TU WIEN Informatics

Enhancing Dashboard Onboarding via LLMs

BACHELORARBEIT

zur Erlangung des akademischen Grades

Bachelor of Science

im Rahmen des Studiums

Software & Information Engineering

eingereicht von

Philipp Holzer
Matrikelnummer 12028208

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.-Prof. DI Dr.techn. Eduard Gröller
Mitwirkung: DI Dr. Vaishali Dhanoa, Aarhus University Denmark

Wien, 3. Jänner 2025

Philipp Holzer   Eduard Gröller

TU WIEN Informatics

Enhancing Dashboard Onboarding via LLMs

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Software & Information Engineering

by

Philipp Holzer
Registration Number 12028208

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.-Prof. DI Dr.techn. Eduard Gröller
Assistance: DI Dr. Vaishali Dhanoa, Aarhus University Denmark

Vienna, January 3, 2025

Philipp Holzer　　Eduard Gröller

Erklärung zur Verfassung der Arbeit

Philipp Holzer

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 3. Jänner 2025

Philipp Holzer

Danksagung

Hiermit möchte ich meinen herzlichsten Dank all denjenigen aussprechen, die mich während der Entstehung meiner Bachelorarbeit unterstützt und begleitet haben.

Mein besonderer Dank gilt zunächst meinem Betreuer Eduard Gröller für seine wertvolle Unterstützung, das geteilte Fachwissen und die konstruktiven Rückmeldungen, die maßgeblich zum Erfolg dieser Arbeit beigetragen haben. Ebenso danke ich Vaishali Dhanoa, die mich von der Zielsetzung bis hin zur Definition der Herausforderungen unterstützt und mich in allen Phasen der Entwicklung motiviert und begleitet hat.

Ein ebenso großes Dankeschön gilt meiner Familie, die mir während meines gesamten Studiums und insbesondere in den intensiven Phasen der Bachelorarbeit stets zur Seite stand. Ihre Unterstützung, ihr Verständnis und ihr Zuspruch haben mir viel Kraft gegeben und waren eine wertvolle Stütze.

Nicht zuletzt möchte ich auch meinen Studienkollegen danken, die diesen Weg mit mir gemeinsam gegangen sind. Durch den Austausch, die gegenseitige Motivation und das gemeinsame Meistern von Herausforderungen wurde das Studium zu einer wertvollen und unvergesslichen Erfahrung.

Allen genannten und nicht genannten Personen, die in dieser Zeit an meiner Seite waren und mich in vielfältiger Weise unterstützt haben, danke ich von Herzen.

Acknowledgements

I would like to express my heartfelt gratitude to all those who supported and accompanied me throughout the creation of my bachelor's thesis.

First and foremost, my special thanks go to my supervisor Eduard Gröller for his invaluable support, the knowledge he shared, and his constructive feedback, which significantly contributed to the success of this work. I would also like to thank Vaishali Dhanoa, who assisted me from the goal-setting phase through to defining challenges and continuously motivated and supported me through every stage of development.

A special thanks goes to my family, who stood by me throughout my entire studies and especially during the intensive phases of my thesis work. Their support, understanding, and encouragement gave me strength and were an invaluable pillar of support.

Last but not least, I would like to thank my fellow students, who walked this path alongside me. Through our exchanges, mutual motivation, and the shared experience of overcoming challenges, my studies became a valuable and unforgettable journey.

To all those named and unnamed who were by my side during this time and supported me in countless ways, I extend my heartfelt thanks.

Kurzfassung

Die Datenvisualisierung bietet starke Einblicke, die bessere Entscheidungen ermöglichen. Allerdings kann ihre Komplexität oft überwältigend wirken. Um ihr volles Potenzial freizusetzen, braucht es einen effektiven Onboarding-Prozess, der Nutzer*innen unabhängig von ihrer Erfahrungsebene eine klare Einführung in den Zweck und die Funktion von Visualisierungen bietet. Traditionelle Onboarding-Methoden, wie etwa statische Tutorials, können häufig die unterschiedlichen Bedürfnisse der Nutzer nicht ausreichend abdecken, was zu Verwirrung und Ineffizienz führt. Diese Arbeit stellt eine innovative Onboarding-Lösung vor, die auf großen Sprachmodellen (LLMs) basiert und darauf ausgelegt ist, personalisierte, kontextbewusste Unterstützung zu bieten, die sich dynamisch an die jeweilige Nutzer*in anpasst.

Unser Ansatz nutzt die Möglichkeiten von Prompt-Engineering, adaptiver Sequenzierung und konversationellen Interaktionen, um ein dynamisches und ansprechendes Onboarding-Erlebnis zu schaffen. Zu den zentralen Funktionen gehören maßgeschneiderte Prompts, die komplexe visuelle Elemente verständlicher machen, adaptive Sequenzierungen, die auf das Verhalten der Nutzer reagieren, sowie individuell angepasste Erklärungen, die sich am Wissensstand der Nutzer orientieren. Diese Funktionen wurden in einem Prototypsystem umgesetzt, das auf Llama 3.1 und ChatGPT 4o basiert, um eine reaktionsschnelle Unterstützung in Echtzeit zu bieten.

Durch die Umwandlung von Dashboards in zugänglichere Werkzeuge macht diese Arbeit die Datenvisualisierung für ein breiteres Publikum nutzbar und verbessert so die Art und Weise, wie Nutzer*innen mit komplexen Daten arbeiten und daraus Erkenntnisse gewinnen.

Abstract

Data visualization offers powerful insights that drive better decision-making, but its inherent complexity can often be challenging. To understand these visualizations, an effective onboarding process is essential, providing clear guidance on the purpose and functionality of visualizations for users of all experience levels. Traditional onboarding approaches, such as static tutorials, frequently fall short in addressing the unique needs of diverse users, leading to confusion and inefficiency. This thesis presents an innovative onboarding solution powered by Large Language Models (LLMs), designed to provide personalized, context-aware assistance that adapts dynamically to each user.

Our approach harnesses the capabilities of prompt engineering, adaptive sequencing, and conversational interactions to create a dynamic, engaging onboarding experience. Key features include custom prompts to clarify complex visual elements, adaptive sequencing that responds to user behavior, and tailored narratives that adjust to users' expertise levels. We implemented these features into a prototype system, powered by Llama 3.1 and ChatGPT 4o, to provide real-time, responsive assistance.

By transforming dashboards into more approachable tools, this work makes data visualization accessible to a wider audience, ultimately enhancing the way users interact with and extract insights from complex data.

