Fine-Tuning Large Language Models for Digital Forensics: Case Study and General Recommendations

GAËTAN MICHELET, Chair for Cybersecurity, University of Augsburg, Augsburg, Germany and School of Criminal Justice, University of Lausanne, Lausanne, Switzerland
HANS HENSELER, Netherlands Forensic Institute, The Hague, Netherlands and Leiden University of Applied Sciences, Leiden, Netherlands
HARM VAN BEEK, Netherlands Forensic Institute, The Hague, Netherlands and Open Universiteit, Heerlen, Netherlands
MARK SCANLON, University College Dublin, Dublin, Ireland
FRANK BREITINGER, Chair for Cybersecurity, University of Augsburg, Augsburg, Germany

Large Language Models (LLMs) have rapidly gained popularity in various fields, including Digital Forensics (DF), where they offer the potential to accelerate investigative processes. Although several studies have explored LLMs for tasks such as evidence identification, artifact analysis, and report writing, fine-tuning models for specific forensic applications remains underexplored. This article addresses this gap by proposing recommendations for fine-tuning LLMs tailored to DF tasks. A case study on chat summarization is presented to showcase the applicability of the recommendations, where us evaluate multiple fine-tuned models to assess their performance. The study concludes with sharing the lessons learned from the case study.

CCS Concepts: • **Applied computing** → **Computer forensics**;

Additional Key Words and Phrases: Digital Forensics Investigation, Fine-tuning, Local Large Language Models (LLM), Chat Logs Summarization, Reporting Automation

ACM Reference format:
Gaëtan Michelet, Hans Henseler, Harm van Beek, Mark Scanlon, and Frank Breitinger. 2025. Fine-Tuning Large Language Models for Digital Forensics: Case Study and General Recommendations. *Digit. Threat. Res. Pract.* 6, 4, Article 21 (December 2025), 18 pages.
https://doi.org/10.1145/3748264

1 Introduction

Large Language Models (LLMs) have impacted many fields, including **Digital Forensics (DF)**, where they have been used for evidence analysis [10] and report writing [19]. However, despite their popularity and to the best of

The Mobi.Doc mobility grant from the University of Lausanne made this project and collaboration possible.
Authors' Contact Information: Gaëtan Michelet (corresponding author), Chair for Cybersecurity, University of Augsburg, Augsburg, Germany and School of Criminal Justice, University of Lausanne, Lausanne, Switzerland; e-mail: gaetan.michelet@uni-a.de; Hans Henseler, Netherlands Forensic Institute, The Hague, Netherlands and Leiden University of Applied Sciences, Leiden, Netherlands; e-mail: h.henseler@nfi.nl; Harm van Beek, Netherlands Forensic Institute, The Hague, Netherlands and Open Universiteit, Heerlen, Netherlands; e-mail: harm.van.beek@nfi.nl; Mark Scanlon, University College Dublin, Dublin, Ireland; e-mail: mark.scanlon@ucd.ie; Frank Breitinger, Chair for Cybersecurity, University of Augsburg, Augsburg, Germany; e-mail: frank.breitinger@uni-a.de.

ACM 2576-5337/2025/12-ART21
https://doi.org/10.1145/3748264

our knowledge, research has focused on general-purpose LLMs rather than models tailored to forensic challenges. This article addresses this gap by exploring the fine-tuning of LLMs for forensic applications. Fine-tuning enables LLMs to be tailored for specific tasks, enhancing the precision and relevance of their output content or format. This is particularly valuable for tasks that are not well supported or where results are unsatisfactory. However, challenges persist in DF, including limited datasets, restricted computational resources for training complex models, and confidentiality constraints that prevent the use of remote computing power, along with a shortage of expertise in model training. Consequently, this article addresses three questions: (1) How can small local LLMs be fine-tuned to address the unique challenges and requirements of DF?; (2) How can we evaluate the results to ensure that the fine-tuning process was successful and identify the best-performing model?; and (3) How beneficial is the fine-tuning process for improving model performance and should practitioners adopt it in forensic workflows?

To address these questions, we review the literature (Section 2) and then provide considerations and recommendations for fine-tuning LLMs in the context of DF, emphasizing aspects such as task definition, base model selection, and dataset preparation (Section 3). Next, we apply these principles to the concrete example of *chat summarization* in Section 4, which is a frequent and time-intensive task, illustrating the practicality of our approach. Given the tremendous number of parameters involved in the fine-tuning process, only a small selection of them has been empirically studied (the rest is generally discussed in the recommendations section). We compare the results of the fine-tuned models against their base counterparts in Section 5 followed by the lessons learned (Section 6). Section 7 concludes this work. In summary, the contributions are:

- We provide recommendations, accessible to DF professionals rather than AI experts. These considerations help determine whether a task can be automated with an LLM and outline what is needed for fine-tuning.
- We apply the proposed fine-tuning framework to a real-world forensic task, describing the task definition, dataset development, model training, and performance comparison.
- We share insights from the fine-tuning process, covering computational power issues, data challenges, and evaluation methods that can be applied to other DF tasks. We also discuss the costs, limits and performance increase tradeoff to determine if the fine-tuning process is beneficial or not.
- We provide the developed datasets and the fine-tuned models to the academic community for further experimentation.

Disclaimer. This article *does not* aim to: (1) provide a step-by-step tutorial for creating datasets and fine-tuning LLMs; (2) provide the best values for the different fine-tuning parameters; (3) present an extensive empirical analysis of the impact of each parameter involved in the fine-tuning process (dataset creation, fine-tuning); or (4) publish ready to use models. While key elements of the fine-tuning process are developed and explained, prior knowledge or experience in deep learning is recommended for the reader.

2 Background and Related Work

2.1 LLMs in the DF Context

The usage of LLMs in a DF context can be divided into two categories. The first focuses on analyzing artifacts from employing the use of LLMs, such as the forensic analysis of the Multi-Agent LLM Platform, AutoGen, [32] and OpenAI's ChatGPT mobile application [3]. These works are considered out of scope. The second aspect surrounds leveraging LLMs to enhance the investigative process. For example, Henseler and van Beek [10] explored the use of ChatGPT as a copilot during the investigation, helping to create queries for the DF as a Service system, Hansken, summarize communications, visualize search results and perform analysis. Michelet and Breitinger [19] tested ChatGPT and Llama-2 capabilities to automate forensic report writing, concluding that LLMs can accelerate the writing but require careful validation. These works are complemented by general studies that discuss potential capabilities, limits, and risks related to the use of LLMs for DF [2, 15, 24, 25]. More details are provided by Wickramasekara et al. [33], who explored various areas in which LLMs can be applied in DF.

2.2 Fine-Tuning LLMs Basics and Applications in Other Domains

This section covers concepts that are directly relevant to the scope of this article, while a comprehensive discussion of all aspects of fine-tuning falls outside our current focus. For more detailed discussions on that topic, we recommend the work by Naveed et al. [21] and Minaee et al. [20].

There are several fine-tuning methods, such as **Supervised Fine-Tuning (SFT)** or **Direct Preference Optimization (DPO)**. During an SFT process, samples composed of an input (prompt/instruction) and an expected output (answer to the prompt/instruction) are fed to the model. The LLM uses the prompt to predict an output using its current parameters (by iteratively predicting the next token until a special "stop" token or the limit of new tokens is reached). A function, usually cross-entropy for LLMs, is used to compute the loss, which measures the difference between the prediction and the expected output. The model then adapts the set of trainable parameters to minimize that loss. DPO optimizes the model based on user preferences or feedback rather than relying solely on predefined reward functions, e.g., users rank multiple responses to the same prompt. This approach helps align the model's behavior more closely with desired outcomes by learning from human evaluations or specific task requirements.

