%** on MRR for Wikipedia content.

| Top-K Chunks | python generate\_report.py

## Generation Metrics



| Method | BLEU | ROUGE-L | BERTScore |

|-----|-----|-----|-----| |---```

| Dense | 0.015 | 0.120 | 0.780 |

| Sparse | 0.022 | 0.145 | 0.820 | |-----|

| Hybrid | 0.018 | 0.135 | 0.810 |

| Answer Generation |

## Question Distribution

| (FLAN-T5-base) |

| Type | Count | Percentage |

|-----|-----|-----|-----| |-----| ## 🔍 Project Overview---

| Factual | 59 | 59% |

| Comparative | 15 | 15% | |

| Inferential | 11 | 11% |

| Multi-hop | 15 | 15% | 

| **Total** | **100** | 100% |

### Generated Answer

+ Source URLs | This project implements a state-of-the-art Hybrid RAG system that:## 🔍 Project Overview

## 📚 Documentation

| Document | Description | Link |

|-----|-----|-----|`-- Combines **dense** (ChromaDB + MiniLM) and **sparse** (BM25) retrieval

| 📄 Submission Guide | Complete submission reference | [SUBMISSION\\_REFERENCE\\_GUIDE.md](#) |

| 📁 Deliverables | Assignment mapping | [SUBMISSION\\_DELIVERABLES.md](#) |

| 🔗 Quick Links | All GitHub links | [QUICK\\_ACCESS\\_LINKS.md](#) |

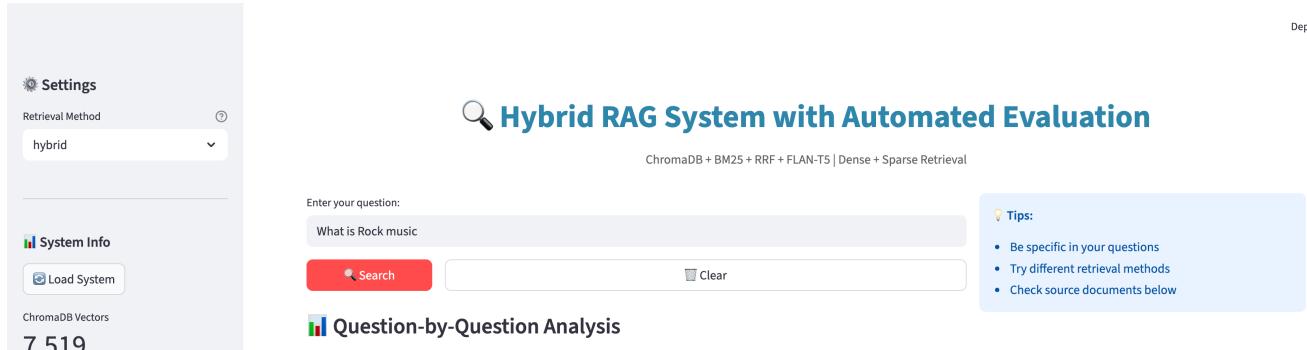
| 📈 Metrics | Metric justification | [docs/METRIC\\_JUSTIFICATION.md](#) |---- Uses **Reciprocal Rank Fusion (RRF)** with k=60 to merge resultsThis project implements a state-of-the-art Hybrid RAG system that:

| 📄 Report | Full evaluation report | [submission/05\\_reports/Hybrid\\_RAG\\_Evaluation\\_Report.md](#) |

📁 Project Structure- Generates answers using **FLAN-T5** language model (248M parameters)- Combines **dense** (ChromaDB + MiniLM) and **sparse** (BM25) retrieval

## 📸 Screenshots

### Query Interface



The screenshot shows the user interface of the Hybrid RAG System. On the left, there's a sidebar with 'Settings' (Retrieval Method: hybrid), 'System Info' (Load System, ChromaDB Vectors: 7.519), and 'Question-by-Question Analysis' (7.519). The main area has a title 'Hybrid RAG System with Automated Evaluation' and a subtitle 'ChromaDB + BM25 + RRF + FLAN-T5 | Dense + Sparse Retrieval'. It features a search bar with the query 'What is Rock music', a 'Search' button, and a 'Clear' button. To the right, there's a 'Tips:' section with three bullet points: 'Be specific in your questions', 'Try different retrieval methods', and 'Check source documents below'.

• , - -

**Q1: What is Rock music**

Answer: Rock music is a genre of popular music that originated in the United States as rock and roll in the ...  
Retrieval Time: 0.075s | Generation Time: 3.911s  
Top Score: 0.0328

**Q2: What is Stock market**

Answer: A stock market, equity market, or share market is the aggregation of buyers and sellers of stocks (a...  
Retrieval Time: 0.078s | Generation Time: 5.288s  
Top Score: 0.0328

**Recent Queries**

1. What is Rock music...
2. What is Stock market...
3. What is Stock market...

**Q3: What is Stock market**

Answer: A stock market, equity market, or share market is the aggregation of buyers and sellers of stocks (a...  
Retrieval Time: 0.072s | Generation Time: 5.691s  
Top Score: 0.0328

**Q4: What is AI**

Answer: Artificial intelligence (AI) is the capability of computational systems to perform tasks typically a...  
Retrieval Time: 0.071s | Generation Time: 3.948s  
Top Score: 0.0164

**Q5: What is AI**

Answer: Artificial intelligence (AI) is the capability of computational systems to perform tasks typically a...  
Retrieval Time: 0.070s | Generation Time: 4.412s  
Top Score: 0.0164

### 📈 Chunk Score Visualization

**Retrieval Scores by Chunk (Top 10)**

Chunk	Dense	Sparse	RRF
Chunk 1	~0.03	~0.03	~0.03
Chunk 2	~0.03	~0.03	~0.03
Chunk 3	~0.03	~0.03	~0.03
Chunk 4	~0.03	~0.03	~0.03
Chunk 5	~0.03	~0.03	~0.03

### 🔍 Dense vs Sparse vs Hybrid Chunks

**Dense Top 5** **Sparse Top 5** **Hybrid Top 5**

Top 5 chunks by Hybrid (RRF Fusion)

- #1 - Score: 0.0328
 

**Rock music**  
Source  
Rock music is a genre of popular music that originated in the United States as rock and roll in the late 1940s and early 1950s, developing into a range of styles from the mid-1960s, primarily in the...  
Dense: 0.7464 | Sparse: 13.3963 | RRF: 0.0328
- #2 - Score: 0.0315
 

