Summary Cycles: Exploring the Impact of Prompt Engineering on Large Language Models' Interaction with Interaction Log Information

Jeremy E. Block **Yu-Peng Chen** **Abhilash Budharapu**

Lisa Anthony **Bonnie Dorr**

{j.block, yupengchen, budharapu.ab}@ufl.edu, lanthony@cise.ufl.edu, bonniejdorr@ufl.edu
University of Florida

Abstract

With the aim of improving *work efficiency*, we examine how Large Language Models (LLMs) can better support the handoff of information by summarizing user interactions in collaborative intelligence analysis communication. We experiment with interaction logs, or a record of user interactions with a system. Inspired by chain-of-thought prompting, we describe a technique to avoid API token limits with recursive summarization requests. We then apply ChatGPT over multiple iterations to extract named entities, topics, and summaries, combined with interaction sequence sentences, to generate summaries of critical events and results of analysis sessions. We quantitatively evaluate the generated summaries against human-generated ones using common accuracy metrics (e.g., ROUGE-L, BLEU, BLEURT, and TER). We also report qualitative trends and the factuality of the output. We find that manipulating the audience feature or providing single-shot examples minimally influences the model's accuracy. While our methodology successfully summarizes interaction logs, the lack of significant results raises questions about prompt engineering and summarization effectiveness generally. We call on explainable artificial intelligence research to better understand how terms and their placement may change LLM outputs, striving for more consistent prompt engineering guidelines.

1 Introduction

Mark M. Lowenthal describes intelligence in three ways: (1) the process of preparing collected intelligence for (often) government consumers; (2) a product of such a process, e.g., a report, database, or "Intellipedia;" (3) the community of people and institutions involved in the preparation, and products, of the intelligence cycle (Lowenthal, 2018). While there is some debate about what is considered intelligence work (Andrew et al., 2019), this domain is characterized by multiple, nonlinear data processing steps in collaboration with multiple departments and people. Tools that could support the distribution of what is known and how the information was derived could be beneficial. Yet, it is challenging to prepare written communication about the precise event sequences that led to a particular outcome from users' memory alone. To address this, analytic provenance has emerged as a promising solution.

Provenance, in this context, refers to the documentation and representation of the process and context underlying an analysis, capturing the steps, data sources, algorithms, and decisions made by an analyst. The promise of provenance is to enable transparency and reproducibility, but listing all the steps leads to a verbose record that may not support these goals. Instead, we apply *analytic* techniques to illicit patterns automatically or visually represent application states over time (Ragan et al., 2015; Xu et al., 2020). When applied to the field of intelligence, often that means capturing *interaction logs* (i.e., recorded steps taken by a user to complete their task) to distill key aspects, facilitating a more comprehensive understanding of the problem-solving process.

The goal of analytic provenance research is therefor focused on illuminating the reasoning behind steps taken and how conclusions are reached. Often, techniques can make steps clear or visualize how often data is examined (Block et al., 2023), but understanding why a step was taken is often more difficult to elucidate from system processes. This is where analytic provenance research seeks to push boundaries, providing more semantically meaningful explanations by looking for patterns among the series of interactions. By incorporating analytic provenance, researchers can effectively communicate the methodology employed, supporting peer review, knowledge exchange, and collaboration.

Resources such as *Papers with Code*, *GitHub*, and the *Open Science Framework* emphasize the

Proceedings of the 4th Workshop on Evaluation and Comparison of NLP Systems, pages 85–99

open-source nature of research and the need to centralize provenance information. However, we have not seen evidence of efficiently processing interaction log information to provide textual summaries with the goal of enabling transparency. By considering interaction logs to describe the steps taken to complete a task, LLMs are uniquely suited to examine patterns in this language and might serve as a general-purpose analysis tool in the analytic provenance toolkit.

This study aims to gain a better understanding of how large language models (LLMs) can expand the possibilities of interaction log information, focusing on a specific set of prompt engineering features. We observe that the LLMs can extract features from an interaction history. We further evaluate the impacts of different prompting effects on the output, engineering prompts to vary the addition of examples and audience description for the LLM. By manipulating these prompts, we aim to investigate how they impact the output generated by the model when presented with interaction log information.

This research seeks to shed light on the intricate relationship between large language models and interaction log data. By examining the effects of prompt engineering features on the model's response, we can gain insights into how to effectively leverage these models for enhancing analytic provenance and, ultimately, the efficient communication of problem-solving in complex domains. The findings from this study will contribute to advancing the field of NLP and inform the development of more sophisticated tools for capturing, summarizing, and leveraging interaction log data in analytic provenance research. Our contributions include the following:

1. A method of recursive prompt reduction with the same LLM.
2. A demonstration of our method on the relevant intelligence and analytic provenance domain.
3. A quantitative analysis of accuracy and factuality among output summaries.
4. A qualitative comparison of output summaries and prompt engineering guidelines.
5. A commentary on the ethical use of large language models for workplace cohesion tasks.

Based on the research contributions completed, we believe that our work will benefit the intelligence field by:

- demonstrating that large language models can be applied to the context of provenance information as a tool for describing how people create intelligence products,
- reporting on the factuality and accuracy of the products to serve as a baseline for future work,
- discussing some concerns about the use of large language models for the production of work reports.

2 Related Work

The NLP field has seen public attention this year from the widespread adoption and use of generative pre-trained models (Zhao et al., 2023). In this work, we explore how LLMs can support analytic provenance research, especially when paired with prompt engineering approaches.

2.1 NLP for Analysing Interaction Logs

Interaction logs come in many forms and can be analyzed in different ways to extract insights. Marin-Castro and Tello-Leal (2021) consider user interaction logs to better understand organizational processes, Hamooni et al. (2016), generate insights from internet-connected devices, and Guo, Yuan, and Wu (2021) identify anomalous activity among network system log messages with a pre-trained encoder model like BERT. In all of these contexts, analytic provenance techniques are applied to make sense of interaction logs and deliver insights in the form of interrelated and hierarchical system diagrams or notifications. This is helpful, especially when examining logs across large organizations or among corpora of captured event messages from heterogeneous sources. But at a smaller day-to-day scale, there are communications among team members and managers that communicate work completed that could use support from analytic provenance techniques.

