MASTERARBEIT | MASTER'S THESIS

Titel | Title

Chatbot for Inflation Data Analysis

verfasst von | submitted by

Polina Samoilenko BSc

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2025

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 645
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Data Science
Betreut von | Supervisor:	Univ.-Prof. Torsten Möller PhD
Mitbetreut von | Co-Supervisor:	Laura Koesten MSc Ph.D.

Kurzfassung

In der heutigen datengetriebenen Welt ist das Verständnis komplexer wirtschaftlicher Daten, wie beispielsweise der Inflation, von entscheidender Bedeutung, jedoch für viele Zielgruppen oft eine Herausforderung. Obwohl Werkzeuge wie der IHS Price Monitor interaktive Visualisierungen von Inflationsmetriken bereitstellen, haben Nutzer häufig Schwierigkeiten, komplexe Diagramme zu navigieren und die Daten zu interpretieren. Ziel dieser Arbeit war die Entwicklung eines Chatbots, der von Large Language Models (LLMs) unterstützt wird, um diese Herausforderungen zu bewältigen und Benutzerpräferenzen zwischen dem Chatbot und dem IHS-Dashboard für den Zugriff auf Inflationsinformationen zu vergleichen. Der Chatbot wurde unter Verwendung von Teilen desselben Datensatzes wie das IHS-Dashboard entwickelt und nutzte ein Multi-Agenten-System, bei dem jeder Agent auf eine spezifische Aufgabe spezialisiert war. Dieser Ansatz ermöglichte es dem Chatbot, mit Antworten in natürlicher Sprache zu reagieren und Visualisierungen als Teil seiner Antworten zu generieren. Zur Bewertung der Leistung des Chatbots und der Benutzerpräferenzen wurde eine empirische Studie mit 10 TeilnehmerInnen durchgeführt. Jeder Teilnehmer/Jede Teilnehmerin erledigte Aufgaben mit beiden Tools. Die Ergebnisse zeigten, dass die TeilnehmerInnen den Chatbot als benutzerfreundlicher empfanden als das IHS-Dashboard. Die Kombination aus textuellen und visuellen Informationen wurde als effektiv und nützlich wahrgenommen. Die Studie identifizierte zudem Verbesserungsbereiche und lieferte wertvolle Erkenntnisse zur Weiterentwicklung des Chatbots. Während der aktuelle Chatbot als praktischer Assistent für die Beantwortung inflationsbezogener Anfragen dient, könnten weitere Optimierungen ihn zu einem vollständig funktionalen Werkzeug für den Zugriff auf Inflationsdaten machen.

Abstract

In today's data driven world, understanding complex economic data, such as inflation, is essential but often challenging for many audiences. Despite the availability of tools like the IHS Price Monitor, which provides interactive visualizations of inflation metrics, users may struggle with navigating complex charts and interpreting the data. This thesis aimed to develop a chatbot powered by Large Language Models (LLMs) to address these challenges and to compare user preferences between the chatbot and the IHS Dashboard for accessing inflation information. The chatbot was built using parts of the same dataset as the IHS Dashboard and utilized a multi-agent system, where each agent specialized in a specific task. This approach enabled the chatbot to respond with natural language answers and generate visualizations as part of its replies. To evaluate the chatbot's performance and user preferences, an empirical study involving 10 participants was conducted. Each participant completed tasks using both tools. The results demonstrated that participants found the chatbot more user-friendly than the IHS Dashboard. The combination of textual and visual information was perceived as both effective and useful. The study also identified areas for improvement, offering valuable insights for refining the chatbot. While the current chatbot serves as a practical assistant for addressing inflation-related queries, further enhancements could establish it as a fully functional tool for accessing inflation data.

Contents

1 Motivation

In today's data-driven world, making complex economic data understandable to a broad public is very important yet challenging, especially regarding inflation. Various sources, including newspapers, books, and online articles, aim to offer detailed insights into inflation in a clear format and language. Additionally, several dashboards have been developed to display economic trends through interactive charts. One such tool, created by researchers at the Institute for Advanced Studies[1] and named IHS Price Monitor[2], showcases data across various countries with metrics such as inflation rates, harmonized index of consumer prices, producer prices, sector contributions to total inflation, etc. While this dashboard is handy for economists, analysts, students, and anyone interested in economics, it may not be intuitive for some users, who might struggle with finding specific information, navigating through complex charts and numbers, or understanding their meaning. These challenges highlight the ongoing need to make this data more comprehensible and to simplify access to necessary information.

My master's thesis project aims to address these challenges by developing a chatbot powered by Large Language Models, or LLMs for short. LLMs are known for being able to understand and generate natural language. This capability is primarily achieved by training on large amounts of internet-sourced data, including websites, books, and code repositories. This enabled them to perform various tasks, including generating code [HBK+21] successfully. Building on this technology, my project develops a chatbot that leverages large language models known as GPT and Claude to make inflation data more accessible and understandable. The chatbot uses the same dataset as the IHS Dashboard to respond to user queries and create or update visualizations based on their requests. This approach lets users quickly obtain and understand inflation information through a simple conversational interface.

[1]https://www.ihs.ac.at/
[2]https://inflation-en.ihs.ac.at/

2 Problem Statement

Current visualization tools utilizing LLMs, such as LIDA [Dib23], Chat2Viz [MS23], and VIDS [HKS23], provide innovative solutions for data visualization but often lack conversational capabilities for answering data-related questions and are not specifically optimized for complex datasets like inflation data used in the IHS Price Monitor. OpenAI's ChatGPT [cha], while capable of engaging in detailed conversations and handling diverse queries, lacks access to the most up-to-date information, which is crucial for analyzing rapidly changing data like inflation rates.

In this project, I aim to create a chatbot that will respond to inflation-related queries and provide visual insights that enhance user understanding and engagement with the data. This chatbot should transform how inflation data is accessed and understood, allowing users to receive real-time, relevant data responses without navigating through complex charts.

The core of this project involves implementing a multi-agent system, where each agent is specialized in specific tasks such as identifying user intent, generating and updating visualization code, handling errors, and answering questions in natural language. The primary challenge lies in the orchestration of these agents to ensure seamless interaction and functionality. Another significant challenge is creating clear system messages for each agent so they understand the tasks they need to accomplish. Additionally, given the dataset's complexity, characterized by a vast amount of information and a complicated hierarchical structure of consumer goods and services, it is crucial to communicate this complexity to the agents so they can perform their tasks efficiently. These challenges are managed through prompt engineering techniques, vital for fine-tuning LLMs to handle such complex tasks. The project also includes an evaluation phase to assess the chatbot's effectiveness and ease of use. During this phase, users interact with the chatbot to solve a specific task by asking inflation-related questions and then provide feedback on the system's functionality and overall experience. The same task is completed using a dashboard to see which tool is more intuitive. The goal is to determine whether the chatbot meets users' needs in real-world scenarios, makes accessing inflation data easier, and gathers insights for future improvements.

2.1 Research Questions

Given these challenges, the project aims to answer the following research questions:

- **RQ1** When is the use of a chatbot more advantageous over the use of a dashboard?

- **RQ2** How accurately can the developed chatbot respond to user queries related to inflation, and under what circumstances does it struggle to provide answers?
- **RQ3** Is it feasible to develop a chatbot that can create visualizations and respond to users' queries in natural language regarding inflation data?

