Find Before you Fine-Tune (FiT): How to Identify Small-Scale LLM Suitable for Cybersecurity Question-Answering Tasks?

Anonymous ACL submission

Abstract

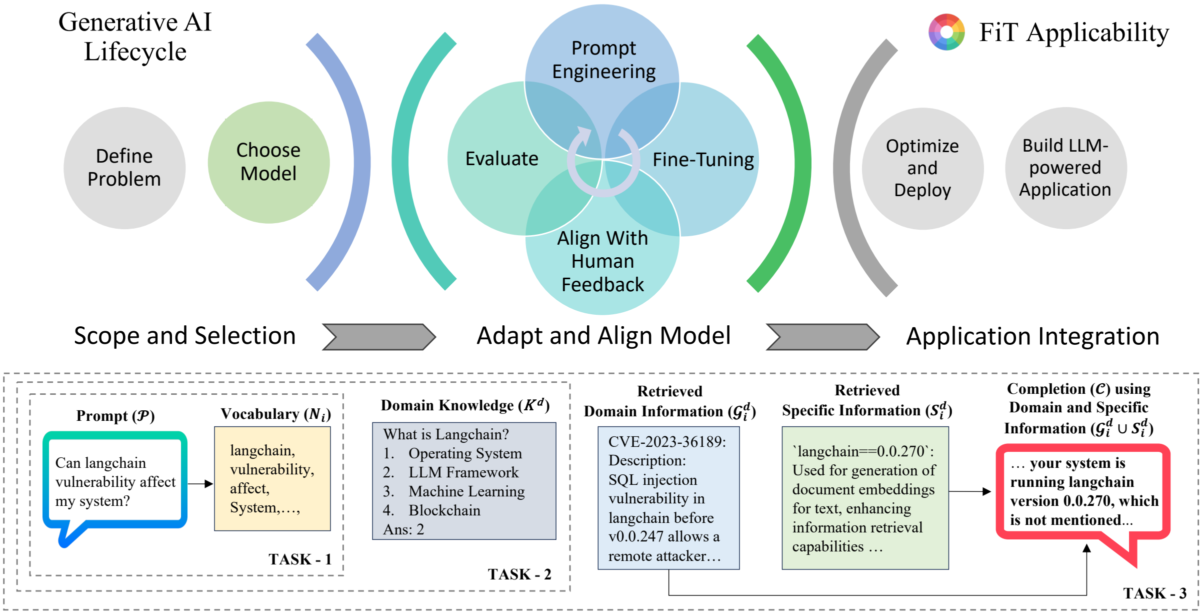
This research presents an innovative automated framework that dynamically pairs Retrieval-Augmented Generation (RAG) with various pre-trained and fine-tuned Large Language Models (LLMs) to enhance their effectiveness in cybersecurity applications. RAG, initially introduced to leverage external knowledge for Languade Models and LLMs, proves insufficient on its own in domains requiring acute precision, like cybersecurity, where it's crucial to distinguish between relevant and irrelevant information. Our framework addresses this gap by evaluating and matching the most appropriate LLM to RAG based on specific cybersecurity tasks. This not only facilitates the provision of contextually accurate and pertinent information but also streamlines the analytical process, significantly saving time for cybersecurity analysts and improving their capability to identify and respond to security threats efficiently. Our findings suggest that instruction tuning causes a knowledge drop, fine-tuning may worsen hallucination in the cybersecurity domain, and the evaluation tasks in our framework are able to predict the post fine-tuning behavior of LLMs.

1 Introduction

Large Language Models (LLMs) have recently delivered a groundbreaking advancement in Natural Language Processing (NLP), with unprecedented language understanding and generation capabilities. Due to this ability, organizations across various industries are adopting LLMs as their primary domain-specific Question-Answering (QA) model to serve numerous business needs. In domains such as cybersecurity, which is often tied to critical infrastructures and national security, incorrect responses could trigger cyber-attacks and breaches, putting the general population at risk. For LLMs to effectively grasp context in the cybersecurity domain, the model must be versed in cybersecurity-related concepts, knowledge, and specific tasks. Although fine-tuning approaches can acquaint LLMs with cybersecurity concepts, they also risk making the model fragile. This has led to problems in real-world applications of LLMs in critical domains (Dahl et al., 2024). The cybersecurity field is abundant with rich knowledge but suffers from a scarcity of labeled data necessary for supervised or semi-supervised fine-tuning. Also, timely and relevant responses are of utmost importance in cybersecurity because cybersecurity information is continuously evolving. Therefore, a method known as Retrieval Augmented Generation (RAG) is utilized, wherein a large language model is coupled with a retrieval system. This system supplies the LLM with current and semantically relevant information, enabling it to produce a meaningful response (Lewis et al., 2020).

For example, a question in a cybersecurity chatbot application can be "*Can langchain vulnerability affect my system?*" In this example, the LLM will be required to understand *What is langchain and its vulnerability?* and *What are the system specifications the user is referring to?*, which are different types of information, but required to generate an accurate answer. Irrespective of the domain of employment, the fundamentals remain the same for a typical knowledge-intensive QA task.

