SYM-AM-24-100

Excerpt from the Proceedings of the Twenty-First Annual Acquisition Research Symposium

Acquisition Research: Creating Synergy for Informed Change

May 8–9, 2024

Published: April 30, 2024

Prepared for the Naval Postgraduate School, Monterey, CA 93943.

Naval Warfare Studies Institute

DDM Department of Defense Management Naval Postgraduate School

The research presented in this report was supported by the Acquisition Research Program at the Naval Postgraduate School.

ACQUISITION RESEARCH PROGRAM
DEPARTMENT OF DEFENSE MANAGEMENT
NAVAL POSTGRADUATE SCHOOL

Large Language Model (LLM) Comparison Research

William M. Fisher—is a data science fellow in the Cost Analysis and Research Division at the Institute for Defense Analyses (IDA). He has a background in applied math and data analytics and has worked on projects on large language models and space policy research. He has a BS in Computational and Applied Mathematics from the University of Chicago.. [wfisher@ida.org]

Nicholas Wagner—is a research staff member in the Cost Analysis and Research Division at the Institute for Defense Analyses (IDA). He specializes in data analytics and machine learning and has experience with projects in personnel, force planning, maintenance, contracts, health care, and AI policy, along with some IDA data governance on the side. He is currently working remotely out of Arizona. He has a BS in Materials Science and Engineering from Arizona State University and a PhD in Materials Science and Engineering from Northwestern University. [nwagner@ida.org]

Kevin Garrison—is a research staff member in the Information Technology and System Division at the Institute for Defense Analyses (IDA). He specializes in data and analytics for the defense department, acquisition oversight, cybersecurity, AI, and machine learning. He previously worked as the Chief of Analytics for the Office of the Department of Defense Chief Information Officer. He has a BS in Astronautical Engineering from the U.S. Air Force Academy. [kgarriso@ida.org]

Abstract

Over the past few years, large language models (LLMs) have rapidly increased in capability, with OpenAI's GPT-4 being the most prominent example. This case study explores two ways that GPT-4 could be used to assist research tasks: data analysis and writing executive summaries. We chose these tasks because they are common to Institute for Defense Analyses (IDA) projects and because they are often presented as tasks appropriate for LLMs. First, we used GPT-4 to conduct tasks such as data cleaning, exploration, modeling, and visualization. We compared the quality and speed to a human doing the same task. We found analysis quality was insufficient when utilizing AI alone, but improved greatly with a human partner. Using GPT-4 saved about 60% of the time on the data analysis assignment and presents an opportunity for significant cost savings in this area. Then, we used the GPT-4 to generate executive summaries (EXSUMs) for three publicly available IDA publications, and we compared these to the human-generated EXSUMs. We found that the LLM-generated EXSUMs often failed to provide appropriate context for more technical papers, but that given the speed that they are generated and their thoroughness, LLMs still present time- and cost-saving opportunities.

Introduction

As large language models (LLMs) have improved and increased in capability over the past few years, many organizations are examining how they can be employed to enhance research productivity. For IDA researchers, an LLM could be used to automate and/or assist tasks such as data analysis and writing executive summaries (EXSUMs). At the moment, the most prominent LLM developer is OpenAI, who released their most recent model, GPT-4, in March 2023. In early July 2023, OpenAI added a new feature to GPT-4 called Code Interpreter, which was then renamed GPT-4.[1] This feature, which is now automatically enabled for all GPT-4 chats, can write and execute Python code, and allows users to upload files and ask it to analyze data, create charts, and edit the files in place. The limitations include a strict time limit for executing code and an inability to save work beyond the immediate session.

In this analysis, we examined these features of GPT-4 to determine the possible time and cost savings of using an LLM for data analysis and writing EXSUMs. First, we measured how long it took both a human and GPT-4 to conduct various data analysis tasks, such as

[1] ChatGPT Plus subscribers can access GPT-4 at https://chat.openai.com/?model=gpt-4-code-interpreter, and they can learn more at https://www.nytimes.com/2023/07/11/technology/what-to-know-chatgpt-code-interpreter.html.

exploration, modeling, and visualization, which provided quantitative measures of time to perform non-CUI data analysis tasks. We then compared analysis quality. For writing EXSUMs, we used the GPT-4 API to generate summaries for publicly available IDA publications and compared the output to the existing EXSUMs for the publications. Through this comparison process, we can determine how much of a quality difference there is in the summaries and how much time a researcher can save by using an LLM to summarize a research paper rather than doing it entirely unassisted.

Dataset

We used two datasets for this analysis. The first comes from Kaggle, a public website, and describes Airbnb activity in New York City during 2023 (Kumar, 2023). The data includes information on prices, reviews, locations, and more. We chose this dataset because it has a variety of options for exploratory analysis and modeling, while being simple enough to ensure we would be able to easily understand and critique the decisions made by GPT-4. We also used a dataset from the Department of Defense Office of Local Defense Community Cooperation on state-by-state spending as a defense-specific example (OLDCC, n.d.). This dataset was used to focus more on GPT-4's data visualization capabilities.

Data Analysis Comparison Methods

To determine how much time GPT-4 can save and how it compares in terms of the quality of its work, we performed data analysis operations on the dataset ourselves before turning to the LLM. For each of the following sections, we briefly describe the work we did with that done by GPT-4 and compare the results. The comparison is based on the estimated time taken for data analysis, as well as the time it would have taken to do everything GPT-4 did. This comparison is not perfect, as GPT-4 sometimes went beyond what we were asking or required steps to be run locally, but the work was similar enough to provide estimates of the time GPT-4 can save on these tasks. The code for this section and the EXSUM generation section can be found at https://code.ida.org/users/wfisher/repos/llm_crp_code/browse.

Data Cleaning and Exploratory Analysis

Human Task Details

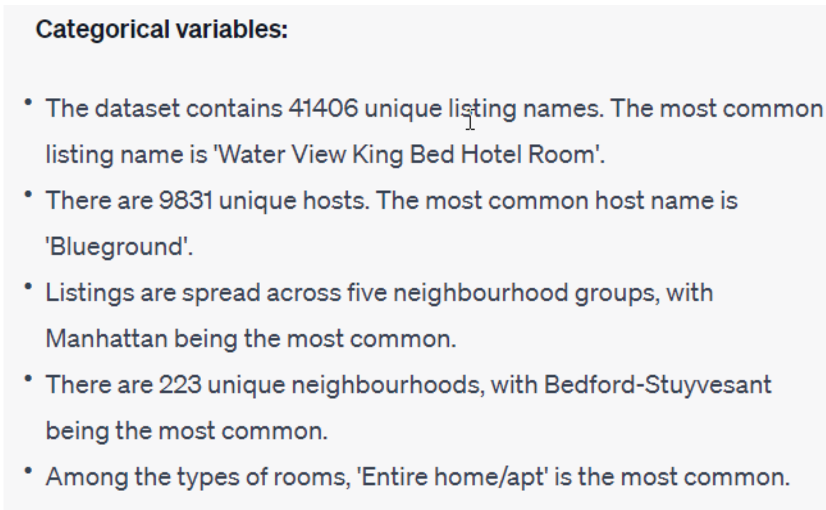
We began data cleaning and exploratory analysis by reviewing the different variables in the dataset to ensure we understood them and agreed with their associated data types. We then performed standard data cleaning operations, such as removing duplicate rows, checking for columns with mixed types, and removing rows with null values. After this, we conducted exploratory analysis by checking the summary statistics and distributions of the numerical variables, as well as unique values of the categorical ones, to better understand the data. We also explored how price varied across different neighborhoods and determined which ones were the most and least expensive.

