Structured Report Generation using Local LLMs for Chat-Based Digital Forensics

Ying Dehing
University of Amsterdam
Amsterdam, The Netherlands

Hans Henseler
University of Applied Sciences Leiden
Leiden, The Netherlands
Netherlands Forensic Institute
The Hague, The Netherlands

Timo Meconi
University of Groningen
Groningen, The Netherlands
Netherlands Forensic Institute
The Hague, The Netherlands

Marcel Worring
University of Amsterdam
Amsterdam, The Netherlands

Harm van Beek
Open University The Netherlands
Heerlen, The Netherlands
Netherlands Forensic Institute
The Hague, The Netherlands

ABSTRACT

Chat conversations are an essential data source in forensic investigations. With the number of chat messages sent and received on a daily basis, manually analysing all these chats is very time-consuming if not impossible. Growing interest exists in leveraging Large Language Models (LLMs) to structure and interpret chat conversations to support investigators with finding relevant messages. Privacy and transparency requirements force LLMs to run locally, on hardware under control of investigative authorities. Our goal in this paper is to evaluate the feasibility of using local LLMs to create structured investigative reports from chat corpora. We adopt a two-stage prompting framework. In stage 1, we split the long conversations in processable parts and, combined with case information, we use local LLMs to create case-specific summaries. In stage 2, these summaries are combined into a single report. We compare the outcome of five local models with (validated) ground-truth output of the cloud model Gemini 2.5 Pro. Results indicate that local LLMs can achieve useful performance in entity and role extraction, but still exhibit gaps in strict citations and chronological consistency. Our findings position local LLMs as triage tools: they can accelerate lead discovery and structure large chat corpora under strict privacy constraints. Their outputs must be treated as provisional and should be used for investigative purposes and explicit validation is required before inclusion in a case report.

KEYWORDS

Digital Forensics, Chat Conversations, LLMs, Prompt Engineering, Timeline Reconstruction, Evaluation

DFDS'26, March 2026, Linköping, Sweden

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The overwhelming amount of chat data stored on digital devices creates challenges for investigators to isolate and structure the most probative traces for a case [16]. LLMs offer promise for triage and structuring unstructured text; however, forensic settings impose constraints on privacy, auditability, and evidentiary citation.

LLMs, based on transformer architectures, can summarise long texts and surface patterns that are otherwise time-consuming to find manually [2, 8, 30, 38]. Their potential value in forensics lies in automating parts of chat analysis and providing structured summaries with explicit references to provenance record [38, 40]. LLMs can be broadly categorised into cloud-hosted models (accessed via an Application Programming Interface (API)) and local models (running on self-hosted infrastructure). While architecturally similar, they differ in performance, deployment, resource requirements, and privacy: cloud providers (e.g., OpenAI, Google) process data on external servers using large models with high performance, whereas local LLMs keep data on investigator-controlled hardware [26], which inherently are much smaller and have lower performance.

Integrating LLMs into digital forensics (DF) introduces risks, including hallucinations and position-bias effects in long contexts, for example, Liu et al. [24] report a U-shaped performance curve depending on where relevant information appears in extended sequences. Although frontier cloud models like Gemini 2.5 pro preview (Gemini 2.5 Pro) [11, 12] support very long contexts and strong reasoning, concerns persist regarding transparency and reproducibility for legally sensitive evidence.

Prior work notes that cloud-hosted LLMs are often unsuitable for sensitive casework due to data-handling policies and the lack of shareable, realistic datasets for benchmarking [5, 38]. This distinction between cloud-hosted and locally based LLMs is particularly relevant in legal contexts that require confidentiality and explicit data-handling guarantees [40]. Local models offer a privacy-preserving alternative but face tighter context limits and performance trade-offs. To enable a controlled comparison between cloud-hosted and locally based LLMs, we use the synthetic *Crystal Clear* (≈ 240k tokens) dataset, designed for DF training, which features stable trace identifiers and shareable licensing. Against this backdrop, our study is guided by the following research question: *To what extent can locally-deployed LLMs support realistic, chat-centric DF tasks compared to a capable*

cloud baseline, given constraints on context length, model capacity and deployment setting?

Motivation and Scope. We investigate whether *local, on-prem* LLMs can operate efficiently in digital forensic workflows, focusing on two key challenges inherent to their use [40]. The first is **hallucinations**, where models produce outputs that appear plausible but contain false or misleading information [41]. In a forensic setting, such inaccuracies can generate false leads or introduce inadmissible evidence [40], often arising from low-quality training data, outdated knowledge, or suboptimal prompt design [30]. The second challenge is the **limited context window**, which restricts how much text a model can process at once. This challenge can complicate the analysis of large datasets, as the dataset might exceed the model's context window. Models may struggle to retain and retrieve information buried in the middle of the input [15, 25]. While a larger context window may improve performance, it also introduces new issues, such as attention decay. To address these challenges, we evaluate local LLMs within a controlled, two-stage chat-centric DF pipeline. In —*Stage 1* the models perform per-part (<29k-token) extraction with strict Trace-ID citation. In *Stage 2*, it synthesises information across parts to produce comprehensive reports. We measure performance on core DF tasks in chat evidence, specifically identifying entities, roles, trace IDs, and timelines under identical prompt constraints, and assess the quality of investigative reports using a forensics-weighted evaluation. We compare five local models against a global baseline (Gemini 2.5 Pro) run in a *manual* two-stage online flow. This setup enables us to investigate how hallucinations and contextual limitations impact forensic reasoning and report quality.

Artefacts. To support replication and reuse, we provide (1) the *Crystal Clear Chats (V3)* benchmark corpus with stable Trace IDs and documented <29k-token splits; (2) a complete prompt set (P1–P5) and a run-time system that implements the two-stage pipeline; and (3) evaluation assets including a whole-report rubric, a timeline-only protocol, a Trace-ID equivalence map, and a validator script.[1]

2 RELATED WORK

Michelet and Breitinger [27] explored the potential of using LLMs such as ChatGPT and Llama to assist in writing forensic case reports. In their evaluation, ChatGPT outperformed Llama, producing more concise and accurate texts, especially for introductory sections. Nevertheless, thorough proofreading and adjustments are necessary to ensure a suitable output, highlighting the importance of human oversight.

Similarly, Scanlon et al. [31] explored ChatGPT's utility in artefact identification, incident response, and programming tasks. They found ChatGPT to be helpful in tasks where there is not only one right or wrong answer, i.e., scenario thinking and instances when one has no idea where to start, thus avoiding the blank page syndrome. However, a decline in performance was noted when tasks became more domain-specific or required contextual knowledge, which could be attributed to the relatively small amount of training data available in the forensic community.

Henseler and Van Beek [17] examined the role of ChatGPT in digital forensic investigations, concluding that while ChatGPT can enhance efficiency, human-readable insights remain valuable. In parallel to improving model capabilities through fine-tuning and prompting techniques, operational solutions are being developed to embed LLMs within forensic investigations. For example, Sharma et al., [32], introduced ForensicLLM, a fine-tuned version of LlaMA-3.1-8B trained on digital forensic literature and artefacts. The evaluations demonstrate that ForensicLLM enhances response accuracy, relevance, and reliability, thereby providing forensic investigators with reliable support in real-world forensic settings.