The promising pairing of RAG-inspired LLMs for cybersecurity becomes challenging when one is asked with questions like "Which LLM to pair with RAG?", "Which RAG-LLM will be effective?" "If fine-tuned LLM is better or pre-trained for pairing with RAG", and others. Answering each of the questions requires substantial computing resources and time. Uniquely to prior efforts in cybersecurity, such as CYBERBENCH (Liu et al.), we introduce a novel framework, FiT, to assess LLM's domain-specific understanding and contextualization ability for knowledge-intensive language tasks in cybersecurity sphere. With FiT evaluations, organizations can easily identify the best-suited LLM for its domain-specific QA task and only fine-tune the



Can langchain vulnerability affect my system?

langchain, vulnerability, affect, System,…,

What is Langchain?
1. Operating System
2. LLM Framework
3. Machine Learning
4. Blockchain
Ans: 2

CVE-2023-36189: Description: SQL injection vulnerability in langchain before v0.0.247 allows a remote attacker…

`langchain==0.0.270`: Used for generation of document embeddings for text, enhancing information retrieval capabilities …

… your system is running langchain version 0.0.270, which is not mentioned...

Figure 1: Implementation scope of FiT benchmark in Generative AI life-cycle (colored area implies applicability) with an example of our three evaluation tasks. By aligning the tasks in a complete process, we visualize the propagation of the domain suitability required to generate Completion ($\mathcal{C}$) for a given Prompt ($\mathcal{P}$).

ones that are likely to perform superior to others, reducing computation cost and deployment time. FiT evaluates through three simple aspects (*Vocabulary, Knowledge, Contextualization*). We demonstrate our research experiment[1] in line with the domain of cybersecurity to demonstrate the robustness of FiT framework. Our contributions are the following.

- We develop a benchmarking framework to assess small-scale LLMs for domain-specific knowledge-intensive QA tasks.
- We showcase FiT evaluation tasks and how they can be utilized to predict post-fine-tuning behavior.
- We demonstrate FiT's effectiveness in critical domains, such as cybersecurity.

The rest of the paper is organized as follows: In Section 2, we formulate the research problem. Section 3 provides the background and related works. Advancing, Section 4 offers a detailed discussion of our FiT evaluations. Then, we present our experiment and findings in Section 5 and 6. Concluding remarks are provided at the end.

2 Objective & Problem Formulation

In this section, we explain the problem definition of our benchmarking framework and its foundations. We begin with defining the implementation scope and then delve into our evaluation tasks. We also provide an example reference of our research scope and evaluation tasks in Fig.[1] for visualization.

Notation	Description
$\mathcal{P}$	User Input Prompt
$\{\mathcal{N}^d \in \mathcal{N}\}$	Domain-specific Vocabulary
$\{\mathcal{K}^d \in \mathcal{K}\}$	Domain-specific Knowledge
$\{\mathcal{G}_i^d \mid \mathcal{G}_i^d \in \mathcal{G}^d\}$	Domain Information for $\mathcal{P}$
$\{\mathcal{S}_i^d \mid \mathcal{S}_i^d \in \mathcal{S}^d\}$	Specific Information for $\mathcal{G}_i^d \cup \mathcal{P}$
$\mathcal{C}$	Completion for $\mathcal{P}$ given $(\mathcal{G}_i^d \cup \mathcal{S}_i^d) \mid \mathcal{K}_d$

Table 1: Description of Notations.

In a typical knowledge-intensive and critical-domain QA task using LLM and RAG, the objective is to generate relevant completion ($\mathcal{C}$) for a given prompt ($\mathcal{P}$), without disclosing sensitive information. We observe, irrespective of the employing domain, primarily two types of information are required to generate $\mathcal{C}$ for a given $\mathcal{P}$. One is domain-specific information ($\mathcal{G}_i^d$) relevant to $\mathcal{P}$, and another is contextual or specific information ($\mathcal{S}_i^d$) that contains specific information required to contextualize $\mathcal{G}_i^d$ for $\mathcal{P}$. Then, utilizing LLM's domain-specific vocabulary ($\mathcal{N}^d$) and knowledge ($\mathcal{K}^d$) final $\mathcal{C}$ is generated. To formulate an evaluation strategy and assess LLM's contextualization abilities for the implementation scope, we define our evaluation in a process-oriented approach and categorize it into three tasks addressing different

[1] GitHub: github.com/XXXX

aspects. The following are the defined tasks:

1. We need to assess LLM's familiarity with domain vocabulary. We can determine through keyword recognition task by instructing LLM to identify important keywords ($\mathcal{N}_i^d$) in $\mathcal{P}$.

2. We need to assess if the LLM possesses domain-specific knowledge. To achieve this, we consider a multiple-choice and general QA task related to the domain to assess LLM's domain-specific understanding ($\mathcal{K}^d$).

3. We need to evaluate whether the LLM has contextualization capabilities for the task scope to comprehend and tailor $\mathcal{G}_i^d$ upon $\mathcal{S}_i^d$ to generate $\mathcal{C}$ for given $\mathcal{P}$, without leaking unnecessary information.

By this approach, we can assess an LLM's suitability for the critical-domain QA task from a security and relevancy standpoint. Furthermore, we analyze the models' behavior post fine-tuning to identify patterns in behavioral changes. This identification will provide an in-depth analysis of model behavior to forecast the suitability of an LLM to be fine-tuned for a domain and task, helping users in curating fine-tuning data aligned with the behavior and objectives to attain maximum outcomes.