Contents

CHAPTER 1

Introduction

Large Language Models (LLMs), such as GPT-3, have recently emerged as powerful tools for natural language processing, revolutionizing the way computers understand and generate text [LM22]. These models, consisting of billions of parameters, excel at creating coherent, contextually accurate responses to diverse inputs. The key to their success lies in leveraging vast amounts of training data—encompassing various topics, languages, and writing styles—to accurately interpret prompts and provide human-like interactions.

LLMs have found applications across many domains, such as answering questions, summarizing information, translating languages, generating creative content, and simplifying complex systems for end users [LM22]. Their versatility in handling both comprehension and communication tasks makes them effective at more than just keyword-based responses, enabling nuanced and context-aware engagement with users. Moreover, LLMs are capable of domain-specific adaptation, allowing them to deliver precise and relevant content by using prompts crafted for specific applications. One such application that stands to benefit significantly from this adaptability is the onboarding process for data visualization dashboards.

Dashboards are instrumental in transforming complex data into understandable visual insights, allowing users across industries like business, healthcare, and education to make informed decisions [DWH+22]. However, these dashboards can be complex, featuring numerous interactive visualizations, data filters, and advanced analytics. This complexity often presents challenges to users without a strong background in data analysis, limiting their ability to gain meaningful insights or understand what the visualizations convey [DWH+22].

Traditional onboarding methods for dashboards, such as static tutorials or pre-defined walkthroughs, tend to be rigid, presenting information in a "one-size-fits-all" format that often fails to address individual users' needs. Such methods lack adaptability, making onboarding overwhelming for some users, while insufficient for others who might require more specific guidance.

This is where LLMs can make a meaningful impact. By leveraging their contextual understanding, LLMs can provide personalized onboarding experiences that dynamically adjust to user interactions and queries [LM22]. Instead of a static script, an LLM can interact with users conversationally, responding to specific questions and providing the right level of information when it is most needed. This approach makes the onboarding process smoother, more engaging, and ultimately more effective, as it adapts to varying levels of user expertise [DHF+25].

In this thesis, we explore how LLMs can enhance the onboarding experience for dashboards by providing dynamic, responsive, and personalized user support [YN24]. Our approach utilizes several key strategies to leverage the power of LLMs in improving user comprehension and engagement:

- **Prompt Engineering for Onboarding Content**: We design prompts tailored to specific visualizations and contexts, enabling LLMs to generate relevant onboarding content that helps users understand dashboard features.
- **Adaptive Sequencing of Explanations**: By utilizing LLMs to determine the optimal order for presenting information, we ensure that onboarding follows a logical progression tailored to user interactions with the dashboard.
- **Dynamic Narratives Based on User Expertise**: The onboarding content is dynamically adapted to match different expertise levels, ensuring that users receive information that is appropriate to their skill level, from novice to expert [CLL+24].
- **Onboarding Process Model Integration**: Drawing on Dhanoa et al.'s process model, we integrate an onboarding loop alongside the dashboard usage loop, which ensures a structured, user-centric approach to onboarding that adapts to specific user needs [DWH+22, DHF+25].
- **Engaging Captions for Data Visualizations**: We employ LLMs to generate captions that go beyond simple data descriptions, offering insights and context that make visual elements more engaging and memorable, thus improving user interaction and understanding [LM22].

By combining the adaptability and contextual understanding of LLMs with effective onboarding strategies, this work aims to make dashboards more approachable for a broad audience, thereby enhancing how users interact with and extract value from data visualization tools. This thesis not only contributes to the field of data visualization but also demonstrates the potential of LLMs in delivering personalized, context-aware user experiences.

CHAPTER 2

Related Work

Our research builds upon various studies and systems that have explored authoring tools, adaptive onboarding methods, and user interaction techniques, all of which contribute to enhancing dashboards for diverse user groups through personalized onboarding. Previous work on LLMs, interactive dashboards, and adaptive visualizations forms the foundation of our approach, enabling a responsive onboarding process tailored to different user needs [YN24, DWH+22, DHF+25].

A key contribution to this domain is Yanez and Nobre's work, which demonstrated the use of GPT-4 to adapt data visualizations based on user characteristics such as cognitive abilities and preferences [YN24]. The authors simulated diverse user personas (as seen in Figure 2.1) to evaluate the impact of personalization, revealing how adapting visual data can improve comprehension for users with varied levels of expertise. This focus on user-centric adaptability aligns well with our goal of enhancing dashboard onboarding through personalized assistance, ensuring users receive context-specific guidance suited to their proficiency and preferences.

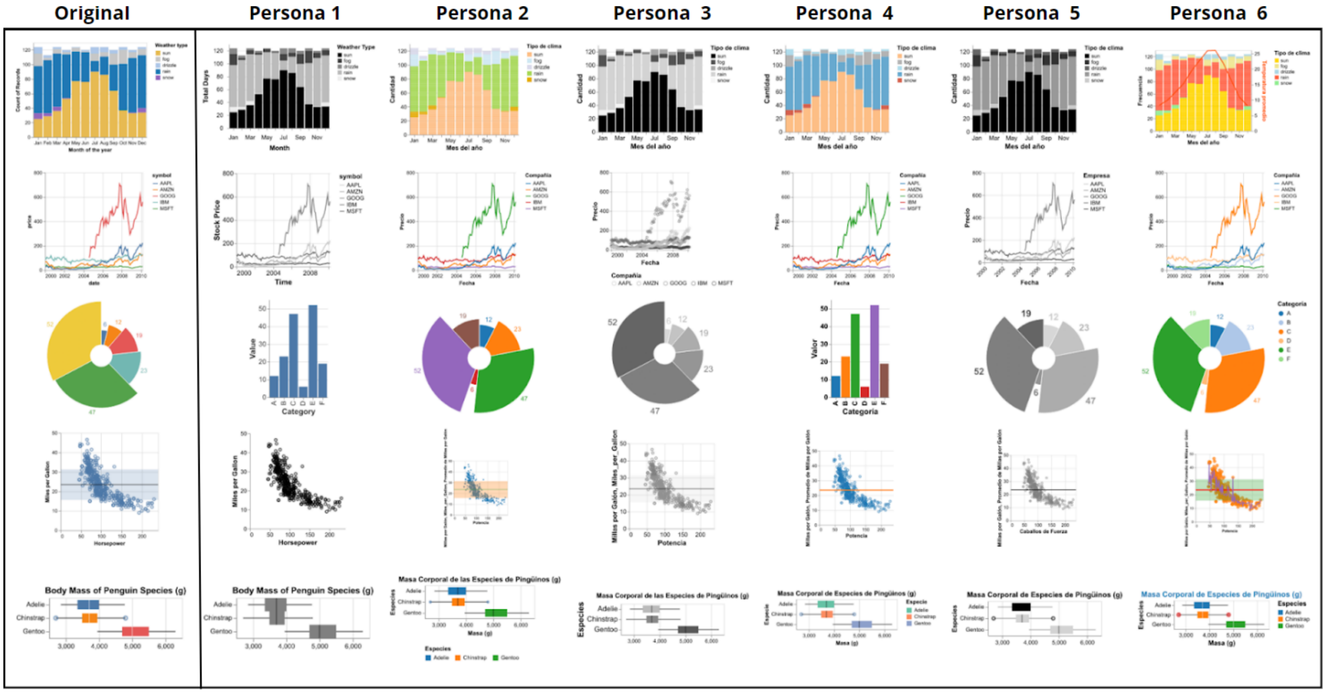

Figure 2.1: A comparative analysis of data visualizations designed for artificial user personas. The first column presents the original visualization set, while the following columns feature customized versions tailored to Personas 1 through 6. [YN24]