Local LLMs can be employed to address the limitations of cloud-based systems. These models run entirely on user-controlled hardware, enabling the data to be processed in trusted environments. Employing local models decreases the risk of data breaches and enhances control over sensitive case material in domains such as law enforcement or defence [3, 22, 23]. The deployment of these models can be optimised using fine-tuning techniques to yield better results. However, we have chosen to work with local models as-they-are, i.e., not fine-tuned. Due to the rapid release cycle of base models (at this moment) fine-tuned models will likely be outdated and resources for fine-tuning are not worth the time and effort [28].

In this context, the Netherlands Forensic Institute (NFI) has developed the Digital Forensics as a Service (DFaaS) platform, Hansken [36, 37]. DFaaS enables investigators to query digital material directly, thereby simplifying the embedding of Artifical Intelligence (AI) in the investigative process. Recent tools such as Hansken Copilot and BelkaGPT illustrate how integrating LLMs into platforms like DFaaS can enhance forensic workflows while preserving privacy [4, 17].

3 METHOD

This section outlines the methodology employed in this research. It first presents a general framework for creating forensic reports using LLMs. This includes a two-stage process to accommodate local LLMs, involving the independent analysis and synthesis of evidence. To systematically assess the quality of the produced reports, a multidimensional evaluation methodology is developed.

3.1 Conceptual Framework and Research Design

This section details the framework that serves as a blueprint for the subsequent experimental design. The framework is illustrated in Fig. 1. The experimental pipeline begins with the collection of input data, which compromises both forensic artefacts (e.g., chat conversations) and contextual information, (e.g., case background and witness statements). A cloud-based LLM is then used to construct a ground-truth report from this data. This benchmark serves as a reference to which reports generated by local LLMs are evaluated. In Experiment 1, a structured prompting strategy is used to guide the behaviour of the local LLMs during report generation. The resulting forensic report are subsequently assessed using a multidimensional metrics framework to systematically score the quality and accuracy of the created output against the benchmark A complete guide of the is provided in the project README [2]

[1] See for the complete repository: https://github.com/NetherlandsForensicInstitute/local-llm-chat-report-benchmark

[2] https://github.com/NetherlandsForensicInstitute/local-llm-chat-report-benchmark/blob/main/README.md

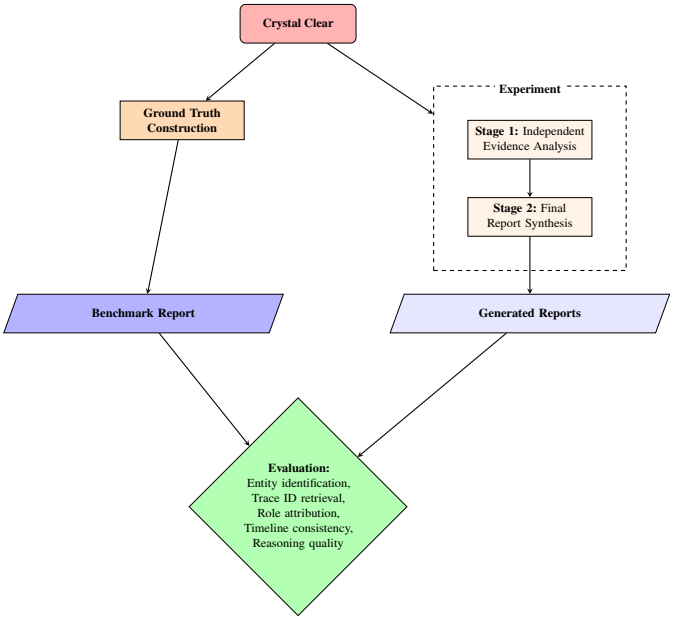

Figure 1: Workflow of the benchmark report, the experiment and the proposed methodology. The benchmark produces the ground truth report. The experiment applies the proposed methodology, consisting of two stages, to create forensic reports for evaluation.

3.2 Methodology for Evaluation

We evaluate report quality with a multidimensional, weighted framework aligned to operational priorities in digital forensics. The goal is to rank models robustly rather than providing an absolute quality measure. While each individual dimension is important in its own right, for the sake of brevity, we also distilled the results into a single summarised value. The weighing of the dimensions was conducted to enable a qualitative comparison of the reports. We deliberately selected dimensions that are relevant to the forensic investigator and aligned with the specific goals of this study. In our view, commonly used standard methods, such as BLUE, BERT, or ROUGE, are not sufficiently robust in capturing semantic nuance to allow for a meaningful comparison of these reports. These metrics compare overlapping n-grams units rather than evaluating the semantic meaning [6]. The assigned weights are therefore based on our informed judgement of what constitutes as relevant criteria. As a result, our approach to combine everything into one number has an inherent subjective component. Scoring follows a semi-automated process: a high-capability LLMs performs an initial comparison against the ground truth, after which human verification corrects potential model bias and validates high-impact items. Partial credit is granted where appropriate (e.g., for timelines and trace IDs). The following dimensions are included in the evaluation:

Entity accuracy (weight 0.25). Measures whether all relevant individuals are correctly identified from the chats. Because missing a key person is more harmful than mentioning an extra irrelevant name, the scoring prioritises recall of actual actors and only lightly penalises spurious additions. Initial scoring is automated; final labels (true/false positive) are verified manually.

Trace ID retrieval (weight 0.20). Assesses whether evidence trace IDs are reproduced accurately and usefully. We award graded credit: *complete* (exact and fully reproducible), *partial* (structure mostly correct but missing child layers), and *root-only* (prefix correct, children incorrect/missing). If two different IDs reference the same underlying message on various devices, they are considered a match. Automated matching is followed by manual cross-checks in Hansken for a sample and all high-impact items.

Role attribution accuracy (weight 0.15). Evaluates whether assigned roles (e.g., leadership, coordination, logistics) are consistent with the conversations. We credit categorical correctness even if the wording differs, and penalise contradictions or speculative roles that are unsupported by the chats. Automated comparison to the ground-truth taxonomy is followed by human review of disagreements.

Timeline consistency (weight 0.15). Checks whether event–date pairs are reconstructed correctly. We use four outcomes: (1) *Correct* (date and event match), (2) *Date correct, event wrong*, (3) *Date wrong, event correct*, (4) *Incorrect* (both wrong). Partial credit is given for cases (2)–(3). Automated alignment produces the initial labels; humans verify edge cases and all critical events.

Factual consistency (weight 0.15). Rates how faithfully the report reflects the source chats (no hallucinations, accurate paraphrases). Claims, names, dates, amounts, and quoted content are checked against the ground truth; unsupported statements are penalised. We combine automated evidence checks with human spot audits focused on high-stakes passages.

Reasoning quality (weight 0.10). Judges the clarity and plausibility of links drawn between events and individuals, the transparency of justifications, and the avoidance of speculative leaps. Reasoning that explicitly ties conclusions to cited messages or trace IDs is rewarded. Automated rubric scoring is double-checked by a human reviewer.

Scoring workflow. For each model, category scores are computed from the automated comparison and corrected by manual verification of entities and trace IDs, as these are considered most important. The final overall score is the weighted sum across categories.

4 EXPERIMENT

This section describes the specific implementation of the methodology outlined above. The benchmark report, experiment, results, and source code are available on GitHub (Footnote 1).

