A Comparative Evaluation of Large Language Models for Named Entity Recognition in Cyber Threat Intelligence

Aykhan Huseynli
Cybersecurity Department, Azerbaijan Technical University, Baku, Azerbaijan

***Abstract*—Cyber Threat Intelligence reports combine analytical prose with dense technical indicators, making structured entity extraction a challenging but operationally valuable task. This study presents a comparative evaluation of three large language models – Claude Sonnet 4.6, GPT-5.4, and LLaMA 4 Scout – on a manually annotated corpus of 21 real-world CTI reports across 15 entity types and 1284 ground truth instances. This study evaluates zero-shot and few-shot prompting conditions and studies the effect of iterative prompt refinement, focusing on explicit format constraints for cryptographic hash entities. Results show that Claude Sonnet 4.6 and GPT-5.4 achieve comparable performance under zero-shot conditions, with LLaMA 4 Scout trailing by a substantial margin. Few-shot prompting consistently reduces hallucination rates, but yields mixed F1 results, with exemplar cardinality emerging as a critical and underappreciated design factor. Entity extraction difficulty varies substantially across types, with technical indicator categories showing near-perfect performance and semantic categories such as tool and target sector posing the greatest challenges across all evaluated models.**

***Keywords*—Cyber threat intelligence; named entity recognition; large language models; prompt engineering**

I. Introduction

The volume of publicly available Cyber Threat Intelligence (CTI) publications has grown substantially. Organizations such as Mandiant, IBM X-Force, ESET, Kaspersky Securelist, Check Point Research, routinely publish detailed campaign analyses covering advanced persistent threat (APT) actors, malware families, and attack infrastructure. Extracting structured knowledge from these reports – threat actors, targeted sectors, malware names, Command and Control (C2) domains and URLs, spear phishing email addresses, and cryptographic indicators – is a prerequisite for populating threat intelligence platforms, maintaining situational awareness, and automating detection engineering pipelines [1].

Traditional Named Entity Recognition (NER) systems require labelled training data and struggle with the fast-moving, inconsistently formatted nomenclature of the CTI domain [2], [3]. Large language models offer a compelling alternative: they can be prompted in a structured fashion without task-specific fine-tuning, and their in-context learning capabilities allow rapid adaptation through prompt engineering. However, multi-model benchmarks of Large Language Models (LLM) performance on CTI-specific NER remain scarce, particularly benchmarks that examine iterative prompt refinement and document the failure modes that emerge in practice.

This study contributes:

- A 21-report, 15-entity-type CTI NER benchmark evaluated across Claude Sonnet 4.6, OpenAI's GPT-5.4, and Meta's LLaMA 4 Scout under zero-shot and few-shot conditions;
- A documented methodology for handling annotation standardization challenges: controlled sector and goal vocabularies, and mixed indicator types in domain/URL/email categories;
- Analysis of two qualitative failure modes unique to CTI NER: tool and malware boundary ambiguity.

The remainder of this study is organized as follows. Section II surveys prior work on CTI named entity recognition, domain-specific language models, and LLM-based information extraction. Section III describes the evaluation corpus, annotation methodology, model selection, prompt design and evolution, and evaluation metrics. Section IV presents experimental results across overall model performance, per-entity-type analysis, prompting condition comparisons, and report-level complexity effects. Section V discusses the implications of these findings for practitioners and researchers. Section VI concludes and outlines directions for future work.

To structure the evaluation and ensure that each experimental component maps to an explicitly stated inquiry, the following five research questions were identified:

RQ1: To what extent can current state-of-the-art LLMs (Claude Sonnet 4.6, GPT-5.4, and LLaMA 4 Scout) perform zero-shot NER on real-world CTI reports across a 15-type entity taxonomy, and how do they compare in terms of macro and micro precision, recall, and F1?

RQ2: Does few-shot prompting with real, out-of-corpus annotated exemplars improve NER accuracy and reduce hallucination rates compared to zero-shot baselines, and are these effects consistent across models and entity types?

RQ3: How does iterative prompt refinement – specifically, adding explicit format constraints for hash entities, CVE identifier requirements, and controlled extraction vocabularies – affect per-entity extraction accuracy across models?

RQ4: Which entity types are systematically most resistant to prompt-based extraction, and what properties of those entities (format ambiguity, semantic boundary overlap, or schema-level

definitional complexity) explain the observed performance hierarchy?

RQ5: How does CTI report length affect LLM extraction performance, and at what scale do current models exhibit measurable degradation?