GPT-4 Task Details and Prompts

We began the second half of the data analysis exercise by using GPT-4 to clean the data and perform an exploratory analysis on the dataset, which allowed us to check for data quality issues and understand the nature of the data it was working with before moving on to modeling and visualization.

Our prompt was, "Clean the data and conduct exploratory analysis on the nature of the different variables in the dataset." We used this prompt because these are often the first steps for a human working with a new dataset and to understand how GPT-4 would respond to broad terminology like "exploratory analysis." In response, it looked for three things in particular: columns with missing values, columns with mixed types, and duplicate rows. GPT-4 quickly removed duplicate rows, as well as a column containing license codes that did not appear to be useful given the amount of missing values and mixed types. However, it inserted zeroes for missing values in the “last_review” and “reviews per month” columns, which is questionable, particularly in regard to using those variables to model something like rental price. For example, more popular, and therefore more expensive, rentals may be reviewed more often than less popular ones, meaning that there could be a relationship between the “last_review” and “reviews per month” variables and rental price that is obscured by inserting zeroes for missing values rather than using something like the mean for the neighborhood.

As far as exploratory analysis, GPT-4 explored descriptive statistics and correlations between numerical variables and frequency distributions for categorical variables. It summarized its main findings in bullet points, which are shown in Figures 1 and 2.


Categorical variables:

- The dataset contains 41406 unique listing names. The most common listing name is 'Water View King Bed Hotel Room'.
- There are 9831 unique hosts. The most common host name is 'Blueground'.
- Listings are spread across five neighbourhood groups, with Manhattan being the most common.
- There are 223 unique neighbourhoods, with Bedford-Stuyvesant being the most common.
- Among the types of rooms, 'Entire home/apt' is the most common.

Figure 1. GPT-4 Findings on Categorical Variables


Numerical variables:

- The average price per night for a listing is about $200, but the range is vast, going from $0 to $99,000. This large range and the high standard deviation suggest that there are some extreme values (outliers) in the price.
- The average minimum number of nights is around 18, but it goes up to 1250, suggesting there are some long-term listings.
- On average, a listing has about 26 reviews. The maximum number of reviews a listing has received is 1842, indicating some very popular properties.
- The 'calculated_host_listings_count' shows that, on average, a host has around 24 listings, but the maximum goes up to 526, suggesting the presence of big players or property management companies.

Figure 2. GPT-4 Findings on Numerical Variables

The bullet points in the two figures provide useful information, such as the skewed distribution of the price, minimum nights, number of reviews, and the most common neighborhood. GPT-4 also makes a significant inference from this exploratory analysis in noting that that the skewed number of listings for hosts may indicate the presence of significant real estate figures and property management companies. It also produced histograms for certain variables, shown in Figure 3 with comments in Figure 4, and calculated correlations between variables, shown in Figure 5 with comments in Figure 6.

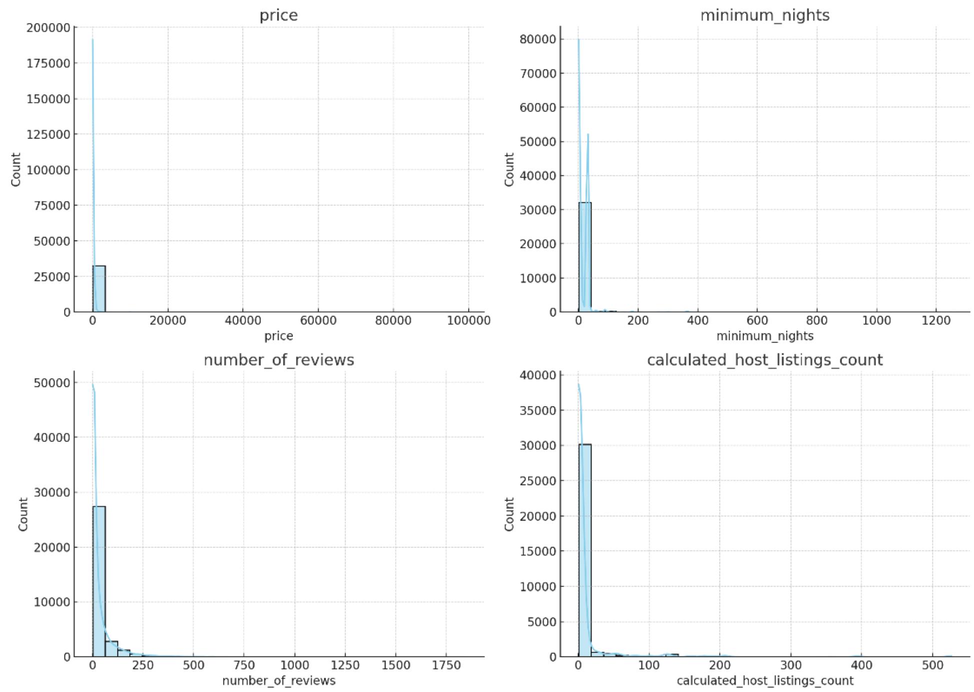


Figure 3. GPT-4 Histograms of Select Numerical Variables

Histograms:

- The `price` histogram shows that the vast majority of prices per night are below $10,000, with a few extreme outliers above that. This is consistent with the statistics we saw earlier.
- The `minimum_nights` histogram reveals that most listings require a stay of fewer than 200 nights, but there are some that require a much longer minimum stay.
- The `number_of_reviews` histogram indicates that most listings have fewer than 200 reviews, but a few have many more.
- The `calculated_host_listings_count` histogram shows that the vast majority of hosts have fewer than 100 listings, but a few hosts have a significantly higher number.