Dhanoa et al. introduced a dual-loop onboarding model consisting of a "dashboard usage loop" and an "onboarding loop", specifically designed to deliver personalized guidance (Figure 2.2). The onboarding loop presents adaptive onboarding artifacts like tutorials or interactive tours that adjust to user needs. This model provides the foundation for creating a structured, interactive onboarding experience. [DWH+22] Inspired by this work, our approach integrates LLMs to extend the onboarding loop concept into a real-time adaptive process. By leveraging LLMs' contextual understanding, we aim to provide a more interactive and dynamic onboarding experience compared to traditional, static tutorials.

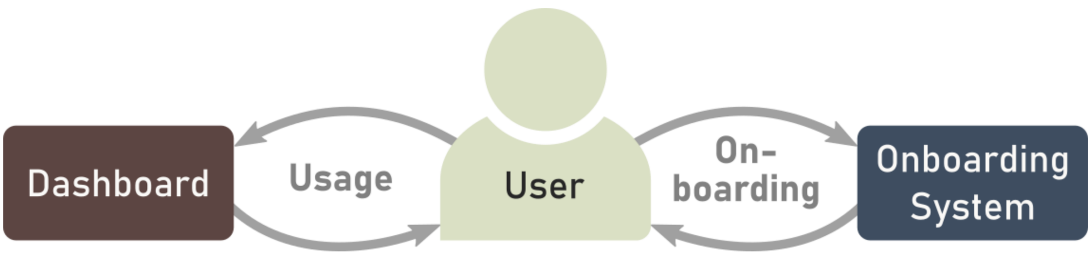

Figure 2.2: Introducing the dashboard onboarding loop in conjunction with the usage loop. [DWH+22]

Furthering this idea, the D-Tour prototype proposed by Dhanoa et al. (Figure 2.3) used semi-automated, narrative-driven guided tours to assist users in navigating dashboards at their own pace, offering more control over the learning process. [DHF+25] This approach reduces cognitive overload, especially for novice users, while still being valuable to experts who may need to navigate advanced features. This notion of narrative-driven exploration directly informs our approach, wherein LLMs create personalized and responsive onboarding sequences, making the learning experience adaptive to each user.

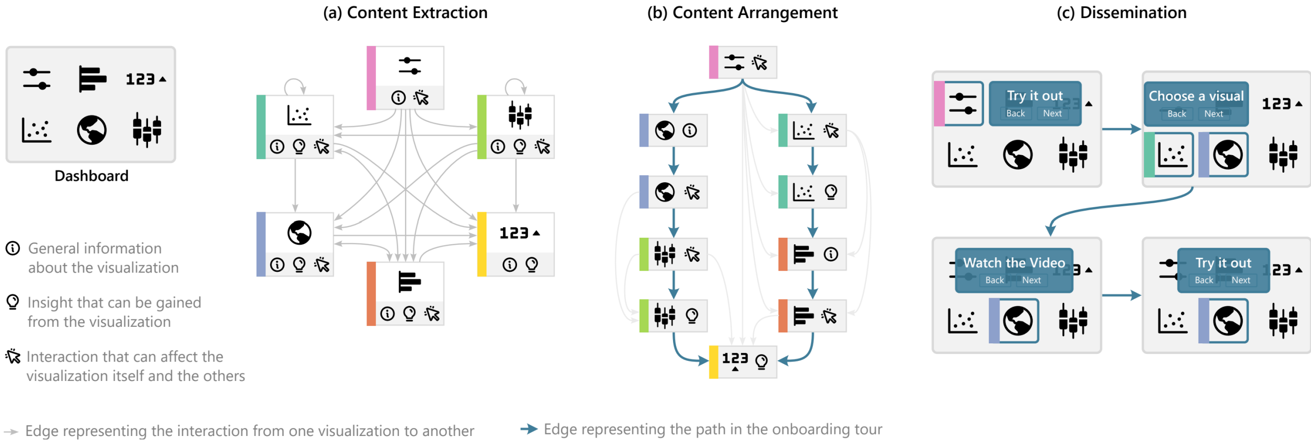

Figure 2.3: Creating and sharing interactive dashboard tours. This workflow outlines the process of designing semi-automated onboarding experiences while maintaining user agency. The content from a visualization dashboard is (a) extracted and transformed into a component graph, (b) organized into an interactive dashboard tour, and (c) distributed to end-users. [DHF+25]

LIDA (Figure 2.4), another tool explored in the literature, utilizes LLMs to generate visualizations and infographic-like summaries that make complex data more accessible [Dib23]. This modular visualization approach breaks down intricate data into user-friendly insights, demonstrating the effectiveness of generating grammar-agnostic and contextually enriched summaries. We build on this approach by employing LLMs to produce similar dynamic, user-specific onboarding content that can explain individual dashboard components comprehensibly and concisely, thus bridging the gap between complex datasets and user understanding.

