AutoFinViz: An LLM-powered Financial Visualization Tool

Hok Yin Yu (Boris)
hokyin.yu@mail.utoronto.ca

Yu Hau (Howard) Chen
howardyh.chen@mail.utoronto.ca

ABSTRACT

Due to the rise of LLMs and machine learning, several noteworthy Automated Visualization (AutoViz) related research papers have emerged, e.g. Voyager 2 [7], Data2Viz [5] ,and LIDA [4]. All of them utilized the power of machine learning or LLMs to improve users' effectiveness and efficiency in the data visualization task. However, their applications are broad, spanning various domains, and they do not specifically cater to the intricate requirements of the financial field. To address this gap, we present AutoFinViz as a tailored solution that integrates LLM capabilities uniquely designed for financial visualization. AutoFinViz aims to navigate the nuances and complexities inherent in the financial sector, harnessing the power of LLMs to provide a user-centric financial data analysis experience. It comprises four modules: Classifier, Summarizer, Question Formulator, and Visualizer. Together, these modules form a seamless pipeline that transforms financial datasets into insightful plots, offering an end-to-end solution for comprehensive financial data analysis. Code is available at this url - `https://github.com/HowardChen123/AutoFinViz`.

Author Keywords

Information Visualization; Large Language Model; Financial Analysis

INTRODUCTION

Visualizing financial datasets, which often include time series data, multi-dimensional variables, and categorical factors, typically requires domain-specific knowledge to ensure accurate representation. Manual visualization tasks in finance can be time-consuming and prone to human error, presenting challenges for individuals with limited visualization experience, particularly when handling vast and varied datasets. Traditional tools, such as Tableau Software [2] and Microsoft Power BI [1], often segregate the processes of exploratory data analysis (EDA), question formulation, and visualization, leading to fragmented workflows.

To streamline this workflow, several noteworthy Automated Visualization (AutoViz) related research papers have emerged, such as Voyager 2 [7], Data2Viz [5], and LIDA [4]. These papers highlight the utilization of machine learning and Large Language Models (LLMs) to enhance users' effectiveness and efficiency in data visualization tasks. However, these tools may not capture the specificity and granularity required for nuanced financial analyses, potentially leading to oversimplified or misleading visuals. The domain-agnostic nature of many current tools can result in the omission of crucial financial indicators or relationships.

Concurrently, Large Language Models like GPT-4 [3] demonstrate their ability to generate and modify visualizations based on textual descriptions and requirements, thanks to their advanced code-writing and natural language understanding capabilities. These models can comprehend intricate financial terminologies and nuances, ensuring visuals are both accurate and relevant. Given the unique characteristics of financial data, there is a pressing need for a tool specifically tailored to the financial domain. An LLM-based tool can ensure visualizations adhere to financial standards and practices, providing professionals with visuals that align with their domain-specific needs.

Thus, in our project, we introduce AutoFinViz, a Python library that integrates LLM capabilities uniquely designed for financial visualization. AutoFinViz is engineered to adeptly navigate the intricacies of the financial sector by leveraging the capabilities of Language Models. It comprises four key modules: Classifier, Summarizer, Question Formulator, and Visualizer. Working in concert, these modules transform complex financial datasets into meaningful and insightful plots. This holistic approach offers an end-to-end solution for conducting comprehensive and insightful financial data analysis.

RELATED WORK

Automated Visualization (AutoViz)

Automated Visualization (AutoViz) has emerged as a prominent topic in recent years, with numerous research papers exploring ways to simplify the visualization process of datasets. Broadly, these approaches fall into two categories: heuristics-based and learning-based.

Heuristics-based approaches involve exploring the properties of data to generate a search space of potential visualizations, subsequently ranking these possibilities and presenting them to users. An exemplary instance of this method is Voyager 2 [7], an innovative mixed-initiative system that integrates both manual and automated chart specification. This approach enables users to engage in open-ended exploration while also assisting in targeted question answering. However, despite its strengths, heuristics-based systems like Voyager 2 can be challenging to maintain, may only cover a limited spectrum of the visualization space, and often fail to fully utilize the valuable information embedded in existing datasets.