4.1 Methodology for Report Creation

This research investigates the capability of local, open-source LLMs to create structured investigative reports in a forensic context. The study follows a comparative design structured around the creation of a benchmark and an experiment. Both the benchmark and the experiment utilise the same dataset, which contains chat conversations from seized mobile devices. This approach is similar to [22] in which only cloud-based models were compared with each other. However, due to the sensitive nature of the data, local models could be better suited for report generation.

As described in Section 3, the benchmark is generated using Gemini 2.5 Pro— a commercial, cloud-based LLM with long-context and high-reasoning capabilities and is subsequently manually verified [11].

The experiment concerns creating forensic reports using local LLMs through a two-stage process involving independent evidence analysis and synthesis. Additionally, the quality of the created reports is evaluated through a multidimensional framework.

Fig. 1 outlines the general research design and relationship between the benchmark report and the experiment.

4.2 Dataset

This study employed datasets derived from a fictitious case called "Crystal Clear", designed for the Hansken Academy [13]. The case revolves around Quan and Joseph, who were arrested at the airport with a suspicious suitcase containing a large amount of money. The datasets contain chat conversations extracted from seven mobile devices involved in this simulated criminal investigation. The datasets comprise a mix of personal messages, spam messages, interactions with service bots, and crucial exchanges related to criminal activities. To limit the total number of tokens, group conversations involving more than 10 participants were excluded from the dataset. Table 6 in Appendix A contains an overview of the devices and their conversations.

4.3 Model Selection and Setup

Five state-of-the-art LLMs were selected for evaluation. At the time of the experiment, these models were selected as they are small enough to run locally while still delivering strong performance. The models were selected based on several technical specifications influencing the output quality, including the context window, parameter count, and quantisation method. The context window refers to the maximum number of tokens a model can process in a single prompt. The context window influences an LLM's output quality, relevance, and coherence and defines how much the model can see and use when creating a response [34]. The specific models and their configurations are detailed in Table 7. Given the rapid pace of development in LLMs, performance characteristics evolve quickly and alternative models might outperform those evaluated here if the expreriment were repeated today. Therefore, our proposed framework can quickly be adapted such that new models can easily be evaluated with minimal effort.

The more parameters the model has and the more precise the quantisation method is, the better the text quality should be [21]. Quantisation is a technique that reduces computational costs by lowering the numerical precision of a model's parameters. A key advantage of quantisation is that it enables LLMs to be deployed in resource-constrained settings, such as on local hardware [18] [7]. The choice for these characteristics also depended on the computing power and time required to create the text.

4.4 Prompt Design

We use a small set of prompts that mirror the experimental workflow: (P1) ground-truth report, (P2) stage-1 per-part summarisation, (P3) stage-2 cross-part synthesis, and (P4) structured whole-report evaluation. We also use a timeline-only evaluation (P5) and two auxiliary prompts for trace-ID equivalence and validation. Full templates are available in the repository[3].

Common design choices. Across prompts we (i) require explicit citation of evidence *trace IDs* with each claim; (ii) prioritise relevance while preserving all potentially probative details; (iii) enforce structured outputs (tables/lists for actors, roles, and timelines) with machine-parsable sections; (iv) keep wording model-agnostic for fair cross-model comparison; and (v) impose an **ID policy** that forbids inventing unseen trace IDs and mandates verbatim reproduction of IDs found in evidence. Prompts further discourage speculative attribution and encourage stating uncertainty explicitly.

*P1 — **Benchmark report**. Purpose:* generate a comprehensive benchmark investigative report from the aggregated dataset. *Inputs:* case background + full chat corpus (all devices). *Output:* a single report with (a) an actor table (names, roles, activities), (b) a chronological timeline (dates/times), (c) inline references to exact trace IDs for each finding, and (d) an explicit statement of the likely organiser/"mastermind" with evidence. *Notes:* the "disciplined detective" persona anchors evidentiary rigour; this report is the comparison target throughout Section 3. For transparency, we also retain the model's hidden "thoughts", if present, as a separate artefact.

*P2 — **Experiment.** Stage 1 per-part summarisation. Purpose:* reduce context while retaining evidentiary specificity. *Inputs:* a *single token-bounded part* of the corpus (post-split; typically ∼29k tokens). *Output:* summaries of *relevant* exchanges (who/what/why) with exact trace IDs; irrelevant content is omitted, but any relevant content must be exhaustively captured to avoid missed clues. *Format:* a record-oriented structure (timestamp, participants, concise fact summary, TraceID) suitable for downstream concatenation. *Note:* Experiment is done following Section 3.

*P3 — **Experiment.** Stage 2 cross-part synthesis. Purpose:* combine P2 outputs into a unified cross-device report. *Inputs:* the P2 summaries (all parts for a given model, or the concatenated summary for Gemini). *Output:* a complete report integrating all summaries, including (i) a comprehensive suspect/actor table (no relevant individual omitted), (ii) an accurate timeline that cites supporting trace IDs for each event, and (iii) consistency checks for names, aliases, and dates. *Constraint:* every claim in the actor table or timeline must be directly supported by at least one cited trace ID.

*P4 — **Whole-report evaluation**. Purpose:* score a model report (P3) against the P1 ground truth. *Inputs:* (1) the P1 report, (2) the model's P3 report, and (3) the trace-ID equivalence map (see Aux. prompts). *Output:* category-level scores (with justifications) and itemised evidence checks for: Entity Accuracy, Role Attribution, Timeline Completeness/Accuracy, Factual Consistency, and Reasoning Quality; plus a final weighted score. *Procedure:* this is the automated first pass for our semi-automated scoring; human review targets high-weight categories and disagreements.

*P5 — **Timeline-only evaluation**. Purpose:* isolate timeline quality from the rest of the report. *Inputs:* ground-truth timeline and the model's timeline. *Output:* one row per model summarising counts

[3] https://github.com/NetherlandsForensicInstitute/local-llm-chat-report-benchmark/blob/main/README.md#prompts–quick-index

of *Correct*, *Wrong event*, *Wrong date*, and *Missed*, under a strict one-to-one alignment policy. *Use:* supplies Table 4 and supports error-type analysis independent of entity/role scoring.

Auxiliary prompts — trace-ID equivalence and validation. **A1)** Create a Trace-ID Equivalence Map: detects duplicate/near-duplicate conversations across devices/exports and outputs an equivalence table used during evaluation to award credit for recognised duplicates. **A2)** Create a Trace-ID Validation Script: generates a Python checker that verifies every mapped ID against the raw corpus, ensuring that evaluation only credits IDs that truly exist.

Output formatting and parsing guarantees. All prompts adopt explicit section headers and bullet/table schemas to support deterministic parsing (e.g., CSV/JSON-friendly tables for roles and timelines). We also require:

- **Trace-ID discipline:** no invented IDs; cite verbatim; list all supporting IDs per claim.
- **Attribution hygiene:** separate facts from hypotheses; label uncertainty.
- **Completeness:** forbid dropping relevant records once marked as relevant in P2.

Rationale for the two-stage design. Local LLMs have smaller context windows; P2→P3 prevents truncation by first extracting part-level salient content (with trace IDs) and then synthesising across parts/devices, preserving auditability and enabling fair cross-model comparison under identical constraints.