Figure 4. GPT-4 Comments on Histograms

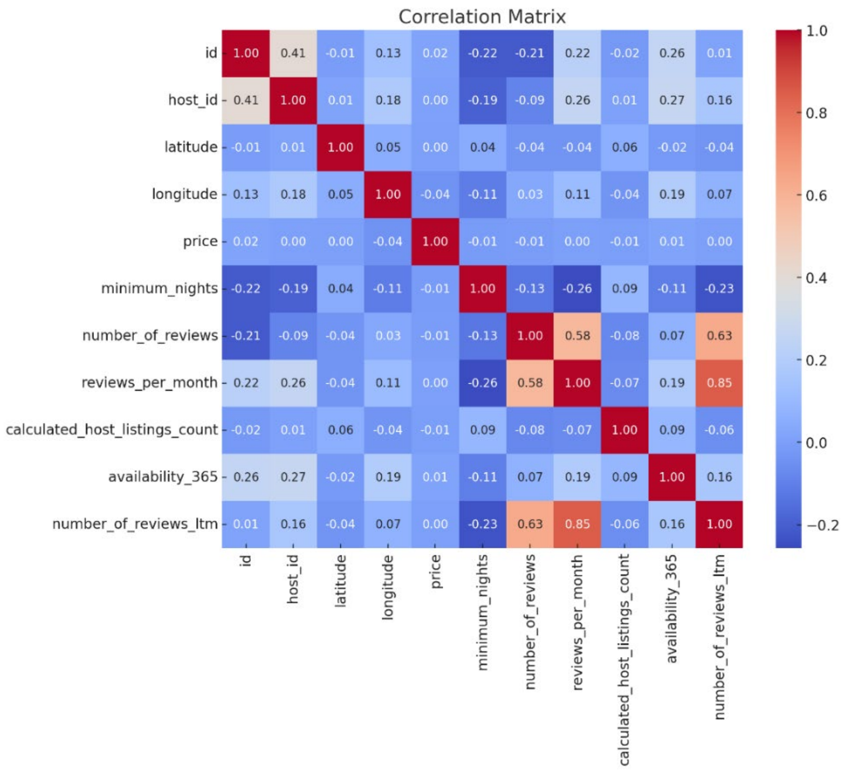
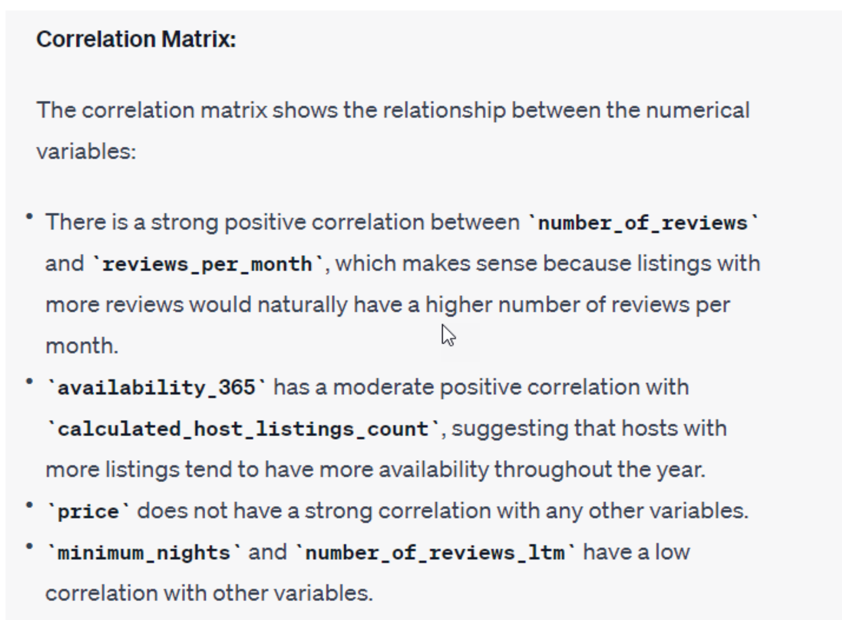


Figure 5. GPT-4 Correlation Matrix between All Numerical Variables



Correlation Matrix:

The correlation matrix shows the relationship between the numerical variables:

- There is a strong positive correlation between `number_of_reviews` and `reviews_per_month`, which makes sense because listings with more reviews would naturally have a higher number of reviews per month.
- `availability_365` has a moderate positive correlation with `calculated_host_listings_count`, suggesting that hosts with more listings tend to have more availability throughout the year.
- `price` does not have a strong correlation with any other variables.
- `minimum_nights` and `number_of_reviews_ltm` have a low correlation with other variables.

Figure 6. GPT-4 Comments on Its Correlation Matrix

Figures 4 and 6 again provide exploratory analysis, noting that price, which is going to be the focus of modeling, does not have a strong correlation with any individual variable and also noting some cutoff numbers for continuous variables. However, some of these cutoff numbers are poorly chosen. For example, GPT-4 notes that the vast majority of prices are below $10,000, but only one percent of prices are above $1,250, meaning that the $10,000/night threshold is only useful for a narrow range of analysis questions. It does note that hosts with more listings should have more availability throughout the year. It is important to remember that GPT-4's conclusions are drawn from machine-readable data used to make the plots and not from any visual understanding. This is particularly noticeable for conclusions drawn about the histograms, where a human would likely choose different cutoff numbers than those used by GPT-4.

Time/Quality Comparison

Whereas identifying and removing rows and columns with missing or unhelpful data required me to program multiple loops to identify and remove these things, GPT-4 completed them instantly. Additionally, as Kaggle did not have a data dictionary for the dataset, we had to manually check and ensure we understood the meanings of variable labels such as availability_365 (the number of days in a year when a listing is available), which GPT-4 was able to do immediately.

For the exploratory analysis, we checked the different distributions of the numerical variables, reviewed distributions of the continuous variables, and checked the most expensive neighborhoods. Given the prompt to conduct exploratory analysis on the nature of the variables in the dataset, GPT-4 went beyond our effort by analyzing the categorical variables, drawing conclusions based on how variables were distributed, and calculating correlations between all the variables.

As GPT-4 went beyond our cleaning and exploratory analysis, there is no exact comparison of how much time was saved by using it, but it would have likely taken around 45 minutes to generate the same visualizations and write the same conclusions. Given that we had to review the writeup and data to filter out the best results, the GPT-4 likely saved around 30 minutes or 66% of our time.

Building a Model

Human Task Details

To begin with the modeling section, we first preprocessed the data by removing columns that would not be relevant, such as "id" and "host_id," and replacing categorical variables with dummy columns. We then split the data into training and testing data and used a linear regression model to predict the price variable. We also looked for other possible modeling approaches for this kind of data. We first used the CatBoost Regressor package, which is meant to work well with high amounts of categorical data. As that package did not improve upon the linear regression, we then used decision tree and random forest regressors to predict the price variable, which produced better results. We used R^2 and RMSE values to measure the model results, which are given in Table 1.

Table 1. Test Set Performance for Models Created Manually

Model	R^2	RMSE
Linear Regression	0.17	189.93
Decision Tree	0.05	238.09
Random Forest	0.42	152.78

GPT-4 Task Details and Prompts

After cleaning and exploring the data, we tested GPT-4's ability to build a model predicting the price variable. This involved deciding what kind of modeling approach to use, what kind of preprocessing steps to take, and how to tune the model for optimized performance.

Our prompt to GPT-4 was, “Generate a multivariate model to predict the price variable, and evaluate the accuracy of your model.” Before making a model, GPT-4 responded with a list of preprocessing steps, shown in Figure 7.