On the other hand, learning-based approaches aim to automatically learn the process of transforming data into visualizations. For instance, Data2Viz [5] exemplifies this approach with an end-to-end trainable neural translation model that automatically generates visualizations from given datasets. This model treats visualization generation as a language translation problem, mapping data specifications to visualization specifications in a declarative language (Vega-Lite). However, this method also has its limitations, such as the need for a collection of training pairs and comparatively weaker error control mechanisms.

LIDA

LIDA represents a significant advancement in the field of automated visualization, distinguished by its utilization of Large Language Models for the automatic generation of grammar-agnostic visualizations and infographics. The tool is structured around four key modules:

- SUMMARIZER: Converts data into a succinct yet comprehensive natural language summary.
- GOAL EXPLORER: Identifies potential visualization goals derived from the provided data.
- VISUALIZATION GENERATOR (VISGENERATOR): Responsible for generating, refining, executing, and filtering visualization code to effectively represent the dataset.
- INFOGRAPHER: Creates stylized graphics that are both aesthetically appealing and faithful to the data, utilizing Information Graphic Models (IGMs).

In addition to these modules, LIDA features a Python API and a hybrid user interface that blends direct manipulation with multilingual natural language capabilities. This interface significantly enhances the user experience, allowing for interactive chart creation, infographics design, and the generation of compelling data stories.

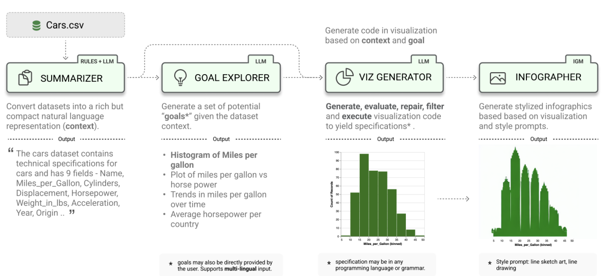

Figure 1. LIDA pipeline

However, we identified a limitation in LIDA's approach: its general application scope does not cater specifically to any domain. This leads to a domain-agnostic issue, particularly noticeable in fields requiring high specificity and granularity, such as Financial Analysis. Despite this, LIDA's well-defined system architecture has been a source of inspiration for the development of AutoFinViz. Furthermore, the evaluation metrics introduced by LIDA, notably the Visualization Error Rate (VER), have proven to be reasonable and applicable to our development process.

AUTOFINVIZ

AutoFinViz, inspired by the system architecture of LIDA, is composed of four pivotal modules: a CLASSIFIER, a SUMMARIZER, a QUESTION FORMULATOR, and a VISUALIZER. Each module plays a crucial role in processing dataset information and generating insightful graphs and plots about the input data. For our Large Language Model (LLM) integration in AutoFinViz, we selected GPT-3.5turbo, leveraging its advanced capabilities. The development of AutoFinViz also involved the use of LangChain, which significantly streamlines the process of integrating LLMs into our system.

In addition, we chose StreamLit for our user interface due to its simplicity and effectiveness in creating interactive web applications. This choice enabled us to craft an intuitive and user-friendly interface for AutoFinViz, enhancing the overall user experience.

AutoFinViz meticulously applies Munzner's Nested Model [6] to deliver a specialized Automated Visualization solution tailored for financial data analysis. At the **Domain Problem Characterization** level, our Classifier module identifies the specific interests of our target audience, whether they be in market, economy, or corporate sectors. Concurrently, the Question Formulator, powered by the extensive knowledge of LLMs, generates relevant questions about the data. In the **Data/Operation Abstraction Design** stage, the Summarizer module translates the dataset into a natural language format, making it understandable to the LLM. Subsequently, the Visualizer module determines the most appropriate visualizations based on the data summary and formulated questions. Furthermore, during the **Encoding/Interaction Technique Design** phase, the Visualizer integrates diverse visual encodings and interactions, employing Retrieval Augmented Generation to identify the most pertinent visualizations for the given data scenario. Although the **Algorithm Design** level does not directly apply to our project, the preceding stages ensure a holistic, domain-specific approach to financial data visualization, closely aligned with Munzner's model and underscoring the innovative use of LLM capabilities in AutoFinViz.

