SYM-AM-25-363

Excerpt from the Proceedings

of the

Twenty-Second Annual Acquisition Research Symposium and Innovation Summit

Volume III

AI-Based DPCAP FAR/DFARS Change Support Tool

Published: May 5, 2025

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

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

To request defense acquisition research, to become a research sponsor, or to print additional copies of reports, please contact any of the staff listed on the Acquisition Research Program website (www.acquisitionresearch.net).

Acquisition Research Program
Department of Defense Management
Naval Postgraduate School

AI-Based DPCAP FAR/DFARS Change Support Tool

Jose Ramirez-Marquez—is an Associate Professor and Division Director of Enterprise Science and Engineering in the School of Systems and Enterprises at Stevens Institute of Technology. His research focuses on the development of mathematical models for the analysis and computation of system operational effectiveness research and explores the interplay between data visualization and analytical decision-making. He conducted funded research for industry and government and published over 100 refereed manuscripts in technical journals, book chapters, and industry reports. He holds degrees from Rutgers University in Industrial Engineering (PhD and MSc) and Statistics (MSc) and from Universidad Nacional Autónoma de México in Actuarial Science. [jmarquez@stevens.edu, 201-216-8003]

Joshua Gorman—is a Graduate Student at Stevens Institute of Technology with a Bachelor of Science in Computer Science. Previously, Josh garnered valuable experience as a research assistant, actively contributing to diverse application development projects ranging from meteor strike modeling to nuclear fallout predictions. He is continuing his education as the accelerated Computer Science Master's program, concentrating in cloud computing and computer networking. [jgorman4@stevens.edu, 201-216-8003]

Akram Amer—is an Undergraduate Senior at Stevens Institute of Technology majoring in Software Engineering with a minor in Computer Science. As a research assistant, he contributed to the DFARS/NDAA AI project by developing backend tools to extract and summarize legislative and regulatory language, including historical analysis of U.S. Code and National Defense Authorization Acts (NDAAs). He also worked on Department of Defense projects analyzing Justification Books, building AI-based tools to process and visualize DoD budget documents and legal codes, with a focus on backend infrastructure, NLP pipelines, and system integration. [aamer@stevens.edu, 201-216-8003]

Douglas J. Buettner—is the AIRC Deputy Chief Scientist at the Stevens Institute of Technology and is an Adjunct Assistant Professor for the University of Utah's Systems Engineering Program. His doctoral research at the University of Southern California's Viterbi School of Engineering examined issues with DoD software-acquisition strategies, identifying acquisition optimizations. His research is found in Acquisition Games and Software Defects: Dynamics of the Software Developer's Dilemma. Buettner has BS and MS degrees in Physics, with a minor in Atmospheric Science from Oregon State University, and his PhD from USC is in Astronautical Engineering. [dbuettne@stevens.edu, 201-216-8300]

Brian Mayer—is a Research Scientist at the Sanghani Center. Brian works with sponsors and collaborators to help apply Sanghani faculty research to solve government, corporate, and societal problems. Brian has helped deploy several long-term government-funded research projects to government sponsors. Brian's research interests are in the areas of decision support, forecasting, intelligence, and human-guided systems. Brian received an MS and BS in Industrial and Systems Engineering from Virginia Tech. [bmayer@cs.vt.edu, 540-231-5907]

Patrick Butler—is a Senior Research Associate at the Sanghani Center. The main thrust of his research focuses on the IARPA-funded EMBERS project that uses open-source indicators such as tweets, news, blog, weather, and so forth, to forecast population-level events such as civil unrest, elections, and epidemics. He graduated with a PhD in computer science from Virginia Tech in 2014, during which time he was awarded a SMART Fellowship. His dissertation was in the area of using knowledge discovery for intelligence analysis. [pabutler@vt.edu 571-858-3107]

Naren Ramakrishnan—is the Thomas L. Phillips Professor of Engineering at Virginia Tech and leads AI and machine learning for Virginia Tech's Innovation Campus. He directs the Sanghani Center for AI and Data Analytics and the Amazon–Virginia Tech Initiative in Efficient and Robust Machine Learning. Ramakrishnan's research interests span data science, forecasting, urban analytics, recommender systems, and computational epidemiology. He is a fellow of the Association for Computing Machinery (ACM), the American Association for the Advancement of Science (AAAS), and the Institute of Electrical and Electronics Engineers (IEEE). [naren@cs.vt.edu, 571-858-3331]

Bradley Freedman—is graduate research assistant at Virginia Tech. [bradleytfreedman@vt.edu, 540-231-5907]

Abstract

The Department of Defense's Defense Pricing, Contracting, and Acquisition Policy Contract Policy Directorate in the Office of the Assistant Secretary of Defense is responsible for periodic updates to the Federal Acquisition Regulation (FAR) and Defense FAR Supplement (DFARS) based on changes in the National Defense Authorization Act (NDAA), Small Business Administration rule changes, U.S. Department of Labor rule changes, or from executive orders. Reading through and assessing these documents for changes that require corresponding changes to acquisition regulations is labor-intensive. Further, when rule changes are proposed to the public for comments, reading and summarizing these public comments can range from straightforward to very labor-intensive.

In this paper, we report our initial research results to greatly improve the efficiency of analyzing the NDAA language for required updates of the FAR and DFARS, and issuance of memoranda and guidance using artificial intelligence, including large language models and advanced natural language processing techniques to provide an improvement in staff efficiency for these laborious tasks.

Keywords: Large Language Models (LLMs); Natural Language Processing (NLP); Department of Defense (DoD); National Defense Authorization Act (NDAA); Federal Acquisition Regulation (FAR); Defense FAR Supplement (DFARS)

Introduction

The Defense Pricing, Contracting, and Acquisition Policy (DPCAP) Contract Policy (CP) Directorate in the Office of the Assistant Secretary of Defense (OSD) is responsible for pricing and contracting policy matters across the Department of Defense (DoD). They execute statutes, executive orders, and policies through the timely update of the Federal Acquisition Regulation (FAR) and Defense FAR Supplement (DFARS) and issuance of memoranda and guidance. Fundamentally, they enable operations through business systems and standards.

The DoD DPCAP is responsible for periodic updates to the DFARS based on changes in the National Defense Authorization Act (NDAA), Small Business Administration (SBA) rule changes, U.S. Department of Labor (USDOL) rule changes or through executive orders (EOs). Reading through the changes made necessary by these multiple sources to complete required DFARS updates is labor-intensive for DPCAP staff. It also requires knowledge of all the rules in the FAR/DFARS to ensure that changes are made appropriately and references are made to the correct sections of the FAR/DFARS.