II. Related Work

Research on information extraction from cybersecurity text predates the LLM era and has been dominated by sequence labeling architectures built on domain-adapted transformer encoders. Reference [2] established a strong transformer-based NER baseline on the Dataset for Named Entity Recognition in Threat Intelligence (DNRTI) corpus – a collection of over 300 publicly available threat intelligence reports – demonstrating that fine-tuned transformer models substantially outperform rule-based and statistical predecessors on CTI entity extraction. Reference [4] extended this direction by pairing BERT with whole-word masking to a BiLSTM-CRF tagging head, improving precision and recall on cybersecurity-specific entities relative to standard BERT fine-tuning. A simplified BERT-CRF architecture – omitting the intermediate BiLSTM layer – was subsequently shown to be competitive on multiple CTI datasets while also being directly comparable to GPT-3.5 in a prompt-based inference setting, providing one of the earliest quantitative bridges between fine-tuned and prompted approaches to CTI NER [5].

Domain-specific pre-training has become a prerequisite for competitive CTI NER performance. CyBERT is the first BERT model fine-tuned on a large corpus of unstructured CTI data using masked language modeling, and demonstrates consistent gains over general-purpose BERT on downstream cybersecurity tasks, including entity extraction [3]. SecureBERT 2.0, built on the ModernBERT architecture and pre-trained on over 13 billion cybersecurity text tokens, achieves state-of-the-art performance on multiple CTI NER benchmarks [6]. These domain language models constitute the principal fine-tuned reference points against which prompt-based LLM approaches are compared in the present work.

The question of what to extract is as consequential as how to extract it. Benchmark construction for CTI NER has proceeded along several lines. FeedRef2022 is a NER dataset derived from 1,854 threat intelligence reports, used to evaluate four pre-trained models on Indicators of Compromise (IoC) extraction [7]. The fragmentation of existing CTI NER datasets is addressed in [8] by harmonizing four prominent corpora – CyNER, DNRTI, APTNER, and Attacker – onto the Structured Threat Information eXpression 2.1 (STIX) standard, consolidating over 50 source tags into 21 coherent entity types and demonstrating that models trained on the unified CyberNER corpus achieve approximately 30% relative F1 improvement over naive concatenation baselines. On the other hand, CTI-HAL, a manually annotated dataset structured according to the MITRE ATT&CK framework with inter-annotator agreement validated using Krippendorff's alpha, provides a rigorous quality benchmark for CTI annotation methodology [9].

Multi-model comparative evaluation of LLMs for CTI tasks remains an underdeveloped area. Reference [10] evaluates GPT-4 for Tactics, Techniques, and Procedures (TTP) identification and threat-actor attribution, finding that standard LLMs generate noisy TTP datasets with low similarity to human-annotated MITRE ATT&CK entries, but that the frequency patterns of extracted techniques nevertheless support attribution performance above a random baseline. The most direct precursor to the present study compares LLaMA2-7B and GPT-4 under zero-shot and few-shot conditions on a CTI dataset that includes entity extraction alongside TTP classification and mitigation generation [11]. The framework presented in [11] demonstrates that commercial models outperform open-weight alternatives under few-shot conditions while acknowledging the data-residency advantage of open-weight deployment – a trade-off that motivates the inclusion of LLaMA 4 Scout as an evaluated model.

The present study extends this body of work in three respects. First, it evaluates three contemporary LLMs, Claude Sonnet 4.6, GPT-5.4, and LLaMA 4 Scout, on a broader entity taxonomy than any prior prompt-based study, covering 15 entity types spanning threat actor attributes, campaign context, and technical indicators of compromise. Second, it isolates the effects of iterative prompt refinement and few-shot exemplar design as experimental variables, documenting failure modes, system utility over-extraction, and hash format confusion, which are specific to the CTI domain and absent from general NER benchmarks. Third, all evaluated models operate without task-specific fine-tuning, providing a prompt-only point of comparison against the fine-tuned supervised baselines surveyed above; Section V examines this comparison in detail.

III. Methodology

A. Evaluation Corpus

The corpus comprises 21 publicly available CTI reports from Mandiant, IBM X-Force, ESET, Kaspersky Securelist, Check Point Research, Trend Micro, Deep Instinct, and Sekoia. Reports were drawn from two established aggregators: APTnotes, which indexes threat intelligence publications up to 2024, and Malpedia, from which additional reports were collected to cover the 2025-2026 period not represented in APTnotes [12], [13]. Reports range from 4,466 to 24,348 tokens in length, with a median of approximately 8,532 tokens. The corpus contains a total of 1,284 manually annotated ground truth entity instances across all 21 reports and 15 entity types.