Before we create and evaluate the model, we need to prepare the data:

1. Handle outliers: Since the `price` variable has some extreme values, we should handle these outliers to make our model more robust. We can cap the prices at a certain threshold, such as the 99th percentile.
2. Feature encoding: The dataset contains some categorical variables. We'll need to convert these into a format that can be used in the model. One common approach is one-hot encoding, which converts each category value into a new column and assigns a 1 or 0 (True/False) value.
3. Feature selection: Not all variables in the dataset may be useful for predicting the price. We'll need to select the features that are most likely to have an impact.
4. Train-test split: We'll split the data into a training set (to train the model) and a test set (t

Regenerate response

Figure 7. GPT-4 Preprocessing Steps

The first two preprocessing steps were fairly straightforward, and GPT-4 handled them without issue. However, the feature selection step was less clear. In response, GPT-4 generated a list of columns to drop. Some of these should likely be dropped, such as the “id” variable, but GPT-4 also chose to drop the “date of last review” variable without explanation. In addition, GPT-4 sent the following message, shown in Figure 8, in response to a preprocessing error.

It appears there was an error because the column `id` was included in the numerical columns for the model, but it was dropped from the dataframe.

Let's correct this by excluding the columns `id`, `host_id`, `latitude`, and `longitude` from the numerical columns. We'll exclude `host_id` because it's an ID that won't carry any useful information for price prediction. And we'll exclude `latitude` and `longitude` for now to keep the model simple. If the model's performance is not satisfactory, we can consider adding these back in.

Let's correct this and run the model again.

Figure 8. GPT-4 Error Explanation

GPT-4 offers little reason for dropping the “latitude” and “longitude” variables beyond “keep[ing] the model simple.” Considering the role location might play in determining Airbnb prices, it seems like a strange decision.

After fixing its preprocessing issues and selecting variables to exclude, the GPT-4 ran a linear regression model on the data to predict the price. The initial performance is superior to the one we trained manually but still poor in absolute terms (RMSE = 133.43 and R^2 = 0.22 on the testing data). Asked to improve the model further, GPT-4 suggests some next steps (see Figure 9).

There are several potential ways to improve the model:

- Adding more relevant features: If more data is available, including additional features that could influence the price (e.g., property size, amenities, ratings) could improve the model.
- Handling non-linear relationships: If the relationship between the predictors and the price is not linear, using a model that can handle non-linear relationships (e.g., polynomial regression, decision trees, or neural networks) could provide better results.
- Tuning the model: Depending on the model used, there may be hyperparameters that can be tuned to improve performance.
- Advanced techniques: More advanced techniques, such as ensemble methods or deep learning, could potentially yield better results, though they may also be more complex and computationally intensive.

Please let me know how you would like to proceed.

Figure 9. GPT-4 Modeling Next Steps

Following these suggestions, we prompted GPT-4 to "try using more advanced modeling techniques to improve the results on the test set." In response, GPT-4 used decision tree and random forest regressors to model the results. GPT-4 achieved much better test set results using the random forest regressor (R^2 = 0.49, RMSE = 112.46) than it did using linear regression or the decision tree regressor (R^2 = 0.12, RMSE = 155.17), but it seemed to be overfitting, which led to the message in Figure 10.

The Random Forest model has a higher RMSE on the training set compared to the Decision Tree, suggesting it's not fitting the training data as closely. However, it performs much better on the test set, with a lower RMSE and higher R^2.

This indicates that the Random Forest model is more generalizable and less prone to overfitting than the single Decision Tree model.

Although the performance on the test set has improved, the R^2 is still relatively low. There may be other factors influencing the price that are not included in the dataset. Additionally, more complex modeling techniques such as gradient boosting or neural networks may provide further improvements.

Please let me know how you would like to proceed.

Figure 10. GPT-4 Model Explanation and Suggestions

We then asked GPT-4 to tune the hyperparameters for the model, but that operation timed out, as did another attempt with a smaller hyperparameter grid. However, the threshold for runtime within GPT-4 is strict, so we copied the code into our machine and waited about 4 minutes for it to run, which it did successfully.

Time/Quality Comparison

When comparing the time taken on modeling tasks, we found that GPT-4 was again faster, but there was less to do in preprocessing than in data cleaning and exploratory analysis so the time benefit was smaller. GPT-4 did make a decision that we had not considered: it removed the extreme values of the price variable, as it was very skewed. But GPT-4 also made a mistake in choosing to remove the longitude and latitude variables.

We also started with a linear regression model. It took us some time to review options and consider the data to select between decision trees and random forest models as an alternative. Meanwhile, as seen in Figure 10, GPT-4 was able to not only determine which model option is better, but which factors to consider and what the next steps might be in just one response. We had the benefit of not working with the strict time limit of the GPT-4 environment, which allowed us to tune the random forest model hyperparameters. While GPT-4 could not execute the hyperparameter tuning code in its environment without timing out, the code for such operations was outputted by GPT-4 very quickly, and we could then run it on our own computer. The end result were models with slightly higher predictive accuracy than the ones we created manually.

Overall, the GPT-4 saved an estimated 10–15 minutes in the preprocessing steps of this exercise and another 35–45 minutes selecting which models to use, quickly setting them up, and then observing their results. Considering our own process took about 75 minutes; GPT-4 would have likely saved 60%–80% of our time.

Data Visualization

Human Task Details

For the last step in the data analysis, we used the data on state-by-state defense spending from the Office of Local Defense Community Cooperation. First, we used the data to generate simple exploratory visualizations on different details of defense spending. We then created more complex geographic visuals of the dataset.

GPT-4 Task Details and Prompts

In addition to cleaning, exploring, and modeling data, data visualizations are often time-consuming. Depending on the library being used, packages can require a lot of manual tuning to achieve the desired results. For the next step of our comparison, we asked GPT-4 to make data visualizations of differing levels of complexity.

GPT-4 is able to run the code for generating simple visualizations requiring only basic libraries like Matplotlib or Seaborn and can produce the graphics in the interface. To demonstrate this, we prompted GPT-4 to “generate a series of exploratory data visualizations for the attached dataset using Seaborn. Visualize details such as the top locations of defense spending, the top contractors, and the leading types of contracts. Assemble the visualizations in a 2x2 grid.” In response, GPT-4 generated the grid of images in Figure 11.

