

Challenge Submission: Overview Paper

Navy AI Utility for Training, Instruction, and Comprehensive Analytic Learning (NAUTICAL)



Instructional System Design (ISD) Analysis Using Al Large Language Models

13 November 2023

Prepared by

Aptima, Inc. 12 Gill Street, Suite 1400 Woburn, MA 01801

Copyright 2023, Aptima, Inc.; Limited Rights in accordance with DFARS 252.227-7013. Distribution authorized to US Government Personnel only





TABLE OF CONTENTS

List of Figures 1
List of Tables
Executive Summary
Problem Statement
Solution
Solution Component: Robust Data Model
Solution Component: Browser Based User Interfaces
Solution Component: Expert Prompt Engineering
Solution Component: Enhanced Excel Documents
Benefits of NAUTICAL System
Summary of Challenge Results
NAUTICAL System Development and Architecture
NAUTICAL User Workflow
Step 1: System, Project, and JDTA Setup 16
Step 2: Product Item Setup and Generation
Step 3: Batch Generation and Data Export
Deep Dive: Retrieval Augmented Generation (RAG) Integration
Deep Dive: Chain-of-Thought
Challenge Questions and Answers
Model Selection
Question: What AI Large Language Models were evaluated for use and selected? Why? 31
LLM Function
Question: How does the LLM Function?
Solution Requirements and Implementation
Question: What are the software and hardware dependencies if the solution were to be
implemented?
Question: What is the prerequisite information/data needed to run if the solution were to be
implemented?
Question: What training of instructional systems designers/analysts is needed if the solution
were to be implemented?
Test and Evaluation
Question: What methods were used to test and validate results?
Barriers
Question: What Barriers might there be for implementation of the proposed solution by the
Navy or end users?
Overall Assessment
Question: What is your overall assessment of challenges and opportunities for
Implementing AI LLMs for this application?
List of Acronyms, Abbreviations, and Symbols
APPENDIX A: APTIMA AI Prompt Engineering Requirements and Guidelines 42



List of Figures

Figure 1. Components of the NAUTICAL Solution	4
Figure 2. Example Jinja-based Prompt Template	7
Figure 3. Example of a Prompt Template after NAUTICAL has Filled in the Variable Data	8
Figure 4. Example System Prompt	9
Figure 5. Example Use of Python Code for Data Manipulation in a Prompt Template	. 10
Figure 6. Development of the Data Model and Human/AI Workflow	. 14
Figure 7. NAUTICAL System Architecture	. 15
Figure 8. Process Workflow for NAUTICAL Users	. 16
Figure 9. Welcome Screen of the NAUTICAL Application	. 17
Figure 10. Project Setup Screen	. 17
Figure 11. Task Setup and Entry Screen	. 18
Figure 12. Product Item Definition Screen	. 19
Figure 13. Prompt Template Definition Screen	. 20
Figure 14. Data Source Setup Screen	. 21
Figure 15. RAG Screen	. 22
Figure 16. Product Item Response Screen	. 23
Figure 17. The Process Flow from a Prompt Template, to a Filled Prompt, to an	
Answer/Rationale from the LLM	. 24
Figure 18. Batch Generation Screen	. 25
Figure 19. Excel Product Export Screen	. 26
Figure 20. Document Product Export Screen	. 26
Figure 21: Raw Retrieved Text based on RAG Prompt	. 28
Figure 22: Refined Retrieved Text based on RAG Prompt	. 28
Figure 23: Prompt Template with RAG Setup	. 29
Figure 24. Chain-of-Thought Prompting within a Single Prompt	. 29
Figure 25. Chain-of-Thought Product Item used as an Intermediary Step in Complex Prompts.	. 30
Figure 26. Sample Prompts Training Document in Development	. 34



List of Tables

Table 1. Summary of Items Produced for Round 1 of the RRL Challenge	. 13	j
Table 2. Top Choice LLM Selection Criteria	. 31	
Table 3. NAUTICAL Self-Assessment	. 35	,



Executive Summary

NAUTICAL (Navy AI Utility for Training, Instruction, and Comprehensive Analytic Learning) delivers transformative efficiency to the Navy's instructional systems design and training evaluation workflows. Developed by Aptima, NAUTICAL seamlessly integrates mature AI capabilities with established methodologies through user-centered design optimized for adoption.

At its core, NAUTICAL is powered by innovative prompt engineering strategies that harness the language proficiency of large language models like OpenAI's ChatGPT. The system decomposes high-level products like task analysis models into discrete, structured data fields. Precision crafted prompts are constructed using these fields to elicit structured and bounded responses from AI. Advanced prompting techniques such as Retrieval Augmented Generation and Chain-of-Thought reasoning enable complex, context-aware results.

NAUTICAL's interfaces empower users to intuitively construct and execute prompts, review AIgenerated outputs, and iterate until achieving required quality thresholds. Batch processes automate product creation, while parallel execution ensures responsive deliveries. Seamless integration of rationale within downstream artifacts provides transparency into AI logic.

Architected on a microservices foundation, NAUTICAL delivers production-grade scalability and reliability while enabling interchangeability of components like LLMs as technology evolves. Open-source building blocks and cloud-native design ensure cost-effective deployment and portability across on-premise, multi-cloud, or hybrid environments.

Extensive user-centered testing with representative instructional designers shaped NAUTICAL's ability to complement existing skills while expanding analytical potential. The system demonstrates the art of possible when thoughtfully leveraging AI to augment human expertise and unlock operational advances.

In summary, NAUTICAL pioneers an innovative prompting methodology that steers advanced natural language models to efficiently generate high-fidelity training artifacts. By meshing AI seamlessly within established processes through deliberative engineering and design, NAUTICAL provides a roadmap for responsibly transitioning emerging capabilities from promise to practice. The solution delivers a strategic advantage today while charting a course for the Navy's continued leadership in leveraging AI.

Problem Statement

The imperative for rapid development and enhancement of training within the United States Armed Forces has never been more critical. The Navy's Ready Relevant Learning (RRL) initiative has been tasked with crafting a modernized training architecture that not only meets but exceeds today's demands. The past 18 months have seen a significant increase in the capabilities of large language models, allowing for meaningful applications across many domains and tasks. The application of these models creates an opportunity for machines to generate products more quickly, and with the same level of efficacy, as human operators; Particularly if humans are in the loop, working with the AI technologies. Harnessing the power of Artificial Intelligence (AI), such as advanced large language models like ChatGPT, can greatly expedite the generation of ISD analysis products, including Task Analysis, Learning Analysis, and Media Selection.



Appropriate use of these models can significantly cut down on the time and costs traditionally associated with these processes while simultaneously elevating the quality of the outcomes.

Solution

Our solution, NAUTICAL (Navy AI Utility for Training, Instruction, and Comprehensive Analytic Learning), was developed in response to the Navy's challenge of accelerating the "Analysis" ISD phase, which is crucial for the efficient flow into the Design, Development, Implementation, and Evaluation stages in the ADDIE ISD model.

NAUTICAL is a transformative solution designed to revolutionize the Navy's instructional design and training evaluation. At the core of NAUTICAL (Figure 1) is a suite of web browserbased user interfaces, underpinned by a robust data model that facilitates an intuitive and seamless user experience. Expertly crafted prompt engineering strategies are employed to leverage the sophisticated capabilities of AI, ensuring the generation of precise, contextually relevant content. This innovative system enhances the Navy's existing Excel workbooks, infusing them with advanced functionalities to better meet the rigorous demands of the Ready Relevant Learning (RRL) initiative. NAUTICAL is engineered to optimize the Instructional Systems Design (ISD) Analysis phase, propelling the Navy towards unprecedented efficiency in developing high-quality training programs. This comprehensive solution is not just a tool—it's a strategic advantage, designed to deliver actionable insights and accelerate "Speed to the Fleet," ensuring that the Navy's training systems are as dynamic and forward-thinking as the personnel they are intended to serve.

Our focus is beyond simply filling in the ISD items for the challenge, but rather aims to develop an augmented workflow for ISDs that utilizes AI technology as a collaborative tool for efficiently developing training solutions.



Browser Based User Interfaces

Interfaces that support ISD/Analysts in their workflow:

- Enable the representation of different product items
- Prompt development templates
 Executing prompts (at individual
- and batch level)Visualizing results of generations
- Configuring data sources
- · Enabling RAG workflow



Expert Prompt Engineering

Prompt Engineering Approach offers many benefits:

- Scalable, adaptable application
- Flexibility across use cases
- More accurate data analysis
 - Reduced work time resulting in cost savings



Robust Data Model

Robust data model that can abstractly represent the

- items needed to do all steps of ISD Analysis:Interfaces are a direct representation of the data
- representation of the data model

47.96	122	BTOR-De	19.95	10.00	49/08/054	BLORDON	10.00
3.11	14.00	14,44	14,95	3,55	107	14.0	9.0
Analyse of All		Sail La	Annual Sector	Petrate	UN Delater	10.00104.0	Lass Transa Long
111	lan.	Der Norderst, Sonerferne eine Ste Sall Ge- Tete Seinerin Jam Tete Gelan fer Mehrenzeiter einer Stellen	And Son Test	The factory of the fa	The Assessment Second S	Particular Ch. National Ch.	Number (number) (1995) Cares Programmer Fransport
ofer bet		processory in the second state of the second s	Antonia.	14.24	INCOME NAME OF A	the section of the	section reprint a second of
1.1.1	٠	California Talla	_	-	_	_	_
thu.	hi.	LOTION PROGRAMMING SHEEPING			-	-	
14418	-	Second participants and an	_	Sec.	Tradhetical record of	sports and leaf-year scales	to be the reader spect
10.444	E.	helpfortering conserving and inclusion		292	STATE Produce destine	and adding publicity ratio is	a animal or cased one disease
10111	<u>e</u> -	internet on a sector of the sector		H	Renders, or relation of the approximited may attract	pertention from (ALE, also terms politicus let noise climpi	a patent tak with mond. a taken word to sportstants dat
	1	Topic conversion of		au las	*		
16,415	14	Name and the second sec					
99-117	-	Per sector a spolarent					
146.118	-	Leansanced courses and the sale					
	24	Property design and residenced		(M)			

Enhanced EXCEL Workbooks

Enhanced Excel Workbooks to display rationales for every AI generated field:

 Ability to view rationales for every data item provides clarity of responses and higher quality output

Figure 1. Components of the NAUTICAL Solution



Solution Component: Robust Data Model

At the heart of NAUTICAL is a robust and extensible data model that supports ISDs in their workflows for the development of analytic work products, such as the Task Analysis Model, Learning Analysis Model, Media Selection Model, Instructional Performance Requirements Document (IPRD), and Instructional Media Requirements Document (IMRD). Key to this data model is the abstraction of individual pieces of data that need to be created into what we call *Product Items* (e.g., an individual cell in a Task Analysis Model) which collectively comprise the *Products* (e.g., the entire Task Analysis Model spreadsheet). For example, the Task Analysis Model *Product*, one *Product Item* is entitled "Skills." As part of the data model, each *Product Item* is defined by a set of fields, including:

- 1. A UID (unique identifier),
- 2. The name of the item,
- 3. The associated *Product*,
- 4. Instructions on how the item is developed,
- 5. Constraints on the item,
- 6. The Definition of the item,
- 7. Examples of created items,
- 8. Criteria for developing the item,
- 9. Relevant Context for the item (e.g., "military operations"),
- 10. Relevant Scales associated with the item (e.g., T/F, High/Medium/Low),
- 11. Options for the item response (e.g., paragraph format, single-item response, multi-item response), and,
- 12. Dependencies on other *Product Items*.

