

On the Impact of Optimization Infrastructure on Large Language Models

Shaul Eliahou-Niv

Aeronautical Engineering, Technion, Haifa, Israel
Corresponding Author Email: seliahou@technion.ac.il

Abstract

In this work, we evaluate a relatively small, general-purpose large language model (LLM) on a specialized and complex Systems Engineering task. Specifically, we employ a Systems Engineering benchmark to accurately measure the model's performance. To improve accuracy and reduce hallucinations, we apply a Retrieval-Augmented Generation (RAG) framework based on a concise corpus composed of Systems Engineering-related textbooks. In addition, we investigate the impact of the RAG corpus on the accuracy of the results. We find that when RAG is combined with a subject-specific benchmark, the framework can effectively act as a judge ranking texts by their relevance to the subject and, when relevant, assessing their completeness relative to the full scope of the benchmark.

Keywords

Artificial intelligence, large language model, retrieval augmented generation, systems engineering.

INTRODUCTION

Improving the performance of large language models can be pursued through several methodological categories, each addressing different aspects of model capability, knowledge integration, and input optimization.

Model-Centric Enhancement Methods

A prominent strategy in this category is finetuning, in which a pre-trained model undergoes additional training on a domain- or task-specific dataset. This approach allows the LLM to obtain high accuracy in applications for certain specialties. On the other hand, it is important to note that finetuning is a complicated and difficult process to carry out. Full fine-tuning process requires a large amount of labeled and sorted high quality data. All the more so, significant computational resources are required, which turn the method to an unrealistic and complicated for everyday routine applications [1] [2] [3].

To deal with the problems of the finetuning approach as described above, scientists have developed a focused and a parameter efficient finetuning approach. These techniques include various directions reported in technical literature such as LoRA, Prefix Tuning and Adapters. While using these techniques the user needs to update only a small portion of the model's parameter meanwhile insuring that the model is adapted to a new domain and emphasizing the characteristics of this domain. Therefore, this method is characterized by using cheaper computational resources and infrastructures [4]. Another approach reported in the literature is focusing on the model itself through continuous and gradual learning. In that approach the model's knowledge is updated progressively over time without a need for full training each time. This idea seems attractive but in practice it involves a lot of complex challenges. The main challenge to encounter

is the problem of 'catastrophic forgetting'. I.e., the situation where the model is required to learn new information each time it is updated, which harms the previous accumulated knowledge already acquired by the model [5].

Knowledge-Based Enhancement Methods

Another approach includes a collection of methods that focuses on enriching the model's knowledge without changing its internal parameters. The most prominent example of the application of this method is the Retrieval Augmented Generation [RAG] which incorporates an external knowledge source or data set from which the relevant information is extracted for the purpose of inference and retrieval. [6]. The RAG method has significant advantages over the other methods previously described. These advantages include, among others, improved factual accuracy and reduced model error rates. The most important advantage is the ability of the model to be continuously updated from the external data set without any need to retrain the basic model [7].

In order to reduce the cognitive load on the LLM model various methods have been developed including a combination of different tools and external plugins. The external tools and the plugins for implantation of this strategy includes data bases, computation engines, or dedicated API's for the specific applications. This method of reducing the cognitive load on the LLM significantly improves and enhance the reliability and the accurate output produced by the LLM.

Input-Optimization Methods

There is also a group of methods that aim to improve the performance of the model by presenting it with information in a better and more organized way. The field called prompt engineering is at the center of these approaches. Methods

such as few-shot prompting, chain of thought, ReAct prompts, and multi-step prompts all help the model think better with improved reasoning quality and more accurately without changing the model itself [8] [9].