However, common business communications are not typically communicated with graphs or charts. To match familiar styles and minimize a need for visualization literacy, there is a need to present insights as text. Liu et al. (2021) generate summarizations from code snippets to make code easier to interpret and maintain, but they rely heavily on graphs as a transition language to map from lines of code to text. Similarly, converting interaction histories into a textual summary is its own challenge. In our case, we explore a technique to automatically

combine contextual information with interaction information to distill a comprehensive textual summary of a user's analysis session.

2.2 Prompt Engineering

Prompt engineering (Beltagy et al., 2022) has emerged as a viable technique for improving the performance of summarization models. By providing explicit instructions to the model, prompt engineering can help facilitate the generation of more accurate summaries.

Firstly, there are few-shot methods (Tsimpoukelli et al., 2021) that recommend providing a task-specific example to improve the accuracy of the expected result. This approach leverages a large pre-trained language model and fine-tunes it on a small example case for effective summarization. For example, Liu et al. (2022) extend this concept by providing unstructured information instead of a single example. Regardless, they show how providing contextual information can support large language reasoning tasks.

Alternatively, Reynolds and McDonell (2021) show how the lack of task-specific examples can also be effective. Several studies have explored the zero-shot paradigm (Ye et al., 2023; Wei et al., 2022), where models are trained to generate summaries without any specific fine-tuning on summarization datasets. Often these approaches rely on prefix-tuning (Zhou et al., 2023) or perturbing the training data with noise (Lewis et al., 2019). regardless, these approaches have shown promise, especially working with generalized pre-trained models (Reynolds and McDonell, 2021)

Finally, Chain-of-thought methods have also gained attention, where an LLM is given a list of steps to complete in addition to the specified content (Wei et al., 2023). Zhang et al. (2023) propose a method to generate summaries by explicitly describing the chain of steps to the model and providing a rationale. This encourages the model to reason more about the prompt and provide more accurate replies. Overall, prompt engineering techniques, including zero-shot, few-shot, and chain-of-thought methods, have shown promise in enhancing summarization performance by providing explicit guidance and controlling the generation process. These approaches influence the methodology presented in this paper.

3 Experimental Procedure

To better understand the expectations and effects of using large language models for summarization of interaction logs, we conduct a handful of experiments, starting with the collection of user feedback from a qualitative study. From this pilot study, we then conduct a series of NLP prompting experiments to compare differences in how the addition of examples and audience types influence model output summaries. Throughout these experiments, we use the OpenAI Chat Completions API with the "gpt-3.5-turbo[1]" model as the LLM for our approach (Brown et al., 2020).

3.1 Pilot Study

Many summarization approaches score summaries based on their coherence, fluency, informativeness, and relevancy (Wu et al., 2020), yet no applicable framework existed for summarizing intelligence work for hand-off communication. We conduct a user study with the primary objective of better understanding which features are preferred by human users in work summaries for different types of audiences. While the details of this study are beyond the scope of this paper, we provide an overview of the methodology used to derive our prompting features. We create an online questionnaire to gather insights from anonymous participants and identify the qualities of summarization that human evaluators find beneficial for peer collaborators and team managers. The study was approved by our institutional review board and aims to understand user preferences for work summaries.

To help participants understand the context, they are asked to review LLM-generated work summaries and rank them according to their communicative support for peer collaborators or team managers. The summaries vary in their generated content and lengths, and participants are asked to quote specific features and textually describe how they are valuable and invaluable. Finally, we also ask participants to classify a set of adjectives (e.g., *accuracy*, *conciseness*, *clarity*, etc.) as core components or non-essential adjectives used to describe peer or manager summaries.

Twenty graduate students pass the attention checks and complete the questionnaire, but due to limited statistical power and the fact that no summary was consistently ranked higher than any other,

[1]Available at https://platform.openai.com/docs/model-index-for-researchers/

we focus on the adjective classifications to draw our conclusions. The study results indicate that most participants believe our eight adjectives are core components of good summaries. However, a small preference exists for certain words and contexts. The results suggest that participants consider *objectivity, relevance, conciseness, and clarity* slightly more essential for a manager's summary but not for their peers. Instead, participants prefer that summaries for peers be *engaging and accurate*. Both *relevance and properly cited* score the same by conditions. Qualitatively, participants highlight how summaries should strike a balance between providing enough detail without being too vague or overly detailed and tailoring the level of information to the user's needs. The findings have guided us in adapting our prompt engineering experiment to identify key features and terms for effective prompting.

3.2 Dataset

We use a set of interaction logs[2] from users completing a 90-minute textual sensemaking task. Originally captured from 24 university students (non-analysis experts), it consists of thousands of user interaction events (e.g., mouseover, click, search, etc.) as they review 103 fictional bank transactions, email intercepts, and other facsimile intelligence reports from the VAST Challenge dataset (Mohseni et al., 2018). To conduct our analysis, we experiment with the interaction logs of the first three users solving the VAST 2010 mini-challenge 1. The size of the chosen context is intentionally not large. We conduct our work on data at a reasonable size for human comprehension to better evaluate and act as a demonstration of our pipeline. This limited size makes it possible for one author to manually write gold-standard summarizations of user analysis processes.

