# LangChain LLM Benchmarking Documentation

## 1. Introduction

This project provides a benchmarking framework for evaluating different LLM models using LangChain. It supports both locally deployed models and API-based models (OpenAI, Cohere, Anthropic).

## 2. Project Structure

The project consists of the following key files:

* • benchmark.py - Main script to run LLM benchmarking.
* • test\_cases.py - Contains test cases for NL to SQL, SQL to NL, and general tasks.
* • models.py - Loads locally deployed models using Hugging Face.
* • models2.py - Loads API-based models from OpenAI, Cohere, and Anthropic.
* • gpu\_monitor.py - Tracks real-time GPU usage during inference.
* • db\_connection.py - Provides a reusable function to connect to PostgreSQL.
* • schema.sql - Creates database schema for SQL-related test cases.
* • insert\_data.sql - Populates the database with random test data.
* • setup\_postgres.sh - Shell script to set up PostgreSQL database.

## 3. Setup Guide

Follow these steps to set up the project:

### Step 1: Set up PostgreSQL

Run the following command to set up the database:

./setup\_postgres.sh

### Step 2: Run the Benchmarking Tool

Run the benchmarking script with a selected model and test category:

python benchmark.py

### Step 3: Connect to the Database

Test the database connection:

python test\_postgres.py

### Step 4: Switch Between Local and API-Based Models

To use locally deployed models, ensure models are loaded from models.py:

from models import get\_model

To use API-based models (OpenAI, Cohere, Anthropic), switch to:

from models2 import get\_model

## 4. Supported Test Categories

* This framework supports various test categories:
* • nl\_to\_sql - Converts natural language queries to SQL.
* • sql\_to\_nl - Converts SQL queries into natural language explanations.
* • nl\_to\_sql\_jp - Japanese version of NL to SQL.
* • sql\_to\_nl\_jp - Japanese version of SQL to NL.
* • general\_knowledge - General question-answering.
* • math - Mathematical reasoning and problem-solving.
* • coding - Programming-related questions.

## 5. Running Benchmarks

To benchmark a specific model, use:

python benchmark.py <model\_name> <test\_category>

### Example Usage

Run a benchmark on GPT-4 for general knowledge questions:

python benchmark.py openai-gpt4 general\_knowledge

## 6. Download Location

All benchmarking results are displayed in the console and logged for further analysis.