Another important choice is the set of targeted parameters during the training process. Fine-tuning all parameters is expensive and may not be feasible for many institutions. A more effective setup, called **Parameter-Efficient Fine-Tuning (PEFT)**, proposes to target a subset or a completely new set of parameters during the fine-tuning process. Hu et al. [11] introduced a PEFT technique called **Low Rank Adaptation (LoRA)**. LoRA freezes the pre-trained model weights and injects trainable rank decomposition matrices into each layer of the Transformer architecture, reducing the number of trainable parameters for downstream tasks. This results in a new file named adapters, containing the set of new parameters. This file is then added on top of the base model during the inference process. A very popular PEFT technique called **Quantized Low Rank Adapters (QLoRA)**, derived from LoRA, uses fewer resources while remaining performant [1].

Fine-tuning LLMs have been explored and tested in different disciplines. For example, Yue et al. [35] fine-tuned LLMs for legal-related tasks, Shestov et al. [26] for vulnerability detection in source code, Tinn et al. [30] for biomedical tasks, and Huang et al. [12] for automatic program repair. However, it has not yet been explored in DF.

2.3 Datasets for Dialogue Summarization

Dialogue summarization datasets consist of a dialogue and a manually created summary thereof. While several of these datasets exist, many are not suitable as they are not chat-like conversations. For instance, Khalman et al. [16] created a dataset using forum conversations, and Feigenblat et al. [5] generated a dataset based on tweets from customer care services. Feng et al. [6] presented a more complete list of datasets. This study uses the SAMSum dataset that "contains natural messenger-like conversations created and written down by linguists fluent in English" [8].

2.4 Fine-Tuning Models for Dialogue Summarization

Different methods (including fine-tuning) were explored to solve the long dialogue summarization problem that was present before the introduction of long-context windows [13, 37, 39]. Tang et al. [28] tested a new fine-tuning approach to help reduce inaccuracies and hallucinations. Zhao et al. [40] stated that dialogue summarization is often domain-specific and not generalizable, before exploring a new fine-tuning method to solve this issue.

Yun et al. [36] ran fine-tuning for dialogue summarization in a customer service context. They experimented with general and specific datasets, demonstrating that domain-specific training datasets helped improve the results.

Some also proposed new models. Zhong et al. [41] created DialogLM using different training techniques. Xiao et al. [34] and Suri et al. [27] used recent models as base models and fine-tuned, respectively, Baichuan and Bio-ClinicalBERT for dialogue summarization.

Tian et al. [29] used a mixture of task-specific experts based on an LLM. They obtained better results than others by letting the model choose the best expert based on the current task.

A survey conducted by Feng et al. [6] provides an overview of what work has been done and what datasets have been created for dialogue summarization.

2.5 Guidelines in the Literature

In the literature, two studies have been identified that present recommendations and guidelines for fine-tuning LLMs. First, Mathav et al. [18] provide recommendations for fine-tuning LLMs for enterprises. They present different components of fine-tuning and provide experiments. They fine-tuned Llama-2 for text and code with different parameters such as the quantization, the LoRA settings, or the dataset used. They also offer a list of guidelines. Second, Qin et al. [23] empirically studied different fine-tuning settings and provided recommendations and guidelines for fine-tuning LLMs based on their results.

3 Recommendations for Fine-Tuning LLMs for DF Investigations

Fine-tuning LLMs requires consideration of five essential areas: (1) the task, (2) the base model, (3) the dataset, (4) the fine-tuning process, and (5) the evaluation. Choices are made in each area, with a direct impact on the quality/applicability of the resulting model. This section details each area and provides recommendations to minimize risks and maximize the process's success rate.

Methodology. During the case study presented in Section 4, several challenges arose, prompting difficult decisions and extensive brainstorming. The recommendations presented here are based on the insights and knowledge gained from these discussions.

3.1 The Task

First, an appropriate task must be identified, i.e., a task that can be automated. The task selection significantly shapes and impacts the other four factors. To meet the tool explainability requirements typically required in the courtroom more easily, a *simpler or more deterministic software* to solve the task, such as using a rule-based system, should be preferred if possible. A clear understanding of the task guides the choice of the base model. It is important to recognize that not all base models perform equally well for different tasks, and this also holds for the fine-tuned model. Tasks that *closely align with next-word prediction*, such as summarizing or identifying relevant elements in a text, are likely to see significant improvement. A major risk with task selection is the possibility that the task may not be automatable. To detect this early and avoid wasting precious resources (in particular, for smaller laboratories), *a preliminary testing of the chosen task* using easily available LLMs (locally or online for non-sensitive data) should be conducted. This evaluation may be done formally (using automatically or manually computed metrics), and/or informally (manual review). Although the former is required at the end of the fine-tuning process to measure the improvement of the fine-tuned model, we consider the latter sufficient for this step.

3.2 The Base Model

Building an LLM from scratch is resource-intensive. Consequently, many models are released by tech companies such as OpenAI (with GPT), Meta (with LLaMA), Google (with Gemma), or Mistralai (with Mistral).[1] These models are released in different flavors, where we focus on three aspects:

— *Number of parameters*: For small forensic laboratories with limited resources (e.g., a single 24 GB GPU), we recommend avoiding *models with more than 8 billion parameters* for fine-tuning, as larger models require more computational power.[2]

[1]Some of these models are open source (Llama, Gemma, or Mistral), while others are not published and are made available through controlled interfaces only (GPT).

[2]While inferences can be run on 24 GB GPU using 11B parameters LLMs, the fine-tuning process is more memory-expensive.

— *Context size*: If the task requires a large amount of data as input (which is sometimes the case with DF tasks), the maximum prompt size must be as lengthy as possible, and selecting a *model with a longer context size* is crucial.
— *Training*: Models can be (pre-)fine-tuned for specific use cases, such as chatbot conversations, which can affect their suitability for the targeted tasks.

The choice of the base model affects the quality variation in two ways: it determines the performance of the base model and the potential for improvement after fine-tuning. Since it can be difficult to predict how a model will perform on a given task and since the waste of resources should be avoided in forensic laboratories, it is *recommended to test the base performance* before commencing the fine-tuning. If the base model performs well, fine-tuning may not even be necessary. In contrast, if it performs poorly, a different model may be better. And if it performs moderately, fine-tuning may help to reach an acceptable level.

Furthermore, it is recommended to use *recent models*, as companies continually improve and often release more powerful and efficient versions. *Open source models with documentation* about their training process are preferable, particularly for our use, as understanding how a model was trained is important to ensure admissibility in court, even if the entire dataset is not publicly available.

If interaction between the user and the model is expected, it is often the preferred option to use an *Instruct fine-tuned model*, e.g., investigation copilot or assistant. These models are optimized to understand and respond in an interactive, structured way. If a raw model is selected for fine-tuning in interactive settings, a chat template with special tokens must be implemented to manage stops and role changes.

3.3 The Dataset

The *quantity and quality* of the data affect the performance of the fine-tuned model and the resulting quality variation. Generally, higher-quality datasets lead to better fine-tuned models and larger datasets lead to greater quality variation.