Data Collection & Classifier

In AutoFinViz, our focus is on comprehensively covering three primary categories of financial datasets: Market Data, Economic Data, and Corporate Financial Data. To ensure a robust and diverse data foundation, we have sourced Market Data, which includes stock and commodity prices, as well as exchange rates, from Yahoo Finance. Our Corporate Financial Data, encompassing transaction and cash flow information, is acquired from MorningStar. Additionally, for Economic Data, we utilize key economic indicators such as GDP, unemployment rates, and inflation figures, sourced from Statistics Canada.

Our aim with AutoFinViz is to excel in the visualization of financial information across these varied domains, providing a comprehensive tool for financial data analysis. The Classifier module plays a pivotal role in this process. Its primary responsibility is to categorize the input datasets into these three distinct categories. To achieve this, we incorporate the column names of the dataset into the prompts used to engage the Large Language Model (LLM). This approach enables the LLM to accurately select one of the categories. Once a category is chosen, we can then apply tailored, category-specific prompts in the subsequent modules to ensure the generation of relevant and precise visualizations.

Summarizer

The Summarizer's objective is to convert raw data into a natural language summary, providing a dense informational context for visualization tasks. This transformation occurs through three stages:

Stage 1: Utilizing a Large Language Model (LLM) to generate new metrics derived from the original dataset.

Stage 2: Providing a comprehensive summary for each column, including data types, statistics, and examples.

Stage 3: Employing the LLM to define and elucidate the meaning and content of each column.

This methodology ensures that the LLM is equipped with the necessary context for accurate data visualization. An example of the summarizer's output is depicted in Fig. 2.

Figure 2. Summarizer Output

Question Formulator

Using the summaries generated by the Summarizer, our Question Formulator module generates insightful questions aimed at data exploration within financial and economic contexts. This is achieved through meticulous prompt engineering, guiding the Large Language Model (LLM) to focus on relevant topics. The process involves multiple layers: the LLM formulates questions, identifies suitable visualization types for these questions, and selects the appropriate x-axis and y-axis columns for graph construction.

In order to facilitate the process of AutoFinViz, the prompts provided to the LLM include a comprehensive list of potential visualization types, tailored to the dataset's category. This specificity helps narrow down the selection of suitable plot types. Additionally, we include descriptions of each graph type in the prompts to aid the LLM in understanding their unique requirements. An example of the output from the Question Formulator can be seen in Fig. 3.

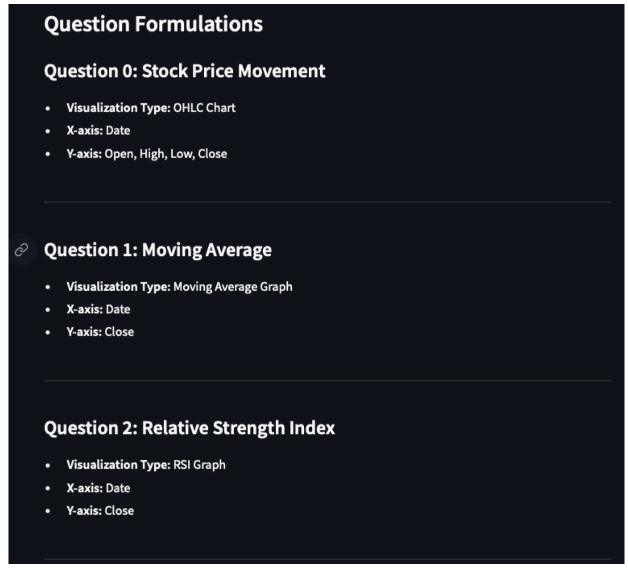

Figure 3. Question Formulator Output

Visualizer

The Visualizer is the final component of AutoFinViz. It utilizes the data summary, question dictionary, and the dataset itself to produce visualizations along with their corresponding code. Initially, we experimented with LIDA's methodology, employing Code Scaffold and Chain-of-Thought for direct code generation. Code Scaffold provides a pre-defined framework to guide the software development process, while Chain-of-Thought outlines the logical reasoning in code generation, offering insights into the decision-making process. However, we found this approach to sometimes yield unstable and less meaningful plots.