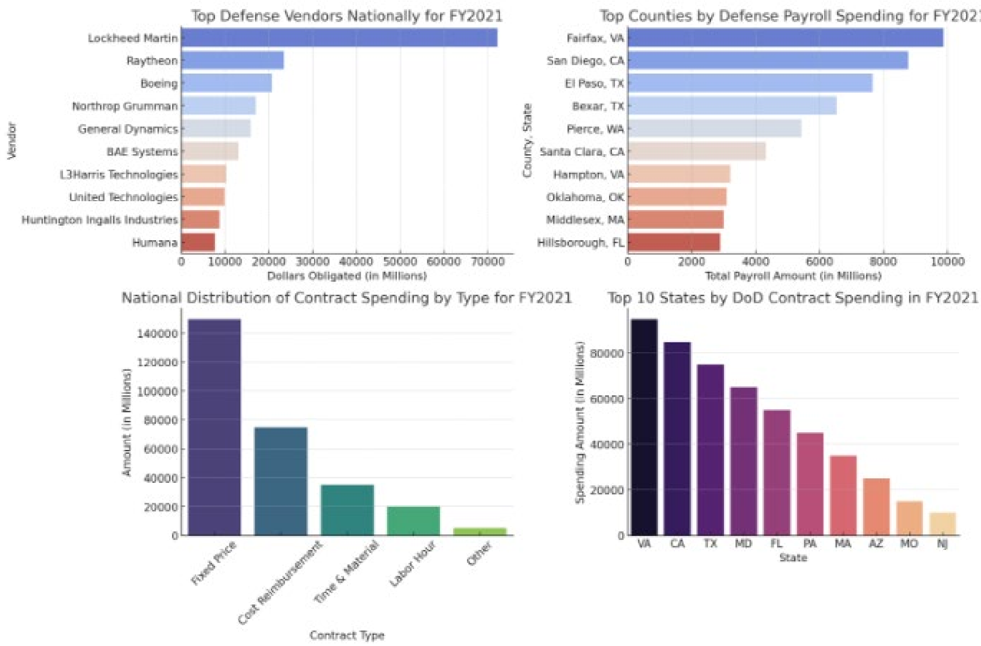


Figure 11. GPT-4 Exploratory Visualizations in Seaborn

It is worth noting that a highly specific prompt may not produce all the requested results. For example, the GPT-4 prompt, “Using the sheet with all state statistics, generate a correlation matrix for all the variables, and make scatter plots with lines of best fit for the three pairs with the highest correlations,” generated and displayed the scatter plots but only generated the correlation matrix data without a visualization. However, when we asked for the correlation matrix and the scatter plots in separate prompts, we received the desired results.

GPT-4 can also struggle with making more complex visualizations. For example, in response to the prompt, “Make two choropleth maps of spending by state for the first and last year in the data side by side,” GPT-4 replied that it could not produce the maps without a U.S. states shapefile, even though it already has access to such a file in the GeoDatasets package. After uploading a shapefile of U.S. states, GPT-4 then produced the choropleth maps shown in Figure 12.

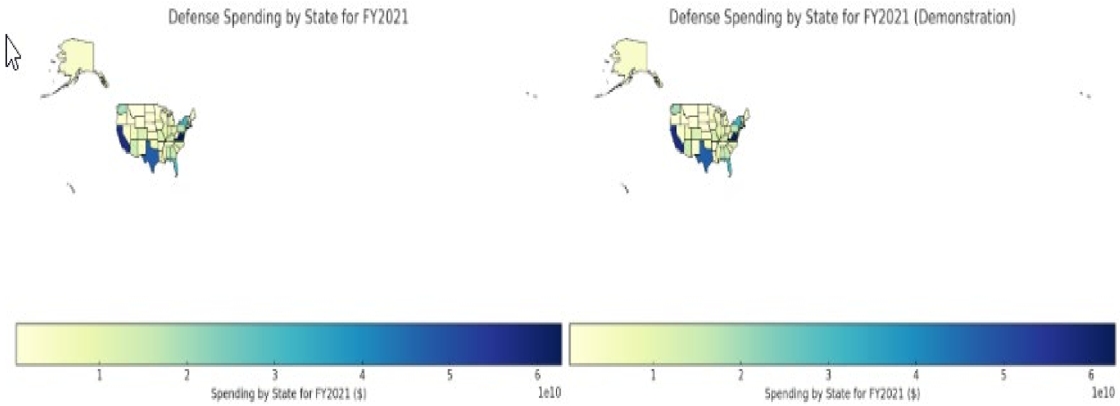


Figure 12. GPT-4-Generated Choropleth Maps

As this map displayed 2021 twice, we asked GPT-4 to run this visualization with 2021 and 2014, which is the earliest year in the data. However, it said a coding environment issue prevented it from doing so. Retrying this in a new chat, GPT-4 was not able to make the maps at all. However, the code used to generate the maps in this format was available and was retrievable so that a human could make the data and aesthetic adjustments necessary, as seen in Figure 13.

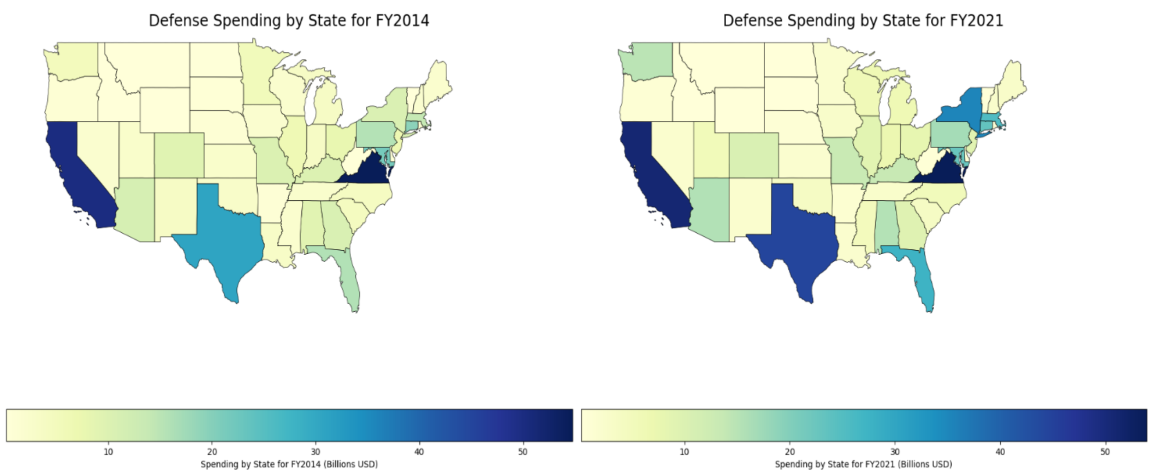


Figure 13. Corrected Choropleth Maps

In another exercise, we asked GPT-4 if it could generate a map with a two-color scale for defense contract spending and defense personnel spending by state, along with a matrix color legend. It again could not correctly merge the U.S. states shapefile with the spending data, so we asked it to generate code for such a map. Although it was able to generate code for making a two-color map or the matrix legend individually, it could not come up with a solution for plotting them together.

Time/Quality Comparison

Manually inputting scales, labels, and color schemes can be a time-consuming process for even simple visualizations, so GPT-4 certainly saves time by producing simple visualizations instantly along with the code to tweak them as needed. Although GPT-4 was unable to display complex visualizations in the interface, it is quite helpful to have an adjustable template available, as thinking through the steps to create such a template and correcting the errors made along the way takes longer than having GPT-4 generate nearly complete code. That being said, it is helpful to know the modules well enough to quickly edit GPT-4 outputs.