3.3 Documents to Context Sources (A)

Before engaging with the interaction logs for context, we need a fairly complete source of reliable contextual information for each of the documents users could interact with in the original analysis database. However, including document content for each interaction would be excessive. Entity extraction has been shown to detect factual inconsistencies (Lee et al., 2022). Also, the inclusion of knowledge before prompting for a specific answer

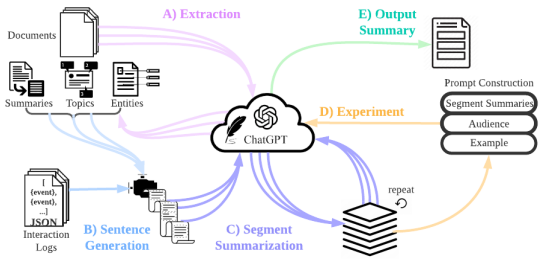

Figure 1: A depiction of our proposed pipeline for making interaction logs into work summaries. We preprocess the document space to A) extract information and B) generate interaction sentences by combining this information with interactions. The generated sentences are C) segmented and summarized to prepare our D) experiments. Finally, we examine the E) output summary.

can also improve model performance at reasoning tasks (Liu et al., 2022). Therefore, we prompt ChatGPT to infer topics, identify entities, and summarize each document in the underlying document dataset as a pre-processing phase (Figure 1A). This allows us to include additional context when an interaction occurs on a document and supports shorter prompt lengths because we can provide document topics instead of an entire document as context. When prompting for this contextual information, we provide precise instructions in terms of output lengths and formatting preferences (i.e., 100 words; JSON format). For a comprehensive overview of the full prompts, please refer to Table 2 in the Appendix or our open-source code.[3]

3.4 Interaction Logs to Sentences (B)

Although ChatGPT is able to handle structured data formats like the ones used for interaction logs (e.g., JSON), directly including the raw interaction logs in an API request will significantly increase the number of tokens. Therefore, we use a sentence-templating approach to preprocess the interaction logs into sentences. Each logged interaction is systematically transformed into a sentence by applying a manually designed template for each interaction type. For example, a search interaction would be converted into the sentence: "The user searched for <*term*>," where '<*term*>' would be substituted with the relevant information from the interaction. We apply this process for all 11 interaction types in

[2]Available for download from https://www.cise.ufl.edu/~eragan/provenance-datasets.html

[3]A version of our approach can be found at https://github.com/jeremy-block/spygest

the dataset[4] to mimic the naive conversion of interactions into sentences. Although this approach creates many similar-sounding sentences, it maintains the original interaction sequence and generates a comprehensive corpus of sentences that preserves the context of user interactions. This process (Figure 1B) allows for subsequent segmentation and prompting processes as described next.

3.5 Segmentation and Token Management (C)

At the time of writing, the OpenAI Chat Completions API has a token limit of 4096.[5] In our use case, a significant challenge arises as the entire interaction session comprises hundreds of interactions, resulting in an average length of 13,788.33 tokens, excluding tokens needed for prompts and responses. To help reduce the number of tokens sent to the API, we draw inspiration from the step-by-step zero-shot chain-of-thought prompting technique (Wang et al., 2023). Our recursive approach (depicted as Figure 1C) involves requesting summaries for smaller segments of the interaction sentences and linking the input of each request with the response from previous requests. By doing so, we not only prompt ChatGPT with "let's think step by step" but also establish distinct steps for the agent to follow.

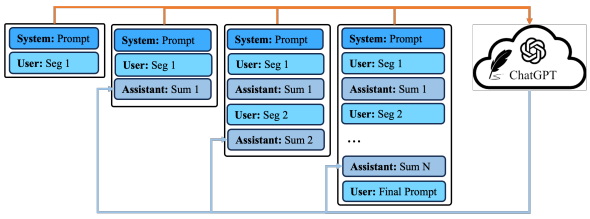

Figure 2: An illustration of our prompting process.

The text corpus describing the entire interaction session is divided into ten segments, determined through a trial-and-error process where test runs are conducted to ensure that the number of tokens remains within the specified limit. The API takes messages as input, where each message is assigned a specific role (i.e., system, user, or assistant). As shown in Figure 2, the entire prompting process is conducted as a conversation that follows a format beginning with a *system* message, followed by a sequence of alternating *user* and *assistant* messages. A total of 11 requests are sent to the API for each interaction log summary, including one request for each segment and a final request for an overall summary. To address the model's "memoryless" nature, all messages are added to a growing list, serving as memory for ChatGPT, with the entire list consistently sent in each request.

3.6 Prompt Design (D)

Prior work has shown that an effective prompt should include clear and specific instructions (Wei et al., 2023; Liu et al., 2022). Our prompting process follows this principle consistently. We use delimiters (e.g., triple backticks) to indicate distinct parts of the input. To construct our prompts, we provide the content in three different message types. The *system* message is the first and explicitly instructs ChatGPT about the task to be executed and the expected behavior. We include the core features from the pilot study here to help the model define its persona. Next, we use alternating user and assistant messages to provide additional context and our final prompt.

As shown in Figure 2, *user* messages either include the segment to be summarized or the final prompt. On the other hand, *assistant* messages are used as a pseudo-memory, only containing summarized segment text returned from earlier API requests. In the *final user* message, detailed persona-specific instructions are included to explore the potential of tailoring the agent's response to specific user needs and expectations. It is here that we specify the different types of audiences and the inclusion of different examples.

3.7 Ground Truth Development (E)

To evaluate the measures described above, we leverage a set of reliable summaries as the gold standard. Often, summarization accuracy is based on human-generated ground truth corpora against which generated summaries are compared (Dernoncourt et al., 2018). Therefore, we create three types of ground truth summaries for each of the three interaction log sessions to use in the evaluation.

First, a set of summaries were crafted by one author for the three interaction log sessions, referred to as the **manual** summary. This was prepared by carefully reviewing each interaction log, paying attention to the think-aloud events, and writing about the major events from the sessions. Additionally, a

[4]We do not use any "think aloud" interaction types because these were manually added by the dataset creators to augment the data and provide some semantic ground truth within the captured logs. Verbal utterances like this are not commonly captured in standard interaction logs, so we choose to exclude them.

[5]https://platform.openai.com/docs/models/gpt-3-5

baseline summary is generated with ChatGPT by following our recursive prompting procedure (Figure 2). In the later prompt engineering experiments, we include example summaries and different adjectives for audience types, but these generated summaries show what ChatGPT does when recursively asked to summarize interaction logs as a baseline. By happenstance, when testing, we noticed that by repeating the pseudo-memory with the final prompt, the resulting summary was consistently shorter. Because automated accuracy measures are sensitive to summary length (Koh et al., 2022; Papineni et al., 2002; Sellam et al., 2020; Snover et al., 2005) we include the summaries with **additional** pseudo-memory context for our evaluation.