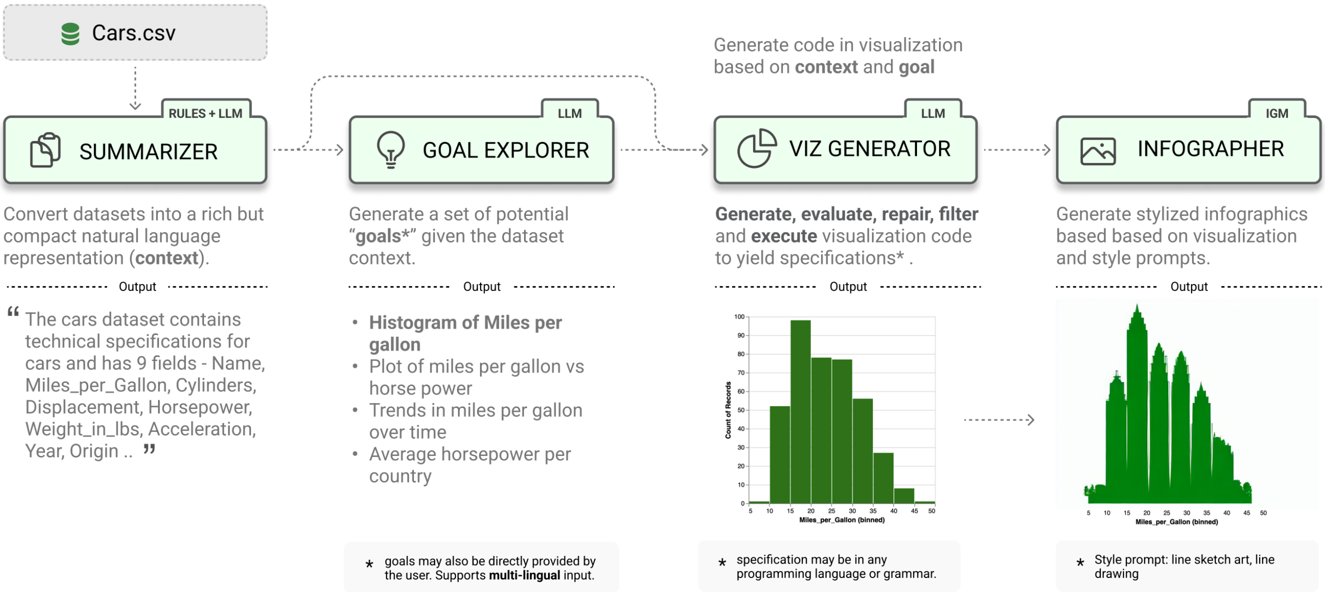

Figure 2.4: LIDA produces visualizations and infographics through four modules: data summarization, goal exploration, visualization generation, and infographic creation. Example outputs from each module are displayed. [Dib23]

Azimuth, introduced by Srinivasan et al., is a dashboard tool tailored for screen reader accessibility, aimed at blind and low-vision users [SHHM23]. It employs structured navigation and detailed descriptions of visual elements, significantly improving accessibility. We draw inspiration from Azimuth by using LLMs to provide structured, descriptive onboarding narratives for dashboard elements, making our solution inclusive for users with varying data literacy levels.

Moreover, Liew and Mueller's work on generating engaging captions for visual data underscores the importance of presenting content effectively [LM22]. Their research highlights how prompt engineering can help create engaging, contextually rich captions, enhancing user understanding. This aligns with our objective to use LLMs to generate meaningful, narrative-driven onboarding content that not only describes visual elements but also situates them within their broader context, making the onboarding process more intuitive and memorable.

In summary, previous studies provide valuable insights into enhancing dashboards through user-specific adaptations, narrative-driven onboarding, and accessible data presentation. Our research builds on these foundations by leveraging LLMs to create a responsive, user-centric onboarding process. By integrating LLMs' adaptability, contextual understanding, and ability to generate dynamic, engaging content, our approach aims to significantly improve dashboard usability for users with diverse expertise and learning preferences. This work contributes to the broader goal of making data visualization more approachable and inclusive, enhancing how users interact with complex data systems.

CHAPTER 3

Concepts

The development of this project followed an iterative approach, where prototyping played a critical role in evaluating design decisions and refining the onboarding experience. The aim was to leverage Large Language Models (LLMs) to provide dynamic, personalized onboarding for data visualization dashboards.

In the concept phase, we began by exploring Copilot+ [Mic24a], which offers AI-driven support features within Power BI. Copilot+ is a generative AI tool that enhances data exploration by helping users interact with their reports more intuitively. The concept phase focused on ensuring that Copilot+ did not interfere with the onboarding experiences we aimed to deliver, as Copilot+ includes features such as generating summaries, creating narratives, and writing queries. Specifically, we tested various Copilot+ functionalities to ensure they complemented our onboarding content without redundancy.

Additionally, a Miro board (Figure 3.1) was employed to brainstorm onboarding features/strategies. The Miro board enabled collaborative idea generation, which included mapping out user journeys, defining onboarding checkpoints, and sketching user interface elements. By consistently reflecting back on user needs, the Miro board ensured that the design remained intuitive and aligned with the target audience's expectations, emphasizing ease of use and minimizing cognitive load for new users.

The thesis was structured into three phases to ensure a logical and effective user journey for the prototype:

- **Prompting LLMs to Generate an Order of Onboarding:** This phase focused on creating an effective sequence for onboarding new users. Key elements such as **sequence design** and **dynamic ordering** were prioritized to ensure that information was structured logically and adaptively. **Contextual relevance** was integrated to refine the process iteratively, addressing diverse user needs.

- **Creating Workflows for Different User Roles:** Recognizing the diversity of user responsibilities, this phase aimed to design role-specific onboarding paths. Components like **branching** and **role-based narratives** ensured that workflows were tailored to individual expertise and objectives.
- **Evaluating Workflows and Stories on Different Dashboards:** A focus on **dashboard diversity** and **visual consistency** ensured that the onboarding experience remained seamless across different dashboards.

Figure 3.1: Overview of the miro board made. The numbers symbolize the priority of the goals for this thesis, 1. to 3. have been implemented.

To maximize the effectiveness of the LLMs, prompt engineering strategies were employed. They involve carefully crafting input prompts to elicit informative and contextually accurate responses from LLMs. For this project, several prompt engineering techniques were employed in this project to achieve a high level of contextual understanding and adaptability:

- **Zero-shot Prompting**: Zero-shot prompting relies on the model's inherent knowledge to answer queries without providing explicit examples or additional context. This approach is ideal for generating straightforward explanations or answering general questions, such as "What does this chart show?" For these scenarios, the prompts were designed to focus on clarity and conciseness, ensuring that users received quick, actionable information about dashboard elements. [DAI24]
- **Few-shot Prompting**: Few-shot prompting involves providing the model with a few examples of desired outputs alongside the main prompt. This technique was instrumental in generating tailored onboarding content, particularly when explaining complex or domain-specific visualizations. By including examples, the model was guided to produce responses that were both contextually relevant and aligned with the style and depth of previous explanations. For instance, when describing advanced filters or multi-layered graphs, examples helped refine the tone and level of detail in the output. [DAI24]
- **Chain-of-Thought Prompting**: Chain-of-thought prompting encourages the model to generate step-by-step responses that outline the logical sequence involved in solving a problem or performing a task. This approach was particularly effective for onboarding users through multi-step processes, such as applying filters, configuring visualizations, or interpreting advanced analytics. By structuring responses into clear, incremental steps, this technique enhanced user comprehension and reduced the potential for errors. [DAI24]
- **Context-Aware Prompting**: Context-aware prompts incorporated details about the visual element (e.g., its type, purpose, or layout) and user interactions (e.g., selected filters or hovered elements). These prompts enabled the model to generate responses that were highly specific to the user's current context, such as describing the relationship between a selected data filter and its impact on a visualization.