Artificial intelligence (AI) is a powerful tool that can accomplish many tasks and improve what humans can accomplish, but it has its limitations, so system development is a deliberate process that should be guided by policy and end use. Limitations can include bias, explainability, and trustworthiness (i.e. the well-known large language model [LLM] hallucination problem). Proper policy and implementation can limit bias, increase accuracy, and improve human effectiveness. When implementing AI solutions, it is important to understand these limitations and to create environments where AI systems and humans work in tandem to obtain the best results possible. For many tasks, the critical importance of human judgment means AI should serve as a complementary tool to improve human efficiency rather than as a standalone solution.

This project establishes a foundation for providing a cost-effective, scalable, semiautomated capability for managing regulatory policy updates, ensuring long-term efficiency and adaptability. This paper shares the process of learning the FAR/DFARS change process and identifying AI methods to make the task easier for DPCAP subject matter experts (SMEs) who are currently executing the tasks and provide additional support in identifying necessary changes. First, the research team worked with DPCAP SMEs to document all the steps of the process (Section 2). Then, a literature review was completed to review and discuss potential solutions (Section 3). Finally, AI methods were developed to automate certain tasks and assist the DPCAP team and were incorporated into an initial prototype to demonstrate those capabilities (Section 4). The team shares benefits, lessons learned, and future work in the conclusion (Section 5)

DPCAP FAR/DFARS Change Process

The first task is to identify change text of interest in the NDAA. Typically, SMEs read the NDAA line by line to identify text of interest. The NDAA is a lengthy document and can take quite a while to review to find all the text of interest, even using standard document search mechanisms. AI natural language processing (NLP) methods can automate all of this.

The second task is, given a text of interest snippet from the NDAA (from Step 1), to identify the locations in the FAR/DFARS that need to be edited. SMEs must rely on their knowledge or keyword searches to associate the change text to the FAR/DFARS sections.

The third task is to generate new or edited text for the FAR/DFARS given the text of interest from the NDAA (Step 1) and the text in the matching FAR/DFARS section (from Step 2). If the NDAA text affects the application of a prior rule, then that rule must be edited to comply. If there is no matching text currently in the FAR/DFARS, then new text must be generated. The proposed FAR/DFARS text must meet the requirements of the NDAA text.

The fourth task is to publish the proposed text for public comment. The proposed text is published in the Federal Register.

The fifth task is to review the comments from the fourth task. Comments are received and posted on www.regulations.gov. The comments are grouped, summarized, and posted on www.regulations.gov and in the Federal Register with the final rule.

The sixth and final task is to make any changes to the proposed FAR/DFARS based on comments received during the comment period.

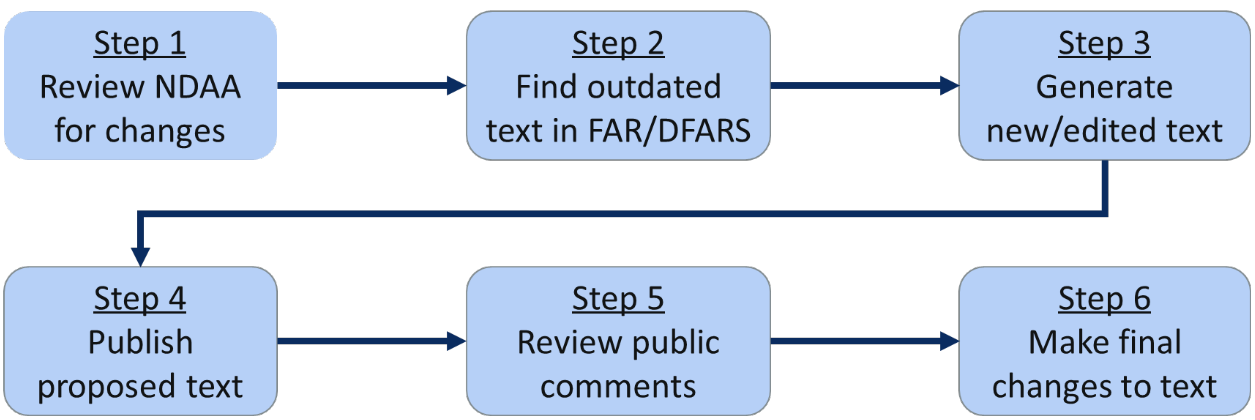

Figure 1. Task Flow Diagram

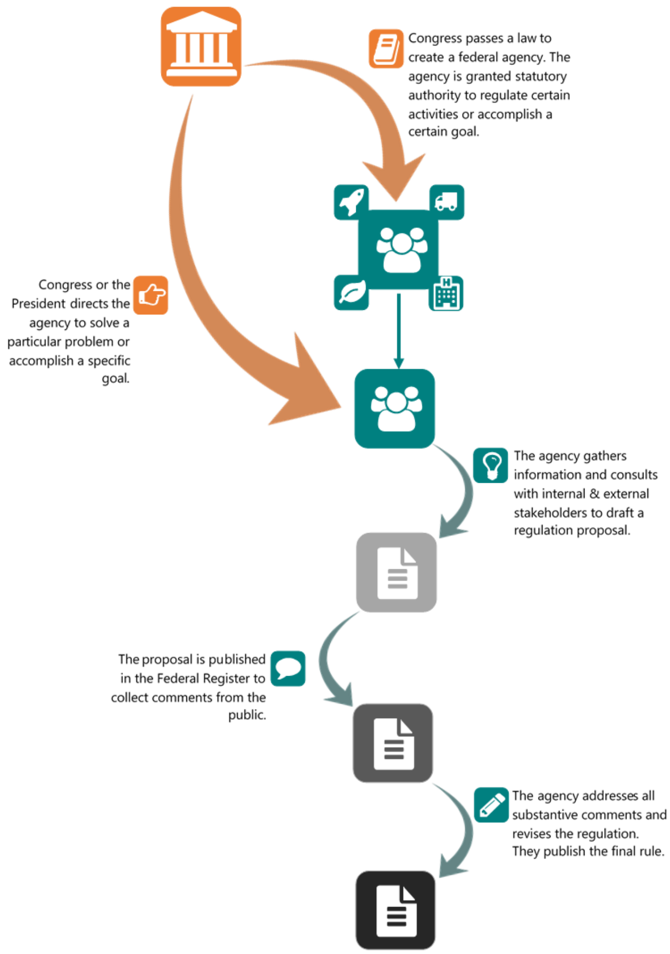

Figure 2. The Rule-making Process: How It Works

(Public Comment Project, n.d.)

Potential AI Solutions and Limitations (Literature Review)

LLMs represent a class of machine learning (ML) systems trained on vast textual datasets that can comprehend and generate human-like text across a wide range of subjects and tasks. These models exhibit several key capabilities, including NLP and natural language (NL) generation with advanced contextual understanding. The research team investigated and discussed the adaptability of current LLM and AI approaches (Antón et al., 2023a, 2023b; Lewis et al., 2020; Neeser et al., 2024; Ramirez-Marquez et al., 2024; Zhang et al., 2024) that offered potential for automation and efficiency enhancement in the context of DFARS. The applications reviewed include automated information extraction, summary generation, query resolution, and the analysis of unstructured data.

