The Death Star Group presents:

GlassBOT

AI-powered career advice built on glassdoor™ review RAG.

Margarita Jonathan Jesse Blair

Purpose

01 Parses job review data from Glassdoor and your resume.

02 Offers career recommendations on companies or roles.

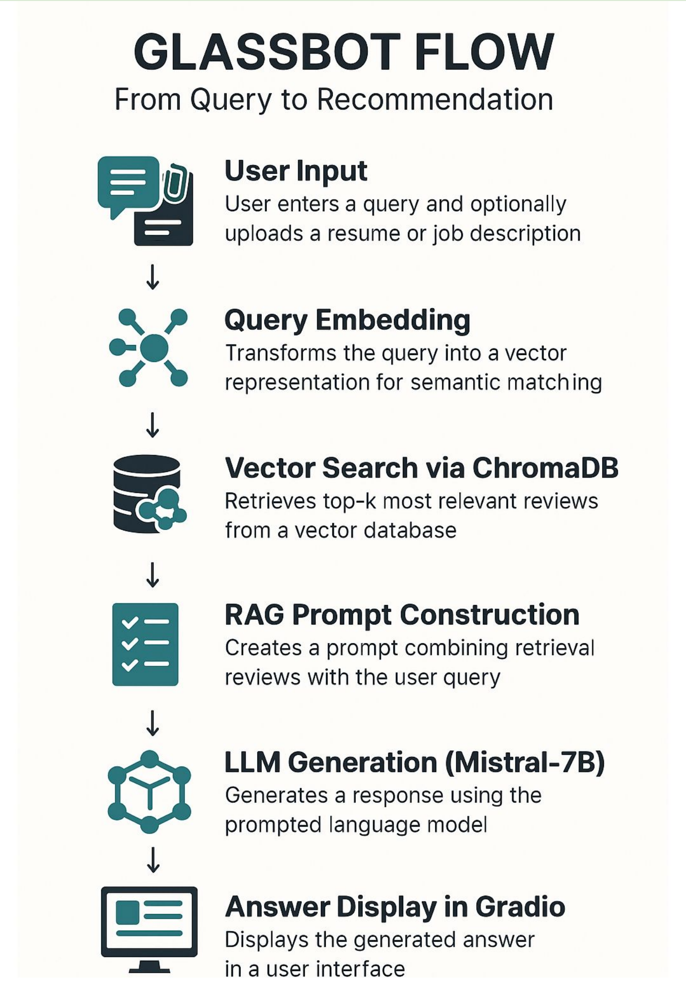
03 Provides pros and cons based on real employee sentiment.

04 Can be filtered by attributes like industry, job role, desired benefits, company size and more.

Overview of GlassBOT

Component	Tool Used	Description
Data Source	Glassdoor Job Reviews CSV + optional resumes	Review text extracted from “pros” and “cons”
Embedding Model	`all-MiniLM-L6-v2` (Sentence Transformers)	Converts text into semantic vectors
Vector Database	ChromaDB (via `chromadb`)	Stores and retrieves relevant reviews by similarity
LLM for Response	Mistral-7B (4-bit via HuggingFace)	Generates career recommendations based on retrieved context
UI	Gradio	User interface to upload documents or ask career-related questions
Environment	Google Colab + Google Drive	Team-accessible development & storage

Workflow


GLASSBOT FLOW
From Query to Recommendation
User Input
User enters a query and optionally uploads a resume or job description
Query Embedding
Transforms the query into a vector representation for semantic matching
Vector Search via ChromaDB
Retrieves top-k most relevant reviews from a vector database
RAG Prompt Construction
Creates a prompt combining retrieval reviews with the user query
LLM Generation (Mistral-7B)
Generates a response using the prompted language model
Answer Display in Gradio
Displays the generated answer in a user interface

Dataset: Glassdoor Job Reviews 2

(Sourced from Kaggle)

01 9.9M reviews x 19 columns

02 Text Columns: Title, pros, cons, job, employee status

03 Numerical Columns: 1-5 rating, Career Opportunities, Compensation and Benefits , Work/Life Balance

04 Categorical Columns: Recommendation on leadership, business outlook and firm general

05 Unique text column: firm link

Data Refinement

Company Name Extraction

- The dataset did not explicitly have Firm Name
- The firm_link column containing a URL-safe string for each company's Glassdoor page
- Wrote a helper function extracts the name and stores it in a new column called firm_name
- 34,369 unique company names

Initial Cleanup

Drop Advice and Index columns (Mostly null values)

Drop all rows with nulls

9.9M reviews to 2.7M (1,728 companies)

Bias Control

1,000 reviews minimum

Random sampling of 1,000 reviews for remaining companies

581,000 reviews from 581 unique companies

File Reduction (not needed)

Chunked
Smaller files ≤ 25,000 rows

Smaller File
Random 500,000 rows
500 reviews

Model Creation and RAG

• Assists users in exploring Glassdoor Job Reviews.
• Provides insights on company reviews, pros and cons, and employee satisfaction.

Key Features

- Company Reviews: Summarizes pros and cons from employee feedback.
- Job Recommendations: Suggests job opportunities based on user interests.
- Diversity Ratings: Shows company ratings on diversity and culture.
- User Engagement: Allows users to upload a resume for analysis.

Data Sources

- Pulls data from the Glassdoor database, including:
- Employee reviews
- Pros and cons information

Purpose

- A valuable tool for job seekers, employers, and career advisors.
- Provides easy access to job market insights for informed decision-making.

Live Demo of the Chatbot

Key Learnings and Results

Improved Answer Quality with Larger LLM: Transitioning from a smaller LLM to **Mistral 7B** significantly improved the relevance and depth of responses, particularly in summarizing sentiment and providing actionable insights.

Scalable Document Handling: Successfully parsed and ingested varied formats (PDF, DOCX, XLSX, CSV), creating a modular ingestion pipeline.

I/O Bottlenecks with Google Drive: Reading/writing directly to Google Drive was a major performance constraint, especially during bulk document ingestion. Using cloud drive storage for frequent read/write operations should be minimized in RAG pipelines.

Optimize for Local Temp Storage: Utilizing local ephemeral storage (`/tmp`) in Google Colab yielded **dramatic performance improvements** in both ingestion and retrieval stages.

Memory Efficiency Matters: LLM inference within Colab has strict resource constraints; quantization and batching strategies were key to staying within available limits.

Prompt Engineering is Critical: Through iterative testing, we evaluated multiple prompt variations—ranging from concise to complex multi-step instructions—to identify a structure that consistently yielded accurate, insightful, and context-aware responses. A well-crafted prompt proved essential to guiding the LLM toward useful and aligned output.

Next steps

1 Upload multiple files

1 Multi-turn convo

1 Change LLM to something more current, larger

1 Add search option to scrape job boards for relevant open listings

1 Use full dataset of 9.9M reviews but weight Firm Name to eliminate bias

1 More data exploration – bias on reviews from current employees and former employees etc.

Thank you!

What a great 6 months!
Many thanks to ALL staff
and classmates. HAGS!