It is also crucial to consider the *source of the data*. Using real case data means that the resulting model cannot be shared and must be trained locally. Publicly available datasets are more flexible, but may not reflect real-world scenarios. If there is no appropriate *dataset available*, which is often the case in DF, a new one can be created manually, automatically (e.g., generating synthetic data using generative AI), or through a combination of both methods. An often overlooked approach is to use an existing dataset and extend it with real or generated data. Each dataset sample should include a prompt (from the *user* role) relevant to the task, the related input data, and the expected model output (from the *assistant* role). For example:

```
[
 {
  “content”: “What is the offense in the following text: ’If you do not deliver on
  time, I will kill you.’”,
 “role”:“user”
 },
 {
  “content”: “Threat to kill”,
  “role”: “assistant”
 }
]
```

where “What is...text:” is the prompt, “If...kill you” is the input data, and “Threat to kill” is the expected model output. If interaction with the model is involved, *each step of the conversation* should have both the user’s prompt and the model’s response, i.e., a list of queries.

The fine-tuning samples are distributed over three subsets: (1) the training subset used by the model to compute the loss and adjust its parameters, (2) the validation subset used to control the generalizability of the learning and limit the overfitting, and (3) the testing subset used to evaluate the results. Conclusions based on this testing subset are run using the base and fine-tuned models. Performance is then measured using the selected metrics and the quality variation between the base and fine-tuned models can be measured.

3.4 The Fine-Tuning Method/Process

During the fine-tuning process, the model adjusts its parameters based on the training subset. The chosen approach, the set of trainable parameters and the loss function all impact the performance of the fine-tuned model and the quality variation. *SFT is generally an effective choice* for fine-tuning an LLM for specific tasks. If computational resources are limited, a PEFT approach is recommended, in particular *LoRA or QLoRA*. Finally, *the cross-entropy loss function* is applicable in most cases.

Another important factor is how the loss function is computed. As explained previously, the purpose of the model is to predict the following token correctly, and to do so, LLMs adjust their parameters based on the cross-entropy loss. This loss can be computed in two ways: (1) The prompt is masked during the loss computation but is still used to serve as context (answer only); and (2) The loss is computed on all tokens from the prompt and the expected answer (prompt + answer). As demonstrated in Section 5.1, computing the *loss on* “*answer only*” seems to perform better for *chat summarization*. Finally, selecting the right hyperparameters is crucial. Key ones include:

— *Number of epochs*: The number of times the complete training dataset is used in the process.
— *Learning rate*: Determines how much the parameters are adjusted in response to the estimated error gradient.
— *Training batch size*: Determines how many samples are processed before updating the model parameters.

Ideally, if the laboratory resources allow it, *different configurations* are tested to identify the best settings. A potential risk is making errors in the fine-tuning configurations, which might lead to an incorrectly fine-tuned model. Although the model may still be useful, it could differ from the initial plan. Another risk is an interruption during the process, which could result in the loss of the current state of the model. Without a checkpointing system, the fine-tuning process needs to be restarted from scratch.

3.5 The Evaluation Metrics

Successful fine-tuning is evaluated by comparing outputs from identical prompts on the base and fine-tuned models against expected results using *appropriate evaluation metrics*. One possibility is human evaluators. Manually reviewing prompts, responses, or entire chats is time-consuming and resource-intensive. If manual evaluation is necessary, it is recommended to use a *double-blind review protocol* with *clear and explicit metrics* to minimize bias and variability between evaluators. For large datasets, an *automated evaluation* should be favored. Depending on the task, an appropriate metric must be selected. Accuracy, precision, recall, or F1-score may, for example, be used for classification tasks. For summarization tasks, metrics such as ROUGE [17], BLEU [22], or BERTScore [38] have established themselves (details see Section 4.5). If an appropriate metric is not available, one may have to create one.

3.6 Fine-Tuning, an Iterative Process?

As each decision impacts the fine-tuning, it should be seen as an iterative process, allowing improvements in parameter choices and steps based on evaluation insights. Obtaining optimal results on the first attempt is rare, especially without prior fine-tuning experience. Therefore, *restarting the process after the initial iteration* can lead to

improvements. This can mean changing the base model, adjusting the dataset, or modifying the hyperparameters. This could also be done before the end of the evaluation if one realizes that better options are available.

4 Case Study

This section follows the recommendations previously presented and defines specific tasks, selects appropriate language models, and constructs a suitable dataset. We then detail the training process and explain the methods used to evaluate and compare the models. The results are analyzed in Section 5.

Our central research question is: *How can we improve chat summarization using fine-tuned language models?* Therefore, we fine-tuned three base LLMs on three chat-log summarization tasks using samples from both manually and automatically generated datasets. Note that the purpose of this case study is not to discuss the impact of each parameter, but rather to provide insight into the fine-tuning process.

The scripts used during this experiment (dataset generation, inference process, and evaluation), as well as additional information about the dataset generation, are available at https://github.com/Michelet-Gaetan/Fine-tuning-LLMs-for-DF-tasks. The scripts used for the fine-tuning process are available in the "alignment book" framework provided by members of the *huggingface* community [31].

4.1 The Tasks

The "chat log summarization" task was chosen for three reasons. First, its importance to investigations [9]. It is frequently encountered by investigators, and the information that chat messages convey can be crucial. Second, reading chat messages, determining their relevance and summarizing them for the report are time-consuming tasks. This is particularly true if the device was extensively used and if multiple messaging applications exist. Lastly, experiments with ChatGPT 3.5 showed promising results, while smaller (local) models lacked quality [19]. Therefore, we considered this task an appropriate candidate for this case study.

The following aspects must be considered: First, chat histories can be extremely long and comprise thousands of messages. They include many different topics, and conversations may have two or more participants (group chats), where each topic consists of a variety of mostly consecutive messages. The amount of time between the discussed topics can vary, as well as the amount of time between messages related to a single topic. Consequently, we decided to narrow down the focus and define the following three tasks (increasing complexity):

Task 1: The first task is what we call *single topic summarization*. The messages related to a single topic are provided to the model, along with the user who sends each message. The model must provide a summary of the discussion.

Task 2: The second task is *topic of interest identification and summarization (2 steps)*. First, chat logs containing messages related to several consecutive topics are provided to the model, along with the sending timestamp and the sender of each message. The model must first provide a list of each topic discussed, including a 1-sentence summary and the timestamps of the first and last messages relating to this topic. During the second step, the user points out the topic of interest for the investigation and asks for a detailed summary of that topic (second output).

Task 3: The third task is *topic of interest identification and summarization (1-step)*. The model receives two distinct elements in the same prompt: (1) the crime investigated, and (2) chat logs containing messages related to consecutive topics, along with the sending timestamp and sender of the message. The model must identify the topic of interest, the timestamp of the first and last messages related to that topic, and provide a summary of the messages related to that topic. Note that it is also possible that none of the discussed topics is related to the investigation of the crime. In that case, the model is expected to answer that none of the topics is of interest.

The tests showed good preliminary results for task 1, with potential for improvement through fine-tuning. However, given that tasks 2 and 3 are more complex, we deemed it necessary to fine-tune the models to increase their performance.

4.2 The Models

Summarization is a common task and thus suitable for the majority of existing models. The following criteria were used to narrow down the candidates:

- *Model size*: Given the hardware, we are limited to models with 8 billion parameters or less.
- *Publisher*: We targeted models published by tech/AI companies. They have experience in LLM creation, and their releases are popular and widely used.
- *Release date*: Recent models tend to provide better results than older ones. The experiment was run from June to August 2024, and we focused on models released a few weeks/months before. Note, some of these models are already considered "legacy" such as Mistral v0.3.
- *Version*: The Instruct fine-tuned version of each model was chosen due to the use case studied in this article, and the possibility offered to the investigator to continue "discussing" with the model after completing the task.
- *Availability*: The chosen models must be open access and be released with information on how the model is structured and trained (base model as well as Instruction fine-tuned model). Although information about the used datasets is provided, the datasets themselves are usually not published.