2.2 Contributions

This thesis contributes to the field by developing a chatbot based on LLMs to effectively manage a complex inflation dataset. The project involves fine-tuning pre-trained models to generate and update visualizations from natural language inputs and provide coherent responses to inflation-related queries. Additionally, this research assesses user satisfaction with the system's responses and the process of retrieving inflation-related information, aiming to improve the chatbots' ability to communicate and interact effectively in economic data analysis.

3 Related Work

In this section, I will explore related works focusing on natural language interfaces for data visualization and studies on user experience and usability in chatbots.

3.1 Natural Language Interfaces for Data Visualization

Natural Language Interfaces (NLIs) for data visualization have significantly evolved. Early NLIs for data visualization used rule-based methods to translate user queries into visual data representations [GDA$^+$15, SLJL70, SBT$^+$16, HSTD18]. Eviza [SBT$^+$16] and Evizeon [HSTD18] were particularly notable for allowing users to interact with and modify visualizations, thus enhancing user engagement with data. Fast et al. [FCM$^+$18] extended these capabilities by incorporating the ability to solve other data science related tasks such as correlation analysis and predictive modeling. Despite their abilities, however, these systems have some constraints on the natural language inputs they can process. They often struggle to comprehend complex queries and have limited flexibility in adapting to varied user inputs [VMLZ21].

The introduction of advanced language models like GPT-3.5 and GPT-4 significantly improved the flexibility of NLIs. Multiple tools have utilized these models to generate Python code to visualize data based on natural language requests [MS23, Dib23, HKS23, NM23]. Maddigan et al. [MS23] developed a tool called Chat2Viz that enables users to input a desired dataset together with their analysis intentions. The system summarizes this information in the prompt and, as a result, renders the desired chart. However, to the best of my knowledge, this tool hasn't been tested on more complex datasets and is limited by its inability to engage in dialog interactions about data. VIDS [HKS23] and LIDA [Dib23] incorporated modular architectures to enhance their functionality. VIDS [HKS23] employed multiple agents, each tasked with specific roles, targeting a more technically skilled audience such as data scientists. It serves as a personal data science assistant, helping users engage more deeply with data by creating data science workflows and generating visualizations. LIDA [Dib23], another comprehensive tool, integrates modules that combine summarization, goal exploration, visualization generation, and infographic production, which together strive to create an automatic generation of grammar-agnostic visualizations and infographics using LLMs.

Despite these advancements, systems are still needed to create visualizations and answer domain-specific questions in natural language. Additionally, only a few systems offer the functionality to modify existing visualizations based on subsequent user input, highlighting a significant area for development.

3.2 User Experience and Usability in Chatbots

User interaction with chatbots has been the focus of various studies. For example, Jordan et al. [JKM23], and Hoque et al. [HMKH24] explored hypothetical interaction scenarios in elections and journalism, respectively. Jordan et al. [JKM23] mainly studied what kind of responses young adults would expect from a chatbot to give in return based on election panel data. These studies help us understand user expectations and usability challenges of conversational interfaces, but they do not involve creating and testing such tools. However, the insights gained are crucial for developing intuitive user-centered interfaces that effectively meet user needs.

4 Chatbot Structure

The central focus of this project is to develop a chatbot capable of performing multiple tasks, including generating visualizations, answering theoretical queries, and responding to data-related questions. The project draws inspiration from LIDA [Dib23] and VIDS [HKS23], which utilize multiple agents, each responsible for a specific task, to achieve these objectives. In this chapter, I will outline the created multi-agent system, detailing how these agents collaborate and describe the underlying dataset used to fine-tune the language models (LLMs). Additionally, I will discuss the user interface and how users can interact seamlessly with the chatbot.

4.1 Dataset

The dataset used to fine-tune the agents is the "Harmonized Index of Consumer Prices" (HICP), provided by the Institute for Advanced Studies. The description of this dataset is sent to all agents, except for the Simplify Query Agent (4.2.6) and Error Agent (4.2.7), as part of the system message. The IHS put this dataset together by combining data from Eurostat [eur] and adding some extra calculations. Although the original dataset used by the IHS for their Price Monitor dashboard is more extensive and more detailed, a smaller version was chosen for this project.

The final dataset is still complex, covering 37 EU countries, Turkey, and seven other regions and monetary unions. It includes data from January 1, 1996, to January 1, 2024. Part of the complexity comes from the fact that it includes 145 categories of consumer goods and services organized hierarchically (see Fig. 4.1). This results in a total of 1,665,964 rows. The data includes price indices, rates of change (monthly, annual, and 12-month moving averages), and weights that show the importance of each item/category and its contribution to the monthly or annual rate of change.

4.2 Multi-agent system

To achieve the desired chatbot functionality, seven agents are created. Two of these agents—the Visualization Agent (4.2.2) and the Query Agent (4.2.5)—were designed to handle more complex tasks and required greater capabilities. They are based on the Claude 3.5-Sonnet model developed by Anthropic [ant24]. This model was chosen over GPT-4 because it offers better performance in code generation while being more cost-effective [ant24]. The other five agents, responsible for more straightforward tasks, are built on OpenAI's GPT-3.5-Turbo model [gpt], providing efficient solutions at a lower cost. It should be noted, though, that at first, Query Agent was based on the

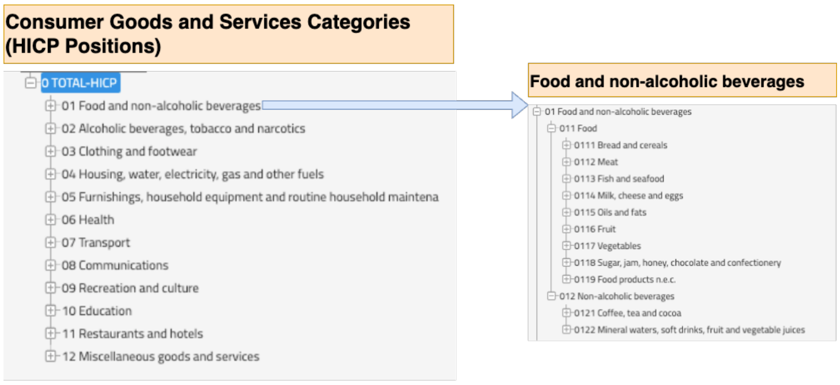


Figure 4.1: Consumer Goods and Services Classification: *The left side of the figure displays all 12 HICP categories. The right side provides a detailed breakdown of the "Food and non-alcoholic beverages" category, showing its breakdown into two subsequent sublevels: Food sublevel and Non-alcoholic beverages sublevel. Each of these sublevels is further divided into more detailed classifications. This hierarchical structure is similar to the other 11 categories.*

GPT-3.5-Turbo model as well, but after some interviews with users, it was noted to underperform, and it was decided to upgrade it to the Claude 3.5-Sonnet model (see 6.2.2 for more details). I will describe each of these agents in detail in the following sections.