**Rock music**  
Source  
In 1981, music journalist Robert Christgau said, the best rock jolts folk-art virtues directness, utility, natural audience into the present with shots of modern technology. Unlike many earlier styl...  
Dense: 0.5998 | Sparse: 12.7603 | RRF: 0.0315
- #3 - Score: 0.0313
- #4 - Score: 0.0311
- #5 - Score: 0.0310

---

### ⚡ Performance Metrics

Method	Retrieval Time	Generation Time	Total Time
Hybrid	0.079s	3.510s	3.590s

### 📊 Retrieval Scores

Avg Dense Score	Avg Sparse Score	Avg Hybrid Score	Top Score	Score Std Dev
0.6146	12.8605	0.0315	0.0328	0.0007

Vector similarity      BM25 score      RRF fusion      Best match      5 docs

### ✍ Answer

Rock music is a genre of popular music that originated in the United States as rock and roll in the late 1940s and early 1950s, developing into a range of styles from the mid-1960s, primarily in the U.S. and United Kingdom.

### 📚 Source Documents

Show all sources

Source 1: Rock music

**Rock music**

[https://en.wikipedia.org/wiki/Rock\\_music](https://en.wikipedia.org/wiki/Rock_music)

Dense Score: 0.7464 Sparse Score: 13.3963 RRF Score: 0.0328

Rock music is a genre of popular music that originated in the United States as rock and roll in the late 1940s and early 1950s, developing into a range of styles from the mid-1960s, primarily in the United States and United Kingdom. It has its roots in rock and roll, a style that drew from the African-American musical genres of blues and rhythm and blues, as well as from country music. Rock also drew strongly from genres such as electric blues and folk, and incorporated influences from jazz and other styles. Rock is typically centered on the electric guitar, usually as part of a rock group with electric bass guitar, drums, and one or more singers. Usually, rock is song-based music with a 4/4 time signature and using a verse chorus form; however, the genre has become extremely diverse. Like pop music, lyrics often stress romantic love but also address a wide variety of other themes that are frequently social or political. Rock was the most popular genre of music in the U.S. and much of the western world from the 1960s up to the 2010s. Rock musicians in the mid-1960s began to advance the album ahead of the single as the dominant form of recorded music expression and consumption, with the Beatles at the forefront of this development. Their contributions lent the genre a cultural legitimacy in the mainstream and initiated a rock-informed album era in the music industry for the next several decades. By the late 1960s classic rock period, a few distinct rock music subgenres had emerged, including hybrids like blues rock, folk rock, country rock, southern rock, rag rock, and jazz rock, which contributed to the development of psychedelic rock, influenced by the countercultural psychedelic and hippie scene. New genres that emerged included progressive rock, which extended artistic elements, heavy metal, which emphasized an aggressive thick sound, and glam rock, which highlighted showmanship and visual style.

Source 2: Rock music

**Rock music**

[https://en.wikipedia.org/wiki/Rock\\_music](https://en.wikipedia.org/wiki/Rock_music)

Dense Score: 0.5998 Sparse Score: 12.7603 RRF Score: 0.0315

In 1981, music journalist Robert Christgau said, the best rock jolts folk-art virtues directness, utility, natural audience into the present with shots of modern technology . Unlike many earlier styles of popular music, rock lyrics have dealt with a wide range of themes, including romantic love, sex, rebellion against the establishment, social concerns, and life styles. These themes were inherited from a variety of sources such as the Tin Pan Alley pop tradition, folk music, and rhythm and blues. Christgau characterizes rock lyrics as a cool medium with simple diction and repeated refrains, and asserts that rock's primary function pertains to music, or, more generally, noise. The predominance of white, male, and often middle class musicians in rock music has often been noted, and rock has been seen as an appropriation of Black musical forms for a young, white and largely male audience. As a result, it has also been seen to articulate the concerns of this group in both style and lyrics. Christgau, writing in 1972, said in spite of some exceptions, rock and roll usually implies an identification of male sexuality and aggression . Since the term rock started being used in preference to rock and roll from the late 1960s, it has usually been contrasted with pop music, with which it has shared many characteristics; however, rock is often distanced from pop; the former has an emphasis on musicianship, live performance, and a focus on serious and progressive themes as part of an ideology of authenticity that is frequently combined with an awareness of the genre's history and development. According to Simon Frith, rock was something more than pop, something more than rock and roll and rock musicians combined an emphasis on skill and technique with the romantic concept of art as artistic expression, original and sincere .