Based on these criteria, we decided to use the three following models: Llama3.1-8B-Instruct [4], Mistral-7B-Instruct-v0.3[3] [14], and gemma-2-2b-it[4] [7]. Note that while a single model is sufficient, we tested three to (1) provide generalization over the impact of the tested variables/parameters, and (2) compare different models. Mistral can be considered standard, while Gemma has only a small number of parameters, which is useful when the available computing power is limited. Llama has an extended context-window size, which allows the user to provide a larger amount of messages. This might be better aligned with reality, where the number of messages retrieved from a device may be large.

4.3 Dataset and Task Descriptions

When analyzing available chat log summarization datasets, none was perfectly suited for the three tasks. Many datasets contained dialogues with associated summaries, but the summaries were too short and the discussions did not align with typical chats or lacked crime-related content. Having a single sentence summary is appropriate to understand the general topic of the conversation (useful for Task 2), but it is not sufficient to get a deep understanding of the discussion. For fine-tuning a model in a forensic context, it is crucial to provide samples similar to those encountered during investigations, i.e., conversations combining casual chat and crime-related topics. Given these limitations, we decided to create a dataset using GPT4 and the existing SAMSum dataset [8]. It aligns with the case study but lacks crime-related chats and has brief summaries. We addressed this by expanding the dataset in a partially reproducible way (see footnote 5). The new dataset contains GPT4-generated chat logs and three different summaries: (1) a manually generated general one-sentence summary describing the overall topic, (2) a manually generated detailed summary, and (3) a GPT4-generated detailed summary.

An additional one-sentence GPT4-generated summary was added during the transformation of the dataset. Having these manually and automatically generated summaries allowed us to derive two versions of each sample.

[3]The Mistral paper relates to version 0.1, but information about modifications operated to create newer versions can be found on the 0.2 and 0.3 models' pages.
[4]All these models can be found on huggingface.co.

During the generation of the manual summaries, all the GPT4-generated conversations were read. Although they did not perfectly reflect reality, we still considered that the quality of these messages was sufficient, e.g., messages did not include typos or abbreviations, and the underlying scenario behind each topic discussed was sometimes close to what is presented in movies or TV shows.

The training/validation dataset contains 172 samples, separated into three distinct sub-datasets that we named S1, S2, and S3. Each set contains approximately 60 samples divided between training and validation in a 4:1 ratio. Some samples were removed before fine-tuning (when the generated chat did not align with the requested topic). Therefore, S1 contains 46 training samples and 12 validation samples; S2 contains 42 validation samples and 12 validation samples; S3 contains 48 training samples and 12 validation samples. These three sets were then independently transformed into their fine-tuning versions and combined when given to the model during fine-tuning to reach 58 (S1), 112 (S1 + S2), and 172 (S1 + S2 + S3) samples. Note that we also tried to keep a balance between the criminal and non-criminal topics discussed in the chat, as well as a balance between the number of participants involved in the discussion (two participants vs. three+ participants).

4.3.1 Fine-Tuning Datasets. The datasets were generated for each task based on our newly created training/validation datasets and the SAMSum dataset. The SAMSum dataset was used to artificially increase the number of topics and messages in a conversation for Tasks 2 and 3 by providing chit-chat topics. We decided to combine one topic of interest (criminal-related or not) with a random number of irrelevant topics. The goal is to simulate a normal chat in which different topics are discussed, including one that is of interest to the investigation. This means that for these tasks, one sample from our new dataset (the topic of interest) was combined with three to six samples from the SAMSum dataset (chit-chat topics). The same process was applied for both manually and automatically generated summaries.

The test dataset used during the evaluation contains 36 samples and was created similarly. The main difference is that the chit-chat topics were also generated through GPT4 to avoid reusing the SAMSum dataset and to prevent any data leakage between the training/validation datasets and the testing one. Once again, each sample of the test dataset is present in both automatically and manually generated versions. The created datasets can be accessed in: https://huggingface.co/GaetanMichelet/datasets.[5] Here is a description of the samples' input and output for each task:

Dataset for Task 1. For the first task, the input consists of a list of messages (with the sender), and the output is a detailed summary of the conversation. The input includes a prompt that specifies the format of the messages and requests a summary.

Example input:

```
Alice: Hi Bob, did you complete the report?
Bob: Not yet, I'll finish it by tonight.
Alice: Please make sure it's done before the meeting tomorrow.
Please provide a detailed summary of the conversation.
```

Example output:

```
Alice asked Bob about the completion of the report. Bob replied that he would finish it
by that night.
Alice reminded him to have it done before the meeting the next day.
```

Dataset for Task 2. Here, the input consists of a set of messages from a conversation that discusses several topics. The messages include timestamps and sender information. The task involves two steps: (1) provide short summaries of each discussed topic, mentioning the timestamps of the first and last messages related to each topic, and (2) summarize a specific topic of interest in more detail, given the topic itself.

[5] The SAMSum dataset license does not allow us to release the fine-tuning dataset directly. Therefore, we share the scripts that enable researchers to build alternative, similar versions to ours (everything should be the same except for the GPT4-generated short summaries).

Example input for step 1:
```
[10:00] Alice: Hey, are we still on for lunch?
[10:05] Bob: Yes, see you at 12.
[11:00] Charlie: Don't forget the meeting at 3 PM.
[11:15] Alice: Thanks for the reminder.
Please provide short summaries of each topic discussed, including the timestamps of the
relevant messages.
```

Example output for step 1:
```
[10:00–10:05] Alice and Bob confirm lunch plans.
[11:00–11:15] Charlie reminds Alice about the 3 PM meeting, and Alice acknowledges.
```

Example input for step 2:
```
Please provide a detailed summary of the topic starting at 11:00 and ending at 11:15.
```

Example output for step 2:
```
Charlie informed Alice about a meeting scheduled at 3 PM. Alice thanked Charlie for the
reminder.
```

Dataset for Task 3. The third task is similar to Task 2 but focuses on a specific crime-related topic within a conversation that includes multiple topics (zero or one criminal and several non-criminal). The input also comprises the name of the crime being investigated. When the sample chat did not contain any criminal topic, the crime investigated was randomly picked from the list of all crimes considered in this experiment. Note that for the test dataset, a random seed was reset between the sample versions using the automatically and manually generated summaries. This ensures that the automatically and manually generated versions of each test sample have the same investigated crime, which is important for the evaluation process.

Example input:
```
Crime Investigated: Unauthorized Access
[09:00] Dave: Did you get into the system?
[09:05] Eve: Yes, I bypassed the firewall.
[09:10] Dave: Excellent. Download the files and delete the logs.
[09:15] Eve: Will do.
[10:20] Dave: By the way, are you coming to the office party tonight?
[10:25] Eve: Yes, looking forward to it!
[10:30] Dave: Great, see you there.
Please provide a detailed summary related to the crime of Unauthorized Access.
```

Example output:
```
The topic of interest for the investigation started at 09:00 and ended at 09:15. Eve
informed Dave that she successfully bypassed the firewall to access the system. Dave
instructed her to download the files and delete the logs, indicating activities related
to unauthorized access.
```

Remark. The number of samples in the created and remixed datasets is limited. Therefore, we focus on quality over quantity. We expected that a small number of high-quality samples would already yield beneficial results. We also expected that having more samples would yield even more beneficial results, but we could not test it. This part of the process was perhaps the most time-consuming part of the case study, but was essential given the results obtained.

