RAG-ATT&CK : Exploring RAG-Assisted Mapping of Cyber Threat Intelligence to MITRE ATT&CK Techniques

Stijn Schuurman
EEMCS
University of Twente
Enschede, Netherlands

***Abstract*—Mapping unstructured Cyber Threat Intelligence (CTI) to MITRE ATT&CK techniques is essential for understanding and mitigating future cybersecurity threats. Existing automated methods require extensive fine-tuning of large language models (LLMs) or require static rules, limiting their adaptiveness to an evolving threat landscape. This work introduces RAG-ATT&CK , an automated mapping system utilizing Retrieval-Augmented-Generation (RAG). RAG-ATT&CK dynamically retrieves relevant MITRE ATT&CK techniques, providing the underlying LLM with factual context for classification, without the need for fine-tuning. While RAG-ATT&CK shows improvements over the baseline LLM system, it does not surpass the state-of-the-art methods. This study highlights the potential of RAG-based systems and offers a comparison to fine-tuning-based systems.**

***Index Terms*—CTI Mapping, MITRE ATT&CK , LLM, RAG, Machine Learning, Embeddings, Transformers**

1. Introduction

In the dynamic field of cybersecurity, the MITRE ATT&CK framework[1] serves as an essential tool to identify and categorize adversarial behaviors. By offering a structured classification of tactics, techniques, and sub-techniques, it provides a unified language that enhances the analysis and response to cyber threats. However, much threat information (CTI) remains unstructured, often presented in natural language, and lacks alignment with established standards like the MITRE ATT&CK framework. Security operators often rely on descriptive text that outlines the tactics, techniques, and procedures (TTPs) employed by threat actors, complicating their ability to effectively identify, prioritize, and respond to threats.

Mapping unstructured CTI to the MITRE ATT&CK framework is a complex and labor-intensive process, requiring well-trained experts with deep domain knowledge. This manual mapping is guided by guidelines[2] from organizations such as Cybersecurity & Infrastructure Security Agency (CISA)[3] and MITRE[4], as well as visualization tools like ATT&CK Navigator[5]. Despite these resources, the process remains prone to inconsistencies and biases among experts, leading to variations in the structured information that is critical for effective threat response.

The objective of this research is to improve the process of mapping each sentence of a CTI report to the applicable MITRE ATT&CK techniques listed in Appendix A.

In efforts to automate CTI mapping, techniques have transcended from rule-based NLP language understanding [1]–[9] to the application of static [10]–[12] embeddings provided by the Word2Vec technology from 2013 [11] and attention-based embeddings [8], [13] provided by transformer models [13] to build ML classifiers. Embeddings are numerical representations of texts, where semantically similar items are positioned closer together. The embeddings capture syntactic relationships and semantic meanings, allowing ML models to process and understand language in a nuanced and efficient manner. The latest advancement goes beyond traditional ML methods, focusing on fine-tuning an LLM [14] to predict results more effectively with the Threat Report ATT&CK Mapper (TRAM)[6], a tool developed by MITRE Engenuity. TRAM is designed to facilitate the automated mapping of threat intelligence reports to MITRE ATT&CK techniques. Despite these advancements, the automated mapping of unstructured CTI to ATT&CK techniques remains imperfect. With ongoing developments in generative large language models (LLMs), there is potential to enhance existing knowledge extraction from CTI reports. This research explores how these LLMs can be integrated into improving CTI mapping of reports to the MITRE ATT&CK framework. This is achieved by leveraging LLMs within a retrieval-augmented generation (RAG) framework. By integrating optimization efforts such as knowledge bases and prompt engineering, our objective is to enhance the mapping of unstructured CTI. This paper aims to explore the capability of LLMs, in conjunction with RAG, to classify CTI into MITRE ATT&CK techniques. Our goal is to provide valuable insights and tools that facilitate more effective automated mapping of unstructured CTI, thereby demonstrating the potential of these methodologies in threat intelligence analysis.

A. Research Questions

Building on the potential of an LLM to enhance the mapping of CTI reports to the MITRE ATT&CK framework,

[1]https://github.com/mitre/cti
[2]https://www.cisa.gov/news-events/news/best-practices-mitre-attckr-mapping
[3]https://www.cisa.gov
[4]https://www.mitre.org
[5]https://github.com/mitre-attack/attack-navigator
[6]https://github.com/center-for-threat-informed-defense/tram

our research is structured around three key investigative steps. First, we establish the baseline performance of an LLM in CTI mapping. Second, we assess how integrating an LLM within a RAG framework influences the mapping. Finally, we evaluate the impact of various optimization efforts on improving this process. These steps lead us to explore the following research questions:

1) *What are the baseline precision, recall, and F1-score metrics for an LLM-powered classifying mechanism when applied to mapping ATT&CK techniques in unstructured CTI reports?*
2) *What is the impact of integrating a Retrieval-Augmented Generation system, powered by a Large Language Model, on the precision, recall, and F1-score metrics when mapping ATT&CK techniques in unstructured CTI reports?*
3) *How do the experiments on the Retrieval-Augmented Generation system affect the precision, recall, and F1-score compared to the baseline metrics in the context of ATT&CK technique identification?*

2. Background Information

This research presents an approach to CTI mapping that differs from traditional data-driven methods, which typically rely on learning from a limited data subset, our approach utilizes an LLM with a broad understanding of language.

To effectively map unstructured CTI reports to a framework like MITRE ATT&CK , the LLM requires additional guidance. Techniques such as knowledge bases and prompt engineering can provide context and instructions.

A. Prompt Engineering

Prompt engineering involves adapting the system's interaction with the LLM by creating a targeted instruction prompt aimed at mapping the CTI to the MITRE ATT&CK framework. Prompt engineering is essential for effectively utilizing LLMs, as it defines how the model interprets and responds to input. The initial message, often referred to as system message, guides the LLMs behavior and focus.

At the heart of prompt engineering is the creation of a system message that specifies the task, constraints and desired outcomes, which has various approaches to execute prompt engineering, such as:

- **Zero-Shot Learning**: Giving the AI a task without any prior examples. Describing the question in detail.
- **One-Shot Learning**: Providing a single example along with the prompt. To help the AI understand the context or format that is desired.
- **Few-Shot Learning**: Providing a set of examples to aid the AI to understand the desired response.
- **Chain-of-Thought Prompting**: Instruct the AI to generate a logical sequence of steps leading to the final answer. Breaks down complex tasks into manageable parts.

In the methodology Sections 3-B2 and 3-B3, we will elaborate on the specific prompts used in our experiments, demonstrating how they are structured to optimize the automated mapping of unstructured CTI to the MITRE ATT&CK framework.

B. Knowlege Bases

Knowledge bases are advanced repositories of information that can aid in enhancing the capabilities of LLMs by providing structured, factual content. While they share similarities with traditional databases, which also store entries in a structured format with fields like identifiers, names, and descriptions, knowledge bases stored in a vector database offer distinct advantages.

In a regular database, entries are retrieved based on exact matches or predefined matching queries, which limits their flexibility and depth of the search. A knowledge base exists inside a vector database, which stores entries as embeddings. This allows for the retrieval of entries based on semantic similarity rather than exact matching. The use of embeddings allows the knowledge base to capture complex relationships and meaning within the data. Since the input to the LLM is unstructured natural text, the natural text can be transformed into an embedded form to retrieve semantically relevant entries from the factual knowledge base.

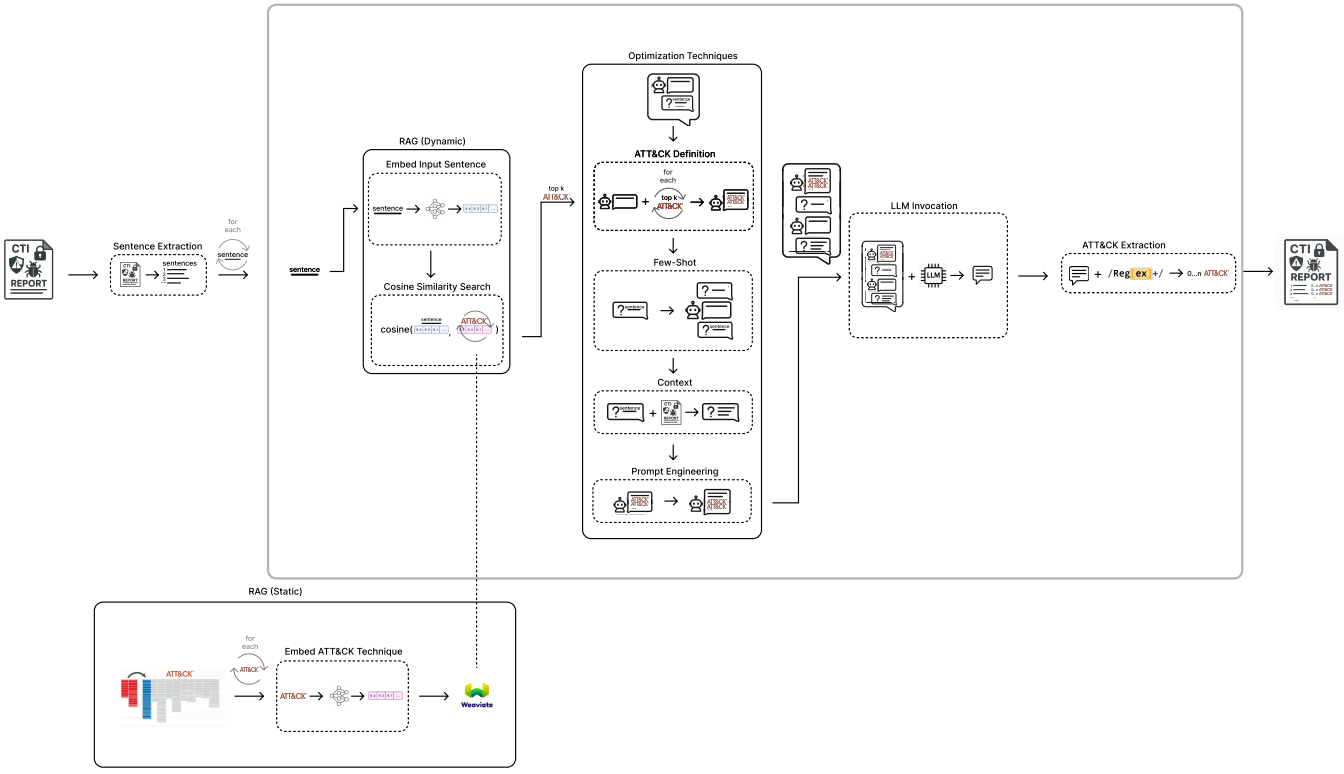
In RAG systems, the knowledge base allows LLMs to access and integrate relevant factual information. Ensuring that the output of the model is not only linguistically correct but also based on accurate and contextually relevant knowledge.

3. Methodology

In this research, we focus on mapping unstructured CTI to MITRE ATT&CK techniques using an LLM as the foundational element of our approach. The steps are illustrated in Figure 1, serving as a visual representation of the system of RAG-ATT&CK . Our objective is to enhance the LLM's capabilities in accurately identifying and classifying ATT&CK techniques from unstructured CTI data. To achieve this, we have implemented a series of optimization efforts that incorporate Retrieval Augmented Generation (RAG) principles. These RAG efforts include embedding model comparison, reranker model application, and various text MITRE ATT&CK technique definition strategies, which dynamically retrieve and integrate additional information into the LLM's input to improve its performance in mapping unstructured CTI to MITRE ATT&CK techniques. A visualization of this process is given in the subfigure RAG (Static) in combination with RAG (Dynamic) in Figure 1.

In our approach, we harness the conversational abilities of a large language model (LLM) to label MITRE ATT&CK techniques within sentences extracted from Cyber Threat Intelligence (CTI) reports. More specifically, for each sentence in the CTI report, we define it as a `user` message and prompt the LLM to complete the dialogue, as illustrated in Figure 1 in subfigure LLM Invocation. This input is then used to generate an `ai` message, providing a response guided by the `system` prompt's instructions detailed in Appendix B. In this research, we aim to enhance the conversational structure of the LLM through few-shot learning, various strategies for context definition, prompt-engineering, and enriched surrounding context of input sentences of the CTI reports. We further investigate how these optimizations impact the model's ability to map unstructured CTI to MITRE ATT&CK techniques,

Fig. 1. RAGATT&CK System Component Overview



as illustrated in Figure 1, specifically in the subfigure titled ”LLM Interaction”.

To provide factual information to the LLM to make decisions on, we utilize RAG principles, which retrieve a subset of the most relevant MITRE ATT&CK techniques to enrich the original `user` message. This approach facilitates a scalable implementation by limiting the amount of ATT&CK technique tags and descriptions that are provided to the LLM. Given the ever-growing number of MITRE ATT&CK techniques—over 650 as of the time of writing—this strategy is essential for maintaining clarity and precision in the model’s responses.

A. Retrieval Augmented Generation (RAG)

The approach of this research is directed towards Retrieval-Augmented Generation (RAG). A method to retrieve factual information from a knowledge base, which is used by the generative LLM to formulate an answer. Refer to Figure 1, subsection titled ”RAG (Static)” & ”RAG (Dynamic)”. In our proposed system, the implementation of RAG involves a systematic process for preselecting relevant MITRE ATT&CK techniques from the knowledge base using an embedding model. This process consists of several key steps designed to ensure effective retrieval and minimize potential model confusion.

The first step involves extracting factual information about the ATT&CK techniques, as defined by the MITRE organization. We utilize the `mitre/cti` project [7] to obtain a comprehensive list of all techniques, including sub-techniques, as defined by MITRE. Next, we employ an embedding model to generate high-dimensional vector representations of these techniques. This model converts words or phrases into dense numerical vectors, capturing the semantic relationships and contextual usage of the terms. By positioning semantically similar items closer together in the embedding space, the model facilitates a data-driven and effective retrieval. The resulting numerical vector representations are stored in a vector database (Weaviate), allowing for efficient querying. Upon LLM invocation, we query the vector database to compare the relevance of the MITRE ATT&CK techniques. Based on this relevance, we can incorporate the tag and definition as defined by MITRE of the selected techniques as additional input to the LLM. This preselection process ensures that the LLM receives factual information relevant to the input, thereby providing a basis for reasoned predictions. By limiting the number of ATT&CK techniques presented to the model, we reduce the likelihood of model confusion [15] compared to supplying the entire set of MITRE ATT&CK techniques.

In our proposed system, we recognize the critical importance of enhancing recall within the RAG implementation. As an imperfect ($< 100\%$) recall rate results in the omission of applicable MITRE ATT&CK techniques, leading to incomplete predictions. When recall is insufficient, the system may fail to identify relevant techniques that are essential for the LLM’s reasoning process. This can ultimately weaken the quality of the model’s outputs. By implementing optimization efforts, we aim to enhance the recall of the retrieval process,

[7] https://github.com/mitre/cti/blob/master/enterprise-attack/enterprise-attack.json

ensuring that the LLM has access to all necessary information for accurate predictions.

1) ATT&CK Description Definition: At its core, our focus is identifying the optimal type of information to store in the knowledge base. To achieve this, we assess various strategies for defining the MITRE ATT&CK techniques and examine how these definitions affect the retrieval process in the vector database. We hypothesize that the amount and clarity of data associated with each technique plays a crucial role in determining how effectively the embedding model can match input sentences to relevant techniques, as the embeddings are generated based on these defined descriptions.

By evaluating different strategies for defining MITRE ATT&CK techniques, we seek to understand how the choice of description impacts the quality of the generated embeddings. This, in turn, directly influences the effectiveness of the retrieval process, ultimately affecting the overall performance of the RAG model in accurately mapping unstructured CTI to relevant techniques.

2) Embedding Models: The RAG principle in our approach relies heavily on an embedding model, which is considered the backbone of the retrieval process. It is the process that converts text input (phrases) into dense numerical vectors in a high- dimensional space, capturing semantic relationships of context and usage of the words. Within this vector space, the model is designed to position semantically similar items closer together in the embedding space. Whether a model successfully achieves this depends on its quality. The quality of the model for a specific domain, such as MITRE ATT&CK techniques, depends on several factors, including the size of the size of the model, its training process, and the quality of the training data. These factors contribute to variability in the embeddings produced by the model's output space. The RAG retrieval process can therefore potentially be enhanced by employing a model that is able to better capture the given domain. When a better quality model is employed, then the input sentence will be compared more effectively, leading to improved retrieval outcomes. This improvement aims to enhance the effectiveness of the retrieval process, ultimately providing a more relevant subset of suggestions for the model to reason upon.

3) Reranker Model: The reranker model is applied subsequently after the subset selection performed by the vector database, which is powered by the embedding model (bi-encoder). The intuition behind this step is that the embedding model generates embeddings for both the user query and all entries in the database independently. Upon receiving the user query, the model embeds the query and compares it to the pre-computed embeddings of all entries in the database using cosine similarity, which is a computationally lightweight metric. However, this precomputational step can lead to information loss in bi-encoders, since the embeddings may not fully capture the nuanced interactions between the query and the entries. Reranker models mitigate the information loss of bi-encoders, by evaluating the actual input sentences directly. They do this by entering both the user query and a single entry from the database into a transformer model, which proceeds an inference step to produce a similarity score. This approach enhances the relevance of the retrieved entries, but it comes with significantly more computational costs. Since reranker models do not rely on pre-computed embeddings, they require more resources, especially as the number of MITRE ATT&CK techniques expands. Henceforth, we implement the reranker following the subset selection performed by the vector database. This strategy limits the computationally intensive operations to a fixed set of entries, improving scalability and efficiency. Focusing the reranking process on a smaller, more relevant subset, we potentially enhance the relevancy of the subset of suggestions for the model while managing computational complexity effectively.

B. Large Language Model Interaction

The RAG integration in RAG-ATT&CK is enhanced with a range of optimization efforts listed above to get the most relevant subset of MITRE ATT&CK techniques for the LLM. The LLM remains the core component responsible for labeling each input sentence from the unstructured CTI report with the applicable MITRE ATT&CK techniques. To enhance the LLM's capabilities in performing this task, we focus on enhancing the conversational structure of the LLM through few-shot learning, various strategies for MITRE ATT&CK technique context definition, prompt-engineering, and enriched surrounding context of input sentences of the CTI reports, further investigating the impact on the model's capability in mapping unstructured CTI to MITRE ATT&CK techniques. See subfigure titled "LLM Interaction" in Figure 1.

1) ATT&CK Description Definition: In the RAG-ATT&CK system, the LLM is provided with a list of MITRE ATT&CK techniques suggested by the retrieval mechanism in the RAG implementation. To investigate. To investigate the impact of context length on the system's performance, we assess various methods for defining these techniques as context for the LLM.