4.5 Procedure

This section summarises how the benchmark is created and how the experiment was executed. Prompt templates (P1–P5 and auxiliary prompts) are described in Section 4.4; a complete, citable reproduction guide (with file paths and artefacts) is provided in the project README.

Benchmark Report Creation. We used a capable global model (Gemini 2.5 Pro) with prompt P1 to produce a single, comprehensive investigative report from the aggregated corpus (V3), using the full dataset and casebackground as inputs. This process was executed in a *single pass* leveraging Gemini 2.5 Pro's long context windows, with Temperature set to 1.0 and Top-P to 0.95, and runs were repeated. Temperature controls the randomness and creativity of the model's responses. A high temperature allows for more diverse outputs, whereas a lower temperature results in more deterministic and consistent responses [33]. Top-p refers to a sampling method, where the LLM selects the next word from the smallest set of tokens whose combined probability exceeds a threshold p, in this case 0.95. This method ensures that only the most likely tokens are considered [35]. The outputs consists of (i) the ground truth report and (ii) the model's optional "thoughts" which were preserved as a separate artefact; both files are listed by filename in the repository README [4]. We have only run this once, as Gemini 2.5 Pro already produced a thorough and accurate report. In contrast to prior work **we did not perform manual validation/correction** of the ground truth in this study.

[4]https://github.com/NetherlandsForensicInstitute/local-llm-chat-report-benchmark/blob/main/README.md#experiment-1--ground-truth-creation

Dataset split (prep for the experiment). To accommodate local context limits, we split V3 into <29k-token parts while preserving conversation/device boundaries where possible. The 29k-token parts is chosen as this is close to the maximum context window of the local LLMs. The dataset overview and links to all parts are provided in the README (Appendix A — Dataset) [5]. In our runs this resulted in `10` parts numbered `01`–`10`.

Experiment — Local & global model evaluation (two-stage). We used a two-stage pipeline to ensure verifiability and cross-part coherence, as shown in Figure 2.

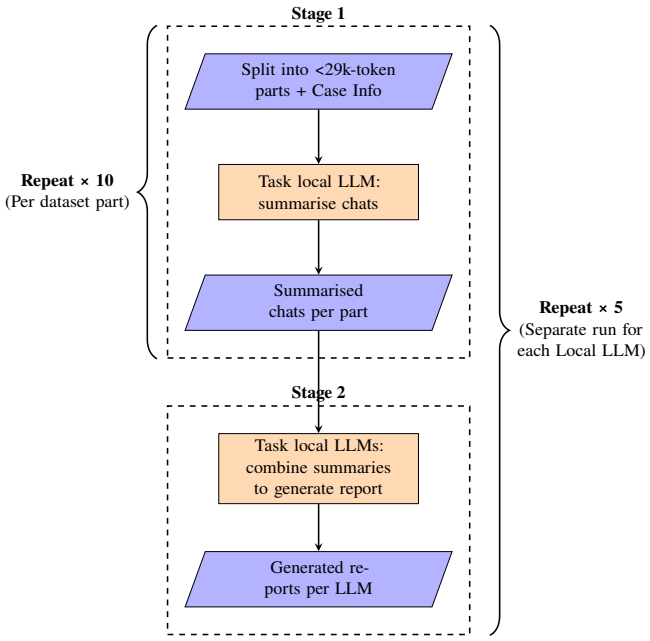

Figure 2: Experiment workflow (per local LLM). Stage 1: per-part analysis—each of ten (<29k-token) dataset parts is summarised with exact trace IDs (P2). Stage 2: cross-part synthesis—summaries are merged into one investigative report with suspect/role table and timeline (P3). This is repeated for five local models. Reports are then scored with the weighted framework using P4 (and timeline-only with P5).

Stage 1 (per-part extraction): For each split part, P2 produces record-oriented summaries of *relevant* exchanges with verbatim Trace IDs; irrelevant content is omitted. The provided runner (`experiment_2.py`)[6] executes Stage 1 for a fixed list of local models and writes per-model, per-part summaries plus timing/token statistics to CSV. The runner includes recovery: if a summary file already exists, that file is skipped.

Stage 2 (cross-part synthesis): All Stage 1 summaries for a model *from parts* `01` *through* `10`*, in order* are concatenated with explicit – `START/END` – markers and passed to P3 to produce one structured report (actor/role table and a chronological, Trace-ID–cited timeline).

[5]https://github.com/NetherlandsForensicInstitute/local-llm-chat-report-benchmark/tree/main/split_chatconversations

[6]https://github.com/NetherlandsForensicInstitute/local-llm-chat-report-benchmark/blob/main/experiment_2.py

Reports are timestamped; tokens and duration are logged. A recovery guard skips Stage 2 only if a final report for that model already exists—*it does not drop any part summaries*. (We assume zero-padded filenames to preserve alphanumeric sort order.)

Global model comparison (Gemini 2.5 Pro, manual two-stage). We followed the same two-stage logic with Gemini 2.5 Pro. Still, we executed it *manually* because Gemini 2.5 Pro was not available through our LM-Studio (OpenAI-compatible) local REST setup and had to be run via the online Gemini interface. Stage 1 produced a single concatenated summary (not per-part); Stage 2 transformed that summary into the final report using P3. Both artefacts are listed in the repository.

Evaluation (semi-automated). We used P4 (whole-report evaluation) to score model reports against the ground truth with category-level justifications and a final weighted score, and P5 (timeline-only evaluation) to isolate alignment errors (*Correct*, *Wrong event*, *Wrong date*, *Missed*) under strict one-to-one matching. Evaluation accounting uses a Trace-ID *equivalence map* (to credit duplicates across devices/exports) and a generated *validator* script (to assert that credited IDs exist in the raw corpus); we also recompute key metrics programmatically for sanity checking. The README (Evaluation & Tables) lists the prompt files, scripts, and table outputs used to produce the reported scores.

5 RESULTS

This section presents the findings of the benchmark creation and the experiment, with a focus on the two-stage methodology.

While the phrasing varied, the content of entities, roles, and timelines remained consistent. For evaluation, we count only the trace IDs that appear in the *timeline* section (IDs elsewhere in the report are ignored). Table 1 summarises the ground-truth report characteristics.

Table 1: Ground-truth report generated with Gemini 2.5 Pro. *Trace ID count refers only to IDs cited in the timeline; IDs elsewhere in the report were ignored for evaluation*.

Model in Google AI Studio	Gemini 2.5 Pro
Prompt plus All conversations	250,990
Time to generate ground truth	86.9s
Length benchmark report	5,307
#Key individuals	8
#Timeline events (with trace IDs)	14
Temperature	1.0 (default)
Top–P	0.95 (default)

To ensure reliability, the report generated by Gemini 2.5 Pro [7] was validated (but not adjusted) against a document containing factual information about the Crystal Clear case. The report accurately identified all key individuals and assigned primary roles and responsibilities to most suspects. It also produced a detailed, accurate timeline that correctly synthesised conversations across multiple people and days, but also noticed that some events were missing that could be considered interesting. In general, these results are consistent with expectations for long-context models: when the full evidence set fits in-window, stronger retrieval–and–synthesis behaviour tends to emerge [20]. Overall, Gemini 2.5 Pro demonstrates strong reasoning while maintaining practical efficiency, identifying all key actors and roles and constructing a coherent narrative of the criminal operation.