4.4 The Fine-Tuning Process

The models are fine-tuned on a standard desktop with a Nvidia RTX 4500 Ada GPU (24 GB of memory). Given the hardware available and the recipes provided by the framework, we decided to use SFT with QLoRA, and the cross-entropy training loss function. QLoRA adapters are created for each fine-tuned model.

The basic setup of the alignment handbook scripts provides a check-pointing system (which was useful given that sometimes errors occurred and stopped the process) and computes the loss function on the prompt and on the answer from the model. By providing the chat template, it was possible to mask the tokens in the prompt during the loss computation. We kept many of the hyperparameter settings as they were provided in the recipes of the alignment handbook, including the LoRA and quantization parameters.

At first, four different configurations were tested, with the "learning rate" and "gradient accumulation steps" varying. The gradient accumulation helps to increase the effective batch size, as the GPU memory was limiting the training batch size to one. The optimal number of epochs was automatically detected using a callback function that stops the training when the validation loss is bigger than the current best loss seven times in a row. The four configurations tested were the four possible combinations with a "learning rate" of 0.0001 or 0.00001 and a "gradient accumulation steps" of 8 or 16. Note that the maximum "number of epochs" was 50 for the 0.0001 "learning rate," and 150 for the 0.00001 "learning rate" (a smaller "learning rate" will increase the "number of epochs" required to reach the optimal validation loss). After a few tests, we noticed that the lowest learning rate (0.00001) was increasing the training time but not improving the optimal validation loss. We decided to focus on the higher learning rate (0.0001) with varying gradient accumulation steps (8 or 16). Each of the fine-tuned models can be found at: https://huggingface.co/GaetanMichelet/models.

4.5 The Evaluation

For our study, we chose three metrics that allow for the automatic comparison of two texts: ROUGE-1/2/L, BLEU-1/2, and BERTScore-F1/RoBERTaScore-F1 (no manual evaluation was performed). These metrics measure different elements and provide insight into the inference quality generated by the model. As these evaluation metrics are specific to our use case (here summarization), we decided to include their description in this section and not in Section 3.5. Other use cases would certainly require different evaluation metrics.

ROUGE [17] and BLEU [22] measure the similarity of texts by computing the co-occurrence of n-grams between the prediction and the reference, with n determined by the user. The number of co-occurring n-grams is then divided by the number of possible n-grams in the candidate (precision/BLEU) and by the number of possible n-grams in the reference (recall/ROUGE). Note that ROUGE provides a ROUGE-L score, which uses the longest common subsequence instead of the co-occurring n-grams. BLEU introduces a brevity penalty in the score computation. When the candidate is much shorter than the reference, the number of potential n-grams will decrease, and a better score will be easier to achieve. The brevity penalty impacts the score when the candidate is shorter than the reference. The list of scores generally used is: ROUGE-1, ROUGE-2, ROUGE-L, BLEU-1, and BLEU-2.

The BERTScore [38] helps measure the semantic similarity between prediction and reference and complements ROUGE and BLEU. This score is computed by embedding each token in the reference and the candidate, before creating couples of tokens (one from the reference, one from the prediction) based on their cosine similarity. Tokens with maximum cosine similarity are grouped, and the precision, recall, and F1-score are computed based on this maximum cosine similarity. Note that it can be applied using BERT and other models such as RoBERTa.

Remark. During the inference process of our base models and fine-tuned models, we deactivated sampling, which is equivalent to setting the temperature parameter to zero ($T = 0$), to disable randomness and better align with practical applications. In standard LLM inference, sampling is typically used to introduce variability by randomly selecting one of the m most probable next tokens. This enhances creativity, but also means that a model may generate different responses for the same prompt. In DF, this variability raises concerns about reproducibility, which is a critical factor in legal proceedings.

5 Results

The experiment involves three base models and three tasks, resulting in nine model-task combinations. Each combination is fine-tuned with three datasets with varying numbers of samples (58, 112, and 172 samples), and available in two versions: with manually or automatically generated summaries. This results in nine model-tasks with three different datasets in two versions, resulting in $9 \times 3 \times 2 = 54$ model-task-dataset-version combinations. Then two configurations are used in each experiment (effective training batch sizes of 8 and 16). Finally, two different cross-entropy loss computations are applied for each configuration ("answer only" and "prompt + answer"), resulting in $54 \times 2 \times 2 = 216$ fine-tuned models to evaluate.

Subsequently, tests are carried out based on the testing dataset to evaluate the quality of the base models and the fine-tuned models for each task. The *metrics* used to measure the quality of the models are all automatically calculated. In Task 2, two inferences are generated, and their results are averaged to produce a single outcome for each metric. Eventually, results from the different fine-tuned and base models are compared to answer the research questions.

Given the number of tests, this section only summarizes the main observations. The detailed results can be downloaded from: https://github.com/Michelet-Gaetan/Fine-tuning-LLMs-for-DF-tasks.

5.1 Impact of the Loss Computation Method

An aspect of fine-tuning language models is how the loss function is computed during training. In the experiments, we compared two methods: "answer only" and "prompt + answer" loss computations. Although the "prompt + answer" loss computation may seem redundant since the prompt is known, it helps the model learn prompt-answer connections and contextual relationships. This improves coherence, especially in multi-turn dialogues or when handling incomplete or noisy prompts. We observed that all fine-tuned models, regardless of the loss computation method used during training, provided better results than the base models. However, the improvement seen was greater with models fine-tuned using the "answer only" loss computation. The findings suggest that while the "prompt + answer" loss computation can have theoretical advantages in certain contexts, the "answer only" loss computation is more efficient and yields better performance for tasks where the focus is on generating accurate answers. For the remainder of this section, models fine-tuned using the "prompt + answer" loss computation will be filtered out of the results.

5.2 Quality of the Base and Fine-Tuned Model

The fine-tuned model outperforms the base model in all metrics, except for a longer runtime. The output of every type of fine-tuned model (manually and automatically generated training samples) was closer to each type of expected output (manually and automatically generated references) than the texts resulting from base models. The results achieved by the base model and all fine-tuned models compared to the expected results generated manually are shown in Figure 1.

The values presented are the averaged results for a specific model (base) or group of models (fine-tuned versions of the base model) for each metric considered during the evaluation. For each metric, the closer to 1, the better the model(s) performed. There is an exception for runtime, where a lower value (closer to 0) indicates better performance. We did not compare the results obtained with the literature presented in Section 2, because previous studies used different models, fine-tuning methods, or datasets (for training and evaluation). Consequently, this study focuses on the differences between the base- and fine-tuned models used herein.

5.3 Impact of the Number of Samples

Interestingly, a small number of 58 training samples, divided into training and validation sets in a 4:1 ratio, was sufficient to improve the quality of the generated text. As the number of samples increased, the quality of inferences improved in general. However, a slight decrease in quality was observed when models fine-tuned with

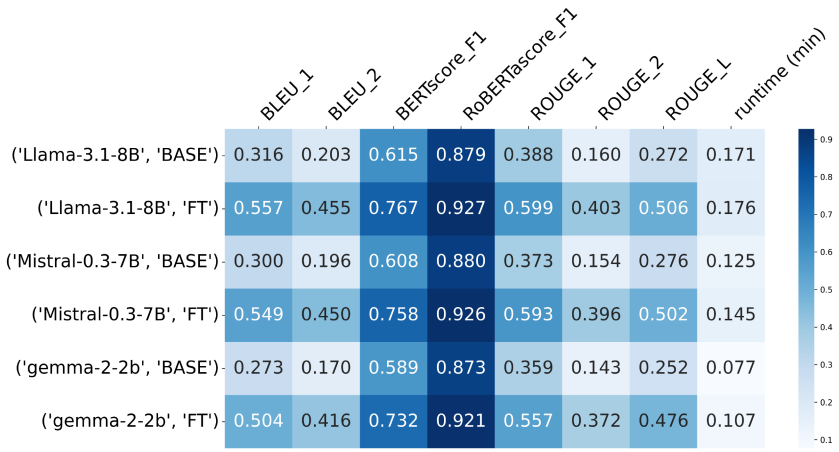

Fig. 1. Metrics computed against manually generated expected output for every fine-tuned and base models.

automatically generated samples were compared against manually generated expected outputs as the sample size increased.