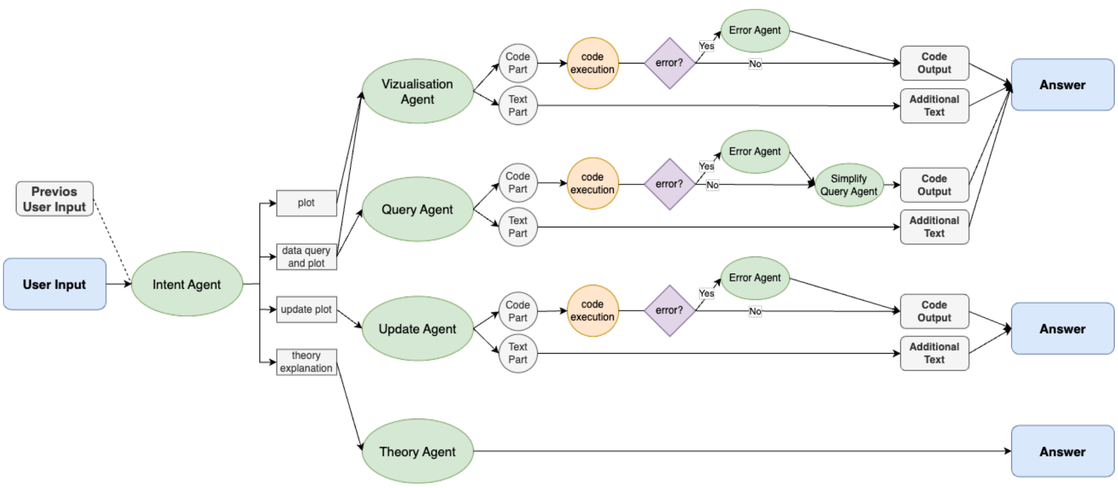


Figure 4.2: Chatbot Architecture

4.2.1 Intent Agent

Since the chatbot is designed to handle various tasks, the user's query must first be filtered to identify their intent. Some queries may require a theoretical explanation, while others might benefit from a visualization. Users might also want to modify an existing plot, such as changing its colors or adding additional information, or having a textual explanation alongside a plot.

The Intent Agent was created to manage this task of filtering and understanding the user's intent. It analyzes both the current query and, if applicable, the previous query to determine the expected response. The agent outputs one of four possible intents: `plot`, `update plot`, `theory explanation`, or `data query and plot`.

For example, a query like *"Explain what inflation is."* would result in a `theory explanation` intent, while *"What was the inflation trend in Austria in 2022?"* would return a `data query and plot` intent. A request such as *"Can you show a bar plot instead of a line plot?"* would trigger the `update plot` intent.

4.2.2 Visualization Agent

The Visualization Agent generates Python code to create plots that satisfy the user's query when the Intent Agent returns a `plot` or `data query and plot` intent. It receives the user's request and, using predefined templates, generates the necessary code to produce the desired visualization.

Inspired by the approach used by Dibia [Dib23], who applied LLMs in a fill-in-the-middle mode developed by Bavarian et al. [BJT$^+$22], the agent works by completing gaps in a pre-existing code template. This template already includes essential components such as library imports, data loading procedures, and additional guiding code to ensure the correct steps are followed (see Fig. 4.3a). Once the code is generated, it is executed, and if any errors occur, they are handled by the Error Agent (see 4.2.7).

In addition to generating the code, the agent must provide a detailed explanation of the selected countries, variables, dates, and categories, along with any calculations performed. An additional text template with all the requirements is also predefined and put into a system message (see Fig. 4.3c). The resulting information is stored in a separate text section of the agent's output. It will be presented to users to help them verify the data and decisions made by the chatbot (refer to 4.3 for more details).

4.2.3 Update Agent

The Update Agent is triggered when the Intent Agent (4.2.1) recognizes that a user wants to modify an existing plot. It takes the previously generated code and the two most recent user requests as input. Its task is to adjust the code according to the user's requests, ensuring the desired updates are applied. As output, the agent provides the updated code and a description of the changes made.

4.2.4 Theory Agent

The Theory Agent is activated when the Intent Agent (4.2.1) identifies a `theory explanation` intent. It is responsible for answering theoretical questions or those related to the dataset's structure. If the question is about inflation theory, the agent first consults the provided

````
```python
#import libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.dates import DateFormatter
import os
#Load the data
main_dir = os.getcwd()
data_dir = os.path.join(main_dir, 'data')
data = pd.read_csv(
        os.path.join(data_dir, 'hvpi_all.csv'),
        parse_dates=['datum'],
        dtype={{'code': str}}
        )
#Filter data for needed country(ies)
country_data = ...
#Filter data for needed category(ies)
selected_categories = ...
country_data = country_data[
        country_data['code'].isin(
        selected_categories)]
categories_names = country_data['name'].unique().tolist()
#Filter data for needed year(s)
...
#Create a plot
plt.figure(figsize = (10,6))
...
plt.grid(True, linestyle='--', alpha=0.7, axis='y')
sns.despine(left = True)
...
fig = plt.gcf()
```
````

(a) Code Template for Visualisation Agent

````
```python
#import libraries
import pandas as pd
import numpy as np

import os
#Load the data
main_dir = os.getcwd()
data_dir = os.path.join(main_dir, 'data')
data = pd.read_csv(
        os.path.join(data_dir, 'hvpi_all.csv'),
        parse_dates=['datum'],
        dtype={{'code': str}})

#Query the data to answer the question
#Filter data for needed category(ies)
selected_categories = ... (e.g ["0"])
data = data[data['code'].isin(selected_categories)]
...

answer = ...

#Provide a reply in natural language
#adding the identified answer variable
reply = '...'
```
````

(b) Code Template for Query Agent

````
```text
- **Countries Selected:**
provide all the country names you have chosen
- **Variables Used:**
provide all variables you used for plotting as they are written in the dataset
- **Dates Used:**
provide the exact year(s) and month(s) you have selected
- **Categories Used:**
provide the EXACT categories names you have selected depending on the digit codes you've filtered in the code
(e.g if you selected category with code 01, say Food and non-alcoholic beverages)
- **Calculations:**
in case you made any calculations inside of a plotting function (like average, sum) mention them, if not put No calculations performed
```
````

(c) Text Template

Figure 4.3: Templates used in the prompts for Visualization and Query Agents

glossary, sourced from the IHS Price Monitor web page[1]. If the answer cannot be found there, it relies on its own knowledge to respond. For questions about the dataset's structure, such as *"What dataset are you using and how is it organized?"*, the agent uses the provided dataset summary to generate a detailed answer.

4.2.5 Query Agent

The Query Agent functions similarly to the Visualization Agent (see 4.2.2), with the main difference being that it generates code to produce a natural language response to accompany the plot generated by the Visualization Agent. This agent is triggered when the Intent Agent (4.2.1) determines that providing a textual explanation in addition to the plot would enhance the user's understanding and returns a `data query and plot` intent.

Like the Visualization Agent, the Query Agent uses a predefined code template that is provided, filling in the necessary gaps to filter the data and perform any calculations required to address the user's query (see Fig. 4.3b). The output is a description of the data used (Fig. 4.3c) and a code that, when executed, generates a natural language

[1] https://inflation-en.ihs.ac.at/

answer. If the code cannot be executed, it is passed to the Error Agent for troubleshooting (see 4.2.7).

Since the initial natural language responses may sound overly robotic, the Simplify Query Agent (4.2.6) is employed to make the reply more user-friendly.