CHAPTER 4

Implementation

Throughout the concept and especially the implementation phase, a number of tools and frameworks were employed to achieve a functional system. The implementation phase began with the use of Microsoft Power BI [Mic24b] and Microsoft Azure to establish the application environment. Microsoft Power BI served as a foundational tool to understand the structure and complexity of existing data visualizations, including assessing how different types of visual elements (e.g., charts, graphs, tables) interacted and how they were structured. This understanding was crucial for designing onboarding prompts that would effectively address users' needs.

The frontend of the onboarding system was developed using Vue.js, along with Vue Flow [CT24] for graph-based visualization and navigation. The dashboard interface (Figure 4.1) shows an overview, where the chat interface for user interaction is positioned on the left side of the dashboard, with pop-ups overlaying the visualization elements to provide targeted assistance and context-sensitive information.

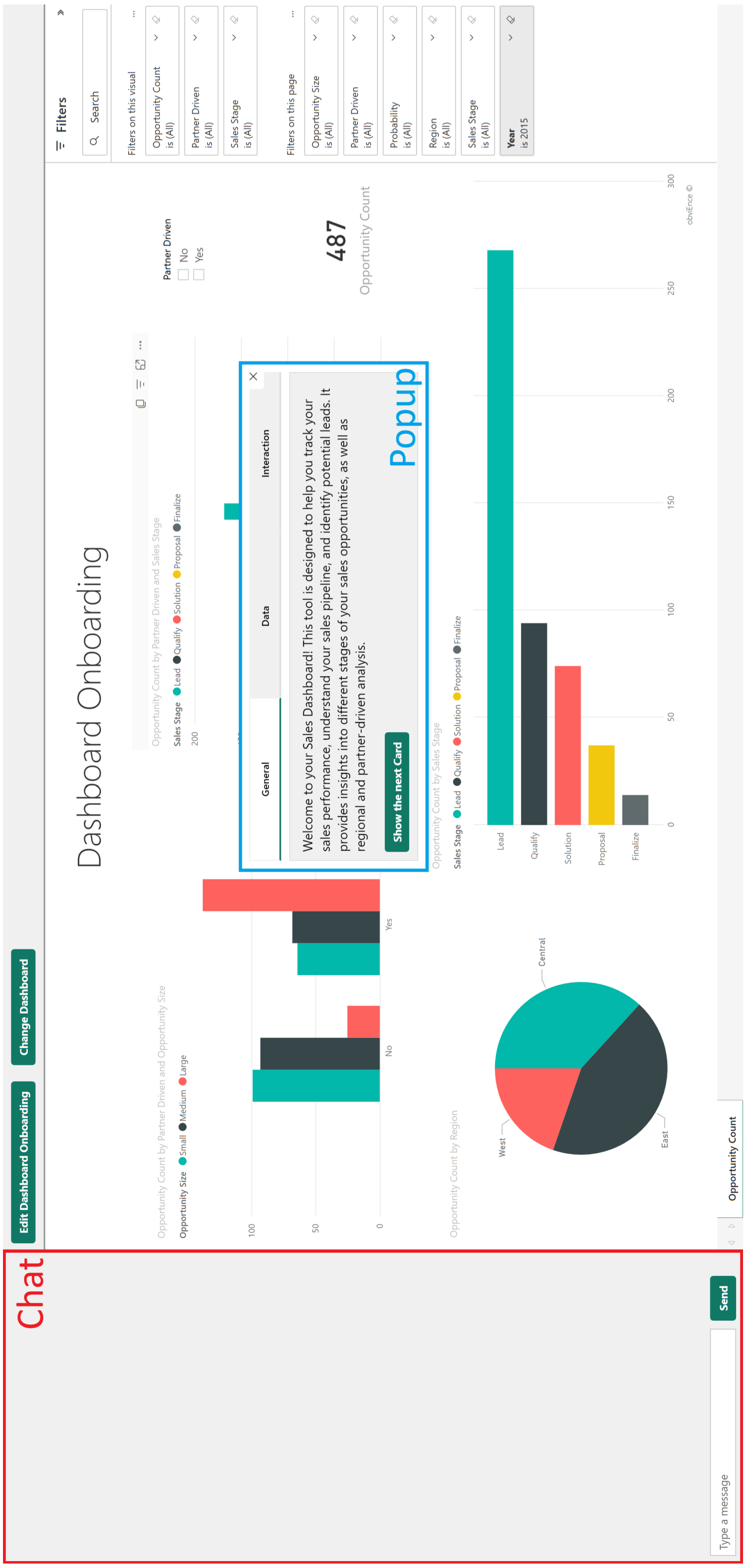

Figure 4.1: Overview of the dashboard view with the chat interface on the left side and popups providing assistance.

Figure 4.2 shows the onboarding interface, featuring the selection of nodes (that are all the individual dashboard visuals of the dashboard selected) shown in Figure 4.1, alongside a grid for selecting the vis/domain expertise level from low to high, which is used to customize the onboarding. This is accessed by clicking on an button in the dashboard view.