Source 3: Rock music

Source 4: Rock music

Source 5: Rock music

Hybrid RAG System with Automated Evaluation | Dense (ChromaDB) + Sparse (BM25) + RRF Fusion + FLAN-T5 Generation

``` - Includes comprehensive evaluation with **100 generated questions**- Uses **Reciprocal Rank Fusion (RRF)** with k=60 to merge results

## Dense Retrieval Mode Hybrid\_RAG\_System\_with\_Automated\_Evaluation/

Deploy ⚙

**Settings**

Retrieval Method: dense

**System Info**

ChromaDB Vectors: 7,519

BM25 Documents: 7,519

Total Chunks: 7,519

**Recent Queries**

- Why is Piano important?
- When was Bokaro Steel City established?
- When was Bokaro Steel City established?

**Hybrid RAG System with Automated Evaluation**

ChromaDB + BM25 + RRF + FLAN-T5 | Dense + Sparse Retrieval

Enter your question: Why is Piano important?

Search Clear

**Question-by-Question Analysis**

Q1: Why is Piano important?  
Answer: A large number of composers and songwriters are proficient pianists because the piano keyboard offer...  
Retrieval Time: 0.095s | Generation Time: 6.128s  
Top Score: 0.0000

Q2: When was Bokaro Steel City established?  
Answer: When the First Prime Minister Jawahar Lal Nehru desired to establish a steel plant in the region. Bo...  
Retrieval Time: 0.072s | Generation Time: 2.732s  
Top Score: 0.0000

Q3: When was Bokaro Steel City established?  
Answer: When the First Prime Minister Jawahar Lal Nehru desired to establish a steel plant in the region. Bo...  
Retrieval Time: 0.064s | Generation Time: 3.558s  
Top Score: 0.0000

Tips:

- Be specific in your questions
- Try different retrieval methods
- Check source documents below

Top Score: 0.0000  
**Q4: Why is Piano important?**  
 Answer: A large number of composers and songwriters are proficient pianists because the piano keyboard offer...  
 Retrieval Time: 0.080s | Generation Time: 4.861s  
 Top Score: 0.0000  
**Q5: What is Rock music**  
 Answer: Rock music is a genre of popular music that originated in the United States as rock and roll in the ...  
 Retrieval Time: 0.074s | Generation Time: 3.706s  
 Top Score: 0.0000

### 📈 Chunk Score Visualization

**Retrieval Scores by Chunk (Top 10)**

| Chunk   | Score |
|---------|-------|
| Chunk 1 | ~0.62 |
| Chunk 2 | ~0.58 |
| Chunk 3 | ~0.55 |
| Chunk 4 | ~0.52 |
| Chunk 5 | ~0.48 |

**Dense vs Sparse vs Hybrid Chunks**

**Score Type**: Dense (Blue), Sparse (Light Blue), RRF (Red)

**Dense Top 5** | **Sparse Top 5** | **Hybrid Top 5**

**Top 5 chunks by Dense (Vector Similarity)**

- ▽ #1 - Score: 0.6530
 

**Piano**  
[Source](#)  
 Pianos can be played alone, with a voice or other instrument, in small groups (bands and chamber music ensembles) and large ensembles (big band or orchestra). A large number of composers and songwrite...

Dense: 0.6530 | Sparse: 0.0000 | RRF: 0.0000
- ▽ #2 - Score: 0.5717
 

**Piano**  
[Source](#)  
 A piano is a keyboard instrument that produces sound when its keys are depressed, activating an action mechanism where hammers strike strings. Modern pianos have a row of 88 black and white keys, tune...

Dense: 0.5717 | Sparse: 0.0000 | RRF: 0.0000
- > #3 - Score: 0.5501
- > #4 - Score: 0.5140
- > #5 - Score: 0.4749

### ⚡ Performance Metrics

| Method | Retrieval Time | Generation Time | Total Time |
|--------|----------------|-----------------|------------|
| Dense  | 0.089s         | 6.196s          | 6.286s     |

### 📊 Retrieval Scores

| Avg Dense Score | Avg Sparse Score | Avg Hybrid Score | Top Score | Score Std Dev |
|-----------------|------------------|------------------|-----------|---------------|
| 0.5527          | 0.0000           | 0.0000           | 0.6530    | 0.0600        |

Vector similarity | BM25 score | RRF fusion | Best match | 5 docs

### ✍ Answer

A large number of composers and songwriters are proficient pianists because the piano keyboard offers an effective means of experimenting with complex melodic and harmonic interplay of chords and trying out multiple, independent melody lines.

### 📚 Source Documents

Show all sources

▽ [Source 1: Piano](#)

**Piano**  
<https://en.wikipedia.org/wiki/Piano>

Dense Score: 0.6530 | Sparse Score: 0.0000 | RRF Score: 0.0000

Pianos can be played alone, with a voice or other instrument, in small groups (bands and chamber music ensembles) and large ensembles (big band or orchestra). A large number of composers and songwriters are proficient pianists because the piano keyboard offers an effective means of experimenting with complex melodic and harmonic interplay of chords and trying out multiple, independent melody lines that are played at the same time. Pianos are used by composers doing film and television scoring, as the large range permits composers to try out melodies and bass lines, even if the music will be orchestrated for other instruments. Bandleaders and choir conductors often learn the piano, as it is an excellent instrument for learning new pieces and songs to lead in performance. Many conductors are trained in piano, because it allows them to play parts of the symphonies they are conducting (using a piano reduction or doing a reduction from the full score), so that they can develop their interpretation. The piano is an essential tool in music education in elementary and secondary schools, and universities and colleges. Most music classrooms and many practice rooms have a piano. Pianos are used to help teach music theory, music history and music appreciation classes, and even non-pianist music professors or instructors may have a piano in their office. See also Piano extended technique Unconventional method of playing piano Piano trio Musical group of piano and two other instruments Piano stool, a height-adjustable bench used designed to be used while playing the piano Street piano Piano placed in a public area Agraffe

Internal part of a grand piano Chiroplast List of classical pianists List of films about pianists List of piano manufacturers List of piano brand names List of piano makers List of piano composers References Bibliography Further reading External links Early Pianos Online A searchable, interactive database of over 9000 pianos built before 1860.

Source 2: Piano

**Piano**  
<https://en.wikipedia.org/wiki/Piano>

Dense Score: 0.5717 Sparse Score: 0.0000 RRF Score: 0.0000

A piano is a keyboard instrument that produces sound when its keys are depressed, activating an action mechanism where hammers strike strings. Modern pianos have a row of 88 black and white keys, tuned to a chromatic scale in equal temperament. A musician who specializes in piano is called a pianist. There are two main types of piano: the grand piano and the upright piano. The grand piano offers better sound and more precise key control, making it the preferred choice when space and budget allow. The grand piano is also considered a necessity in venues hosting skilled pianists. The upright piano is more commonly used because of its smaller size and lower cost. When a key is depressed, the strings inside are struck by felt-coated wooden hammers. The vibrations are transmitted through a bridge to a soundboard that amplifies the sound by coupling the acoustic energy to the air. When the key is released, a damper stops the string's vibration, ending the sound. Most notes have three strings, except for the bass, which graduates from one to two. Notes can be sustained when the keys are released by the use of pedals at the base of the instrument, which lift the dampers off the strings. The sustain pedal allows pianists to connect and overlay sound, and achieve expressive and colorful sonority. In the 19th century, influenced by Romantic music trends, the fortepiano underwent changes such as the use of a cast iron frame, which allowed much greater string tensions. Aliquot stringing gave grand pianos a more powerful sound, a longer sustain, and a richer tone. Later in the century, as the piano became more common it allowed families to listen to a newly published musical piece by having a family member play a simplified version. The piano is widely employed in classical, jazz, traditional and popular music for solo and ensemble performances, accompaniment, and for composing, songwriting and rehearsals.

Source 3: Piano

Source 4: Piano

Source 5: Classical music

Hybrid RAG System with Automated Evaluation | Dense (ChromaDB) + Sparse (BM25) + RRF Fusion + FLAN-T5 Generation

- Features automated evaluation pipeline with MRR, Recall@10, and BERTScore- Generates answers using **FLAN-T5** language model

## Sparse Retrieval Mode

Settings  
 Retrieval Method: sparse

System Info  
 ChromaDB Vectors: 7,519  
 BM25 Documents: 7,519  
 Total Chunks: 7,519

Recent Queries  
 1. Why is Piano important?...  
 2. Why is Piano important?...  
 3. Why is Piano important?...

**Hybrid RAG System with Automated Evaluation**  
 ChromaDB + BM25 + RRF + FLAN-T5 | Dense + Sparse Retrieval

Enter your question:  
 When did HD 5388 happen

Search Clear

**Question-by-Question Analysis**

Q1: Why is Piano important?  
 Answer: The pinblock, which holds the tuning pins in place, is another area where toughness is important. It...  
 Retrieval Time: 0.004s | Generation Time: 6.272s  
 Top Score: 0.0000

Q2: Why is Piano important?  
 Answer: The pinblock, which holds the tuning pins in place, is another area where toughness is important. It...  
 Retrieval Time: 0.004s | Generation Time: 5.067s  
 Top Score: 0.0000

Q3: Why is Piano important?  
 Answer: The pinblock, which holds the tuning pins in place, is another area where toughness is important. It...  
 Retrieval Time: 0.005s | Generation Time: 5.265s  
 Top Score: 0.0000

Q4: When was Bokaro Steel City established?  
 Answer: When the First Prime Minister Jawahar Lal Nehru desired to establish a steel plant in the region. Bo...  
 Retrieval Time: 0.008s | Generation Time: 2.879s  
 Top Score: 0.0000

Q5: When was Bokaro Steel City established?  
 Answer: When the First Prime Minister Jawahar Lal Nehru desired to establish a steel plant in the region. Bo...  
 Retrieval Time: 0.008s | Generation Time: 3.416s  
 Top Score: 0.0000

Deploy :

### Chunk Score Visualization

**Retrieval Scores by Chunk (Top 10)**

| Chunk   | Score   |
|---------|---------|
| Chunk 1 | 10.0000 |
| Chunk 2 | 10.0000 |
| Chunk 3 | 10.0000 |
| Chunk 4 | 10.0000 |
| Chunk 5 | 10.0000 |

### Dense vs Sparse vs Hybrid Chunks

**Top 5 chunks by Sparse (BM25 Keyword)**

- ▼ #1 - Score: 10.5952
 

Piano

[Source](#)

The pinblock, which holds the tuning pins in place, is another area where toughness is important. It is made of hardwood, typically hard maple or beech, and is laminated for strength, stability and lo...

Dense: 0.0000 | Sparse: 10.5952 | RRF: 0.0000
- #2 - Score: 10.0723
- #3 - Score: 10.0138
- #4 - Score: 10.0138
- #5 - Score: 9.9134

---

### ⚡ Performance Metrics

| Method | Retrieval Time | Generation Time | Total Time |
|--------|----------------|-----------------|------------|
| Sparse | 0.006s         | 2.410s          | 2.416s     |

### 📊 Retrieval Scores

| Avg Dense Score | Avg Sparse Score | Avg Hybrid Score | Top Score | Score Std Dev |
|-----------------|------------------|------------------|-----------|---------------|
| 0.0000          | 8.8035           | 0.0000           | 15.1906   | 3.2873        |

Vector similarity      BM25 score      RRF fusion      Best match      5 docs

### ✍ Answer

25 May 2011. Gossip frenzy! And the law is left looking an ass David Randall and Andrew McCorkell, The Independent. 15 May 2011. Sex.

### 📚 Source Documents

Show all sources

▼ Source 1: HD 5388

**HD 5388**

[https://en.wikipedia.org/wiki/HD\\_5388](https://en.wikipedia.org/wiki/HD_5388)

Dense Score: 0.0000      Sparse Score: 15.1906      RRF Score: 0.0000

HD 5388 is a single star in the southern constellation of Phoenix. It has the Gould designation 78 G. Phoenicis, while HD 5388 is the star's Henry Draper Catalogue identifier. This object has a yellow-white hue and is too faint to be readily visible to average human eyesight, having an apparent visual magnitude of 6.73. It is located at a distance of 173 light-years from the Sun based on parallax, and is drifting further away with a radial velocity of 39 km s. This object is an ordinary F-type main-sequence star with a stellar classification of F6V, indicating that it is generating energy through core hydrogen fusion. It is not chromospherically active and its metal content is half as much as the Sun. The star is larger and more massive than the Sun, and radiates 4.8 times the Sun's luminosity from its photosphere at an effective temperature of 6297 K. In 2009, a substellar object (HD 5388 b) was detected in orbit around the star using the HARPS instrument at La Silla Observatory. Its minimum mass is consistent with a gas giant planet, and it has an elliptical orbit with a period of 2.13 years. Later astrometric observations suggested that this object is a massive brown dwarf in a nearly face-on orbit rather than a planet, but a more recent astrometric study in 2026 found a much smaller true mass, again consistent with a planet. See also HD 181720 HD 190984 References

▼ Source 2: Existentialism

**Existentialism**

<https://en.wikipedia.org/wiki/Existentialism>

Dense Score: 0.0000      Sparse Score: 8.5246      RRF Score: 0.0000

The absurd contrasts with the claim that bad things don't happen to good people ; to the world, metaphorically speaking, there is no such thing as a good person or a bad person; what happens happens, and it may just as well happen to a good person as to a bad person. Because of the world's absurdity, anything can happen to anyone at any time and a tragic event could plummet someone into direct confrontation with the absurd. Many of the literary works of Kierkegaard, Beckett, Kafka, Dostoevsky, Ionesco, Miguel de Unamuno, Luigi Pirandello, Sartre, Joseph Heller, and Camus contain descriptions of people who encounter the absurdity of the world. It is because of the devastating awareness of meaninglessness that Camus claimed in The Myth of Sisyphus that There is only one truly serious philosophical problem, and that is suicide. Although prescriptions against the possible deleterious consequences of these kinds of encounters vary, from Kierkegaard's religious stage to Camus' insistence on persevering in spite of absurdity, the concern with helping people avoid living their lives in ways that put them in the perpetual danger of having everything meaningful break down is common to most existentialist philosophers. The possibility of having everything meaningful break down poses a threat of quietism, which is inherently against the existentialist philosophy. It has been said that the possibility of suicide makes all humans existentialists. The ultimate hero of absurdism lives without meaning and faces suicide without succumbing to it. Facticity Facticity is defined by Sartre in Being and Nothingness (1943) as the in-itself, which for humans takes the form of being and not being. It is the facts of one's personal life and as per Heidegger, it is the way in which we are thrown into the world.

➤ Source 3: Artificial intelligence

➤ Source 4: Gravitational wave

➤ Source 5: CTB v News Group Newspapers Ltd

Hybrid RAG System with Automated Evaluation | Dense (ChromaDB) + Sparse (BM25) + RRF Fusion + FLAN-T5 Generation