The final field, Dependencies, is of particular importance as it defines how different pieces of data feed into each other. For example, in the Learning Analysis Model, the *Product Item*, Voice Audio, depends on a previous *Product Item*, Audio Description. That is, the determination of whether the tasks require the use of voice audio relies on a previously developed description of the audio features required for the task.

Each *Product Item* will have one or more *Prompt Templates* associated with it. The *Prompt Template* provides a way to structure the information related to the *Product Item* into a text prompt that can be sent to the LLM. Within the *Prompt Template* users can identify which LLM to utilize for the item (e.g., ChatGPT 4 vs. LLAMA2), what data sources may be necessary to include (e.g., instructional manuals or look-up tables), an overall strategy for generating the answer, a system prompt, and the structured prompt itself. The system prompt is useful in LLMs to provide contextual information for the language model, such as the format for the response.

Solution Component: Browser Based User Interfaces

To build a comprehensive solution capable of addressing the Navy's RRL Challenge, and more generally the need for AI-driven instructional design, Aptima has developed a suite of user interfaces that enhance the ISD workflow and its underlying data model, offering an intuitive user experience. The NAUTICAL application utilizes a series of web-based interfaces that enable users to:

• Create and edit ISD projects, work products, and data flows.



- Define Product Items and their associated data.
- Develop Prompt Templates.
- Import external data sources.
- Develop Retrieval Augmented Generation (RAG) and Chain-of-Thought (CoT) workflows.
- Iteratively query the LLMs, revise prompts, and review answers from the LLMs.
- Generate entire Products at once through batch processing and visualize the results.
- Export Products into different formats, including Excel workbooks and Word documents.

These interfaces are tailored to represent an array of product items, enabling a seamless transition from concept to execution. NAUTICAL provides intuitive templates for both system and product item prompt development, coupled with validation features that ensure the accuracy and efficiency of our template strategies. The execution of prompts can be performed individually or in batches, offering both precision and scalability. Visualizing the outcomes of data generation enable users to quickly identify missing or incorrect data. Our toolset extends to the curation of data sources, streamlining the process of definition and population. Aptima is at the forefront of integrating RAG workflows (discussed in later sections), ensuring that every generated piece of data is enhanced with the most relevant and contextually rich information. This comprehensive interface ensemble empowers users to achieve new levels of productivity and data-driven decision-making, setting the stage for the Navy to lead with information superiority.

Solution Component: Expert Prompt Engineering

The NAUTICAL application enables the development of a wide variety of prompts, from simple question and answers or fill-in-the-blanks to complex multi-step CoT and RAG approaches. Using templates, we can define reusable prompt strategies that can be used across multiple Product Items, increasing the efficiency of the ISD workflow.

Within a Prompt Template, Aptima has defined four (4) overall strategies:

- 1. Generate: Utilize LLMs to produce responses for the Product Item.
- 2. Copy: Combine previously generated responses into a single Product Item.
- 3. Manual: Allow users to enter responses manually for a Product Item.
- 4. Python: Enable users to directly write and execute Python code for advanced data manipulations.

In addition, NAUTICAL utilizes a templating language called Jinja

(<u>https://palletsprojects.com/p/jinja/</u>) that enables users to integrate plain-language prompts with more complex variables and dependencies. Figure 2 provides an example Prompt Template for a Product Item.





Figure 2. Example Jinja-based Prompt Template

Each of the double-bracketed ("{{}}") items represents a reference to another piece of data in the data model. This could include information about the specific task (e.g., {{ task.description }}), information from the Product Item fields (e.g., {{ product_item.definition }}), or information from other dependent Product Items (e.g., {{ responses["LA_41_00"].answer }}). When the prompts are sent to the LLMs, these pieces of information are filled in by the NAUTICAL system. The above template is populated as shown below in Figure 3. This enables the development of extremely generalizable, interdependent, and reusable prompts.





Figure 3. Example of a Prompt Template after NAUTICAL has Filled in the Variable Data

NAUTICAL also utilize system prompts with the LLMs to provide additional context, formatting information, or other relevant instructions to the LLM. These system prompts can also utilize Jinja and dependencies in the data model to create complex responses. An example of a system prompt used regularly for this effort is provided in Figure 4.



•	Respond as factually as possible based on the information provided.
•	You may make inferences if they are solidly grounded, but do not make anything up.
•	Don't give information not mentioned in the CONTEXT INFORMATION.
•	Do not include any direct reference to the task description in the answer.
•	Keep the answer to the specified format. Do not add any introductory words or sentences.

Figure 4. Example System Prompt

The versatility of this prompting strategy enables us to create increasingly complex and interdependent prompts within NAUTICAL. Two advanced techniques utilized in this effort are RAG and CoT prompting. These are discussed in detail in subsequent sections of this document, but both enable arbitrarily complex patterns of reasoning and inclusion of external data sources to be implemented within NAUTICAL.

As a way of prompt bounding, our overall prompting strategy utilizes requests to the LLM to provide both an answer and a rationale for that answer. Aptima has found, through extensive empirical testing, that this strategy increases the accuracy and interpretability of the LLM responses immensely. By asking the LLM the explain its reasoning when providing an answer, we find that it prods the LLM to more precisely consider how its answers are generated. In addition, it provides the ISD a clear explanation of the answer and increases their confidence in results, especially in cases where they may disagree with the answer, or the answers provided are not immediately intuitive.

Finally, we note that while LLMs can provide accurate, intuitive, and reasoned results in many cases, they are not the appropriate solution in every scenario. In cases where a product item relies less on natural language reasoning and more on data manipulation, we have integrated the ability to write and execute Python code through the NAUTICAL prompt templates. This enables advanced users to manipulate data in the underlying model in complex ways. An example is shown in Figure 5.



```
import re
table=data sources["11 Media Selection Rubric"].content
raw sensory requirements = responses["MS 09 00"].answer
sensory requirements = [x.strip().rstrip('.') for x in
re.split(';|\n|,',raw sensory requirements.strip()) if x]
if responses["LA 89 00"].answer.strip():
 sensory_requirements.append('Complex Controls')
if responses["LA 90 00"].answer.strip():
 sensory_requirements.append('Complex Scenarios')
if responses["LA 91 00"].answer.strip():
 sensory requirements.append('Out-of-Tolerance Feedback')
if responses["LA 92 00"].answer.strip():
 sensory requirements.append('Fault Insertion')
lines = table.strip().split('\n')
headers = lines[0]
rows = lines[2:]
def clean row(row):
     cleaned = row.lstrip().lstrip('|').lstrip().rstrip().rstrip()
    return [x.strip() for x in cleaned.split('|')]
h = clean row(headers)[1:]
lookup = \{\}
for row in rows:
   r = clean row(row)
   name = r[0]
   r = r[1:]
    result = {}
    for i in range(len(h)):
       result[h[i]] = 1 if r[i] == '1' else 0
    lookup[name] = result
missing srs = []
counts = \{x:0 \text{ for } x \text{ in } h\}
for sr in sensory requirements:
    if sr not in lookup:
       missing srs.append(sr)
        continue
    media values = lookup[sr]
    for media, count in media values.items():
       counts[media]+=count
results = []
for k,v in counts.items():
   results.append((k,v))
results.sort(key = lambda x: x[1], reverse=True)
answer='\n'.join([f"{k} = {v} matches" for k,v in results])
if missing srs:
   rationale='error for sensory requirements:\n\n' + '\n'.join(missing srs)
else:
    rationale=''
```



Solution Component: Enhanced Excel Documents

In our approach to tackle the RRL Challenge, Aptima also enhanced the existing Navy Excel workbooks by integrating them with the rationale provided by the LLM responses. This fusion allows for ISDs and other training developers to work with known product formats (such as the Excel Training Analysis Model) but still enjoy the benefits of interacting with the LLMs responses and reasoning. Starting within the NAUTICAL web application, users can export products into intermediary Excel files that contain both the answers to each product item as well as the rationale provided by the LLM for those answers. The enhanced Excel workbooks provide



input capabilities to enable users to quickly import the answers and rationale into the spreadsheets. This rationale is then presented within the workbook via a custom macro: when a user selects a data cell, a comment box dynamically appears, revealing the reasoning behind the data. This unique feature provides reviewers with invaluable insights and recorded thought processes from the system that were previously unrecorded. Our solution not only enriches the data with context but also fosters an informed review process, elevating the level of detail and understanding in data-driven decisions for the Navy's operational excellence.

Benefits of NAUTICAL System

While we anticipate that many challenge participants will utilize LLMs and prompt engineering in their responses, we believe that there are several unique benefits to the NAUTICAL system and the development of Aptima's solution.

- 1. Savings in ISD/Analyst time and reduced cost. The NAUTICAL system enhances current ISD workflows through intuitive interfaces and templates. This enables ISDs to focus on strategies for answering questions and deriving results, while automating many of their repeatable tasks. Once the system is set up, prompt templates and product items are created, and the data sources identified, ISDs can batch process and review results quickly. In the first round of the challenge, NAUTICAL was able to batch generate responses to all five documents (TA, LA, MS, IPRD, IMRD) that are comprised of over 200 unique product items across all ten challenge tasks in less than two hours. For an ISD tasked with producing products for hundreds or even thousands of unique tasks across multiple JDTAs, this can reduce their processing time from months to days.
- 2. Increased likelihood of adoption and reduced training requirements for ISD/Analysts. Our solution is meticulously designed to dovetail with the established workflows of ISDs and analysts, aiming to enhance adoption rates while minimizing the learning curve. Throughout the development phase, the Aptima team placed a strong emphasis on usability, conducting rigorous testing with representative users to refine the interfaces. This iterative process, rooted in actual user experiences and feedback, has been instrumental in shaping a system that feels intuitive and familiar to its users. By anticipating the needs of ISDs and analysts, the Aptima team has crafted a suite of interfaces that not only align with current user practices but also elevates the entire workflow experience. The result is a user-centric platform that simplifies the transition to advanced AI support, ensuring that the focus remains on the critical analytical work, not on navigating the toolset. Our proactive approach to usability testing underscores our commitment to delivering a solution that is as effective in practice as it is innovative in concept.
- 3. Enhanced support of analysts. In our endeavor to augment the capabilities of ISDs, the Aptima team recognizes the meticulous efforts that go into the current process, where the primary focus is on efficiency and data integrity. To complement this process, without imposing additional tasks on the analysts, we have introduced an innovative enhancement



to the existing workbooks. Our solution leverages AI to articulate the rationale behind each data entry, encapsulating the 'why' in a concise, accessible format. These AIgenerated insights are seamlessly integrated as hover-over popup comments within each field of the workbook, offering on-demand clarity without disrupting the workflow. This subtle yet powerful feature serves not just as a tool for quality control and deeper understanding but also as a bridge fostering trust between the analyst and AI, subtly enriching the current methodology with a layer of transparency and context that supports the ISDs' valuable work.