The entity taxonomy spans three functional groups. Threat actor attributes comprise threat_actor_name and threat_actor_country. Campaign context entities include target_country, target_sector, malware_name, goal, tool, and vulnerability. Technical indicators of compromise cover IP, domain, URL, email, sha256, sha1, and md5. Together, these 15 types reflect the full informational structure encountered in real-world CTI publications, from strategic attribution to low-level network observables.

Entity annotation was performed by two annotators working at the Computer Emergency Response Team, with a focus on CTI, each independently processing the full text of every CTI report and recording entity spans across all 15 entity types. To assess the reliability of the entity definitions, agreement between the two independent passes was measured before reconciliation as a pairwise entity-level F1, in which a span counts as agreed only when both its boundary and its entity type match. The annotators achieved a pairwise entity-level F1 of 0.87, pooled

across all 15 entity types, indicating high agreement. Disagreement was concentrated almost entirely in the tool and malware categories, consistent with the boundary ambiguity between these types discussed in Section V, while the remaining thirteen categories showed near-complete agreement. Following each independent pass, the two annotation sets were compared document by document. Spans on which both annotators agreed were accepted into the ground truth directly. Divergences, including differences in entity boundary, entity type assignment, and decisions to include or exclude ambiguous candidate spans, were resolved through discussion until consensus was reached. Where recurring disagreements revealed an underspecified boundary condition in the annotation guidelines (most frequently at the tool and malware entities), an explicit clarification rule was formulated, documented, and applied retroactively across all affected instances in the corpus. The resulting ground truth, therefore, reflects a jointly validated, consensus-based interpretation of each entity instance rather than the judgment of a single annotator, following the dual-annotator reconciliation practice established in comparable CTI corpus construction efforts [8].

Entity annotation was performed without imposing a fixed schema at the outset. Instead, annotators recorded entity types as they naturally emerged from the reports, following an inductive approach. The 15 entity types reported in this study represent the full set of categories identified through this process – no entity type present in the source documents was excluded a priori. This approach ensures the taxonomy reflects the actual informational structure of real CTI publications rather than a theoretically predetermined set.

During the first iterations, models performed extremely poorly on attack goal and target sector entities, which is explained by values varying substantially in the source document language. A single concept might appear as "national government", "government entity", or "public sector," depending on the author and publication style. Similarly, goal descriptions ranged from "data theft" to "espionage" to "collection of intelligence". To ensure consistent extraction and enable fair evaluation, it was decided to provide models with a controlled vocabulary for the targeted sector, covering 18 types and aligned to the STIX. STIX language provides a standardized format for representing and sharing CTI to ensure interoperability across security tools and organizations [14]. On the other hand, with regard to common attack goals observed in the reports, a pool of 5 values was provided. Without this constraint, macro F1 on these categories would be artificially depressed by surface-form disagreements between model outputs and ground semantically equivalent truth annotations.

In addition to that, in early evaluation iterations, all three models exhibited a systematic pattern of returning natural-language vulnerability descriptions rather than CVE identifiers when asked to extract the vulnerability entity type. For example, when a report referenced CVE-2026-21509, models would return "Microsoft Office Security Feature Bypass Vulnerability" rather than the CVE identifier string. This occurred despite the entity type being named vulnerability. The prompt was revised to explicitly require the CVE-YYYY-NNNNN format and to instruct models to return the identifier string only, ignoring descriptions without an associated CVE code. The final evaluations show CVE F1 = 1.000 across all three models, confirming full resolution of this issue.

B. Models

Three models are evaluated in this study. Claude Sonnet 4.6 (claude-sonnet-4-6), GPT-5.4 (gpt-5.4), and LLaMA 4 Scout (meta-llama/llama-4-scout-17b-16e-instruct). All three models were queried with temperature = 0 and top-p = 0 to ensure deterministic, reproducible outputs.

C. Prompt Design and Evolution

Two distinct types of prompts were used during the evaluation. In a zero-shot setting, the model receives only a task description and the target document, with no examples of completed extraction Fig. 1. The model must rely entirely on its pretrained knowledge and the instructions provided in the prompt to produce the output. Zero-shot prompting is the simpler baseline: it requires no curated examples and makes the fewest assumptions about what the model has seen during training.

```
    You are a cybersecurity threat intelligence analyst. Your task is to
extract named entities from the CTI report provided below and return only
a valid JSON object that strictly conforms to the predefined schema. For
any entity types not present in the document, return empty arrays ([]).
Do not include any text, explanation, or formatting outside the JSON
output. Ensure that all occurrences of each entity type are extracted
across the entire document – do not limit extraction to the first in-
stance. In particular, for hash-related fields, include every identified
value.

    . . .
```

Fig. 1. Zero-shot prompt.