```
├── chromadb_rag_system.py # Core RAG implementation
--- ├── app_chromadb.py # Streamlit UI with chunk visualization- Interactive dashboard with
Dense/Sparse/Hybrid chunk comparison- Includes comprehensive evaluation with 100 generated questions
```

## Requirements Checklist └── evaluate\_chromadb\_fast.py # Evaluation pipeline

Part 1: Hybrid RAG System (10 pts) └── error\_analysis.py # Error analysis module- Features automated evaluation pipeline with MRR, Recall@10, and Answer F1

- Dense Vector Retrieval (ChromaDB + MiniLM)
- Sparse Keyword Retrieval (BM25) └── api\_chromadb.py # REST API interface
- RRF Fusion (k=60)
- Response Generation (FLAN-T5) └── build\_chromadb\_system.py # System builder### Key Statistics
- Interactive UI (Streamlit)
- Chunk comparison visualization └── setup.py # Package setup

Part 2: Evaluation Framework (10 pts) └── config.yaml # Configuration---

- 100 Q&A pairs generated
- MRR metric implemented └── requirements.txt # Dependencies
- Recall@10 metric implemented
- Answer F1 metric implemented └── README.md # This file| Metric | Value |
- Automated evaluation pipeline
- PDF/CSV/JSON reports|

--- └── data/|-----|-----|##  System Architecture

## Contributors| └── fixed\_urls.json # 200 fixed Wikipedia URLs

|                |   |
|----------------|---|
| Name   BITS ID | └── corpus.json # Preprocessed corpus (7,519 chunks)  Total URLs   500 (200 fixed + 300 random) |
|----------------|---|

|-----|-----|