By incorporating these three types of ground truth summaries, we can compare how recursively asking large language models to generate work summaries compares to manually written reports from interaction logs.

4 Results

Our goal with this work is to demonstrate the simplicity of a recursive summarization technique for communicating user interaction logs. Overall, the generated summaries are promising and may offer a realistic possibility for generating sufficient support for report generation with human refinement. In this section, we offer a handful of observations.

4.1 Quantifiable Objective Metrics

Our work examines the impact of various prompt designs on the two quantifiable measures of interest (i.e., our dependent variables), namely factuality and accuracy. In our experiment, we manipulate two independent variables: the **target audiences** and the **prompt engineering strategies**, each of which has four different levels. The target audiences are characterized by the core features identified in our pilot study. The four levels include no audience (none), self, peer, and manager. The prompt engineering strategies are manipulated by how examples were provided to the LLM. The four levels include no examples (Zero-Shot), providing a manual summary (One-Shot), providing a masked manual summary (One-Shot + Hint), and providing a masked template (Hint). We examine interaction logs from three participants, resulting in the analysis of 48 summaries (i.e., 3 (participants) x 4 (types of audiences) x 4 (types of provided examples)).

Factuality A known challenge with abstractive summarization is the chance of the model generating inaccurate information (i.e., hallucinations (Ji et al., 2023; Gabriel et al., 2021)). For this reason, we evaluate the factuality of the base summaries. Some techniques try to calculate factuality automatically but are either not trained on our specific use case (Ribeiro et al., 2022) or struggle to decompose summaries into reliable chunks for comparison (Glover et al., 2022). Instead, we use the FRANK framework defined by Pagnoni et al. (2021) to manually determine the percent of factual phrases in our generated summaries.

Using the same entity definitions presented in the FRANK framework, the three baseline summaries (i.e., the None x None condition) for each of the three participants are coded. Semantic Frame Errors occur when predicates, entity mentions, or circumstance details are inaccurate. Discourse Errors describe when pronouns or entailments are incorrect. Content Verifiability Errors describe when the content is essentially hallucinated or dramatically inconsistent. Finally, we choose to also count the frequency of repeated phrasing as an additional error type. One author manually applies this code to individual phrases of a summary and counts the occurrence of different types of errors. These error counts are then divided by the total number of phrases in a summary to calculate the factuality percentage.

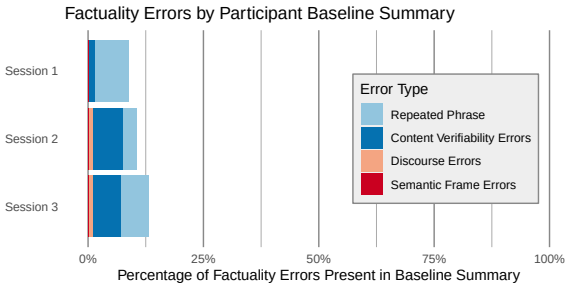

Figure 3: a representation of the relative percentage of different error types for the baseline summaries for each of the three participant interaction logs

In Figure 3, we see very few factual errors among the three participants examined. As (Pagnoni et al., 2021) discuss, transformer models have been shown to have fewer semantic frame errors than LSTM (Hochreiter and Schmidhuber, 1997) models, but, as we see with our results, there are still discourse errors. We also observe more repetition of sequences of words. This may be due to the fundamental functionality of transformer mod-

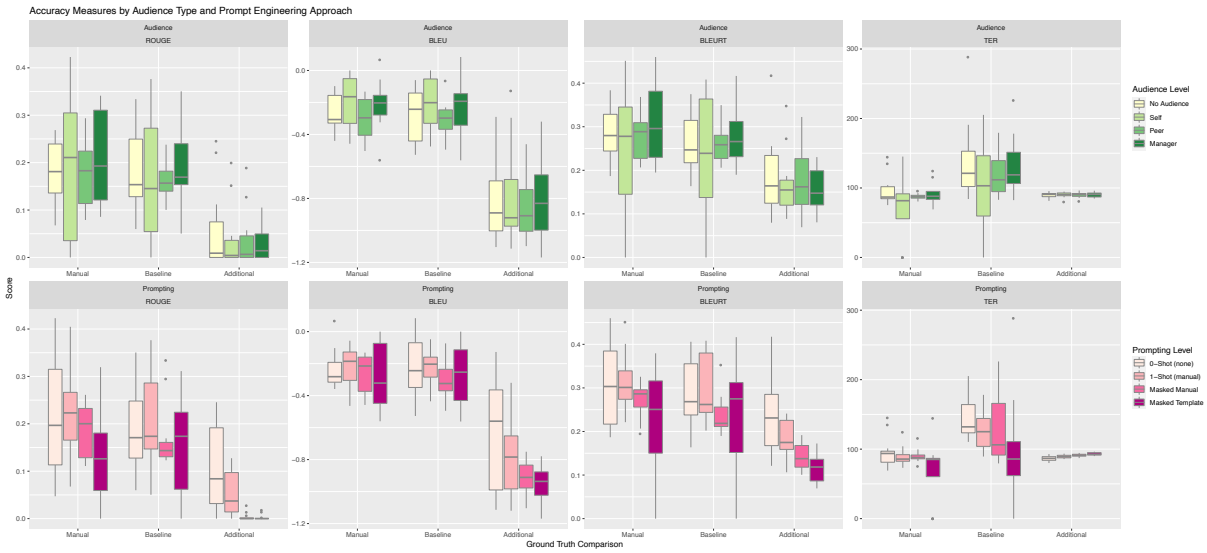

Figure 4: We show the distribution of each automated measure (i.e., ROUGE-L, BLEU, BLEURT, and TER) compared across the three ground-truth summaries: One manually generated by a human, one generated by the baseline model without any prompt engineering, and one generated in the same way but with the prompt provided twice. Notice that these scores show very little variation among each group showing that the independent variables of the Audience type or Prompt Engineering approach have little influence on the accuracy of the measure.

els (Vaswani et al., 2017), where each word is generated with a certain probability. Given this context, some words like "arms dealing, fraud, and illegal possession of arms, as well as events related to sickness, health issues, and business success" may be repeated by the model because it frequently saw them appear together or were defined in the initial system message. Regardless, we see high factuality scores across the baseline summaries for each participant, leading us to consider other dependent features.