5.1 Evaluation

The experiment evaluates five local models in the two-stage pipeline described in Section 4: (i) per-device chat summarisation (Stage 1) and (ii) cross-device report synthesis (Stage 2). Prompts and inputs follow Section 4.4, and scoring uses the weighted framework. Ground truth is the Gemini-generated report. We organise findings around three criteria: (1) *forensic task performance*; (2) *overall report quality*, and (3) *computational efficiency*.

5.1.1 Computational Efficiency. In line with our research questions on computational efficiency, Table 2 shows that the reasoning models impose a clear time penalty, with significant differences between them. Among the reasoning models, OpenAI Gpt-oss-20b (GPT-OSS-20B) is the fastest overall, followed by Qwen 3 14B (Qwen-3-14B) and Microsoft Phi 4 Reasoning (Phi-4-Reasoning) . The non-reasoning Google Gemma-3 12B (Gemma-3 12B QAT) remains competitive, while Google Gemma-3 27B (Gemma-3 27B QAT) is an outlier with prohibitive runtime. Detailed per-summary timings are provided in a supplementary table.[8]

Table 2: Runtime of all models in Stage-1 (summaries) and Stage-2 (reports). Stage-1 times include both summarisation and any associated *thinking* (thoughts) steps. Note that GPT-OSS-20B, Qwen-3-14B, and Phi-4-Reasoning are reasoning models, which typically incur extra computation time compared to instruction-tuned models.

Model	s1 avrg	s1 total	s2 total	total
GPT-OSS-20B	1m10s	11m37s	2m25s	14m2s
Qwen-3-14B	1m36s	15m57s	2m27s	18m24s
Gemma-3 12B QAT	1m53s	18m54s	2m47s	21m41s
Phi-4-Reasoning	8m0s	1h20m3s	18m0s	1h38m3s
Gemma-3 27B QAT	56m44s	9h27m21s	1h35m10s	11h2m31s

5.1.2 Forensic task performance. How accurately do the models identify relevant individuals? Across models, all key actors are recovered from the chats. Gemma-3 12B QAT and Qwen-3-14B also introduce an irrelevant individual (Charlie and Kate, who are false positives). GPT-OSS-20B fails to reconcile nb/NB with Nerijus Bos. Phi-4-Reasoning achieves the highest overall identification rate, whereas Gemma-3 27B QAT shows the most over-inclusion. See figure in our repository[9] for the per-model identification patterns.

How well do the models assign roles to identified individuals? Role attribution shows strong categorical agreement on central figures (e.g., Joseph and Quan are consistently assigned to Logistics;

[7] https://github.com/NetherlandsForensicInstitute/local-llm-chat-report-benchmark/blob/main/output/evaluation/gemini-2.5-pro-report-from-summaries-evaluation.md

[8] https://github.com/NetherlandsForensicInstitute/local-llm-chat-report-benchmark/blob/main/latex_figures_and_tables/tables/table_4_stage_1_timing.md

[9] https://github.com/NetherlandsForensicInstitute/local-llm-chat-report-benchmark/blob/main/latex_figures_and_tables/figures/4_roles_from_new_experiment.pdf

leadership/coordination is stably attributed to Antonio and Nerijus). Disagreement increases for peripheral actors, where labels vary in specificity and certainty. The complete figure can be seen in Github (See footnote 9) for category-level consistency and model-specific labels.

Table 3: Retrieved IDs. Columns: Corr. = fully retrieved trace IDs, Part. = small details missing, R.O. = root-only, Miss. = missed event, Perf. = Trace ID Retrieval Score.

Models	Corr.	Part.	R.O.	Miss.	Perf.
Gemini 2.5 Pro (GT)	14	0	0	0	1
Gemini 2.5 Pro (staged)	13	0	0	1	0.93
GPT-OSS-20B	13	0	0	1	0.93
Phi-4-Reasoning	10	0	2	2	0.76
Qwen-3-14B	7	0	0	7	0.50
Gemma-3 12B QAT	1	1	2	10	0.17
Gemma-3 27B QAT	0	0	4	10	0.10

How reliably do the models retrieve trace IDs required for evidentiary traceability?
Table 3 shows three clear performance tiers in trace ID recovery. Gemini 2.5 Pro (staged) and GPT-OSS-20B closely match the ground truth, recovering nearly all trace IDs exactly. Phi-4-Reasoning and Qwen-3-14B achieve partial recovery with an increasing number of incomplete or root-only matches. In contrast, the QAT Gemma variants perform poorly, rarely recovering full trace IDs. Overall, higher-end (cloud and reasoning) models demonstrate stronger evidentiary traceability, whereas the smaller QAT models often fail to reconstruct long, structured identifiers end-to-end.[10]

How consistent are model-generated timelines with the ground truth?
Table 4 reveals substantial variation in timeline fidelity across models. Gemini 2.5 Pro (staged) achieves the strongest overall performance, with high coverage and few errors. Among local/OSS models, Phi-4-Reasoning performs best, followed by GPT-OSS-20B, which achieves moderate correctness but tends to over-generate spurious events. The QAT Gemma variants struggle to align with the ground truth, producing many incorrect or extraneous events, while Qwen-3-14B under-generates and misses a significant fraction of ground-truth events. Overall, even the strongest models fail to comprehensively cover all events, whereas weaker models either omit many ground-truth items or introduce substantial noise. Some models reported correct events, but the ground truth report, in fact, ignored that. Category definitions (Correct, Wrong, Wrong event, Wrong date, Missed) follow the one-to-one alignment prompt used for scoring.

5.1.3 Overall report quality. Table 5 summarises overall quality and efficiency across models. GPT-OSS-20B achieves the highest overall quality, while also being the fastest, indicating a favourable quality-runtime tradeoff. Phi-4-Reasoning exhibits strong performance, with strengths in entity and timeline accuracy, albeit at a substantially higher computational cost. Qwen-3-14B sits mid-pack, combining moderate quality with competitive runtime. In contrast, the QAT Gemma variants lag in both quality and practicality, with Gemma-3 27B QAT exhibiting extreme runtime overhead and Gemma-3 12B QAT underperforming across most dimensions. The full rubric, scoring prompt, and example I/O are available in our repository [11]. Reasoning models GPT-OSS-20B, Qwen-3-14B and Phi-4-Reasoning appear to perform better overall.

Table 5: Performance scores and total generation time (Stage 1 summaries + thoughts, and Stage 2 report).

	GPT-OSS-20B	Phi-4-Reasoning	Qwen-3-14B	Gemma-3 27B QAT	Gemma-3 12B QAT
Entity	0.81	**0.88**	0.81	0.81	0.69
Trace ID	**0.93**	0.76	0.50	0.10	0.17
Role	**0.71**	0.43	0.43	**0.71**	0.33
Timeline	0.64	**0.79**	0.43	0.57	0.29
Factual	**0.77**	0.73	0.73	0.69	0.61
Reasoning	**0.60**	0.50	**0.60**	0.50	0.40
Weighted final	**0.77**	0.71	0.60	0.57	0.43
Total time	14m 2s	1h 38m 3s	18m 24s	11h 2m 31s	21m 41s

Time–quality trade-off. The scatterplot in figure 3 illustrates the tradeoff between the quality and stage 2 runtime. GPT-OSS-20B and Qwen-3-14B emerge as the most efficient models, combining relatively high scores with low latency. Phi-4-Reasoning achieves strong quality, but at a higher time cost. In contrast, the QAT Gemma variants offer weaker quality-efficient tradeoffs, whereas Gemma-3 27B QAT is a clear runtime outlier for only modest quality. Overall, reasoning-oriented models tend to score higher, but efficiency varies substantially across deployments.