- 4. Extensible to future models and other applications. A standout feature of NAUTICAL is its inherent extensibility, accommodating the integration of various large language models (LLMs) and other applications with minimal effort. It is built with the flexibility to interchange data models as needed, supporting both external and internal hosting options. This flexibility means that as new and more advanced LLMs emerge, NAUTICAL can incorporate these tools seamlessly, without the need for a system overhaul or re-engineering of the underlying methodology. Furthermore, the system's design allows for individual product items to be powered by the most suitable model available, optimizing for accuracy and relevance in data generation. Such versatility ensures that NAUTICAL remains at the cutting edge, ready to harness future advancements in AI and machine learning to continually enhance the ISD process. This strategic foresight positions NAUTICAL as a sustainable, long-term solution for the Navy's instructional design needs, ensuring it remains a dynamic and adaptable asset in the ever-changing technological landscape.
- 5. Better quality results. Our tailored prompt engineering methodology is at the forefront of delivering superior quality results. By focusing on structured data fields rather than relying solely on direct language input, we establish a more granular level of control and precision in prompt creation. This data-driven strategy allows for the iterative refinement of prompts, ensuring that each iteration is informed by the last, thereby progressively enhancing the accuracy and relevance of the output. Integrating RAG and bespoke Python code into our system further elevates the quality of results. RAG brings contextually rich and informed content into the equation, pulling from a vast corpus of data to support the AI's responses. Meanwhile, custom Python code enables sophisticated data processing capabilities, from complex calculations to the manipulation of large datasets, which traditional prompt methods cannot handle. This combination of structured data fields, RAG, and Python scripting transcends the capabilities of standard prompt engineering. It not only refines the input for the AI but also tailors the AI's thought process, guiding it towards outputs that are deeply informed and meticulously aligned with the task at hand. Our approach ensures that the quality of the results is not left to chance but is instead a product of deliberate design and strategic engineering, setting a new benchmark for excellence in AI-assisted analysis.



Summary of Challenge Results

Our submission includes the five work products (Task Analysis Model, Learning Analysis Model, Media Selection Model, IPRD, and IMRD), which we believe demonstrate a robust and compelling solution to this challenging problem. In total, the NAUTICAL system developed 216 product items across the five products. A summary of each product and response strategies is shown in Table 1. We note that for the later products, namely the IMRD and IPRD, the Aptima team did not attempt many of the items due to both the time constraints of the challenge and the fact that many of the items in those documents fall outside the scope of the information included in this effort. Based on our initial efforts, though, we are confident that NAUTICAL can successfully be applied to all steps of the ISD process given appropriate access to data resources.

Product / Strategy	LLM Generated Responses	Manual Responses	Copied (from another item) Responses	Python Responses	Total
TA	29	1	4	0	34
LA	87	0	12	2	101
MS	28	0	20	1	49
IPRD	8	0	5	0	13
IMRD	4	0	15	0	19
Total	156	1	56	3	216

 Table 1. Summary of Items Produced for Round 1 of the RRL Challenge

NAUTICAL System Development and Architecture

Our solution to the RRL challenge relies on an extensive analysis of the problem and the tailoring of a robust software system to meet those needs. The diagram in Figure 6 depicts the approach for developing the NAUTICAL solution through a structured process of reviewing and categorizing the challenge data and ISD workflows, developing a data model and system to support those analyses, iterative development of prompts and other data strategies, and batch generation and export of analysis products.





Figure 6. Development of the Data Model and Human/AI Workflow

The development of NAUTICAL started with a robust analysis of each of the challenge data products, including the model workbooks, task list, and other background data documents. These informed the development of the data model and decomposition of each model/document feature into a unique product item. The Aptima team then categorized each product item, established dependencies between product items, and identified strategies for producing each item (e.g., generate from an LLM, copy from another item, manually entry by ISD/SME, and/or derived from Python code). Once the product items and their corresponding prompt templates were entered into the software system, the work of iterative prompt development began. In each case, strategies for the prompts were developed (examples of these strategies and sample prompts are included as **Appendix A**) and refined through interactions with the LLMs. In addition, when external data sources or complex queries were needed, the team developed RAG and chain of through strategies for each item. Finally, the results for each product item (and the corresponding rationale produced by the LLM when applicable) were reviewed before performing a batch execution, export from the software, and import into the final model workbooks/Word documents.

In terms of system architecture and software development, The NAUTICAL system (as depicted in Figure 7) is built on top of Aptima's Enterprise Analytics Services Platform (EASP) Discourse, a microservice architecture that supports the development of generative AI-based applications. EASP Discourse includes a library of LLM services that share a common API,



backend infrastructure for data storage, prompt engineering and experimentation UIs, LLM finetuning workflows, and Retrieval Augmentation Generation (RAG) support.



Figure 7. NAUTICAL System Architecture

The Discourse microservice architecture is built on <u>Yet Another Analytic Dataflow Architecture</u> (YAADA), an open-source data and analytics framework developed at Aptima. It uses OpenSearch to store document-oriented data such as the Prompt repository, the Data Source repository (used for RAG), and the Product Response database that keeps a history of all generations. Discourse also maps some data into specialized databases such as Weaviate (a vector database for text embeddings) and Neo4j (a graph database) to support query semantics such has vector similarity or graph traversal that aren't supported by the primary document database. By running LLMs as microservices with common APIs, it is possible to swap in and out LLM implementations as the state of the art evolves. Additionally, LLM services can represent self-deployed models running in AWS or can act as pass throughs to cloud-hosted inference services such as OpenAI, Anthropic Claude, or Hugging Face Inference API.



NAUTICAL User Workflow

This section provides a guide to using the NAUTICAL system, an introduction to the application interfaces, and additional details on the prompt engineering strategies developed and employed as part of our solution. The process workflow for users of NAUTICAL is summarized in Figure .



Figure 8. Process Workflow for NAUTICAL Users

In order to realize the generation approach and Human/AI workflow mentioned above in the short time required by the RRL Challenge, we developed a prototype frontend tailored to the ISD domain using Gradio, a popular open-source Python toolkit for building user interfaces around machine learning models. While Gradio is fairly limiting in the UI features it supports and not sufficient for deployment of a mature application for wide user adoption, it is extremely fast and adaptable for rapid prototyping, and allows the same developers doing ML/analytic development to implement UI with no frontend engineering expertise required. The Gradio interface supports the ISD workflow through a series of tabs that support iterative prompt engineering, response testing, batch generation, and product visualization and export.

Step 1: System, Project, and JDTA Setup

Users of NAUTICAL are welcomed to the application with an initial screen depicting the NAUTICAL logo as shown in Figure 9. Incidentally, the NAUTICAL logo was developed by AI through the integration of ChatGPT 4 and DALL-E 3 image generation capabilities. Future implementations of NAUTICAL will include authorization and authentication services to provide an additional layer of security for the system.





Figure 9. Welcome Screen of the NAUTICAL Application

After loading the system, the user can then define the parameters of the project that they are working on (Figure 10). This includes providing a name for the project (e.g., for the RRL challenge, we might call the project "CONSTRUCTION ELECTRICIAN ISD"). The user can also set global parameters for the project such as a global system prompt that will be used as default for newly created prompt templates. This is useful for consolidating system prompt strategies into an easily reusable format. Future iterations of NAUTICAL may include the ability to set up project access permissions, collaboration tools, definition of products, and other high-level project information.



Figure 10. Project Setup Screen

The next step in NAUTICAL involves entering the JDTA information for the project (Figure 11). Individual tasks can be entered through the interface for each job (e.g., Construction



Electrician) and duty (e.g., electrical and electronic system components). Future iterations of NAUTICAL will also include the ability to batch upload sets up of tasks and other JDTA information directly from source documents.

Welcome	Project Setup	Task	Product Item	Prompt Template	Data Source	RAG Template	Product Item Response	Batch Generation	Excel Product Export	Document Product Export
Task ID										
CE-TA-1.1	.а									
Job										
CONSTRU	ICTION ELECTRICI	AN .								
Rating										
CE										
Duty										
ELECTRIC	AL AND ELECTRON	VIC SYSTER	M COMPONENTS							
0										
Row										
1										
Description	1									
Develop e	electrical and elect	ronic syste	em component wi	ring diagrams						
Alias										
T1										
			Save					Del	lete	

Figure 11. Task Setup and Entry Screen

Step 2: Product Item Setup and Generation

Next, the ISD will begin defining each of the products and their associated product items (Figure 12). The interface guides users through entry of each relevant piece of information for a product item. These include:

- A UID (unique identifier for each product item)
- A link to the associated product (e.g., Task Analysis Model)
- A column or ordering value for the item (this enables users to control the order in which items appear in exported documents)
- A name for the product item
- A determination of whether this item is defined at the task level (i.e., each task will have a unique response for the item) or at the cross-task level (i.e., the product item response will cover multiple tasks)
- Dependencies to other product items (users can add one or more product items from a drop-down list to establish dependencies)
- Individual data features for the product item (these are all free text entry, allowing the user to tailor information to each product item), including:
 - Instructions
 - Constraints
 - Definition
 - o Examples
 - o Criteria



- o Context
- o Scale
- Options

Welcome Project Setup Task	Product Item Prompt Template Dat	ta Source RAG Temp	late Product Item Response	Batch Generation Excel	Product Export Document Product Expo		
Product Item ID	oduct Item ID Product ID Column		Name		Task Level		
TA_17 •	TA *		Standard		SINGLE		
Prompt Template Dependencies							
RAG_RESPONSE_00 - Content Types	*				× ×		
Instructions			Constraints				
determine the performance standards start each answer with the word "with" or "within" or "in accordance with"; list only the criteria with the actions performed							
Definition			Example				
Standards describes the minimum acce quantity, and quality that will ensure su objective, observable, and measurable, Quality, Quantity, % accuracy). May also standards, or other publications/manu	ptable level of task performance to include as accessful accomplishment of the task. Standar , and should relate to the task under study (Ti o refer to the use of technical or equipment m als.	ccuracy, speed, rds should be ime, Rate, Speed, nanuals, military	with 80% accuracy; within 30 mi	nutes;			
Criteria			Context				
none			military environment/operations				
Scale			Options				
none			none				
	Save			Delete			

Figure 12. Product Item Definition Screen

After defining the products and their associated product items, users are then taken to the Prompt Template screen (Figure 13). This is where the "brains" of NAUTICAL are configured. When creating a prompt template, users are asked to provide the following:

- A UID (unique identifier)
- A name for the template
- An associated product item (selected from a drop-down)
- A persona (e.g., an ISD or a SME) this enables the LLMs to respond based on different personas to different prompts
- A selection of a prompt strategy, including:
 - Generate from LLMs
 - Copy from another source
 - o Manually enter
 - Python code
- A series of selection options, including:
 - Product default should this be the default template for the selected product item?



- Batch generate should this template be included when batch generations are run?
- Sub Task Aliases enables substitution of task names/ids with aliases for some use cases
- Selection of which LLM to use. Each prompt template can have a different LLM associated with it, enabling the strengths of each LLM to be harnessed for different purposes. LLM parameters can also be set here, such as the model temperature (i.e., how volatile can the model responses be)
- Selection of a RAG workflow (See the RAG section for discussion)
- Selection of external data sources to use
- Definition of the system prompt
- Definition of the prompt using the Jinja templating language

To ease in development of the templates, the left side of the screen displays available Jinja variables to use based on the selected dependences, RAG workflows, and external data sources.