Antón et al.'s (2023b) work was particularly relevant to the current project as it leveraged NLP and generative AI to enhance the identification of critical programs within DoD Comptroller Justification Books (J-Books) and improve the understanding of their budgetary implications. The work contained two phases, the first focused on utilizing NLP pattern matching to systematically extract and analyzing J-Book sections across different DoD branches, enabling the automated identification of key terms and their contextual significance, while the last incorporated analytics to aggregate data into portfolio budgets and integrated OpenAI's LLM to associate textual data with financial insights and visual analytics, ultimately enhancing decision-making and budget analysis.

Also relevant during the discussion was the work of Ramirez-Marquez et al. (2024), which explores the application of NLP techniques to enhance talent management and workforce adaptability within the DoD. Through analysis of text data from government, industry, and academic reports, NLP algorithms can automatically identify critical skills for the DoD workforce, particularly in acquisition and defense operations. The approach supports decision-makers by providing actionable insights to optimize talent acquisition, training, and resource allocation. This NLP-driven approach strengthens the DoD's ability to strategically develop and deploy personnel by automating skill identification that enhances workforce agility, reduces skill gaps, and improves operational readiness—critical factors in addressing evolving geopolitical challenges.

Fuzzy (also called approximate) string matching algorithms[1] are techniques used to compare and find similarities between text strings, even if they are not identical. These algorithms are useful when dealing with variations in spelling, typos, or slightly different wordings. In the context of LLMs, fuzzy string matching helps in

- **text preprocessing:** standardizing and normalizing text by identifying similar words or phrases.
- **information retrieval:** matching user queries with relevant documents, even if the wording differs.
- **entity recognition:** identifying names, locations, or terms that may appear in different forms.

When applied to LLMs, these techniques improve the model's ability to process and relate different text inputs, enhancing tasks such as document analysis, search functionality, and automated text generation.

Task 1: Identifying Text of Interest in the NDAA

One solution to this process is to identify keywords that typically indicate a FAR/DFARS change is necessary and use those keywords to find text of interest in the NDAA through simple searching. The limitation of this solution is that the keywords may be used many times not in a section that requires a FAR/DFARS change.

Another approach is to take ground truth examples (i.e., previous text that led to a FAR/DFARS change) and use AI to learn textual patterns that indicate a FAR/DFARS change. Document embeddings are a way to numerically represent documents of any length as vectors (Antón et al., 2023a, 2023b). The different sections or sentences can be compared to the ground truth examples using cosine similarity (distance between) or Jaccard similarity (Thada & Jaglan, 2013) to determine the similarity of the two vectors and the likelihood that the tested text indicates a FAR/DFARS change. The limitation of this solution is that the ground truth example text may be too broad and contain text that is not particularly indicative of a change. The embedding model may find similarities based on topic (e.g., acquisition system, country) versus impact. One way to mitigate this concern is to find many (hundreds/thousands) of ground truth examples and determine the features that are similar across the examples to determine what features should be looked for in the NDAA text segments. Another way to mitigate this problem

[1] See, for example, Approximate String Matching (n.d.).

is to use an LLM to identify similarities across ground truth examples which can be used to search for text of interest in the NDAA.

A final suggested approach is to prompt LLMs to find text similar to the ground truth examples in a new NDAA using Retrieval Augmented Generation (RAG) to find references instead of generating text (Neeser et al., 2024).

Task 2: Identify Matching Text in the FAR/DFARS

It should be noted that the NDAA text may reference previous NDAA text (in a prior year) that led to a rule change. In this case, the FAR/DFARS section that resulted from the previous NDAA text is likely the section of the FAR/DFARS we want to identify.

One tool for accomplishing this is ElasticSearch (Lewis et al., 2020). ElasticSearch performs highly efficient keyword searching. The limitation of this solution is that words may be the same but will be used in a different context or for a different purpose.

Another approach is to use document embeddings similar to Task 1, but in this scenario, we can use LlamaIndex (Zhang et al., 2024) and Chroma vector stores, which are vector search solutions. These vectors are embeddings of the documents. A vector database is a collection of data that stores information as mathematical representations. We can search for vectors that are similar to the vector that represents the identified text of interest. This allows a more efficient search for multiple terms and more contextual information to be understood and returned (Schwaber-Cohen, 2024). The limitation of this solution is the same as it was for Task 1. One solution to this problem is to only search based on section titles, but this may not be specific enough.

If there is no matching text currently in the FAR/DFARS, then that likely means a new rule must be generated.

Task 3: Generate FAR/DFARS Text

If we have examples of known NDAA text (text of interest), matching FAR/DFARS text, and the newly proposed change to the FAR/DFARS text we can train a model on this information using *few-shot prompting* (Relevance AI, n.d.). The limitation of this solution is that few-shot prompting models may be overfit to the specific examples provided, leading to poor results when applied to new/different data. The success of few-shot prompting heavily relies on the quality and relevance of the examples provided. Poorly chosen or irrelevant examples can lead to inaccurate or nonsensical outputs.

Another approach is to use well-trained LLMs. LLMs are excellent tools for generating text. They can even generate text in specific styles and tones (Ullah, n.d.). If given the NDAA text of interest that indicates the new requirement, it could be asked to generate a rule or edit a current rule (if provided the current FAR/DFARS text). The limitation of this solution is that LLMs are liable to hallucinate. The FAR/DFARS rules provide specific information (e.g., numerical/section references), which are ripe for hallucination. There are a couple of ways to mitigate this. Numbering in FAR/DFARS is often a reference to other sections of that document and often those numbers may be new because the rules are new. New numbers could be provided to the LLM or entered after. Additionally, any text with references would be marked for review. Currently, SMEs use placeholders for these numbers when drafting the text manually. Also instead of asking the LLM to write the entire text we could ask the LLM just to rewrite the language that needs to be updated. The last potential mitigation is to fine-tune the LLM on a large set of government language documents or use RAG so it can learn these references better, effectively creating a reference library (Lewis et al., 2020).

To further enhance the output, we can ask the LLM to break down the prior rule before editing, similar to Chain of Thought (CoT) prompting (Wei et al., 2023). This helps the LLM perform more complex reasoning tasks by breaking down the problem into a series of intermediate steps. In this way, we are guiding the LLM to the solution instead of just asking for the output.