Modern LLM's have a large, but limited context window, which defines the maximum number of tokens that can be included in the conversational structure before the model is unable to generate a response. Despite the context window limit not being reached, there exists research by Liu et al. [15], that suggests that LLM's struggle to effectively access and utilize information in long input contexts. Degrading performance significantly when the position of relevant information changes. The study highlights that performance is often lowest when models must extract information from the middle of long input contexts, known as model confusion.

Given this limitation, adding more context, knowledge, and understanding related to the mapping task does not necessarily lead to better performance. To address this, we investigate the impact of context length by varying the method of defining MITRE ATT&CK techniques. When specifying the selected subset of MITRE ATT&CK techniques in the RAG system using only their names and tags, the context remains concise. In contrast, providing the full descriptions as defined by MITRE results in a significantly longer context, which may hinder the model's ability to access relevant information located in the middle of the long input context.

2) Few-Shot Learning: Unlike traditional machine learning approaches that require extensive training datasets with labeled entries to train a multi-label classifier, our method leverages the capabilities of LLM's to perform few-shot learning. This allows us to instruct the pre-trained model using a limited set of pre-executed examples rather than relying on a large labeled dataset.

Traditional fine-tuning involves repeated gradient to the original model weights, requiring a substantial corpus of example tasks [16]. This method of adapting an LLM to perform the labelling of unstructured CTI to MITRE ATT&CK techniques is performed by the TRAM project [17]. In this approach, a batch of examples is processed, and subsequently, the model's gradients are updated to reflect these examples. However, we leverage few-shot learning instead of traditional fine-tuning. Few-shot learning, a form of in-context learning, does not update the model's gradients during the final training step. Instead, the model is provided with a set of examples during the model's inference phase, allowing the model to utilize these examples in its context while making predictions.

A study by Brown et al. reports that a 175 billion parameter LLM can nearly match the performance of state-of-the-art fine-tuned systems. Given the evolving threat landscape in cybersecurity and rapidly innovating property of LLM's, we have deliberately chosen to opt for the promising few-shot learning method. This approach retains flexibility in the use of the domain-specific MITRE ATT&CK knowledge and the underlying LLM. Since all LLMs operate based on conversational interactions, we retain the flexibility to swap the underlying model while maintaining the few-shot learning instructions within the context.

3) Prompt Engineering: In the few-shot learning section 3-B2, we introduce the concept of zero-shot learning, which involves describing the task to the model without providing examples. Defining the instructions is commonly referred to as prompt engineering.

In our approach, we utilize prompt engineering to clearly articulate the task that we aim to perform with the system, mapping unstructured CTI to MITRE ATT&CK techniques. This technique not only helps to clarify the intended objective but also facilitates the incorporation of our optimization efforts.

For instance, in the RAG implementation described in Section 3-A, we provide the model with a selected subset of MITRE ATT&CK techniques. Specifically, we systematically describe all the techniques in the subset using their respective tags, names, and descriptions using the following template:

`[technique: {tag} - {name} - description: '''{page_content}''']` These descriptions are then integrated into the system prompt, utilizing a specific placeholder that indicates where the data will be inserted (Appendix D). This approach aims to ensure that the model understands the nature of the information and task it is processing.

4) Context Integration: This section outlines the methodology for incorporating additional context into the input sentences. Experts mapping CTI to MITRE ATT&CK techniques [18] often utilize the adversary's target and intention amongst a paragraph or subsection of the report to judge and identify utilized techniques. Thus, adding context has the potential to enhance model's ability to better understand and capture the nuances of the sentences.

Unstructured CTI reports describe cybersecurity-related events in natural language. Sentences describing the use of a specific MITRE ATT&CK technique by an adversary requires more than just that isolated sentence; they need additional context to accurately understand the events of the adversary surrounding the cyber attack or compromise. To mimic the insights of a trained domain expert, we implement supplementary contextual information in the form of additional contextual sentences to enhance the surrounding context.

4. Dataset

To evaluate the performance of RAG-ATT&CK , we utilize the TRAM v1.3 dataset [8] [14], which is labeled by the MITRE organization. This dataset annotates MITRE ATT&CK techniques at the sentence level and includes metadata for reconstructing the full CTI document.

The metadata is particularly valuable as it provides the surrounding context for the sentences that need to be mapped to the applicable ATT&CK techniques. This surrounding context of the sentence can be retrieved because the sentences are annotated with metadata specifying the original report from which they originate. By utilizing this metadata, the proposed system can directly reconstruct the surrounding sentences and use it in the context of the model.

This TRAM dataset exists in two distinct versions. Both versions contain the same sentences of the CTI reports. However, the single-label version features a single-label classification, meaning that each entry is associated with only one MITRE ATT&CK technique. In contrast, the multi-label version is a multi-label classification, allowing each entry to have none, one, or multiple MITRE ATT&CK technique labels. The multi-label dataset is preferable because it captures all relevant ATT&CK techniques for each sentence, while the single-label version assigns only one label, even when multiple techniques are applicable or when none are applicable.

The TRAM v1.3 [9] multi-label dataset comprises 5,089 sentences labeled with techniques, focusing on a subset of 50 ATT&CK (sub-)techniques, which limits our evaluation pipeline accordingly. A list of these techniques can be found in Appendix A. The dataset was originally constructed by the MITRE organization as part of their Threat Report ATT&CK Mapper [9] project, which automates the mapping of CTI reports to MITRE ATT&CK techniques. To support their machine learning approach for fine-tuning a model, they produced this labeled dataset in two distinct versions. We discuss the dataset and its caveats in Section 6-C, where we address our concerns regarding its application and limitations.

[8] https://github.com/center-for-threat-informed-defense/tram/releases

[9] https://github.com/center-for-threat-informed-defense/tram/blob/f29793d8d665f7f552898696e00065ef24a29a20/data/tram2-data/multi_label.json

5. Evaluation

RAG-ATT&CK is implemented in Python and used to compare its performance in performance metrics (e.g. precision, recall, F1-score) to its baseline LLM integration, which is an approach of our system that merely uses an LLM without any optimization efforts applied. Table 2 gives an overview of the performance scores achieved by all compared optimization efforts.

A. Experimental Setup

Our research contains a wide range of experiments, involving the expansion and exchange of system elements. However, throughout these experiments, we consistently utilize the following components, as illustrated in Figure 1.

First off, RAG-ATT&CK utilizes the conversational capabilities of an LLM to perform the mapping of unstructured CTI to MITRE ATT&CK techniques. This is done by specifying the system prompt within the `system` message, which defines the task and provides context for the model. The input sentence from the CTI report is integrated as the `user` message, allowing the LLM to process the information individually. Upon invocation of the LLM with these messages, the LLM generates an `ai` message. This is illustrated in subfigure "LLM Invocation" in Figure 1.

To extract the applicable MITRE ATT&CK techniques predicted by the LLM, we process the output text using the regular expression pattern: `\b(T\d{4}(\.\d{3})?)\b`, to accurately extract the labels of the MITRE ATT&CK techniques identified by the model. This regex pattern is specifically designed to capture identifiers that follow the MITRE ATT&CK framework format, precise extraction from the text. The extracted ATT&CK techniques serve as the prediction performed by the LLM. The step of extracting the MITRE ATT&CK techniques in the RAG-ATT&CK system is illustrated in subfigure "ATT&CK Extraction" in Figure 1.

The described components above are essential for RAG-ATT&CK to function. Each experiment introduces changes to the described setup. In order to assess the effectiveness of these optimization efforts, we utilize the LLaMA 3.1:8b-instruct-q4_0 [19] [10] model, an open-source LLM released by Meta-AI, to support our research efforts. This choice aligns with our commitment to transparency, continuity, and reproducibility, allowing other researchers to easily access and validate our methodology and findings. Moreover, the smaller size of the LLaMA 3.1:8b-instruct-q4_0 model reduces hardware requirements, enabling a broader range of researchers to verify the results. We also apply the larger equivalent Llama-3.3-70B model in later experiments, as well as the DeepSeek-R1-70B model to obtain improved performance scores. We will now elaborate on each experiment in more detail. An overview of the performance of each experiment is provided in Table 2.

B. Evaluation Metrics

The evaluation of the RAG-ATT&CK system will be systematically assessed using key metrics, including recall, precision, and F1-score, which are particularly critical in the context of threat detection. A high recall is essential to minimize the risk of overlooking critical threats. A high precision reflects reliability in mapping text to ATT&CK techniques.

The fundamentals to the performance metrics in this multi-label classification are:

- *True Positive (TP)*: Instances where the system correctly predicts the presence of an ATT&CK technique in the text, aligning with the ground-truth label in the dataset.
- *False Positive (FP)*: Instances where the system predicts the presence of an ATT&CK technique in the text, but this prediction does not match the ground truth (i.e., the technique is not present according to the dataset).
- *False Negative (FN)*: Instances where the system fails to identify an ATT&CK technique that is present according to the ground truth in the dataset.

Given the multi-label nature of mapping CTI to MITRE ATT&CK techniques, it is not required to focus on True Negative (TN) values. With a total of 50 labels and most entries in the dataset having between 0 and 8 associated labels (with a maximum of 15), it results in a large number of rather meaningless TN values. Moreover, the targeted performance metrics, recall, and precision, do not include TN counts in their formulas. Neither does the F1-score, as it is derived from recall and precision.

Micro-averaging and macro-averaging methods provide metrics that assess the multilabel system's performance in identifying both common and rare ATT&CK techniques. These metrics report on the system's strengths and weaknesses in mapping ATT&CK techniques, aiding in further system refinement and appliance. Detailed class-specific results for the experiments are in the Appendix, while summarized results can be found in tables such as Table 3 and Table 2.

C. Baseline LLM

The results of this experiment establish the baseline performance for RAG-ATT&CK . This baseline reflects the interaction a typical user has with an LLM, where both the task of mapping unstructured CTI to MITRE ATT&CK techniques and the specific sentence to be mapped are specified. From this baseline, we continuously aim to improve the RAG-ATT&CK system by applying the defined optimization efforts. This baseline experiment is conducted by defining the prompt in Appendix B as the `system` message within the conversational input of the LLM. This prompt is crucial as it defines the domain of MITRE ATT&CK techniques and presents the model with a subset of 50 MITRE ATT&CK techniques (refer to Appendix A) on which the RAG-ATT&CK system is implemented. Defining these two messages in the conversational history of the LLM is illustrated at the top of the "Optimization Efforts" subfigure in Figure 1. The optimization efforts defined after the definition of this conversational history are not applied in this experiment. As always, the invocation step and extraction of predicted MITRE ATT&CK tags, described in Section 5-A, follow this setup.

Evaluating the entire dataset results in a macro-average score of 4.30%, 14.07%, 4.27% for the precision, recall and

[10] https://ollama.com/library/llama3.1:8b-instruct-q4_0

F1-score respectively for the BaseLLM experiment, as listed as the first experiment in Table 2. Detailed scores for each class are available in Appendix J.

The baseline performance of the experiment is underwhelming, especially when compared to the high expectations set by recent advancements in LLMs, research efforts in the field, and consumer adoption. Upon manual inspection, we observed that the model frequently misunderstood the MITRE ATT&CK technique tags. For instance, the MITRE ATT&CK technique T1190, which is defined as "Exploit Public-Facing Application" [11], was inaccurately inferred by the LLM as "Automated Collection". Such discrepancies highlight the model's challenges in accurately mapping tags to their correct definitions, complicating the effective mapping of unstructured CTI to MITRE ATT&CK mapping. While we have not quantified the total number of misinterpreted MITRE ATT&CK technique labels across the entire output, we did measure the number of tags that were not present in the original prompt and were therefore incorrectly used by the model. By extracting all MITRE ATT&CK tags from the generated `ai` message produced by the LLM, we identified how many tags were not included in the system prompt. This resulted in a total of 2,223 out of the 23,944 tags predicted by the model being classified as "unlisted predictions."

Futhermore, manual analysis of the first ten predictions revealed that all entries were incorrectly inferred. A summary of these predictions is present in Table 1, displaying a misalignment between the predicted name and actual name of the MITRE ATT&CK technique. Displaying a potential explanation for the poor performance metrics observed in the experiment.

TABLE 1
COMPARISON OF PREDICTED TAGS TO MITRE DEFINITIONS

Tag	Predicted	MITRE
T1003.001	Remote File Access	OS Credential Dumping: LSASS Memory
T1547.001	Use Alternate Authentication Material	Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder
T1190	Automated Collection	Exploit Public-Facing Application
T1484.001	Exfiltration Over Alternative Network Channel	Domain or Tenant Policy Modification: Group Policy Modification

1) Llama-3.3-70B : The same experiment was conducted using a larger and more efficient LLaMA model, specifically Llama-3.3-70B , to assess the impact of model size on the BaseLLM setup. This experiment highlights the necessity of optimization efforts in model interaction, demonstrating that simply employing a bigger LLM does not guarantee perfect mapping of unstructured CTI to MITRE ATT&CK techniques. Notably, the recall improved from a macro-average of 14.07% with the Llama-3.1-8B model to 42.80% with the Llama-3.3-70B model. This increase in recall is partially related to the higher number of predicted ATT&CK tags. The larger Llama-3.3-70B LLM predicted a total of 39.503 ATT&CK tags, compared to the 23.944 predicted by the smaller Llama-3.1-8B version of the model.

In this experiment, precision also increased, rising from a macro-average of 4.30% with Llama-3.1-8B LLM to 5.96% with the Llama-3.3-70B LLM. While this increase in precision is not substantial, the stable precision combined with the significant improvement in recall indicates an enhanced understanding of the task by the system. Notably, the Llama-3.1-8B model predicted 23,944 techniques, with 2,223 being unlisted, whereas the Llama-3.3-70B model predicted 39,503 techniques, with 2,412 unlisted. This suggests that the larger Llama model is more effective in mapping unstructured CTI to MITRE ATT&CK techniques. A detailed overview of the performance metrics for this experiment can be found in Appendix S.

Despite the improved performance of the larger LLM, we aim to implement additional optimization efforts into the RAG-ATT&CK system to further enhance its performance.

D. BaseRAG

In the BaseRAG approach, we aim to enhance the LLM's ability to make informed predictions by supplying it with factual definitions of relevant MITRE ATT&CK techniques. Through a process called RAG, introduced in Section 3-A.

For each input sentence from a CTI report, instead of directly prompting the model with this sentence as a `user` message, we first query the vector database. This process involves embedding the input sentence and comparing its embedding to the stored embeddings of the 50 ATT&CK techniques. Illustrated in subfigure "RAG (Static)" in Figure 1.

We achieve this by embedding all 50 MITRE ATT&CK techniques (see Appendix A) using an embedding model. Specifically, in this experiment, we employ the `nomic-ai/nomic-embed-text-v1.5` [20] model, which generates numerical representations of text. As an embedding model, the model is designed to position semantically similar items closer together in the embedding space. Each individual MITRE ATT&CK technique from the list of 50 is embedded, and stored in the vector database for efficient retrieval. In the "RAG (Static)" subfigure in Figure 1, it can be seen that each MITRE ATT&CK technique is embedded by the "Embed ATT&CK Techniques" step and is consequently stored in the "Weaviate" vector database.

In contrast to the BaseLLM approach, we leverage this vector database to retrieve relevant information. Specifically, we select the top-k most similar ATT&CK techniques based on the cosine similarity measure, as defined in Equation 1. The "RAG (Dynamic)" subfigure in Figure 1 displays this runtime usage of the vector database to compare each entry to the embedding of the input sentence in steps "Embed Input Sentence" & "Cosine Similarity Search".

$$\text{Cosine Similarity}(A, B) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad (1)$$

We set $k = 10$ based on the distribution of labels in the dataset, where all but one entry contains eight or fewer labels.

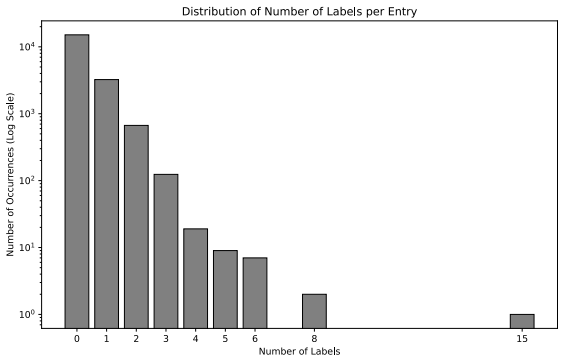
[11] https://attack.mitre.org/techniques/T1190/

TABLE 2
PERFORMANCE METRICS ACROSS DIFFERENT EXPERIMENTS AND LLMS

LLM	Variant	Support		Micro Average			Macro Average		
		Unlisted[A]	Total[B]	Precision	Recall	F1	Precision	Recall	F1
BaseLLM									
Llama-3.1-8B (19)	-	2.223	23.944	2.95%	12.46%	4.77%	4.30%	14.07%	4.27%
Llama-3.3-70B (30)	-	2.412	39.503	5.18%	37.33%	9.09%	5.96%	42.80%	9.25%
DeepSeek-R1-70B (37)	-	9.479	172.904	1.59%	50.59%	3.09%	2.16%	51.92%	3.92%
BaseRAG									
Llama-3.1-8B (22)	-	659	25.641	11.34%	55.08%	18.81%	13.56%	58.69%	19.47%
OptimizedRAG									
Llama-3.1-8B (17)	7 Techniques	662	24.694	10.34%	48.34%	17.04%	11.46%	48.71%	16.84%
Llama-3.1-8B (25)	10 Techniques	809	37.925	8.77%	63.27%	15.40%	10.65%	65.98%	16.84%
LLama-3.3-70B (38)	10 Techniques	790	56.023	7.32%	78.59%	13.39%	8.64%	82.45%	14.71%
DeepSeek-R1-70B (39)	10 Techniques	1472	141594	2.96%	**80.75%**	5.72%	3.27%	**83.71%**	6.08%
DescriptionRAG (with OptimizedRAG)									
Llama-3.1-8B (23)	Full	872	1.418	**24.18%**	2.57%	4.64%	**22.28%**	4.24%	6.76%
Llama-3.1-8B (24)	Summarized	1.383	39.111	8.73%	64.01%	15.36%	9.95%	67.84%	16.22%
Llama-3.3-70B (31)	Full	7.960	79.607	5.70%	79.47%	10.64%	6.12%	82.78%	10.85%
Llama-3.3-70B (33)	Summarized	1.320	64.306	6.44%	78.90%	11.91%	7.02%	82.37%	12.33%
Few-ShotRAG (with OptimizedRAG)									
Llama-3.1-8B (28)	Summarized	1.838	29.293	10.07%	53.74%	16.96%	12.54%	60.80%	18.92%
Llama-3.3-70B (29)	None	911	34.204	10.57%	68.42%	18.31%	12.76%	76.89%	20.15%
Llama-3.3-70B (32)	Full	7.796	38.941	10.90%	65.99%	18.71%	11.40%	68.52%	18.25%
Llama-3.3-70B (34)	Summarized	2.707	35.013	11.06%	69.47%	19.08%	12.09%	75.91%	19.50%
DeepSeek-R1-70B (35)	Summarized	3.550	95.206	4.39%	78.18%	8.31%	4.53%	80.12%	8.30%
ContextRAG (with OptimizedRAG)									
Llama-3.1-8B (26)	2 Sentences	1.668	38.792	7.09%	51.16%	12.45%	9.23%	54.20%	14.21%
Llama-3.1-8B (27)	1 Sentence	1.403	37.397	7.76%	54.31%	13.58%	10.21%	57.59%	15.60%
PromptEngineeringRAG (with OptimizedRAG & Few-ShotRAG)									
DeepSeek-R1-70B (36)	Reasoning	201	23.099	13.29%	59.15%	**21.70%**	16.35%	63.96%	**23.93%**

[A] Unlisted predictions occur when the model predicts techniques outside the specified MITRE ATT&CK subset.
[B] Total predictions by the model, which vary per input CTI sentence.

