Bryan McKinney Presents - A Minimal Conversational Prototype

Agenda

- The Problem at Hand
- Multiple Ways to Approach
- My Solution
- Results
- Future Plans
- Questions

The Problem At Hand:

Not everyone can be a data expert

It is often times difficult, or even time consuming, for people that are not inherently using the data on a daily basis to truly understand the data present to them, even if presented in a well done BI tool. A minimal conversational prototype could allow for an end user to query the data in a more intuitive way with little to no understanding of how a traditional BI tool or query language operates.

Multiple Ways to Approach

01

In A Perfect World

Given a good sized budget and time, Ideally this project would involve 3rd party LLM's (Large Language Models). This would include a decent amount of time to prepare the data for these LLM's. Annotating the data, classifying the data and validating the data. Once the data has been prepped, we connect the data using the LLM API to query the data as asked.

02

Budget Contrains Us All

Given constraints of money and time, a solid middle gound option would be to create some data pre-processing layer to constantly monitor and correct the incoming data. For this task we could use an already in use tool such as dbt to ensure data integrity. Then we could create an in-house LLM to query the data. This LLM would not be nearly as robust as a 3rd party option, but can still deliver favorable results.

03

The MVP (Minimal Viable Product)

Given the constraints of only having a limited data set, and a few hours to complete the task, we can create a very simplistic NLP (Natural Language Processing) model to allow us to query the data in much less robust way. This process will have issues around phrasing of the question and will have to be manually updated when new cases are added. However the pre-processing of the data remains similar to approach 02.

My Solution: Steps to create the MVP

1. **Data Discovery -** A deep dive to understand the data presented and document obvious transforms/ translations that need to occur
2. **Data Cleaning -** Typically this would be completed as part of the BI transfer layer
3. **NLP to Understand the question -** Use off the self NLP libraries in Python to help understand what the question is asking and then assign the queries a type (Counting, Filtering, Aggregate…) as well as determine what columns of data we need to use.
4. **Code Functions to perform the tasks** - Using python to create multiple functions to define query types (Counting, Filtering, Aggregate…).
5. **Testing -** Run different query types and examples to ensure the appropriate actions are taken in the code.
6. **Deployment -** Running the code in Git CodeSpace to ensure end to end functionality

My Solution: Data Discovery / Cleaning

During Discovery there were a few data points that needed to be addressed:

1. Payment Frequency needed to be encoded
2. All Dollar signs ("$") needed to be removed
3. All commas needed to be removed
4. All columns that are representative of dollar amounts need to be cast as float

For cleaning it was decided to just adjust the columns within the python dataframe. Given more time I would have created a pre-processing script to sit on top of the data directly and only ingest the clean data.

```
sample_data = pd.read_csv('saas_customer_data.csv')

# Data Cleaning steps of sample_data
sample_data.columns = [col.lower().replace(' ', '_') for col in sample_data.columns]
# Harcoded clean up based on current data set, need to add in more dynamic cleaning
# for future state
sample_data['new_freq'] = sample_data['payment_frequency'].case_when(
    [
        (sample_data['payment_frequency'] == 'Monthly', 12),
        (sample_data['payment_frequency'] == 'Quarterly', 4),
        (sample_data['payment_frequency'] == 'Annually', 1)
    ]
)
sample_data['billings'] = sample_data['billings'].str.replace('$', '')
sample_data['billings'] = sample_data['billings'].str.replace(',', '').astype(float)
sample_data['contacts'] = sample_data['contacts'].str.replace(',', '').astype(float)
sample_data['yearly_revenue'] = sample_data['billings'] * sample_data['new_freq']
sample_data = sample_data.sort_values(by='yearly_revenue', ascending=False)
```

My Solutions: How the Code Works

The Question

A Question is Asked

The Translator

NLP determines the type of question and columns used

The Data Handler

Looking at the Data the correct calculations are ran

The UI

The code give back a user friendly response

Example:

The Question

How many Customers are there?

The Translator

Intent: "count"
Column: "customer"

The Data Handler

Counts all customers in the data set

The UI

"There are a total of 100 Customers."

Results - Pretty good for a MVP

The MVP can accuratley compute 5 different intents, and more than 15 distinct functions.

Given a bit more time, and larger dataset, it would be fairly easy to increase these numbers dramatically and have the tool be even more robust.

Intent	Query Type
Count	how many, how much, count, quantity, number of, total
Filter	where, with, having, >, < , equals, between, contains
Aggregate	avg, mean, sum , total, max, min, top, correlation
Group	group by, per, each, for every, by category, breakdown
Sort	sort, order, rank, top, bottom, highest, lowest

Results - The Proof

```
=== Minimal Conversational Prototype Demo ===

Available columns:
Available columns: customer_id, company_name, payment_frequency, clos
s, advanced_api_access, api_rate_limit_increase, advanced_security, h
ular_users, monthly_active_users, paid_users, contacts, workflows, ir
ight_users, all_answers_contacts, all_resolutions, new_freq, yearly_r

================================================

Q: How many distinct customers do we have?
A: Total number of records: 100
Result: 100

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

Q: What are the top 10 company_name based on yearly_revenue?
A: Calculated top for column 'company_name'
Result:
33    Company 34
52    Company 53
31    Company 32
8      Company 9
37    Company 38
34    Company 35
9     Company 10
70    Company 71
7      Company 8
38    Company 39
Name: company_name, dtype: object
```

```
Q: Is there a correlation between contacts and workflows?
A: Calculated correlation for column 'contacts'
Result:
some correlation between variables

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

Q: What is the average yearly revenue by customer?
A: Calculated average for column 'yearly_revenue'
Result: 4437.52
```

Future Plans

01

Data Updates

Of course step one would be to connect this to a DB. However I would also plan to use a service like Data Robot to ensure the data was kept clean and consistent. This would include data validation steps to ensure data was always actionable.

02

LLMs

Once the data is coming in clean and classified I would use a LLM such as Claude.ai to replace the bulk of the MVP code. This would be in the form of API calls in which Claude.ai would have access to the data and be able to provide with more in depth natural language abilities without the need to code manually for new cases.

03

Use for Exec Team Queries

I would sit this project on top of the data that exec teams would care to utilize so they can ask questions of the data such as revenue projections given certain parameters , or top 5 reasons for customer churn, or which employee has the highest customer success scores.

Questions

Pitch

Want to make a presentation like this one?

Start with a fully customizable template, create a beautiful deck in minutes, then easily share it with anyone.

Create a presentation (It’s free)