Structured prompts use predefined templates, formats, or rules such as JSON schemas, tables or rule-based instructions to ensure that the model's responses are consistent and to direct the model to give the type of the results expected by the user. When the model is bounded by a clear framework, it is less likely to "runaway" to unexpected interpretations. Constrains can also be added to the model they help it to stay focused and to give consistent responses. When structured prompts are used the model responses are more consistent, the model output become clear and easy to read due to the fact that the model understands exactly what is expected of it. The main insight is that when the information is well structured and well bounded, the model's performance improves significantly [10]. It is important to note in this context the self-consistency methods. Self-consistency methods make the model give several different answers, and then to choose the one that sounds most logical and appropriate. The process is like running several models together within the prompt itself and instead of the model giving one answer and trusting that it is correct the model is asked to generate several answers. An external system or the model itself compares the answers and chooses the answer that repeats the most or sounds most coherent or the answer that is most statistically likely. All of this leads to a result that is more stable and has fewer hallucinations and is more accurate when the models are allowed to generate only one answer.

Of all the methods reviewed above, we chose in this study to use methods for improving model performance that are based on adding external knowledge rather than retraining the model. We focused mainly on the RAG architecture and infrastructure for improving model parameters that were particularly suited to dealing with complex engineering problems while continuously improving parameters based on data collected during work. It is true that sometimes it seems that fine-tuning (which belongs to methods that focus on changing the model itself) can be good solution, but it is also has clear disadvantages. One of the disadvantages in the fine-tuning model is that the model is exposed to only a relatively small number of examples, especially compared to the huge amount of text on which it was originally trained. While fine tuning can strengthen the model's ability to work in a particular domain or follow specific patterns, its success depends heavily on how relevant the examples are to the topic? How high quality they are? And how consistent they are? If any of these things are inaccurate, the effectiveness of fine-tuning is significantly reduced.

Fine-tuning offers several advantages, including improved domain-specific accuracy, stronger contextual alignment within the target domain, and the ability to control stylistic tone during generation.

However, this capacity is less important for us, because our research focuses on calibrating the system using a benchmark consisting of a large number of multiple-choice questions. Although a model that has undergone fine-tuning can provide similar benefits to those offered by the RAG method, there are some important differences between the approaches. In practice, in the case we are investigating, fine-tuning requires transforming all of our data into a collection of prompt-answer pairs, because the additional training process must work on labeled data. This creates a significant problem in our research: since the source material is long texts in the style of lectures or book chapters, it is difficult to know how to transform each part of such text into a clear and meaningful prompt. Even if we were to succeed in devising a suitable prompt for each section, another difficult question remains: how to correctly divide the long text into sections? This is not trivial at all. Because of all of this, it is clear that fine-tuning requires a lot of preliminary manual work, and its results depend very much on the quality and accuracy of the data generated from this process.

Compared to the limitations of the fine-tuning method, especially in the case of the research we are conducting in terms of data handling and model training, RAG has several advantages relevant to our research, as listed below. Using RAG allows partial update of the system's knowledge simply by adding new information to the external knowledge base, which can immediately utilize during inference. This is a significant advantage of the RAG method over fine-tuning. In environments where knowledge is updated rapidly, a fine-tuning model may become obsolete quickly, because the learned parameters remain static after training. This limitation is especially significant in dynamic organizations characterized by frequent changing documentation. Under such conditions, maintaining a fine-tuned model will require periodic retraining, more complex and resource-intensive process than updating documents in a RAG based knowledge base. In addition to all these limitations of data organization and updating, there are also computational hardware limitations. Fine-tuning requires dedicated hardware such as GPUs and can be resource intensive, especially for large models.

It also requires a considerably higher level of machine-learning expertise compared with the relatively modest skill set needed to implement a straightforward RAG pipeline. For these reasons, a concise, unlabeled knowledge base within a frequently updated domain is far better suited to a RAG-based approach. The primary barriers to fine-tuning in our context are the absence of a high-quality labeled dataset and the substantial computational and expertise requirements associated with the method. Consequently, we adopt a RAG-centered strategy in this research project.

MOTIVATION AND FRAMEWORK

In this research, we used the RAG infrastructure that we developed in a preliminary study, and which is described in detail in our previous paper [11]. This infrastructure is

illustrated in Figure 1 below.