In a few-shot setting, the prompt is augmented with several examples – pairs of input text and expected output – that demonstrate the desired extraction behavior before the model encounters the target document Fig. 2. The rationale is that concrete examples communicate extraction conventions more reliably than abstract definitions alone: a model that sees a correctly annotated excerpt is better positioned to replicate that annotation style than one given only a textual description of what each entity type means.

```
    APT28, also known as Fancy Bear, is a Russian espionage group that
targeted German government agencies and French energy companies. The
campaign deployed X-Agent malware and leveraged CVE-2021-34527 (Print-
Nightmare). Mimikatz was used for credential dumping. C2 infrastructure:
185.220.101[.]45, update[.]evil-domain[.]com

    Output: { "threat_actor_name": ["APT28"], "threat_actor_country":
["russia"], "target_country": ["germany", "france"], "target_sector":
["government", "energy"], "malware_name": ["x-agent"], "goal": ["Espio-
nage"], "tool": ["mimikatz"], "ip": ["185.220.101.45"], "domain": ["up-
date.evil-domain.com"], "url": [], "email": [], "sha256": [], "sha1":
[], "md5": [], "vulnerability": ["CVE-2021-34527"]
```

Fig. 2. Example given with a few shots.

In this study, the few-shot condition prepends two fully annotated CTI report excerpts to the user prompt before the target document is presented. These exemplars were drawn from two published reports deliberately excluded from the 21-report evaluation corpus to prevent data leakage. The two excerpts were selected jointly such that, together, they provide at least one positive instance of each of the 15 entity types, ensuring the model has a concrete reference for every category it is expected to extract. Several entity types are represented by multiple instances within a single excerpt, so the exemplars demonstrate not only

the presence of each category but also the extraction of multiple entities of the same type. The use of real, annotated excerpts rather than synthetically constructed examples is a methodological requirement, since synthetic examples risk being unnaturalistic and may not reflect the surface-form variation that characterizes actual CTI publications.

The choice of two exemplars reflects a deliberate balance between coverage and prompt length. Two excerpts are sufficient to instantiate every one of the 15 entity types, with several types already represented by multiple instances, so adding further exemplars would not improve entity-type coverage. Prompt length, however, is a binding constraint, as reports in the evaluation corpus reach 24,348 tokens, and Section IV documents measurable extraction degradation on the longest reports under the existing prompt length, consistent with attention dilution over long contexts. Increasing the exemplar count would lengthen every prompt and amplify this effect across the corpus, not only on outlier reports. The two-exemplar configuration, therefore, represents the smallest prompt that achieves full entity-type coverage, which is the relevant design constraint for evaluation across reports of variable length.

The initial prompts defined each entity type with a brief natural-language description and requested a structured JavaScript Object Notation (JSON) response. Hash entities were described generically as "cryptographic file hash" without a format specification. Post-hoc analysis of initial results revealed that GPT and LLaMA systematically confused SHA-1 (40 hex characters) with SHA-256 (64 hex characters), treating all long hex strings as SHA-256 and returning near-zero SHA-1 recall.

D. Evaluation Metrics

All extractions are evaluated against ground truth annotations using exact string matching after lowercasing and whitespace normalization.

Precision – measures the proportion of extracted entities that are correct:

$$P = TP / (TP + FP), \quad (1)$$

where, TP is the number of correctly extracted entity instances, and FP is the number of extracted entities absent from the ground truth.

Recall – measures the proportion of ground truth entities successfully extracted:

$$R = TP / (TP + FN), \quad (2)$$

where, FN is the number of ground truth entities the model failed to extract.

F1 Score – providing a single balanced measure when both missed and spurious extractions carry cost.

$$F1 = 2PR / (P + R), \quad (3)$$

Micro F1 – aggregates TP, FP, and FN globally across all entity instances before computing Eq. (1), (2), and (3), weighting each entity occurrence equally regardless of type:

$$F1(micro) = 2TP / (2TP + FP + FN), \quad (4)$$

Macro F1 – the unweighted mean of per-entity-type F1 scores, treating each of the N = 15 entity types equally, regardless of its frequency in the corpus:

$$F1(macro) = (1/N) \times \Sigma F1_i, \quad (5)$$

where, $F1_i$ is the F1 score for entity type i and N = 15.