OpenAI's ChatGPT 4o can be used to generate summaries. ChatGPT 4o is representative of the capabilities of NIPRGPT, which is available internally to the DoD (Secretary of the Air Force Public Affairs, 2024). Llama (Grattafiori et al., 2024; Meta, n.d.) and Phi3 (Abdin et al., 2024; Microsoft/Phi-3CookBook, n.d.) are alternative open-source models that can be used. These models are small but still highly capable. They are deployed locally, which ensures complete control over sensitive data and documents, and are therefore compliant with government data protection regulations. Additionally, there is no reliance on external cloud services or data transfers. Ollama (n.d.) is also a framework that could help leverage various LLMs.

Task 5: Review Comments

Another task that LLMs excel at is summarization. The LLM will be prompted to simply summarize the comments (Zhang et al., 2023). Conveniently, the Regulations.gov API makes public comments to FAR/DFARS changes easily aggregable for input to an LLM. Document embeddings and clustering (Campello et al., 2013; Lloyd, 1982) can be used concurrently with LLMs to help structure and group the comments into similar topics and categories for more concise and usable summarizations. The limitation of this solution is that using LLMs incurs a cost, either time, resource, or monetary. The more data (text) you feed it, the more you use. Some proposed FAR/DFARS changes have thousands of comments. To mitigate the concern of costing too much, we propose narrowing down the number of comments fed into the LLM. We suggest doing this using the document embeddings and clustering results mentioned previously.

Task 6: Use Comments to Make Changes to the Proposed Text

This can be accomplished similarly to Task 3 but using the comments summary as an additional input and using the draft rule instead of the old rule. This will have similar limitations.

Proposed AI Solution Pipeline

We took the requirements from the SMEs and developed a prototype that demonstrates the potential to integrate the proposed techniques into the SME process in a way that enhances their effectiveness and efficiency while mitigating concerns due to limitations of the techniques and tools.

Task 1: Identifying Text of Interest in the NDAA

The research team worked on automating the process to identify language in the NDAA that could trigger regulatory changes. The team extracted relevant sections using key phrases used by the DPCAP staff to identify language that signals regulatory actions. The team mapped NDAA sections to historical DFARS rules through fuzzy string matching, ensuring accurate alignment. Additionally, the team developed a reusable, automated workflow for regulatory mapping, which can be extended to FAR and other regulatory frameworks for future applications. The team focused on

- **modification of existing tools:** Adapt existing AIRC NLP and LLM tools to ingest and analyze NDAA documents and historical FAR/DFARS language spanning 5 or more years.

- **keyword search and validation:** Develop AI-based algorithms to identify keywords and phrases in NDAA documents that signal potential DFARS changes. Validate identified changes by cross-referencing with historical documents.
- **change identification and suggestion:** Identify and track historical changes that correspond to DFARS and public comments.

To extract NDAA sections, NDAA data is read from Excel files containing sections of legislative text. A list of key phrases (e.g., “shall update regulations,” “modifies existing policy”) created with the help of SMEs is used to identify sections relevant to regulatory changes. Extracted sections are then stored with metadata, including source sheet and row index for traceability.

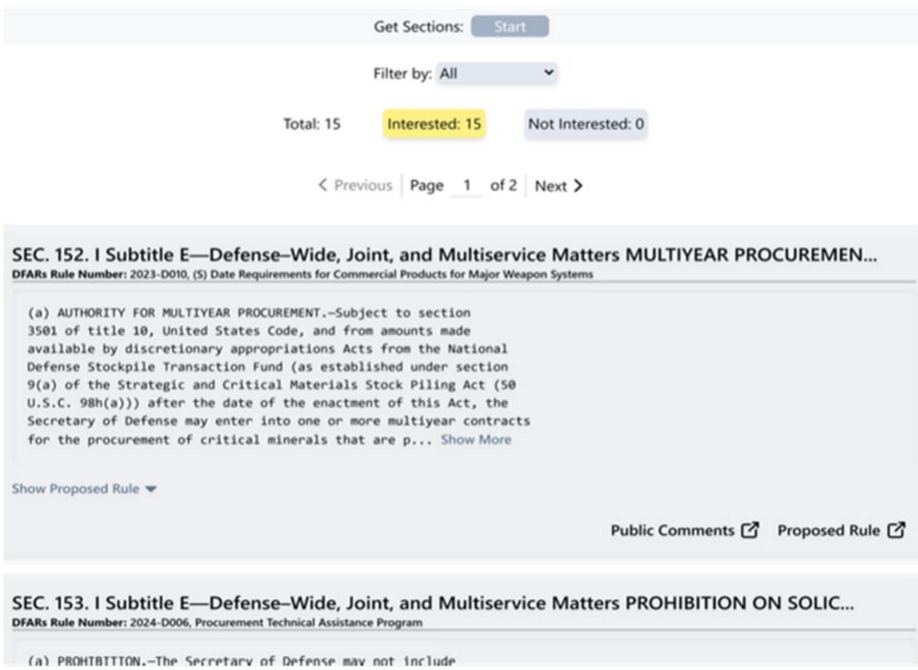

Figure 1. Screenshot of Demonstration Tool Extracting Potential Rule Change from an NDAA

Task 2: Identify Matching Text in the FAR/DFARS

To identify matching FAR/DFARS rules, first the rules must be extracted. DFARS rules are sourced from Word documents containing regulatory details. A parser scans document tables, extracting

- rule number (e.g., 252.225-7000)
- rule name
- comments link (if applicable)
- final rule notice

The extracted rules are then stored in a structured format for efficient lookup.

Once the rules are extracted, the NDAA sections can be mapped to these extracted rules. Both the NDAA and DFARS text undergo preprocessing to normalize wording and remove extraneous characters. Each NDAA section is compared to DFARS rules using fuzzy string matching (SequenceMatcher algorithm). The system selects the best-matching DFARS rule for each NDAA section based on similarity scores.

Figure 4. Screenshot of Demonstration Tool Providing Historical Proposed Rule Changes and Comments

Task 3: Generate (Draft) FAR/DFARS Text

We utilized some of the proposed techniques to create a tool that takes an NDAA year and section as well as a DFARS section as input (presumably as outputs from Task 2 and 3) and generates the text for a new rule and the corresponding text to be published with a proposed rule in the Federal Register (see Figures 5–7).