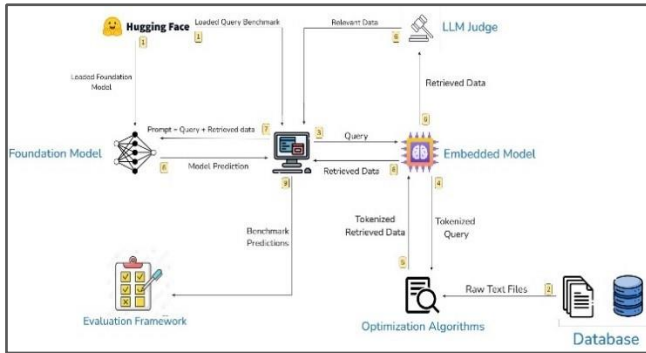


Figure 1. RAG System Workflow

This diagram outlines the RAG system's process, starting with loading the foundation model and leading through data retrieval, prediction, and evaluation stage to evaluate performance and to calibrate the model, we executed the benchmark on our custom RAG infrastructure, comprising 1,144 systems engineering questions, distributed across systems engineering topics. The questions taken from Hugging Face SysEngBench <https://huggingface.co/datasets/rabell/SysEngBench>. The distribution of question categories is illustrated in the accompanying diagram [11].

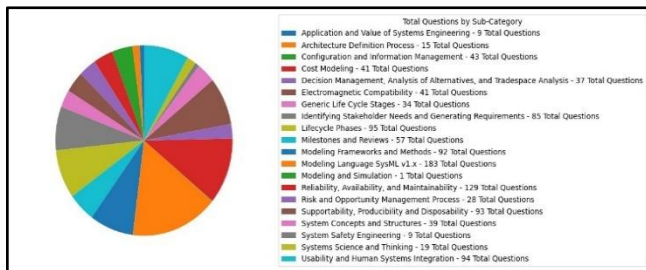


Figure 2. The distribution of the benchmark questions categories

In a preliminary study, we evaluated a relatively small, general-purpose large language model on a highly specific and well-defined Systems Engineering task. To do so, we employed a Systems Engineering benchmark designed to accurately measure model performance. To improve accuracy and reduce hallucinations, we incorporated a Retrieval-Augmented Generation (RAG) framework, supported by a concise corpus constructed from Systems Engineering-related engineering textbooks. The model exhibited measurable improvements when augmented with RAG capabilities, indicating that the corpus contributed knowledge that enabled the base model to answer a greater number of questions correctly. It is reasonable to infer that this improvement resulted from the corpus containing information at least partially aligned with Systems Engineering, the domain assessed by the benchmark. Conversely, had the corpus been less relevant to the subject matter or significantly narrower in scope, addressing only a limited aspect of Systems Engineering the contribution of the RAG mechanism would likely have been substantially reduced, and in some cases might even have degraded the

model's baseline performance.

In the present study, our objective is to investigate how both the size and the content of the corpus influence the accuracy of the large language model.

RESULTS AND DISCUSSION

System Calibration

In order to evaluate the impact of the RAG infrastructure we developed, an initial calibration was performed to establish the system's capabilities and baseline performance. The evaluation was conducted using 1,144 questions from the benchmark, as described in the figures below.

Figure 3 presents a comparison of the performance of the Gemma 2 2B model in terms of accuracy, comparing the baseline model with the enhanced model, segmented by different Systems Engineering topics according to the categorization adopted in the benchmark. Across all topics, a significant improvement was observed, with an overall average improvement of 27.77%.

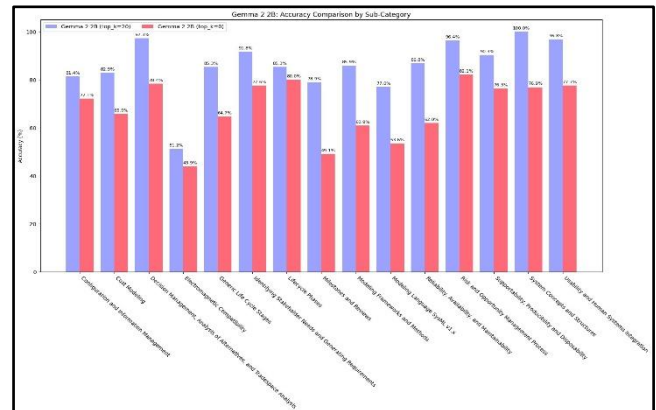


Figure 3. Model Accuracy by categories

Figure 4 illustrates the impact of the infrastructure on model performance in terms of the confidence score, again segmented by the same topics. In this case as well, a significant improvement was achieved across all evaluated topics, with an overall average improvement of 46.66%.