Fig. 2. Distribution of Number of Labels per Entry



This distribution is shown in Figure 2. The single outlier, which contains a total of 15 ATT&CK techniques, is a list of ATT&CK techniques in a CTI report that was entered into the dataset as a single sentence.

For the 10 preselected techniques obtained from the vector database, we supply the tag and the name to the LLM in the following format: `[technique: {tag} - {name}]`. This information is integrated into the context of the RAG-ATT&CK system by modifying the `system` message to include the retrieved context, replacing the previous list of 50 MITRE ATT&CK techniques. The sentence from the CTI report is supplied as the `user` message again. Figure 1 displays the inclusion of the selected techniques in the arrow between subfigures "RAG (Dynamic)" & "Optimization Efforts". The step "ATT&CK Definition" identifies the method for including the context of the selected techniques, explored in Section 5-F1.

The macro-average scores for the BaseRAG experiment (6) are 58.69% for recall, 13.56% for precision, and 19.47% for the F1-score. The full set of results for each class and each metric can be found in Appendix K. These performance scores represent a substantial increase from the BaseLLM setup for the RAG-ATT&CK system.

Providing the LLM with factual definitions of relevant MITRE ATT&CK techniques, selected through the vector database's retrieval mechanism, has increased the macro-average recall from 14.07% to 58.69% when comparing the BaseLLM to the BaseRAG experiment result. Both approaches predicted approximately the same number of

ATT&CK techniques (see Table 2), with the BaseLLM predicting a total of 23.944 techniques, slightly fewer than the 25.641 predictions of the BaseRAG experiment. Notably, the number of unlisted ATT&CK techniques predicted by the LLM has been significantly reduced in this experiment, from 2.223 to 659. Additionally, the macro-average precision score increased by approximately the same magnitude as the recall scores, indicating improved quality in the predictions made by the RAG-ATT&CK system. Given the approximately equal amount of predictions, this demonstrates that the BaseRAG approach is more capable of performing the task with the preselected and directed addition of factual definitions of relevant MITRE ATT&CK techniques. Shown in the performance overview in Table 2, particularly in the F1 scores.

E. Optimization Efforts

1) Retrieval Mechanism BaseRAG:
The retrieval mechanism in the BaseRAG setup serves as the driving force behind RAG-ATT&CK . By providing factual data descriptions of the MITRE ATT&CK techniques to the LLM, this mechanism enables the model to make informed predictions on the applicable MITRE ATT&CK techniques relevant to the input sentence. The retrieval mechanism in RAG-ATT&CK is depicted in Figure 1 as the combination of the steps outlined in subfigures "RAG (Static)" and "RAG (Dynamic)".

As described in the methodology (Section 3-A). An imperfect ($< 100\%$) recall rate can lead to the omission of applicable MITRE ATT&CK techniques in the retrieval mechanism of the RAG-ATT&CK system. When recall is insufficient, the system may fail to identify relevant techniques. Recall is crucial in this retrieval mechanism because it functions as the suggestion system for the RAG-ATT&CK implementation. It lists potential MITRE ATT&CK techniques, and the LLM then evaluates whether to include each technique in the final decision.

Enhancing the recall of this data retrieval system will ultimately provide the LLM with a more comprehensive range of suggested content. In this subsection, we focus on the optimization efforts related to improving the retrieval mechanism within RAG-ATT&CK .

The standard retrieval mechanism of the BaseRAG system achieves a recall @10 of 80.28%, as shown in Table 3, where it is listed as experiment #1. The table also includes recall metrics for other cut-offs, such as @1, @5, and @20. With this set of experiments, we aim to optimize the recall metrics for the different cut-offs to come closer to the perfect (100%) recall rate.

To evaluate the improvements to the retrieval mechanism compared to the mechanism featured in the RAG: Base Integration, we analyze recall metrics at the following cutoffs: @1, @5, @10, and @20. These @k cutoffs represent the number of top-ranked results used for the evaluation. For example, @5 measures recall based on the top 5 retrieved techniques, assessing how many of the actual relevant techniques appear within the top 5 predictions. By examining different cut-offs, we can assess the retrieval mechanism's performance at varying levels, providing insights into potential cutoff points for techniques provided by the LLM. The outcomes of the improvement efforts are summarized in Table 3.

We compare the results of our efforts within the three optimization effort domains and ultimately to the implementation of the retrieval mechanism of the RAG: Base Integration, which features the attributes listed in the first row of Table 3. The comparison is grounded in the recall metrics outlined earlier in this section.

2) Description Definition: Changing the description that is stored in the knowledge base, changes the outcome of the embedding model in step "Embed ATT&CK Technique" of subfigure "RAG (Static)" in Figure 1, which influences the results of the retrieval step performed by the retrieval mechanism of RAG-ATT&CK in step "Cosine Similarity Search" in subfigure "RAG (Dynamic)". Therefore, we focus on five methods to define the MITRE ATT&CK technique description in the knowledge base. These are full description, summary, first paragraph, and two splitted versions with a chunking strategy. With this approach, we evaluate through experiment 1 till 6 (see Table 3) how these varying levels of ATT&CK technique description definitions influence the retrieval mechanism of RAG-ATT&CK .

We fixate the embedding model to the `nomic-ai/nomic-embed-text-v1.5` embedding model for this experiment and do not utilize a reranker.

Full Description

The full descriptions are what is used in the BaseRAG experiment as the definition of the MITRE ATT&CK techniques in the knowledge base. In this providing full descriptions of the MITRE ATT&CK techniques. These are directly derived from the official MITRE knowledge base, the `mitre/cti` project [21]. Isolating the retrieval mechanism and evaluating it on the defined performance metrics yields a recall @10 rate of 80.28%. Serving as a baseline for this subset of experiments.

First Paragraph

Shortening the MITRE ATT&CK techniques to just the first paragraph retains key information but omits details about connections to other techniques and example cases. As a result, the selection process may focus more on the general application of the technique, rather than its specific use cases or its connections to hacker groups. Potentially optimally generalizing the resulting embedding. This method is implemented by splitting the ATT&CK technique definitions from the `mitre/cti` project at the first two newline characters and taking the first segment of the result.

The performance score for recall @10 is 77.06%. Despite the omission of additional relational data and examples, it did not result in an improved score over the baseline. This suggests that the absence of more interconnected context limits the ability of the model to enhance retrieval performance.

Summarized Description

This method involves creating a condensed version of the MITRE ATT&CK technique descriptions, aiming to highlight

TABLE 3
RECALL ACROSS VARIATIONS FOR OPTIMIZEDRAG

Embedding Model	Description Method	Reranker Model	Precision@1	Recall@1	Recall@5	Recall@10	Recall@20
Baseline Configuration							
nomic-embed-text (1)	Full	None	9.44	37.44%	69.06%	80.28%	89.08%
Description Variants (nomic-embed-text, No Reranker)							
nomic-embed-text (4)	First Paragraph	None	9.05%	35.57%	64.46%	77.06%	89.64%
nomic-embed-text (3)	Summary	None	9.84%	38.87%	**71.74%**	**83.93%**	**92.34%**
nomic-embed-text (5)	Relevancy 1	None	**10.28%**	**40.94%**	66.21%	77.61%	86.74%
nomic-embed-text (6)	Relevancy 2	None	9.80%	38.66%	62.01%	71.70%	82.00%
Embedding Model Variants (Full Description, No Reranker)							
stella-en-500M-v5 (8)	Full	None	11.01%	43.93%	78.13%	88.46%	94.94%
ATTACK-BERT (7)	Full	None	10.61%	42.22%	75.66%	86.05%	94.08%
text-embedding-3-large (12)	Full	None	**11.70%**	**46.68%**	**81.33%**	**90.31%**	**96.36%**
SecureBERT-Plus (16)	Full	None	7.15%	27.91%	59.69%	75.21%	88.01%
Reranker Variant (nomic-embed-text, Full Description)							
nomic-embed-text (10)	Full	ms-marco-MiniLM-L-6-v2	8.88%	35.01%	63.24%	74.59%	79.77%
Verification Experiment							
ATTACK-BERT (14)	Summary	None	10.53%	41.99%	74.81%	86.26%	94.01%

the most relevant aspects while reducing complexity. By retaining the core definition of each technique, this approach allows information to be drawn from the entire text rather than just the first paragraph. The condensed versions are generated using the Llama-3.1-8B [19] LLM, utilizing the system prompt defined as the `system` message in Appendix C.

The process is conducted iteratively, providing the full descriptions of all MITRE ATT&CK techniques one at a time to generate their summaries. Each iteration resets the chat context, ensuring that the model focuses on the current technique being summarized without retaining previous messages or interactions. The prompt is specifically designed to create a shortened version of the description, preserving the core information.

The summarized descriptions yield a recall @10 rate of 83.93%, representing an improvement over the retrieval mechanism benchmark compared to the full description baseline score. Although this method shares a similar intention to shorten or condense the description as the First Paragraph approach, it retains more crucial information necessary for the embedding model to enhance retrieval performance.

Chunked Description

In this approach, MITRE ATT&CK technique descriptions are divided into smaller segments, named chunks. Aiming to facilitate efficient retrieval and processing through a more directed and specific matching process. This experiment tested two chunk sizes: larger chunks with a maximum length of 750 characters and an overlap of 80 characters, and smaller chunks with a maximum length of 375 characters and an overlap of 50 characters. The choice of specific character limits and overlap values is based on balancing the retention of sufficient context with optimizing processing efficiency. The overlap is crucial as it ensures that key contextual information is retained, minimizing the risk of losing important connections between related concepts when consecutive chunks are retrieved.

The recall @10 rates were 77.61% and 71.70% for the larger and smaller chunks, respectively. These performance scores suggest that while chunking increases granularity, it may not provide the contextual benefit needed to surpass the baseline score in the retrieval mechanism.

To conclude the MITRE ATT&CK technique description definition experiments, our findings display the varying effectiveness of different description strategies on retrieval performance. Suggesting that Summarized Description approach made the `nomic-ai/nomic-embed-text-v1.5` embedding model best capable of retrieving the top 10 most relevant MITRE ATT&CK techniques while being the only approach capable of surpassing the baseline Full Description performance.

3) Embedding Models: The effectiveness of the retrieval mechanism in RAG-ATT&CK is heavily dependent on the quality of the embedding model. Embedding models serve as the backbone of the retrieval mechanism, by converting textual inputs into dense numerical vectors in a high-dimensional space. These generated vectors capture semantic relationships and contextual nuances, which inherently place semantically similar items closer together. The process of embedding textual inputs to vector representations in RAG-ATT&CK is illustrated in Figure 1, within subfigures "RAG (Dynamic)" & "RAG (Static)", as the "Embed..." steps. What we define as quality for an embedding model for a given domain, such as MITRE ATT&CK techniques, is dependent on the size of the model, its training process, and the quality of its training data. A higher quality embedding model for the MITRE ATT&CK domain would allow for more effective comparisons of input sentences to the stored MITRE ATT&CK techniques, improving the retrieval outcomes.

To find the best quality embedding model for the MITRE ATT&CK domain in the RAG-ATT&CK system, we con-

ducted experiments while fixating the ATT&CK description definition method to a Full Description and not utilizing a reranker model. Allowing for a direct comparison between the baseline performance score.

In total, we evaluated four candidate models to replace the baseline configuration `nomic-ai/nomic-embed-text-v1.5` model.

- Top 10 MTEB leaderboard Embedding Model: As of February 3, 2025, the `stella-en-400M-v5` model ranks #6 on the leaderboard. Table 3, experiment 8.
- MITRE ATT&CK specific embedding model: The `ATTACK-BERT` model is a fine-tuned BERT embedding model. Table 3, experiment #7.
- Cybsersecurity focused model: The `Secure-BERT-Plus` is a fine-tuned BERT model trained using a range of cybersecurity-related data. Table 3, experiment #16.
- Proprietary model: The `open-ai/text-embedding-3-large` model is a proprietary model of which the weights are not published. Available only through the OpenAI API. Table 3, experiment #12.

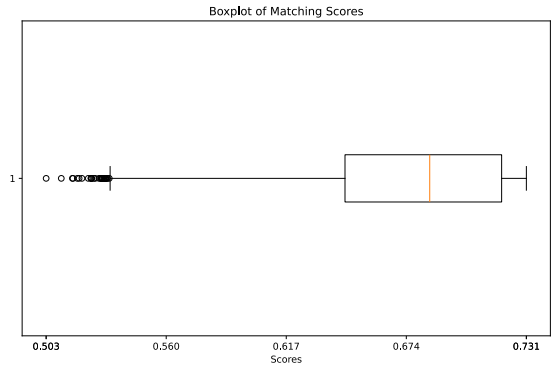
Among the models tested, the `text-embedding-3-large` model came out as the top performer, achieving a recall @10 rate of 90.31%. Both the `ATTACK-BERT` and `stella-en-400M-v5` models outperformed the baseline `nomic-embed-text-v1.5` model. All three of these models can capture the nuanced differences between the ATT&CK technique descriptions. Despite `Secure-BERT-Plus`'s focus on cybersecurity data, it failed to achieve a score higher than the baseline model.

Despite the superior performance of the `text-embedding-3-large` model on the dataset, we chose to employ the `ATTACK-BERT` model in our optimized use case due to its non-proprietary nature and its reduced computational complexity relative to the `stella-en-400M-v5`. The `ATTACK-BERT` model provides a balanced approach, optimizing both performance and computational efficiency, making it the model of choice for RAG-ATT&CK .

4) Reranker Model: The integration of a reranker model into our system aims to alleviate the limitations of the embedding model's retrieval mechanism. The embedding model creates embeddings of all MITRE ATT&CK techniques and stores them in a database. When a query is prompted to the database, it compares the embedding of that query to the stored embeddings of the MITRE ATT&CK techniques, potentially leading to information loss. The reranker model mitigates this by directly evaluating the input sentence to the MITRE ATT&CK descriptions.

The `ms-marco-MiniLM-L-6-v2` reranker model is integrated into RAG-ATT&CK by adding it as a step following the retrieval step by the embedding model. The preselection step performed by the embedding model is crucial, as processing the entire set of MITRE ATT&CK techniques through a reranker model would be computationally infeasible. The reranker model processes the 15 preselected techniques, by inputting both the CTI input sentence and each technique into a transformer model to produce a similarity score.

Fig. 3. Box Plot Similarities Matching Scores



Examining the outcome of this reranker experiment showed us that the recall @10 had decreased to 74.59% when compared to the baseline. While reranking models theoretically have the potential to enhance the retrieval mechanism, our findings show that the reranker model is not effective in the MITRE ATT&CK domain. The full results can be found in Tabel 3, in experiment #10. As illustrated in Figure 3, the similarity scores between the CTI input sentence and the MITRE ATT&CK technique definition range from a minimum of 0.503 to a maximum of 0.731. This distribution indicates that there are no very poor matches (scores below 0.25), but also no highly similar matches (scores above 0.90). Notably, even when examining scores that should ideally indicate no match at all, the minimum score remains at 0.500. This shows the challenge faced by the current models in effectively discriminating between the relevance of closely related items within the vector database. We speculate that the homogeneity in the nature of the data is complicating the identification of meaningful differences over an embedding model. Therefore, we suspect that a reranker model is more beneficial in scenarios where the retrieved results vary more significantly. Thus, we will not apply a reranker model in the optimized retrieval mechanism of RAG-ATT&CK .

To conclude the optimization experiments for the retrieval mechanism, we have combined the chosen configurations of the three optimization domains. For the Description Definition experiment, we selected the Summarized method; for the Embedding Model experiment, we selected the `ATTACK-BERT` model; and we deliberately opted out of applying a reranker model. This combination represents verification experiment #14 in Table 3, resulting in a recall @10 rate of 86.26%. This is a negligible difference from experiment #7, where the same embedding model utilized the Full Description definition method. Both Description Definition approaches result in a higher recall rate across all cut-offs, demonstrating that Description Definition does not critically impact the `ATTACK-BERT` model's performance. Therefore, we choose

not to apply the Summarized method to reduce complexity.

This final optimized retrieval mechanism is integrated into the RAG-ATT&CK system, employing the Llama-3.1-8B LLM in experiment #25 in Table 2, named OptimizedRAG. Upon evaluation, this configuration achieves macro-average performance scores of 65.98% for recall, 10.65% for precision, and 16.84% for the F1-score.

5) Llama-3.3-70B : The OptimizedRAG experiment was also conducted using the larger and more efficient Llama-3.3-70B model to evaluate the impact of model size on the optimized retrieval setup (see Table 2). This experiment underlines the significance of the applied optimization efforts, illustrating that the enhancements can lead to improvements in performance, even with larger models.

Notably, the recall increased from 65.90% with the smaller Llama-3.1-8B model to 82.45% with the Llama-3.3-70B model. The larger model has the ability to predict a more comprehensive set of relevant ATT&CK tags. This can be related to the increased number of techniques predicted, increasing from 37.925 with the Llama-3.1-8B LLM to 56.023 with the Llama-3.3-70B LLM. As a consequence, precision has decreased from 10.65% with the smaller model to 8.64% with the larger model. Despite the larger Llama-3.3-70B benefitting from the RAG integration by improving over the BaseLLM experiment with the larger LLM, it does not manage to perform better than the smaller equivalent on the OptimizedRAG experiment.

F. LLM Interaction

We now evaluate various optimization efforts applied to the interaction with the LLM. The range of experiments aims to optimize the conversational structure of the LLM to optimize the mapping of unstructured CTI to MITRE ATT&CK techniques. These components of RAG-ATT&CK are collectively depicted in the subfigure "Optimization Efforts" in Figure 1.

