# **🧠 Explainable RAG Pipeline: Concept-based & Mechanistic Interpretability**

## **🚀 Overview**

This project demonstrates a Google Colab-compatible Proof of Concept (PoC) that adds interpretability to a Retrieval-Augmented Generation (RAG) pipeline using two complementary methods:

* **Concept-Based Explanation**
* **Mechanistic Interpretability**

The goal is to make RAG outputs more **transparent**, **traceable**, and **human-aligned**, helping both researchers and practitioners understand *why* a model responded in a certain way.

## **📦 Setup**

# ✅ Run in Google Colab or local environment with:

pip install torch transformers matplotlib seaborn

pip install numpy==1.23.5 # for compatibility

pip install transformer-lens==1.9.0 --no-deps

## **🧩 1. Concept-Based Explanation**

### **✅ Description**

This module uses **sentence embeddings** (e.g., BERT) to measure how much a generated response aligns with predefined human concepts such as:

* Positive Sentiment
* Negative Sentiment
* Technical Language
* Casual Language

### **⚙️ How It Works**

* Average the embedding of multiple examples per concept.
* Generate embedding for the new text.
* Compute **cosine similarity** between the text and each concept vector.
* Visualize the relevance via bar charts.

### **🔍 Output Example**

Explaining: "I really didn't enjoy this movie, it was boring and too long."

Concept Relevance Scores:

negative\_sentiment: 0.82

casual\_language: 0.61

technical\_language: 0.23

positive\_sentiment: 0.01

## **⚙️ 2. Mechanistic Interpretability**

### **✅ Description**

This module uses **TransformerLens** to uncover which parts of the transformer model (e.g., GPT2) are responsible for generating specific outputs.

### **🔬 What It Does**

* Tokenizes prompt
* Runs forward pass while capturing internal activations
* Visualizes **attention flow** (e.g., which word focused on which)
* Can extend to **logit lens**, **activation patching**, **head ablation**

### **🔍 Output Example**

A heatmap showing attention from layer 0, head 0:

[I] → I

[really] → didn't

[didn't] → enjoy

[enjoy] → movie

## **🧠 Comparison Table**

| **Aspect** | **Concept-Based Explanation** | **Mechanistic Interpretability** |
| --- | --- | --- |
| Focus | Semantic concept alignment | Internal component behavior |
| Level | Input/output space | Hidden layers, heads, neurons |
| Tools | BERT, cosine similarity | TransformerLens, cache tracing |
| Output | Concept scores, bar chart | Attention heatmap, token tracing |
| Purpose | Human-level explanation | Model reasoning deconstruction |

## **📊 Usage**

* Use the **Concept Explainer** to check what *kind* of knowledge the model reflects.
* Use **Mechanistic Interpretability** to identify *which token, head, or layer* influenced that generation.

## **📁 File Structure (in Colab)**

📦 Explainable-RAG

├── concept\_explainer.py # Concept-based explanation module

├── mechanistic\_interpreter.py # TransformerLens-based analyzer

├── concept\_examples.json # Concept definitions

├── README.md # Project overview

## **🙋‍♂️ Author & Contact**

**Bunni** Full-stack AI/ML Developer  
 PoC built for Explainable AI research using LLMs + RAG  
 Reach out for collaboration or expansion.

bunnibunni269@gmail.com

## **🔗 Related**

* [HuggingFace Transformers](https://huggingface.co/transformers)
* [Concept Activation Vectors (TCAV)](https://arxiv.org/abs/1711.11279)