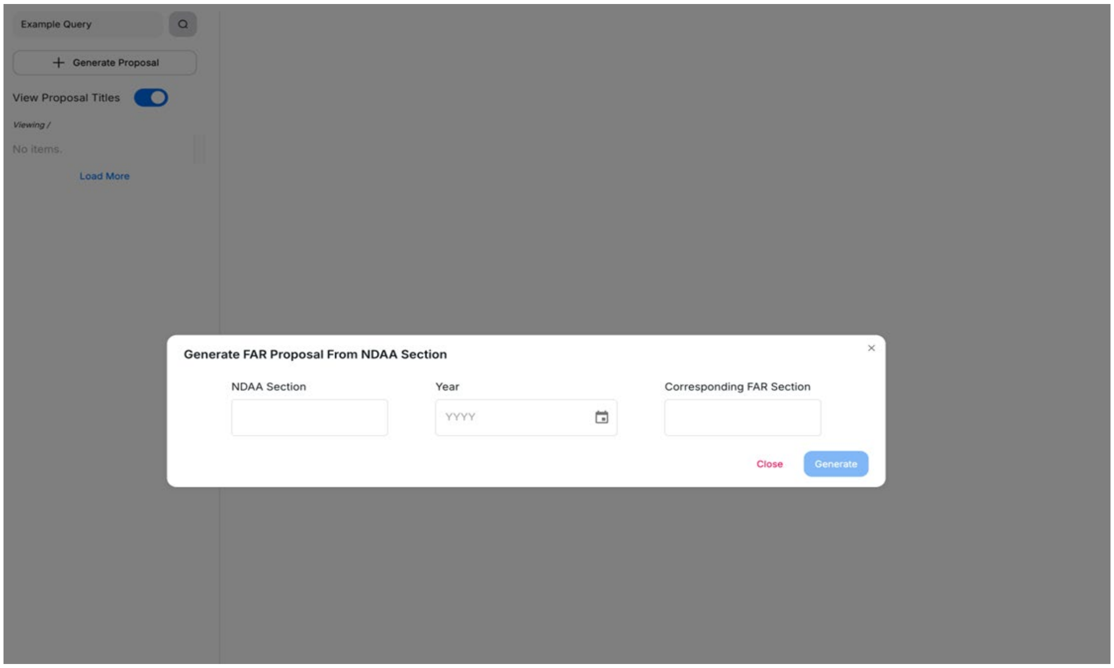

Figure 5. Inputting the NDAA Year and Section Along With the DFARS Section

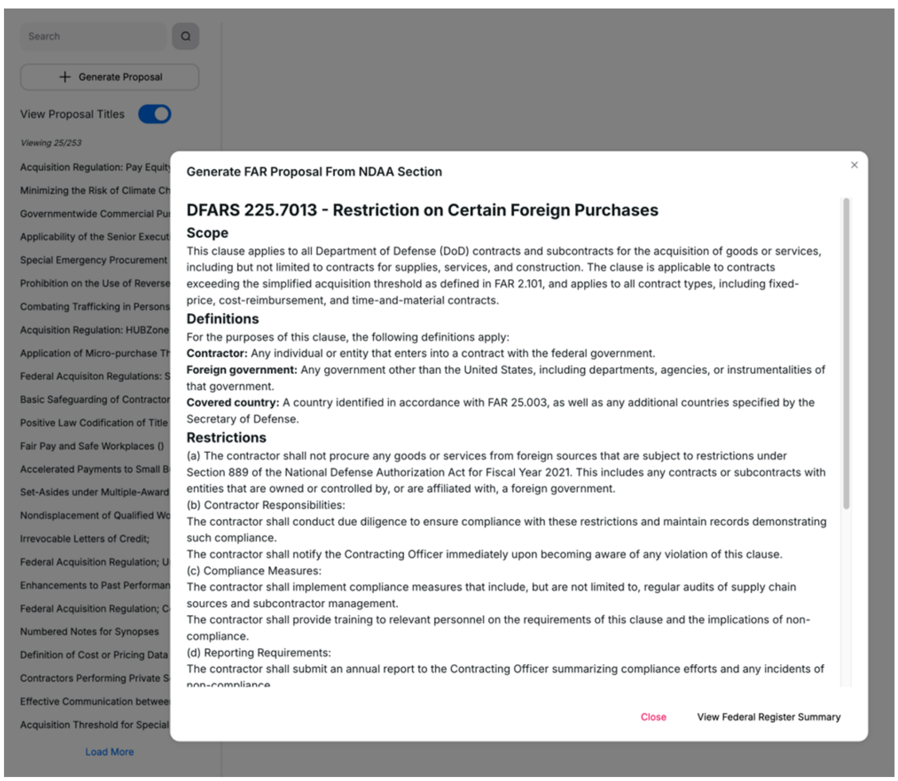

Figure 6. The Draft Proposed Rule Text Output by the Tool

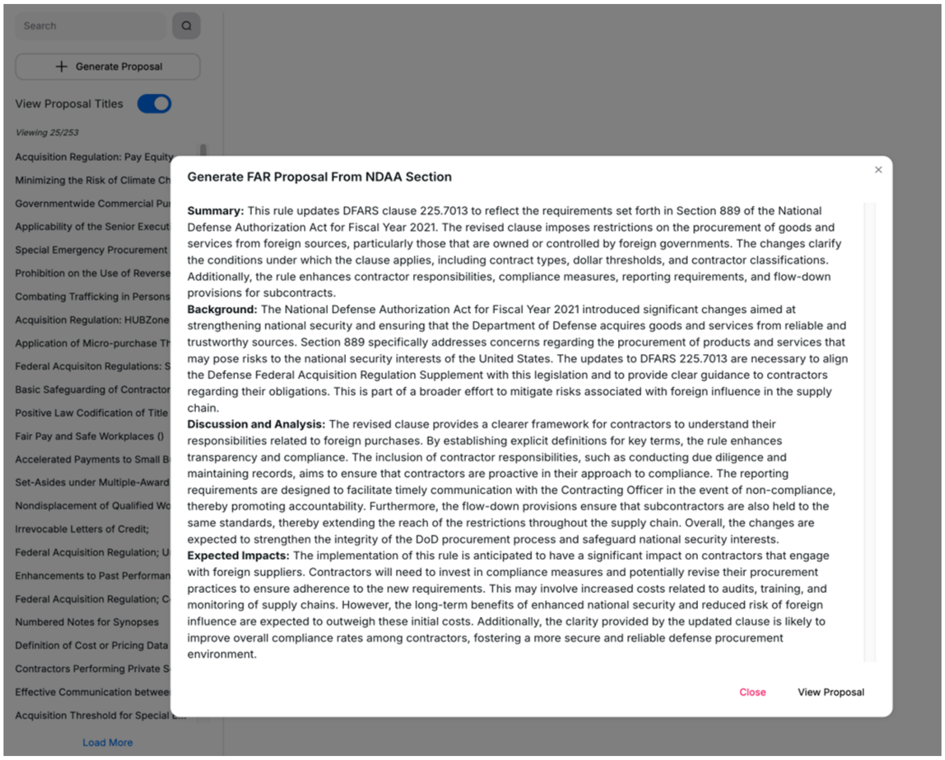

Figure 7. The Draft Text to be Published With the Proposed Rule in the Federal Register

The tool uses CoT prompting and requests specific steps and outputs to be performed to accomplish the task (see Figure 8). Virginia Tech researchers defined these steps. If SMEs provided a more accurate and thorough breakdown of the rule, the output could be improved through a more knowledgeable CoT process.