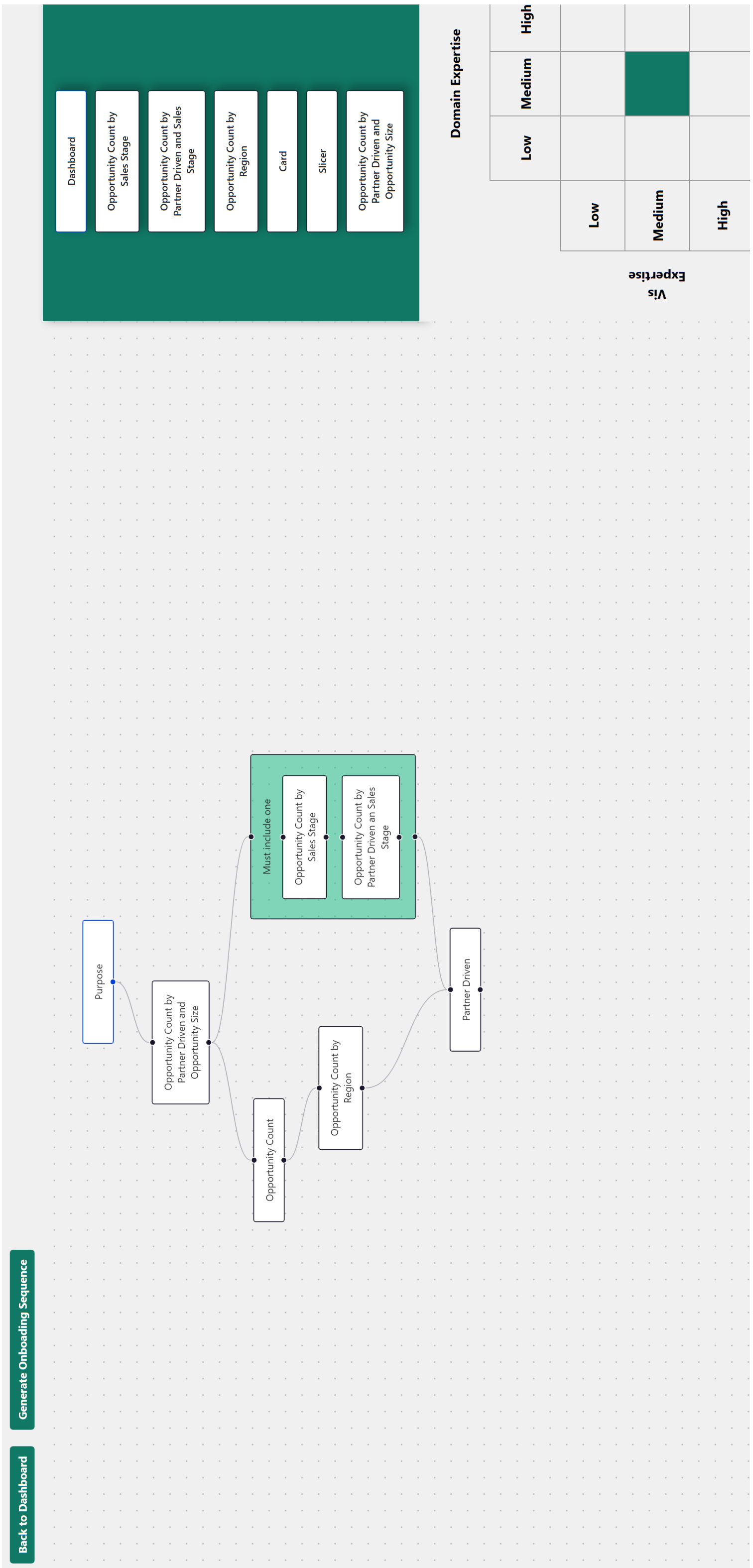

Figure 4.2: The onboarding interface for the dashboard shown in Figure 4.1, featuring an node selection on the right based on the individual visuals of the dashboard and an expertise level customization.

To even retrieve data/visuals from Power BI, we utilized the `powerbi-client` library, allowing seamless integration between Power BI reports and our onboarding platform. This integration allowed for the extraction of visual elements, layout information, and underlying data directly from Power BI, as seen in the Vue.js script in Listing 4.1.

```
report.on('loaded', function () {
  report
    .getPages()
      .then((pages) => {
        pages.forEach((page) => {
          page.getVisuals().then((visuals) => {
            visuals.forEach(async (visual) => {
              console.log(visual)
              let layout = visual.layout

              delete layout.displayState
              delete layout.z

              try {
                const visualDataCSV = await visual.exportData(
                  models.ExportDataType.Summarized,
                  1000
                )
                const visualData = parseCSV(visualDataCSV.data)
                store.addNode({
                  name: visual.title || visual.name,
                  type: visual.type,
                  node: 'default',
                  layout: layout,
                  data: visualData
                })
              } catch (error) {
                console.error(`Error exporting data for visual ${
                    visual.name}:`, error)
              }
            })
          })
        })
      })
    .catch((error) => {
      console.error('Error getting report pages:', error)
    })
})
```

Listing 4.1: The frontend function for getting the visuals and their respective summarized data from the dashboard.

The backend was implemented using Python with Flask, utilizing Socket.IO for real-time chat capabilities. Two LLMs, Llama 3.1 and ChatGPT 4o, were integrated to generate responses based on user prompts. Llama 3.1 was accessed using OOlama's API. ChatGPT 4o was accessed through the OpenAI API. That enables flexibility between models for conversational output depending on the user's expertise level and the nature of the question. The following Listings show the backend function for generating responses using Llama 3.1 (Listing 4.2) and ChatGPT 4o (Listing 4.3) respectively.

```
import requests
from flask import jsonify
import os

def llama_generate_response(prompt):
  url = 'http://localhost:11434/api/generate'
  payload = {
    "model": "llama3.1",
    "prompt": prompt,
    "stream": False
  }
  headers = {'Content-Type': 'application/json'}

  response = requests.post(url, json=payload, headers=headers)
  response_data = response.json()

  return response_data.get('response', '')
```

Listing 4.2: Backend function using Llama 3.1

```
from openai import OpenAI
import os

# function to interact with OpenAI API
def openai_generate_response(prompt, max_tokens, system_message):
  key = os.getenv('OPENAI_KEY')
  if key is None:
    print("error: OPENAI_KEY is not set.")

  client = OpenAI(
    api_key=os.getenv('OPENAI_KEY'),
  )
  response = client.chat.completions.create(
    model="chatgpt-4o-latest",
    messages=[
      {"role": "system", "content": system_message},
      {"role": "user", "content": prompt}
    ],
    max_tokens=max_tokens
  )
  return response.choices[0].message.content
```

Listing 4.3: Backend function using ChatGPT 4o

This setup allowed the system to offer personalized onboarding by using different models to respond to user queries and provide nuanced explanations based on the user's chosen expertise level. To achieve this, prompt engineering was applied to optimize the interaction between users and the LLMs. Significant time of the implementation involved crafting specialized prompts to generate onboarding content dynamically. To maximize the effectiveness of prompt engineering, each prompt was designed to include one or more of the following components:

- **Visual Context**: Specific details about the dashboard element, such as its type (e.g., bar chart, scatter plot, table) and its function or purpose.
- **User Queries**: Anticipated questions or actions from users, tailored to varying levels of expertise (e.g., "What does this pie-chart indicate?").
- **Targeted Instructions**: Detailed guidance on how to interact with or interpret the visual element, ensuring clarity and usability.