To address this, we shifted to using Retrieval-Augmented Generation. RAG combines information retrieval with natural language generation, allowing the LLM to access relevant data from an external source, such as our database of sample plots, and generate content based on this information. This method enhances the relevance and quality of the generated content. By retrieving plot information relevant to the dataset's category, the LLM is better equipped to generate executable code that aligns with the dataset's context, as illustrated in Fig. 6. The execution of this generated code, resulting in higher-quality visualizations, is depicted in Fig. 5. This approach significantly improved the stability and quality of the visualizations produced by AutoFinViz.

User Interface

User Interface was implemented using StreamLit and it consists of several sections.

OpenAI API Key and Data Upload

This section of the interface is dedicated to inputting the OpenAI API Key, an essential requirement for the application's functionality. Additionally, it provides the capability for users to upload their dataset, as illustrated in Fig. 7. Upon successful upload, users can preview a sample of the dataset through a table view.

Figure 4. Visualizer Output - Visualization

Figure 5. Visualizer Output - Visualization Samples

```
import pandas as pd
import plotly.graph_objects as go

def plot(df: pd.DataFrame):
    """
    Plots an OHLC (Open, High, Low, Close) chart using Plotly, suitable for financ

    Parameters:
    df (pd.DataFrame): DataFrame containing the OHLC data.

    Returns:
    fig: Plotly figure object representing the OHLC chart.
    """

    # Ensure the date column is in datetime format
    df['Date'] = pd.to_datetime(df['Date'], infer_datetime_format=True, errors='co

    #OHLC Plot
    fig = go.Figure(go.Ohlc(x=df['Date'], open=df['Open'], high=df['High'], low=df
    fig.update_layout(title="Stock Price Movement")

    return fig

# DO NOT MODIFY ANYTHING BELOW
# Dataframe 'df' is already defined so NO NEED TO CONSTRUCT DATAFRAME "df"
fig = plot(df)
```

Figure 6. Visualizer Output - Code

Summarization

Upon the upload of a dataset, the SUMMARIZER module is automatically activated, producing a concise summary of the dataset's contents. This summary is immediately displayed to the user. An exemplary display of the summarization output is presented in Fig. 2.

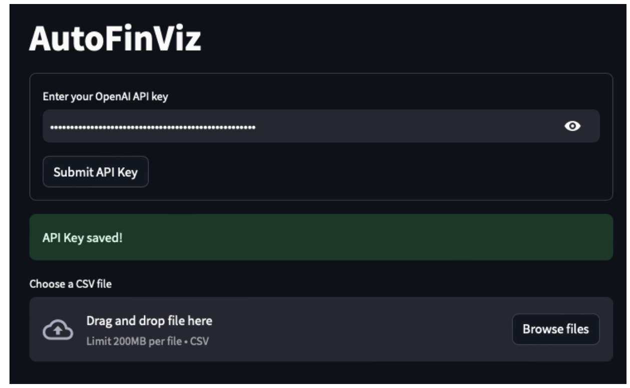

Figure 7. AutoFinViz API

Question Formulation and Visualization

Following the summarization of the data, the application activates the Question Formulator. This module presents a curated list of potential visualizations to the user, as depicted in Fig. 3. Subsequently, the Visualizer module is engaged, generating the executable code for the chosen visualization. This process culminates in the display of the final visualization, as shown in Fig. 4.

EVALUATION

In evaluating the effectiveness of AutoFinViz, we adopted the Visualization Error Rate (VER) metric, originally used by LIDA, to assess the reliability of our visualization pipeline.