1) Description Definition: In this section, we outline the employed strategies to provide MITRE ATT&CK technique descriptions to the LLM, focusing on three distinct approaches: no description, full description, and summarized description. With this approach, we can evaluate how varying levels of added contextual information impact the model's performance in mapping CTI to MITRE ATT&CK techniques. This experiment differs from the Retrieval Mechanism Description Definition, listed as experiment #3 till #6 in Table 3 since this experiment aims to assess how the LLM, instead of the retrieval mechanism, interacts with various levels of information provided as MITRE ATT&CK definitions. As a result, this component utilizing the ATT&CK descriptions is depicted as step "ATT&CK Definition" as part of subfigure "Optimization Efforts in Figure 1.

No Description

The previous RAG experiments utilized a RAG system that supplied the LLM with only the tags and the name of the MITRE ATT&CK techniques. This approach serves as a baseline for comparison. For completeness, the context template for this approach was: `[technique: {tag} - {name}]`. For the smaller Llama-3.1-8B LLM, this relates to experiment #25 in Table 2.

Full Description

In this phase, we enhance the input to the LLM by providing full descriptions of the MITRE ATT&CK techniques. These are directly derived from the official MITRE knowledge base [21]. This provides the model with access to comprehensive and accurate information about each technique. We aim to determine whether this increased level of detail improves the model's ability to predict the MITRE ATT&CK techniques. To do so, we adapt the context template to `[technique: {tag} - {name} - description: '''{page_content}''']`. This includes the full description as defined by the MITRE ATT&CK framework [22] instead of just the tag identifier and name. Applying the integration of MITRE ATT&CK definition to the smaller Llama-3.1-8B model resulted in significant model confusion. This addition led to the model predicting only a total of 1.418 MITRE ATT&CK techniques in the output, of which 872 unlisted predictions. When compared to the approach where no ATT&CK technique descriptions were provided, the recall rate declined from 65.98% to 4.24%. This approach scored the lowest macro-average recall rate amongst all performed experiments in this paper, displaying model confusion in the smaller Llama-3.1-8B LLM. In contrast, the larger Llama-3.3-70B LLM (experiment #31) shows a slightly decreased macro-average precision compared to the approach without technique definitions, while maintaining a similar recall rate in Table 2. Although the larger model does achieve an improved F1-score with the full descriptions, it shows no significant signs of model confusion. Indicating that model confusion is more prevalent in the smaller model.

Summarized Description

This third and final phase of this experiment involves supplying the LLM with summarized descriptions of the MITRE ATT&CK techniques. These summaries are, like in the retrieval mechanism experiments, generated based on the full descriptions to capture the essential elements of each technique, while not incorporating less critical details. The summarized descriptions are created using the prompt listed in Appendix C. With this, we aim to evaluate whether a concise informative description can enhance the model's performance. The results indicate that the reduction in context length by providing summarized MITRE ATT&CK description definitions potentially eliminates model confusion in the smaller Llama-3.1-8B LLM. However, it does provide a substantial improvement over the no-description comparison. It is likely that the summarized information is so generalized that the model already contains this information within its existing knowledge. For the larger Llama-3.3-70B LLM, the use of summarized descriptions slightly improves the F1-score from 10.85% to 12.33% compared to the no-description equivalent. This improvement may potentially be related to the reduction of noise from other techniques being relationally mentioned in the full description definitions.

To conclude, neither the Full Description nor the Summarized Description definition approach resulted in a substantial improvement over the No Description approach in mapping unstructured CTI to MITRE ATT&CK techniques.

2) Few-Shot Learning:
The implementation of our proposed system involves a system prompt, defined as the `system` message. This definition of the task is referred to as prompt engineering, and in the context of few-shot learning, it can also be considered zero-shot learning, as it defines the task without providing an example (see 2). Upon receiving an input sentence from the unstructured CTI report, our system processes it as a `user` message and formulates an `ai` message the response. Our few-shot learning approach leverages the conversational aspect of the LLM to place examples within the context. We integrate a total of three example label processes as `user` and `ai` messages:

- **Single Technique Example**: An input sentence labeled with a single MITRE ATT&CK technique, demonstrating the task of labeling the input sentence with a domain-specific label.
- **Multiple Techniques Example**: An input sentence labeled with two MITRE ATT&CK techniques, demonstrating the task again, as well as the possibility of having multiple domain-specific labels, highlighting the multi-label classification problem.
- **No Technique Example**: An input sentence labeled with no MITRE ATT&CK techniques, showcasing the scenario in which no applicable MITRE ATT&CK techniques exist for the input sentence, highlighting the multi-label classification problem once more.

The set of examples generalizes the task from the minimal context provided to the model. These examples aim to improve the performance of the LLM to map unstructured CTI to MITRE ATT&CK techniques. The integration of example mappings is defined as conversational history, as illustrated in step "Few Shot" as part of subfigure "Optimization Efforts" in Figure 1.

Given the inconclusive benefits of the Description Definition methods in the previous experiment (Section 5-F1), this Few-Shot experiment was evaluated on the larger Llama-3.3-70B LLM on all three description methods to assess their impact on further optimizations before drawing a definitive conclusion on what approach is the best. The performance scores defined in Table 2 as experiments #29, 32, and 34 show no significant difference in performance scores amongst the various Description Definition methods. Despite this, the integration of the example labeling in the Few-Shot Learning integration shows improved performance across all technique description definition methods for the Llama-3.3-70B LLM. The macro-average F1-score for the Llama-3.3-70B LLM increased from 14.71% to 20.15% for the No Description approach, from 10.85% to 18.25% for the Full Description approach, from 12.33% to 19.50% for the Summarized Description approach. This indicates that Few-Shot Learning effectively aids in defining the mapping task of unstructured CTI to MITRE ATT&CK techniques.

3) Context: We integrate surrounding context into the system by modifying the system prompt to describe the presence of additional context alongside the sentence that requires to be mapped to the applicable MITRE ATT&CK techniques. This approach aims to replicate the understanding and methodology of a domain expert, potentially resulting in more accurate mapping of unstructured CTI to MITRE ATT&CK techniques.

Context Enrichment
To provide the model with the additional context in which the sentence that has to be mapped is present, we employ a strategy that retrieves 'n' number of sentences prior to and after each sentence. This approach partially reconstructs the paragraph in the CTI report surrounding the input sentence. In the event of not having enough surrounding sentences, the context will be shorter. The integration of context in RAG-ATT&CK is depicted in the step labeled "Context" within the subfigure "Optimization Efforts" in Figure 1. In this process, the original sentence from the CTI report is used alongside the originating CTI report, to reconstruct the paragraph. The input sentence is then replaced with the paragraph in the conversational history of the LLM.

Integration into the Input
The integration of the prompt that defines the inclusion of the surrounding context can be found in Appendix E. Additionally, we modify the `user` message by annotating it with the surrounding context, structured as follows: `The main sentence is: '''{main_sentence}'''. The main sentence is placed in the context: '''{input}'''. Please label the main sentence.`.

Neither the integration of one sentence before and after nor the integration of two sentences resulted in improved performance in mapping unstructured CTI to MITRE ATT&CK techniques. This LLM interaction optimization effort was evaluated only on the Llama-3.1-8B LLM. These results suggest that the sentences in the labeled dataset were labeled strictly on the individual sentence itself, without consideration of the surrounding context. Further manual investigation is required to reach a definitive conclusion.

6. Discussion

The integration of RAG in our system, RAG-ATT&CK , has demonstrated internal improvements in performance, particularly when compared to our baseline (BaseLLM) experiments. These enhancements underscore the potential of RAG to map unstructured CTi to MITRE ATT&CK techniques by leveraging retrieval mechanisms and LLM's.

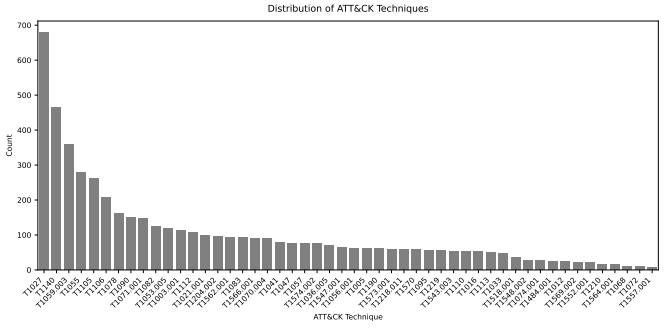
A. Comparison to State-of-the-Art Approaches

While we have made notable internal advancements, our approach has not surpassed the performance of existing methods in the field. Our best result, achieved in experiment 36 as shown in Table 2, falls short when compared to the current State-of-the-Art approach, TRAM v1.3 [14]. Table 4 provides

TABLE 4
COMPARISON OF RAG-ATT&CK AGAINST STATE-OF-THE-ART

Tool	Micro Average			Macro Average		
	Precision	Recall	F1	Precision	Recall	F1
RAG-ATT&CK	13.29%	**59.15%**	21.70%	16.35%	**63.96%**	23.93%
TRAM v1.3 [14]	**73.60%**	40.00%	**51.80%**	**36.50%**	21.20%	**25.10%**

Fig. 4. TRAM [14] Dataset Class Distribution



a detailed comparison of these tools, using the metrics utilized in our evaluation.

When comparing RAG-ATT&CK to TRAM v1.3, it is observed that TRAM v1.3 demonstrates a vastly superior precision at 73.60%. This difference can be attributed to the class imbalance present in the dataset, as illustrated in Figure 4. The difference between micro average and macro average precision for TRAM v1.3 suggests that its performance is particularly strong in the majority class, which substantially, elevates its micro average precision score greatly compared to RAG-ATT&CK .

However, at the macro-average level, where precision across all classes is weighted equally, TRAM v1.3 maintains superior performance. Despite the difference in precision being less significant here, it remains substantial, indicating that RAG-ATT&CK is not as capable as TRAM v1.3 in consistently identifying MITRE ATT&CK techniques across diverse classes. Highlighting the area where RAG-ATT&CK could be enhanced, improving the precision across all classes.

Although RAG-ATT&CK underperforms in terms of precision, it demonstrates notable strength in recall at the macro-average level, achieving more than three times the recall of TRAM v1.3. This substantial difference in macro-average recall is significant enough that RAG-ATT&CK nearly compensates for its lower precision, achieving an F1-score of 23.93% compared to TRAM v1.3 with 25.10%. In contrast, the smaller difference in the micro-average metrics suggests that RAG-ATT&CK does not benefit as much from the majority class(es) shown in Figure 4 as TRAM v1.3.

B. RAG or Fine-Tuning

In the previous Section 6-A, we discussed the comparison between RAG-ATT&CK and the current State-of-the-Art, TRAM v1.3 [14]. Both tools leverage an LLM as the core of their implementation. However, RAG-ATT&CK focuses on integrating a RAG with the LLM, while TRAM v1.3 employs fine-tuning on a foundation model. This sparks an engaging discussion between the two approaches.

In the context of this research, both RAG-ATT&CK and TRAM v1.3 are tasked with classifying the same dataset for evaluation, thereby addressing the same question. The goal is to produce an identifier for a MITRE ATT&CK technique. To enforce this output format in RAG-ATT&CK , which utilizes a RAG system, it is required to specify the desired format in the prompt within the LLM's conversational history. Specifically, for MITRE ATT&CK technique tags, the format should be Txxxx.xxx, and this request must be repeated for each input. In contrast, the fine-tuning approach incorporates the desired output format contained within the dataset used for the fine-tuning by adjusting the weights to align the output with this specific format. During inference, the fine-tuned model is more likely to respond in the format embedded in the training data, as its weights are optimized to recognize and produce this format.

Fine-tuning adjusts the model's weights to incorporate domain-specific knowledge, allowing it to leverage existing foundational knowledge to predict correct values. By embedding this information into the model's architecture, the model can utilize it when answering questions. Depending on the type of fine-tuning, a selection, or all weights are updated. Adjusting weights by fine-tuning is different from RAG, where the domain-specific MITRE ATT&CK information is included in the context during invocation. While this means access to the information, it may not effectively integrate concepts across different data pieces to formulate an answer, nor will it adhere to the tone, or format of the dataset. For example, fine-tuning with cybersecurity content enables the model to better relate the term "shell" to a command-line interface, commonly used by computers, rather than a hard protective case, commonly occurring in nature.

For generating high-quality answers, fine-tuning offers significant advantages in mapping unstructured CTI to MITRE ATT&CK techniques by embedding knowledge directly into the model's weights, enhancing the versatility of the information. As a result, the fine-tuning approach used in TRAM v1.3, based on the previous generation SciBert LLM, outperforms the RAG approach, which employs the latest generation reasoning models such as DeepSeek-R1-70B , as displayed by superior F1-scores in Table 4.

RAG, however, does provide benefits, mainly adaptability to new information. Unlike fine-tuning, which requires retraining whenever there is new domain knowledge or an update to the foundational model. However, the RAG system of RAG-ATT&CK can incorporate updates by simply adding the entry

to the vector database used during LLM inference. This facilitates immediate integration of the latest information. Moreover, RAG systems can directly cite the context from which the system derives the answer, allowing for traceability of information sources. While RAG can identify the sources of the information, it may optimally leverage the retrieved data. As of now, newer generation models like Llama-3.3-70B and DeepSeek-R1-70B require fine-tuning to be compatible with the TRAM v1.3 system.

C. Dataset

The MITRE TRAM dataset [9] is a multi-label dataset, meaning that each entry can have zero or more MITRE ATT&CK technique labels associated with it. This dataset serves as a ground truth in our approach aimed at automating the mapping of CTI reports to MITRE ATT&CK techniques. In other approaches, such as TRAM itself, it is used for training and evaluating the machine learning models. The dataset consists of CTI reports that have been split up into individual sentences, with each entry potentially linked to multiple MITRE ATT&CK technique labels.

Within the dataset, some sentences are notably short, sometimes consisting of only one to four words. Despite the sentences not describing an activity or appliance of a MITRE ATT&CK technique, the entry is still labeled with an associated ATT&CK technique. Additionally, sentences within the same paragraph share overlapping labels, adding complexity to the multi-label classification task. Approaches like TRAM perform training on the samples in the dataset to achieve high-performance scores during the subsequent testing. However, reasoning-based methods that do not utilize fine-tuning, such as our RAG-ATT&CK approach, do not learn these data nuances. The overlap in labels across sentences in sparsely defined paragraphs, combined with labels associated with sentences that do not describe an activity, raises questions about the specificity of the dataset's labeling. Potentially requiring further investigation if interested in advancing the field of automated mapping of MITRE ATT&CK techniques.

D. Evaluation Structure

The complexity of the multi-label mapping of unstructured CTI to MITRE ATT&CK techniques, combined with the specificity of the dataset, potentially suggests the need for a more flexible evaluation approach. The current evaluation procedure requires an exact match between the labeling of ATT&CK techniques and the select input sentence. When considering this evaluation structure critically, one may conclude that such a strict evaluation structure may not be necessary, or beneficial. The mapping of unstructured CTI to MITRE ATT&CK techniques does not need to be word-for-word, nor even sentence-by-sentence, as currently required. The usage of ATT&CK techniques may be described throughout a paragraph or in just a few words within natural text. Thus, it might be more meaningful to recognize the usage of MITRE ATT&CK techniques throughout the text, rather than restricting the classification to individual whole sentences.

Restricting the classification to a sentence-to-sentence-based approach could potentially limit the progression of automated CTI mapping to MITRE ATT&CK techniques because machine learning models could learn the improper nuances defined in the sentences, rather than in the actual natural text. A new proposal for a new labeling and subsequent evaluation mechanism could be a future research direction.

A proposed new evaluation mechanism could allow for a range of flexibility, allowing for more dynamic mapping and detection of MITRE ATT&CK techniques in unstructured CTI. Currently, unstructured CTI is mapped in a structured manner.

7. Related Work

A set of related works explored various ways to extract CTI information and label CTI report sentences, automating the mapping of unstructured CTI to MITRE ATT&CK techniques.

1) CTI Knowledge Extraction: Recent work by Cheng et al. [23] represents an advancement in the field of CTI knowledge extraction and Cybersecurity Knowledge Graph (CSKG) construction by leveraging In-Context Learning (ICL) with LLMs. This approach enables data-efficient extraction of cybersecurity entities and relationships without training on extensive datasets or fine-tuning. Firstly, the methodology utilizes an automatic prompt construction strategy that retrieves demonstrations for extracting relevant cybersecurity entities and relations. This process identifies a subset of example mapping deemed relevant to the input through the cosine similarity 1 metric applied on transformer-enabled embeddings. The results are further refined using an entity alignment technique, which involves coarse-grained grouping and fine-grained merging to maintain a diverse and relevant set of examples. This allows the model to adapt to the specified task. Our approach aligns with the principles of CTINexus [23] by employing a RAG system. Similarly, RAG-ATT&CK seeks to maximize the relevancy of the supplied information to the LLM, allowing for effective data usage. Also, our system applies few-shot learning, which is a method for improving task adaption with limited data [16], by applying a small number of example outputs. While CTINexus [23] focuses on constructing CSKGs, instead of mapping entities and actions to ATT&CK techniques, both approaches overlap in their goal of extracting cyber threat information from natural text and converting it into a desired format through learning from instructions and examples.

A. TRAM v1.2

Threat Report ATT&CK Mapper [12] maps to techniques from the ATT&CK framework. TRAM is an open-source tool developed by the cybersecurity community in collaboration with MITRE, designed to closely align with the ATT&CK framework. A significant contribution of TRAM v1.2 [9] is its user-friendly interface, which allows users to utilize implemented machine learning (ML) methods to map new unstructured CTI to techniques in the ATT&CK framework. This version uses traditional ML models, including Logistic Regression, Naive Bayes, and Multi-Layer Perceptron. TRAM v1.2 facilitates a graphical interface that enables users to run

[12]https://github.com/center-for-threat-informed-defense/tram

these models directly and retrieve annotated versions of the original CTI reports uploaded to the platform. Specific details of the implementation of supervised learning methods can be found in the TRAM GitHub repository[13].

B. TRAM v1.3

An approach with transformer-based LLM's is TRAM. The updated TRAM v1.3 platform [14] employs transformer-based methods, specifically fine-tuning the SciBERT model on a dataset of the 50 most common ATT&CK techniques[14], each with multiple labeled examples[15]. This fine-tuned SciBERT model can consequently be used for inference for input sentences from CTI reports to automate the mapping of unstructured CTI to MITRE ATT&CK techniques.