3 Background and Related Work

The application of pre-trained LLMs in specialized domains has been an active area of research in recent times (Ranade et al., 2021). In this section, we briefly look into the pre-requisite background and related developments in the context of our benchmarking approach.

3.1 LLM, RAG, and Fine-tuning

Large Language Models (LLMs) have revolutionized Natural Language Processing (NLP) thanks to transformer architectures (Vaswani et al., 2017) that offer remarkable parallelization (Min et al., 2023). These models are pre-trained on massive crawls of Internet text with large parameter sizes and have exceptional learning capabilities. Despite their superiority in learning intricate language patterns and structures, these models may occasionally produce seemingly plausible yet inaccurate predictions and face challenges when addressing problems that require specialized domain knowledge. There are many possible reasons (Wang et al., 2023) for the failure of general-purpose LLMs to answer factual questions accurately in a closed domain, such as a *deficit in domain knowledge* (e.g., a language model may lack comprehensive expertise in a specific domain to which it has not been exposed), *outdated information* (e.g., LLMs may have a cutoff date set by their training data, so any developments post-training won't be known to the model without external input), and *forgetting* (e.g., language models may experience catastrophic forgetting (Kirkpatrick et al., 2017) during additional training, where they lose prior knowledge gained before fine-tuning).

To mitigate the knowledge deficiency within pre-trained LLMs for domain-specific tasks, an additional knowledge ingestion step is required. The two most common approaches currently practiced for external knowledge ingestion are Retrieval Augmented Generation (RAG) and Fine-tuning. The first approach, introduced around mid-2020 by Lewis et al. (2020), aims to enhance the capabilities of LLMs for knowledge-intensive tasks. The core idea is to leverage external knowledge sources to overcome the knowledge deficiency limitations of pre-trained LLMs. The process works by providing the model with an auxiliary knowledge base, which could be a corpus of relevant documents, a structured database, or any other source of domain-specific information. When presented with an input query, the RAG architecture then searches through this knowledge base to identify the most relevant documents or passages. These retrieved information sources are then seamlessly integrated into the input, providing the LLM with additional context and background related to the query.

As language models continue to grow in size and complexity, updating all of their parameters becomes an increasingly demanding computational task, making it inefficient and cost-prohibitive. This presents a substantial challenge when attempting to *finetune* these large language models for specific downstream applications, especially in scenarios where the available hardware infrastructure and computational resources are limited. This has led the research in exploring parameter efficient tuning methods that aims to attain optimal performance for specific tasks while minimizing the number of tunable parameters. Some efforts in this direction that mainly focus on developing efficient tunable modules for LLMs include *adapters based (Houlsby et al., 2019), prompt based (Lester et al., 2021), LoRA(Valipour et al.,*

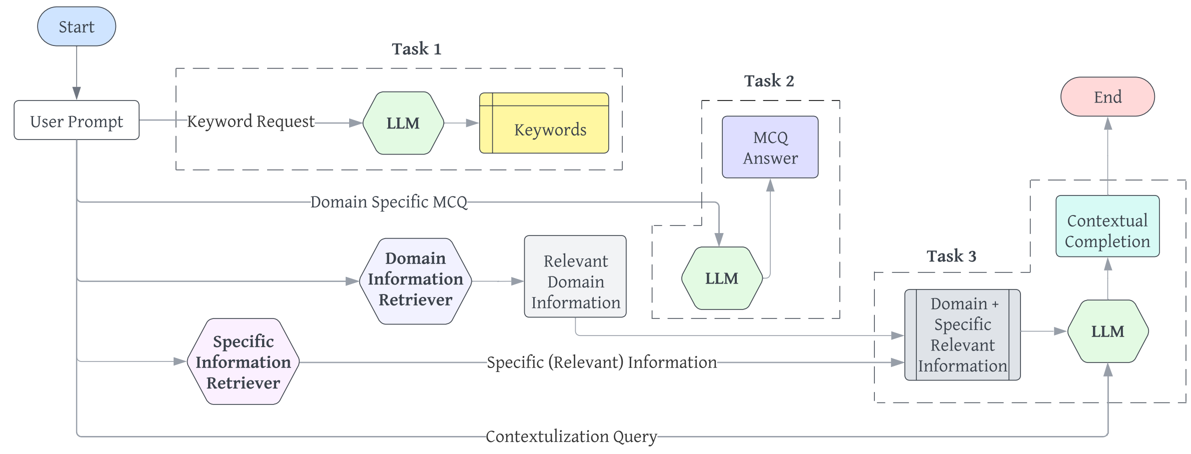

Start
Task 1
User Prompt
Keyword Request
LLM
Keywords
Task 2
MCQ Answer
End
Domain Specific MCQ
Contextual Completion
Domain Information Retriever
Relevant Domain Information
LLM
Task 3
Domain + Specific Relevant Information
LLM
Specific Information Retriever
Specific (Relevant) Information
Contextulization Query

Figure 2: Flowchart of FiT benchmark evaluation process. The process depicts the three evaluation tasks in line with the deployment scope. Task 1 is the vocabulary assessment, Task 2 is for domain-specific knowledge analysis, and finally, Task 3 assesses the LLM's contextualization ability for relevant and secured generation.

2022), QLoRA(Dettmers et al., 2024), and hybrid (Mao et al., 2021) among others.