Welcome Project Setup Task Produc	ct Item Prompt Template Data Sou	arce RAG Template Product Item R	esponse Batch Generation Excel Pr	oduct Export Document Product Export		
Product Item Variables	Prompt Template ID	Name	Product Item	Persona		
Templates use Jinja templating language. See this <u>lutorial</u> or this <u>article</u> for a quick	TA_17_00 -	Standard	TA_17 - Standard -	Instructional System Designer 🛛 🗕 🛨		
explanation of of how jinja works.	Generation Strategy		Generation Model ID			
Template variables:	GENERATE		hosted-gpt-4			
<pre>> {{ prompt_template.persona }} > {{ product_item.name }}</pre>	Product Default		Model Temperature	0.35		
<pre>% {{ product_iten.instructions }}</pre>	🕑 Batch Generate					
<pre>- {{ product_item.constraints }} v {{ product_item.definition }}</pre>	Sub Task Aliases					
<pre>: {{ product_iten.example }}</pre>	RAG Templates		Data Sources			
<pre>>> {{ product_iten.criteria }}</pre>						
<pre>4 { product_item.context }}</pre>						
<pre>>> [] product_item.scale [] >> [5 product_item.scale []</pre>	System Prompt		Prompt			
Global Variables	Respond as factually as possible based make inferences if they are solidly grou	on the information provided. You may nded, but do not make anything up.	{{ task.description }}			
<pre>- {{ global_vars["system_prompt*].value }} - rs</pre>	Don't give information not mentioned i Do not include any direct reference to t	n the CONTEXT INFORMATION. he task description in the answer.	To determine the {{ product_item.name }}, consider this definition of {{ product_item.name }}; f product_item.definition }}			
<pre>global_vars["project_name"].value }}</pre>						
Task Variables			Also, consider this criteria that may apply: {{ product_item.criteria }}			
∴ {{ task.id }}			Apply any constraints identified: {{ pro	duct_item.constraints))		
<pre>= {{ task.job }}</pre>			Use these examples in your analysis and to construct your response:			
<pre>> {{ task.duty }} = {{ task.description }}</pre>			{{ product_item.example }}			
Dependent Item Response			Please provide a response in the format:			
<pre>variableS : {! rosponses["RAG_RESPONSE_00"].answer]]</pre>			Answer: list individual standards in paragraph format with items separated by semi-colons Rationale: an explanation of your answer			
ະ {{ responses["RAG_RESPONSE_00"].rationale]}	Sa	ave	D	elete		

Figure 13. Prompt Template Definition Screen

The next screen provides the ability to upload external data sources into NAUTICAL. This data can then be directly accessed through the prompt templates or as part of the RAG process. The



system utilizes markdown formats to represent the information in the data sources. This allows the data to be readable by both humans, the NAUTICAL system, and the LLMs.

Welcome Project Setup Task Product Item Prompt Template	Data Source RAG Template	Product Item Response	Batch Generation Ex	cel Product Export Document Product Export
Data Source ID	Description			Should this be made available as a RAG source?
MIL-HDBK-29612-3A.md	• MIL-HDBK-29612-3A.m	Use for RAG		
Content				
 > DEPARTMENT OF DEFENSE HANDBOOK MIL-HDBK-29612-3A [31 August 2001](underline) Supersedes MIL-HDBK-29612-3 a) July 1999 > DEVELOPMENT OF > INTERACTIVE MULTIMEDIA INSTRUCTION (IMI) (PART 3 OF 5 PARTS) !](Imedia/image1.png)(width="2.1110148731408573in" height="2.0260411196600177in") This Handbook is for guidance only. Do not cite this document as a requirement. > AMSC N/A AREA SESS > DISTRIBUTION STATEMENT A. Approved for public release; distribution is > unlimited. > Elif _bookmark0.anchorjFOREWORD This handbook is intended for guidance only. This handbook cannot be cited as a requirement. If it is, the contractor does not have to comply. 3. MIL-HDBK-29612-3 is Part 3 of 5 Parts. Part 3 provides guidance that may be used by DoD (all Services) and industry for the planning, design, development, implementation, evaluation, and management of Interactive Multimedia Instruction (IMI) products. Use of this 				
guidance is not mandatory. 4. Part 1, MIL-HDBK-29612-1, DoD Handbook, Guidance for Acquisition of Training Data Products and Services, provides guidance for the				
Save			Delete	

Figure 14. Data Source Setup Screen

The next screen (Figure 15) enables users to set up RAG workflows. As discussed in the RAG section, this process enables the combination of specific data source information with the LLM prompts to provide more contextually aware and precise responses. Within a RAG template, users can define the set of data sources they want to include, the LLM they want to utilize and its associated parameters, and a RAG prompt that is used to retrieve the data and send to the LLMs. This screen allows users to iteratively develop RAG workflows that then can be used within the prompt template screen.



Welcome Project Setup Task Produ	ct Item Prompt Template Data Sou	rce RAG Template Product P	tem Re	sponse	Batch Generatio	n Excel Product Export	Document Product Export
Template Help	Create RAG Template			Test	RAG Gene	ration	
Templates use <u>linja</u> templating language. See this <u>tutorial</u> or this <u>article</u> for a quick explanation of of how jinja works. Template variables:	RAG Template ID RAG_Navy_Instructional_Style_(+	RAG Template Description Using Navy Instructional Style Gu	iide	Testing (Option Model b templat	t Model ID nal) o test RAG te with. Defaults to rom template	Task ID (Optional) Single or Multiple tasks to be used for generating a RAG response	Fill
Single Task ◇ [[task.job]]	Retrieval Parameters Data Sources			hoste	ed-gpt-4 -	CE-TA-1.1.1 ×	
 §{ task.duty }} §{ task.description }} 	Navy ILE Instructional Content Style Instructional Content Style Guide	Guide_20070815 - Navy ILE ×		Filled F Testing	RAG Template Purposes Only. Task	model id, and filled prompt wil	I not be automatically saved
Multi Task		× •		The fi	ive content types a	and their definitions are:	
<pre>0 {{ task[0].10 }} 0 [[task[0].job]] 0 {{ task[0].duty }} </pre>	Number of documents to refined query engine output. retrieve Image: Check this to use raw source or refined query engine output. Image: Check this to use raw source or refined query engine output. Image: Check this to use raw source or refined query engine output. Image: Check this to use raw source or refined query engine output. Image: Check this to use raw source or refined query engine output. Image: Check this to use raw source or refined query engine output. Image: Check this to use raw source or refined query engine output. Image: Check this to use raw source or refined query engine output. Image: Check this to use raw source or refined query engine output. Image: Check this to use raw source or refined query engine output. Image: Check this to use raw source or refined query engine output. Image: Check this to use raw source output. Ima			 Concept: A concept is a category that includes multiple examples. It comprises a group of objects, ideas, or events that are represented by a single word or term, and share common features. Facts: Facts are unique and specific information usually represented in the form of a statement. Procedure: A procedure is a sequence of steps that are followed systematically to achieve a task or make a decision. A procedure contains directions or procedural tasks that are done in the same way every time. 			
 ii task[0].description is {% for t in task %} {{ t.id }} {% 							
enutor 3}							
	Model Temperature	0.35		 Process: A process is a flow of events that identify how som Topics that list a chain of events that are performed by an org represent a process. 			y how something works. I by an organization usually
	RAG Prompt List the 5 content types and their definitions			5. Principle: A principle consists of directions that outline guidelines for action in which people must adapt the rules to various situations. Principles typically require a person to make decisions when applying them.			
				Data Se	ource ID	Data Source Description	Save as Data Source
	Save	Delete					

Figure 15. RAG Screen

The next step of the workflow involves generating responses for each product item (Figure 16). Through this interface, users can select a product item/prompt template and apply it to one or more tasks. This will retrieve the prompt template and fill in the information from the variety of sources, including Jinja variables, RAG workflows, and direct data sources. Users can see how the filled in prompts look and make any necessary adjustments to the templates. From this screen, they can send the prompt to the LLM and visualize the response, including both the answer from the LLM and rationale for that answer provided by the LLM. Users can then mark these responses as:

- UNREVIEWED: the default label after the response is first generated
- ACCEPT: the user accepts the LLMs response without modification
- ACCEPT WITH MODIFICATION: the user can edit the response and accept the modified result
- REJECT: The user rejects the result of this run

Multiple runs of a product item can be conducted, and each is labeled with a timestamp so users can return to previous iterations in their process.



Welcome Project Setup	Task Product Item	Prompt Template Da	ata Source RAG Temp	plate Product Item Response E	Batch Generation Excel Product Export	t Document Product Export			
Product Item ID TA_17 +	Prompt Template ID TA_17_00 - Standarc +	Task ID CE-TA-1.1.1	Model ID + hosted-gpt-4	Response Timestamp	Fill	Generate New			
Filled System Prompt Respond as factually as po solidly grounded, but do n Don't justfy your answers Don't give information not Do not include any direct n	ssible based on the informatic tot make anything up. : mentioned in the CONTEXT IN reference to the task descriptio	on provided. You may make	e inferences if they are	Filled Prompt To determine the Standard, consider this definition of Standard: Standards describes the minimum acceptable level of task performance to include accuracy, speed, quantity, and quality that will ensure successful accomplishment of the task. Standards should be objective, observable, and measurable, and should relate to the task under study (Time, Rate, Speed, Quality, Quantity, % accuracy). May also refer to the use of technical or equipment manuals, military standards, or other publications/manuals. Also, consider this criteria that may apply: none Apply any constraints identified: start each answer with the word "with" or "within" or "in accordance with"; list only the criteria without the actions performed Use these examples in your analysis and to construct your response: with 80% accuracy; within 30 minutes; Please provide a response in the format: Answer: list individual standards in paragraph format with items separated by semi-colons Rationale: an explanation of your answer					
Answer Within a time frame of 2 h Code (NEC); using industry	ours; with at least 90% accurac y-standard symbols and notati	ry; in accordance with the N Ion.	National Electrical	Rationale The task of developing electrical a reasonable time frame of 2 hours i functionality, Adherence to the NE standards, Using industry-standar by other professionals in the field.	nd electronic system component wiring di s set. A high level of accuracy (90%) is requ C ensures that the diagrams are complian d symbols and notation ensures that the d	agrams is complex, hence a sired to ensure safety and t with widely accepted liagrams can be understood			
Feedback		Feedback C	Comment		Save				

Figure 16. Product Item Response Screen

As another illustration of the prompt template filling process, the flow from template, to filled prompt, to LLM response is shown in Figure 17.



Prompt Template

As a {{ prompt_template.persona }}, I am performing a task analysis for a {{ task.job }}. I need to determine the {{ product_item.name }} for the task:

{{ task.description }}

To determine the {{ product_item.name }}, consider this definition that may apply: {{ product_item.definition }}

Here are some examples: {{ product_item.example }}

Please provide the appropriate {{ product_item.name
}} for this context: {{ product_item.context}} from
these options: {{ product_item.options }}

Please provide a response in the format:

Answer: single option item Rationale: an explanation of your answer

Answer

4 - Catastrophic

Rationale

Developing electrical and electronic system component wiring diagrams is a high-risk task in a military environment. The construction electrician's role involves building, maintaining, and repairing power distribution systems, which are vital to the operations of any military base or outpost. Any error in the wiring diagram could result in serious malfunctions of equipment, potentially leading to injuries or loss of life among personnel, severe damage to the equipment itself as well as impacting mission success. Therefore, due to these high stakes' outcomes from inadequate performance on this task in a military context, it is considered catastrophic.

Filled Prompt

As a Instructional System Designer, I am performing a task analysis for a CONSTRUCTION ELECTRICIAN. I need to determine the Criticality of Performance for the task:

Develop electrical and electronic system component wiring diagrams

To determine the Criticality of Performance, consider this definition that may apply: Criticality of Performance points to the need for selecting tasks for training that are essential to job performance, when required, even though the tasks may not be performed frequently. Criticality is a measure of how essential a task is to job performance. The consequences of inadequate performance on certain tasks could result in injury to personnel, loss of life, or damage to equipment. Inadequate performance could have a serious impact on the mission, the operation, the product, the equipment, or the operator.