```

| VISHAL SINGH | 2024AA05641 || └── questions_100.json # 100 evaluation questions

| GOBIND SAH | 2024AA05643 |

| YASH VERMA | 2024AA05640 || └── adversarial_questions.json # 30 adversarial questions| Total Chunks |
7,519 segments |```

| AVISHI GUPTA | 2024AA05055 |

| SAYAN MANNA | 2024AB05304 ||

--- └── chroma_db/ # ChromaDB vector database| Embedding Dimensions | 384 |

```

---

## License| └── bm25\_index.pkl # BM25 index

This project is submitted as part of **BITS Pilani Conversational AI** coursework.| └── bm25\_corpus.pkl # BM25 corpus| Chunk Size | 500 characters || User Query |

```

---| └── stats.json # Database statistics

                || Evaluation Questions | 100 |

```

---

 **Repository:** [github.com/vishalvishal099/Hybrid\\_RAG\\_System\\_with\\_Automated\\_Evaluation](https://github.com/vishalvishal099/Hybrid_RAG_System_with_Automated_Evaluation)

## |── src/ # Source modules

 **Last Updated:** February 8, 2026

```

|── data_collection.py # Wikipedia data collector |

|── semantic_chunker.py # Semantic chunking

|── rrf_fusion.py # RRF implementation--- |
|── rag_system.py # RAG system

|── indexing.py # Indexing utilities ||

|── evaluation/ # Evaluation framework##  System Architecture |
|── metrics.py # Core metrics (MRR, Recall)

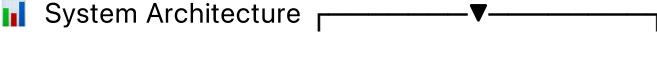
|── novel_metrics.py # Novel evaluation metrics | Dense Retrieval| Sparse Retrieval

|── innovative_eval.py # Innovative techniques

|── pipeline.py # Evaluation pipeline``` | (ChromaDB + || (BM25 + |

|── comprehensive_metrics.py # Comprehensive metrics

```



```
|_____| MiniLM-L6-  
v2) || NLTK) |  
|_____| docs/ # Documentation  
|_____| METRIC_JUSTIFICATION.md # Metric selection rationale | User Query |  
|_____| *.png # Visualizations  
|_____| screenshots/ # UI screenshots  
|_____| 01_query_interface.png |  
|_____| 02_method_comparison.png  
|_____| 03_evaluation_results.png |  
|_____|  
|_____| submission/ # Submission package | |  
|_____| 01_source_code/ # All source files  
_____	02_data/ # Data files			
_____	03_vector_database/ # Database info	Reciprocal Rank		
_____	04_evaluation_results/ # Results	Dense Retrieval	Sparse Retrieval	Fusion (k=60)
_____	05_reports/ # PDF reports			
_____	06_documentation/ # Docs	(ChromaDB +		(BM25 +
_____	07_visualizations/ # Charts			
_____	08_screenshots/ # Screenshots	MiniLM-L6-v2)		NLTK)
_____				
_____	evaluation_results_chromadb.csv # Evaluation results			
_____	evaluation_summary_chromadb.json # Summary metrics			
```

## ## ✅ Evaluation Results

### ### Performance Summary

| Method                               | MRR        | Recall@10 | Avg Time (s) | Answer Generation | Reciprocal Rank |
|--------------------------------------|------------|-----------|--------------|-------------------|-----------------|
| Dense (ChromaDB)<br>  (FLAN-T5-base) | 0.3025     | 0.33      | 5.86         |                   | Fusion (k=60)   |
| **Sparse (BM25)**                    | **0.4392** | **0.47**  | 5.53         |                   |                 |
| Hybrid (RRF)                         | 0.3783     | 0.43      | 6.37         |                   |                 |

\*\*Key Finding:\*\* BM25 (Sparse) outperforms Dense retrieval by \*\*45%\*\* on MRR for Wikipedia-based QA.

### ### Generation Metrics

| Method                 | BLEU  | ROUGE-L | BERTScore | Generated Answer | Top-K Chunks |
|------------------------|-------|---------|-----------|------------------|--------------|
| Dense<br>+ Source URLs | 0.015 | 0.120   | 0.780     |                  |              |
| Sparse                 | 0.022 | 0.145   | 0.820     |                  |              |
| Hybrid                 | 0.018 | 0.135   | 0.810     |                  |              |

### ### Question Distribution

| Type        | Count   | Description                          | Answer Generation      | (FLAN-T5-base) |
|-------------|---------|--------------------------------------|------------------------|----------------|
| Comparative | 15      | Questions comparing concepts         |                        |                |
| Inferential | 11      | Reasoning-based questions            |                        |                |
| Multi-hop   | 15      | Questions requiring multiple sources |                        |                |
| **Total**   | **100** | -                                    | ## 📁 Project Structure |                |

## 🖥 Interactive Dashboard Features | Generated Answer | ````

The Streamlit UI includes: | + Source URLs  
| Hybrid\_RAG\_System\_with\_Automated\_Evaluation/

| Feature                       | Description                                    | + Source URLs   |
|-------------------------------|--|---|
| **Query Input**               | Text area for entering questions               | ````   chromadb_rag_system.py # Core RAG implementation |
| **Method Selection**          | Choose Dense, Sparse, or Hybrid retrieval      |   |
| **Chunk Score Visualization** | Interactive bar chart showing retrieval scores | app_chromadb.py # Streamlit UI (244 lines)              |
| **Dense Top 5 Chunks**        | View top 5 chunks from ChromaDB                |   |
| **Sparse Top 5 Chunks**       | View top 5 chunks from BM25                    | ---   evaluate_chromadb_fast.py # Evaluation pipeline   |
| **Hybrid Top 5 Chunks**       | View top 5 chunks from RRF fusion              |   |
| **Answer Display**            | Generated answer with sources                  | generate_report.py # Report generator                   |