Maintain the same DFARS clause number and add or update sub clause numbers as necessary.

Maintain the same DFARS clause title.

Clarify under which conditions the clause applies, including specific contract types, dollar thresholds, or contractor classifications.

Maintain and amend the requirements from the current DFARS clause. Include contractor responsibilities, compliance measures, reporting requirements, and procedures.

Maintain references to other relevant FAR, DFARS, or government regulations and add any new references needed as a result of the NDAA text.

If and only if there are any flow-down provisions that specify when the clause must be included in subcontracts and, if so, under what conditions include or add them.

Provide a summary of the change, background of the change, discussion and analysis of the change and expected impacts of the change to be published in the Federal Register.

Figure 8. CoT Requests for Generating a Draft Rule and Corresponding Federal Register Text

Documents will be formatted similarly to the expected output but not exactly. The output of the tool (i.e., LLM) will need to be edited by an SME but can serve as a starting point. Reference and identification numbers will often have placeholders.

Note that as we test examples of the rule change, we are testing if the change already has a finalized rule that the tool will refer to for a new (or changed) rule. This is done instead of using the prior rule change and therefore impacts the output of the proposed text slightly. The research team tested the tool on both finalized rules and rules that were not yet finalized and expects to do more testing in a follow-on phase of this work.

Currently, the tool is only created and tested to generate DFARS text but should be easily extendable/applicable to creating FAR text.

Additionally, the tool currently requires a previous DFARS rule to be edited. If the NDAA requires a new rule to be created, the tool is not set up to generate that rule from scratch but could be modified (or improved) to accomplish this task as well. Tests were also not performed on the ability to generate a completely new rule. We tested our model and process on comments received from several proposed regulations. Example screenshots are shown in the figures that follow.

Task 5: Review Comments (Summarize)

We developed a prototype that can analyze, group, and summarize comments from any FAR/DFARS rule posted on www.regulations.gov. The user of the developed tool has the ability to select the proposed regulation they would like to review (see Figure 9), which is normally aligned on the left side of the page. The tool gets comments via the Regulations.gov API.

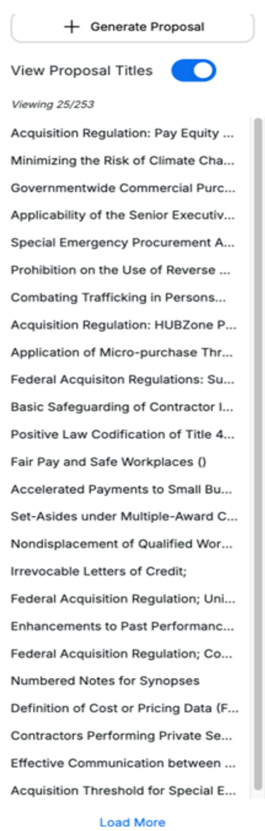

Figure 9. Users Can Select Any Proposed Rule Listed on Regulations.gov From the Column on the Left

Duplicate comments are removed, and similar comments (based on semantic meaning) are grouped in up to six groups. The groups are then summarized by an LLM (in this case GPT 4o; see Figure 10).

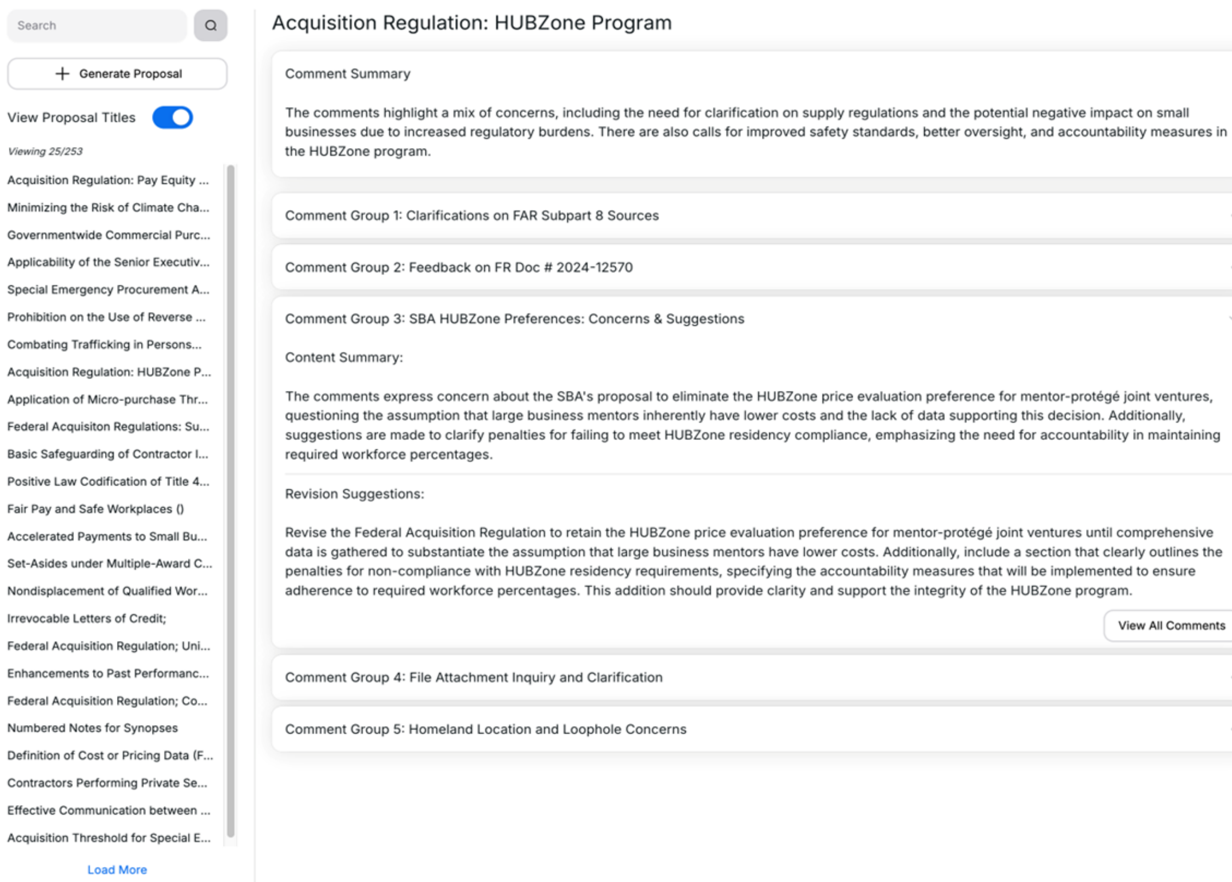

Figure 10. Groups of Similar Comments Are Summarized, and Suggestions for Changes Are Provided

If there are sufficient comments, a subset of the group's comments can be used for the summarization to reduce LLM token requests without drastically affecting the summary, since comments have been grouped based on similarity.