5.4 Impact of Manually vs. Automatically Generated Summaries

We observed that, in general, fine-tuned models performed better when evaluated against summaries of the same type as those used during training, i.e., models fine-tuned on manual summaries produced outputs that more closely matched the manually generated reference summaries (and vice versa).

This indicates that the alignment between the type of training data and the evaluation data plays a significant role in the quality performance of the model. The models appear to learn and replicate the stylistic and content patterns specific to the type of summaries on which they were trained.

In practice, this means that the training dataset seems to guide the model toward a particular writing style and set of elements considered relevant to accomplish the task at hand. This can be beneficial, especially if the fine-tuning process is achieved or at least supervised by an investigator who knows the writing style of the institution and the pieces of information required for the task at hand. This will ensure that the model adopts the targeted writing style and selects the elements relevant to the task at hand.

Note that there is an exception for task 3, where the models fine-tuned on the manual summaries produced outputs that matched more closely the manually and automatically generated reference summaries. This difference might be explained by the creation process of the fine-tuning dataset for task 3. Variability between the samples generated from automatic and manual summaries was introduced when the chat did not contain any criminal topics. In this scenario, the crime investigated was randomly picked from the list of all crimes considered in the experiment. This means that for a given sample with no crime-related chat, the crime investigated in the manual summary version might be different from the crime investigated in the automatic summary version. This variation might have led to better training for the models fine-tuned on manual summaries for task 3.

5.5 Impact of the Configuration

The batch size is a hyperparameter that influences the learning process by determining the number of samples the model processes before updating its parameters. Smaller batch sizes lead to more frequent weight updates with higher variance gradients, which can help the model capture diverse patterns and nuances in the data. In contrast, larger batch sizes result in more stable and smoother gradient updates, which may benefit learning from data with consistent patterns.

In our experiments, the differences between the two configurations (different training batch sizes) were not particularly significant overall. However, we observed that *Configuration 1* (smaller batch size) tended to perform better when fine-tuning with manually generated training samples. Conversely, *Configuration 2* (larger batch size) showed slightly better performance when fine-tuning with automatically generated training samples.

A hypothesis that could be made based on these observations is the following: a smaller batch size might be more effective when working with manually generated data, possibly because of the greater variability and richness of human-created content. The higher variance in gradients with smaller batches allows the model to adapt to the diverse examples more effectively. A larger batch size might be more suitable for automatically generated data, where consistency and homogeneity are higher, providing stable gradient estimates that reinforce the prevalent patterns in the data.

5.6 Discussion of the Results

The fine-tuned models outperformed the base models across all metrics (except runtime), a consistent result for each tested model: Gemma-2, Llama-3.1, and Mistral-v0.3. Although the sample size is too small to generalize, it is promising to obtain measurable improvements. Specifically, the results showed that 58 samples (12 for validation and 46 for S1 training) had an impact, which, given the lack of datasets in DF, is encouraging. Moreover, models trained on both manually and automatically generated samples outperformed the base models, regardless of the expected output type. This could help address the dataset availability issue through automated dataset generation. More tests are needed to see if there will be an impact on real-world chat conversations.

Observations support the recommendation of computing the loss function on "answer only." The runtime increases when the loss is computed on "prompt + answer." After a manual investigation, many inferences made by that kind of fine-tuned model presented repetitions of the last sentence(s) until the maximum number of new tokens was reached. This long sentence could partially explain the diminution of the score on each metric.

Lastly, larger batch sizes appeared to be more effective for automatically generated samples. Although a hypothesis was proposed, the exact reason for this remains unclear. It is important to note that this observation is specific to the small-scale experiment and may not hold in a different context.

5.7 Limitations

We identified the following limitations:

— *The tasks*: The tasks tested in our case study focused on summarization (an already well-suited LLM capability), excluding other types of task where the outcome may be different and limiting the generalizability of the results.
— *The models*: The case study explored models from 2B to 8B parameters, excluding smaller and larger models. Probably, larger models perform better and smaller ones worse. Having access to larger models (by removing the hardware limitation) could lead to a different impact of the fine-tuning process. The difference between the base and fine-tuned versions could be reduced due to a better base quality or increased due to a higher (re)training potential. Further testing is needed to determine the impact of larger models on the fine-tuning process.
— *The datasets*: The small size of the datasets and their composition, made from synthetic data and from small chats never exceeding the maximum number of tokens supported by the models, limit the generalizability of the results and their applicability to real-world investigations. Chat logs may contain slang, intertwined topics, and have multiple languages, challenges that were not addressed in this study. Finally, the time and computational costs associated with dataset generation may pose a barrier in smaller laboratories.
— *The fine-tuning*: We analyzed a limited subset of hyperparameters, leaving the impact of others unexplored.
— *The evaluation*: The metrics used cannot capture bias, such as parts of the chat that are ignored or emphasized by the model in the generated summaries, or misinterpretations of the context/sentiment of the conversation,

which could be solved by implementing a manual evaluation. In addition, the performance of the current analysis and summary writing methods was not considered. This makes a formal comparison between the performances of the fine-tuned models and the currently used methods impossible.

Despite limitations, our work demonstrates the feasibility of fine-tuning models. It also offers insights into the performance increase obtained and the influence of key parameters, including dataset size, model size, and loss computation.

6 Lessons Learned and Insights from Fine-Tuning LLMs

Preliminary testing is important, although we did not conduct it extensively across all tasks, as we were convinced that fine-tuning would ultimately lead to better outcomes. This mistake could have led to a waste of time/resources if tasks 2 and 3 had been performed differently. In the future, we plan to adopt a more rigorous testing methodology before fine-tuning LLMs.

Another lesson learned is that finding appropriate datasets for a given task is challenging and the creation or adaptation of existing datasets is time-consuming. The preparation of the dataset accounted for approximately a third of the time spent on the experiment. On a positive note, as highlighted in Section 5, even a small number of high-quality samples can enhance the performance of a model. In addition, sharing a (modified) dataset may be prohibited due to licenses.

Many decisions were impacted by the available computing resources, in our case, a single 24 GB GPU (costing around $ 2,000). The complexity of acquiring computational power is enhanced by the impossibility of renting online resources when working on real-case datasets.

The fine-tuning process can be intimidating with no or minimal prior knowledge. Although existing frameworks such as the "alignment handbook"[6] provide a good starting point by sharing existing scripts and configuration files, it is a steep learning curve. It was difficult to find and understand appropriate metrics. Manual evaluation is difficult to set up in a systematic and sound way. It is time-consuming and requires opinions from different evaluators.

The last lesson learned is that fine-tuning for DF is currently not (sufficiently) beneficial. Although the presented fine-tuning process is applicable and yields better performances, this improvement is strongly outweighed by (1) the rapid evolution of base LLMs that quickly outperform a fine-tuned version of an older model, and (2) the associated time and computational costs. Although we encourage research exploring fine-tuning for DF, we do not recommend implementing it in practice yet.