4.2.6 Simplify Query Agent

As referenced in 4.2.5, the Simplify Query Agent works in tandem with the Query Agent. Its primary task is to enhance the user-friendliness and readability of the natural language responses generated by the Query Agent, ensuring they are clearly structured and easy to understand.

4.2.7 Error Agent

The Error Agent is activated whenever an error occurs after executing the code generated by the Visualization Agent (4.2.2), Update Agent (4.2.3), or Query Agent (4.2.5). It receives both the problematic code and the corresponding error message as input, attempting to resolve the issue. The Error Agent makes up to two attempts to fix the error; if unsuccessful, the error is returned to the user.

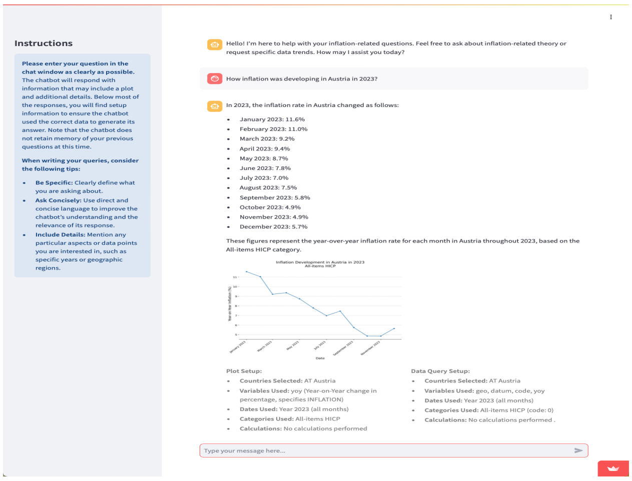
4.2.8 Agent Interaction and Workflow

Figure 4.2 illustrates the overall architecture of how the agents interact to process user queries. When the user asks a question, the Intent Agent (4.2.1) first classifies it based on one of four intents. The appropriate agent(s) are activated depending on the identified intent.

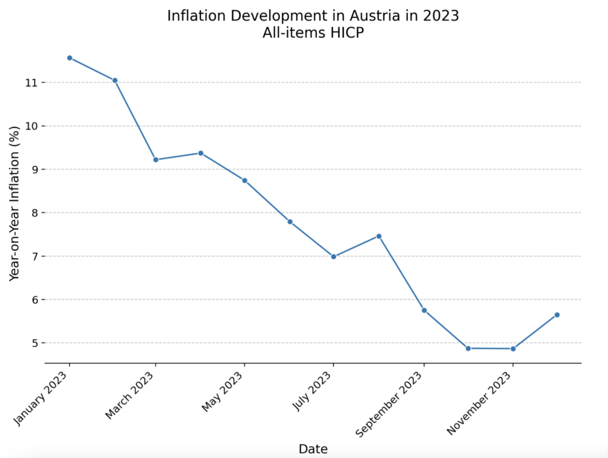
For example, if the query requires visualization, modification, or a data query, the Visualization (4.2.2), Update (4.2.3), or Query Agents (4.2.5) will handle the task respectively. Their output is then sent for execution. If an error occurs during execution, the Error Agent (4.2.7) steps in and attempts to resolve the issue up to two times. Once the Query Agent (4.2.5) generates a response, the Simplify Query Agent (4.2.6) refines the natural language output, making it more user-friendly. Finally, the complete response, whether it involves text, a plot, or both, is sent back to the user.

4.3 User Interface

The user interface was developed using Streamlit, a Python framework commonly used for creating interactive web applications, particularly in Data Science projects [stra]. Streamlit allows Python scripts to be easily converted into web apps. During the evaluation phase, the chatbot was deployed on Streamlit Community Cloud [strb], enabling users to access the chatbot through a web link.



(a) Frontpage



(b) Enlarged Plot

Figure 4.4: Streamlit Chatbot Interface

Figure 4.4a shows the layout of the chatbot interface. On the left, a set of instructions offers guidelines on how to use the chatbot, while at the top, the bot displays a welcome message. At the bottom of the interface, users can input their queries in the provided field. The example in the figure demonstrates a typical output from the chatbot, which includes a natural language response and a plot. Below the plot are sections labeled "Plot Setup" and "Data Query Setup". These sections specify the data sources and processes used by the Visualization Agent to create the plot and by the Query Agent to generate the accompanying text (see 4.2.2 and 4.2.5 for details). Users can enlarge the plot for closer inspection (see Fig. 4.4b).

5 Evaluation Design

After developing the chatbot, collecting feedback to evaluate its performance and understand user preferences was crucial. The main objectives of this feedback were to determine whether users would prefer the chatbot or the IHS Price Monitor for solving tasks, evaluate their satisfaction with the chatbot, assess its overall usefulness, and measure its effectiveness in addressing their questions. A user study was conducted to gather this information.

5.1 Participants

To evaluate the chatbot, 10 participants were recruited, labeled P1 through P10. This included five individuals from the Institute for Advanced Studies (P1-P5), who were familiar with the dashboard, and five (P6-P10), found among personal contacts, who had never used it before (see Table 5.1).

Five participants were between 18-24 years of age, four between 25-34 years of age, and one between 35-44. The gender distribution was even, with five identifying themselves as women and five as men. Educationally, nine hold university degrees, and one has a baccalaureate.

Regarding occupation status, four participants were employed and students, one was employed and self-employed, three were solely students, and two were exclusively employed. The participants' areas of expertise included six in Business and Economics, two in the Sciences (Biology, Chemistry, Mathematics, Physics), one in Computer Science/Computer Engineering/IT, and one with expertise in Computer Science/Computer Engineering/IT and Language and Linguistics.

Regarding their familiarity with inflation, four participants were very knowledgeable and able to teach the subject at a university level, three were moderately familiar, and three had basic familiarity (Fig. 5.1). In terms of prior chatbot usage, one participant had no previous experience, while nine had used chatbots of varying kinds: five had only interacted with ChatGPT, one had used both ChatGPT and a variety of LLM/GPT-based custom chatbots, and another had experienced both ChatGPT and JenniAI (Fig. 5.1).

5.2 Evaluation Metrics

Two metrics were utilized to evaluate the chatbot: the System Usability Scale (SUS) [Bro96] and open-ended questions. The SUS was slightly adapted for this project by substituting the word "system" with "chatbot" in its questions, following the approach used by Coperich et al. in their study [CCN]. Open-ended questions were designed to gain

Table 5.1: Participant Demographics.

Note: Participants marked with (*) used the the chatbot before the Query Agent was upgraded. Others used it after the upgrade to the advanced Claude 3.5-Sonnet model.