Here are some examples: the probable consequences of inadequate performance of such combat tasks as "Identifying enemy aircraft" could be loss of life and equipment. Another example, the probable criticality of performance of the task "Write trip report" is negligible. If this task were selected for training, it would be on the basis of factors other than probable criticality of performance

Please provide the appropriate Criticality of Performance for this context: military environment/operations from these options:

- 1 Minor; 2 - Marginal;
- 3 Critical:
- 4 Catastrophic

Please provide a response in the format:

Answer: single option item Rationale: an explanation of your answer

Figure 17. The Process Flow from a Prompt Template, to a Filled Prompt, to an Answer/Rationale from the LLM

Step 3: Batch Generation and Data Export

After completing a set of prompts for a product, the user may want to generate all the responses for a product at once. This is accomplished through the batch generation screen shown in Figure



18. Users can select a product and run a batch process for a single product item across all tasks (i.e., one column of a model spreadsheet), a single task across all product items (i.e., one row of a model spreadsheet), or the entire product (i.e., fill in all the cells). The batch processing engine utilizes a number of intelligent features to make this process efficient and accurate. Because there may be complex dependencies between different product items, the batch engine will identify the correct order to generate responses so that pre-conditions for an item are met and so that cyclical dependencies don't cause the system to get stuck in a loop. In addition, the system can identify if a previous batch run was halted and resume execution from that point. Finally, the batch engine utilizes parallel API calls to the LLMs to speed processing while also monitoring rate limits that may be imposed by the LLMs (e.g., only 5 queries per minute or allowed or only 10000 tokens per minute are allowed).

Product		Product Item	ask		Response Sta	atus			
ТА			ALL		ALL		*) _	Ger	ierate
product_item_id	.▲ p	<pre>prompt_template_id</pre>	task_id	model_ic	ŧ.		generation_strategy		status
TA_05	Ţ	A_05_00	CE-TA-1.1.9				COPY		planned
TA_06	Ţ	A_06_00	CE-TA-1.1.1	hosted-ą	gpt-4		GENERATE		planned
TA_06	т	A_06_00	CE-TA-1.1.10	hosted-g	gpt-4		GENERATE		planned
TA_06	т	'A_06_00	CE-TA-1.1.2	hosted-g	gpt-4		GENERATE		planned
TA_06	T	A_06_00	CE-TA-1.1.3	hosted-g	gpt-4		GENERATE		planned
TA_06	Т	A_06_00	CE-TA-1.1.4	hosted-a	gpt-4		GENERATE		planned
TA_06	т	A_06_00	CE-TA-1.1.5	hosted-g	gpt-4		GENERATE		planned
TA_06	т	A_06_00	CE-TA-1.1.6	hosted-g	gpt-4		GENERATE		planned
TA_06	T	A_06_00	CE-TA-1.1.7	hosted-g	gpt-4		GENERATE		planned
TA_06	т	A_06_00	CE-TA-1.1.8	hosted-a	gpt-4		GENERATE		planned
TA 06	T	FA 06 00	CE-TA-1.1.9	hosted-	got-4		GENERATE		planned

Figure 18. Batch Generation Screen

After products are created within NAUTICAL (either through batch generation, individual response generation, or a combination of both), the user can then visualize the results and export the data through one of two screens: an Excel product export (Figure 19), or a Document product export (Figure 20), depending on the format of the product.



TA_23 * TA_24 *TA_25 *TA_26 *TA_27 *TA_28 *TA_29 *TA_39 *TA_33 *TA_33 *TA_34 *TA_35 *TA_36 *TA_37 *TA_38 *TA_38 *Finger manuals and publica tion: tormanial trip:al manuals and publica: tion: tormanial trip:al manuals and publica: tormanial trip:TA_28 *TA_29 *TA_30 *TA_33 *TA_33 *TA_34 *TA_35 *TA_36 *TA_37 *TA_38 *TA_38 *Informa Tools; tion: tion: tormanial trip:Informa Tools; tion: tormanial trip:Ta <th>Product TA</th> <th>÷</th> <th>Job CONS</th> <th>STRUCTION EI</th> <th>LE +</th> <th>ELECTRICAL AN</th> <th>D ELE +</th> <th>Sort row</th> <th>÷</th> <th>Answer/ Answe</th> <th>Rationale</th> <th>÷</th> <th>Feedback Filter ACCEPT × ACCEPT_WITH UNREVIEWED</th> <th>MODIFICAT × × •</th> <th></th> <th>Refresh</th> <th></th>	Product TA	÷	Job CONS	STRUCTION EI	LE +	ELECTRICAL AN	D ELE +	Sort row	÷	Answer/ Answe	Rationale	÷	Feedback Filter ACCEPT × ACCEPT_WITH UNREVIEWED	MODIFICAT × × •		Refresh	
	TA_23 A Finger Dexteri ty; Informa tion Orderin g; Manual Dexteri ty; Near Vision; Problem Sensiti vity; Selecti ve Attenti	TA_24 A Common Hand Tools; Ladders; Persona 1 Protect ive Equipme nt (PPE); Special Tools; Test	TA_25 ▲ al manuals and publica tions; COMMAVA AIRFORIN ST (Series), The Naval Aviatio n Mainten ance Program (NAMP); Eloctri	TA_26 ▲ Ashore; Afloat	3 - Critica l	TA_28 * Electri cal Hazard; Burn/He at Hazard; Chemica l Hazard (c.g., Lithium or Battery Acid); Damage to Ship or Equipme at:	TA_29 ▲ 3 - Critica 1	TA_30 A , environ mont for several roasons ; it ensures the continu ous operati on of essenti al equipme nt, contrib utes to	TA_31 ▲ 1 - Low delay toleran ce	3 - At least monthly , but less than twice per week	4 - More often than other tasks	4 - Task may requi up to month or mo to learn	4 TA_35 A ctable conditi ons that may exist in a militar y ire operati o 1 on; h potenti ore al time constra n ints under high- pressur	4 - Task may require up to 1 month or more to learn	4 - 75 to 100% of the personn el perform this task	2 - Less time spent on this task than most other tasks	3 Ta fi: po: ed wi 6 mo af as en



IPRD			•)	Refresh
Document Title	MUMMEL REQUIREMENTS DOCUMENT	Header NAUTICAL	Footer	
lumn ≜ name ▲	answer Training Systems Requirements,	Acquisition and Management, OPNAVINST 1500.74	6D (July 2021)	▲ rationale
(U) Training Task Lists (NAWCTSD)	For CONSTRUCTION ELECTRICIAN, 1 Requirements Workbook Tab 9C.	O training tasks were identified for training	g. This represents NTP	The total number of entries in the response that match the criteria for inclusion is ten. All of these are "1-Over- train" which is included in the criteria. Therefore, all ten entries are counted towards the total number of training tasks identified for the project named "CONSTRUCTION ELECTRICIAN".
		Generate File		

Figure 20. Document Product Export Screen



Deep Dive: Retrieval Augmented Generation (RAG) Integration

Retrieval Augmented Generation represents a significant increase in utility to natural language processing and large language models. At its core, RAG is a hybrid approach to language generation integrating information retrieval systems and generative capabilities of LLMs. This approach allows access to vast amounts information without the explicit training of a model on new or unseen text. Using language model with RAG enables them to generate responses that are not only relevant, reducing hallucinations, but also contextually enhanced.

The RAG system dynamically fetches pertinent information from corpora in response to a query. NAUTICAL utilizes an open-source library, llama-index to orchestrate three components of RAG: retrieval, refinement, and synthesis. Each component plays a critical role in the functionality.

- 1. **Retrieval**: The first step involves identifying and extracting relevant information from a set of user-defined data sources/knowledge repositories. The data sources are currently constrained to what the user defines since an answer may be an aggregation from multiple sources. The current system utilizes a vector similarity search between the query and documents to determine relevance between prompt and text chunks. Additionally, there are keyword-based searches are utilized to increase quality of retrievals. Techniques can be varied depending on nature of the corpus and requirements of the prompt.
- 2. **Refinement**: Following retrieval, various text chunks deemed relevant to the prompt are identified, but they often contain extraneous information. To address this, a refinement system is employed. This system processes the retrieved chunks, distilling them down to more concise segments that directly pertain to the original query. This refinement is executed through a sub-generative process, where a Large Language Model (LLM) is fed both the query and the retrieved text. The LLM then synthesizes this information to output a focused and relevant answer.
- 3. **Synthesis**: Once the raw context is retrieved and/or refined, the final step is to synthesize a new answer based on this new information. This new information enhances the original user prompt, providing additional context for the user query. Integrating this carefully selected information with the user's prompt, whether using a pre-trained or fine-tuned model, enhances the accuracy of the generated responses.

NAUTICAL utilizes RAG in a unique way by allowing users to bring in multiple concepts. Documents, including manuals and specifications, are ingested, segmented, and stored in vector databases for efficient retrieval. Within NAUTICAL, in the RAG Template Screen, these documents are chosen as possible resources and a query about the document can be asked. For example, if a manual outlines the tools required for fixing a light fixture, a user could query, "What tools are needed to fix a light fixture?" In the RAG template screen, a user would cite the manual(s) as a data source, add the prompt, the number of raw text chunks to retrieve, and if they would like it refined. The system would then return an answer, in the form of the raw relevant text or a refined distilled answer based on their configuration.



Within the context of the Ready Relevant Learning Challenge, we ingested documents such as MIL-HDBK-29612A series, an extensive document containing information about different processes. Queries such as "What sources of data can be used for the ISD/SAT process?" a RAG template can be setup and our system will output answers such as those shown in the following figures.







Figure 22: Refined Retrieved Text based on RAG Prompt

RAG templates can be made for a set of relevant prompts that may be specific to a task but share the same data source. Their answers are fed into the Prompt Template as additional resources for a more complex prompt that may need additional information from different data sources. Users can add as many RAG templates as they would like to fully form a final prompt to generate a product such as a Task Analysis or the Instructional Performance Requirements Document.



RAG Templates	Data Sources
RAG_Template_1 - ISD/SAT Data Sources × × -	· []
System Prompt	Prompt
	Using one of the following resources , recommend a set of documents for the set of tasks {{ rag_responses["RAG_Template_1"].response }}
	Tasks: {% for t in task %} {{ t.description }} {% <u>endfor</u> %}

Figure 23: Prompt Template with RAG Setup

Deep Dive: Chain-of-Thought

Chain-of-Thought prompting is the process of creating complex prompts through multi-step reasoning and intermediate steps. This approach has been shown to significantly improve responses provided by LLMs for complex queries. Aptima has integrated two CoT strategies into NAUTICAL. The first approach utilizes a single prompt in which multiple steps of reasoning are explicitly requested from the LLM. Figure 24 provides an example of a prompt template, utilizing Jinja, that defines a multi-step query.

```
Here is a text with placeholders <answerl> and <answer2>. Please replace <answerl>
with <collective tasks> and <answer2> with <training tasks>, based on the following
steps and data:
First step:
Calculate the total number of entries in this response:
{% for r in responses["TA 40 00"] %} {{r.answer}} {% endfor %}
Store the response as <training tasks>.
Second step:
Calculate the total number of entries in this response that equal any of these, {{
product item.criteria }}, as the matching criteria:
{% for r in responses["LA 26 00"] %} {{r.answer}} {% endfor %}
Store the response as <collective tasks>.
Thir<u>d step:</u>
Use these variables in the response:
<Project Name> = {{ task[0].job }}
<answer1> = <collective tasks>
<answer2> = <training tasks>
Fourth Step:
Use this as the format for the response with these substitutions:
Example = {{ product item.example }}
Please provide a response in the format:
Answer: formatted statement per the example without changing any other part of the
text, including grammar and punctuation.
Rationale: an explanation of your answer
```

Figure 24. Chain-of-Thought Prompting within a Single Prompt



In this example, we explicitly define the steps of the reasoning process (e.g., "First step:"), the data needed to be considered at each step, and intermediary variables in which to store the results of each step (e.g., "<training_tasks>").