7 Conclusion

To explore the underdeveloped area of fine-tuning LLMs for DF, we conducted a comprehensive literature review, examined key fine-tuning elements, and performed experiments as part of a case study. This article has made several contributions to the field of fine-tuning LLMs for DF. First, we provided practitioner-focused recommendations for model fine-tuning. Second, we ran a case study to fine-tune LLMs specifically for chat summarization in forensic contexts. Third, we shared valuable insights and challenges encountered during this process. Lastly, we made chat-summarization datasets and fine-tuned models available to the community. These efforts allowed us to consolidate our findings and answer the research questions posed in the introduction:

How Can Small Local LLMs Be Fine-Tuned to Address the Unique Challenges and Requirements of DF? This work showed evidence that, despite the unique challenges of DF, it is feasible to fine-tune models. We developed a set of recommendations that offer a foundation for those looking to fine-tune models in DF. The five key components identified are: the task, the base model, the dataset, the fine-tuning process, and the evaluation metrics. Our results indicate that even small datasets can lead to measurable improvements in model performance.

[6] https://github.com/huggingface/alignment-handbook.

How Can We Evaluate the Results to Ensure That the Fine-Tuning Process Was Successful and Identify the Best-Performing Model? An assessment is needed to determine whether the fine-tuning has been successful. In our case study, we demonstrated how to train various models and compare their performance for a specific task.

How Beneficial Is the Fine-Tuning Process for Improving Model Performance, and Should Practitioners Adopt It in Forensic Workflows? While improvements are observed, they are not sufficient given the costs, limitations, and the likelihood of better base models emerging soon. We do not yet recommend a practical implementation.

Future research may build on this work by exploring the case study in greater depth, particularly by: (1) testing and evaluating the fine-tuning process on real-case data, (2) exploring its usability for different forensic tasks, (3) considering a wider range of base models, (4) increasing the dataset size and the sample variability, (5) expanding the number of hyperparameters tested throughout the fine-tuning process, (6) incorporating manual evaluation, and (7) evaluating the currently used methods to compare their performances against the base and fine-tuned models.

It remains also important to consider the ethical aspects of the use of LLMs in the context of DF. Evitable errors in court must be avoided, and therefore any result obtained, or text generated through the use of an LLM, must be validated.[7] We also recommend being transparent towards the use of LLMs and specifying in the report if an LLM was used and, if so, for which purpose.

Acknowledgments

First and foremost, we would like to express our deepest gratitude to the University of Lausanne, and, particularly, to the Mobi.Doc committee. We also thank Edwin Rijgersberg for helping with the use of the alignment handbook. Finally, we thank all the reviewers. The relevant and helpful feedback they provided allowed us to improve this work.

Declaration of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article.

References

[1] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*. A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36, Curran Associates, Inc., 10088–10115. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2023/file/1feb87871436031bdc0f2beaa62a049b-Paper-Conference.pdf

[2] Ricardo J. Dinis-Oliveira and Rui M. S. Azevedo. 2023. ChatGPT in forensic sciences: A new Pandora's box with advantages and challenges to pay attention. *Forensic Sciences Research* 8, 4 (Nov. 2023), 275–279. DOI: https://doi.org/10.1093/fsr/owad039

[3] Evangelos Dragonas, Costas Lambrinoudakis, and Panagiotis Nakoutis. 2024. Forensic analysis of OpenAI's ChatGPT mobile application. *Forensic Science International: Digital Investigation* 50 (2024), 301801. DOI: https://doi.org/10.1016/j.fsidi.2024.301801

[4] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The Llama 3 Herd of models. arXiv:2407.21783. Retrieved from https://arxiv.org/abs/2407.21783

[5] Guy Feigenblat, Chulaka Gunasekara, Benjamin Sznajder, Sachindra Joshi, David Konopnicki, and Ranit Aharonov. 2021. TWEETSUMM—A dialog summarization dataset for customer service. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.), Association for Computational Linguistics, 245–260. DOI: https://doi.org/10.18653/v1/2021.findings-emnlp.24

[6] Xiachong Feng, Xiaocheng Feng, and Bing Qin. 2022. A survey on dialogue summarization: Recent advances and new frontiers. arXiv:2107.03175. Retrieved from https://arxiv.org/abs/2107.03175

[7] Gemma Team: Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, et al. 2024. Gemma 2: Improving open language models at a practical size. arXiv:2408.00118. Retrieved from https://arxiv.org/abs/2408.00118

[7] Note that even though validation is required, efficiency is still improved through the identification of potentially interesting chats and the drafting of their summary.

[8] Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum Corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*. Lu Wang, Jackie Chi Kit Cheung, Giuseppe Carenini, and Fei Liu (Eds.), Association for Computational Linguistics, 70–79. DOI: https://doi.org/10.18653/v1/D19-5409

[9] Christopher Hargreaves, Frank Breitinger, Liz Dowthwaite, Helena Webb, and Mark Scanlon. 2024. DFPulse: The 2024 digital forensic practitioner survey. *Forensic Science International: Digital Investigation* 51 (2024), 301844. DOI: https://doi.org/10.1016/j.fsidi.2024.301844

[10] Hans Henseler and Harm van Beek. 2023. ChatGPT as a Copilot for investigating digital evidence. In *Proceedings of the Third International Workshop on Artificial Intelligence and Intelligent Assistance for Legal Professionals in the Digital Workplace (LegalAIIA '23)*, Vol. 3423. CEUR Workshop Proceedings, 58–69. Retrieved from https://ceur-ws.org/Vol-3423/paper6.pdf

[11] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-rank adaptation of large language models. arXiv:2106.09685. Retrieved from https://arxiv.org/abs/2106.09685

[12] Kai Huang, Xiangxin Meng, Jian Zhang, Yang Liu, Wenjie Wang, Shuhao Li, and Yuqing Zhang. 2023. An empirical study on fine-tuning large language models of code for automated program repair. In *Proceedings of the 38th IEEE/ACM International Conference on Automated Software Engineering (ASE '23)*. IEEE Press, 1162–1174. DOI: https://doi.org/10.1109/ASE56229.2023.00181

[13] Yongbin Jeong, Ju-Hyuck Han, Kyung Min Chae, Yousang Cho, Hyunbin Seo, KyungTae Lim, Key-Sun Choi, and Younggyun Hahm. 2023. Teddysum at MEDIQA-Chat 2023: An analysis of fine-tuning strategy for long dialog summarization. In *Proceedings of the 5th Clinical Natural Language Processing Workshop*. Tristan Naumann, Asma Ben Abacha, Steven Bethard, Kirk Roberts, and Anna Rumshisky (Eds.), Association for Computational Linguistics, 394–402. DOI: https://doi.org/10.18653/v1/2023.clinicalnlp-1.42

[14] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7B. arXiv:2310.06825. Retrieved from https://arxiv.org/abs/2310.06825

[15] Sreya, E. K. Sakshi, and Manisha Wadhwa. 2023. Enhancing digital investigation: Leveraging ChatGPT for evidence identification and analysis in digital forensics. In *Proceedings of the 2023 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, 733–738. DOI: https://doi.org/10.1109/ICCCIS60361.2023.10425000

[16] Misha Khalman, Yao Zhao, and Mohammad Saleh. 2021. ForumSum: A multi-speaker conversation summarization dataset. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.), Association for Computational Linguistics, 4592–4599. DOI: https://doi.org/10.18653/v1/2021.findings-emnlp.391

[17] Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, 74–81. Retrieved from https://aclanthology.org/W04-1013

[18] Raj J. Mathav, V. M. Kushala, Harikrishna Warrier, and Yogesh Gupta. 2024. Fine tuning LLM for enterprise: Practical guidelines and recommendations. arXiv:2404.10779. Retrieved from https://arxiv.org/abs/2404.10779