3.2 LLM Benchmarking

Numerous benchmarking frameworks have been developed to evaluate the performance of both general and domain-specific language models across a diverse array of NLP tasks. Some benchmarks, such as GLUE-X (Yang et al., 2022) and Prompt-Bench (Zhu et al., 2023), assess the capabilities of language models for general task, including robustness to *out-of-distribution* and *adversarial inputs*, while others like KOLA (Yu et al., 2023) evaluate their knowledge and reasoning abilities. In addition to general task benchmarks, there are benchmarks tailored for specific domains. For instance, MultiMedQA is a benchmark for medical question-answering focused on medical exams, research, and consumer healthcare. MATH (Singhal et al., 2023), on the other hand, evaluates AI models' reasoning and problem-solving skills in mathematics. Similarly, there are multi-task benchmarks in the cybersecurity domain, such as CyberBench (Liu et al.), and SecureBert (Aghaei et al., 2022) for sentiment analysis and NER.

4 FiT Benchmark

In this section, we explain our FiT benchmark and its evaluation tasks to assess LLM's suitability for a domain-specific QA task through RAG. We describe each evaluation task as exemplified in Fig.[2], for visualization.

4.1 Task 1: Vocabulary Assessment

To evaluate a language model for a domain-specific task, ensuring the model has a basic understanding of its vocabulary is paramount. This linguistic understanding allows a language model to comprehend the given input and communicate using similar vocabulary. Therefore, to assess this ability, we consider *Keyword Recognition (KR)* task. KR is an NLP task that involves identifying important entities in an unstructured text. To achieve this, the LLM is instructed to identify the keywords in the input prompt ($\mathcal{P}$). With this evaluation, we can assess two aspects of LLM's overall accuracy. The first assessment involves the number of correct keyword identifications. We determine the model's domain-oriented linguistic understanding as vocabulary drastically differs depending on the domain and scope of implementation. For instance, a medical QA vocabulary differs from an e-commerce or cybersecurity vocabulary. Second, understanding keywords specific to instruction implies their proportional relation to understanding the given task. Hence, mathematically the task can be defined as, let $\mathcal{P} = \{\mathcal{P}_i\}_{i=1}^{n}$ be the set of prompts, d be the domain, and $\mathcal{N}^d$ be the set of vocabulary with respect to domain d. There exists $\mathcal{N}_i^d$ for $\mathcal{P}_i$, where $\overline{\mathcal{N}_i^d}$ is the expected output. Hence, the LLM's vocabulary assessment score $\Phi(\mathcal{N}^d|\mathcal{P})$ can be calculated as,

$$\Phi(\mathcal{N}^d|\mathcal{P}) = \frac{\sum_{i=1}^{n} \Delta\{\mathcal{L}(\mathcal{P}_i)|\overline{\mathcal{N}_i^d}\}}{n} \tag{1}$$

where Δ is the F_1 function upon predicted and expected response, $\mathcal{L}$ is the model, and $n > 0$.

Task-1: Vocabulary Assessment ($\mathcal{N}^d$)

Instruction: Print the keywords from the following ...
$\mathcal{P}_i$: Can langchain vulnerability affect my system?
$\mathcal{N}_i^d$: langchain, vulnerability, affect, system

4.2 Task 2: Knowledge Analysis

Defining knowledge is a philosophical question, far beyond our research's scope. However, we can conduct a passive qualitative analysis of an LLM's reasoning capabilities to quantify its knowledge regarding a certain context or domain. For example, if an LLM can comprehend the context of the question, it can generate a relevant answer through its reasoning capabilities. Hence, we consider multiple-choice question-answering (MCQ) tasks for analyzing the models' domain-specific knowledge. We can then compare the output with human-generated ones to obtain quantifying measures. Finally, we extend these quantifying metrics to assess the whole knowledge base of an LLM regarding a specific domain. Formally, let $\mathcal{P} = \{\mathcal{P}_i\}_{i=1}^n$ be a set of MCQ problems, $\alpha = \{\alpha_i^1 ... \alpha_i^m\}_{i=1}^n$ be the set of all answers, with each $\mathcal{P}_i$ having m possible answers and γ_i be the set of correct answers. Hence, knowledge score $\Phi(K^d|\mathcal{P})$ can be calculated as,

$$\Phi(K^d|\mathcal{P}) = \frac{\sum_{i=1}^n \{(\alpha_i^j = \gamma_i|\mathcal{P}_i)\}_{j=1}^m}{n} \tag{2}$$

Task-2: Knowledge Analysis ($\mathcal{K}^d$)

Instruction: Answer the correct choice for the question ...
$\mathcal{P}_i$: A hash function guarantees the integrity of a message. It guarantees that the message has not be
1: Replaced
2: Overview
3: Changed
4: Violated