In terms of making visualizations, GPT-4 likely saved around 30–40 minutes. Much of the time savings comes from the lessened need to remember different syntaxes for the various visualizations, even though GPT-4 is unable to generate them in its interface. Producing our own initial exploratory visualizations with maps took about 70 minutes, so GPT-4 would have saved around 50% of our time.

Summary and Takeaways

GPT-4 can be a time saver for a researcher, particularly when used for initial analysis of a data set. Although one needs to read through its data cleaning decisions, GPT-4 recognizes the most important fixes needed before analysis and executes them much faster than a researcher. When modeling, it quickly programmed and tested different modules to determine the best option. However, more computationally intensive tuning is probably better done locally due to the strict timeout limit for code run within the GPT-4 interface. GPT-4 can quickly and capably generate simple visualizations and provide a starting point for complex visualizations; in fact, this is largely true of its capabilities in general, as it can complete simple tasks or those needed at the start of an assignment, but it requires human input for more complicated tasks. It is important to carefully read its responses and specify the changes you want it to make or implement them locally.

The use of short and specific prompts seems to be best practice for GPT-4. Prompts that are too long may not achieve everything desired, and prompts that are too broad may lead it to make inappropriate choices or decisions regarding the data. However, one should not assume that GPT-4 cannot perform a task if it fails initially. If an initial prompt does not generate the appropriate response for a problem, a user can ask GPT-4 to regenerate a response or use another more specific prompt. A user can also ask GPT-4 for further explanation on a decision it made, which can clarify decisions it made or help the user determine if a step must be approached differently. Similarly, even if GPT-4 does something well the first time around, it may struggle with the same task when asked to do it later.

GPT-4 seems most effective to a researcher as an assistant rather than a tool that will do all the work itself. For this exercise, it saved an estimated total of 1.75 to 2.15 hours of work across the different tasks, as shown in Table 2. This amounted to saving around 60% of the total time for the assignments. It could have saved more time with a more complicated dataset that required additional cleaning and preprocessing or if given more specific prompts. On the other hand, this dataset was not very large (41,410 rows), which allowed GPT-4 to run most of the code in the interface and self-correct if it detected issues; it is possible that GPT-4 may not be able to execute as many steps in the interface for a dataset containing millions of rows and the output could therefore require more review and correction from the user. That being said, GPT-4 also has value as a collaborative assistant given its ability to provide frameworks and a starting point for more complex assignments. Regardless, given that the subscription price for ChatGPT Plus is $20 a month and IDA Research Associates have an estimated average hourly rate of ~$50, this could save hundreds of dollars or more on regular data analysis assignments. Additionally, as people become more familiar with GPT-4 and the best principles for using it, it is possible that the time and cost savings coming from its use could improve beyond the numbers found in this study and that the differences in quality could be reduced.

Table 2. Summary of Time Saved and Quality Comments from GPT-4

Task	Total Time Saved (minutes)	Percent of Time Saved (%)	Quality Comments
Cleaning and Exploratory Analysis	30	67	More thorough in investigating the nature of the different variables and made important contextual inferences
Modeling	45–60	60–80	Tested the same models that we did, but made questionable preprocessing decisions and could not run everything locally
Visualization	30–40	43–57	Quickly makes simple visualizations, but requires some human adjustments for more complex ones
Total	105–130	55–68	

EXSUM Comparisons

In this section, we used GPT-4 to generate EXSUMs from publicly released IDA reports and compared the results to the published EXSUMs. The goal was to explore if and how LLM-generated EXSUMs could save time that researchers currently use to summarize their work and could also better capture all the elements of a report.

Generating an EXSUM Using GPT-4

Currently, GPT-4 has a context limit of 128,000 tokens, which translates to roughly 96,000 words. This means that the full text of almost all the publicly released IDA research papers can be uploaded as part of a prompt to GPT-4 and then queried. For each of the following papers, the full text (except for the original EXSUM) was uploaded into the ChatGPT 4 interface, and the model was prompted to "write a detailed executive summary for this paper as if you were one of the authors. Include information from all major sections of the paper." We chose this prompt instead of "Write an executive summary for this paper" because GPT-4 tended to respond with EXSUMs much shorter than the original ones, and we thought a more detailed EXSUM would be a better point of comparison.

Comparisons

In this section, we compare the EXSUMs of three papers with the GPT-4-generated EXSUMs. We also note how the summaries compare in terms of clarity and how well they captured the content of the paper. For these comparisons, it is important to note that the criteria for what makes an EXSUM better or worse is subjective and can be dependent on who the audience is. For this study, we judged the EXSUMs based on how well they represented the overall content of the paper. It took GPT-4 about 30 seconds to generate the EXSUM for each paper.

Paper 1: Factors Limiting the Speed of Software Acquisition

Link: https://www.ida.org/research-and-publications/publications/all/f/fa/factors-limiting-the-speed-of-software-acquisition

Real EXSUM for Paper 1

Improving the agility of defense acquisition is a high priority goal for both the Office of the Secretary of Defense and the Military Departments. Improving the speed at which the Department of Defense (DoD) can develop, deploy, and update software-enabled capabilities would enable more general acquisition agility, given modern defense systems' critical dependence on software. Given the need to speed up software acquisition and sustainment, it is important to understand the fundamental factors that limit how quickly software can be developed, deployed, and upgraded. These factors fall into seven principal categories; in rough order of importance: 1. Required functionality – what you need the software to do (and not do). 2. Architecture – the organizing structure of the software and its operating environment. 3. Technology maturity – to what extent the intended design uses novel solutions. 4. Resources – the people, skills, funds, data, and infrastructure needed to do the work. 5. Testing strategy – acquiring the information to fix defects early in development. 6. Contract structure – the alignment of contractor incentives with DoD satisfaction. 7. Change management – the processes for trading-off performance, schedule, cost, and sustainability. This short paper considers each category in this taxonomy in turn and examines how each affects the pace of development. Ultimately, we conclude that it will only be possible to field new software-enabled capabilities quickly on an ongoing basis if earlier programs have invested the time and effort to create an environment that supports rapid capability insertion. Ideally, this environment would include modular (and preferably open) software architecture, adequate data rights, platforms with excess space and power available, an industrial base that can provide enough people with the right skills, curated input and training data, developmental test infrastructure (including modeling and simulation resources where appropriate), and localized change management authority within the developer/stakeholder team. Putting these enabling environmental features into place will often require accepting delay, up-front expense, and reduced capability in the initial increments of those platforms. Unless Service leadership accept this reality and empower new system developers to preserve these features even when faced with cost overruns, schedule delays, and demands for greater capability up front, they will not happen. Absent this kind of empowerment, software capability insertion will continue to be as slow, expensive, and unreliable as it is today.