More complex CoT prompts are supported in NAUTICAL using intermediary Product Items. This is particularly useful when the result of an intermediary step in a CoT may be used by multiple Product Items in the system or when the chains of thought are so complex that an LLM cannot effectively respond within a single prompt. For example, in Figure 25 shows a prompt template for a CoT item that is used as a single step in a series of prompts for deriving a final product item response. Separating steps of the CoT reasoning into multiple product items is particularly useful for helping analysts to organize their workflow and efficiently reuse items that may feed multiple parallel chains of thought or even branching workflows. The generalizability of this strategy in NAUTICAL enables arbitrarily complex lines of reasoning to be implemented and executed.

Prompt Template ID	Name	Product Item	Persona
Cot_MS_11_12_00 +	Compare against Learning/Assessment Strategy	CoT_MS_11_12 - Compare against 👻	Instructional System Designer 🛛 👻
Generation Strategy		Generation Model ID	
GENERATE		hosted-gpt-4	
Product Default		Model Temperature	0.35
🛃 Batch Generate			
Sub Task Aliases			
RAG Templates		Data Sources	
		RRL_FY22_01 - Content Delivery Modes	0
		11_Media_Selection_Definitions - Instruction	ional Types Definitions 🗴 🗙 -
System Prompt		Prompt	
		As a {{ prompt_template.persona }}, I am perf	forming a task analysis for a {{ task.job }}.
		I need to {{ product_item.name }} for: Objective Statement: {{ responses["MS 03 0	0"].answer }}
		Use this source data:	
		Learning Strategy: {{ responses["MS_11_00"]	.answer]}
		Assessment Strategy: {{ responses["MS_12_0 Sensory Requirements: {{ responses["MS_09	00"].answer }} 00"].answer }}
		Media Characteristics: {{ responses["MS_10_	00"].answer]}
		Use these as your options:	
		Options: {{ responses["CoT_MS_09_01"].ans	wer }}
		Use this data to inform your choice:	
		Criteria: {{ product_item.criteria}}	"I content II
		Definitions: {{ data_sources[KRL_F122_01 Definitions: {{ data_sources["11_Media_Sele	ction_Definitions"].content }}
		Please provide the appropriate recommenda	tions from the list of options without using a

Figure 25. Chain-of-Thought Product Item used as an Intermediary Step in Complex Prompts



Challenge Questions and Answers

Model Selection

Question: What AI Large Language Models were evaluated for use and selected? Why?

Aptima has work with Large Language Models since Open AI released GPT-2 in 2019. We have employed a Chief Computational Scientist and have had a longstanding technical division dedicated to Department of Defense projects related to AI and Machine Learning. Although we did not complete a formal analysis of all possible models as part of this challenge, we have been continuously tracking the top performing commercial and open-source LLMs with licenses that are compatible with commercial use.

For the challenge, we identified a handful of selection criteria and compared four of the most promising LLMs given the challenge parameters and timeline; Open AI GPT 3.5, Open AI GPT 4, Claude, and Llama 2.

Selection criteria. For LLM selection, we carefully evaluated several criteria for their impact on performance and utility:

- **Self-Hostable:** Essential for data privacy and compliance with governance standards, particularly with sensitive information.
- **Context Window Size:** Important for handling complex queries, a larger context window allows for more comprehensive and informed responses by considering a broader scope of information.
- **Output Format Respect:** The LLM should accommodate user-defined response formats, like bullet points or single answers, to meet specific clarity and documentation needs.
- **Task-Specific Training:** A model's training for chat completion or question answering significantly influences its suitability for generating conversational dialogue or providing precise factual information, respectively.
- **API Access:** In order to incorporate into NAUTICAL, having access to consumer chat application is not enough, we need access to a programmatic API.

Selection Considerations	Open AI GPT 3.5	Open AI GPT 4	Anthropic Claude	Llama 2
Self hosted or Commercial	Commercial	Commercial	Commercial	Open- source/Self hosted
Context window size	4096 tokens	8192 tokens	100,000 tokens	4096 tokens

 Table 2. Top Choice LLM Selection Criteria



Selection Considerations	Open AI GPT 3.5	Open AI GPT 4	Anthropic Claude	Llama 2
Ability to respect outputs format	Yes	Yes	Yes	Did not work without finetuning
Task Specific Training	Chat	Chat	Chat/Q&A	Chat/Completion
API Access	Yes	Yes	Aptima doesn't have API access yet	Self hosted

Aptima did not get programmatic access to the Anthropic Claude model in time for use in this challenge but given its large context window it is extremely promising. We will explore its use in the future. Llama 2 is the most promising commercially available open-source model right now, and it has a larger context window than most other open-source models, but the base 13B model that we run internally at Aptima did not adequately respect the generation formatting requirements we had. Therefore, we settled on primarily using GTP-4. It has the largest context window of models we have access to and does a good job generating outputs in the format that we needed. Llama 2 is still extremely promising, and a self-hostable model will be required in the future to generate products where there are greater privacy needs (CUI/Classified data). With task-specific finetuning (potentially using GPT-4 generations as a baseline), we think that Llama 2 models will likely become viable for use in NAUTICAL.

LLM Function

Question: How does the LLM Function?

The NAUTICAL system developed by Aptima provides a structured framework for generating analysis products using large language models (LLMs). The selected LLMs function as part of a broader, integrated NAUTICAL solution that optimizes the natural language capabilities of LLMs through an innovative set of tools and interfaces described in earlier sections. It allows users to craft customized prompt templates utilizing advanced techniques like RAG and CoT prompting to enhance the contextual awareness and reasoning capabilities of the LLMs. Relevant data sources can be pulled into prompts through RAG, with CoT guiding the models through complex multi-step logic. The prompts are engineered to elicit specific formats of responses from the LLMs. Multiple LLMs like ChatGPT, Claude, and LLAMA can be used, with the user selecting the best model for each prompt. The responses from the LLMs are captured, labeled based on quality, and can be iteratively improved through prompt revisions. Full products are efficiently generated through batch parallel prompting. By tailoring prompts and incorporating external data, the NAUTICAL system is able to optimize the natural language capabilities of LLMs for generating high-quality analysis products.



Solution Requirements and Implementation

Question: What are the software and hardware dependencies if the solution were to be implemented?

NAUTICAL is designed to run on flexible cloud-based infrastructure, with core dependencies being access to managed large language model services and GPU computing for inferencing. The NAUTICAL system leverages a microservice architecture built on open-source frameworks including Yet Another Analytic Dataflow Architecture (YAADA) and tools like Gradio for rapid UI prototyping. This allows the system to be deployed on commodity infrastructure and scale horizontally. Core software dependencies include databases like OpenSearch for prompt storage and vector databases for retrieval. Compute depends on access to GPUs for running inference on large language models. As NAUTICAL matures, the UI can be rebuilt leveraging common web frameworks like React. An API gateway manages access to the various microservices. The open-source foundations make NAUTICAL adaptable across on-premise, cloud, and hybrid environments. When using publicly available, commercial LLMs, NAUTICAL has no special hardware requirements. In the future, when NAUTICAL starts operating in higher security settings and thus needs to host its own LLMs, NAUTICAL will require GPU-accelerated hardware for LLM inferences and will need occasional access to large multi-GPU high VRAM infrastructure for model fine-tuning.

Question: What is the prerequisite information/data needed to run if the solution were to be implemented?

To implement the NAUTICAL system, the key prerequisite is access to the source data that will be used to generate the analysis products. This includes documentation on the jobs, duties, and tasks that will drive the analysis, such as training manuals, job task data, equipment specs, etc. Additionally, template versions of the target analysis products are needed, for example blank workbooks for the Task Analysis and Learning Analysis Models. These provide the structure. Subject matter experts familiar with the job roles and tasks are critical for evaluating and refining the AI-generated products. Access to scaling compute infrastructure is required to run inference on the large language models. And prompts need to be engineered for the specific products, which is facilitated through NAUTICAL's templating system. But foundational data on the job tasks and a vision for the end products are the minimum needed to tailor NAUTICAL for a new implementation.

Question: What training of instructional systems designers/analysts is needed if the solution were to be implemented?

The training required for instructional system designers and analysts to effectively leverage the NAUTICAL solution is intentionally minimal, thanks to its intuitive interfaces designed based on user feedback. Fundamentally, around 2-4 hours of hands-on training would allow ISDs to learn the overall workflow - setting up projects, tasks, products, prompt templates, data sources, running generations, reviewing outputs, and iterating as needed. NAUTICAL's goal is to augment existing ISD skills, not replace them. Aptima has drafted an initial training guide for ISDs to facilitate learning the process of AI prompt engineering (Figure 26). More advanced features like selecting LLMs, tuning parameters, and developing custom RAG workflows do



require deeper expertise. But ISDs can utilize NAUTICAL's core capabilities out-of-the-box after modest training, given its alignment with established workflows. Ongoing guidance on designing optimal prompts and evaluating AI-generated content would complement the hands-on training. With an intuitive UI and templates designed for their existing skills, most ISDs can quickly get up to speed on leveraging NAUTICAL's automation and AI capabilities following limited onboarding.



Figure 26. Sample Prompts Training Document in Development

Test and Evaluation

Question: What methods were used to test and validate results?

The NAUTICAL system underwent extensive testing and validation throughout its development for the RRL challenge. The core approach was iterative - prompts and products were generated, reviewed by subject matter experts including experienced ISDs, and then refined through prompt engineering based on the feedback. This cycle was repeated across the five core products for the ten challenge tasks. Both individual prompts and full batch generations were tested to validate accuracy. In total, over 200 product items were generated and validated across 100+ iterative cycles. Additional validation involved working with representative users to test NAUTICAL's interfaces and workflows for usability and areas of improvement. Testing also evaluated the quality and relevance of RAG retrievals and CoT reasoning. LLM selection and parameter tuning was tested empirically to optimize performance. And the enhanced Excel exports were validated to ensure properly formatted imports. Through these comprehensive testing methods, NAUTICAL was rigorously validated both in terms of output accuracy and usability by representative users in the ISD domain.



Barriers

Question: What Barriers might there be for implementation of the proposed solution by the Navy or end users?

Several barriers could hinder adoption of the NAUTICAL system. As an AI-based solution, skepticism around accuracy is a common initial barrier with end users until the value is proven. Related is trust in the rationale behind AI-generated products – however, NAUTICAL's intentional *explainability* helps build confidence. Technically, availability of compute infrastructure and access to LLM APIs could pose challenges. Data security and protection of proprietary information when leveraging external systems is another barrier. The flexibility of NAUTICAL's architecture aims to mitigate these. Changing workflows and tools, even when value is clear, also faces inertia. NAUTICAL's goal of tightly integrating with existing ISD processes helps, but organizational change management is key. Potential IP issues around open-source foundations may need resolution. User-friendly interfaces and training help drive adoption, but don't remove organizational resistance. Overall, while the value proposition is strong, thoughtful change management and emphasis on accuracy, security, explainability, and flexible technical delivery can help NAUTICAL overcome barriers to implementation.