γ_i: (3) Changed

4.3 Task 3: Contextualization Analysis

Contextualization refers to understanding and generating text based on nuances and relationships between multi-faceted information. For example, a question can be *What potential impact could the CVE-2023-3894 vulnerability have on the integrity of our TOML configuration data?* In this case, the LLM will require the multi-faceted information of *CVE-2023-3894* and referring *TOML server*. For knowledge-intensive QA tasks, RAG is employed as a bridge to overcome the LLMs' knowledge deficiency. Contextualization ability allows an LLM to generate relevant and reliable answers from this additional retrieved multi-faceted information by following instructions. Assessing this ability is critical, since in numerous scenarios such as recommendation, privacy, etc., the domain information must be tailored in completion depending on the specifics. Without accurate information contextualization, the LLM is prone to leak sensitive information or provide misinformation. Therefore, by this task, we can evaluate a model's complete understanding of the problem context, provided information, and reliability to generate a response concerning additional factors. To assess this ability to perform knowledge-intensive and domain-specific QA tasks, we consider contextualized RAG (Greshake et al., 2023) as our final evaluation task. Comparing the generated response with a Subject Matter Experts (SMEs) ground truth, through correctness and similarity metric, we can assess the models contextualization and data security. We express this mathematically as follows. Let $\mathcal{P} = \{\mathcal{P}_i\}_{i=1}^n$ be a set of knowledge-intensive questions related to the application domain, $\mathcal{G}_i^d$ be the domain, and $\mathcal{S}_i^d$ be the specific information, for $\mathcal{P}_i$, and $\overline{\mathcal{C}_i}$ be the expected answer. Therefore, we can calculate the contextualization score $\Phi(\mathcal{C}|\mathcal{P})$ between $\mathcal{C}_i$ and $\overline{\mathcal{C}_i}$ by,

$$\Phi(\mathcal{C}|\mathcal{P}) = \frac{\sum_{i=1}^n \Omega\{\mathcal{L}(\mathcal{P}_i|\mathcal{G}_i^d \cup \mathcal{S}_i^d)|\overline{\mathcal{C}_i}\}}{n} \tag{3}$$

where Ω is the contextualization score calculation function, $\mathcal{L}$ is the model, and $n > 0$

Task-3: Contextualization ($\mathcal{C}$)

Instructions:
Given the following retrieved knowledge, answer ...
$\mathcal{G}_i^d$**:**
CVE ID: CVE-2023-36189
Description: SQL injection vulnerability in langchain before v0.0.247 allows a remote attacker to obtain ...
CVE ID: CVE-2023-36188
Description: An issue in langchain v.0.0.64 allows a remote attacker to execute arbitrary code via the PALChain ...
$\mathcal{S}_i^d$**:**
‘langchain==0.0.270‘: Used for generation of document embeddings for text, enhancing information retrieval capabilities ...
$\mathcal{P}_i$: Can langchain vulnerability affect my system?
$\mathcal{C}_i$: Langchain has multiple reported vulnerabilities ... your system is running langchain version 0.0.270, which is not mentioned to be vulnerable in the provided information...

5 Experiment & Evaluation

In this section, we discuss the experiment dataset description, and evaluations. First, we start by describing our experimental data with relevance to our three evaluation tasks. Then we delve deep into discussing the quantitative and qualitative evaluations, followed by fine-tuning details. For the scope of the research, cybersecurity itself being an information critical domain, we have considered it in our experiments.

5.1 Data description and Preparation

In our evaluation, we perform three separate tasks. For task 1, we curated sample questions relevant to cybersecurity and their corresponding keywords. For task 2, we considered computer security MCQ questions from MMLU dataset (CAIS, 2024). For task 3, our evaluation dataset is divided into two parts: a cybersecurity information repository ($\mathcal{G}^d$) and a QA specific information repository ($\mathcal{S}^d$). In our case we considered NIST (NIST, 2024) as our domain information repository and we curated organization-specific infrastructure wiki as our specific information repository. Since this information is often sensitive, we supplanted this with some synthetic data. We then curated questions that require both information to generate a relevant response alongside ground-truths. In our evaluations, we performed two types of fine-tuning. One is knowledge-focused (Finetuned-1) dataset and another is instruction-focused (Finetuned-2) dataset. For knowledge-focused dataset we retrieved Cisco Talos (Cisco, 2024) dataset and generated QA pairs, and for instruction biased dataset we created a training split from our evaluation dataset. All the curated dataset and fine-tuned models will be disclosed at appropriate locations.

5.2 Experiment Infrastructure

For our evaluations we considered 5 open-sourced 7-billion parameter 4-bit quantize QA purpose LLMs, namely [*Llama-2-7b*[2], *Mistral-7b*[3], *Prometheus-7b*[4], *WestLake-7b*[5], *and WestSeverus-7b*[6]]. Additionally, to compare the model performance relative to one of the current state-of-the-art, we have considered *GPT-3.5-Turbo* [7]. Furthermore, for specific information retrieval, we have implemented *ChromaDB*[8] as our vector storage. Our experiment was performed over *Intel i9-12900 with GeForce RTX™ 3090Ti* and 128 GB of RAM.

5.3 FiT Evaluation

We employ two evaluation criteria to measure the effectiveness of our approach including qualitative and quantitative. The qualitative evaluation is conducted by leveraging Subject Matter Experts (SMEs) to judge the generated responses across three tasks whereas the quantitative evaluation is conducted by leveraging Ragas (Es et al., 2023) framework. We provide the evaluation results in Table 2 and 3.