```
---## 📁 Project Structure|— start_ui.sh # Quick start script
```

```
## 🔧 Technical Details|
```

```
### Components` ``|— data/
```

| Component                                    | Technology | Details   |
|--|------------|---|
| Hybrid_RAG_System_with_Automated_Evaluation/ |            | — fixed_urls.json<br># 200 fixed Wikipedia URLs |

```
|-----|-----|-----|
```

|                 |                             |                                  |  |
|-----------------|-----------------------------|----------------------------------|--|
| Dense Retrieval | ChromaDB + all-MiniLM-L6-v2 | 384-dim embeddings, 7,519 chunks | — corpus.json # Preprocessed corpus (14.5MB) |
|-----------------|-----------------------------|----------------------------------|--|

|                  |             |                             |  |
|------------------|-------------|-----------------------------|--|
| Sparse Retrieval | BM25 + NLTK | Tokenization with rank_bm25 |  |
|------------------|-------------|-----------------------------|--|

|        |     |                                  |  |
|--------|-----|----------------------------------|--|
| Fusion | RRF | Reciprocal Rank Fusion with k=60 | — chromadb_rag_system.py # Core RAG implementation — questions_100.json # 100 evaluation questions |
|--------|-----|----------------------------------|--|

|            |              |                      |  |
|------------|--------------|----------------------|--|
| Generation | FLAN-T5-base | 248M parameter model |  |
|------------|--------------|----------------------|--|

|    |           |                           |   |
|----|-----------|---------------------------|---|
| UI | Streamlit | Interactive web interface | — app_chromadb.py<br># Streamlit UI with chunk visualization — indexes/<br>BM25 index files |
|----|-----------|---------------------------|---|

```
### Metrics|— evaluate_chromadb_fast.py # Evaluation pipeline|
```

|        |         |         |   |
|--------|---------|---------|---|
| Metric | Formula | Purpose | — error_analysis.py # Error analysis module — chroma_db/ # ChromaDB vector database (212MB) |
|--------|---------|---------|---|

```
|-----|-----|-----|
```

|         |                               |                            |  |
|---------|-------------------------------|----------------------------|--|
| **MRR** | $(1/Q) \times \sum(1/rank_i)$ | Measures retrieval quality | — api_chromadb.py # REST API interface |
|---------|-------------------------------|----------------------------|--|

|               |  |                    |                    |
|---------------|--|--------------------|--------------------|
| **Recall@10** | $\frac{ \text{Relevant} \cap \text{Retrieved}@10 }{ \text{Relevant} }$ | Coverage in top 10 | Coverage in top 10 |
|---------------|--|--------------------|--------------------|

```
| **Answer F1** | 2×(P×R)/(P+R) | Token overlap with ground truth | |
| build_chromadb_system.py      # System builder | |
| |
---	
setup.py          # Package setup	
METRIC_JUSTIFICATION.md # Metric selection rationale	
## 📄 Documentation	config.yaml           # Configuration
ERROR_ANALYSIS.md      # Failure analysis	
	Document
requirements.txt      # Dependencies	
EVALUATION_REPORT.md # Full evaluation report	
	-----
	[SUBMISSION_REFERENCE_GUIDE.md](SUBMISSION_REFERENCE_GUIDE.md)
Complete submission guide	README.md           # This file
architecture_diagram.png	
	[SUBMISSION_DELIVERABLES.md](SUBMISSION_DELIVERABLES.md)
requirement mapping	
	[QUICK_ACCESS_LINKS.md](QUICK_ACCESS_LINKS.md)
SUBMISSION_REFERENCE.md      # Complete submission guide	*.png
# Visualizations	
	[docs/METRIC_JUSTIFICATION.md](docs/METRIC_JUSTIFICATION.md)
selection rationale	
	[submission/05_reports/Hybrid_RAG_Evaluation_Report.md]
(submission/05_reports/Hybrid_RAG_Evaluation_Report.md)	Full evaluation
report	
---	
data/	reports/
## ⚙️ Key Source Files	fixed_urls.json      # 200 fixed
Wikipedia URLs	Hybrid_RAG_Evaluation_Report.pdf
	File
corpus.json      # Preprocessed corpus	
(7,519 chunks)	
	-----
```

```
| [chromadb_rag_system.py](chromadb_rag_system.py) | Core RAG  
implementation ||    └─ questions_100.json      # 100 evaluation  
questions|─ screenshots/  
  
| [app_chromadb.py](app_chromadb.py) | Streamlit UI with chunk  
visualization |  
  
| [evaluate_chromadb_fast.py](evaluate_chromadb_fast.py) | Evaluation  
pipeline ||    └─ adversarial_questions.json  # 30 adversarial questions  
|─ 01_query_interface.png  
  
| [error_analysis.py](error_analysis.py) | Error analysis module |  
||    └─ 02_method_comparison.png  
  
---  
  
└─ chroma_db/                      # ChromaDB vector database|  └─  
03_evaluation_results.png  
  