	Participant ID	**Gender**	**Age Group**	**Highest Education Level**	**Occupation**	**Area of Expertise**
Experts	P1*	Male	35-44	University degree	Employed	Business and Economics
	P2*	Male	25-34	University degree	Employed, Self-employed	Business and Economics
	P3*	Male	18-24	Baccalaureate	Student, Employed	Business and Economics
	P4	Female	18-24	University degree	Student	Business and Economics
	P5	Female	25-34	University degree	Student, Employed	Business and Economics
Non-Experts	P6	Female	18-24	University degree	Student, Employed	Computer Science / Computer Engineering / IT
	P7	Female	18-24	University degree	Student	Sciences (Biology, Chemistry, Physics, Mathematics)
	P8*	Male	25-34	University degree	Student, Employed	Computer Science / Computer Engineering / IT
	P9	Female	18-24	University degree	Student	Business and Economics
	P10	Male	25-34	University degree	Employed	Sciences (Biology, Chemistry, Physics, Mathematics)

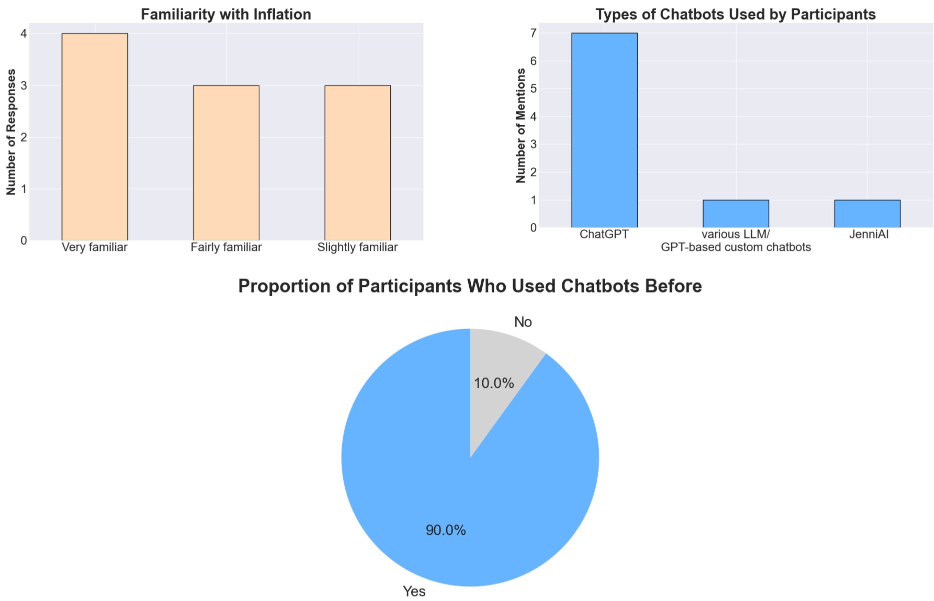


Figure 5.1: Overview of Participants' Inflation Knowledge Level and Familiarity with Chatbots

deeper insights into the participants' experiences and feedback concerning the chatbot. These questions included:

1. Describe your experience in solving the task. Which tool (chatbot or dashboard) helped you more effectively, and why?
2. What issues or challenges did you encounter while using the chatbot?
3. How useful were the visualizations and the natural language replies in helping you find answers to your questions?
4. How helpful was having query and plot setups listed under the visualizations?
5. What improvements would you suggest to enhance the chatbot's functionality?
6. How much did you trust the answers, and why?

5.3 Interview Procedure and Tasks

Each interview was conducted separately via Zoom [zoo] and was video and audio recorded, with written consent obtained from the participants. Additionally, each interview was transcribed using *Otter.ai* [ott] to facilitate the analysis. Before the interviews, participants were asked to complete a demographic questionnaire (more details can be found in section 5.1) and sign a consent form.

At the start of each interview, the procedure was thoroughly explained. Participants were then asked to perform a task involving the analysis of inflation trends in European countries of their interest. They were instructed to identify contributing factors to inflation in these countries and discover three new insights. Participants were required first to solve the task using the chatbot and subsequently perform the same task using the Price Monitor dashboard, allowing for a direct comparison of the usability of each tool. During these interactions, participants were encouraged to verbalize all their thoughts and impressions about the responses they received, their satisfaction with them, and whether the outputs met their expectations.

After using both tools, participants were asked to respond to the open-ended questions outlined in 5.2. Afterward, they had to complete a System Usability Scale questionnaire on Google Forms. The parts of the interview where the procedure was explained and the SUS questionnaire was completed were not recorded or transcribed.

On average, interactions with the chatbot and the dashboard, including the time spent on open-ended questions, lasted 31 minutes, of which 15 minutes were spent interacting with the chatbot. The shortest session was 13 minutes, and the longest was 46 minutes.

6 Findings

In the following section, I will present the findings, which include results from the System Usability Scale, participant feedback on the chatbot, and observations of agents' behavior from a technical perspective. This information was collected through interviews and analyzing logs generated during participant interactions with the chatbot.

6.1 SUS

The average System Usability Scale score among the 10 participants was 86.75, with a standard deviation 8.14. Scores ranged from a high of 100 to a low of 70. Specific responses to questions 1, 6, and 10 stood out because they were quite different from the rest. For example, in question 1, one participant disagreed with the statement that they would like to use this chatbot frequently, pointing to their lack of interest in inflation as the reason. In contrast, in question 10, one person strongly agreed that they would need to learn many things before using this chatbot (see Fig. 6.1 for details). It remains unclear what influenced such a response: whether it was the chatbot itself or other factors, as the participant did not provide further clarifications. Regarding question 6, *"I thought there was too much inconsistency in this chatbot"*, the reason for one participant selecting "Strongly Agree" is that some aspects of the chatbot's functionality need improvements.

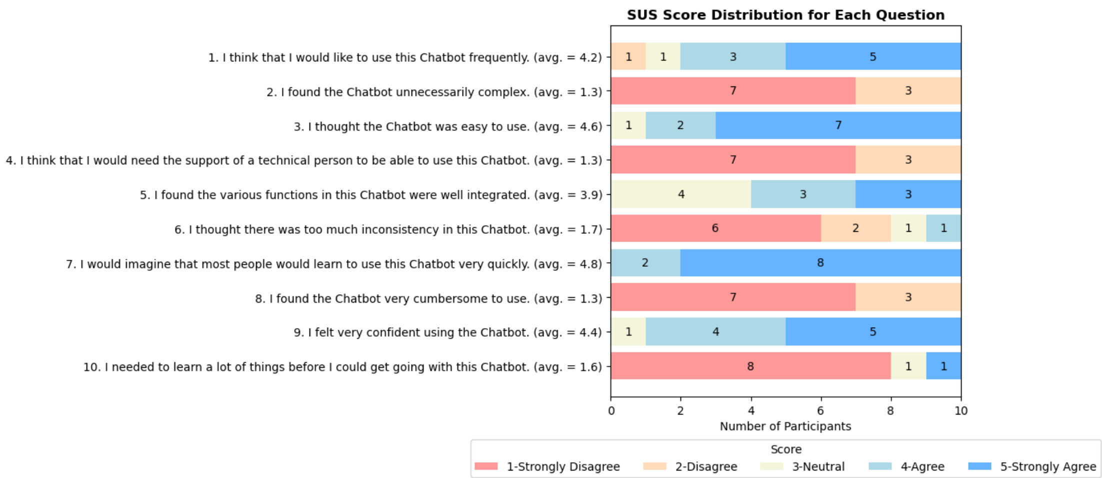


Figure 6.1: System Usability Scale Results

6.2 Participants' Feedback

This subsection is organized into several topics that were discussed during participant interviews. These topics are: dashboard and chatbot comparison, visualizations and natural language replies, plot and query setups, trust in the provided information, identified problems, and suggested improvements. Below, I provide gathered feedback for each of these topics.