Visualization Error Rate is calculated as the percentage of generated visualizations that lead to code compilation errors. This metric is crucial for gauging the reliability of the AutoFinViz pipeline. It not only provides insights into the system's current dependability but also helps evaluate the impact of various modifications, such as prompt engineering or updates to the scaffold. This evaluation is vital in ensuring that AutoFinViz remains a robust and reliable tool for financial data visualization.

The formula for VER is given as:

$$VER = \frac{E}{T} * 100$$

Where E represents the number of generated visualizations with code compilation errors, and T is the total number of generated visualizations.

At present, our system has achieved a Visualization Error Rate of approximately 4.5% across our sample datasets, indicating a high level of reliability and efficiency in the AutoFinViz pipeline.

FUTURE WORKS

Looking ahead, we plan to engage financial experts to review the plots generated by AutoFinViz. Their expertise will be invaluable in assessing the quality and relevance of our visualizations, ensuring they meet the high standards required in financial analysis. Another key goal is to enhance AutoFinViz's robustness across a wider range of financial datasets.

This improvement will enable users to delve into a broader array of financial scenarios, enriching their analytical capabilities.

Additionally, exploring real-time data streams represents a significant extension of our project. In the fast-paced financial sector, the ability to perform immediate analysis and make timely decisions is crucial. To support this, we recognize the need to optimize the efficiency of AutoFinViz, enabling it to handle real-time data streaming effectively. These advancements will ensure that AutoFinViz remains at the forefront of financial data visualization and analysis.

CONCLUSION

In this project, we introduced AutoFinViz, an innovative Financial Visualization Tool powered by Large Language Models (LLMs). By adapting the design architecture from LIDA and harnessing the LLM's capabilities in visualization generation, AutoFinViz specifically addresses the limitations of current domain-agnostic Automated Visualization tools in the context of financial data analysis. It provides an end-to-end solution that encompasses everything from classifying dataset categories to visualizing insightful, financially-focused graphs. This approach significantly simplifies the process of deriving meaningful insights from raw datasets for users.

Furthermore, we developed a user-friendly interface to enhance accessibility and exploration of our library. In assessing the effectiveness of AutoFinViz, we employed visualization error rates as our primary evaluation metric. Our aspiration with AutoFinViz is to streamline the visualization workflow for the general public, making financial data analysis more approachable and insightful. We believe that AutoFinViz has the potential to be a valuable tool in simplifying and enhancing the understanding of financial datasets for a wide range of users.

ACKNOWLEDGMENTS

We extend our gratitude to Prof. Fanny Chevalier and TA Zixin Zhao for their invaluable and insightful feedback on our project. Their guidance significantly contributed to the refinement and success of our work. Additionally, we are thankful to all our classmates who took the time to review our project video. Their constructive comments played a crucial role in the successful completion of our research on AutoFinViz.

REFERENCES

[1] 2023. Microsoft Power BI. `https://powerbi.microsoft.com/`. (2023). Accessed: 2023-12-22.

[2] 2023. Tableau Software. `https://www.tableau.com/`. (2023). Accessed: 2023-12-22.

[3] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of Artificial General Intelligence: Early experiments with GPT-4. (2023).

[4] Victor Dibia. 2023. LIDA: A Tool for Automatic Generation of Grammar-Agnostic Visualizations and Infographics using Large Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*. Association for Computational Linguistics, Toronto, Canada, 113–126. DOI: `http://dx.doi.org/10.18653/v1/2023.acl-demo.11`

[5] Victor Dibia and Çağatay Demiralp. 2018. Data2Vis: Automatic Generation of Data Visualizations Using Sequence to Sequence Recurrent Neural Networks. (2018).

[6] TamaraMunzner. 2009. A Nested Model for Visualization Design and Validation. (2009).

[7] Kanit Wongsuphasawat, Zening Qu, Dominik Moritz, Riley Chang, Felix Ouk, Anushka Anand, Jock Mackinlay, Bill Howe, , and Jeffrey Heer. 2017. Voyager 2: Augmenting visual analysis with partial view specifications. *ACM CHI* (2017).