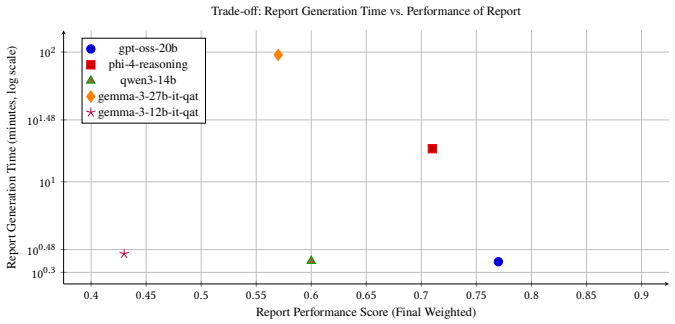

Figure 3: Trade-off (log-scaled) between overall report performance (Final Weighted score) and Stage-2 report generation time. Lower is faster; higher is better.

6 DISCUSSION

How should these findings be interpreted within the context of chat-centric digital forensics? Specifically, to what extent do local LLMs reliably structure entities and roles while preserving evidentiary

[10] https://github.com/NetherlandsForensicInstitute/local-llm-chat-report-benchmark/blob/main/README.md#timeline-only-scoring

[11] https://github.com/NetherlandsForensicInstitute/local-llm-chat-report-benchmark/tree/main/output/evaluation

Table 4: Timeline Comparison. GT = #ground-truth events; Model = #model events. ✓ correct actors/action and time; ✗ wrong event; ● same situation/actors but semantics wrong; ▲ actors/action match but time wrong; Missed: GT-(✓+●+▲).

Ground Truth	GT	Model	✓ Correct	✗ Wrong	● Wrong event	▲ Wrong date	Missed
Gemma-3 12B QAT	14	12	3	6	1	2	8
Gemini 2.5 Pro (staged)	14	19	12	6	0	1	1
Gemma-3 27B QAT	14	22	5	14	3	0	6
GPT-OSS-20B	14	23	8	13	2	0	4
Phi-4-Reasoning	14	20	9	8	1	2	2
Qwen-3-14B	14	8	2	2	4	0	8

justifications (trace-IDs, timelines), and under which verification protocols and dataset constraints can their outputs be considered operationally trustworthy?

6.1 Interpretation of Key Findings

Factual accuracy and contextual reasoning. Local models reliably surfaced key *entities* and produced usable *role* assignments under strict prompting, but residual role misattribution and terminology drift persisted. Two factors likely contributed: (i) Stage-1 summaries inevitably compress evidence; any omission propagates to Stage-2 and weakens cross-part inferences, and (ii) the need to reconcile aliases across parts in Stage-2 increases the risk of conflating actors. Even so, the study shows that local LLMs can consistently identify core roles in a criminal network and hold value in assisting investigators, particularly when structuring relevant artefacts.

Trace-ID retrieval and evidentiary precision. Verbatim citation of trace IDs proved to be a challenge. Despite the equivalence map and validator, the local models frequently produced partial or malformed IDs—consistent with tokenisation issues for structured strings. This limits the direct evidentiary use: outputs are best treated as leads that *must be verified* (IDs and linked snippets) before being included in a forensic narrative.

Timeline construction. All models recovered salient events, yet timelines remained incomplete and occasionally inconsistent. The timeline-only analysis exposed characteristic failure modes—*missed*, *wrong date*, and *wrong event*—that align with long-context position effects and the difficulty of stitching evidence across parts. In practice, the two-stage design helps constrain drift, but human review should focus on event coverage and date alignment, with targeted checks on cross-part links. Some local models actually reported relevant events that were not included in the ground truth, which suggests that a two-stage approach may have some benefits. We have not investigated this further.

6.2 Implications for Practice and Research

Our results situate local LLMs as promising but imperfect collaborators in chat-centric investigations. In line with Michelet and Breitinger's findings on draft generation and triage value, despite factual imperfections [27], the local models here consistently surfaced salient entities and produced workable role hypotheses. Where they fell short was in evidentiary precision—most notably verbatim *Trace-ID* citation and the completeness/consistency of chronological narratives—which are precisely the behaviours that matter most for admissible, auditable reporting.

A practical reading of these findings is that local models can lower the cost of early sense-making without replacing expert judgement. The two-stage design means that omissions at Stage 1 can propagate to Stage 2, influencing role attribution and event linking. Our timeline-only analysis made this visible through characteristic failure modes (*missed*, *wrong date*, *wrong event*). At the same time, evaluation hygiene—specifically, using a *Trace-ID equivalence map* to avoid double-penalising duplicates and a *validator* to ensure IDs exist in the raw corpus—proved essential to interpreting performance credibly.

Operationally, the fully automated on-prem pipeline reframes the integration question from feasibility to oversight: where should human effort be concentrated? Our experience points to two areas: (i) targeted verification of *Trace-ID* references and cross-links that underpin claims, and (ii) consolidation of timelines across parts, where minor omissions can have outsized effects on narrative coherence. Under this regime, local LLMs are best treated as triage partners that accelerate organisation and hypothesis formation. At the same time, investigators retain responsibility for validating the cited evidence and chronology before including them in the evidentiary narrative.

6.3 Limitations

While this study provides valuable insights into the potential of using local LLMs to enhance digital forensic reports, several limitations must be acknowledged:

Synthetic case and external validity. We use the synthetic *Crystal Clear* case to enable sharing and controlled scoring. Synthetic evidence is cleaner and more coherent than real casework, which often contains contradictory, incomplete, or noisy artefacts [14]. As a result, absolute performance may be optimistic and generalisation to messy, multi-artefact investigations remains to be demonstrated.

Model behaviour, omissions, and automation. Local LLMs lag behind a strong cloud baseline in terms of factual precision (verbatim Trace-ID citation) and temporal consistency. Because Stage 1 operates on <29k-token parts, any omission there propagates into Stage 2 and can affect roles/timelines. Hallucinations were infrequent but present, and the pipeline still requires operator supervision (prompting choices, reruns, sanity checks)—it is not yet a push-button, fully automated report generator.

Scope and scalability constraints. This study focused on a single crime scenario. While this allowed for a controlled environment, it also narrows the applicability of the findings. Different types of crime (e.g., financial fraud or cyberstalking) involve different artefacts,

timelines, and behavioural patterns, which may pose different or additional challenges for LLM-based analysis. We also focus on chat messages; other artefacts (e.g., call logs, browser history) may change outcomes and limit generalisability. Stage 1 scales to large corpora via per-part chunking, but Stage 2 does not scale. As collections grow, the *concatenated* Stage 1 summaries for a model can exceed the Stage 2 context window, causing synthesis to fail. In such cases, an intermediate "Stage 1.5" becomes necessary, possibly introducing new omissions or broken cross-references.