C. Natural Language Processing

Works like TTPDrill [5] use NLP techniques like POS-Tagging and TF-IDF (Term Frequency-Inverse Document Frequency), combined with the BM25 weighting scheme to classify threat actions into a structured set of patterns. In contrast to TTPDrill, other works, such as the study by Orbinato et al. [8], specifically focus on the MITRE ATT&CK framework. The study applies and compares various methods of applying NLP techniques such as Stopword Removal, Stemming, and Lemmatization. They then build ML classifiers including Naive Bayes, Logistic Regression, and Multi-Layer Perceptron to perform the mapping to the MITRE ATT&CK framework. Similarly, TRAM v1.2 applies similar methods and provides a graphical interface that enables users to run these models and retrieve annotated versions of the original CTI reports. As well as Ayoade et al.'s [10] work explores ML methods for mapping to the MITRE ATT&CK framework, specifically by utilizing the Support Vector Machine (SVM) method. Additionally, the EAGLE framework [1] emphasizes NLP methods, including Coreference Resolution, and Named Entity Recognition, which have not been highlighted in the other works discussed in this paper. These methods are compared to each other and collectively aim to extract and classify relevant information from unstructured CTI data.

D. Embedding NLP Techniques

The development of embedding models has inspired a range of works to utilize the Word2Vec technology from 2013 [11], which captures semantic relationships between words. rcATT by Legoy et al. [24] utilizes this technology to train a ML classifier capable of ranking the relevance of words compared to techniques in the MITRE ATT&CK framework.

8. Conclusion

In this work, we introduced RAG-ATT&CK , a novel approach for automating the mapping of unstructured CTI to MITRE ATT&CK techniques. Unlike existing methods, our approach does not require fine-tuning of an LLM or the training of an ML classifier. While we achieved performance improvements over the baseline LLM conversational setup, our approach did not consistently rival or surpass the current state-of-the-art in the multi-label classification of unstructured CTI to MITRE ATT&CK techniques.

Multi-label classification of MITRE ATT&CK techniques presents a significant challenge due to the vast number of possible label combinations. The proposed RAG system encounters difficulties in selecting the correct MITRE ATT&CK techniques with sufficient precision. Although RAG-ATT&CK is provided a qualitative set of suggestions through RAG, it struggles to accurately distinguish the applied techniques within the sentences of the CTI report.

Further research is needed to explore the more promising method of fine-tuning. By leveraging the latest advancements in LLM's, we can potentially enhance the effectiveness of mapping unstructured CTI to MITRE ATT&CK techniques. This approach could extend beyond current fine-tuning implementations on outdated models as TRAM v1.3 [14] and improve upon the less efficient RAG systems such as RAG-ATT&CK . Ultimately, this would result in a system ready for production use. Currently, the existing technologies are not ready for deployment in multi-label environments.

[13]https://github.com/center-for-threat-informed-defense/tram/blob/main/src/tram/ml/base.py

[14]https://github.com/center-for-threat-informed-defense/tram/blob/main/data/ml-models/bert_model/classes.txt

[15]https://github.com/center-for-threat-informed-defense/tram/blob/main/data/tram2-data/single_label.json

REFERENCES

[1] SoK: Threat Intelligence Processing - Unleashing the Real Power of Natural Language Processing for Cyber Threat Intelligence (EAGLE). (To be published).

[2] Ziyun Zhu and Tudor Dumitras. ChainSmith: Automatically Learning the Semantics of Malicious Campaigns by Mining Threat Intelligence Reports. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 458–472, April 2018.

[3] Ghaith Husari, Xi Niu, Bill Chu, and Ehab Al-Shaer. Using Entropy and Mutual Information to Extract Threat Actions from Cyber Threat Intelligence. In *2018 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 1–6, November 2018.

[4] Marc Mezard and Andrea Montanari. Information, Physics and Computation. 2008.

[5] Ghaith Husari, Ehab Al-Shaer, Mohiuddin Ahmed, Bill Chu, and Xi Niu. TTPDrill: Automatic and Accurate Extraction of Threat Actions from Unstructured Text of CTI Sources. In *Proceedings of the 33rd Annual Computer Security Applications Conference*, ACSAC '17, pages 103–115, New York, NY, USA, December 2017. Association for Computing Machinery.

[6] Ziyun Zhu and Tudor Dumitraş. FeatureSmith: Automatically Engineering Features for Malware Detection by Mining the Security Literature. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 767–778, Vienna Austria, October 2016. ACM.

[7] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The Stanford CoreNLP Natural Language Processing Toolkit. In Kalina Bontcheva and Jingbo Zhu, editors, *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[8] Vittorio Orbinato, Mariarosaria Barbaraci, Roberto Natella, and Domenico Cotroneo. Automatic Mapping of Unstructured Cyber Threat Intelligence: An Experimental Study, August 2022. arXiv:2208.12144.

[9] The Center for Threat-Informed Defense. center-for-threat-informed-defense/tram v1.2, September 2020. (Available on GitHub).

[10] Gbadebo Ayoade, Swarup Chandra, Latifur Khan, Kevin Hamlen, and Bhavani Thuraisingham. Automated Threat Report Classification over Multi-Source Data. In *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*, pages 236–245, October 2018.

[11] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space, September 2013. arXiv:1301.3781 [cs].

[12] Valentine Legoy, Marco Caselli, Christin Seifert, and Andreas Peter. Automated Retrieval of ATT&CK Tactics and Techniques for Cyber Threat Reports, April 2020. arXiv:2004.14322 [cs].

[13] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Neural Information Processing Systems*, June 2017.

[14] The Center for Threat-Informed Defense. center-for-threat-informed-defense/tram v1.3, October 2024. (Available on GitHub).

[15] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the Middle: How Language Models Use Long Contexts, November 2023. arXiv:2307.03172 [cs].

[16] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners, July 2020. arXiv:2005.14165.

[17] Center for Threat Informed Defense. center-for-threat-informed-defense/tram at b50bfd19ba0e7f79f06999b82ec62eecb5ec5963.

[18] Best Practices for MITRE ATTCK Mapping.

[19] Abhimanyu Dubey, Abhinav Jauhri, Ahmad Al-Dahle, and Egor Lakomkin. The Llama 3 Herd of Models, August 2024. arXiv:2407.21783.

[20] Zach Nussbaum, John X. Morris, Brandon Duderstadt, and Andriy Mulyar. Nomic Embed: Training a Reproducible Long Context Text Embedder, February 2025. arXiv:2402.01613 [cs].

[21] mitre/cti, October 2024. original-date: 2017-06-05T16:18:24Z.

[22] mitre-attack/attack-stix-data, October 2024. original-date: 2021-05-20T16:47:13Z.

[23] Yutong Cheng, Osama Bajaber, Saimon Amanuel Tsegai, Dawn Song, and Peng Gao. CTINEXUS: Leveraging Optimized LLM In-Context Learning for Constructing Cybersecurity Knowledge Graphs Under Data Scarcity, October 2024. arXiv:2410.21060.

[24] vlegoy. vlegoy/rcATT, October 2024. original-date: 2019-09-07T23:33:52Z.

APPENDIX A
TRAM-V1.3 SUPPORTED CLASSES

'T1003.001', 'T1005', 'T1012', 'T1016', 'T1021.001', 'T1027', 'T1033', 'T1036.005', 'T1041', 'T1047', 'T1053.005', 'T1055', 'T1056.001', 'T1057', 'T1059.003', 'T1068', 'T1070.004', 'T1071.001', 'T1072', 'T1074.001', 'T1078', 'T1082', 'T1083', 'T1090', 'T1095', 'T1105', 'T1106', 'T1110', 'T1112', 'T1113', 'T1140', 'T1190', 'T1204.002', 'T1210', 'T1218.011', 'T1219', 'T1484.001', 'T1518.001', 'T1543.003', 'T1547.001', 'T1548.002', 'T1552.001', 'T1557.001', 'T1562.001', 'T1564.001', 'T1566.001', 'T1569.002', 'T1570', 'T1573.001', 'T1574.002'

APPENDIX B
LLM PROMPT

Task Instructions:

Label the following sentences with a corresponding MITRE ATT&CK Technique. If no applicable technique exists, please respond with the string "No Mapping."

Guidelines:

- You may assign multiple MITRE ATT&CK techniques to a single sentence if relevant. - Limit your mappings to the following MITRE ATT&CK techniques:

"' T1003.001, T1005, T1012, T1016, T1021.001, T1027, T1033, T1036.005, T1041, T1047, T1053.005, T1055 T1056.001, T1057, T1059.003, T1068, T1070.004, T1071.001, T1072, T1074.001, T1078, T1082, T1083, T1090 T1095, T1105, T1106, T1110, T1112, T1113, T1140, T1190, T1204.002, T1210, T1218.011, T1219, T1484.001, T1518.001 T1543.003, T1547.001, T1548.002, T1552.001, T1557.001, T1562.001, T1564.001, T1566.001, T1569.002, T1570, T1573.001, T1574.002 "'

Output Format:

Your response should be formatted as follows: 'Txxxx.xxx'

APPENDIX C
LLM PROMPT TECHNIQUE DESCRIPTION SUMMARIZIATION

> Can U create a summary of the following MITRE ATT&CK technique description. Only include information that is helpful in the process of mapping the technique to natural text. Keep the original way of writing the description please. Only output the text itself, no additional information. Since it is a summary, make it shorter than the original text please.

APPENDIX D
BASE RAG PROMPT

Task Instructions:

Label the sentences provided by the user with a corresponding MITRE /attack Technique. If no applicable technique exists, please respond with the string "No Mapping."

Guidelines:

- You may assign multiple MITRE /attack techniques to a sentence if relevant. - Limit your mappings to the following MITRE /attack techniques:

Begin list MITRE /attack techniques. "' context "' —

Now create a list of applicable MITRE /attack techniques to the sentence inputted by the user.

Output Format:

Your response should be formatted as follows: 'Txxxx : Name'

APPENDIX E
CONTEXT RAG PROMPT

Task Instructions:

Label the main sentence provided by the user with a corresponding MITRE ATT&CK Technique. The main sentence is the one that appears between the context sentences. If no applicable technique exists, please respond with the string "No Mapping."

Guidelines:

- The user will provide a full context that includes text before, the main sentence to be labeled, and text after.

- You may assign multiple MITRE /attack techniques to the main sentence if relevant. - Limit your mappings to the following MITRE /attack techniques:

Begin list MITRE /attack techniques. "' context "' —

Now create a list of applicable MITRE /attack techniques to the main sentence in the provided context.

Output Format:

Your response should be formatted as follows: 'Txxxx : Name'

APPENDIX F
RAG PROMPT ENGINEERED

You are a professional MITRE /attack technique labeller. Using the provided context, label the sentence provided by the user with a corresponding MITRE /attack Technique or Sub-technique. If no applicable technique exists, please respond with the string ”No Mapping.”

Guidelines:

- Techniques represent specific behaviors to achieve a goal, often a single step in a string of activities intended to complete the adversary’s overall mission. Sub-techniques provide more granular descriptions of techniques. - You may assign multiple MITRE /attack techniques to a single sentence if relevant. - Limit your mappings to the following MITRE /attack techniques:

— “‘ context “‘ —

Instructions:

1. Look for signs of adversary behavior, try to identify initial access as well as post-compromise activity. (Not all behaviors may translate into (sub-)techniques, technical details can build on each other to inform an understanding of the overall adversary behavior and objectives.) 2. Translate the behavior into a Tactic, focusing on why the adversary performed the behavior. 3. Identify the (sub-)Technique that applies to the behavior, considering the specific goals of the adversary. Map to the parent technique only if there is not enough context to identify a sub-technique.

Output Format:

Your response should be formatted as follows: ‘Txxxx : Name‘ or ‘Txxxx.yyy : Name‘ for sub-techniques.

APPENDIX G
RELATED WORK OVERVIEW

TABLE 5
RELATED WORK TABLE

Paper	Approach Type	Tool	MITRE ATT&CK Technique Mapping
	Information Assessment	VECTR	✓
	Information Assessment	Rabobank/DeTTECT	✓
	Information Assessment	SenseOn	✓
	Information Assessment	cisagov/decider	✓
	Information Assessment	Caldera	✓
Using Entropy and Mutual Information to Extract Threat Actions from Cyber Threat Intelligence	NLP	ActionMiner	✗
SoK Threat Intelligence Processing - USENIX (EAGLE) [1]	NLP	EAGLE	✓
FeatureSmith: Automatically Engineering Features for Malware Detection by Mining the Security Literature [6]	NLP	FeatureSmith	✗
TTPDrill: Automatic and Accurate Extraction of Threat Actions from Unstructured Text of CTI Sources [5]	NLP	TTPDrill	✗
Automated Retrieval of ATT&CK Tactics and Techniques for Cyber Threat Reports [12]	NLP (Word2Vec) Machine Learning	vlegoy/rcATT	✓
Automatic Mapping of Unstructured Cyber Threat Intelligence: An Experimental Study: (Practical Experience Report) [8]	NLP Machine Learning Transformed-based language modeling	Apply several state-of-the-art classification models	✓
Automated Threat Report Classification over Multi-Source Data [10]	NLP Machine Learning	Ayoade et al.	✓
Center for Threat Informed Defense [9]	NLP	TRAM v1.2	✓
Center for Threat Informed Defense [14]	Transformer Machine Learning	TRAM v1.3	✓

APPENDIX H
LLM EXPERIMENTS (LLAMA3.1:8B-INSTRUCT-Q4_0)

TABLE 6
ATTRIBUTES OF THE LLM EXPERIMENTS

Experiment	LLM	Embedding Model	ATT&CK definition	MITRE Prompt	Context	Few-Shot
BaseLLM (19)	llama3.1:8b-instruct-q4_0	-	-	No	-	-
BaseRAG (20)	llama3.1:8b-instruct-q4_0	nomic-embed-text	Name, Tag	No	-	-
RAGSimilarity (17)	llama3.1:8b-instruct-q4_0	ATTACK-BERT	Name, Tag	No	-	-
RAGPrompt (18)	llama3.1:8b-instruct-q4_0	ATTACK-BERT	Name, Tag	Yes	-	-
RAGDescription	llama3.1:8b-instruct-q4_0	ATTACK-BERT	Name, Tag, Description	Yes	-	-
RAGDescription2	llama3.1:8b-instruct-q4_0	ATTACK-BERT	Name, Tag, Summary (Description)	Yes	-	-
RAGContext	llama3.1:8b-instruct-q4_0	ATTACK-BERT	Name, Tag, Summary (Description)	Yes	2	-
RAGContext2	llama3.1:8b-instruct-q4_0	ATTACK-BERT	Name, Tag, ??	Yes	3	-
RAGFewShot	llama3.1:8b-instruct-q4_0	ATTACK-BERT	Name, Tag, ??	Yes	-	Yes

APPENDIX I
EXP 17 - RAG: SIMILARITY RETRIEVAL OPTIMIZED: K=7

Class	Recall	Precision	F1-Score	Support
T1003.001	0.6161	0.2123	0.3158	112
T1005	0.6557	0.0328	0.0625	61
T1012	0.7500	0.0381	0.0726	24
T1016	0.8462	0.0419	0.0799	52
T1021.001	0.7677	0.2815	0.4119	99
T1027	0.6298	0.1647	0.2612	678
T1033	0.4894	0.0578	0.1034	47
T1036.005	0.5211	0.0600	0.1076	71
T1041	0.7250	0.0454	0.0854	80
T1047	0.8701	0.0833	0.1521	77
T1053.005	0.7797	0.1949	0.3119	118
T1055	0.7266	0.1777	0.2855	278
T1056.001	0.9194	0.2336	0.3725	62
T1057	0.7403	0.0686	0.1256	77
T1059.003	0.4749	0.2791	0.3516	358
T1068	0.8000	0.0271	0.0525	10
T1070.004	0.6333	0.1326	0.2192	90
T1071.001	0.5473	0.0936	0.1599	148
T1072	0.6000	0.0073	0.0145	10
T1074.001	0.4615	0.0152	0.0294	26
T1078	0.5776	0.1144	0.1910	161
T1082	0.3415	0.0938	0.1471	123
T1083	0.3871	0.0611	0.1056	93
T1090	0.2819	0.3043	0.2927	149
T1095	0.4211	0.0485	0.0870	57
T1105	0.2069	0.1731	0.1885	261
T1106	0.2077	0.1265	0.1572	207
T1110	0.3077	0.1416	0.1939	52
T1112	0.3178	0.1453	0.1994	107
T1113	0.5510	0.4154	0.4737	49
T1140	0.2618	0.2198	0.2390	466
T1190	0.4918	0.0592	0.1056	61
T1204.002	0.2708	0.0807	0.1244	96
T1210	0.3529	0.0234	0.0440	17
T1218.011	0.3833	0.0639	0.1095	60
T1219	0.4286	0.1218	0.1897	56
T1484.001	0.5417	0.1066	0.1781	24
T1518.001	0.2162	0.0191	0.0352	37
T1543.003	0.5472	0.1676	0.2566	53
T1547.001	0.4462	0.0939	0.1551	65
T1548.002	0.6538	0.1453	0.2378	26
T1552.001	0.2273	0.0279	0.0498	22
T1557.001	0.1429	0.0116	0.0215	7
T1562.001	0.2688	0.1337	0.1786	93
T1564.001	0.2500	0.0417	0.0714	16
T1566.001	0.4667	0.1757	0.2553	90
T1569.002	0.2727	0.1132	0.1600	22
T1570	0.2881	0.1018	0.1504	59
T1573.001	0.3333	0.0673	0.1120	60
T1574.002	0.3553	0.0823	0.1337	76
Micro-average	0.4834	0.1034	0.1704	5143
Macro-average	0.4871	0.1146	0.1684	5143
Weighted-average	0.4834	0.1551	0.2161	5143

