StatGPT

An AI-based SDMX Query Building Assistant

OCTOBER 29, 2023

Jeff Danforth (IMF)
Ilya Gorelik (EPAM)
Outline

- Objectives
- Business capabilities
- Architecture and how it works
- Technical challenges
- Lessons learned
- Demo
- Next steps
StatGPT Objectives
StatGPT Capabilities: Generative

- Generative
  - Generate SDMX Queries
  - Generate SQL Queries
  - Generate REST Queries
  - Generate Code Snippets and Scripts

I would like GDP indicators for Colombia and its neighbors last 10 years and forecast for 2023

TIME_PERIOD: 'startPeriod': '2013', 'endPeriod': '2023'
COUNTRY:
  - [233] Colombia
  - [218] Bolivia
  - [253] El Salvador
  - [278] Nicaragua
  - [293] Peru
  - [299] Venezuela
  - [309] Eastern Caribbean Currency Union
INDICATOR:
  - [NGDP_RPCH] Gross domestic product, constant prices, National Currency, percent change
  - [NGDPD] Gross domestic product in U.S. dollars
  - [NGDPDPC] Gross domestic product, current prices, U.S. dollars, per capita
FREQUENCY:
  - [A] Annual
StatGPT Capabilities: Visualization

- Visualization
- Render Table Data
- Render Charts
- Render Queries
- Render Code Snippets
StatGPT Capabilities: Integration

- Integrate natural language query builder into existing dissemination tools

- Visualize data using existing interface

- Analyze data using existing Excel add-in
How it works

1. Process the query through embedding model to semantically represent original request as a query vector

2. Pass the vector to the vector DB that contains vectors for all indicators, countries and groups known to the system

3. Retrieve top-k relevant contexts measured by cosine distance between the query and embedded items in the knowledge base

4. Add relevant contexts to original user query and send it to LLM for parsing

5. Build SDMX query based on the response from LLM

6. Send SDMX query to SDMX API to get the data

7. Generate response using the retrieved data
Technical Challenges

- Embeddings distance not always gives all relevant items, specifically for longer prompts

- Inclusion of multiple indicators and countries creates a large context for LLM (affects cost and speed of execution)

- Queries like “Give me data for France and all its neighbors” require a pre-flight LLM request to retrieve an actual list of neighbors

- Keeping only necessary and sufficient context during the conversation with LLM is not trivial

- Indicator prioritization (key indicators) is not supported by embeddings and must be implemented separately
Lessons Learned

- Concurrency is limited to capabilities of LLM (LLM is a bottleneck)
- LLM load-balancing is crucial for robust implementation (multiple accounts might be required)
- GPT-4 is an optimum model for the task (GPT-4-32K is slower and more expensive, GPT-3.5 is not that precise)
- Exposing the tool to internal users is much easier than making it available to public (cost and performance considerations)
DEMO
Give me monthly inflation data for Jamaica and 5 similar countries for the last 10 years

**User:** Give me monthly inflation data for Jamaica and 5 similar countries for the last 10 years

**Assistant:** I have updated the form based on your request for monthly inflation data for Jamaica and 5 similar countries for the last 10 years. Here's the updated form:


Please let me know if you need any further changes or if this is good to proceed.
StatGPT

Click to display/hide charts

Click to display/hide Excel input

Click to display/hide SDMX URL


Click to display/hide code snippet

```python
provider = sdmx.Request('IMF_RES')
data_msg = provider.data('WEO', key=['COUNTRY': [233, 218, 253, 268, 278, 293, 299]])
```
Next Steps

- More training of the model with Fund users
- Implementing within IMF IT architecture
- Data transformations
- International collaboration?
QUESTIONS?
Architecture

- Integrates into existing IMF architecture
- Explanation of what AI Dial is?