7 CONCLUSIONS AND FUTURE WORK

The results show that all local LLMs demonstrate promising capabilities in identifying key individuals, particularly **Phi-4-Reasoning**. However, all models struggle with more complex tasks. Models failed to retrieve complete trace IDs or construct coherent timelines. Trace ID retrieval was inconsistent, with only **GPT-OSS-20B** achieving high accuracy (0.93). The other models frequently failed to retrieve correct trace IDs, and creating a coherent timeline proved to be challenging. These challenges reflect difficulties in maintaining contextual continuity across fragmented evidence.

These findings suggest that local LLMs are not yet suitable for fully autonomous use in digital forensic investigations. Their current limitations, i.e., their susceptibility to omissions and inconsistencies, necessitate human-in-the-loop feedback. Nonetheless, they offer value as assistive tools, particularly for initial evidence triage and report structuring under expert supervision. Enhancing their performance will require fine-tuning on forensic data and improving prompt engineering. With continuous development, local LLMs could benefit the forensic investigative process while preserving privacy and transparency.

Ethical Considerations. While LLMs can support DF investigations, they must remain assistive tools—not replacements for human expertise. Over-reliance risks bypassing expert analysis and validation [38]. At the same time, abstention is not value-neutral: law enforcement faces rising caseloads across DNA, controlled substances, and digital forensics, and routine casework already squeezes time for innovation. If responsibly deployed LLMs can shorten reporting cycles without degrading quality, refusing to use them may delay justice and slow progress. Ethical responsibility thus cuts both ways: guard against misuse while enabling society to benefit from efficiency gains.

Human investigators should therefore define clear use boundaries—e.g., assistance in data analysis or evidence extraction—not legal judgment. Maintain strict human oversight, and ensure every LLM-surfaced finding is independently verified in a forensic tool by (1) confirming the trace ID is valid and (2) confirming that the trace ID resolves to an artefact that actually supports the claimed finding. Transparency of the LLM itself is less critical when the chain from claim → trace ID → artefact is reproducibly validated.

Future Work. Real-world digital forensic investigations often include additional artefacts (call logs, documents, emails, browser history). Future work will test whether local LLMs can operate across these heterogeneous sources while maintaining the quality observed on chat data—focusing on cross-artefact consistency, provenance via trace IDs, and robustness under realistic evidence noise.

Beyond the current two-stage pipeline, we will explore human-in-the-loop agentic RAG: specialised on-prem agents (e.g., Device Analyst, Timeline Builder, Auditor) that iteratively retrieve, reason, and verify against evidence stores, with investigators approving or correcting intermediate claims. This draws on recent search-augmented reasoning (think→search→answer) to decide when to retrieve, what to query, and when evidence is sufficient [19]. The aim is simple: assistants that not only summarise but also justify—while humans stay firmly in the loop.

REFERENCES

[1] Marah Abdin et al. 2025. Phi-4-Reasoning Technical Report. arXiv:2504.21318 [cs.AI] https://arxiv.org/abs/2504.21318.

[2] Jonathan Adkins, Ali Al Bataineh, and Majd Khalaf. 2024. Identifying Persons of Interest in Digital Forensics Using NLP-Based AI. *Future Internet* 16, 11 (2024), 426.

[3] Isabel Barbera. 2025. *AI Privacy Risks & Mitigations – Large Language Models (LLMs)*. Support Pool of Experts report. European Data Protection Board. https://www.edpb.europa.eu/system/files/2025-04/ai-privacy-risks-and-mitigations-in-llms.pdf Prepared under the EDPB Support Pool of Experts programme (April 2025).

[4] Belkasoft. 2025. BelkaGPT: The First Offline AI Assistant for DFIR Investigations. https://belkasoft.com/belkagpt. Accessed: 2025-04-04.

[5] Frank Breitinger and Alexandre Jotterand. 2023. Sharing Datasets for Digital Forensic: A Novel Taxonomy and Legal Concerns. *Forensic Science International: Digital Investigation* 45 (July 2023), 301562. https://doi.org/10.1016/j.fsidi.2023.301562

[6] Mohammed Khalid Hilmi Briman and Beytullah Yildiz. 2024. Beyond ROUGE: A Comprehensive Evaluation Metric for Abstractive Summarization Leveraging Similarity, Entailment, and Acceptability. *International Journal on Artificial Intelligence Tools* 33, 05 (Aug. 2024), 2450017. https://doi.org/10.1142/S0218213024500179

[7] Iván Palomares Carrascosa. 2025. Using Quantized Models with Ollama for Application Development. https://machinelearningmastery.com/using-quantized-models-with-ollama-for-application-development/

[8] Maxim Chernyshev, Zubair Baig, and Robin Ram Mohan Doss. 2023. Towards Large Language Model (LLM) Forensics Using LLM-based Invocation Log Analysis. In *Proceedings of the 1st ACM Workshop on Large AI Systems and Models with Privacy and Safety Analysis*. 89–96.

[9] Gemini Team. 2025. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. arXiv:2507.06261 [cs.CL] https://arxiv.org/abs/2507.06261.

[10] Gemma Team. 2025. Gemma 3 Technical Report. arXiv:2503.19786 [cs.CL] https://arxiv.org/abs/2503.19786.

[11] Google DeepMind. 2024. Gemini 2.5 Pro. https://deepmind.google/technologies/gemini/. Large language model developed by Google DeepMind.

[12] Google DeepMind. 2025. *Gemini 2.5 Pro Preview Model Card*. Technical Report. Google DeepMind. https://storage.googleapis.com/model-cards/documents/gemini-2.5-pro-preview.pdf Last updated May 9, 2025.

[13] Hansken Team. 2022. Train-the-Trainer for Hansken for Advanced Users. https://www.hansken.nl/latest/news/2022/09/07/train-the-trainer-for-hansken-for-advanced-users Accessed: Mar. 14, 2025.

[14] Shuang Hao, Wenfeng Han, Tao Jiang, Yiping Li, Haonan Wu, Chunlin Zhong, Zhangjun Zhou, and He Tang. 2024. Synthetic Data in AI: Challenges, Applications, and Ethical Implications. arXiv:2401.01629 [cs.LG] https://arxiv.org/abs/2401.01629

[15] Amey Hengle, Prasoon Bajpai, Soham Dan, and Tanmoy Chakraborty. 2024. Multilingual Needle in a Haystack: Investigating Long-Context Behavior of Multilingual Large Language Models. arXiv:2408.10151 [cs.CL] https://arxiv.org/abs/2408.10151

[16] Hans Henseler and Timo Meconi. 2025. Deep Reasoning and Large Context Windows: Next-Generation AI in Digital Forensic Investigations. In *Proceedings of the Digital Forensic Research Conference Europe (DFRWS EU)*. https://dfrws.org/wp-content/uploads/2025/03/Deep-Reasoning-and-Large-Context-Windows-Next-Generation-AI-in-Digital-Forensic-Investigations.pdf

[17] Hans Henseler and Harm MÃ 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 2023) (CEUR Workshop Proceedings, Vol. 3423)*. Braga, Portugal, 58–69. https://ceur-ws.org/Vol-3423/paper6.pdf

[18] Hugging Face. 2025. Quantization. https://huggingface.co/docs/optimum/concept_guides/quantization. Accessed: 2025-05-15.