APPENDIX J
EXP 19 - BASELLM LLAMA-3.1-8B

Class	Recall	Precision	F1-Score	Support
T1003.001	0.3304	0.0381	0.0683	112
T1005	0.2787	0.0116	0.0222	61
T1012	0.3333	0.0066	0.0129	24
T1016	0.4231	0.0213	0.0405	52
T1021.001	0.2323	0.0367	0.0634	99
T1027	0.0442	0.0680	0.0536	678
T1033	0.0851	0.0078	0.0143	47
T1036.005	0.1408	0.0146	0.0264	71
T1041	0.1500	0.0130	0.0239	80
T1047	0.0649	0.0234	0.0344	77
T1053.005	0.3305	0.0209	0.0394	118
T1055	0.5971	0.0661	0.1190	278
T1056.001	0.1452	0.0219	0.0381	62
T1057	0.0779	0.0335	0.0469	77
T1059.003	0.1061	0.0754	0.0882	358
T1068	0.4000	0.0381	0.0696	10
T1070.004	0.1000	0.1125	0.1059	90
T1071.001	0.0541	0.0421	0.0473	148
T1072	0.0000	0.0000	0.0000	10
T1074.001	0.0769	0.0370	0.0500	26
T1078	0.0745	0.1429	0.0980	161
T1082	0.1951	0.0264	0.0465	123
T1083	0.0753	0.0393	0.0517	93
T1090	0.0403	0.0217	0.0282	149
T1095	0.0702	0.0123	0.0209	57
T1105	0.0230	0.0385	0.0288	261
T1106	0.0242	0.1562	0.0418	207
T1110	0.0000	0.0000	0.0000	52
T1112	0.0467	0.0714	0.0565	107
T1113	0.0204	0.0303	0.0244	49
T1140	0.0150	0.1522	0.0273	466
T1190	0.4262	0.0319	0.0594	61
T1204.002	0.0833	0.0289	0.0429	96
T1210	0.2353	0.0114	0.0218	17
T1218.011	0.0333	0.0033	0.0060	60
T1219	0.0179	0.0060	0.0090	56
T1484.001	0.0417	0.0040	0.0073	24
T1518.001	0.0811	0.0056	0.0104	37
T1543.003	0.3774	0.0259	0.0485	53
T1547.001	0.2308	0.0172	0.0321	65
T1548.002	0.1154	0.0259	0.0423	26
T1552.001	0.2273	0.0208	0.0382	22
T1557.001	0.2857	0.0076	0.0149	7
T1562.001	0.0860	0.1778	0.1159	93
T1564.001	0.0000	0.0000	0.0000	16
T1566.001	0.0556	0.0781	0.0649	90
T1569.002	0.0000	0.0000	0.0000	22
T1570	0.0847	0.1515	0.1087	59
T1573.001	0.0333	0.0606	0.0430	60
T1574.002	0.0658	0.1111	0.0826	76
Micro-average	0.1246	0.0295	0.0477	5143
Macro-average	0.1407	0.0430	0.0427	5143
Weighted-average	0.1246	0.0661	0.0527	5143

APPENDIX K
EXP 22 - BASERAG

Class	Recall	Precision	F1-Score	Support
T1003.001	0.5625	0.3539	0.4345	112
T1005	0.4754	0.0670	0.1174	61
T1012	0.6667	0.0792	0.1416	24
T1016	0.6346	0.0669	0.1211	52
T1021.001	0.7071	0.3448	0.4636	99
T1027	0.5501	0.2013	0.2947	678
T1033	0.4043	0.0931	0.1514	47
T1036.005	0.2958	0.0778	0.1232	71
T1041	0.6750	0.0829	0.1477	80
T1047	0.6623	0.2742	0.3878	77
T1053.005	0.6525	0.5347	0.5878	118
T1055	0.7230	0.2436	0.3645	278
T1056.001	0.9194	0.2938	0.4453	62
T1057	0.6883	0.0710	0.1286	77
T1059.003	0.4832	0.3078	0.3761	358
T1068	0.6000	0.0349	0.0659	10
T1070.004	0.5778	0.1948	0.2913	90
T1071.001	0.4392	0.1193	0.1876	148
T1072	0.3000	0.0050	0.0098	10
T1074.001	0.4231	0.0462	0.0833	26
T1078	0.5528	0.1127	0.1872	161
T1082	0.5691	0.0872	0.1512	123
T1083	0.7204	0.0748	0.1355	93
T1090	0.5235	0.2583	0.3459	149
T1095	0.7368	0.0487	0.0913	57
T1105	0.1839	0.1322	0.1538	261
T1106	0.1643	0.3953	0.2321	207
T1110	0.8462	0.2056	0.3308	52
T1112	0.6822	0.1370	0.2281	107
T1113	0.7755	0.1810	0.2934	49
T1140	0.4678	0.2264	0.3051	466
T1190	0.7869	0.0422	0.0801	61
T1204.002	0.3438	0.0459	0.0810	96
T1210	0.5294	0.0078	0.0153	17
T1218.011	0.8667	0.1130	0.2000	60
T1219	0.8036	0.0470	0.0888	56
T1484.001	0.8750	0.0698	0.1292	24
T1518.001	0.7027	0.0321	0.0615	37
T1543.003	0.6226	0.1701	0.2672	53
T1547.001	0.8462	0.1024	0.1827	65
T1548.002	0.0769	0.1176	0.0930	26
T1552.001	0.5909	0.0267	0.0512	22
T1557.001	0.5714	0.0656	0.1176	7
T1562.001	0.4086	0.1583	0.2282	93
T1564.001	0.5000	0.0494	0.0899	16
T1566.001	0.7778	0.1431	0.2418	90
T1569.002	0.2273	0.0267	0.0478	22
T1570	0.6441	0.0605	0.1106	59
T1573.001	0.6000	0.0818	0.1440	60
T1574.002	0.9079	0.0690	0.1283	76
Micro-average	0.5508	0.1134	0.1881	5143
Macro-average	0.5869	0.1356	0.1947	5143
Weighted-average	0.5508	0.1894	0.2545	5143

APPENDIX L
EXP 23 - RAG DESCRIPTION FULL

Class	Recall	Precision	F1-Score	Support
T1003.001	0.0714	0.3636	0.1194	112
T1005	0.0492	0.3000	0.0845	61
T1012	0.0417	0.1429	0.0645	24
T1016	0.0577	0.1579	0.0845	52
T1021.001	0.0202	0.1538	0.0357	99
T1027	0.0059	0.2500	0.0115	678
T1033	0.0426	0.1667	0.0678	47
T1036.005	0.0141	0.2000	0.0263	71
T1041	0.0500	0.3333	0.0870	80
T1047	0.0390	0.2308	0.0667	77
T1053.005	0.0593	0.3889	0.1029	118
T1055	0.0144	0.2353	0.0271	278
T1056.001	0.0000	0.0000	0.0000	62
T1057	0.0519	0.2857	0.0879	77
T1059.003	0.0000	0.0000	0.0000	358
T1068	0.2000	0.2500	0.2222	10
T1070.004	0.0556	0.2778	0.0926	90
T1071.001	0.0270	0.2857	0.0494	148
T1072	0.0000	0.0000	0.0000	10
T1074.001	0.0769	0.5000	0.1333	26
T1078	0.0745	0.4615	0.1283	161
T1082	0.0488	0.2500	0.0816	123
T1083	0.0430	0.2222	0.0721	93
T1090	0.0201	0.3750	0.0382	149
T1095	0.0175	0.2500	0.0328	57
T1105	0.0115	0.1429	0.0213	261
T1106	0.0097	0.2000	0.0184	207
T1110	0.0000	0.0000	0.0000	52
T1112	0.0093	0.1000	0.0171	107
T1113	0.0000	0.0000	0.0000	49
T1140	0.0064	0.2500	0.0126	466
T1190	0.2131	0.6842	0.3250	61
T1204.002	0.0208	0.2000	0.0377	96
T1210	0.0588	0.1111	0.0769	17
T1218.011	0.0167	0.2000	0.0308	60
T1219	0.0000	0.0000	0.0000	56
T1484.001	0.0417	0.3333	0.0741	24
T1518.001	0.0000	0.0000	0.0000	37
T1543.003	0.0755	0.3333	0.1231	53
T1547.001	0.0462	0.2500	0.0779	65
T1548.002	0.0000	0.0000	0.0000	26
T1552.001	0.1364	0.5000	0.2143	22
T1557.001	0.2857	1.0000	0.4444	7
T1562.001	0.0000	0.0000	0.0000	93
T1564.001	0.0000	0.0000	0.0000	16
T1566.001	0.0444	0.2857	0.0769	90
T1569.002	0.0000	0.0000	0.0000	22
T1570	0.0508	0.3000	0.0870	59
T1573.001	0.0000	0.0000	0.0000	60
T1574.002	0.0132	0.1667	0.0244	76
Micro-average	0.0257	0.2418	0.0464	5143
Macro-average	0.0424	0.2228	0.0676	5143
Weighted-average	0.0257	0.2198	0.0436	5143

APPENDIX M
EXP 24 - RAG DESCRIPTION SUMMARIZED

Class	Recall	Precision	F1-Score	Support
T1003.001	0.7679	0.2028	0.3209	112
T1005	0.5738	0.0253	0.0484	61
T1012	0.6250	0.0246	0.0474	24
T1016	0.7500	0.0319	0.0612	52
T1021.001	0.7273	0.2441	0.3655	99
T1027	0.6077	0.1865	0.2854	678
T1033	0.6809	0.0597	0.1098	47
T1036.005	0.5634	0.0598	0.1081	71
T1041	0.7500	0.0502	0.0941	80
T1047	0.7403	0.2036	0.3193	77
T1053.005	0.7712	0.1957	0.3122	118
T1055	0.7662	0.2165	0.3376	278
T1056.001	0.9355	0.2468	0.3906	62
T1057	0.7792	0.1033	0.1824	77
T1059.003	0.4777	0.2429	0.3220	358
T1068	0.7000	0.0437	0.0824	10
T1070.004	0.7000	0.1230	0.2093	90
T1071.001	0.6081	0.0889	0.1552	148
T1072	0.2000	0.0033	0.0064	10
T1074.001	0.5385	0.0172	0.0333	26
T1078	0.6522	0.1240	0.2083	161
T1082	0.6667	0.0840	0.1492	123
T1083	0.7957	0.0519	0.0975	93
T1090	0.6510	0.2160	0.3244	149
T1095	0.7018	0.0553	0.1026	57
T1105	0.4138	0.1271	0.1944	261
T1106	0.3527	0.1125	0.1706	207
T1110	0.8654	0.1429	0.2452	52
T1112	0.6916	0.1407	0.2338	107
T1113	0.8163	0.2500	0.3828	49
T1140	0.5172	0.2939	0.3748	466
T1190	0.8033	0.0447	0.0847	61
T1204.002	0.7917	0.0424	0.0804	96
T1210	0.6471	0.0099	0.0194	17
T1218.011	0.8667	0.0295	0.0571	60
T1219	0.6607	0.0509	0.0945	56
T1484.001	0.9167	0.0551	0.1040	24
T1518.001	0.7027	0.0144	0.0282	37
T1543.003	0.8302	0.0695	0.1283	53
T1547.001	0.8615	0.0885	0.1605	65
T1548.002	0.8846	0.1095	0.1949	26
T1552.001	0.7273	0.0227	0.0441	22
T1557.001	0.5714	0.0177	0.0343	7
T1562.001	0.5054	0.0879	0.1497	93
T1564.001	0.5625	0.0299	0.0568	16
T1566.001	0.8889	0.0938	0.1697	90
T1569.002	0.2273	0.0424	0.0714	22
T1570	0.5593	0.0714	0.1267	59
T1573.001	0.6167	0.0639	0.1158	60
T1574.002	0.9079	0.0611	0.1145	76
Micro-average	0.6401	0.0873	0.1536	5143
Macro-average	0.6784	0.0995	0.1622	5143
Weighted-average	0.6401	0.1513	0.2298	5143

APPENDIX N
EXP 25 - RAG: SIMILARITY RETRIEVAL OPTIMIZED: K=10

Class	Recall	Precision	F1-Score	Support
T1003.001	0.6161	0.2828	0.3876	112
T1005	0.4590	0.0262	0.0495	61
T1012	0.6667	0.0370	0.0700	24
T1016	0.7885	0.0347	0.0665	52
T1021.001	0.7273	0.3547	0.4768	99
T1027	0.6504	0.1326	0.2203	678
T1033	0.4681	0.0582	0.1035	47
T1036.005	0.3521	0.0665	0.1119	71
T1041	0.7500	0.0542	0.1011	80
T1047	0.7143	0.1667	0.2703	77
T1053.005	0.7966	0.2741	0.4078	118
T1055	0.6691	0.2391	0.3523	278
T1056.001	0.8871	0.2941	0.4418	62
T1057	0.6623	0.0687	0.1245	77
T1059.003	0.4274	0.3214	0.3669	358
T1068	0.6000	0.0429	0.0800	10
T1070.004	0.7000	0.1671	0.2698	90
T1071.001	0.6216	0.1132	0.1915	148
T1072	0.2000	0.0028	0.0054	10
T1074.001	0.5000	0.0219	0.0419	26
T1078	0.6584	0.1160	0.1972	161
T1082	0.6829	0.0602	0.1106	123
T1083	0.8280	0.0425	0.0809	93
T1090	0.5436	0.2883	0.3767	149
T1095	0.8596	0.0463	0.0878	57
T1105	0.4138	0.1343	0.2028	261
T1106	0.4396	0.1218	0.1908	207
T1110	0.8654	0.1613	0.2719	52
T1112	0.7944	0.1171	0.2041	107
T1113	0.8163	0.2500	0.3828	49
T1140	0.6116	0.1851	0.2841	466
T1190	0.8361	0.0330	0.0634	61
T1204.002	0.4688	0.0366	0.0680	96
T1210	0.5294	0.0106	0.0207	17
T1218.011	0.9000	0.0506	0.0958	60
T1219	0.6964	0.0595	0.1096	56
T1484.001	0.8333	0.0844	0.1533	24
T1518.001	0.6757	0.0170	0.0331	37
T1543.003	0.7925	0.0955	0.1704	53
T1547.001	0.9077	0.0493	0.0936	65
T1548.002	0.8846	0.0950	0.1716	26
T1552.001	0.6818	0.0321	0.0613	22
T1557.001	0.7143	0.0246	0.0476	7
T1562.001	0.3871	0.0750	0.1257	93
T1564.001	0.7500	0.0399	0.0757	16
T1566.001	0.9111	0.1137	0.2022	90
T1569.002	0.1818	0.0519	0.0808	22
T1570	0.5932	0.0653	0.1176	59
T1573.001	0.5667	0.0450	0.0834	60
T1574.002	0.9079	0.0627	0.1173	76
Micro-average	0.6327	0.0877	0.1540	5143
Macro-average	0.6598	0.1065	0.1684	5143
Weighted-average	0.6327	0.1493	0.2246	5143

APPENDIX O
EXP 26 - RAG OPTIMIZED K=10 CONTEXT 2

Class	Recall	Precision	F1-Score	Support
T1003.001	0.5714	0.2025	0.2991	112
T1005	0.4918	0.0181	0.0349	61
T1012	0.4583	0.0317	0.0593	24
T1016	0.5577	0.0306	0.0579	52
T1021.001	0.5152	0.2684	0.3529	99
T1027	0.6003	0.1323	0.2168	678
T1033	0.2553	0.0472	0.0797	47
T1036.005	0.2817	0.0407	0.0710	71
T1041	0.6000	0.0403	0.0755	80
T1047	0.6623	0.1536	0.2494	77
T1053.005	0.8136	0.2297	0.3582	118
T1055	0.6475	0.2287	0.3380	278
T1056.001	0.5484	0.2519	0.3452	62
T1057	0.4675	0.0642	0.1129	77
T1059.003	0.1788	0.2294	0.2009	358
T1068	0.6000	0.0349	0.0659	10
T1070.004	0.4000	0.0893	0.1460	90
T1071.001	0.4527	0.0770	0.1316	148
T1072	0.2000	0.0024	0.0047	10
T1074.001	0.5385	0.0188	0.0363	26
T1078	0.5404	0.1037	0.1740	161
T1082	0.4146	0.0533	0.0944	123
T1083	0.7957	0.0376	0.0717	93
T1090	0.4899	0.2441	0.3259	149
T1095	0.7544	0.0546	0.1018	57
T1105	0.2146	0.1016	0.1379	261
T1106	0.2899	0.1263	0.1760	207
T1110	0.7500	0.2053	0.3223	52
T1112	0.6542	0.1515	0.2460	107
T1113	0.5510	0.3000	0.3885	49
T1140	0.4442	0.1837	0.2599	466
T1190	0.8525	0.0662	0.1228	61
T1204.002	0.3333	0.0201	0.0379	96
T1210	0.4118	0.0058	0.0114	17
T1218.011	0.7333	0.0263	0.0508	60
T1219	0.6071	0.0518	0.0955	56
T1484.001	0.8333	0.0687	0.1270	24
T1518.001	0.4054	0.0082	0.0160	37
T1543.003	0.7925	0.0794	0.1443	53
T1547.001	0.8308	0.0443	0.0842	65
T1548.002	0.8077	0.1010	0.1795	26
T1552.001	0.5000	0.0284	0.0538	22
T1557.001	0.7143	0.0202	0.0392	7
T1562.001	0.5269	0.0543	0.0985	93
T1564.001	0.3125	0.0094	0.0182	16
T1566.001	0.8222	0.0636	0.1181	90
T1569.002	0.1364	0.0380	0.0594	22
T1570	0.4237	0.0681	0.1174	59
T1573.001	0.4500	0.0388	0.0714	60
T1574.002	0.8684	0.0667	0.1239	76
Micro-average	0.5116	0.0709	0.1245	5143
Macro-average	0.5420	0.0923	0.1421	5143
Weighted-average	0.5116	0.1300	0.1885	5143

APPENDIX P
EXP 27 - RAG OPTIMIZED K=10 CONTEXT 1

Class	Recall	Precision	F1-Score	Support
T1003.001	0.6161	0.2456	0.3511	112
T1005	0.4590	0.0178	0.0343	61
T1012	0.5000	0.0310	0.0584	24
T1016	0.5769	0.0304	0.0577	52
T1021.001	0.5354	0.2994	0.3841	99
T1027	0.6239	0.1480	0.2392	678
T1033	0.3191	0.0573	0.0971	47
T1036.005	0.2817	0.0382	0.0672	71
T1041	0.6625	0.0460	0.0860	80
T1047	0.6883	0.1715	0.2746	77
T1053.005	0.8475	0.2481	0.3839	118
T1055	0.6583	0.2395	0.3512	278
T1056.001	0.6774	0.2727	0.3889	62
T1057	0.5584	0.0776	0.1363	77
T1059.003	0.2318	0.2660	0.2478	358
T1068	0.7000	0.0437	0.0824	10
T1070.004	0.5333	0.1297	0.2087	90
T1071.001	0.5473	0.0948	0.1617	148
T1072	0.3000	0.0039	0.0076	10
T1074.001	0.4615	0.0182	0.0350	26
T1078	0.5466	0.1019	0.1717	161
T1082	0.5041	0.0596	0.1066	123
T1083	0.7527	0.0360	0.0688	93
T1090	0.4564	0.2688	0.3383	149
T1095	0.7719	0.0582	0.1082	57
T1105	0.2375	0.1163	0.1562	261
T1106	0.2705	0.1299	0.1755	207
T1110	0.8077	0.2270	0.3544	52
T1112	0.7290	0.1560	0.2570	107
T1113	0.6122	0.3371	0.4348	49
T1140	0.4936	0.2114	0.2960	466
T1190	0.8525	0.0660	0.1225	61
T1204.002	0.4167	0.0261	0.0491	96
T1210	0.5294	0.0077	0.0152	17
T1218.011	0.7500	0.0277	0.0535	60
T1219	0.6607	0.0535	0.0989	56
T1484.001	0.8750	0.0805	0.1474	24
T1518.001	0.4865	0.0095	0.0186	37
T1543.003	0.7547	0.0891	0.1594	53
T1547.001	0.8154	0.0453	0.0859	65
T1548.002	0.8846	0.0983	0.1769	26
T1552.001	0.6364	0.0333	0.0632	22
T1557.001	0.5714	0.0192	0.0372	7
T1562.001	0.4839	0.0530	0.0955	93
T1564.001	0.3750	0.0133	0.0256	16
T1566.001	0.8889	0.0735	0.1358	90
T1569.002	0.1818	0.0533	0.0825	22
T1570	0.3390	0.0563	0.0966	59
T1573.001	0.4500	0.0405	0.0743	60
T1574.002	0.8816	0.0767	0.1412	76
Micro-average	0.5431	0.0776	0.1358	5143
Macro-average	0.5759	0.1021	0.1560	5143
Weighted-average	0.5431	0.1447	0.2081	5143