[19] Gaëtan Michelet and Frank Breitinger. 2024. ChatGPT, llama, can you write my report? An experiment on assisted digital forensics reports written using (local) large language models. *Forensic Science International: Digital Investigation* 48 (2024), 301683. DOI: https://doi.org/10.1016/j.fsidi.2023.301683

[20] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. arXiv:2402.06196. Retrieved from https://arxiv.org/abs/2402.06196

[21] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2025. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology* (June 2025). Just accepted. DOI: https://doi.org/10.1145/3744746

[22] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL '02)*. Association for Computational Linguistics, 311–318. DOI: https://doi.org/10.3115/1073083.1073135

[23] Ruiyang Qin, Dancheng Liu, Chenhui Xu, Zheyu Yan, Zhaoxuan Tan, Zhenge Jia, Amir Nassereldine, Jiajie Li, Meng Jiang, Ahmed Abbasi, et al. 2025. Empirical guidelines for deploying LLMs onto resource-constrained edge devices. *ACM Transactions on Design Automation of Electronic Systems* (May 2025). Just accepted. DOI: https://doi.org/10.1145/3736721

[24] Mark Scanlon, Frank Breitinger, Christopher Hargreaves, Jan-Niclas Hilgert, and John Sheppard. 2023. ChatGPT for digital forensic investigation: The good, the bad, and the unknown. *Forensic Science International: Digital Investigation* 46 (2023), 301609. DOI: https://doi.org/10.1016/j.fsidi.2023.301609

[25] Mark Scanlon, Bruce Nikkel, and Zeno Geradts. 2023. Digital forensic investigation in the age of ChatGPT. *Forensic Science International: Digital Investigation* 44 (Mar. 2023), 301543. DOI: https://doi.org/10.1016/j.fsidi.2023.301543

[26] Aleksei Shestov, Rodion Levichev, Ravil Mussabayev, Evgeny Maslov, Pavel Zadorozhny, Anton Cheshkov, Rustam Mussabayev, Alymzhan Toleu, Gulmira Tolegen, and Alexander Krassovitskiy. 2025. Finetuning large language models for vulnerability detection. *IEEE Access* 13 (2025), 38889–38900. DOI: https://doi.org/10.1109/ACCESS.2025.3546700

[27] Kunal Suri, Prakhar Mishra, Saumajit Saha, and Atul Singh. 2023. SuryaKiran at MEDIQA-Sum 2023: Leveraging LoRA for clinical dialogue summarization. arXiv:2307.05162. Retrieved from https://arxiv.org/abs/2307.05162

[28] Xiangru Tang, Arjun Nair, Borui Wang, Bingyao Wang, Jai Desai, Aaron Wade, Haoran Li, Asli Celikyilmaz, Yashar Mehdad, and Dragomir Radev. 2022. CONFIT: Toward faithful dialogue summarization with linguistically-informed contrastive fine-tuning. In

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (Eds.), Association for Computational Linguistics, 5657–5668. DOI: https://doi.org/10.18653/v1/2022.naacl-main.415

[29] Yuanhe Tian, Fei Xia, and Yan Song. 2024. Dialogue summarization with mixture of experts based on large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.), Association for Computational Linguistics, 7143–7155. DOI: https://doi.org/10.18653/v1/2024.acl-long.385

[30] R. Obert Tinn, Hao Cheng, Yu Gu, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2023. Fine-tuning large neural language models for biomedical natural language processing. *Patterns* 4, 4 (2023), 100729. DOI: https://doi.org/10.1016/j.patter.2023.100729

[31] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Shengyi Huang, Kashif Rasul, Alvaro Bartolome, Alexander M. Rush, and Thomas Wolf. 2023. The Alignment Handbook. Version 0.3.0.dev0, Apache-2.0 License. Retrieved from https://github.com/huggingface/alignment-handbook

[32] Clinton Walker, Taha Gharaibeh, Ruba Alsmadi, Cory Hall, and Ibrahim Baggili. 2024. Forensic analysis of artifacts from Microsoft's multi-agent LLM platform AutoGen. In *Proceedings of the 19th International Conference on Availability, Reliability and Security (ARES '24)*. ACM, New York, NY, Article 198, 9 pages. DOI: https://doi.org/10.1145/3664476.3670908

[33] Akila Wickramasekara, Frank Breitinger, and Mark Scanlon. 2025. Exploring the potential of large language models for improving digital forensic investigation efficiency. *Forensic Science International: Digital Investigation* 52 (2025), 301859. DOI: https://doi.org/10.1016/j.fsidi.2024.301859

[34] Jianfei Xiao, Yancan Chen, Yimin Ou, Hanyi Yu, Kai Shu, and Yiyong Xiao. 2024. Baichuan2-Sum: Instruction finetune Baichuan2-7B model for dialogue summarization. *In Proceedings of the 2024 International Joint Conference on Neural Networks (IJCNN)*, 1–8. DOI: https://doi.org/10.1109/IJCNN60899.2024.10650513

[35] Shengbin Yue, Wei Chen, Siyuan Wang, Bingxuan Li, Chenchen Shen, Shujun Liu, Yuxuan Zhou, Yao Xiao, Song Yun, Xuanjing Huang, and Zhongyu Wei. 2023. DISC-LawLLM: Fine-tuning large language models for intelligent legal services. arXiv:2309.11325. Retrieved from https://arxiv.org/abs/2309.11325

[36] Jiseon Yun, Jae Eui Sohn, and Sunghyon Kyeong. 2023. Fine-tuning pretrained language models to enhance dialogue summarization in customer service centers. In *Proceedings of the Fourth ACM International Conference on AI in Finance (ICAIF '23)*. ACM, New York, NY, 365–373. DOI: https://doi.org/10.1145/3604237.3626838

[37] Longxiang Zhang, Renato Negrinho, Arindam Ghosh, Vasudevan Jagannathan, Hamid Reza Hassanzadeh Thomas Schaaf, and Matthew R. Gormley. 2021. Leveraging pretrained models for automatic summarization of doctor-patient conversations. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.), Association for Computational Linguistics, 3693–3712. DOI: https://doi.org/10.18653/v1/2021.findings-emnlp.313

[38] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. arXiv:1904.09675. Retrieved from https://arxiv.org/abs/1904.09675

[39] Yusen Zhang, Ansong Ni, Tao Yu, Rui Zhang, Chenguang Zhu, Budhaditya Deb, Asli Celikyilmaz, Ahmed Hassan Awadallah, and Dragomir Radev. 2021. An exploratory study on long dialogue summarization: What works and what's next. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.), Association for Computational Linguistics, 4426–4433. DOI: https://doi.org/10.18653/v1/2021.findings-emnlp.377

[40] Lulu Zhao, Fujia Zheng, Weihao Zeng, Keqing He, Weiran Xu, Huixing Jiang, Wei Wu, and Yanan Wu. 2022. Domain-oriented prefix-tuning: towards efficient and generalizable fine-tuning for zero-shot dialogue summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (Eds.), Association for Computational Linguistics, 4848–4862. DOI: https://doi.org/10.18653/v1/2022.naacl-main.357

[41] Ming Zhong, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2022. DialogLM: Pre-trained model for long dialogue understanding and summarization. *Proceedings of the AAAI Conference on Artificial Intelligence* 36, 10 (June 2022), 11765–11773. DOI: https://doi.org/10.1609/aaai.v36i10.21432

Received 9 May 2025; revised 9 May 2025; accepted 7 July 2025