Accuracy to ground truths Determining the accuracy of a summary can be a challenge, and various factors must be considered, such as cohesion, readability, conciseness, information-richness, precision, quality of input text and summarization algorithm used, length of the summary and human evaluation (Gupta and Gupta, 2019). Instead, we apply an ensemble of summary accuracy measures to help determine a general sense of accuracy. The set of accuracy criteria selected requires the generated summaries to be compared to some ground truth. As described in Section 3.7, we designed three types of ground truth. One summary is written by an author (i.e., manual), our LLM pipeline generates another (i.e., baseline), and another generated version where the pseudo-memory is repeated in the final prompt (i.e., additional). Koh et al. (2022), suggest that Rouge-L aligns with human expectations, but BLEU (Papineni et al., 2002) and BLEURT (Sellam et al., 2020) are also popular for abstractive text summarization evaluations. TER can also describe accuracy, by converting one string of text to another and counting the number of changes (e.g., insertions, deletions, etc.) (Snover et al., 2005). We choose a handful of techniques to get a general sense of the accuracy of our various ground truth summaries (i.e., Manual, Baseline, and Additional) given different audience types and prompt engineering strategies.

In Figure 4, we see a variety of ranges for each accuracy measure (i.e., each facet of the chat). Looking at the Audience levels (i.e., green hues) and Prompt Engineering (i.e., magenta hues), we see little variation among these levels too. Alternatively, we see more differentiation based on the ground truth summary comparison (i.e., the horizontal grouping), signaling that the summary we used to compare the accuracy may have more influence on the score than either of our experimental factors (i.e., Audience and Prompt Engineering).

4.2 Qualitative Observations

The evaluation of the system's performance reveals several notable qualities. Firstly, providing context and requesting summarization recursively proves to be a viable technique for this context. LLMs, like ChatGPT, identify key phrases and reinforce them in their summary. The system incorporates entities and topics from the dataset into the gen-

erated summaries, showcasing its proficiency in identifying relevant concepts.

However, certain aspects remain ambiguous and raise intriguing points for discussion. One notable aspect is the pipeline's goal-oriented focus on generating final summaries. The phrasing used in the summaries strongly implies that all the information provided is intricately connected to the given goal. Consequently, every detail recorded in the interaction log is considered relevant to the process of solving the puzzle at hand. This behavior is likely a direct reflection of the task outlined in our prompt. In the initial system message to ChatGPT, we explicitly mention that the interaction logs depict someone "trying to investigate an event in the intelligence domain."

It is from this perspective that the model operates, and as a result, the generated summary naturally strives to establish connections between all available information (i.e., provenance sentences) and the specified goal (i.e., summarize the steps taken). The absence of unrelated or misleading information in the underlying dataset further reinforces the challenge of disambiguating between intentional deductions and serendipitous insights. Within the dataset, there are few instances of red herrings or other relevant fallacies designed to divert the analyst's attention heavily. Consequently, when reading the generated summaries, it is not easy to distinguish between insights that the model intentionally identified as relevant behaviors toward the goal and those that were stumbled upon serendipitously.

Another intriguing observation is the system's tendency to adopt phrasing from prompt engineering examples, even if it struggles to calculate the described pattern accurately. Looking at the output of summaries where an example is provided shows that 9/48 summaries include percentages of topics covered. In the 0 Shot (i.e., Baseline) summaries, the inclusion of percentages was never generated by default and only appears after seeing the structure demonstrated in one of the masked prompts. This suggests that the system draws inspiration from provided examples and incorporates their phrasing into the output, potentially refining the final structure.

Still, despite the seeming agency to control the output's phrasing, the percentages and values are incorrect. Even when the percentages provided by the manually generated example are accurate, the returned output generates its own (incorrect) value for these phrases. Since transformer models are optimized to predict the next word in a phrase, the system appears to rely on identifying relevant terms and phrases from the corpus rather than more preferred behaviors, like performing deeper statistical analysis or ranking different behaviors as more relevant than others.

Incorporating a statistical determination layer into the preprocessing pipeline could enhance the ability to identify patterns beyond linear descriptions. On the other hand, while there are common evaluation measures for evaluating summarization, we are unaware of benchmarks that evaluate the ability of language models to group and consolidate information by examining the relative semantic meaning of concepts. Optimization in this direction may improve LLMs in the analytic provenance context and likely many more.

5 Discussions and Future Work

In this work, we explore the factors of audience and example inclusion as a demonstration of applying prompt engineering to generate work summaries in the intelligence domain. While we have not found other evidence of a methodology where the proposed pipeline consults a large language model, the pre-processing steps taken on the dataset documents are inspired by the chain-of-thought prompting strategies. We use a series of prompts to extract information from documents and segment an interaction log to build up a complete summary prompt and discuss the results.

Our independent variables are derived from our pilot study, where users identify essential elements of a work summary. Yet, we do not see strong effects on baseline summary factuality or accuracy when adjusting the audience or the inclusion of examples. Instead, in our testing, we observe different important factors. We observe differences in summary lengths when we included contextual information twice. Therefore we use two different kinds of ground truth (i.e., baseline and additional) to account for this. This leads us to think about how specific wording in the prompt messages may noticeably impact the focus of the output.

Novel methods may emerge that afford the direct manipulation of prompt wording. For example, it would be interesting to investigate how opposite terms, antonymic to the adjectives used in our study, may impact the model's attention. Additionally, ab-

lation studies that target the specific adjectives we use may offer fascinating insights into which terms make the biggest difference. Regardless of the technique employed, studies exploring the influence of individual terms do not, to our knowledge, have consistent summarization evaluation criteria, thus calling attention to a need for more established evaluation methods.