GPT-4 EXSUM for Paper 1

The paper, "Factors Limiting the Speed of Software Acquisition," delves into the intricate landscape of defense software development within the Department of Defense (DoD), pinpointing the multifaceted hurdles that impede rapid deployment and suggesting a holistic overhaul to enhance efficiency and agility. We begin by dissecting the prevailing acquisition framework, highlighting its misalignment with the fast-paced nature of software development. Our analysis identifies key bottlenecks, including rigid procurement processes, outdated regulatory requirements, and a culture resistant to change. We proceed to compare traditional acquisition models with agile methodologies, underscoring the latter's potential to foster iterative development, stakeholder collaboration, and responsiveness to changing requirements. Through case studies and contemporary examples, we demonstrate how adopting agile practices within the DoD can significantly reduce development timelines while maintaining high standards of quality and security. Our recommendations are structured around three pillars: policy reform, process innovation, and personnel development. We advocate for policy adjustments that recognize the unique characteristics of software as a capability, propose the adoption of agile and DevSecOps processes to enhance flexibility and efficiency, and stress the importance of cultivating a workforce skilled in modern software development techniques. The paper concludes with a call to action for the DoD to embrace these changes, emphasizing that the ability to rapidly field software capabilities is crucial for maintaining national security in an era of technological evolution.

Assessment

Both the GPT-4-generated and the original EXSUM describe the paper in largely the same ways, but the original EXSUM focuses on highlighting specific terms. The categories for limiting software acquisition are clearly broken out and labeled in the original EXSUM, whereas they are grouped into a smaller group of broader categories in the GPT-4 version. This is also the case for the proposed solutions in the paper. If specifying problem and solution terms is a priority for the authors, they must indicate that when prompting GPT-4 for an EXSUM; otherwise the generated summary covers the material in the paper similarly to the original one.

Paper 2: Forecasting Demand for Air National Guard Enlisted Initial and Technical Schooling

Link: https://www.ida.org/-/media/feature/publications/f/fo/forecasting-demand-for-air-national-guard-enlisted-initial-technical-schooling/p-32920.ashx

Real EXSUM for Paper 2

Air National Guard (ANG) airmen frequently experience delays of up to a year to enter training, which harms unit readiness and contributes to inefficient resource allocation. These delays may be caused, at least in part, by inaccurate forecasts in earlier years of training needs: the demand for basic military training (BMT) and for technical (or “tech”) school slots. To satisfy planning and budgeting requirements, the ANG must request BMT and tech school training slots up to five years in advance. Currently, individual ANG Career Field Managers (CFMs) collect data through various methods and forecast school slot demand generated by airmen who exit the ANG or switch occupations, and by end strength adjustments to particular career fields. More precise and reliable ANG school slot demand forecasts may help to reduce entry delays. To assist in developing ANG school slot demand forecasts, the Institute for Defense Analysis (IDA) produces annual “exit” forecasts for each career field subdivision, commonly referred to as three-digit Air Force Specialty Code (AFSC), for each June from 2022 through 2026. We employ the Retention Prediction Model (RPM), a machine learning (ML) capability developed by IDA, to forecast five-year retention probabilities for each airman in service in June of 2021. Since individuals can switch occupations during their time in service, we layer a Markov transition model over the raw RPM exit forecasts to predict the AFSC from which enlisted personnel will exit the ANG in the future. We then use both of these probabilities to forecast the expected number of airmen who will exit the ANG from each AFSC. This method does not account for individuals not yet in ANG service who will both join and exit during the forecasting window of July 1, 2022 to June 30, 2026. To illustrate the performance of this forecasting method, we compare the forecasts this method would have produced for each June from 2017 through 2021 to actual exits during the same period. This exercise indicates an average mean absolute error (by exit share) of 7% across all years, representing a 70% improvement from a baseline extrapolation model using historic attrition rates at the three-digit AFSC level. While these exit forecasts represent only one component of the demand generating process, this method represents a substantial improvement in ANG school slot and general attrition forecasting capabilities.

GPT-4 EXSUM for Paper 2

The research paper presents a comprehensive study aimed at improving the forecasting of demand for Air National Guard (ANG) enlisted initial and technical schooling. It is structured into several major sections, each contributing to the development, validation, and application of a novel forecasting model. Initially, the paper outlines the motivation behind the study, emphasizing the challenges faced by the ANG in accurately predicting training needs due to fluctuating service member exits and the consequential impact on operational readiness. This section sets the stage for the importance of developing a more precise forecasting model. The core of the paper introduces the Retention Prediction Model-ANG (RPM-ANG), a sophisticated statistical model designed to predict the likelihood of individual service members exiting the ANG. The model leverages historical data, incorporating various factors such as service duration, rank, and specialty, to estimate exit probabilities. This section includes a detailed description of the model’s methodology, the statistical techniques employed, and the rationale behind the selection of predictive variables. Following the model description, the paper discusses the integration of the RPM-ANG with a transition model that converts exit forecasts into specific demands for schooling slots across different specialties. This part elaborates on the simulation techniques used to model the flow of personnel through the ANG's career lifecycle, highlighting how the model accounts for the dynamic nature of training requirements. In the final sections, the paper presents a series of case studies and simulations to validate the forecasting model. It compares the model's predictions with actual training demands, demonstrating the model's accuracy and reliability. The conclusion emphasizes the potential of the RPM-ANG to significantly enhance the ANG's training planning and resource allocation processes, ultimately leading to improved readiness and efficiency.

Assessment

The GPT-4 EXSUM of this paper is very different from the original. Although it talks about each section of the paper, it includes fewer specific details. For example, the GPT-4 summary does not go into as much depth on the motivations for the study or on specific details of the model developed to predict ANG needs. Both general and specific EXSUMs can be appropriate depending on the audience, but certain details—in this case, including the overall improvement over the existing prediction structure—should be included. Additionally, the GPT-4 EXSUM refers to the model as the RPM-ANG, even though the paper itself does not. The issues with this EXSUM may be due to the more technical content of the paper, as it describes the creation of a model with techniques that may be difficult for GPT-4 to describe. It is also about twice as long as Paper 1, which makes it more difficult to generate a comprehensive EXSUM.

Paper 3: Quantifying and Visualizing Forecast Uncertainty with the FIFE

Link: https://www.ida.org/-/media/feature/publications/q/qu/quantifying-and-visualizing-forecast-uncertainty-with-the-fife/p-31857.ashx

Real EXSUM for Paper 3

Survival analysis can be a useful tool for modeling the attrition of service members, particularly when it comes to forecasting future states of survival for those members. Government sponsors are often interested in predicting these attrition rates at future time points. The Institute for Defense Analyses (IDA) has developed a tool for this purpose: The Finite Interval Forecasting Engine (FIFE) (Institute for Defense Analyses 2021). FIFE is a forecasting tool that produces predictions through the use of various modeling frameworks, including deep neural networks and gradient boosted trees. FIFE combines methods from both survival analysis and multivariate time series analysis to predict future states of survival, along with total counts of attrition, for service members at various future points in time.