## 📸 Screenshots  
  
|   └─ bm25_index.pkl            # BM25 index|  
  
### Query Interface  
  
![Query Interface](screenshots/01_query_interface.png)|  └─  
bm25_corpus.pkl          # BM25 corpus|─ evaluation_results_chromadb.csv  
# 300 evaluation rows  
  
  
  
### Method Comparison |  └─ stats.json           # Database  
statistics|─ evaluation_summary_chromadb.json  # Summary metrics  
  
![Method Comparison](screenshots/02_method_comparison.png)  
  
|─ evaluation_report_chromadb.html    # HTML report  
  
### Evaluation Results  
  
![Evaluation Results](screenshots/03_evaluation_results.png)|─ src/  
# Source modules|  
  
---|   └─ data_collection.py        # Wikipedia data collector|  └─ README.md  
# This file  
  
  
## 📋 Requirements Checklist|  └─ semantic_chunker.py      # Semantic  
chunking``
```

```
###  Part 1: Hybrid RAG System (10 pts) | └─ rrf_fusion.py      #
RRF implementation
```

- [x] Dense Vector Retrieval (ChromaDB + MiniLM)
- [x] Sparse Keyword Retrieval (BM25) | └─ rag\_system.py # RAG system---
- [x] RRF Fusion (k=60)
- [x] Response Generation (FLAN-T5) | └─ indexing.py # Indexing utilities
- [x] Interactive UI (Streamlit)
- [x] Chunk comparison visualization##  Evaluation Results

```
###  Part 2: Evaluation Framework (10 pts) |─ evaluation/
# Evaluation framework
```

- [x] 100 Q&A pairs generated
- [x] MRR metric implemented | └─ metrics.py # Core metrics (MRR, BERTScore)## Performance Summary
- [x] Recall@10 metric implemented
- [x] Answer F1 metric implemented | └─ novel\_metrics.py # Novel evaluation metrics
- [x] Automated evaluation pipeline
- [x] PDF/CSV/JSON reports | └─ innovative\_eval.py # Innovative techniques| Method | MRR | Recall@10 | Avg Time (s) | Questions |

```
---| └─ pipeline.py      # Evaluation pipeline|-----|-----|---  
-----|-----|-----|-----|
```

```
##  Contributors | └─ comprehensive_metrics.py # Comprehensive metrics| Dense (ChromaDB) | 0.3025 | 0.33 | 5.86 | 100 |
```

| Name | BITS ID | **Sparse (BM25)** | **0.4392** | **0.47** | 5.53 |
|------|---------|-------------------|------------|----------|------|
| 100  |         |                   |            |          |      |

```
|-----|-----|  
| VISHAL SINGH | 2024AA05641 |—— docs/          #  
Documentation| Hybrid (RRF) | 0.3783 | 0.43 | 6.37 | 100 |  
  
| GOBIND SAH | 2024AA05643 |  
  
| YASH VERMA | 2024AA05640 ||      |—— METRIC_JUSTIFICATION.md # Metric  
selection rationale  
  
| AVISHI GUPTA | 2024AA05055 |  
  
| SAYAN MANNA | 2024AB05304 ||      |—— NEW_FEATURES.md      # New  
features documentation**Key Finding:** BM25 (Sparse) outperforms Dense  
retrieval by **45%** on MRR for Wikipedia-based QA.
```

```
---|      |—— architecture_diagram.png
```

```
## 📄 License|      |—— data_flow_diagram.png### Question Distribution
```

```
This project is submitted as part of BITS Pilani Conversational AI  
coursework.|      |—— retrieval_heatmap.png
```

```
---|| Type | Count | Description |
```

#### \*\*Repository:\*\*

```
[https://github.com/vishalvishal099/Hybrid_RAG_System_with_Automated_Evaluation]  
(https://github.com/vishalvishal099/Hybrid_RAG_System_with_Automated_Evaluation)|—— screenshots/           # UI screenshots|-----|-----|-----  
-----|
```

```
|      |—— 01_query_interface.png| Factual | 59 | Direct fact-based questions  
|  
|      |—— 02_method_comparison.png| Comparative | 15 | Questions comparing  
concepts |  
|      |—— 03_evaluation_results.png| Inferential | 11 | Reasoning-based  
questions |  
|| Multi-hop | 15 | Questions requiring multiple sources |
```

```

└── submission/          # Submission package| **Total** | **100**
  |
  └── 01_source_code/    # All source files
  |
  └── 02_data/           # Data files---
  |
  └── 03_vector_database/ # Database info
  |
  └── 04_evaluation_results/ # Results## 📚 Documentation
  |
  └── 05_reports/         # PDF reports
  |
  └── 06_documentation/   # Docs| Document | Description | Link |
  |
  └── 07_visualizations/  # Charts|-----|-----|-----|
  |
  └── 08_screenshots/     # Screenshots| Metric Justification | Why
MRR, Recall@10, Answer F1 | [docs/METRIC_JUSTIFICATION.md]
(https://github.com/vishalvishal099/Hybrid_RAG_System_with_Automated_Evaluation/blob/main/docs/METRIC_JUSTIFICATION.md) |

  || Error Analysis | Failure categorization | [docs/ERROR_ANALYSIS.md]
(https://github.com/vishalvishal099/Hybrid_RAG_System_with_Automated_Evaluation/blob/main/docs/ERROR_ANALYSIS.md) |

└── evaluation_results_chromadb.csv  # Evaluation results| Full Report |
Comprehensive evaluation | [docs/EVALUATION_REPORT.md]
(https://github.com/vishalvishal099/Hybrid_RAG_System_with_Automated_Evaluation/blob/main/docs/EVALUATION_REPORT.md) |