6.2.1 Dashboard and Chatbot Comparison

In comparing the IHS Dashboard and the chatbot for accessing inflation data, participants offered various opinions, highlighting each tool's advantages and disadvantages. Most found the chatbot more user-friendly for obtaining general overviews and specific answers. For example, P2 and P9 noted the chatbot's straightforwardness and ease of use compared to the dashboard, which they mentioned requires some prior knowledge. P9 mentioned, *"I found the Chatbot easier to use because, yeah, I could just very easily direct it and then find the information I was looking for without having to figure out what the structure behind the dashboard is."*

However, some participants noted that while the chatbot was straightforward, it sometimes struggled with accuracy unless the questions were well formulated. For example, P8 commented, *"I had to reformulate the sentences, which is always the case with these chatbots anyways, but I actually got to have the data that I was interested in"* and P3 observed, *"The chatbot definitely has its perks, but it has some issues with the accuracy and how you work the question to really give you the thing that you want"*.

Additionally, participants highlighted the chatbot's ability to perform data aggregation and manipulation as a positive feature (P3 and P8). P8 further mentioned that it would be helpful to see how these calculations are performed. On the other hand, it was noted that the dashboard could be more advantageous for users seeking very detailed and specific data, provided they know how to navigate the tool effectively (P2, P3, P4).

The feedback suggests that the chatbot provides a more user-friendly and efficient experience for users unfamiliar with the IHS Price Monitor. The statement of P7 could summarize this, where they mentioned the chatbot to be *"a lot more"* helpful because the dashboard *"has a lot of information, but for me, it had an overwhelming amount of information"*. This underscores that while the dashboard is powerful for those adept at using it, the chatbot offers a more accessible option for most users.

6.2.2 Visualizations and Natural Language Replies

All participants found the combination of textual and visual information good and useful. Some were pleasantly surprised to see visualizations included in the responses (P4, P8, P9). However, P3 experienced difficulties when the textual and visual information did not align: *"It should show the same thing in general. So, if you have some result in the text, and then the plot shows you a different thing (even if both are correct), it's confusing. You might question the validity of the answer"*. This issue may have been caused by the Query Agent using a simpler GPT-3.5-Turbo model, which might have led to less accurate results. Consequently, I upgraded the Query Agent to the more advanced Claude 3.5-Sonnet model. Following this upgrade, no further issues were reported regarding this matter. Participants who used the chatbot before the upgrade are marked with an asterisk in Table 5.1.

6.2.3 Plot and Query Setups.

Another opinion, which was interesting to discover, concerned the plot and query setups provided under the plot as a part of the response. Views on this feature varied among participants. P1, P3, and P7 felt these setups were unnecessary if one only needed a quick answer. P4 commented, *"It's a bit much because almost half of the page you have just information on what it did. But on the other hand, if I would need to quickly know which information I asked the chatbot, yes, it is nice to have it at one glance, it's good it's there"*. There were suggestions to make these setups optional, with P2 and P3 proposing a toggle feature that allows users to access more detailed information on demand. This could satisfy those who prefer quick information and those interested in more detailed insights about the information used for the answer.

P1 and P2 mentioned that understanding these setups requires familiarity with the system and some prior knowledge of the variables. P4 suggested adding additional explanations of the variables to enhance understanding.

P4, P7, and P9 noted they didn't pay attention to the setups but appreciated knowing they were available. For example, P7 stated, *"Psychologically, it was really nice to know"* they were there. Conversely, participants like P6 and P10 were indifferent, noting that the setups did not disturb or significantly impact their chatbot use.

6.2.4 Trust in the Provided Information.

Regarding trust in the chatbot, feedback varied slightly among participants. P2, P4, (both of which had an economic background) and P6 (who has a Computer Science background) had complete trust in the answers. P6 additionally expressed full confidence in *"all of the numbers and visualizations"* when they found something reasonable. P5 and P8 also trusted the answers because they knew the source of the data. For example, P5 mentioned, *"I did trust the answers because I know what dataset it's accessing, so I do know that the dataset that it's accessing is trustworthy"*. Conversely, P9 felt hesitant due to not knowing where the chatbot sourced its general information. P3 suggested providing a link to the data source would be useful.

Additionally, P7 and P10 stated that they would double-check the provided information if they needed to report it or required *"very trustworthy"* information.

6.2.5 Identified Problems and Suggested Improvements.

In addition to the improvements identified in previous paragraphs, such as adding the source of the data to increase trust and enabling plot and query setups with a click, further enhancements were proposed. P6 suggested adding tables alongside the text to summarize all information nicely. P4 appreciated the idea of having interactive plots.

Other improvements were suggested after participants noted some issues with the chatbot. P1, P5, and P8 observed the chatbot's inability to recall information from earlier in the conversation, which complicated accessing necessary data. P5 mentioned having *"difficulties accessing information that it was given prior, or that it generated prior"*. P1 pointed out, *"every new question is a new conversation if you don't explicitly refer to the other one. This comes unnatural and unintuitive"*. From these challenges arose the suggestion to equip the chatbot with a memory feature.

Since P3 raised concerns about mismatches between the text responses and plots, they suggested adding a capability that ensures the content aligns with the user's query and that both the plot and the query are consistent before being presented to the user. This issue aligns with challenges faced by P7 and P10, who noted that one of their questions was misunderstood or incorrectly interpreted. P7 described feeling confused after receiving *"the graph with country comparison that provided me the different time period of inflation"*.

The final suggestion was to keep previously generated plots on the page rather than removing them with each new question, as noted by P6 and P8. P6 also mentioned that this issue could be addressed by providing a table with summarized information.

6.3 Chatbot Agents' Behaviour

This subsection looks at how the chatbot responded to user queries, focusing on which agents were used most, which ones worked well, and where some improvements might be needed.

As mentioned in Section 4.2.1, every time a user asks a question, the Intent Agent must identify its intent so that the right agent can take over. This agent's accuracy is crucial because it determines if the user gets the desired response. Figure 6.2 shows how often each intent was identified. On average, each participant asked 8 questions. The chart shows that most questions were identified with `data query and plot` intent, which means the Query Agent and Visualization Agent have participated the most in delivering the answer. However, The Visualization Agent was never triggered on its own because no `plot` intents were identified. Once, the Intent Agent failed to recognize an intent, leading to responses like *"The question can't be answered"*, shown on the plot as "Other".