From the two figures, it is evident that for certain topics a particularly substantial improvement was achieved in both metrics, while for others the improvement was more moderate. This interesting finding will be further investigated in future work, including an examination of the correlation between the accuracy metric and the confidence score. These findings will support a meaningful, precise, and coordinated calibration of the developed infrastructure for addressing Systems Engineering challenges, tailored to the specific topic being addressed.

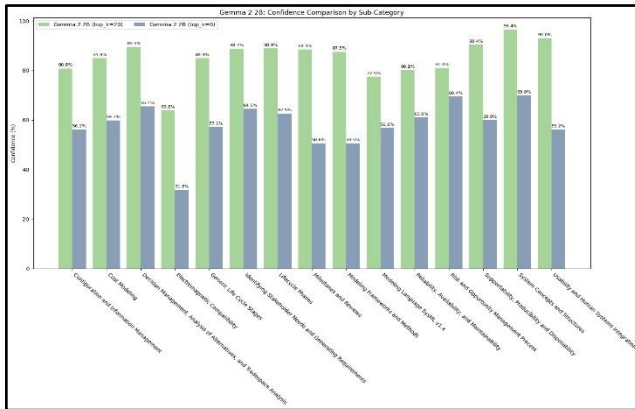


Figure 4. Model Confidence Score by categories

We employed the Mistral-7B model, which demonstrated both the capacity to improve when used within a RAG framework and a high sensitivity to the quality of the data provided. For each text under evaluation, we constructed a dedicated RAG configuration with a strictly limited corpus consisting solely of that individual text.

For the initial calibration and the system testing, we ran the entire benchmark test which included 1,144 multiple choice questions divided by topics as described in the previous chapter. We measured the overall accuracy of the model within this RAG setup. This overall accuracy essentially serves as a score assigned to the corresponding text. We divided the tested texts into two groups that differed radically from each other. The first group included texts arranged in ascending order of relevance from sources that were completely unrelated to any engineering (for example the first Harry Potter book) to materials that were highly relevant to various engineering fields and systems engineering theories (such as the INCOSE Systems Engineering Handbook, which is a recognized industry standard [12]).

The second group comprised highly relevant texts arranged in increasing order of scope, beginning with a single chapter from the INCOSE Systems Engineering Handbook and extending to the full textbook.

For each text, the resulting score reflects the total accuracy achieved by the model when assisted by a RAG framework that retrieves information exclusively from that text. Consequently, higher scores indicate stronger model performance under the corresponding RAG configuration.

Section 1 – Comparing the Relevance of Texts

In this section, we evaluated a series of texts arranged in increasing order of relevance to the field of Systems Engineering:

- Harry Potter and the Sorcerer's Stone
- Introduction to Algorithms
- Elementary Principles of Chemical Processes
- INCOSE Systems Engineering Handbook

The numerical value associated with each text represents the accuracy achieved by the foundation model when supplemented with retrieved data from the respective text.