Hallucination Rate – the hallucination rate quantifies the proportion of extracted entities whose surface form does not appear in the source document:

$$HR = N_a / N_e, \quad (6)$$

where, N_a is the number of extractions whose surface form does not appear in the source document and N_e is the total number of extractions.

IV. Experimental Results

A. Overall Model Performance

Tables I and II present the macro and micro performance of all models under both conditions using the final prompt. Fig. 3 visualizes micro F1 scores.

TABLE I. Macro Performance

Model & Condition	Macro P	Macro R	Macro F1
Claude 4.6 ZS[a]	0.516	0.533	0.514
Claude 4.6 FS[b]	0.520	0.524	0.510
GPT-5.4 ZS	0.531	0.527	0.523
GPT-5.4 FS	0.530	0.498	0.503
LLaMA 4 Scout ZS	0.459	0.474	0.453
LLaMA 4 Scout FS	0.422	0.404	0.415

[a] Zero-Shot

[b] Few-Shot

TABLE II. Micro Performance

Model & Condition	Micro P	Micro R	Micro F1
Claude 4.6 ZS	0.782	0.890	0.832
Claude 4.6 FS	0.802	0.875	0.837
GPT-5.4 ZS	0.799	0.866	0.831
GPT-5.4 FS	0.815	0.792	0.803
LLaMA 4 Scout ZS	0.700	0.754	0.726
LLaMA 4 Scout FS	0.746	0.548	0.632

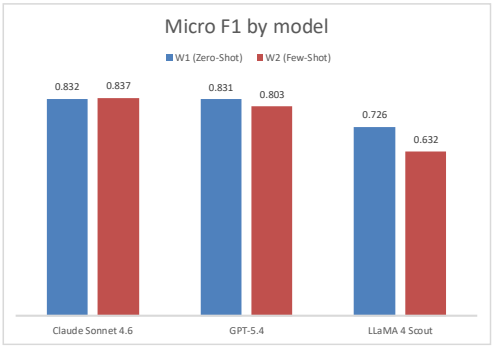

Fig. 3. Micro F1 by model and condition.

Claude Sonnet 4.6 and GPT-5.4 are closely matched under zero-shot, achieving micro F1 of 0.832 and 0.831, respectively. GPT-5.4 holds a slight advantage in macro F1 (0.523 vs 0.514), reflecting marginally better performance on low-frequency entity types. Claude achieves a higher micro recall (0.890 vs 0.866), suggesting a preference for breadth of extraction over precision. LLaMA 4 Scout trails by a substantial margin (micro F1 = 0.726), driven primarily by its poor tool entity performance and aggressive over-extraction in certain categories.

The few-shot condition produces mixed effects across all three models. Claude improves in micro F1 (+0.005) while reducing hallucination; GPT-5.4 declines (-0.028), driven primarily by the DynoWiper cardinality anchoring failure discussed in *subsection D*; LLaMA 4 Scout also declines substantially (-0.094), consistent with a pattern of few-shot conditioning providing marginal or neutral benefit over strong zero-shot baselines.

B. Per-Entity-Type Performance

Tables III and IV present F1 scores disaggregated by entity type for all three models under both the zero-shot and few-shot conditions using the final version of the prompt. Performance across the 15 entity types is highly non-uniform, spanning a range from perfect extraction to near-failure within the same model and the same report. First, entity type is a stronger predictor of extraction difficulty than model identity – the rank ordering of entity types by F1 is largely preserved across all three models, suggesting that the challenge lies in the entity itself rather than in model-specific capabilities. Second, the few-shot condition does not produce uniform improvement; its effect is entity-dependent, providing measurable benefit on some categories while introducing regression on others, most notably malware name for GPT-5.4 and URL for LLaMA 4 Scout. Third, the two entity categories with the most complex annotation boundaries – tool and target sector – are consistently the weakest across all models and both conditions, confirming that prompt-based extraction struggles most where the distinction between a positive and negative instance requires operational domain knowledge that cannot be fully encoded in a natural-language definition.

The following discussion groups entity types into three performance tiers and examines the underlying drivers of each.

1) High-performing categories: Vulnerability identifier extraction achieves F1 = 1.000 across all three models, a result of both the highly distinctive CVE identifier format and the explicit format constraint introduced in response to early description-extraction failures. IP address extraction is similarly reliable (0.927–0.983 F1), as dotted-decimal notation is unambiguous. Threat actor country (0.950) and SHA-1/SHA-256 (0.807–0.964) are also well-extracted following the revised prompt.