Finally, corresponding to the chain-of-thought nature of the work presented, there are obvious future directions that could consider how the prompting process could involve human users to adjust and modify the prompt in real time. It would be helpful to have domain experts rank the summaries and use these rankings to fine-tune the prompting process. Additionally, giving users interface controls that manipulate the generated prompt by using prompt engineering guidelines could be imagined for future exploration into model behaviors. It is also interesting to consider the downstream tasks from a work summary and how different generation methods are perceived and may influence future work by human users.

Ultimately, in this work, we observe the feasibility of generating human-sounding summaries of work from user interaction logs, but they tend to list steps completed without a hierarchical structure that captures the concepts that are most important or structures the content to flow like a story. Perhaps future work could explore how additional analysis layers, prompt engineering interfaces, or human feedback may help summaries acquire a sense of structured storytelling.

6 Conclusion

By harnessing LLMs, researchers can enhance transparency, reproducibility, and collaboration, improving problem-solving communication. In this work, we showcase the potential of ChatGPT to generate work summaries from data analysis interaction logs and the associated document contexts. By manipulating prompt engineering features, we investigate the impact of different prompts on the LLM's output in the intelligence domain. We develop a recursive prompt reduction method to handle token limitations and evaluated prompt examples and audience types, both quantitatively and qualitatively. While we show the potential LLMs have for automating work summaries from provenance information, we find few consistent impacts of these factors on summary accuracy. Instead, we recognize that more reliable prompt engineering guidelines will be helpful when developing more sophisticated tools to analyze provenance information and control generated output.

As has become a common discussion within the research community (Ray, 2023; Maslej et al., 2023), the need to better understand these models and their impact on society is critical. While what we demonstrate shows promise for productivity increases, there are tradeoffs that come from automation that will impact how we individually engage with society. Therefore, we complete this work with a discussion of the various limitations of what we proposed and the ethical considerations of LLM usage in workplace cohesion tasks.

Limitations

We conduct our testing on a single dataset and among three users' interaction histories to examine if large language models can be used to make work summaries. The 103 textual documents included in the VAST dataset are small enough that we can conduct and test our summarization pipeline. Since the data context is at a human-comprehensible scale, we can ask for summaries, entity extraction, and topic modeling while also writing gold-standard summaries and verifying the content.

The results of the demonstrated technique are promising, but additional complications are likely to be introduced when applied to larger scales of data. For example, challenges exist where the underlying document dataset is restricted due to privacy concerns (e.g., healthcare records or government intelligence) or its temporal dynamism (e.g., social media posts or stock market movements). Capturing static, secure snapshots of the data an analyst is working with to conduct our approach will require additional consideration by the research community.

Also, while the data context we demonstrate contains some typos and misspellings of names, it would be beneficial to explore how this approach applies in multilingual contexts. Often intelligence work deals with content in foreign languages, and applying an approach that introduces machine translation or additional lingual morphologies, will support the promise of our proposed technique.

Ethics Statement on Broader Impact

The emergence of LLMs shows promise for enhancing bureaucratic activities and enhancing efficiency.

As AI technologies advance, we are witnessing significant shifts in how individuals refer to and discuss the concepts of artificial intelligence. However, the use of LLMs to automate processes that involve generating human-like text raises important ethical considerations pertaining to human work and the creation of knowledge. LLMs will fundamentally change how people work, necessitating new skills in editing and engineering results. There are unexplored possibilities for extending LLMs' impact on workplace activities and beyond. The effort to achieve explainability in LLMs is challenging, but the ambition to identify weaknesses, biases, and boundaries is encouraging (Agarwal et al., 2022).

Unfortunately, this work does little to mitigate the potential drawbacks of large language models, but we hope to demonstrate a methodology for elucidating underlying system behaviors for system designers who can then improve the models. The data we used in our demonstration was collected for research purposes with individuals' informed consent that their interactions would be interpreted in the future (Mohseni et al., 2018). In our work, we have demonstrated how LLMs can serve as an essential lynchpin for novel applications and evaluation methodologies.

In a broader way, concerns still exist regarding the detection and propagation of harmful and inaccurate information by generative models. Our experiments demonstrate the model's hyperfixation on the terms provided in the system prompt, which leads to assumptions about the goal of the interaction log's content and purpose. Behaviors like this compromise the accuracy of reports and ultimately could dissolve user trust.

Apart from improving model accuracy, emphasizing AI literacy is crucial to recognizing technology faults and differences. While it is delusional to assume that the public will ever deeply understand the workings of AI tools, the effort by designers to encode best practices into tools and ensure societally-aligned responsible usage is a necessary first step. We call attention to these ethical considerations and promote the responsible use of LLMs in generating summaries of individual work.

References

Chirag Agarwal, Eshika Saxena, Satyapriya Krishna, Martin Pawelczyk, Nari Johnson, Isha Puri, Marinka Zitnik, and Himabindu Lakkaraju. 2022. Openxai: Towards a transparent evaluation of model explanations. *arXiv preprint arXiv:2206.11104*.

Christopher M. Andrew, Richard J. Aldrich, and Wesley K. Wark, editors. 2019. *Secret intelligence: a reader*, second edition. Routledge.

Iz Beltagy, Arman Cohan, Robert Logan IV, Sewon Min, and Sameer Singh. 2022. Zero- and few-shot NLP with pretrained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 32–37, Dublin, Ireland. Association for Computational Linguistics.

Jeremy E. Block, Shaghayegh Esmaeili, Eric D. Ragan, John R. Goodall, and G. David Richardson. 2023. The influence of visual provenance representations on strategies in a collaborative hand-off data analysis scenario. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):1113–1123.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Franck Dernoncourt, Mohammad Ghassemi, and Walter Chang. 2018. A repository of corpora for summarization. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Saadia Gabriel, Asli Celikyilmaz, Rahul Jha, Yejin Choi, and Jianfeng Gao. 2021. Go figure: A meta evaluation of factuality in summarization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, page 478–487. Association for Computational Linguistics.