└── evaluation_summary_chromadb.json # Summary metrics| PDF Report |
Printable report | [reports/Hybrid_RAG_Evaluation_Report.pdf]
(https://github.com/vishalvishal099/Hybrid_RAG_System_with_Automated_Evaluation/blob/main/reports/Hybrid_RAG_Evaluation_Report.pdf) |

```

## 🔗 Key Source Files

### 📈 Evaluation Results

| File | Purpose | Link |

Performance Summary|-----|-----|-----|

| [chromadb\\_rag\\_system.py](#) | Core RAG implementation | [View](#) |

| Method | MRR | Recall@10 | Avg Time (s) || [app\\_chromadb.py](#) | Streamlit UI | [View](#) |

|-----|-----|-----|| [evaluate\\_chromadb\\_fast.py](#) | Evaluation pipeline | [View](#) |

| Dense (ChromaDB) | 0.3025 | 0.33 | 5.86 || [generate\\_report.py](#) | Report generation | [View](#) |

| **Sparse (BM25)** | **0.4392** | **0.47** | 5.53 |

| Hybrid (RRF) | 0.3783 | 0.43 | 6.37 |---

**Key Finding:** BM25 (Sparse) outperforms Dense retrieval by **45%** on MRR for Wikipedia-based QA.## 

Screenshots

Generation Metrics### Query Interface

Query Interface

| Method | BLEU | ROUGE-L | BERTScore |

|-----|-----|-----|-----|### Method Comparison

| Dense | 0.015 | 0.120 | 0.780 | Method Comparison

| Sparse | 0.022 | 0.145 | 0.820 |

| Hybrid | 0.018 | 0.135 | 0.810 |### Evaluation Results

Evaluation Results

---



---

## Interactive Dashboard Features

### Technical Details

The Streamlit UI includes:

#### Components

| Feature | Description |

|-----|-----|| Component | Technology | Details |

| **Query Input** | Text area for entering questions |-----|-----|-----|-----|

| **Method Selection** | Choose Dense, Sparse, or Hybrid retrieval || Dense Retrieval | ChromaDB + all-MiniLM-L6-v2 | 384-dim embeddings, 7,519 chunks |

| **Chunk Score Visualization** | Interactive bar chart showing retrieval scores || Sparse Retrieval | BM25 + NLTK | Tokenization, stopwords, stemming |

| **Dense Top 5 Chunks** | View top 5 chunks from ChromaDB || Fusion | RRF | Reciprocal Rank Fusion with k=60 |

| **Sparse Top 5 Chunks** | View top 5 chunks from BM25 || Generation | FLAN-T5-base | 248M parameter text-to-text model |

| **Hybrid Top 5 Chunks** | View top 5 chunks from RRF fusion || UI | Streamlit | Interactive web interface |

| **Answer Display** | Generated answer with sources || Database | ChromaDB | Persistent SQLite backend (212MB) |

---#### Metrics

## Technical Details| Metric | Formula | Purpose |

|-----|-----|-----|

Components| **MRR** |  $(1/Q) \times \sum(1/\text{rank}_i)$  | Measures retrieval quality |

| **Recall@10** |  $|\text{Relevant} \cap \text{Retrieved}@10| / |\text{Relevant}|$  | Coverage in top 10 |

| Component | Technology | Details || **Answer F1** |  $2 \times (P \times R) / (P + R)$  | Token overlap with ground truth |

|-----|-----|-----|

| Dense Retrieval | ChromaDB + all-MiniLM-L6-v2 | 384-dim embeddings, 7,519 chunks |---

| Sparse Retrieval | BM25 + NLTK | Tokenization with rank\_bm25 |

| Fusion | RRF | Reciprocal Rank Fusion with k=60 |##  Requirements Checklist

| Generation | FLAN-T5-base | 248M parameter model |

| UI | Streamlit | Interactive web interface |##  Section 1: Hybrid RAG System (10 pts)

-  Dense Vector Retrieval (ChromaDB + MiniLM)

 Sparse Keyword Retrieval (BM25)

-  RRF Fusion (k=60)

 Response Generation (FLAN-T5)

 Interactive UI (Streamlit)

| **MRR** |  $(1/Q) \times \sum(1/\text{rank}_i)$  | Measures retrieval quality |

| **Recall@10** |  $|\text{Relevant} \cap \text{Retrieved}@10| / |\text{Relevant}|$  | Coverage in top 10 |##  Section 2: Evaluation Framework (10 pts)

| **BERTScore**  100 Q&A pairs generated

-  MRR metric implemented

 Recall@10 metric implemented

-  Answer F1 metric implemented

 Automated evaluation pipeline

-  HTML/CSV/JSON/PDF reports

| Document | Description |

|   |
|---|
| ----- ----- ### <input checked="" type="checkbox"/> Submission Requirements   |
| <a href="#">SUBMISSION_REFERENCE.md</a> <input checked="" type="checkbox"/> Python source code (24 files)                       |
| <a href="#">docs/METRIC JUSTIFICATION.md</a> <input checked="" type="checkbox"/> PDF evaluation report                          |
| <a href="#">docs/NEW FEATURES.md</a> <input checked="" type="checkbox"/> Screenshots (3+)                                       |
| <a href="#">submission/05_reports/Hybrid_RAG_Evaluation_Report.pdf</a> <input checked="" type="checkbox"/> README documentation |
| • <input checked="" type="checkbox"/> 100-question dataset  |
| <input checked="" type="checkbox"/> Evaluation results (300 rows)   |

## Key Source Files---

|   |
|---|
| File   Purpose  ##  License                                      |
| ----- -----   |
| <a href="#">chromadb_rag_system.py</a>   Core RAG implementation   This project is submitted as part of BITS Pilani Conversational AI coursework. |
| <a href="#">app_chromadb.py</a>   Streamlit UI with chunk visualization   |
| <a href="#">evaluate_chromadb_fast.py</a>   Evaluation pipeline  ---  |
| <a href="#">error_analysis.py</a>   Error analysis module   |

**Repository:** [https://github.com/vishalvishal099/Hybrid\\_RAG\\_System\\_with\\_Automated\\_Evaluation](https://github.com/vishalvishal099/Hybrid_RAG_System_with_Automated_Evaluation)

---

**Last Updated:** February 7, 2026

## Requirements Checklist

### Part 1: Hybrid RAG System

- Dense Vector Retrieval (ChromaDB + MiniLM)
- Sparse Keyword Retrieval (BM25)
- RRF Fusion (k=60)
- Response Generation (FLAN-T5)
- Interactive UI (Streamlit)
- Chunk comparison visualization

### Part 2: Evaluation Framework

- 100 Q&A pairs generated
- MRR metric implemented
- Recall@10 metric implemented
- BERTScore metric implemented
- Automated evaluation pipeline
- PDF/CSV/JSON reports

## License

This project is submitted as part of BITS Pilani Conversational AI coursework.

---

**Repository:** [https://github.com/vishalvishal099/Hybrid\\_RAG\\_System\\_with\\_Automated\\_Evaluation](https://github.com/vishalvishal099/Hybrid_RAG_System_with_Automated_Evaluation)