2) Moderate-performing categories: Malware name extraction achieves F1 of 0.753-0.857. Claude shows high recall (0.926) but lower precision (0.708), suggesting a tendency to include related tool names or sub-components alongside the primary malware families. GPT-5.4 is better balanced (P=0.833, R=0.882). Domain and URL extraction show consistent moderate performance (~0.62-0.76 F1), with false positives arising from vendor website links, CDN hosts, and report-source URLs that appear in CTI documents alongside malicious infrastructure.

TABLE III. PER-ENTITY F1 RESULTS USING ZERO-SHOT APPROACH

	Claude Sonnet 4.6	GPT-5.4	LLaMA 4 Scout
Entity Type	**Zero-Shot**		
Threat Actor Name	0.923	0.923	0.881
Threat Actor Country	0.974	1.000	0.942
Target Country	0.904	0.828	0.831
Target Sector	0.615	0.538	0.438
Malware Name	0.800	0.619	0.697
Goal	0.780	0.810	0.706
Tool	0.611	0.700	0.468
IP Address	0.982	0.956	0.903
Domain	0.690	0.703	0.762
URL	0.800	0.735	0.581
Email	0.895	0.919	0.862
SHA-256	0.948	0.917	0.840
SHA-1	0.931	0.929	0.771
MD5	0.806	0.707	0.476
CVE	1.000	1.000	1.000

TABLE IV. PER-ENTITY F1 RESULTS USING FEW-SHOT APPROACH

	Claude Sonnet 4.6	GPT-5.4	LLaMA 4 Scout
Entity Type	**Few-Shot**		
Threat Actor Name	0.872	0.923	0.857
Threat Actor Country	0.974	0.950	0.950
Target Country	0.931	0.868	0.829
Target Sector	0.642	0.725	0.468
Malware Name	0.803	0.857	0.753
Goal	0.810	0.829	0.698
Tool	0.582	0.673	0.416
IP Address	0.983	0.956	0.927
Domain	0.688	0.740	0.733
URL	0.760	0.731	0.624
Email	0.850	0.842	0.786
SHA-256	0.947	0.912	0.852
SHA-1	0.964	0.947	0.807
MD5	0.800	0.745	0.452
CVE	1.000	1.000	1.000

Beyond C2 and phishing infrastructure, CTI reports routinely reference legitimate third-party services as part of their technical narrative, e.g., links to malware samples hosted on VirusTotal or code repositories on GitHub. During corpus annotation, such references were deliberately excluded from the ground truth for domain and URL entity types. Similarly, domain names associated with legitimate content delivery networks or cloud platforms, even when referenced in the context of an attack, were omitted when the domain itself is not malicious infrastructure. The rationale for this exclusion is operational: automated ingestion of these indicators into Intrusion Detection and Prevention Systems (IDS/IPS) or firewall blocklists would block access to widely-used legitimate services, causing service disruption disproportionate to any security benefit. Only the full URL path pointing to a malicious file or payload was retained as a ground truth instance in such cases, as this specificity limits the risk of adverse blocking effects while preserving the actionable indicator.

3) Low-performing categories: The tool entity type returns the lowest F1 scores across all models: Claude 0.582, GPT-5.4 0.673, LLaMA 0.416. The LLaMA precision of 0.288 – meaning fewer than 3 in 10 extracted "tools" are correct makes this the single most problematic category in the evaluation. Further comparison of extracted and ground truth values reveals six distinct root causes:

a) Malware/tool boundary confusion: LLaMA repeatedly extracted malware families as tools. In the OilRig report, LLaMA extracted “danbot”, “mango”, “marlin”, “milan”, “oilcheck”, “oilforcegtx”, “shark”, and “solar” as all malware families in the OilRig toolkit as tool entities, while the ground truth annotates these as “malware_name”. Similarly, in the Gamaredon report, LLaMA extracted “GammaLoad” and “GammaStealer” as tools. Claude 4.6 and GPT-5.4 showed this error rarely.

b) System utility over-extraction: All three models extracted operating system utilities and scripting engines – PowerShell, mshta.exe, vbscript, and WinRAR as attack tools. The annotation schema restricts the tool entity type to threat-actor-specific offensive utilities (e.g., Cobalt Strike, Mimikatz, Ligolo) and excludes general system components, even when those components appear in an attack chain. This boundary is conceptually grounded in the notion of Living Off the Land Binaries (LOLBins): legitimate, pre-installed, and typically digitally signed executables that adversaries abuse because their use blends with normal system activity and evades signature-based detection. While LOLBins are tactically significant for malware analysts performing dynamic analysis, they provide limited value to CTI analysts performing strategic or operational threat intelligence work – their appearance in a campaign report indicates technique, not infrastructure or tooling unique to the threat actor. Since LOLBins are exploited across virtually every threat actor and campaign, their extraction does not contribute to attribution, actor profiling, or indicator-based detection. Accordingly, they were excluded from the tool's ground truth, and model extractions of such binaries were scored as false positives.

