

Improving Question-Answering Performance over Tabular Data with LangChain and LangSmith

By Harrison Chase



What is LangChain

Framework for building context-aware reasoning applications

Core, modular building blocks

- Integrations with LLMs, vector stores, embedding models
- Implementations of prompt templates, text splitters, agents, tools, memory, etc

End-to-end chains

- Assembling these modular components for specific use cases

What is LangSmith

Platform for managing context-aware reasoning applications

Debugging, logging, monitoring

Testing, evaluation, datasets

All connected

The Problem

One big use case is question answering over tabular data

Most of the focus in the industry has been on unstructured data

Tabular data actually harder because it can have aspects of unstructured data but people also expect to be able to ask analytical questions

For a while, people had been asking for better support for this

The Solution

Collect Data

Define evaluator



Benchmark current solution

Experiment

Benchmark new solution

Repeat until satisfied

Collect Data



Ask the CSV App

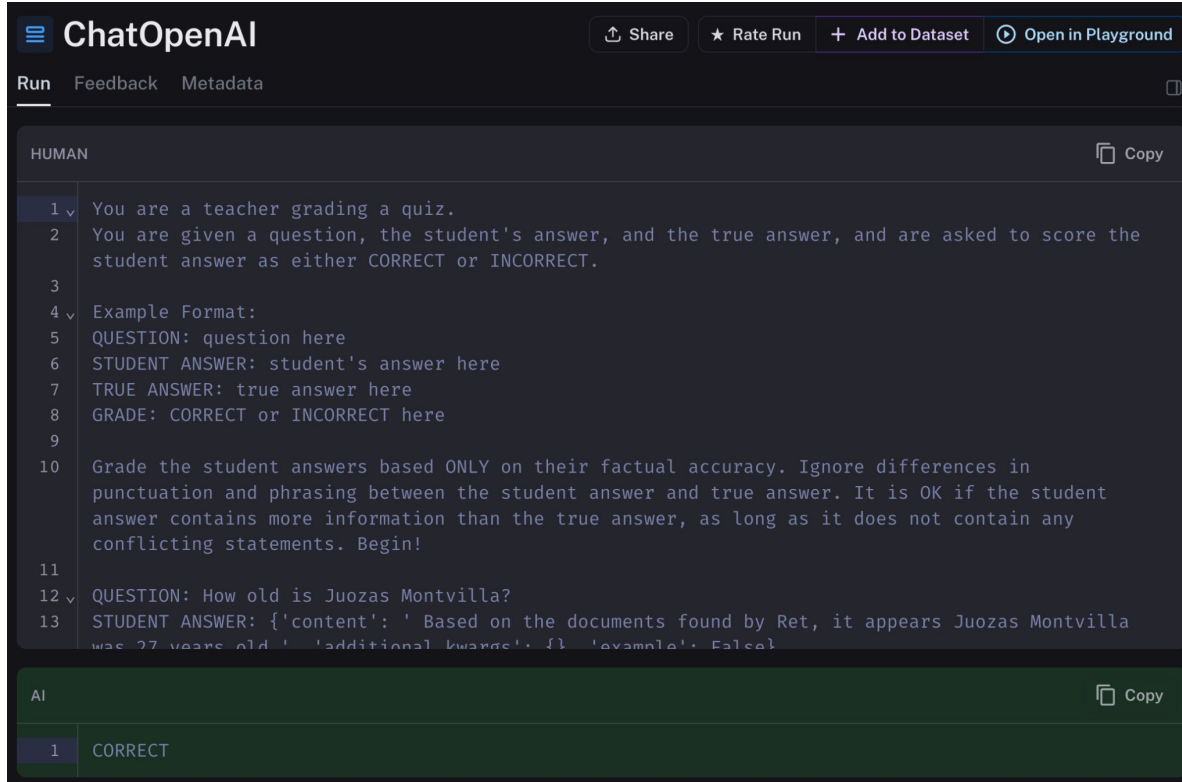
Most 'question answering' applications run over unstructured text data. But a lot of the data in the world is tabular data! This is an attempt to create an application using [LangChain](#) to let you ask questions of data in tabular format. For this demo application, we will use the Titanic Dataset. Please explore it [here](#) to get a sense for what questions you can ask. Please leave feedback on well the question is answered, and we will use that improve the application!

Enter your question:

The person in cabin C128 was Mr. Fletcher Fellows Williams-Lambert.

Feedback:

Define Evaluator



The screenshot shows the ChatOpenAI interface. At the top, there are navigation buttons: 'Share', 'Rate Run', 'Add to Dataset', and 'Open in Playground'. Below these are tabs for 'Run', 'Feedback', and 'Metadata'. The main content area is divided into two sections: 'HUMAN' and 'AI'. The 'HUMAN' section contains a list of instructions and a question. The 'AI' section shows the model's response.

ChatOpenAI Share Rate Run Add to Dataset Open in Playground

Run Feedback Metadata

HUMAN Copy

- 1 You are a teacher grading a quiz.
- 2 You are given a question, the student's answer, and the true answer, and are asked to score the student answer as either CORRECT or INCORRECT.
- 3
- 4 Example Format:
- 5 QUESTION: question here
- 6 STUDENT ANSWER: student's answer here
- 7 TRUE ANSWER: true answer here
- 8 GRADE: CORRECT or INCORRECT here
- 9
- 10 Grade the student answers based ONLY on their factual accuracy. Ignore differences in punctuation and phrasing between the student answer and true answer. It is OK if the student answer contains more information than the true answer, as long as it does not contain any conflicting statements. Begin!
- 11
- 12 QUESTION: How old is Juozas Montvilla?
- 13 STUDENT ANSWER: {'content': ' Based on the documents found by Ret, it appears Juozas Montvilla was 27 years old.', 'additional_kwarg': {}, 'example': False}

AI Copy

- 1 CORRECT

Old Solution

For CSV:

- Pandas Agent (can write and run Pandas code)

For SQL:

- SQL Chain (can write and run SQL)

Some Pain Points

- Multihop questions
- Questions about text columns
 - Even simple ones like name or song title
- Multihop questions involving text columns
- (sidenote) poorly formatted data

New Solution

Agent with multiple tools

- Vectorstore - index over textual data
- Pandas/SQL - perform analytic queries

Agent allows it to perform multiple steps

- Really important for high cardinality textual columns!