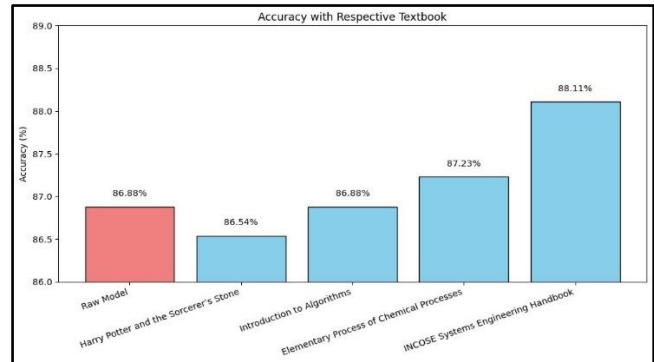


Figure 5. Accuracy with perspective textbooks

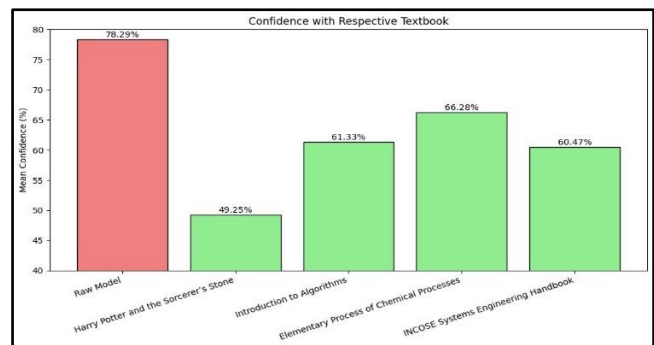


Figure 6 Confidence score with Perspective Textbooks

The phenomenon of a reduced confidence score resulting from the use of a RAG framework has been discussed extensively in our previous publications. The increase in accuracy is accompanied by a decrease in confidence because the model is required to address topics on which it was not directly trained.

Section 2 - Comparing Texts' Scope

In this section we tested segmented chunks of a highly relevant text, to see whether the RAG and benchmark can detect how “complete” a text is. We used the “INCOSE Systems Engineering Handbook” segmented to a different number of chapters, with the idea being that a segment which contains a smaller number of chapters has a more limited scope. The number associated with each segment is the accuracy of the foundation model enhanced with retrieved data from the segment in question. The finding depicted in fig. 7 show significant improvement in the accuracy parameter relative to the basic model, when adding the first relevant segment to the RAG corpus. The rate of the accuracy improvement becomes moderate as we add additional relevant materials to the corpus, (in this case additional segments from the INCOSE Systems Engineering Handbook). This experiment shows that when a collection of relevant texts is incorporated into the RAG corpus, the performance of the model improved significantly, even if there is not many such texts or they are created dynamically during the system use. This finding clearly indicates that the ability to update the model dynamically is a distinct advantage over other methods. The findings depicted in fig. 8 show the variation of the confidence score parameter depending on the enrichment of the corpus. In this case, it is

expected to see that the confidence level decreases when the model is required to deal with cases that is not previously recognize based on the initial data it was trained on, but as the RAG corpus is enriched with relevant data, the confidence parameter improves.

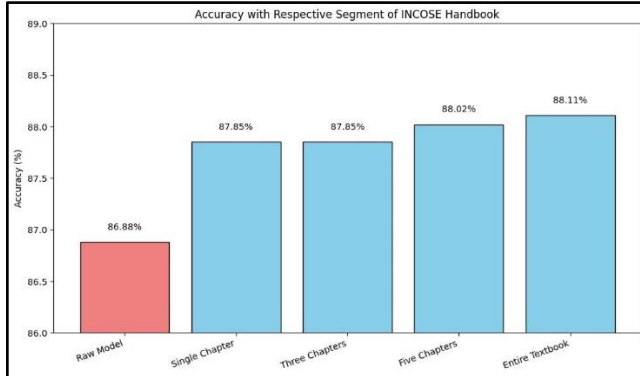


Figure 7. Accuracy with Respective Segment of INCOSE Handbook

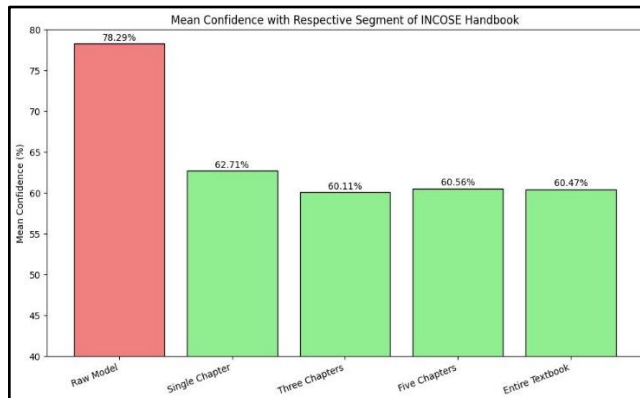


Figure 8. Confidence Score with Respective Segment of INCOSE Handbook

From the results of this experiment, we observe a significant increase in the confidence score compared with the experiment described in Section 1. Even with a corpus containing only a single chapter from the INCOSE Systems Engineering Handbook, the confidence rises sharply, followed by a more moderate increase as the corpus expands. This finding indicates that a corpus containing relevant text substantially improves the model's performance, largely independent of the total amount of material. In addition, the pronounced increase in the confidence scores relative to the experiment described in Section 1 suggests that the model exhibits a higher degree of certainty when providing its answers compared with the first experiment.