John Glover, Federico Fancellu, Vasudevan Jagannathan, Matthew R. Gormley, and Thomas Schaaf. 2022. Revisiting text decomposition methods for nli-based factuality scoring of summaries. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, page 97–105. Association for Computational Linguistics.

Haixuan Guo, Shuhan Yuan, and Xintao Wu. 2021. Logbert: Log anomaly detection via bert. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Som Gupta and S. K Gupta. 2019. Abstractive summarization: An overview of the state of the art. *Expert Systems with Applications*, 121:49–65.

Hossein Hamooni, Biplob Debnath, Jianwu Xu, Hui Zhang, Guofei Jiang, and Abdullah Mueen. 2016. Logmine: Fast pattern recognition for log analytics. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, page 1573–1582, New York, NY, USA. Association for Computing Machinery.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):248:1–248:38.

Huan Yee Koh, Jiaxin Ju, He Zhang, Ming Liu, and Shirui Pan. 2022. How far are we from robust long abstractive summarization? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, page 2682–2698. Association for Computational Linguistics.

Hwanhee Lee, Cheoneum Park, Seunghyun Yoon, Trung Bui, Franck Dernoncourt, Juae Kim, and Kyomin Jung. 2022. Factual error correction for abstractive summaries using entity retrieval. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, page 439–444. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022. Generated knowledge prompting for commonsense reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3154–3169, Dublin, Ireland. Association for Computational Linguistics.

Shangqing Liu, Yu Chen, Xiaofei Xie, Jingkai Siow, and Yang Liu. 2021. Retrieval-augmented generation for code summarization via hybrid gnn.

Mark M. Lowenthal. 2018. *The future of intelligence*. Polity.

Heidy M. Marin-Castro and Edgar Tello-Leal. 2021. Event log preprocessing for process mining: A review. *Applied Sciences*, 11(22).

Nestor Maslej, Loredana Fattorini, Erik Brynjolfsson, John Etchemendy, Katrina Ligett, Terah Lyons, James Manyika, Helen Ngo, Juan Carlos Niebles, Vanessa Parli, and et al. 2023. *The AI Index 2023 Annual Report*. AI Index Steering Committee.

Sina Mohseni, Andrew Pachuilo, Ehsanul Haque Nirjhar, Rhema Linder, Alyssa M. Pena, and Eric D. Ragan. 2018. Analytic provenance datasets: A data repository of human analysis activity and interaction logs. *CoRR*, abs/1801.05076.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Eric D Ragan, Alex Endert, Jibonananda Sanyal, and Jian Chen. 2015. Characterizing provenance in visualization and data analysis: an organizational framework of provenance types and purposes. *IEEE transactions on visualization and computer graphics*, 22(1):31–40.

Partha Pratim Ray. 2023. Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems*.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI EA '21, New York, NY, USA. Association for Computing Machinery.

Leonardo F. R. Ribeiro, Mengwen Liu, Iryna Gurevych, Markus Dreyer, and Mohit Bansal. 2022. FactGraph: Evaluating factuality in summarization with semantic graph representations. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3238–3253, Seattle, United States. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. Bleurt: Learning robust metrics for text generation.

Mathew Snover, Bonnie Dorr, Richard Schwartz, John Makhoul, Linnea Micciulla, and Ralph Weischedel. 2005. A study of translation error rate with targeted human annotation. Technical report, Technical Report LAMP-TR-126, CS-TR-4755, UMIACS-TR-2005-58, University of ....

Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill. 2021. Multimodal few-shot learning with frozen language

models. In *Advances in Neural Information Processing Systems*, volume 34, pages 200–212. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Hanlu Wu, Tengfei Ma, Lingfei Wu, Tariro Manyumwa, and Shouling Ji. 2020. Unsupervised reference-free summary quality evaluation via contrastive learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3612–3621, Online. Association for Computational Linguistics.

Kai Xu, Alvitta Ottley, Conny Walchshofer, Marc Streit, Remco Chang, and John Wenskovitch. 2020. Survey on the analysis of user interactions and visualization provenance. *Computer Graphics Forum*, 39(3):757–783.

Seonghyeon Ye, Doyoung Kim, Joel Jang, Joongbo Shin, and Minjoon Seo. 2023. Guess the instruction! flipped learning makes language models stronger zero-shot learners. In *The Eleventh International Conference on Learning Representations*.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. 2023. Multimodal chain-of-thought reasoning in language models.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large language models are human-level prompt engineers.

A Document Context Creation

Here we provide the specific preprocessing prompts sent to ChatGPT to get the Topics, Entities, and summary for each document in the dataset. We specify the length of topics, types of entities, and the number of words to generate short document contexts that include essential information.

Topics prompt: `Act as an intelligence analyst, your task is to determine topics that are being discussed in classified documents. Determine up to 5 topics in the document delimited by triple backticks. Make each item one to 2 words long. Format your response as "a list of items separated by commas". Document: <`*content*`>`

Entities prompt: `Act as an intelligence analyst, your task is to identify named entities in classified documents. There are 4 entities, which are "person, organization, location, and miscellaneous" from CoNLL-2003. Identify the entities in the document delimited by triple backticks. Format your response in a JSON format. Document: <`*content*`>`

Summary prompt: `Act as an intelligence analyst, your task is to generate a short summary of classified documents. Summarize the document delimited by triple backticks in at most 100 words. Document:` ```` ```<`*content*`>``` ````

B Segment Summarization

segmentation: because there is a token limitation for a single request, we ask the LLM to summarize the previous interactions and use this shorter interaction history as its memory. This process is similar to the use case of a chatbot in which an LLM summarizes previous conversation and uses that summary as its memory, instead of using the entire raw conversation log as the memory

System Prompt: ``` ``Act as an intelligence analyst, your task is to generate a summary of the interaction logs of a user who was trying to investigate an event in the intelligence domain. The logs are written in sentences. The entire interaction is divided into 10 segments. You will be summarizing the entire interaction session step by step by summarizing one segment at a time. When you are summarizing a segment, make

sure you take into account summaries of previous segments. Please summarize a segment in at most 100 words. The goal is to communicate findings and progress in a collaborative investigation scenario. Please focus on these core features delimited by triple backticks when you summarize:```*<terms for Audience level. See Appendix Table 1>*```"

User Prompt for each segment: "Summarize the sentences describing the interactions of segment 1 delimited by triple backticks in at most 100 words. Make sure you take into account summaries of previous segments. Description: ```*<segment N from interaction sentences generated in preprocessing stage>*```"

C Independent Variables

Based on the findings of the pilot study, we examined how an audience may influence summarization techniques. Similarly, we wanted to examine how various prompt engineering approaches like zero-shot and few-shot may impact summaries in our chain-of-thought-inspired approach.