Overall Assessment

Question: What is your overall assessment of challenges and opportunities for Implementing AI LLMs for this application?

The NAUTICAL system demonstrates the immense opportunity in leveraging AI and LLMs to automate and augment the ISD analysis process. Turning repetitive yet meticulous tasks like product generation over to the machines while integrating human expertise allows each to focus on their strengths. However, thoughtfully engineered prompts are critical to unlock the value - blanketing LLMs with data generates marginal quality improvements. Integrating NAUTICAL's structured templating system with advanced methods like RAG and CoT prompting is key to steering model outputs. Change management poses challenges as workflows evolve - interfaces tailored to current skills and transparent explainability of model rationale help drive adoption. Technically, dependence on cloud API availability could introduce risk. But the microservice architecture provides flexibility. Overall, the breakthrough capabilities of LLMs are clear, but only with deliberate user-centered design and smart prompt engineering can that potential be realized. Aptima's approach with NAUTICAL offers a model for thoughtfully integrating AI to drive innovation while mitigating disruption of proven methodologies.

Throughout the RRL challenge, we continuously rated ourselves using the Challenge Scoring Rubric (Table 3). We considered both the strengths of our solution and captured weaknesses for consideration in future development. While we acknowledge the inherent bias in our self-assessment scores, we believe the associated reasoning is fair and accurate given the rating scale anchors provided and the four-week challenge timeline.

Table 3. NAUTICAL Self-Assessment



Category	Subcategory	Self Rating/ Total Points	Reasoning
Innovation (30%)	Prompt Engineering Strategy (10%)	9/10	 Pros: Structured data model decomposes products into discrete fields, allowing prompts to be highly targeted. Templating language enables reusable, customizable prompt patterns. Advanced techniques like RAG and CoT expand context and reasoning. Iterative refinement based on SME feedback improves quality over time. Support for multiple LLMs caters to model strengths. System and format prompts provide guidance to bound responses. Integration of rationale/explanations adds transparency. Cons: Requires careful tuning and testing to optimize prompts. Some very complex prompts pushed limits of LLM reasoning. Data dependence issues caused repetitiveness if not structured optimally.
	Chain-of- Thought/ Tree-of- Thought Strategy (10%)	8/10	 Pros: Allows complex reasoning processes to be broken into logical steps. Intermediary steps can feed multiple product items, improving reusability. Separates complex tasks into more manageable prompts. Provides structure to guide LLM through multi-step logic. Modular approach allows branching and recombining of steps. Promotes transparency into full reasoning process. Cons: Requires significant planning to map out full tree of reasoning. Easy to end up with convoluted trees if not carefully designed. Challenging to manage data flow through all possible branches. LLMs can still lose track of context across long chains.
	Use of Collaborative Agents Strategy	2/10	 Pros: Agents can provide real-time support answering user questions through a chat interface (future implementation). Aptima used LLMs as collaborative agents to refine prompt strategies. Complex workflows like tree-of-thought Cons: Increased coordination overhead to manage agent collaboration.



Category	Subcategory	Self Rating/ Total Points	Reasoning
			 Agents may make incorrect suggestions or perform flawed reasoning. Blending agent and human outputs risks inconsistencies. Agents would require extensive training data to be effective.
Risk Mitigation (25%)	Prompt Iteration including Self- Reflection and Revision (10%)	7/8	 Pros: Enables continuous improvement of prompts over time. Refining prompts is lower effort than refining outputs. Self-reflection by reviewing outputs identifies areas for improvement. Revision closes the feedback loop to enhance quality. Structured data model simplifies prompt adjustments. Templating facilitates prompt reusability. Aligns well with agile, iterative methodologies. Cons: Time-intensive process, especially with large prompt sets. Risk of overfitting prompts to narrow use cases. Requires diligence to regularly review and update.
	Prompt Bounding (10%)	6/8	 Pros: System prompts provide overall guidance and expectations. Requiring rationales constrains rambling responses. Providing examples bounds appropriate tone and formats. Constraint fields limit scope for each product item. Iterative refinement helps converge on ideal bounds. Const Requires extensive testing to determine optimal bounds. Overconstraining prompts risks stifling model creativity. Dynamic balancing of structure and flexibility is challenging. LLMs may still stray outside expected bounds at times.
	Data Security and Privacy Measures (5%)	5/9	 Pros: The microservices architecture could allow compartmentalization of data access. Managed cloud services may provide some built-in security capabilities. Encryption will likely used for sensitive data transmission and storage. Basic access controls will eventually be implemented. Cons: Integration with external LLM APIs introduces potential data exposure risks. Open-source components may contain vulnerabilities open to exploitation



Category	Subcategory	Self Rating/ Total Points	Reasoning
			 Distributed architecture presents challenges for holistic data security. Protecting proprietary data is difficult when using public cloud. Lack of multi-tenant isolation could allow data leakage between tenants
Feasibility & Integration (25%)	Technical 7/8 Feasibility (10%)		 Pros: The system is built on proven open-source frameworks like YAADA and common cloud technologies, suggesting technical viability. The microservices architecture and containerization provide flexibility and scalability in deployment. Use of managed services like OpenSearch offloads infrastructure management complexity. Main technical dependencies are access to LLM APIs and GPUs, which are maturing capabilities. Interchangeable components lower risk of vendor lockin as AI landscape evolves. Cons: Integration with external LLM APIs could present latency and availability challenges. Data security protections when transmitting sensitive data may be insufficient. Multi-tenant performance and isolation needs more testing under load.
	Ease of Integration (10%)	6/8	 Pros: Web UI provides intuitive application access requiring minimal training. Aligns analysis activities with current ISD methodologies. Microservices enable incremental integration, limiting scope changes. Use of excel and word for import/export facilitates adoption. Open architecture offers flexibility for integration with other systems. Cons: Migration from legacy systems requires change management. Integrating new data sources into NAUTICAL not turnkey. Depends on availability and reliability of external LLM APIs. Security considerations around transmitting/storing data externally.
	Project Management (5%)	NA	We believe this effort was effectively managed but defer on scoring this category.



Category	Subcategory	Self Rating/ Total Points	Reasoning
Impact & Expansion (15%)	Potential Impact on Instruction Systems Design and Courseware Development (10%)	9/10	 Pros: Automates time-intensive analysis activities like task/learning analysis. Generates artifacts consistently with enhanced speed and scale. Integrates AI-generated products into existing workflow tools. Provides transparency into model reasoning to build trust. Structured data framework improves courseware consistency. Accelerates speed to capability by streamlining development. Cons: Model inaccuracies may reduce trust and require oversight. Change management still needed to adopt new workflows. Less flexibility for customization compared to manual methods.
	Potential for Expansion (5%)	8/10	 Pros: Microservices and open standards allow incremental expansion. Componentization makes adding new models/data sources straightforward. Core data framework is extensible to new products. Cloud infrastructure provides ability to scale on demand. API gateway enables integration with external systems. Active open-source community around foundations like YAADA. Cons: Adding advanced features like collaborative agents requires dev work. Heavily dependent on cloud economics and vendor ecosystems. Expanding to new data domains involves tailoring prompts and models. Adoption challenges moving beyond initial applications.
Robustness (5%)	Robustness of Solution (5%)	4/5	 Pros: Microservices architecture promotes resilience and failover capabilities. Use of managed cloud services adds redundancy. Multiple LLM options prevent vendor lock-in. Open-source software increases ability to detect/address vulnerabilities. Designed for iterative refinement of artifacts over time.



Category	Subcategory	Self Rating/ Total Points	Reasoning
			 Integration with external LLM APIs introduces fragility. Lack of detail on disaster recovery provisions. Multi-tenant performance impacts not fully characterized. Security protections may be insufficient for sensitive data. Limited discussion around stress testing and fault tolerance.



List of Acronyms, Abbreviations, and Symbols

Acronym/Abbreviation/Symbol	Definition
ADDIE	Analysis, Design, Development, Implementation, and Evaluation
AI	Artificial Intelligence
API	Application Programming Interface
СоТ	Chain-of-Thought
GPU	Graphics Processing Unit
IMRD	Instructional Media Requirements Document
IPRD	Instructional Performance Requirements Document
ISD	Instructional System Design
ISD/SAT	Instructional System Design/Systems Approach to Training
JDTA	Job Duty Task Analysis
LA	Learning Analysis
LLM	Large Language Model
MS	Media Selection
NAUTICAL	Navy AI Utility for Training, Instruction, and Comprehensive Analytic Learning
OAuth	Open Authorization
RAG	Retrieval Augmented Generation
RRL	Ready Relevant Learning
SME	Subject Matter Expert
ТА	Task Analysis
UI	User Interface
UID	Unique Identifier
YAADA	Yet Another Analytic Dataflow Architecture



APPENDIX A: APTIMA AI Prompt Engineering Requirements and Guidelines



Aptima AI Prompt Engineering

Requirements and Guidelines for AI Prompt Engineering

Document Version: 1.0

Rev Date: November 2023



General Prompt Engineering Approaches

Introductory Framing

Introductory framing prompts are those that set the context or provide a lead-in to the main content of the response. They are often used to specify the subject or to introduce the purpose of the information that follows. Here are some examples:

- 1. Task Description:
 - "In the task of analyzing historical events, the following methods are employed:"

2. Question Answering:

• "To answer your question about the life cycle of stars, one must consider:"

3. Explanation Lead-in:

"When explaining the concept of supply and demand in economics, it's important to note:"

4. Guidance or Instruction:

• "For successfully completing a marathon, the recommended training schedule should include:"

5. Summary Introduction:

• "Summarizing the key points of the recent health study, we find that:"

6. Proposal or Suggestion:

• "To improve workplace productivity, the following strategies should be implemented:"

7. Hypothesis Setting:

- "The hypothesis that dark matter constitutes the majority of matter in the universe is supported by:"
- 8. Comparative Analysis:
 - "Comparing the two literary works, several thematic similarities emerge, such as:"
- 9. Predictive Statement:
 - "Looking ahead to the next decade in technology, we anticipate that:"

10. Instructional Objective:

• "The learning objectives for the course on environmental science include:"

To avoid these types of framing, you would instruct the LLM to directly provide the information without such introductions. For example, if you wanted information on the life cycle of stars without an introductory statement, you might say:

• "Describe the life cycle of stars directly without an introductory statement."

Chain-of-Thought Prompting

Here are several examples of CoT prompting strategies. We use two approaches, one with separate product items that do interim calculations, and when possible, inline prompt CoT.

Inline Prompt CoT

This is from section 3.1.1 of the IPRD and generates the calculations needed to replace the quantities in the paragraph.



The Collective TTL is a subset of the overall TTL for tasks that require the participation assistance of multiple individuals. **<answer1>** tasks were categorized as collective tasks, **<answer2>** were selected as Training Tasks. The Collective TTL follows in the embedded file.

Linked File Placeholder Embedded File 5. (U) Collective TTL

- Collective tasks or subtasks require the awareness, coordination, or cooperation of multiple individuals.
- Training will include group situations.

• Of the **<answer2>** tasks and subtasks selected for training, **<answer1>** were categorized as collective.

Pie Chart Graph Placeholder Figure 6. (U) Quantity of Collective Tasks

Prompt:

Here is a text with placeholders <answer1> and <answer2>. Please replace <answer1> with <collective_tasks> and <answer2> with <training_tasks>, based on the following steps and data:

First step: Calculate the total number of entries in this response: {% for r in responses["TA_40_00"] %} {{r.answer}} {% endfor %} Store the response as <training tasks>.