APPENDIX Q
EXP 28 - RAG OPTIMIZED K=10 FEW-SHOT LEARNING

Class	Recall	Precision	F1-Score	Support
T1003.001	0.7054	0.2821	0.4031	112
T1005	0.5410	0.0393	0.0733	61
T1012	0.4583	0.0317	0.0593	24
T1016	0.6923	0.0437	0.0823	52
T1021.001	0.5960	0.3533	0.4436	99
T1027	0.3909	0.2392	0.2968	678
T1033	0.3617	0.0605	0.1037	47
T1036.005	0.4225	0.0679	0.1170	71
T1041	0.7125	0.0581	0.1074	80
T1047	0.6883	0.2246	0.3387	77
T1053.005	0.8051	0.1885	0.3055	118
T1055	0.7050	0.2576	0.3773	278
T1056.001	0.8871	0.3873	0.5392	62
T1057	0.6234	0.1098	0.1868	77
T1059.003	0.4469	0.2802	0.3445	358
T1068	0.7000	0.0795	0.1429	10
T1070.004	0.6889	0.1766	0.2812	90
T1071.001	0.3649	0.0826	0.1347	148
T1072	0.3000	0.0055	0.0108	10
T1074.001	0.5385	0.0302	0.0573	26
T1078	0.5404	0.1526	0.2380	161
T1082	0.5935	0.0973	0.1672	123
T1083	0.7527	0.0541	0.1009	93
T1090	0.4497	0.2965	0.3573	149
T1095	0.8421	0.0544	0.1021	57
T1105	0.2529	0.2031	0.2253	261
T1106	0.3285	0.1593	0.2145	207
T1110	0.7115	0.2741	0.3957	52
T1112	0.7290	0.2281	0.3474	107
T1113	0.8571	0.2515	0.3889	49
T1140	0.3519	0.2095	0.2626	466
T1190	0.5902	0.0651	0.1173	61
T1204.002	0.3750	0.0427	0.0766	96
T1210	0.4706	0.0080	0.0157	17
T1218.011	0.9167	0.0413	0.0791	60
T1219	0.6786	0.0446	0.0837	56
T1484.001	0.9167	0.0482	0.0917	24
T1518.001	0.6216	0.0147	0.0287	37
T1543.003	0.7547	0.0946	0.1681	53
T1547.001	0.8615	0.0926	0.1672	65
T1548.002	0.8846	0.2054	0.3333	26
T1552.001	0.6364	0.0335	0.0636	22
T1557.001	0.4286	0.0429	0.0779	7
T1562.001	0.4624	0.1468	0.2228	93
T1564.001	0.6875	0.0468	0.0876	16
T1566.001	0.8778	0.1193	0.2101	90
T1569.002	0.0909	0.0312	0.0465	22
T1570	0.4746	0.0793	0.1359	59
T1573.001	0.7000	0.0487	0.0910	60
T1574.002	0.9342	0.0856	0.1569	76
Micro-average	0.5374	0.1007	0.1696	5143
Macro-average	0.6080	0.1254	0.1892	5143
Weighted-average	0.5374	0.1781	0.2457	5143

APPENDIX R
EXP 29 - LLAMA-3.3-70B FEWSHOTRAG NO DESCRIPTIONS

Class	Recall	Precision	F1-Score	Support
T1003.001	0.8393	0.1728	0.2866	112
T1005	0.6393	0.0262	0.0503	61
T1012	0.7500	0.0433	0.0818	24
T1016	0.7885	0.0441	0.0836	52
T1021.001	0.9192	0.2345	0.3737	99
T1027	0.5413	0.2583	0.3497	678
T1033	0.8511	0.0618	0.1153	47
T1036.005	0.6197	0.0609	0.1110	71
T1041	0.8375	0.0526	0.0990	80
T1047	0.8312	0.2433	0.3765	77
T1053.005	0.8559	0.3176	0.4633	118
T1055	0.8022	0.2777	0.4126	278
T1056.001	0.9516	0.3450	0.5064	62
T1057	0.7922	0.0915	0.1640	77
T1059.003	0.5642	0.2172	0.3137	358
T1068	1.0000	0.0273	0.0532	10
T1070.004	0.8556	0.1273	0.2216	90
T1071.001	0.7162	0.0741	0.1343	148
T1072	0.4000	0.0066	0.0129	10
T1074.001	0.8077	0.0253	0.0490	26
T1078	0.7391	0.1183	0.2039	161
T1082	0.7236	0.0773	0.1396	123
T1083	0.7849	0.0761	0.1388	93
T1090	0.8054	0.1156	0.2022	149
T1095	0.6842	0.0576	0.1063	57
T1105	0.5287	0.1637	0.2500	261
T1106	0.4106	0.2068	0.2751	207
T1110	0.8846	0.1818	0.3016	52
T1112	0.8318	0.2580	0.3938	107
T1113	0.8980	0.3465	0.5000	49
T1140	0.3884	0.4239	0.4054	466
T1190	0.8361	0.0662	0.1227	61
T1204.002	0.6042	0.0295	0.0563	96
T1210	0.7647	0.0147	0.0288	17
T1218.011	0.9000	0.1862	0.3086	60
T1219	0.7321	0.0823	0.1480	56
T1484.001	0.9583	0.1631	0.2788	24
T1518.001	0.7297	0.0515	0.0963	37
T1543.003	0.8302	0.1429	0.2438	53
T1547.001	0.8769	0.1615	0.2727	65
T1548.002	0.8846	0.1797	0.2987	26
T1552.001	0.9091	0.0374	0.0718	22
T1557.001	0.7143	0.0276	0.0532	7
T1562.001	0.8387	0.0757	0.1388	93
T1564.001	0.9375	0.0446	0.0852	16
T1566.001	0.8778	0.1186	0.2090	90
T1569.002	0.4091	0.0350	0.0645	22
T1570	0.8644	0.1030	0.1841	59
T1573.001	0.8167	0.0515	0.0969	60
T1574.002	0.9211	0.0764	0.1411	76
Micro-average	0.6842	0.1057	0.1831	5143
Macro-average	0.7689	0.1276	0.2015	5143
Weighted-average	0.6842	0.1900	0.2680	5143

APPENDIX S
EXP 30 - BASELLM LLAMA-3.3-70B-INSTRUCT

Class	Recall	Precision	F1-Score	Support
T1003.001	0.6518	0.1798	0.2819	112
T1005	0.3279	0.0333	0.0605	61
T1012	0.4583	0.0243	0.0461	24
T1016	0.5000	0.0648	0.1148	52
T1021.001	0.5152	0.1032	0.1720	99
T1027	0.2183	0.2852	0.2473	678
T1033	0.3404	0.0667	0.1115	47
T1036.005	0.1690	0.0288	0.0493	71
T1041	0.5625	0.0548	0.0999	80
T1047	0.5065	0.0489	0.0892	77
T1053.005	0.6441	0.1434	0.2346	118
T1055	0.5971	0.1315	0.2156	278
T1056.001	0.6452	0.0691	0.1248	62
T1057	0.6104	0.0444	0.0828	77
T1059.003	0.4609	0.1500	0.2263	358
T1068	0.3000	0.0158	0.0300	10
T1070.004	0.6889	0.0435	0.0819	90
T1071.001	0.6081	0.0403	0.0755	148
T1072	0.0000	0.0000	0.0000	10
T1074.001	0.4615	0.0125	0.0243	26
T1078	0.4099	0.1564	0.2264	161
T1082	0.1382	0.0590	0.0827	123
T1083	0.3226	0.0771	0.1245	93
T1090	0.5436	0.1280	0.2072	149
T1095	0.0702	0.0303	0.0423	57
T1105	0.1226	0.1280	0.1252	261
T1106	0.0435	0.0393	0.0413	207
T1110	0.2308	0.0472	0.0784	52
T1112	0.4112	0.1189	0.1845	107
T1113	0.0612	0.0106	0.0180	49
T1140	0.0150	0.0405	0.0219	466
T1190	0.5902	0.1176	0.1962	61
T1204.002	0.4167	0.0153	0.0295	96
T1210	0.4118	0.0101	0.0196	17
T1218.011	0.5000	0.0252	0.0480	60
T1219	0.4821	0.0333	0.0622	56
T1484.001	0.3333	0.0073	0.0142	24
T1518.001	0.0270	0.0083	0.0127	37
T1543.003	0.6415	0.0660	0.1197	53
T1547.001	0.8462	0.0377	0.0722	65
T1548.002	0.9231	0.0411	0.0787	26
T1552.001	0.5000	0.0162	0.0313	22
T1557.001	0.7143	0.0032	0.0064	7
T1562.001	0.5806	0.0727	0.1292	93
T1564.001	0.1875	0.0023	0.0045	16
T1566.001	0.9222	0.0497	0.0943	90
T1569.002	0.4545	0.0134	0.0260	22
T1570	0.1356	0.0237	0.0403	59
T1573.001	0.4000	0.0340	0.0627	60
T1574.002	0.6974	0.0292	0.0561	76
Micro-average	0.3733	0.0518	0.0909	5143
Macro-average	0.4280	0.0596	0.0925	5143
Weighted-average	0.3733	0.1054	0.1338	5143

APPENDIX T
EXP 31 - LLAMA 70B OPTIMIZEDRAG FULL DESCRIPTION

Class	Recall	Precision	F1-Score	Support
T1003.001	0.8750	0.0676	0.1256	112
T1005	0.7377	0.0151	0.0295	61
T1012	0.7917	0.0192	0.0374	24
T1016	0.8846	0.0323	0.0623	52
T1021.001	0.9192	0.1210	0.2139	99
T1027	0.7935	0.1193	0.2074	678
T1033	0.9362	0.0368	0.0709	47
T1036.005	0.7042	0.0312	0.0598	71
T1041	0.8500	0.0318	0.0613	80
T1047	0.8831	0.0722	0.1335	77
T1053.005	0.8390	0.1612	0.2705	118
T1055	0.8022	0.1475	0.2492	278
T1056.001	0.9677	0.1222	0.2170	62
T1057	0.8182	0.0563	0.1054	77
T1059.003	0.6089	0.2401	0.3444	358
T1068	0.9000	0.0060	0.0118	10
T1070.004	0.8222	0.0566	0.1059	90
T1071.001	0.7838	0.0606	0.1125	148
T1072	0.4000	0.0032	0.0063	10
T1074.001	0.7308	0.0117	0.0230	26
T1078	0.8323	0.0860	0.1558	161
T1082	0.8130	0.0620	0.1151	123
T1083	0.8817	0.0405	0.0775	93
T1090	0.8121	0.0843	0.1527	149
T1095	0.8070	0.0392	0.0747	57
T1105	0.6437	0.0833	0.1476	261
T1106	0.5362	0.0847	0.1463	207
T1110	0.8846	0.0546	0.1029	52
T1112	0.8972	0.1064	0.1903	107
T1113	0.8980	0.2328	0.3697	49
T1140	0.7661	0.1445	0.2431	466
T1190	0.9672	0.0306	0.0594	61
T1204.002	0.9167	0.0326	0.0629	96
T1210	0.8824	0.0093	0.0183	17
T1218.011	0.9000	0.0473	0.0899	60
T1219	0.7857	0.0496	0.0933	56
T1484.001	0.9583	0.0372	0.0717	24
T1518.001	0.7297	0.0181	0.0352	37
T1543.003	0.8491	0.0404	0.0772	53
T1547.001	0.9077	0.0540	0.1020	65
T1548.002	0.8846	0.0368	0.0707	26
T1552.001	0.8636	0.0173	0.0338	22
T1557.001	0.7143	0.0036	0.0072	7
T1562.001	0.8817	0.0355	0.0682	93
T1564.001	1.0000	0.0106	0.0211	16
T1566.001	0.9556	0.0541	0.1023	90
T1569.002	0.8182	0.0252	0.0489	22
T1570	0.8305	0.0473	0.0895	59
T1573.001	0.8167	0.0321	0.0617	60
T1574.002	0.9079	0.0468	0.0890	76
Micro-average	0.7947	0.0570	0.1064	5143
Macro-average	0.8278	0.0612	0.1085	5143
Weighted-average	0.7947	0.0966	0.1646	5143

APPENDIX U
EXP 32 - LLAMA-3.3-70B FEWSHOTRAG FULL DESCRIPTION

Class	Recall	Precision	F1-Score	Support
T1003.001	0.8571	0.1282	0.2230	112
T1005	0.6066	0.0265	0.0508	61
T1012	0.7917	0.0378	0.0722	24
T1016	0.7885	0.0509	0.0956	52
T1021.001	0.9192	0.1970	0.3244	99
T1027	0.6873	0.1907	0.2986	678
T1033	0.9149	0.0802	0.1475	47
T1036.005	0.6479	0.0554	0.1020	71
T1041	0.8375	0.0557	0.1044	80
T1047	0.8701	0.1274	0.2222	77
T1053.005	0.8559	0.2911	0.4344	118
T1055	0.8022	0.2090	0.3316	278
T1056.001	0.9677	0.2381	0.3822	62
T1057	0.7922	0.0926	0.1658	77
T1059.003	0.5559	0.2888	0.3801	358
T1068	0.9000	0.0154	0.0304	10
T1070.004	0.7889	0.1212	0.2101	90
T1071.001	0.7365	0.0899	0.1602	148
T1072	0.5000	0.0130	0.0253	10
T1074.001	0.5769	0.0265	0.0508	26
T1078	0.7516	0.1258	0.2155	161
T1082	0.6829	0.1007	0.1755	123
T1083	0.8172	0.0748	0.1371	93
T1090	0.7651	0.1293	0.2211	149
T1095	0.6842	0.0552	0.1022	57
T1105	0.6092	0.1438	0.2326	261
T1106	0.3285	0.1450	0.2012	207
T1110	0.7885	0.1231	0.2130	52
T1112	0.4953	0.1815	0.2657	107
T1113	0.7143	0.3571	0.4762	49
T1140	0.3176	0.3466	0.3315	466
T1190	0.7049	0.0686	0.1250	61
T1204.002	0.6458	0.0600	0.1098	96
T1210	0.6471	0.0238	0.0458	17
T1218.011	0.6167	0.1028	0.1762	60
T1219	0.6071	0.0980	0.1687	56
T1484.001	0.7500	0.1364	0.2308	24
T1518.001	0.4865	0.0664	0.1169	37
T1543.003	0.6604	0.1292	0.2160	53
T1547.001	0.6923	0.1772	0.2821	65
T1548.002	0.8462	0.1732	0.2876	26
T1552.001	0.6364	0.0345	0.0654	22
T1557.001	0.2857	0.0060	0.0117	7
T1562.001	0.8065	0.1040	0.1843	93
T1564.001	0.3750	0.0193	0.0367	16
T1566.001	0.8222	0.0904	0.1628	90
T1569.002	0.4091	0.0429	0.0776	22
T1570	0.6271	0.0969	0.1678	59
T1573.001	0.6167	0.0613	0.1114	60
T1574.002	0.6711	0.0929	0.1632	76
Micro-average	0.6599	0.1090	0.1871	5143
Macro-average	0.6852	0.1140	0.1825	5143
Weighted-average	0.6599	0.1662	0.2426	5143

APPENDIX V
EXP 33 - DESCRIPTIONRAG SUMMARIZED

Class	Recall	Precision	F1-Score	Support
T1003.001	0.8571	0.0920	0.1661	112
T1005	0.8033	0.0175	0.0343	61
T1012	0.7500	0.0213	0.0413	24
T1016	0.9038	0.0394	0.0754	52
T1021.001	0.9293	0.1287	0.2260	99
T1027	0.7876	0.1362	0.2322	678
T1033	0.9149	0.0415	0.0794	47
T1036.005	0.8028	0.0335	0.0643	71
T1041	0.8375	0.0324	0.0625	80
T1047	0.8442	0.0897	0.1621	77
T1053.005	0.8644	0.1744	0.2902	118
T1055	0.8345	0.1739	0.2878	278
T1056.001	0.9516	0.1595	0.2731	62
T1057	0.8312	0.0674	0.1246	77
T1059.003	0.6592	0.1852	0.2892	358
T1068	0.9000	0.0085	0.0169	10
T1070.004	0.8556	0.0601	0.1123	90
T1071.001	0.7905	0.0624	0.1157	148
T1072	0.6000	0.0046	0.0091	10
T1074.001	0.6538	0.0103	0.0202	26
T1078	0.8447	0.0823	0.1499	161
T1082	0.8049	0.0717	0.1316	123
T1083	0.8602	0.0500	0.0945	93
T1090	0.8188	0.0885	0.1597	149
T1095	0.7193	0.0354	0.0675	57
T1105	0.6130	0.1155	0.1944	261
T1106	0.5024	0.1077	0.1773	207
T1110	0.9038	0.0790	0.1453	52
T1112	0.8505	0.1280	0.2225	107
T1113	0.8980	0.2914	0.4400	49
T1140	0.6974	0.1610	0.2616	466
T1190	0.9016	0.0297	0.0576	61
T1204.002	0.8854	0.0258	0.0502	96
T1210	0.8235	0.0088	0.0173	17
T1218.011	0.9000	0.0554	0.1044	60
T1219	0.7500	0.0473	0.0890	56
T1484.001	0.9583	0.0461	0.0880	24
T1518.001	0.7568	0.0278	0.0536	37
T1543.003	0.8679	0.0604	0.1129	53
T1547.001	0.9231	0.0833	0.1529	65
T1548.002	0.8846	0.0622	0.1162	26
T1552.001	0.9545	0.0214	0.0418	22
T1557.001	0.7143	0.0069	0.0138	7
T1562.001	0.8817	0.0557	0.1048	93
T1564.001	0.9375	0.0137	0.0271	16
T1566.001	0.9222	0.0657	0.1227	90
T1569.002	0.5909	0.0231	0.0444	22
T1570	0.8814	0.0521	0.0984	59
T1573.001	0.8333	0.0320	0.0617	60
T1574.002	0.9342	0.0435	0.0831	76
Micro-average	0.7890	0.0644	0.1191	5143
Macro-average	0.8237	0.0702	0.1233	5143
Weighted-average	0.7890	0.1056	0.1798	5143