[19] Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-R1: Training LLMs

to Reason and Leverage Search Engines with Reinforcement Learning. https://arxiv.org/abs/2503.09516. arXiv:2503.09516 [cs.CL]
[20] Steven Johnson. 2024. In the Context of Long Context. https://adjacentpossible.substack.com/p/in-the-context-of-long-context. Substack essay, accessed 2025-10-03.
[21] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling Laws for Neural Language Models. https://arxiv.org/abs/2001.08361. arXiv:2001.08361 [cs.LG]
[22] Kyung-Jong Kim, Chan-Hwi Lee, So-Eun Bae, Ju-Hyun Choi, and Wook Kang. 2025. Digital Forensics in Law Enforcement: A Case Study of LLM-driven Evidence Analysis. *Forensic Science International: Digital Investigation* 54 (Sept. 2025), 301939. https://doi.org/10.1016/j.fsidi.2025.301939
[23] BV Pranay Kumar and MD Shaheer Ahmed. 2024. Beyond Clouds: Locally Runnable LLMs as a Secure Solution for AI Applications. *Digital Society* 3, 3 (2024), 49.
[24] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the Middle: How Language Models Use Long Contexts. arXiv:2307.03172 [cs.CL] https://arxiv.org/abs/2307.03172
[25] Daniel Machlab and Rick Battle. 2024. LLM In-Context Recall is Prompt Dependent. arXiv:2404.08865 [cs.CL] https://arxiv.org/abs/2404.08865
[26] Malcolm. 2025. Local LLMs vs Cloud-Based AI: The Complete Business Guide to Choosing Your AI Automation Strategy. https://www.flowio.co.uk/blog/business-automation/local-llms-vs-cloud-based-ai-the-complete-business-guide/. Reading time: 6 minutes; published on July 11, 2025.
[27] 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. https://doi.org/10.1016/j.fsidi.2023.301683 DFRWS EU 2024 - Selected Papers from the 11th Annual Digital Forensics Research Conference Europe.
[28] 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. *Digital Threats: Research and Practice* (July 2025), 3748264. https://doi.org/10.1145/3748264
[29] OpenAI. 2025. gpt-oss-120b & gpt-oss-20b Model Card. arXiv:2508.10925 [cs.CL] https://arxiv.org/abs/2508.10925.
[30] Gabrijela Perković, Antun Drobnjak, and Ivica Botički. 2024. Hallucinations in llms: Understanding and addressing challenges. In *2024 47th MIPRO ICT and Electronics Convention (MIPRO)*. IEEE, 2084–2088.
[31] 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.
[32] Binaya Sharma, James Ghawaly, Kyle McCleary, Andrew M Webb, and Ibrahim Baggili. 2025. ForensicLLM: A local large language model for digital forensics. *Forensic Science International: Digital Investigation* 52 (2025), 301872.
[33] SungUk Shin and Youngjoon Kim. 2025. Enhancing Graph Of Thought: Enhancing Prompts with LLM Rationales and Dynamic Temperature Control. https://openreview.net/forum?id=l32IrJtpOP
[34] Swimm Team. 2025. LLM Context Windows: Basics, Examples & Prompting Best Practices. https://swimm.io/learn/large-language-models/llm-context-windows-basics-examples-and-prompting-best-practices. Accessed: 2025-05-15.
[35] Chenxia Tang, Jianchun Liu, Hongli Xu, and Liusheng Huang. 2024. Top-$n\sigma$: Not All Logits Are You Need. https://arxiv.org/abs/2411.07641
[36] Ruud van Baar, Harm van Beek, and E van Eijk. 2014. Digital forensics as a service: A game changer. *Digital Investigation* 11 (2014), S54–S62.
[37] Harm van Beek, Erwin van Eijk, Ruud van Baar, Mattijs Ugen, Jörgen Bodde, and Allard Siemelink. 2015. Digital forensics as a service: Game on. *Digital Investigation* 15 (2015), 20–38. https://doi.org/10.1016/j.diin.2015.07.004 Special Issue: Big Data and Intelligent Data Analysis.
[38] 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.
[39] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 Technical Report. arXiv:2505.09388 [cs.CL] https://arxiv.org/abs/2505.09388
[40] Zhipeng Yin, Zichong Wang, Weifeng Xu, Jun Zhuang, Pallab Mozumder, Antoinette Smith, and Wenbin Zhang. 2025. Digital Forensics in the Age of Large Language Models. *arXiv preprint arXiv:2504.02963* (2025).
[41] Jie Zhang, Haoyu Bu, Hui Wen, Yongji Liu, Haiqiang Fei, Rongrong Xi, Lun Li, Yun Yang, Hongsong Zhu, and Dan Meng. 2025. When llms meet cybersecurity: A systematic literature review. *Cybersecurity* 8, 1 (2025), 1–41.

A USED DATASET AND ITS SPECIFICATIONS

Table 6: Column C4V3 is the Crystal Clear Chat Corpus (V3). This dataset is split into ∼29k-token chunks without breaking conversations. Rows 1–10 are the split parts with the number of conversations, tokens, and device composition (device name with number of conversations). Note: Part 1 is small because a very large conversation is placed in Part 2; similarly, Part 3 is small because a very large conversation is moved to Part 4.

Part	# Conv.	Tokens	Devices (# conversations)
C4V3	**171**	**240,681**	All devices across seven mobiles
1	8	7,557	01 iPhone 7 (8)
2	12	28,463	01 iPhone 7 (12)
3	9	6,032	01 iPhone 7 (9)
4	1	29,864	01 iPhone 7 (1)
5	2	26,240	01 iPhone 7 (2)
6	35	28,410	02 iPhone 6s Plus (16), 03 Samsung S20FE (19)
7	41	27,922	03 Samsung S20FE (7), 04 Motorola moto G9 plus (27), 05 Huawei P smart (7)
8	22	28,340	05 Huawei P smart (18), 06 Samsung Galaxy J7 (4)
9	10	28,936	06 Samsung Galaxy J7 (10)
10	31	28,917	06 Samsung Galaxy J7 (12), 07 iPhone11 (19)

B MODEL PARAMETERS

Table 7: Models used and their parameters. All models were run with temperature=1 and top-p=0.95. All non-cloud models were run on NVIDIA RTX 4500 ADA.

Model	Hugging Face ID / File	Context	Quant.
Gemini 2.5 Pro (reasoning model) [9]	— (cloud-hosted)	1,048,576	NA
Gemma-3 27B QAT [10]	gemma-3-27B-it-qat-GGUF/ gemma-3-27B-it-QAT-Q4_0.gguf	32,768	Q4_0 (QAT)
Gemma-3 12B QAT [10]	gemma-3-12B-it-qat-GGUF/ gemma-3-12B-it-QAT-Q4_0.gguf	32,768	Q4_0 (QAT)
Phi-4-Reasoning (reasoning model) [1]	Phi-4-reasoning-GGUF/ Phi-4-reasoning-Q4_K_M.gguf	32,768	Q4_K_M
Qwen-3-14B [39]	lmstudio-community/Qwen3-14B-GGUF	32,768	Q4_K_M
GPT-OSS-20B (reasoning model) [29]	openai/gpt-oss-20b	32,768	MXFP4