5.3.1 Quantitative Evaluation

For the quantitative evaluations, our focus is to quantify the three evaluation tasks. For task 1 (KR), we compute the F1 score between the prediction and ground truths. Task 2 (MCQ) is a binary classification problem. Hence, we compute the accuracy of correct classifications. For task 3 (Contextualization), we consider similarity and correctness metric from Ragas framework. Through similarity metric, we can assess information leakage, and correctness metric assess the relevant contextualization. We opted for this framework because other popular frameworks such as BLEU (Papineni et al., 2002) and ROUGE (Rouge, 2004) are primarily tailored for evaluating machine translation tasks, and text summarization tasks.

5.3.2 Qualitative Evaluation

For qualitative evaluation, we engaged two cybersecurity SMEs to assess FiT's contextual response generation concerning pre-trained models. They evaluated question-answer pairs for factual correctness and contextual relevancy using a 5-point Likert scale (Allen and Seaman, 2007), ranging from 1 (indicating "Factually Incorrect and Contextually irrelevant") to 5 (indicating "Factually Accurate and Contextually relevant"). Inter-rater agreement was analyzed using the Fleiss Kappa measure (McHugh, 2012) as depicted in Table 3, showing strong agreement for most models (gpt-3.5-turbo at 0.861, llma2-7b at 0.845, prometheus-7b at 0.864, westlake-7b at 0.944, and westseverus-7b at 0.868), though moderate for mistral-7b at 0.782.

5.4 Fine-tuning

To analyze the model behavior post fine-tuning we performed QLoRA (Dettmers et al., 2024), a PEFT (Ding et al., 2023) fine-tuning technique over

[2] huggingface.co/meta-llama/Llama-2-70b-chat-hf
[3] huggingface.co/mistralai/Mistral-7B-Instruct-v0.2
[4] huggingface.co/AiMavenAi/AiMaven-Prometheus
[5] huggingface.co/senseable/WestLake-7B-v2
[6] huggingface.co/FelixChao/WestSeverus-7B-DPO-v2
[7] platform.openai.com/docs/models/gpt-3-5-turbo
[8] trychroma.com

Table 2: FiT evaluation results for pre-trained, knowledge-focused, and instruction focused fine-tuning.

Model	Pretrained				Finetuned-1				Finetuned-2			
	Task 1	Task 2	Task 3		Task 1	Task 2	Task 3		Task 1	Task 2	Task 3	
	F1	Score	Sim	Cor	F1	Score	Sim	Cor	F1	Score	Sim	Cor
gpt-3.5-turbo	0.85	0.76	0.92	0.77	–	–	–	–	–	–	–	–
llama2-7b	0.62	0.51	0.91	0.78	0.35	0.32	0.87	0.75	0.48	0.31	0.92	0.79
mistral-7b	0.47	0.59	0.90	0.72	0.27	0.39	0.86	0.74	0.43	0.18	0.86	0.76
prometheus-7b	0.76	0.75	0.92	0.73	0.65	0.61	0.85	0.73	0.55	0.16	0.86	0.75
westlake-7b	0.77	0.71	0.92	0.74	0.59	0.65	0.85	0.73	0.73	0.08	0.93	0.79
westseverus-7b	0.74	0.72	0.89	0.69	0.69	0.66	0.89	0.76	0.73	0.12	0.91	0.78

Table 3: Fleiss Multirater Kappa Analysis

Model	Kappa (K)	Standard Error
gpt-3.5-turbo	0.861	0.080
llama2-7b	0.845	0.084
mistral-7b	0.782	0.082
prometheus-7b	0.864	0.077
westlake-7b	0.944	0.081
westseverus-7b	0.868	0.078

the open-source models with knowledge-focused and instruction-focused datasets. We kept all other hyper-parameters such as rank (64), batchsize (4), epochs (5), etc, constant for both fine-tuning.

6 Findings and Limitations

In domains like cybersecurity, where knowledge is dynamic, fine-tuning with knowledge-focused dataset, as evidenced in *Finetuned-1: Task 1*, is not helpful because prompts often contain keywords and context unknown to the model. Hence, we observe a decrement in *Task 2* equally. Furthermore, knowledge-focused fine-tuning led to more hallucination and less instruction following (Task-3-Similarity). From a practical standpoint, in a dynamic domain, if we lack relevant data, using pre-trained models is a better approach. Conversely, instruction-focused tuning led to significant knowledge drop. The significant knowledge drop in *Task 2* in *Finetuned-2* can be attributed to the rigorous instruction tuning, where we deliberately instructed the model not to provide an answer if it is unsure about the result. The model thus became conservative in its output generation throughout. This finding follows abstention, the intuition presented in Xin et al. (2021). Additionally, different LLMs learning mechanism directly impacts post-fine tuning performance. Mistral being one of the prominent LLMs performed relatively poor than others. It might be referred to its sliding window attention mechanism of learning. In our pre-trained evaluation, Wesklake and Westseverus yielded superior results in cybersecurity knowledge and contextualization across the three tasks. After fine-tuning, we observed the same pattern, further emphasizing the utility of our three different tasks in predicting post-fine-tuning performance.

Apart from the findings, further experiment over more number of models will deliver more insights. We only chose 7-billion models which are more popular to adhere the scope of our research. Previous studies have suggests that even with small-size LLMs, it it possible to achieve equivalent performance of 70-biilion models in domain-specific tasks (Liu et al.). Furthermore, our benchmark only focuses on the contextualization and information security aspect. We did not consider other evaluation aspects such as toxicity, truthfulness, etc. in our evaluation and behaviour analysis. Finally, due to the confidentiality reasons evaluation was conducted over synthetic data.