The `update plot` intent was relatively rare among participants, but for some, it appeared for nearly half of the inquiries. While this intent was correctly identified for such requests like *"stack the two lines to stacked bar chart"*, or *"Can you now zoom in on just Austria and France?"*, it was wrongly identified for a query *"what contributes to Belgium's low inflation rate"*. This question was asked right after a question about Belgium's inflation rate, which had a `theory explanation` intent, making an `update plot` intent inappropriate as there was no chart to modify. Furthermore, even when `update plot` was correctly identified, the Update Agent sometimes struggled to modify previous charts as needed significantly, leading to user confusion. For instance, a participant requested:

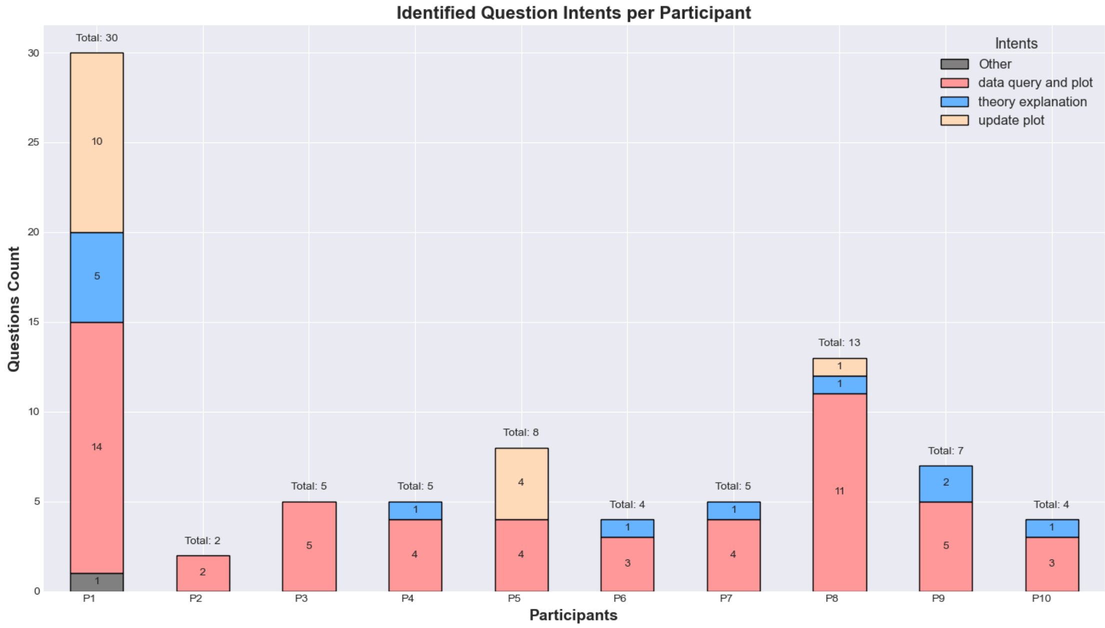
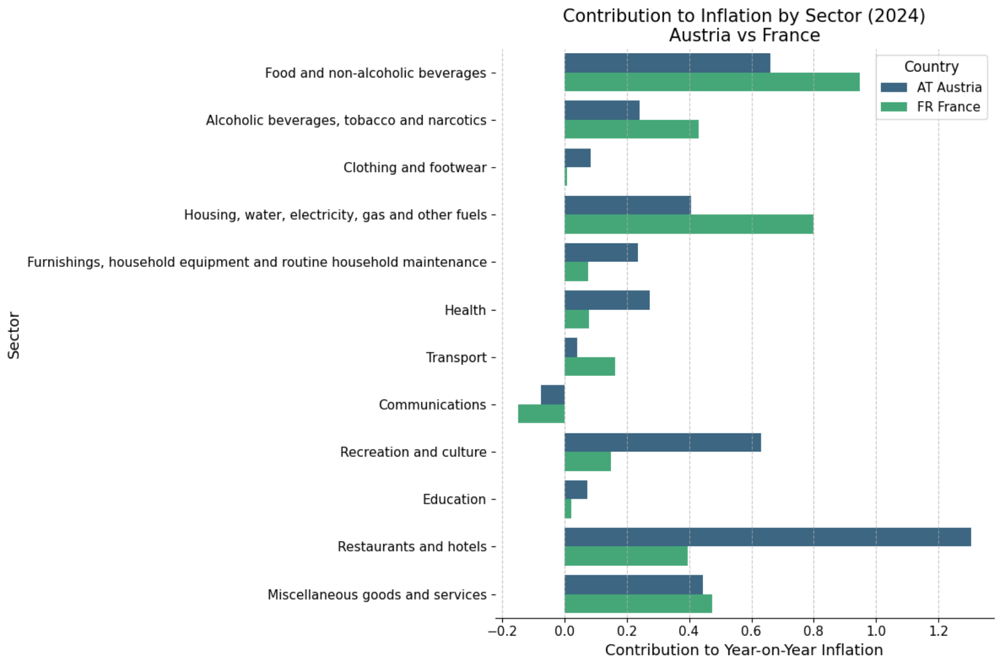


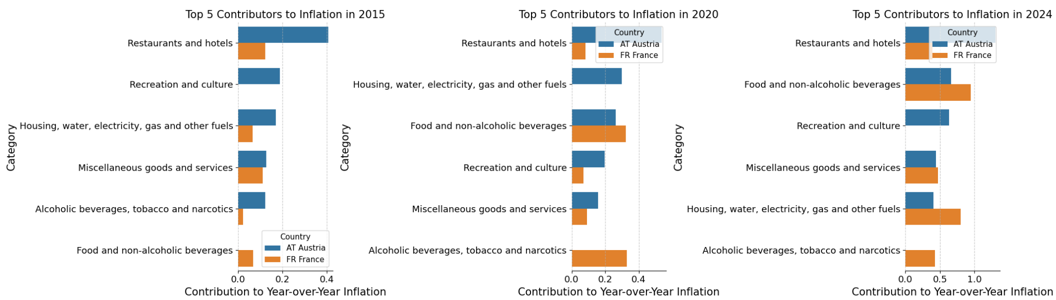
Figure 6.2: Question Intents per Participant

"Could you please show me the same graph you created before, but just for the years 2015, 2020, and 2024—plot three different graphs, one for each year". The bot, however, overlaid the plots for each year, which contradicted the participant's request (see Fig. 6.3a). Despite some attempts to rephrase the query, the bot repeatedly produced the incorrect chart. It was not until the query was reformulated as *"I want to see for Austria and France what were the drivers behind inflation in the years 2015, 2020 and 2024. I would be interested in seeing what industry/sector contributed to inflation most. Plot 3 graphs side-by-side, one for each year."*, that the Intent Agent identified as `data query and plot` intent, which resulted in a correct output, as seen in Fig. 6.3b.

The `theory explanation` intent was also rarely triggered, likely because the task that participants had to solve did not require many theoretical questions. Queries classified as theoretical included requests like *"Why can the inflation level be different in different countries?"*, *"Hello! Could you please tell me about inflation? I am new to the topic."*, and *"What does negative inflation mean in terms of transport?"*. However, for questions like *"Why is Belgium's inflation rate so low?"*, or *"What contributes to Belgium's low inflation rate."*, one participant expected a plot showing specific contributors to the low inflation instead of a theoretical, abstract explanation. This would also be a correct understanding from the authors' point of view and yet these were incorrectly categorized by the chatbot as the `theory explanation` intent.



(a) Wrong output from the Update Agent



(b) Correct output from the Visualization Agent

Figure 6.3: Examples of the Agents' Outputs

7 Discussion

Based on the findings, participants generally found the chatbot user-friendly, especially for straightforward tasks and getting a general overview. It was perceived as more intuitive than the IHS Price Monitor, particularly for users unfamiliar with the dashboard's structure. While the chatbot showed promise, some limitations emerged that could be addressed in future work.