We discuss methods for quantifying uncertainty in these survival forecasts, both for individual probabilities of exit, and aggregated total exits. While FIFE currently uses advanced approaches for maximizing forecasting performance, through the use of LightGBM for gradient-boosted trees, and Keras for neural networks, there are little-to-no implemented methods for measuring uncertainty in these predictions. Point estimates for future values of interest can be close to the truth, but they are never correct. In some classification prediction problems, future realized data that occurs far from the training data can be classified incorrectly with high probability. Use of prediction intervals around those point estimates leads to appropriate understanding of the accuracy of these predictions. Having a probability distribution associated with a future value, instead of just a point estimate, facilitates understanding of the uncertainty associated with forecasts.

We define relevant terminology in relation to prediction uncertainty and address how these terms differ based on field of study, and we discuss the properties of prediction intervals and describe the specifics of our objective in adding methods to FIFE that can produce these intervals. Our literature review investigates differing approaches to quantifying forecast uncertainty, which includes generic methods and learner-specific methods. We then implement a few of these approaches in FIFE and discuss their performance.

GPT4 EXSUM for Paper 3

The paper, developed at the Institute for Defense Analyses, introduces the Finite Interval Forecasting Engine (FIFE) - a tool designed for predicting future attrition rates of service members. It aims to enhance forecasting by integrating survival analysis and multivariate time series analysis, producing both individual and aggregated attrition forecasts. A significant contribution of this work is the development and implementation of methods to quantify uncertainty in these forecasts, acknowledging the inherent limitations of point estimates. The authors explore both generic and learner-specific methods for creating prediction intervals, notably Chernoff Bounds, MC Dropout for deep neural networks, and Stochastic Gradient Langevin Boosting for gradient boosted trees, among others. The paper emphasizes the importance of understanding and quantifying different types of uncertainties: approximation, model (epistemic), and data (aleatoric) uncertainties. Through extensive analysis, the paper demonstrates the application and effectiveness of these methods within FIFE, providing detailed discussions on parameter tuning for both simulated and real data. The discussion and future work sections highlight the ongoing challenges and potential directions for refining uncertainty quantification in forecast models, emphasizing the practical implications for decision-making processes based on forecasted attrition rates.

Assessment

This is the most technical of the three sample papers and contains more equations and mathematical terminology. Similar to Paper 2, the GPT-4-generated summary provides fewer details than the human-generated summary does on the use of different technical methods and how each contributed to developing prediction intervals for the model in question. While there is no omission of a key final result in this EXSUM as there was in the previous example and nothing inaccurate is stated, the GPT-4 generated summary provides a much less comprehensive view of the paper than the human one. This may be for the same reasons we suggested in the second example, as this is also a longer and more technical paper than Paper 1. That said, the GPT-4-generated EXSUM does summarize the content across the paper, even if it does so in a more general way than does the printed version.

Overall Performance

It is difficult to determine EXSUM quality for a given paper due to the possibility of specific parts that should be highlighted for a specific audience. However, across the three papers we examined, GPT-4 was able to at least generate a summary that captured the content of the major sections of the paper without any clear inaccuracies. The GPT-4 summary was most similar to the human summary for the first paper, which was the shortest and least technical of the three examined. However, for the two longer and more technical papers, the GPT-4 summary included fewer specific details and explanations. Although this may be appropriate for less technical audiences in some circumstances, the key findings of the papers were also less clear in these summaries.

Additional prompt engineering and tweaking may improve the quality of generated summaries—for example, the summaries generated for the more technical papers could be improved if users followed up with a prompt instructing GPT-4 to include more details from a certain section or to include a specific result. We adjusted our own prompt from the basic “Write an executive summary for the following paper,” and asking for more detail in a vague manner resulted in noticeably different EXSUMs. Additional prompt engineering and tweaking could improve the quality of generated summaries—simply adding the descriptor “detailed” results in an objectively better summary of a paper. That being said, we also experimented with adding phrases such as “contextualize technical information” and “write this for a senior executive,” but they did little to change the quality of the generated summary. Running the same prompt twice can result in two different summaries, so there is some variance in the consistency of LLM-generated summaries. It is possible that newer LLMs, such as GPT-4-32k, will not require splitting up short- and medium-length papers and then joining the summaries together, but that process will likely still be required for longer papers. Also, even if an LLM-generated EXSUM does not fully detail the key points the researcher would like it to focus on, it still has the potential to save time by creating an initial starting point for a draft. After all, many of the LLM-generated summaries are not too far off from the published human EXSUMs.

In general, though the LLM-generated summaries required edits, they provided starting points for EXSUMs that could be made more technical without too much effort. Alternately, a more specific prompt could be all that is needed for a result that better matches the author's intent. This may provide an opportunity to save time for RSMs writing EXSUMs and money for IDA at large. The cost for a ChatGPT Plus subscription is $20/month. In comparison, generating an EXSUM may take anywhere from 1–2 hours for shorter papers or 3–4 hours for longer ones. As the estimated average hourly rate for RSMs is ~$100, this could result in anywhere from $95 to $395 in savings.

Conclusion

GPT-4 provides opportunities for researchers to save time on both data analysis and writing EXSUMs, thus potentially reducing project costs. For data analysis work, researchers can use the GPT-4 feature to automatically clean data and develop simple visualizations, as well as write starter code for more complex tasks that require some human adjustments afterward but ultimately save time. In regard to EXSUMs, GPT-4 does seem to work better with shorter and less-technical papers, but it generates a summary that only requires some editing based on what the author thinks should be the focus, and that is certainly faster than starting from scratch.

Even if the first output produced by GPT-4 is not perfect, its ability to quickly answer prompts provides opportunities to self-correct, or a researcher can prompt further in a way that provides a satisfactory answer while still saving time. That being said, GPT-4 requires careful supervision. For data analysis, this means reviewing the decisions that it makes in its analysis and checking the code that it runs in its interface. For EXSUMs, that means checking that the content of the summary is completely accurate and that it captures the most important findings from the paper. Going forward, some of these issues may be resolved as new LLM models are released by OpenAI or other LLM developers. In particular, it may end up being best to use multiple LLMs that are specialized for certain tasks, such as an LLM designed for data analysis and an LLM designed for summarization.

References

Lu, Y. (2023, July 11). What to know about CHATGPT's new Code interpreter feature. *The New York Times*. https://www.nytimes.com/2023/07/11/technology/what-to-know-chatgpt-code-interpreter.html

Kumar, J. (2023, July 6). *New York City Airbnb 2023, public data*. Kaggle. https://www.kaggle.com/datasets/godofoutcasts/new-york-city-airbnb-2023-public-data?resource=download

OLDCC. (n.d.). *Defense spending by state - Fiscal year 2021*. https://oldcc.gov/dsbs-fy2021

Acquisition Research Program
Department of Defense Management
Naval Postgraduate School
555 Dyer Road, Ingersoll Hall
Monterey, CA 93943

www.acquisitionresearch.net