Conclusion

Cybersecurity is tied to critical infrastructure underscoring the importance of investigating LLMs in this domain, as they can potentially lead to significant consequences if sensitive information is compromised. Our research focuses on assessing the suitability of LLMs paired with RAG for knowledge-intensive QA tasks from contextualization and information security standpoint. To do so, we develop a novel benchmark that assess LLMs' domain understanding and forecast post-fine-tuning behavior through three tasks. According to our observation, instruction focused tuning reduces knowledge and knowledge focused tuning reduces instruction following behavior. Hence, in critical domains where data is dynamic, users can benefit from pre-trained models to strike the perfect balance between knowledge and instruction following.

Ethics Statement

Our research is based on a dataset that does not contain any sensitive information. To obtain cyber threat intelligence, specifically Common Vulnerabilities and Exposures (CVEs), we use web crawlers that make API calls, strictly following the limitations specified by authorized sources. We prioritized the privacy of our human evaluators by thoroughly anonymizing their identities, ensuring that no personally identifiable information is accidentally disclosed. Furthermore, we confirm that our research is in line with the ethical standards stated in the ACL Ethics policy to the best of our knowledge.

References

Ehsan Aghaei, Xi Niu, Waseem Shadid, and Ehab Al-Shaer. 2022. Securebert: A domain-specific language model for cybersecurity. In *International Conference on Security and Privacy in Communication Systems*, pages 39–56. Springer.

I Elaine Allen and Christopher A Seaman. 2007. Likert scales and data analyses. *Quality progress*, 40(7):64–65.

CAIS. 2024. Measuring massive multitask language understanding. huggingface.co/datasets/cais/mmlu.

Cisco. 2024. National vulnerability database. talosintelligence.com.

Matthew Dahl, Varun Magesh, Mirac Suzgun, and Daniel E. Ho. 2024. Hallucinating law: Legal mistakes with large language models are pervasive. https://hai.stanford.edu/news/hallucinating-law-legal-mistakes-large-language-models-are-pervasive.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2023. Ragas: Automated evaluation of retrieval augmented generation. *arXiv preprint arXiv:2309.15217*.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. 2023. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, pages 79–90.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Zefang Liu, Jialei Shi, and John F Buford. Cyberbench: A multi-task benchmark for evaluating large language models in cybersecurity.

Yuning Mao, Lambert Mathias, Rui Hou, Amjad Almahairi, Hao Ma, Jiawei Han, Wen-tau Yih, and Madian Khabsa. 2021. Unipelt: A unified framework for parameter-efficient language model tuning. *arXiv preprint arXiv:2110.07577*.

Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. 2023. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*, 56(2):1–40.

NIST. 2024. National vulnerability database. nist.gov.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Priyanka Ranade, Aritran Piplai, Sudip Mittal, Anupam Joshi, and Tim Finin. 2021. Generating fake cyber threat intelligence using transformer-based models. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE.

Lin CY Rouge. 2004. A package for automatic evaluation of summaries. In *Proceedings of Workshop on Text Summarization of ACL, Spain*, volume 5.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. 2023. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180.

Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. 2022. Dylora: Parameter efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. *arXiv preprint arXiv:2210.07558*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, et al. 2023. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity. *arXiv preprint arXiv:2310.07521*.

Ji Xin, Raphael Tang, Yaoliang Yu, and Jimmy Lin. 2021. The art of abstention: Selective prediction and error regularization for natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1040–1051.

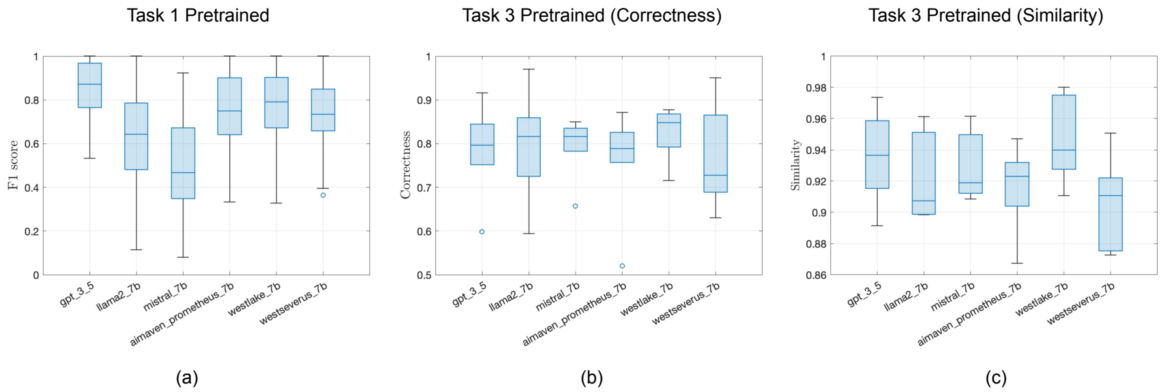
Linyi Yang, Shuibai Zhang, Libo Qin, Yafu Li, Yidong Wang, Hanmeng Liu, Jindong Wang, Xing Xie, and Yue Zhang. 2022. Glue-x: Evaluating natural language understanding models from an out-of-distribution generalization perspective. *arXiv preprint arXiv:2211.08073*.