CONCLUSIONS

In this research, we built a well-tuned RAG infrastructure to deal with complex engineering problems using LLM. We also evaluated the system's performance in dealing with various methodologies in the field of systems engineering. We used a benchmark to evaluate the system's performance. We measured important parameters such as the system's

accuracy and the system's confidence scores. We found the trend of change of these two parameters as a function of the data sets we used. We showed that this system can be used also for performance evaluation of various engineering documents.

These findings suggest that with a substantially larger benchmark one comprising a significantly greater number of questions it may be possible to accurately and automatically assess the relevance of a given textbook to a particular subject, as the increased granularity would allow for more precise differentiation among texts.

Our results further indicate that this scoring methodology is more effective for evaluating relevance than scope. Although the relative ordering of the segmented text units was consistent, the performance gap between using a single chapter and the full textbook was unexpectedly small. This implies that the model can benefit from retrieved information even when the exact answer is not explicitly present in the corpus, provided that the retrieved content is sufficiently immersed in the terminology and conceptual structures of the target domain. In addition, we have shown that using the RAG method allow us to constantly update the data set, compared to retraining the model as required by traditional methods. Furthermore, the user privet database is kept privet where compartmentalization or commercial confidentially is required.

Acknowledgement

The author would like to thank Mr. Ofir Ben-Avi, an undergraduate student from the Computer Department at Technion, for his assistance with the software issues.

This research was made possible thanks to the generous support of the Bernard M. Gordon Center for Systems Engineering at the Technion.

REFERENCES

- [1] Devlin, J., Chang, M.W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL. <https://arxiv.org/pdf/1810.04805>.
- [2] Liu, Y., et al. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692. <https://arxiv.org/pdf/1907.11692>.
- [3] Hu, E., et al. (2022). LoRA: Low-Rank Adaptation of Large Language Models. ICLR. https://proceedings.iclr.cc/paper_files/paper/2025/file/906c860f1b7515a8ffec02dcac74048-Paper-Conference.pdf.
- [4] Li, X. L., & Liang, P. (2021). Prefix-Tuning: Optimizing Continuous Prompts for Generation. ACL. <https://doi.org/10.48550/arXiv.2101.00190>.
- [5] Parisi, G. I., et al. (2019). Continual Lifelong Learning with Neural Networks: A Review. Neural Networks. <https://doi.org/10.48550/arXiv.1802.07569>.
- [6] Lewis, P., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. NeurIPS. <https://doi.org/10.48550/arXiv.2005.11401>.
- [7] Guu, K., et al. (2020). REALM: Retrieval-Augmented Language Model Pre-Training. ICML. Brown, T. <https://doi.org/10.48550/arXiv.2002.08909>.

-
- [8] B., et al. (2020). Language Models are Few-Shot Learners. NeurIPS. <https://doi.org/10.48550/arXiv.2005.14165>.
- [9] Kojima, T., et al. (2022). Large Language Models are Zero-Shot Reasoners. NeurIPS. <https://doi.org/10.48550/arXiv.2205.11916>.
- [10] Wei, J., et al. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. <https://doi.org/10.48550/arXiv.2201.11903>.
- [11] Eliahou-Niv, S.: Leveraging the performance of Large Language Models in Systems Engineering applications JISEM Vol. 10 No. 27s 2025. <https://doi.org/10.52783/jisem.v10i27s.4524>.
- [12] INCOSE. "Systems Engineering Handbook", Version 2a June 2004. <https://productrealize.ir/library/INCOSE-Systems-Engineering-Handbook-A-%E2%80%9CWhat-To%E2%80%9D-Guide-For-.pdf>.