The interface of the tool allows the user to see all comments that are in a group (see Figure 11).

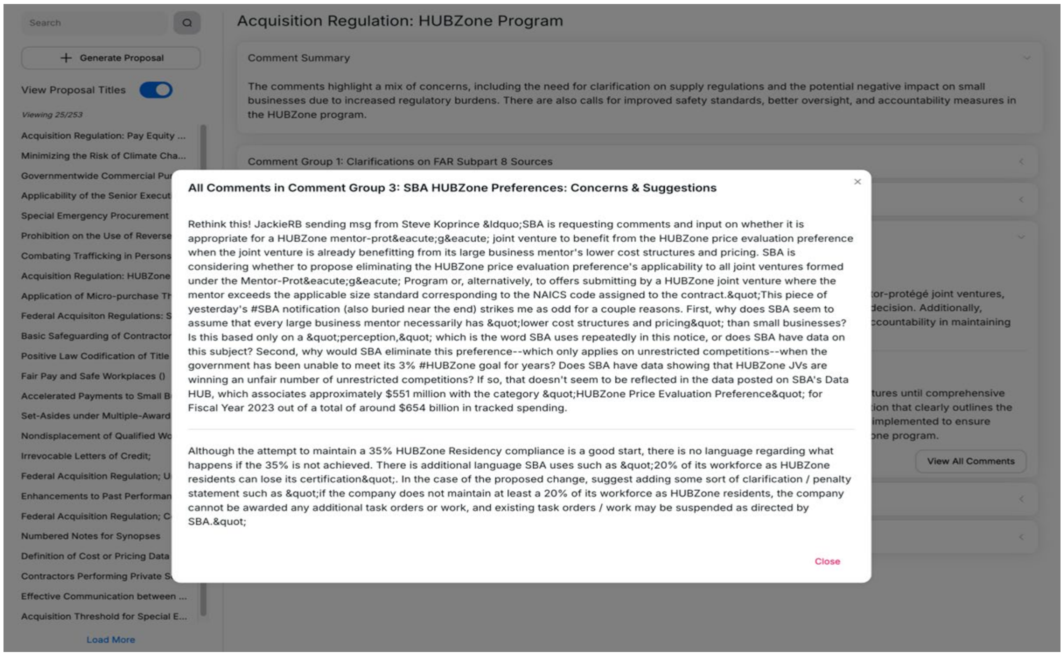

Figure 11. Actual Text of Comments Sorted by Group

Finally, the tool provides an overarching summary for the entire set of comments no matter how many groups and comments there are for the proposed rule (see Figure 12).

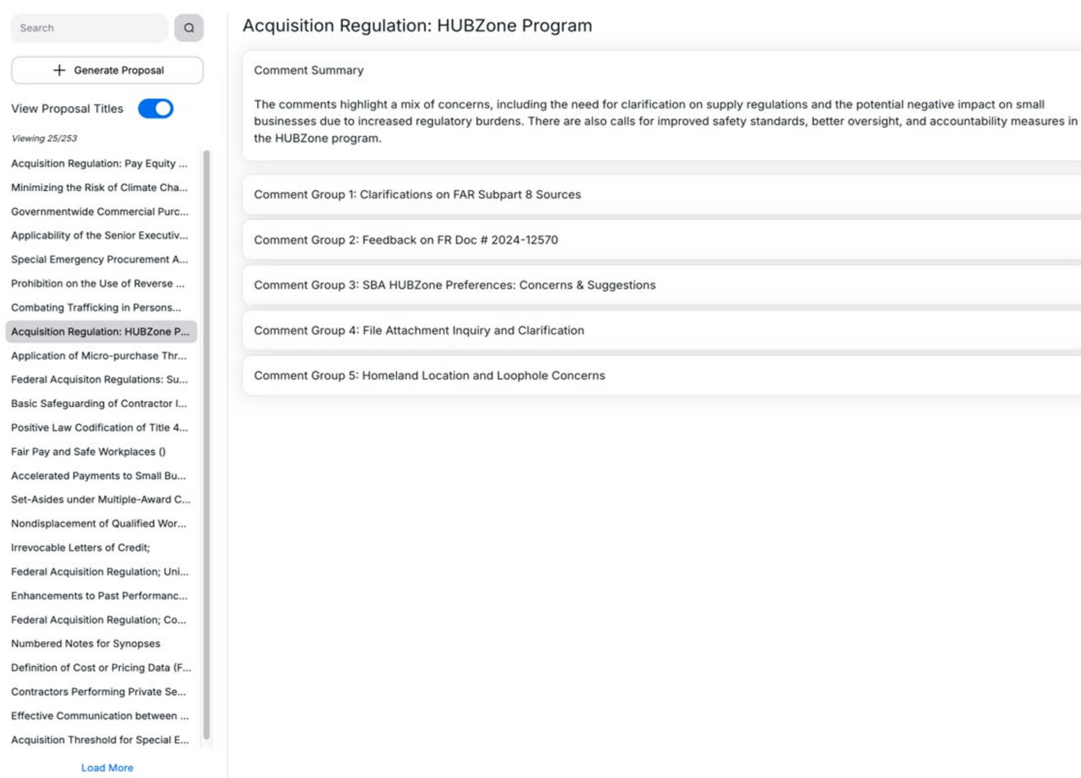

Figure 12. A Summary of All the Comments Regardless of Group

Task 6: Use Comments to Make Changes to the Proposed Text

For each group of comments, our tool provides edit recommendations to the rule based on what the comments are suggesting (see Figure 6). The user can choose whether to incorporate the suggestions from these comments or not.

The research team also tested providing the suggestions to the LLM along with the full text from the proposed rule (available through www.regulations.gov), but the text was often too long and more than the LLM token limits. There are solutions to this problem, but they were not addressed for this demonstration, since editing text was demonstrated for Task 3. Similar results (likely better because the text would actually be passed in) to Task 3 can be expected if an LLM were tasked to edit the proposed rule based on the comment suggestions shown.

Conclusions

Through the understanding that the research team gained about the FAR/DFARS change process from working with the DPCAP team, there is clearly an opportunity for efficiency and assistance through the careful implementation of AI techniques. Some techniques have limitations and therefore should be considered and implemented cautiously, but in many cases, some mitigations can be incorporated. The research team created an automated tool using AI to extract, analyze, and map NDAA sections to DFARS rules. These outputs are then fed to an LLM tool for generating proposed rule text and assessing public comments. The tool identifies key regulatory triggers and current changes, and streamlines the assessment of public comments on proposed rule changes. An executable web interface was delivered to DPCAP that integrates manual and automated steps, significantly reducing labor-intensive tasks for DPCAP staff.