Jifan Yu, Xiaozhi Wang, Shangqing Tu, Shulin Cao, Daniel Zhang-Li, Xin Lv, Hao Peng, Zijun Yao, Xiaohan Zhang, Hanming Li, et al. 2023. Kola: Carefully benchmarking world knowledge of large language models. *arXiv preprint arXiv:2306.09296*.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Neil Zhenqiang Gong, Yue Zhang, et al. 2023. Promptbench: Towards evaluating the robustness of large language models on adversarial prompts. *arXiv preprint arXiv:2306.04528*.

A Appendix

This section contains supplementary materials, including visualization figures for Task 1 and Task 3, for pretrained, Finetuned-1, and Finetuned-2 models, which were omitted from the main paper due to space limitations.


Task 1 Pretrained

Task 3 Pretrained (Correctness)

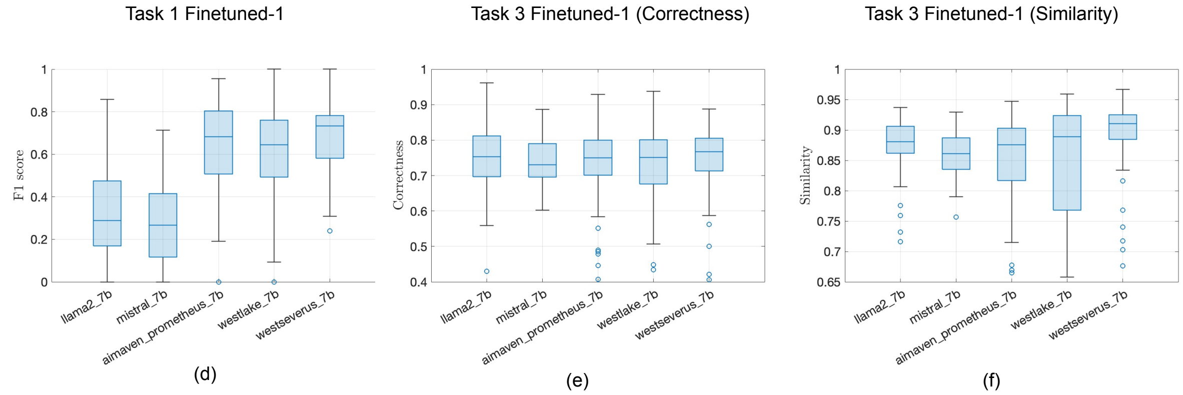
Task 3 Pretrained (Similarity)

(a)

(b)

(c)

Figure 3: Graphical depiction of performance of pre-trained models on Task 1 and Task 3. Figure (a) represents F1 scores on vocabulary assessment tasks. Figure (b) represents performance of the models on contextual response completion tasks in terms of correctness of completion, while Figure (c) represents similarity scores of completions.


Task 1 Finetuned-1

Task 3 Finetuned-1 (Correctness)

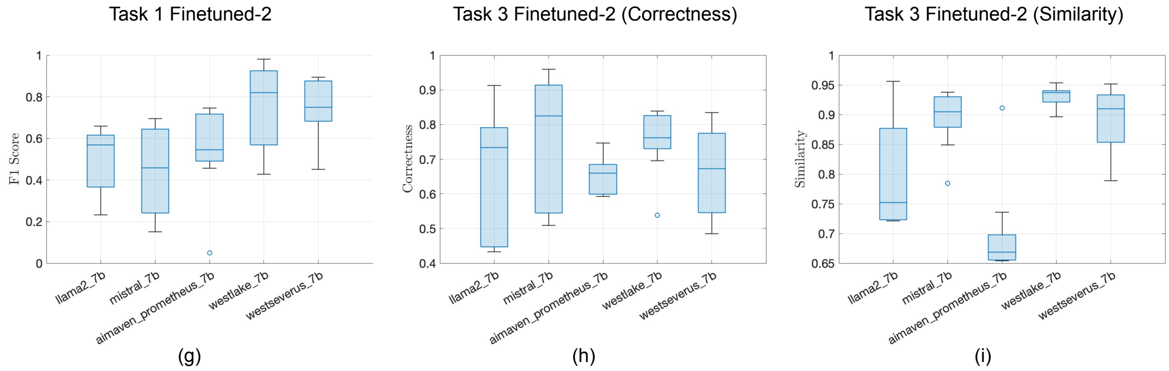
Task 3 Finetuned-1 (Similarity)

(d)

(e)

(f)

Figure 4: Graphical depiction of performance of Finetuned-1 models on Task 1 and Task 3. Figure (a) represents F1 scores on vocabulary assessment tasks. Figure (b) represents performance of the models on contextual response completion tasks in terms of correctness of completion, while Figure (c) represents similarity scores of completions.


Task 1 Finetuned-2

Task 3 Finetuned-2 (Correctness)

Task 3 Finetuned-2 (Similarity)

(g)

(h)

(i)

Figure 5: Graphical depiction of performance of Finetuned-2 models on Task 1 and Task 3. Figure (a) represents F1 scores on vocabulary assessment tasks. Figure (b) represents performance of the models on contextual response completion tasks in terms of correctness of completion, while Figure (c) represents similarity scores of completions.