Below is the prompt for generating onboarding content for a given dashboard with different visualizations (Listing 4.4).

```
You are an assistant that generates the onboarding for the given dashboards.
Your task is to generate onboarding messages for a user interface based on a given sequence of steps and associated data.
Instructions:
  - Understand the Sequence: Review the provided sequence of steps, which is represented as a Vue flow graph in JSON format with nodes and edges.
  - Analyze Node Data: For each node in the sequence, examine the associated data provided.
  - Generate Onboarding Content: For each step, produce:
    - General: A clear and concise description of the steps purpose.
    - Data: Any relevant data or information that enhances understanding.
    - Interaction: Specific instructions or interactions the user can perform at this step.
```

Listing 4.4: System-Prompt for generating onboarding content. The result of this prompt is a json array containing the content of the general, data and interaction tabs as shown in the popup on Figure 4.1

Also a model for following the logical progression of the onboarding based on user interactions is implemented. Figure 4.2 illustrates this adaptive flow using a decision tree model based on the nodes. These nodes are an representation of all visuals the dashboard consists of, gathered by 4.1. Following Dhanoa et al. [DHF+25], the expertise is displayed via a grid for the domain / vis experience for that particular dashboard that the user can choose from. The system then evaluates these existing nodes and the user's expertise level (persona) to determine the onboarding dynamically (Listing 4.5).

```
You are an expert assistant specialized in generating onboarding
    sequences for dashboards. Your task is to create a comprehensive
    and user-friendly onboarding sequence based on the provided nodes
    and persona.
Guidelines:
  - Analyze the provided nodes and persona to understand the context
    and requirements.
  - Create a logical flow of steps that guide the user through the
    dashboard effectively.
  - Use parent and child nodes to indicate decision points and
    branches.
  - Ensure the sequence is clear, concise, and easy to follow.
  - Return the result as a JSON object containing two keys: "nodes"
    and "edges".
  - Strictly follow the structure of the provided templates for nodes
     and edges.
```

Listing 4.5: System-Prompt for generating the sequence. The templates that contain valid vue flow representation of nodes and edges in json are provided seperatly.

For the sequence (Listing 4.5), the system uses a combination of backend logic and API integrations. Prompts were dynamically constructed based on user interactions, and the LLM-APIs (either Llama 3.1 or ChatGPT 4o) were invoked to generate responses. The backend processed inputs, formulated structured prompts, and parsed the LLM's output to ensure it adhered to the required format, which is shown for the dashboard sequence generation (Listing 4.6).

```
    @app.route('/onboarding/sequence', methods=['POST'])
    def onboarding_sequence():
        # retrieve data sent with the request

        data = request.get_json()
        nodes = data.get('nodes', [])
        persona = data.get('persona', '')

        # constructing the prompt
        prompt = ...

        # call LLM-API
        response = openai_generate_response(prompt=prompt, ...)

        # parse the response as JSON
        try:
            json_response = json.loads(response_text)
        except json.JSONDecodeError:
            json_start = response_text.find('{')
            json_end = response_text.rfind('}') + 1
```

```python
            json_text = response_text[json_start:json_end]
            json_response = json.loads(json_text)

        return jsonify(response)
```

Listing 4.6: Simplified version (excluding the prompt show in Listing 4.5) of the onboarding sequence generation. The code retrieves the data sent with the request, constructs the prompt calls the api for the llm (Listing 4.3) and parses the response as an json object before returning it.

In essence, these adaptive, LLM-driven prompts allow the system to respond intelligently and in real time, providing the right level of onboarding instruction depending on a user's role, expertise level, and curiosity. By combining data extracted from Power BI with a carefully orchestrated sequence of prompts and decision logic, the onboarding experience becomes both flexible and contextually grounded.

CHAPTER 5

Results

The prototype was evaluated using two sample Power BI dashboards, "Opportunity Count Overview" (Figure 4.1) and "Revenue Opportunities Report" (Figure 5.1) to gauge how well the LLM-based onboarding adapts to different data structures and visual elements. These dashboards contained various charts, filters, and interactive components, presenting an ideal test environment for assessing the system's ability to provide tailored guidance.

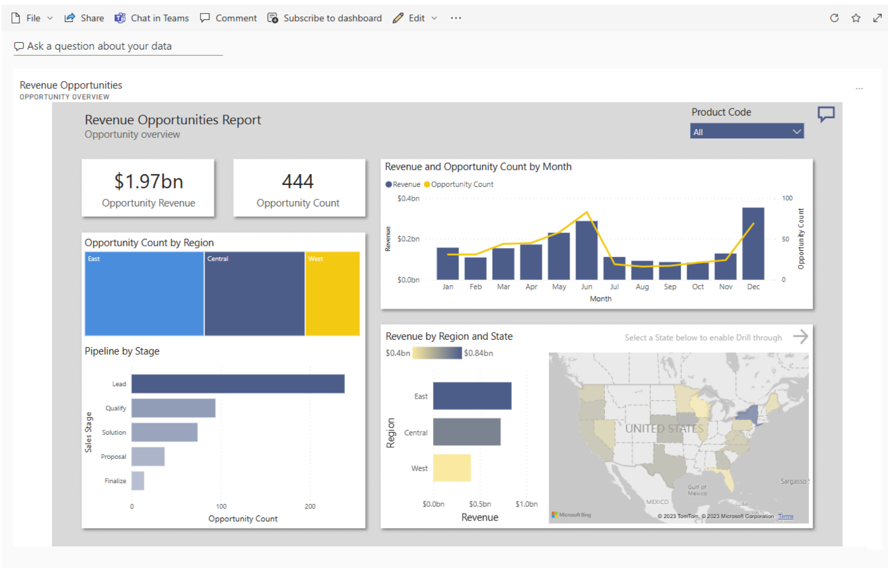

Figure 5.1: The revenue opportunities sample provides a report and semantic model for a software company with direct and partner sales channels, enabling sales managers to monitor opportunities and revenue by region, deal size, and channel.

When a user first accesses the dashboard, the system automatically detects the constituent visual elements such as bar charts, pie charts, and tables by querying the Power BI report via the `powerbi-client` library (see Listing 4.1). The user is then offered an onboarding interface (Figure 4.2) where they can select which visualizations they need help understanding or interacting with, specify their domain knowledge and visualization proficiency on a low-to-high grid, receive context-specific explanations, and engage in conversational guidance through a chat interface positioned on the left side of the dashboard. These explanations appear in pop-ups over the selected visuals, describing why the visual matters, how the data is calculated, and which interactions are possible. The LLMs (Llama 3.1 or ChatGPT 4o) respond in real time to provide clarifications, suggest next steps or propose alternative views of the data. Through this process, the user experiences an adaptive and personalized onboarding session. For instance, an expert in data visualization but new to the domain might see fewer explanations of basic chart elements, while a domain expert with limited visualization experience will receive more fundamental guidance on interpreting chart layouts and applying filters.