C.1 Audience

We direct the ChatGPT prompt with the terms (see Appendix Table 1) derived from the pilot study. These terms appear in the process of generated segments (see Appendix B) and the final prompt construction (see Appendix Table 2).

Table 1: **Pilot Study Core Features/Terms** As identified by the user study described in Section 3.1, we explicitly list the terms suggested as core features for summarization. In the pilot study, a discussion about summaries for an individual is not included, so we combined all the terms for this case.

Audience Level	Suggested Terms
None	N/A
Self	objectivity, relevance, conciseness, clarity, engaging, accuracy, proper citation, coherence.
Peer Collaboration	engaging, accuracy.
Team Manager	objectivity, relevance, conciseness, clarity.

C.2 Examples

We also systematically vary the inclusion of an example in the **Final User message**. Below are examples of the content sent to ChatGPT for the first interaction log. These examples would be customized for each user session.

None: N/A; like a zero-shot approach.

Manual example: "Please provide the overall summary based on the example delimited by triple backticks. Example: ```This session began by searching for the word "Nigeria" and looking at the documents returned. They noted that Dr. George and Mikhail emailed and then transitioned to searches about "Kenya" and the Middle East. At this time, they were reviewing people like Leonid Minsky and Anna Nicole Smith. By the end of the session, they had transitioned to exploring documents from Russia and middle eastern countries. They searched for "death," "kasem" and "dubai." In the end, they returned to some of the same documents they had opened at the beginning but also opened many different documents for the first time. Out of the 46 topics and 102 documents, they reviewed 39 topics, opened 45% of the total documents at least once, and spent an average of 30 seconds with each document. The people they returned to most frequently were Leonid Minsky, Mikhail Dombrovski, and Dr. George.```"

Masked manual example: "Please provide the overall summary based on the example delimited by triple backticks. Example: ```This session began by searching for [KEYWORD1] and looking at the documents returned. They noted that [KEYWORD2] and [KEYWORD3] emailed and then transitioned to searches about [KEYWORD4] and [KEYWORD5]. At this time, they were reviewing people like [KEYWORD6] and [KEYWORD7]. By the end of the session, they had transitioned to exploring documents from [KEYWORD8] and [KEYWORD9]. They searched for [[KEYWORD10], [[KEYWORD11] and [KEYWORD12]. In the end, they returned to some of the same documents they had opened at the beginning but also

opened many different documents for the first time. Out of the [NUMBER] topics and [NUMBER] documents, they reviewed [NUMBER] topics, opened [NUMBER]% of the total documents at least once, and spent an average of [NUMBER]"```

Masked template: "Please provide the overall summary using the template delimited by triple backticks. Example: ```They focused on [NUMBER] main topics in this analysis session, exploring [PERCENTAGE] of the documents. The topics that received the most attention were [TOPICS]. They started searching for [KEYWORD1], before transitioning to [KEYWORD2] and finally looking for [KEYWORD3]. They conducted NUMBER searches throughout their session. [CONCLUSION]```"

D Ground Truth Descriptions

We used three different ground truths as an evaluation standard and tweaked the process based on two different independent variables. The first is the **Manual** summary seen in Appendix C.2. This is custom for each user's session and contains accurate and factual information written by one author.

The **Baseline** summary was generated by ChatGPT without any additional prompting. This means there were no specifications about an audience or example provided.

The **Additional** summary was also generated by ChatGPT but simply had the segment messages repeated in the final prompt. By repeating the user and system messages in the final prompt, we noticed the summary was shorter, which could influence accuracy calculations.

Table 2: **Final Prompt Construction** The final prompt to ChatGPT is generated from the variations shown in this table. Each accuracy experiment designates some vertical combination of the following strings of text, choosing one audience level and one example level (4 x 4). This final prompt combines with all the prepended messages that contain the initial system message as well as the pairs of user and assistant segmentation summaries.

Please provide a comprehensive summary of the entire interaction based on the summaries of *user.numSegments* segments in at most *finalLength*.				
Audience	None	Self	Peer	Manager
	N/A	Please avoid being too vague and overly detailed.	Your audience will be a peer who is more comfortable working with team members' uncertainty and hedged statements. More specifically, you should follow a list of instructions delimited by triple backticks. Instructions: 1. Provide the context of the analysis by offering starting points and providing more details later. 2. Being entirely objective is less important for peer collaboration than being accurate or relevant to their peers. 3. Including the opinions of the author in their summary can provide contextual data (e.g., hedge statements or other personal theories) about the state of the investigation. 4. Please avoid being too vague and overly detailed.	Your audience will be a manager who expects to see summaries with a high information density in each sentence and still provide context for the investigation without offering too many details to invite the manager to do the task themselves. More specifically, you should follow a list of instructions delimited by triple backticks. Instructions: 1. Should not focus on the specific statistics but focus on the general behaviors. 2. Please provide a sense of how much work was completed. 3. Please use more descriptive language. 4. Please avoid being too vague and overly detailed.
Example	None	Manual	Masked	Template
	N/A	*Human-Generated Ground Truth*	*Human-Generated Ground Truth* but nouns replaced with masks (e.g., [number], [topic], [percentage], etc.)	*Generic summary template for any summary. All values are masked.*