c) Name normalization failures: This issue was observed on all models. Exact string matching penalizes near-correct extractions. "cobalt strike beacon" was extracted instead of "cobalt strike"; "atera agent" instead of "atera"; "connectwise" instead of "connectwise remote access"; "fast reverse proxy server" instead of "fast reverse proxy server (frps)". These represent correct entity identification with imprecise boundary delineation – a known limitation of exact-match evaluation in NER.

d) Analyst-related software extraction: LLaMA extracted legitimate security research tools such as “Wireshark”, “procmon”, “fiddler”, and “ollydbg” and assigned them to the tools category, despite them being mentioned in attack investigation sections. Claude 4.6 and GPT-5.4 avoided this error.

e) Target sector – target sector achieves moderate F1 (0.468–0.725). Despite the controlled vocabulary provided in the prompt, models vary in the granularity at which they extract sector entities. LLaMA tends to over-enumerate sectors when a report mentions multiple victim organizations (high FP), while Claude and GPT are more conservative. A residual source of false negatives is sectors implied but not explicitly named in the report text.

f) MD5 hashes – MD5 recall is the primary challenge (0.384-0.674). MD5 hashes frequently appear alongside SHA-256 values, and models tend to prioritize the longer hash. Additionally, 32-character hex strings appear in non-hash contexts and, despite a strict definition in the last prompt, while reducing false positives, also excluded some true MD5 values.

C. Zero-Shot Vs. Few-Shot Performance and Hallucination

LLaMA 4 Scout shows a substantial decline under the few-shot conditions, with micro F1 dropping from 0.726 to 0.632 (-0.094) and macro F1 from 0.453 to 0.415 (-0.038). This pattern is broadly consistent with GPT-5.4, which also declines under a few-shot prompt. The decline is distributed across a range of entity types: on reports with high entity density, the few-shot exemplars appear to constrain extraction breadth without meaningfully improving precision, yielding a net negative trade-off. Hallucination rate decreases from 0.208 to 0.162, indicating that exemplars do successfully suppress ungrounded extractions – but the precision gain is offset by increased false negatives.

LLaMA shows greater sensitivity to few-shot cardinality anchoring than the commercial models. On reports with long entity lists, the model tends to match the exemplar entity count more rigidly, missing entities beyond the first few instances. This pattern is less pronounced but qualitatively similar to the GPT-5.4 DynoWiper failure described in the following section.

The most consistent effect of few-shot prompting is a reduction in hallucination rate. Across both prompt versions and all models, the few-shot approach reduces hallucination by 0.028-0.037, confirming that exemplars constrain the output space and discourage ungrounded entity generation. Hallucination rates by model and workflow are depicted in Fig. 4.

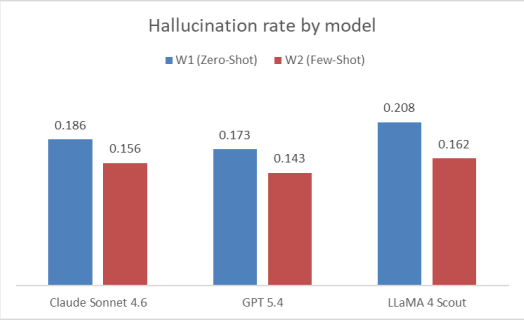

Fig. 4. Hallucination rate by model.

However, hallucination reduction does not translate uniformly to F1 improvement. For Claude, reduced false positives are partially offset by increased false negatives, yielding a net micro F1 gain of +0.005. For GPT-5.4, the hallucination reduction is overwhelmed by the catastrophic DynoWiper failure (*subsection D*), depressing corpus-wide micro F1 by -0.028.

Claude produced identical extraction results between using zero-shot and few-shot prompts on five reports: APT28 CVE-2026-21509, Gamaredon, ITG05/Headlace, BugSleep, and LIGHTSHOW Pt. 1. This circumstance can be defined as "ceiling stability". On reports with high entity density, highly distinctive entity formats, or strong structural regularity, zero-shot extraction already achieves maximum performance, and few-shot exemplars provide no additional signal. This finding suggests that document-adaptive prompting – applying few-shot conditioning selectively to reports where zero-shot confidence is low may be more efficient than uniform few-shot conditioning.