7.1 Limitations

The multi-agent system of the chatbot performed well overall, but it was observed during the evaluation phase that less concrete questions tended to receive less accurate responses. This was mainly because the chatbot couldn't reference previous queries in the dialogue. The only exception when it could access previous queries was when the identified intent was to update the plot—this triggered the Update Agent to modify a previously generated chart. However, there were instances where the Intent Agent struggled to correctly recognize when the Update Agent was needed.

Additionally, some participants noticed occasional mismatches between the plots and the natural language responses. This happened because the two outputs were generated by separate agents, which could result in inconsistencies. Despite these challenges, the multi-agent system and the chatbot showed strong potential, and with further adjustments, it could deliver even better results.

The evaluation involved a relatively small group of 10 participants, each asking, on average, eight questions. Due to the design of the task and time constraints, it was noticed that the evaluation was limited to specific types of queries and interaction styles. This means the findings might not fully represent how a broader audience would interact with the chatbot or the challenges they might face.

7.2 Future Work

To overcome these limitations and improve the chatbot's functionality, several areas for future development have been identified:

Adding a memory feature. One way to address the misinterpretation of user queries is by introducing a memory feature. This would help the chatbot understand user intents more clearly and reduce the need for participants to clarify or restate their requests repeatedly. A memory feature could also allow the Visualization Agent to update previously generated plots directly, as it would retain access to past data and code. This could potentially eliminate the need for the Update Agent. However, implementing memory would increase costs since it would require more tokens for storing previous queries.

Modifying intents and the multi-agent system. After adding a memory feature, it might be worth exploring how removing the Update Agent and relying solely on the Visualization Agent for plot modifications would impact the system's performance. This could streamline the process and reduce complexity. Additionally, since the Intent Agent never identified the `plot` intent during evaluation, and participants appreciated the combination of visualizations and natural language responses, future iterations could simplify intents to two categories: `theory` intent and `data query and plot` intent.

It might be worth combining these two agents to address mismatched outputs between the Visualization Agent and the Query Agent in future studies. On the one hand, this could lead to more consistent outputs with no contextual differences between the plots and natural language replies while also reducing costs. On the other hand, it's possible that the quality of both plots and natural language responses could decrease. However, with the development of more advanced models, this approach could yield promising results.

Long-Term Interaction Studies. To gain deeper insights, future research could explore how people interact with the chatbot over an extended period. Observing daily interactions could uncover more diverse use cases and provide a clearer picture of what further enhancements the chatbot might need.

7.3 Implications

This project is a solid starting point, laying the groundwork for creating a fully functional tool to access inflation data. In its current state, the chatbot already demonstrates potential as a practical assistant, capable of answering inflation-related questions in natural language, providing visualizations, and delivering meaningful insights to users. The findings from the evaluation sessions offer valuable guidance for further improvements, helping to turn the chatbot into an even more robust and reliable resource.

8 Conclusion

This project focused on developing an LLM-based chatbot capable of answering inflation-related questions using visualizations and natural language responses. A multi-agent system was implemented to achieve this, with each agent assigned to a specific task. The evaluation showed that participants found the chatbot more user-friendly than the IHS Price Monitor. Many noted that the dashboard requires familiarity to navigate efficiently, while the chatbot offered a more intuitive experience. Currently, the chatbot can already serve as a practical assistant for inflation-related questions, and with some further improvements, it has the potential to complement the IHS Dashboard by simplifying and streamlining access to inflation data.

Bibliography

[ant24] Introducing the next generation of claude, 2024.

[BJT+22] Mohammad Bavarian, Heewoo Jun, Nikolas Tezak, John Schulman, Christine McLeavey, Jerry Tworek, and Mark Chen. Efficient training of language models to fill in the middle, 2022.

[Bro96] John Brooke. *SUS – a quick and dirty usability scale*, pages 189–194. 01 1996.

[CCN] Katherine M. Coperich, Elizabeth A. Cudney, and Harriet Black Nembhard. Continuous improvement study of chatbot technologies using a human factors methodology.

[cha] Chatgpt.

[Dib23] Victor Dibia. Lida: A tool for automatic generation of grammar-agnostic visualizations and infographics using large language models, 2023.

[eur] Eurostat: Hicp - monthly data (index).

[FCM+18] Ethan Fast, Binbin Chen, Julia Mendelsohn, Jonathan Bassen, and Michael S. Bernstein. Iris: A conversational agent for complex tasks. CHI '18, page 1–12, New York, NY, USA, 2018. Association for Computing Machinery.

[GDA+15] Tong Gao, Mira Dontcheva, Eytan Adar, Zhicheng Liu, and Karrie G. Karahalios. Datatone: Managing ambiguity in natural language interfaces for data visualization. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, page 489–500, New York, NY, USA, 2015. Association for Computing Machinery.

[gpt] Gpt-3.5 turbo.

[HBK+21] Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, and Jacob Steinhardt. Measuring coding challenge competence with apps, 2021.

[HKS23] Md Mahadi Hassan, Alex Knipper, and Shubhra Kanti Karmaker Santu. Chatgpt as your personal data scientist, 2023.

[HMKH24] Md Naimul Hoque, Ayman Mahfuz, Mayukha Kindi, and Naeemul Hassan. Towards designing a question-answering chatbot for online news: Understanding questions and perspectives, 2024.

[HSTD18] Enamul Hoque, Vidya Setlur, Melanie Tory, and Isaac Dykeman. Applying pragmatics principles for interaction with visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):309–318, 2018.

[JKM23] Bernhard Jordan, Laura Koesten, and Torsten Möller. Chatting about data - interacting with voice interfaces to engage with election panel data. In *Proceedings of the 5th International Conference on Conversational User Interfaces*, CUI '23, New York, NY, USA, 2023. Association for Computing Machinery.

[MS23] Paula Maddigan and Teo Susnjak. Chat2vis: Generating data visualisations via natural language using chatgpt, codex and gpt-3 large language models. *IEEE Access*, PP:1–1, 01 2023.

[NM23] David Noever and Forrest McKee. Numeracy from literacy: Data science as an emergent skill from large language models, 2023.

[ott] Otter.ai.

[SBT+16] Vidya Setlur, Sarah E. Battersby, Melanie Tory, Rich Gossweiler, and Angel X. Chang. Eviza: A natural language interface for visual analysis. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, page 365–377, New York, NY, USA, 2016. Association for Computing Machinery.

[SLJL70] Yiwen Sun, Jason Leigh, Andrew Johnson, and Sangyoon Lee. Articulate: A semi-automated model for translating natural language queries into meaningful visualizations. pages 184–195, 01 1970.

[stra] Streamlit.

[strb] Streamlit community cloud.

[VMLZ21] Henrik Voigt, Monique Meuschke, Kai Lawonn, and Sina Zarrieß. Challenges in designing natural language interfaces for complex visual models. In Su Lin Blodgett, Michael Madaio, Brendan O'Connor, Hanna Wallach, and Qian Yang, editors, *Proceedings of the First Workshop on Bridging Human–Computer Interaction and Natural Language Processing*, pages 66–73, Online, April 2021. Association for Computational Linguistics.

[zoo] Zoom.

9 Acknowledgements

I acknowledge the use of ChatGPT as a tool to assist in improving the language, grammar, and readability of this thesis. The content and structure were entirely my own creation. After utilizing the tool, I carefully reviewed and edited the text to ensure its accuracy and alignment with my original work.