Second step: Calculate the total number of entries in this response that equal any of these, {{ product_item.criteria }}, as the matching criteria: {% for r in responses["LA_26_00"] %} {{r.answer}} {% endfor %}

Store the response as <collective_tasks>.

Third step: Use these variables in the response: <Project Name> = {{ task[0].job }} <answer1> = <collective_tasks> <answer2> = <training_tasks>

Fourth Step: Use this as the format for the response with these substitutions: Example = {{ product_item.example }}

Please provide a response in the format:



Answer: formatted statement per the example without changing any other part of the text, including grammar and punctuation. Rationale: an explanation of your answer

Here is another more complex prompt used for IPRD Section 3.1.3:

Here is a text with placeholders: <criteria1_tasks>, <criteria2_tasks>, <criteria3_tasks>, <criteria4_tasks>. Please replace the placeholders with generated data, based on the following steps and data:

First step: Calculate the total number of entries in this response: {% for r in responses["TA_40_00"] %} {{r.answer}} {% endfor %} Store the response as <training_tasks>.

Second step: Calculate the total number of entries in the responses for each of these, {{ product_item.criteria }}, as the matching criteria: {% for r in responses["LA_26_00"] %} {{r.answer}} {% endfor %}

Store the responses as <criteria1_tasks>, <criteria2_tasks>, <criteria3_tasks>, <criteria4_tasks>.

Third step: Use these variables in the response: <Project Name> = {{ task[0].job }} <criteria1_tasks> <criteria2_tasks> <criteria3_tasks> <triteria4_tasks> <training_tasks>

Fourth Step: Use this as the format for the response with these substitutions: Example = {{ product_item.example }}

Please provide a response in the format: Answer: formatted statement per the example without changing any other part of the text, including grammar and punctuation. Rationale: an explanation of your answer



Global System Prompts

- Respond as factually as possible based on the information provided.
- You may make inferences if they are solidly grounded, but do not make anything up.
- Don't give information not mentioned in the CONTEXT INFORMATION.
- Do not include any direct reference to the task description in the answer.
- Generate the answer without using a lead-in statement.
- Do not change any other part of the text, including grammar and punctuation.

System Prompts

Stop adding	Don't justify your answers.
justification text to	
answer fields.	

Copy Prompts

Copy prompts are used to copy a previous answer stored in a specific product item into the current item. This can be from the same source or a previous source, i.e. from TA to LA, LA to LA.

Copy Another Field

In the template, choose the "Copy" as the Generation Strategy and use this prompt/format in the prompt field:

Answer: {{ responses["TA_16_00"].answer }} Rationale: {{ responses["TA_16_00"].rationale }}

Add Placeholder Text

This is used when filling in a field with a static text string, i.e. ISD Input, Direct Source. Complete the following tabs/fields using this approach:

Product Item:

- 1. Instructions box: Fill in the answer with the option.
- 2. Options box: ISD Input (or Direct Source, etc.)
- 3. Save

Prompt Template:

- 1. Generation Strategy: COPY, Product Default (Checked), Batch Generate (Checked)
- 2. Prompt box: {{ product_item.options }}



Generate Prompts

General Prompt Statements

Set PERSONA and TASK	As a {{ prompt_template.persona }}, I am performing a task analysis for a {{ task.job }}.
Task to Perform	I need to determine the {{ product_item.name }} for the task: {{ task.description }} Instructions: {{ product_item.instructions }}
Source Data	Use this source data: Objective Statement: {{ responses["LA_14_00"].answer }} Learning Objective Type: {{ responses["LA_15_00"].answer }} Complete task list: {{ data_sources["JDTA_Task_List"].content }}
Data Information	Use this data to inform your choice: Definitions: {{ product_item.definition }} Criteria: {{ product_item.criteria}}
Examples	Examples: {{ product_item.example }}
Request	Please provide the appropriate {{ product_item.name }} from these options: {{ product_item.options }}
Format Response	Please provide a response in the format: Answer: single option item Rationale: an explanation of your answer

Format Types

Instructions	Prompt Format
Enter Text; Use	Please provide a response in the format:
semicolons to	Answer: list in paragraph format with items separated by semi-colons,
separate multiple	without explanation, rationale, or without using a lead-in statement.
entries	Rationale: an explanation of your answer
Single Number	Please provide a response in the format:
	Answer: single number
	Rationale: an explanation of your answer
	Please provide a response in the format:
	Answer:
	Rationale: an explanation of your answer
Select from Drop	Please provide a response in the format:
Down	Answer: single option item



Rationale: an explanation of your answer

Lookup Prompts

Example using LA_19

Product Item

Instructions	Using the first column as lookup, respond with the term from the second
	column.
Criteria	Use the Verb to KSA Level table for lookup.
Prompt Template	Choose the template for the item to use for the lookup.
Dependencies	

Product Template

Set PERSONA and TASK	As a {{ prompt_template.persona }}, I am performing a task analysis for a {{ task.job }}.
Task to Perform	I need to determine the {{ product_item.name }} for the task: {{ task.description }}
Source Data	Use this source data: Objective Statement: {{ responses["LA_14_00"].answer }} Learning Objective Type: {{ responses["LA_15_00"].answer }}
Data Information	Use this data to inform your choice: Definitions: {{ product_item.definition }} Criteria: {{ product_item.criteria}}
Examples	Examples: { product_item.example }}
Request	Please provide the appropriate {{ product_item.name }} from these options: {{ product_item.options }}
Format Response	Please provide a response in the format: Answer: single option item Rationale: an explanation of your answer

Data Sources Choose the lookup data source from the Data Source list.



System Prompt	Respond as factually as possible based on the information provided. You may make inferences if they are solidly grounded, but do not make anything up.
Prompt	Using the identified verb, {{ responses["LA_13_00"].answer }}, answer the following: {{ product_item.instructions }}
	Data Source: {{ data_sources["10_LA_Logic_H-I"].id }}, {{ data_sources["10_LA_Logic_H-I"].name }} Data Content: {{ data_sources["10_LA_Logic_H-I"].content }}
	Answer: single item without explanation or description Rationale: explain how you found the answer

LA_20, LA_21, LA_22

Product Template

Data Sources	Choose the lookup data source from the Data Source list.
System Prompt	Respond as factually as possible based on the information provided. You may make inferences if they are solidly grounded, but do not make
	anything up.
Prompt	Using the identified verb, {{ responses["LA_19_00"].answer }}, answer the following: {{ product_item.instructions }} Data Source: {{ data_sources["10_LA_Logic_L-O"].id }}, {{ data_sources["10_LA_Logic_L-O"].name }} Data Content: {{ data_sources["10_LA_Logic_L-O"].content }}
	Answer: single item without explanation or description
	Rationale: explain how you found the answer

Lookup using Python Code

Prompt	As a {{ prompt_template.persona }}, I am performing a task analysis for a {{ task.job }}.
	I need to determine the {{ product_item.name }} for {{ task.description }}.
	<pre>table_data: {{ data_sources["10_LA_Logic_R-W"].content }}</pre>
	<pre># Let's find the index of the column we're interested in. learning_outcome_index = table_data[0].index({{ responses["LA 21 00"].answer }})</pre>



<pre># Now let's filter out the values. presentation_learning_outcome_values = [row[learning_outcome_index] for row in table_data[1:] # Skip the header row if row[0] == "Presentation" and row[learning_outcome_index] != "NA"]</pre>
presentation_learning_outcome_values Please provide a response in the format: Answer: single items separated by semi-colons without explanation or
description. Do not include NA in your answer. Rationale: explain how you found the answer

Concatenated Prompts

Produces a	Make a concatenated list in paragraph form, without a lead-in statement,				
concatenated	from the following items:				
answer from					
other product	# Given data list				
items	data_list = [
	Audio Fidelity: {{ responses["LA_40_00"].answer }};				
	Appearance Fidelity: {{ responses["LA_52_00"].answer }};				
	Spatial Fidelity: {{ responses["LA_53_00"].answer }};				
	Motion Fidelity: {{ responses["LA_61_00"].answer }};				
	Tactile Fidelity: {{ responses["LA_66_00"].answer }};				
	Stimuli Fidelity: {{ responses["LA_71_00"].answer }};				
	Response Fidelity: {{ responses["LA_81_00"].answer }};				
	Ambience Fidelity: {{ responses["LA_83_00"].answer }};				
	Format Fidelity: {{ responses["LA_85_00"].answer }};				
	Content Fidelity: {{ responses["LA_87_00"].answer }};				
	# Filtering out "N/A" and joining the remaining items with a semicolon				
	filtered_list = [item for item in data_list if item != "N/A"]				
	formatted_paragraph = '; '.join(filtered_list)				
	formatted_paragraph				
	Please provide a response in the format, without explanation, rationale, or				
	without using a lead-in statement.:				
	Answer: formatted_paragraph				
	Rationale: an explanation of how you arrived at your answer				
1					



Specific Examples

LA	33	Audio	Cues
_			

Product Item ID	Product ID	Column	Name	Task Level	
LA_33 •	LA •	33	Audio Cues	SINGLE	•
Prompt Template Dependencies					
LA_32_00 - Audio Description ×					× •
Instructions		Con	istraints		
Determine whether or not the task requires audio cues for adequate training					
Definition		Exa	mple		
Sounds or signals received through the sense	of hearing.		one		4
Criteria		Con	ntext		
Audio Cues = Signals received through the ser Auditory Feedback = Feedback presented aur display presenting information capable of be Face to Face Communication = Feedback pre Tonal Audio = A very limited source of sound heres whistles sizes, hells, humore at	nse of hearing. ally by means of an auditory display to the train ing heard. sented by direct verbal means to the trainee. or noise that is used, rather than speech, for sign	ee. Auditory display is a aling or warning, e.g.,	ilitary environment/operations		li
Scale		Opt	ions		
none		A	udio Cues; N/A		
		1.			

Product Item

Compare the recommendations from xxx against the types of audio provided as criteria to determine if xxx is recommended. If any of the recommendations match the criteria for xxx, then respond with "Audio Cues" and if not, respond with N/A.

Audio Cues = Signals received through the sense of hearing.

Auditory Feedback = Feedback presented aurally by means of an auditory display to the trainee. Auditory display is a display presenting information capable of being heard.

Face-to-Face Communication = Feedback presented by direct verbal means to the trainee. **Tonal Audio** = A very limited source of sound or noise that is used, rather than speech, for signaling or warning, e.g., horns, whistles, sirens, bells, buzzers, etc.

Voice Audio = The source of sound is a live or recorded human voice.

Full Audio = A source of sound that contains all the significant elements of the sound and is suited to demanding sound recognition exercises.

Ambient Audio = A complex sound environment with sounds emanating from various sources and from various directions, including background noise and task-significant sounds.

Prompt Template:



Compare the recommendations from {{ responses["LA_32_00"].answer }} for the related duty and task against the types of audio provided as criteria to determine if {{ product_item.name }} is/are recommended. If any of the recommendations match the criteria for {{ product_item.name }}, then respond with {{ product_item.name }} and if not, respond with "N/A".

Related Job: {{ task.job }} Related Duty: {{ task.duty }} Related Task: {{ task.description }} Criteria: {{ product_item.criteria }} Options: {{ product_item.options }}

Please provide a response in the format:

Answer: single option item Rationale: an explanation of your answer