The team would like to perform extensive testing of the tool on the NDAA implementation tracker for a specified period of years to validate its effectiveness and compare it to the current process. Additionally, updates can be made to improve performance and usability.

The prototype can be implemented immediately as is but has some limitations. We propose future development to address these limitations. Currently, the tool only finds and edits DFARS rules. The tool needs to be extended to work on the FAR as well. The LLM generation is relying on LLM knowledge of DFARS rules, but ingesting the raw FAR/DFARS text should enhance generated text. Another way to greatly improve generated text would be to have additional discussions with the SMEs to determine the structure and general requirements of FAR/DFARS rules. This would help inform the CoT prompting utilized by the LLM.

The current prototype is also limited to only suggesting edits to prior rules and only one rule at a time. Some rules require new DFARS sections/subsections, and in many cases, rules are updated together when an NDAA applies to multiple rules. A tool that matches this process would be more natural and easier to use for the DPCAP SMEs.

An easy implementation would be to allow SMEs to select which FAR/DFARS sections to update based on which NDAA rules, which would automatically feed the LLM rule generator (Step 3).

Finally, comments currently only result in the suggested changes, but a similar but more closed and refined process to Step 3 can be used to truly generate the final rule in Step 6.

Acknowledgements

We would like to thank Ms. Dawn Messer, OSD Defense Pricing, Contracting and Acquisition Policy, for her sponsorship and support of this analytical effort.

The views, findings, and conclusions in this document are solely those of the authors and do not necessarily reflect the views or positions of the U.S. government (including the DoD or any other government personnel) or the Stevens Institute of Technology.

References

Abdin, M., et al. (2024, August 30). *Phi-3 technical report: A highly capable language model locally on your phone*. ArXiv. https://www.doi.org/10.48550/arXiv.2404.14219

Antón, P. S., Buettner, D. J., & Ramirez-Marquez, J. E. (2023a). *ChatGPT budget justification book program element summarization: A proof-of-concept demonstration*. https://acqirc.org/wp-content/uploads/2024/05/03.-WRT-1079-AIRC-2023-TR-022-v2.0.pdf

Antón, P. S., Buettner, D. J., Ramirez-Marquez, J. E., Gorman, J. & Patel, A. (2023b). *DoD budget justification book keyword search association: A proof-of-concept demonstration*. https://acqirc.org/wp-content/uploads/2024/05/02.-WRT-1079-AIRC-2023-TR-021-v2.0.pdf

Approximate String Matching. (n.d.). In *Wikipedia*. Retrieved December 9, 2024, from https://en.wikipedia.org/wiki/Approximate_string_matching

Campello, R. J. G. B., Moulavi, D., & Sander, J. (2013). Density-Based Clustering Based on Hierarchical Density Estimates. In J. Pei, V. S. Tseng, L. Cao, H. Motoda, & G. Xu (Eds.), *Advances in Knowledge Discovery and Data Mining* (pp. 160–172). Springer. https://www.doi.org/10.1007/978-3-642-37456-2_14

Cer, D., et al. (2018, April 12). *Universal sentence encoder*. ArXiv. https://www.doi.org/10.48550/arXiv.1803.11175

Elastic. (n.d.). *Elastic: The search AI company*. Retrieved December 9, 2024, from https://www.elastic.co/

Grattafiori, A., et al. (2024, November 23). *The Llama 3 herd of models*. ArXiv. https://www.doig.org/10.48550/arXiv.2407.21783

Le, Q. V., & Mikolov, T. (2014, May 22). *Distributed representations of sentences and documents*. ArXiv. https://www.doi.org/10.48550/arXiv.1405.4053

Lewis, P., et al. (2020, December). Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (pp. 9459–9474).

LlamaIndex. (n.d.). *Using vector stores*. Retrieved February 6, 2025, from https://docs.llamaindex.ai/en/stable/community/integrations/vector_stores/

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Trans. Inf. Theory*, *28*(2), 129–137. https://www.doi.org10.1109/TIT.1982.1056489

Meta. (n.d.). *Introducing Meta Llama 3: The most capable openly available LLM to date*. Retrieved April 19, 2025, from https://ai.meta.com/blog/meta-llama-3/

Microsoft/Phi-3CookBook. (n.d.). *Jupyter notebook*. Retrieved December 9, 2024, from https://github.com/microsoft/Phi-3CookBook

Neeser, A., Butler, P., Mayer, B., & Ramakrishnan. (2024). *N. Bot automation using LLMs and plugins* [Unpublished manuscript].

Ollama. (n.d.). Retrieved December 9, 2024, from https://ollama.com

Public Comment Project. (n.d.). *How public comment works*. https://publiccommentproject.org/how-it-works

Ramirez-Marquez, J. E., Shafovaloff, G., Krzysko, M., & Verma, D. (2024, May). A semiautomated framework leveraging NLP for skill identification and talent management of the acquisition workforce in the Department of Defense. *Proceedings of the NPS Acquisition Research Symposium*.

Relevance AI. (n.d.). *Teach LLMs to mimic your style*. Retrieved December 9, 2024, from https://relevanceai.com/docs/example-use-cases/few-shot-prompting

Schwaber-Cohen, R. (2024, December 9). *What is a vector database & how does it work? Use cases + examples*. Pinecone. Retrieved December 9, 2024, from https://www.pinecone.io/learn/vector-database/

Secretary of the Air Force Public Affairs. (2024, June 10). *Department of the Air Force launches NIPRGPT*. https://www.af.mil/News/Article-Display/Article/3800809/department-of-the-air-force-launches-niprgpt/

Thada, V., & Jaglan, V. (2013, August). Comparison of Jaccard, Dice, Cosine similarity coefficient to find best fitness value for web retrieved documents using genetic algorithm. *International Journal of Innovations in Engineering and Technology*, *2*, 202–205.

Ullah, M. A. (n.d.). *Fine-tuning large language models to replicate scholarly writing styles: A step-by-step guide*. Medium. Retrieved December 9, 2024, from https://medium.com/@asadullah321/fine-tuning-large-language-models-to-replicate-scholarly-writing-styles-a-step-by-step-guide-945ac3a6d90f

Wei, J., et al. (2023, January 10). *Chain-of-thought prompting elicits reasoning in large language models*. ArXiv. https://www.doi.org/10.48550/arXiv.2201.11903

Zhang, T., Ladhak, F., Durmus, E., Liang, P., McKeown, K., & Hashimoto, T. B. (2023, January 31). *Benchmarking large language models for news summarization*. ArXiv. https://www.doig.org/10.48550/arXiv.2301.13848

Zhang, T., Ladhak, F., Durmus, E., Liang, P., McKeown, K., & Hashimoto, T. B. (2024, January). Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, *12*, 39–57. https://doi.org/10.1162/tacl_a_00632

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

www.acquisitionresearch.net