The results exhibit dynamic prompt generation, where each onboarding message is generated via prompts (Listings 4.4 and 4.5) incorporating the user's selections, the dashboard's visuals, and the user's declared expertise. This approach replaces static tutorial scripts with concise, context-aware explanations. A decision-tree model, combined with the "grid" input of user expertise, provides adaptive sequencing rather than a rigid, linear tutorial. During testing, novice users were guided through basic tasks first, while advanced users were shown immediately how to apply complex filters. Because the system integrates seamlessly with Power BI's API, real-time information on each visual is obtained. Layout properties are used to position pop-ups, and summarized datasets inform the explanatory content, ensuring that the onboarding remains relevant even if the dashboard data changes.

The LLM-based approach unlocks new possibilities by offering immediate, user-specific explanations that align with the user's chosen expertise. The onboarding sequence can adapt based on user queries, personalizing the path for different roles, such as sales managers or business analysts. By moving beyond stating "what" a visual shows to explaining "why" it matters—such as highlighting the relevance of analyzing forecasted revenue—this system enables more context-aware storytelling. These features go beyond traditional "one-size-fits-all" tutorials, providing deeper context for each visual element and handling user questions on demand.

Several challenges arose during prototype development. Integrating smoothly with the `powerbi-client` library required multiple iterations to ensure correct retrieval of visual layouts and data summaries, especially as dashboards updated. The prompts also needed tuning to balance detail and brevity, ensuring thorough yet digestible explanations. Managing diverse user contexts demanded a carefully debugged decision-tree approach, with tests across different combinations of domain and visualization expertise to ensure logical consistency. By successfully resolving these challenges, the prototype shows how real-time, adaptive onboarding can reduce the friction of learning new dashboards.

CHAPTER 6

Conclusion and Future Work

This thesis set out to address the challenges associated with onboarding users to complex data dashboards by leveraging the adaptability and contextual awareness of Large Language Models (LLMs). We proposed and implemented an LLM-driven onboarding prototype that offers personalized, dynamic guidance, making data dashboards more approachable for users with diverse expertise levels compared to reading and understanding dashboards independently. By utilizing prompt engineering, adaptive sequencing, and conversational interactions, the prototype demonstrated the potential of LLMs to generate context-specific onboarding content that can enhance comprehension and user engagement.

Through the research, we found that LLMs are effective in producing descriptive and tailored onboarding experiences, especially when careful prompt engineering is used to ensure accuracy and relevance. Adaptive sequencing based on visual hierarchy proved to be particularly beneficial while testing the prototype with multiple dashboards, as it allowed information to be presented logically and progressively according to a users need. Furthermore, integrating a conversational chat feature added an extra layer of adaptability, allowing users to ask clarifying questions and receive personalized responses, further enhancing the onboarding process.

While the prototype provides a promising approach to enhancing dashboard usability, several areas remain open for further exploration and improvement. First, transforming the onboarding experience into a more interactive and multimedia-rich format by adding audio could significantly enrich user engagement. Future iterations of the prototype could incorporate video timelines, Text-to-Speech (TTS) capabilities, and interactive visual elements, enabling users to navigate content in more varied ways, revisit specific sections seamlessly, and experience a more immersive learning journey.

Moreover, moving from a prototype to an easy to deployable solution will require enhancing the system's robustness and generalizability. This includes refining the codebase, optimizing LLM integration and developing a more streamlined, user-friendly interface. Further work could also explore the integration of alternative LLMs or different configurations to enhance response times, accuracy, and support for accessibility, such as ensuring compatibility with screen readers and other assistive technologies.

Additionally, future research could benefit from extensive user studies to evaluate and quantify the effectiveness of LLM-driven onboarding in real-world settings. Evaluations could compare the LLM-based approach against traditional onboarding methods to measure improvements in usability, user satisfaction, and learning outcomes. By adopting a more rigorous evaluation strategy, we can provide concrete evidence of the impact that LLM-driven onboarding has on enhancing users' ability to interact with data dashboards effectively.

List of Figures

Bibliography

[CLL+24] Kiroong Choe, Chaerin Lee, Soohyun Lee, Jiwon Song, Aeri Cho, Nam Wook Kim, and Jinwook Seo. Enhancing Data Literacy On-demand: LLMs as Guides for Novices in Chart Interpretation . *IEEE Transactions on Visualization & Computer Graphics*, (01):1–17, June 2024.

[CT24] Burak Cakmakoglu and Vue Flow Team. Vue flow: Interactive graphs and flow visualization for vue 3. `https://vueflow.dev/`, 2024. Version 1.41.4, Last accessed: 2024-11-05.

[DAI24] DAIR.AI. Prompt engineering guide. `https://www.promptingguide.ai`, 2024. Last updated on September 9, 2024.

[DHF+25] Vaishali Dhanoa, Andreas Hinterreiter, Vanessa Fediuk, Niklas Elmqvist, Eduard Gröller, and Marc Streit. D-tour: Semi-automatic generation of interactive guided tours for visualization dashboard onboarding. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):721–731, 2025.

[Dib23] Victor Dibia. LIDA: A tool for automatic generation of grammar-agnostic visualizations and infographics using large language models. In Danushka Bollegala, Ruihong Huang, and Alan Ritter, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 113–126, Toronto, Canada, July 2023. Association for Computational Linguistics.

[DWH+22] Vaishali Dhanoa, Conny Walchshofer, Andreas Hinterreiter, Holger Stitz, Eduard Groeller, and Marc Streit. A process model for dashboard onboarding. *Computer Graphics Forum*, 41(3):501–513, 2022.

[LM22] Ashley Liew and Klaus Mueller. Using large language models to generate engaging captions for data visualizations. *NLVIZ: Exploring Research Opportunities for Natural Language, Text, and Data Visualization (NLVIZ, Workshop jointly held with IEEE VIS)*, 2022.

[Mic24a] Microsoft Documentation Team. Intro to copilot for power bi. `https://learn.microsoft.com/en-us/power-bi/create-reports/copilot-introduction`, October 2024. Last accessed: 2024-11-05.

[Mic24b] Microsoft Power BI Team. Power bi playground. `https://playground.powerbi.com/en-us/`, 2024. Last accessed: 2024-11-05.

[SHHM23] Arjun Srinivasan, Tim Harshbarger, Darrell Hilliker, and Jennifer Mankoff. Azimuth: Designing accessible dashboards for screen reader users. In *Proceedings of the 25th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '23, New York, NY, USA, 2023. Association for Computing Machinery.

[YN24] Fernando Yanez and Carolina Nobre. User-Adaptive Visualizations: An Exploration with GPT-4. In Daniel Archambault, Ian Nabney, and Jaakko Peltonen, editors, *Machine Learning Methods in Visualisation for Big Data*. The Eurographics Association, 2024.