APPENDIX W
EXP 34 - LLAMA-3.3-70B FEWSHOTRAG SUMMARIZED

Class	Recall	Precision	F1-Score	Support
T1003.001	0.8482	0.1661	0.2778	112
T1005	0.6885	0.0302	0.0579	61
T1012	0.7083	0.0483	0.0904	24
T1016	0.7692	0.0571	0.1064	52
T1021.001	0.8990	0.2253	0.3603	99
T1027	0.6770	0.1957	0.3036	678
T1033	0.8511	0.0789	0.1444	47
T1036.005	0.7183	0.0675	0.1235	71
T1041	0.8250	0.0572	0.1070	80
T1047	0.8052	0.1890	0.3062	77
T1053.005	0.8390	0.3046	0.4470	118
T1055	0.8058	0.2403	0.3702	278
T1056.001	0.9516	0.2864	0.4403	62
T1057	0.7792	0.1045	0.1843	77
T1059.003	0.5698	0.2277	0.3254	358
T1068	0.9000	0.0223	0.0435	10
T1070.004	0.8222	0.1201	0.2096	90
T1071.001	0.7568	0.0873	0.1565	148
T1072	0.4000	0.0081	0.0159	10
T1074.001	0.6154	0.0276	0.0529	26
T1078	0.7516	0.1222	0.2103	161
T1082	0.6829	0.1121	0.1927	123
T1083	0.7634	0.0825	0.1488	93
T1090	0.8054	0.1236	0.2143	149
T1095	0.6316	0.0537	0.0990	57
T1105	0.5517	0.1773	0.2684	261
T1106	0.4010	0.1697	0.2385	207
T1110	0.8846	0.1479	0.2534	52
T1112	0.7850	0.2054	0.3256	107
T1113	0.8776	0.3739	0.5244	49
T1140	0.3047	0.3326	0.3180	466
T1190	0.8525	0.0673	0.1247	61
T1204.002	0.8333	0.0412	0.0785	96
T1210	0.8235	0.0182	0.0356	17
T1218.011	0.8833	0.1380	0.2387	60
T1219	0.7143	0.0830	0.1487	56
T1484.001	0.9167	0.1140	0.2028	24
T1518.001	0.5946	0.0853	0.1492	37
T1543.003	0.8113	0.1243	0.2155	53
T1547.001	0.8923	0.2021	0.3295	65
T1548.002	0.8846	0.1544	0.2629	26
T1552.001	0.9091	0.0357	0.0687	22
T1557.001	0.7143	0.0198	0.0385	7
T1562.001	0.8387	0.1074	0.1905	93
T1564.001	0.7500	0.0421	0.0797	16
T1566.001	0.8889	0.0977	0.1760	90
T1569.002	0.4545	0.0386	0.0712	22
T1570	0.8136	0.0943	0.1690	59
T1573.001	0.8000	0.0550	0.1030	60
T1574.002	0.9079	0.0833	0.1527	76
Micro-average	0.6947	0.1106	0.1908	5143
Macro-average	0.7591	0.1209	0.1950	5143
Weighted-average	0.6947	0.1704	0.2516	5143

APPENDIX X
EXP 35 - DEEPSEEK OPTIMIZEDRAGFEWSHOTSUMMARIZED

Class	Recall	Precision	F1-Score	Support
T1003.001	0.8929	0.0645	0.1203	112
T1005	0.6557	0.0127	0.0248	61
T1012	0.7500	0.0138	0.0271	24
T1016	0.8077	0.0246	0.0477	52
T1021.001	0.9091	0.0862	0.1575	99
T1027	0.8068	0.0962	0.1719	678
T1033	0.8085	0.0310	0.0598	47
T1036.005	0.7746	0.0313	0.0602	71
T1041	0.8125	0.0232	0.0452	80
T1047	0.8182	0.0560	0.1048	77
T1053.005	0.8559	0.1146	0.2022	118
T1055	0.8561	0.1289	0.2241	278
T1056.001	0.9516	0.0827	0.1523	62
T1057	0.7792	0.0519	0.0972	77
T1059.003	0.5810	0.1390	0.2244	358
T1068	0.9000	0.0059	0.0117	10
T1070.004	0.8444	0.0398	0.0761	90
T1071.001	0.7500	0.0445	0.0841	148
T1072	0.4000	0.0018	0.0036	10
T1074.001	0.6923	0.0092	0.0181	26
T1078	0.7950	0.0657	0.1213	161
T1082	0.7642	0.0526	0.0984	123
T1083	0.7849	0.0486	0.0916	93
T1090	0.8389	0.0693	0.1280	149
T1095	0.8246	0.0251	0.0486	57
T1105	0.5709	0.1049	0.1772	261
T1106	0.5266	0.0686	0.1214	207
T1110	0.9038	0.0389	0.0747	52
T1112	0.8131	0.0774	0.1413	107
T1113	0.8776	0.1086	0.1933	49
T1140	0.8219	0.0887	0.1601	466
T1190	0.8525	0.0161	0.0317	61
T1204.002	0.8125	0.0227	0.0441	96
T1210	0.7647	0.0058	0.0115	17
T1218.011	0.8833	0.0341	0.0656	60
T1219	0.6964	0.0243	0.0470	56
T1484.001	0.9583	0.0223	0.0436	24
T1518.001	0.6757	0.0133	0.0261	37
T1543.003	0.8113	0.0325	0.0625	53
T1547.001	0.8769	0.0535	0.1008	65
T1548.002	0.9231	0.0300	0.0580	26
T1552.001	0.9091	0.0177	0.0348	22
T1557.001	0.8571	0.0029	0.0058	7
T1562.001	0.8925	0.0331	0.0637	93
T1564.001	1.0000	0.0075	0.0148	16
T1566.001	0.9444	0.0304	0.0589	90
T1569.002	0.5000	0.0127	0.0249	22
T1570	0.8475	0.0423	0.0806	59
T1573.001	0.7667	0.0204	0.0397	60
T1574.002	0.9211	0.0346	0.0667	76
Micro-average	0.7818	0.0439	0.0831	5143
Macro-average	0.8012	0.0453	0.0830	5143
Weighted-average	0.7818	0.0720	0.1284	5143

APPENDIX Y
EXP 36 - DEEPSEEK OPTIMIZEDRAG FEWSHOTRAG + PROMPT ENGINEERING

Class	Recall	Precision	F1-Score	Support
T1003.001	0.8839	0.0651	0.1213	112
T1005	0.6885	0.0144	0.0281	61
T1012	0.6667	0.0127	0.0250	24
T1016	0.6731	0.0225	0.0435	52
T1021.001	0.8889	0.0909	0.1649	99
T1027	0.7935	0.0999	0.1775	678
T1033	0.8511	0.0352	0.0676	47
T1036.005	0.7042	0.0302	0.0580	71
T1041	0.8125	0.0254	0.0492	80
T1047	0.8312	0.0612	0.1141	77
T1053.005	0.8559	0.1157	0.2038	118
T1055	0.8094	0.1255	0.2173	278
T1056.001	0.9516	0.0803	0.1481	62
T1057	0.7922	0.0526	0.0987	77
T1059.003	0.5335	0.1370	0.2180	358
T1068	0.9000	0.0061	0.0121	10
T1070.004	0.8222	0.0422	0.0803	90
T1071.001	0.7703	0.0475	0.0895	148
T1072	0.5000	0.0024	0.0048	10
T1074.001	0.5385	0.0075	0.0148	26
T1078	0.7764	0.0694	0.1275	161
T1082	0.7317	0.0526	0.0981	123
T1083	0.7957	0.0512	0.0963	93
T1090	0.8658	0.0765	0.1406	149
T1095	0.7895	0.0243	0.0471	57
T1105	0.5632	0.1120	0.1869	261
T1106	0.4734	0.0667	0.1169	207
T1110	0.9038	0.0395	0.0757	52
T1112	0.7290	0.0772	0.1395	107
T1113	0.8571	0.1114	0.1972	49
T1140	0.7661	0.0830	0.1498	466
T1190	0.8525	0.0168	0.0330	61
T1204.002	0.7917	0.0235	0.0456	96
T1210	0.7647	0.0062	0.0123	17
T1218.011	0.8833	0.0330	0.0636	60
T1219	0.7143	0.0264	0.0509	56
T1484.001	0.9583	0.0241	0.0469	24
T1518.001	0.7297	0.0146	0.0287	37
T1543.003	0.8113	0.0339	0.0650	53
T1547.001	0.9077	0.0576	0.1084	65
T1548.002	0.8462	0.0317	0.0611	26
T1552.001	0.9091	0.0177	0.0347	22
T1557.001	0.8571	0.0031	0.0061	7
T1562.001	0.8710	0.0335	0.0644	93
T1564.001	1.0000	0.0079	0.0156	16
T1566.001	0.9333	0.0311	0.0602	90
T1569.002	0.3636	0.0100	0.0194	22
T1570	0.8305	0.0435	0.0827	59
T1573.001	0.7500	0.0210	0.0408	60
T1574.002	0.9211	0.0342	0.0659	76
Micro-average	0.7601	0.0447	0.0845	5143
Macro-average	0.7843	0.0462	0.0844	5143
Weighted-average	0.7601	0.0729	0.1292	5143

APPENDIX Z
EXP 37 - DEEPSEEK-R1-70B BASELLM

Class	Recall	Precision	F1-Score	Support
T1003.001	0.9464	0.0131	0.0258	112
T1005	0.6230	0.0051	0.0102	61
T1012	0.4167	0.0016	0.0033	24
T1016	0.4423	0.0053	0.0105	52
T1021.001	0.6162	0.0114	0.0225	99
T1027	0.5457	0.0644	0.1152	678
T1033	0.5957	0.0048	0.0096	47
T1036.005	0.5775	0.0075	0.0147	71
T1041	0.7000	0.0090	0.0178	80
T1047	0.6104	0.0085	0.0167	77
T1053.005	0.8305	0.0156	0.0306	118
T1055	0.8597	0.0332	0.0640	278
T1056.001	0.8710	0.0090	0.0178	62
T1057	0.6104	0.0086	0.0171	77
T1059.003	0.4804	0.0343	0.0640	358
T1068	0.5000	0.0012	0.0024	10
T1070.004	0.8111	0.0084	0.0167	90
T1071.001	0.7162	0.0135	0.0266	148
T1072	0.3000	0.0008	0.0016	10
T1074.001	0.6538	0.0032	0.0064	26
T1078	0.5776	0.0281	0.0535	161
T1082	0.3984	0.0156	0.0301	123
T1083	0.6022	0.0161	0.0314	93
T1090	0.5369	0.0319	0.0603	149
T1095	0.1053	0.0037	0.0072	57
T1105	0.4253	0.0571	0.1007	261
T1106	0.0580	0.0111	0.0186	207
T1110	0.5192	0.0250	0.0478	52
T1112	0.3458	0.0269	0.0500	107
T1113	0.1837	0.0081	0.0155	49
T1140	0.0579	0.0384	0.0462	466
T1190	0.6230	0.0164	0.0320	61
T1204.002	0.6562	0.0140	0.0274	96
T1210	0.2353	0.0052	0.0102	17
T1218.011	0.3500	0.0224	0.0420	60
T1219	0.4643	0.0311	0.0584	56
T1484.001	0.7917	0.0075	0.0148	24
T1518.001	0.0541	0.0198	0.0290	37
T1543.003	0.6415	0.0268	0.0515	53
T1547.001	0.7692	0.0361	0.0690	65
T1548.002	0.6923	0.0657	0.1200	26
T1552.001	0.5909	0.0120	0.0236	22
T1557.001	0.5714	0.0039	0.0077	7
T1562.001	0.6022	0.1451	0.2338	93
T1564.001	0.5000	0.0083	0.0163	16
T1566.001	0.8111	0.0581	0.1084	90
T1569.002	0.0909	0.0058	0.0109	22
T1570	0.1356	0.0279	0.0462	59
T1573.001	0.2333	0.0170	0.0317	60
T1574.002	0.6316	0.0380	0.0717	76
Micro-average	0.5059	0.0159	0.0309	5143
Macro-average	0.5192	0.0216	0.0392	5143
Weighted-average	0.5059	0.0322	0.0565	5143

APPENDIX
EXP 38 - LLAMA3.3-70B OPTIMIZEDRAG 10 TECHNIQUES

Class	Recall	Precision	F1-Score	Support
T1003.001	0.8839	0.1193	0.2102	112
T1005	0.7213	0.0167	0.0326	61
T1012	0.7917	0.0268	0.0518	24
T1016	0.8462	0.0327	0.0629	52
T1021.001	0.9293	0.1691	0.2862	99
T1027	0.7802	0.1369	0.2329	678
T1033	0.9362	0.0393	0.0755	47
T1036.005	0.7042	0.0412	0.0778	71
T1041	0.8625	0.0380	0.0728	80
T1047	0.8831	0.1347	0.2337	77
T1053.005	0.8644	0.2198	0.3505	118
T1055	0.7914	0.1911	0.3079	278
T1056.001	0.9677	0.2532	0.4013	62
T1057	0.8182	0.0613	0.1141	77
T1059.003	0.5726	0.2553	0.3531	358
T1068	1.0000	0.0115	0.0227	10
T1070.004	0.8444	0.0837	0.1523	90
T1071.001	0.7973	0.0591	0.1100	148
T1072	0.5000	0.0054	0.0107	10
T1074.001	0.8077	0.0128	0.0252	26
T1078	0.7888	0.0946	0.1690	161
T1082	0.8211	0.0629	0.1169	123
T1083	0.8710	0.0508	0.0960	93
T1090	0.8389	0.0930	0.1674	149
T1095	0.7544	0.0393	0.0747	57
T1105	0.6398	0.0907	0.1588	261
T1106	0.4831	0.1481	0.2268	207
T1110	0.9038	0.1516	0.2597	52
T1112	0.8692	0.1615	0.2723	107
T1113	0.8980	0.3492	0.5029	49
T1140	0.7854	0.1927	0.3095	466
T1190	0.9016	0.0349	0.0672	61
T1204.002	0.8438	0.0311	0.0600	96
T1210	0.8824	0.0095	0.0188	17
T1218.011	0.9167	0.1056	0.1893	60
T1219	0.7679	0.0660	0.1215	56
T1484.001	0.9583	0.0865	0.1586	24
T1518.001	0.7297	0.0271	0.0523	37
T1543.003	0.8679	0.0732	0.1351	53
T1547.001	0.9077	0.0787	0.1448	65
T1548.002	0.8846	0.0780	0.1433	26
T1552.001	0.8636	0.0228	0.0444	22
T1557.001	0.7143	0.0113	0.0222	7
T1562.001	0.8387	0.0516	0.0971	93
T1564.001	0.9375	0.0176	0.0345	16
T1566.001	0.9333	0.0747	0.1383	90
T1569.002	0.8182	0.0410	0.0781	22
T1570	0.7966	0.0733	0.1343	59
T1573.001	0.7833	0.0365	0.0697	60
T1574.002	0.9211	0.0575	0.1082	76
Micro-average	0.7859	0.0732	0.1339	5143
Macro-average	0.8245	0.0864	0.1471	5143
Weighted-average	0.7859	0.1236	0.2035	5143

APPENDIX

EXP 39 - DEEPSEEK-R1-70B OPTIMIZED RAG NO TECHNIQUES

Class	Recall	Precision	F1-Score	Support
T1003.001	0.7500	0.2927	0.4211	112
T1005	0.5082	0.0292	0.0552	61
T1012	0.5417	0.0915	0.1566	24
T1016	0.6731	0.0716	0.1294	52
T1021.001	0.8081	0.4061	0.5405	99
T1027	0.5059	0.2327	0.3188	678
T1033	0.6596	0.0909	0.1598	47
T1036.005	0.6197	0.0506	0.0935	71
T1041	0.7375	0.0742	0.1349	80
T1047	0.6623	0.4048	0.5025	77
T1053.005	0.6780	0.3571	0.4678	118
T1055	0.5791	0.2771	0.3749	278
T1056.001	0.7419	0.3833	0.5055	62
T1057	0.6364	0.1701	0.2685	77
T1059.003	0.3659	0.2729	0.3126	358
T1068	0.8000	0.0297	0.0573	10
T1070.004	0.6778	0.1517	0.2480	90
T1071.001	0.5743	0.1257	0.2063	148
T1072	0.2000	0.0043	0.0084	10
T1074.001	0.5769	0.0286	0.0545	26
T1078	0.6957	0.1384	0.2309	161
T1082	0.5935	0.0955	0.1646	123
T1083	0.6989	0.1272	0.2152	93
T1090	0.7047	0.3344	0.4536	149
T1095	0.5439	0.0909	0.1558	57
T1105	0.4828	0.1538	0.2333	261
T1106	0.3430	0.2198	0.2679	207
T1110	0.6538	0.2048	0.3119	52
T1112	0.5701	0.2430	0.3408	107
T1113	0.7347	0.5294	0.6154	49
T1140	0.5386	0.3786	0.4446	466
T1190	0.7213	0.0753	0.1364	61
T1204.002	0.7083	0.0362	0.0688	96
T1210	0.5294	0.0203	0.0390	17
T1218.011	0.6667	0.2222	0.3333	60
T1219	0.6429	0.0909	0.1593	56
T1484.001	0.8750	0.1858	0.3066	24
T1518.001	0.5135	0.0660	0.1169	37
T1543.003	0.7170	0.1919	0.3028	53
T1547.001	0.8000	0.2080	0.3302	65
T1548.002	0.9231	0.2553	0.4000	26
T1552.001	0.6364	0.0388	0.0731	22
T1557.001	0.5714	0.0494	0.0909	7
T1562.001	0.7312	0.1151	0.1988	93
T1564.001	0.6875	0.0364	0.0692	16
T1566.001	0.8111	0.1017	0.1807	90
T1569.002	0.5455	0.0822	0.1429	22
T1570	0.6102	0.1622	0.2562	59
T1573.001	0.6333	0.0538	0.0992	60
T1574.002	0.8026	0.1210	0.2103	76
Micro-average	0.5915	0.1329	0.2170	5143
Macro-average	0.6396	0.1635	0.2393	5143
Weighted-average	0.5915	0.2150	0.2946	5143