D. Case of Anchoring Bias and GPT-5.4

The DynoWiper report by ESET contains 22 malware entities in its ground truth: a catalogue of the Sandworm threat actor's full wiper toolkit, including "ArguePatch", "AwfulShred", "CaddyWiper", "HermeticWiper", "Industroyer2", "NotPetya", "Prestige", and others. Under zero-shot, GPT-5.4 extracts 22 malware names (micro F1 = 0.921). Under few-shot, GPT-5.4 extracts exactly one malware name: "dynowiper" – the tool named in the report title. All 21 remaining malware names become false negatives. For this reason, the micro F1 for this report dropped to 0.516.

We attribute this to quantity anchoring bias: the two few-shot exemplars each contained several malware entities. When the target report contains 22, GPT-5.4 anchors on exemplar cardinality and treats the document's primary malware (the title-referenced tool) as the sole extraction target, discarding the remainder as background. The same pattern partially affects target_sector (7 ground truth entities resulted in 1 extracted) and target_country (reduced from 2 to 1).

Practical implication here is that few-shot exemplars for CTI NER must be matched not only in entity type coverage but in entity count distribution. An exemplar with 2 malware names is actively misleading when applied to reports cataloguing more than 20.

E. Report Complexity and Performance

Small (less than 6k tokens) and medium (6-12k tokens) reports achieve comparable average micro F1 (0.773 and 0.787, respectively). Large reports (more than 12k tokens) drop to 0.641, driven by the four longest reports: APT28 Phantom Net Voxel (24,348 tokens), LIGHTSHOW Pt. 1 (17,956), OilRig (15,928), and Sandworm-related report (19,693). These reports show disproportionately high false negative rates, consistent with attention dilution over long contexts. Table V presents the full per-report breakdown.

TABLE V. PER-REPORT MICRO F1 BY MODEL UNDER ZERO-SHOT CONDITION

Report	Input Tokens	Claude F1	GPT F1	LLaMA F1
1	19,693	0.777	0.792	0.791
2	8,087	0.719	0.742	0.652
3	24,348	0.651	0.458	0.360
4	10,595	0.951	0.944	0.715
5	5,448	0.842	0.914	0.632
6	8,763	0.848	0.921	0.795
7	7,756	0.738	0.800	0.531
8	6,661	0.462	0.467	0.424
9	5,921	0.552	0.593	0.727
10	11,364	0.877	0.915	0.884
11	10,232	0.901	0.891	0.727
12	4,466	0.818	0.806	0.746
13	7,130	0.636	0.538	0.500
14	7,320	0.951	0.918	0.812
15	8,532	0.979	0.978	0.936
16	15,928	0.819	0.906	0.635
17	10,012	0.837	0.878	0.900
18	17,956	0.593	0.589	0.321
19	5,960	0.800	0.769	0.588
20	10,874	0.905	0.878	0.794
21	5,425	0.930	0.930	0.948

V. DISCUSSION

A. Overall Model Comparison

At the aggregate level, Claude Sonnet 4.6 and GPT-5.4 are interchangeable for zero-shot CTI NER within measurement precision. The practical choice depends on operational priorities. Claude's higher recall (0.890 vs 0.866) is preferable when missing indicators are more costly than reviewing false positives, which is the typical operational stance for threat intelligence ingestion. GPT-5.4's marginally better macro F1 (0.523 vs 0.514) may reflect better handling of low-frequency entity types such as goal and tool.

LLaMA 4 Scout trails the commercial models at micro F1 = 0.726 zero-shot and 0.632 few-shot. The model shows particular weakness in tool entity precision (0.288) and target sector recall (0.431). As an open-weight model, it remains attractive for data-residency-constrained or air-gapped environments. The performance gap relative to Claude and GPT is substantial but likely

closeable with domain-specific fine-tuning on a CTI NER corpus.

B. Prompt Specificity as a High-Leverage Intervention

The single most impactful change across the study (discussed in the Methodology section), adding hash length constraints, produced larger improvements for the affected entity types than the entire few-shot conditioning effort. This establishes a design principle: for entity types with formally constrained structures (hashes, CVE identifiers, IP addresses), precise format specification in the system prompt is a prerequisite, not a refinement. The same principle motivated the controlled vocabulary intervention for sector and goal, and the explicit CVE format requirement for vulnerability.

However, the MD5 case illustrates the inherent precision/recall trade-off in format-constrained extraction. The revised prompt required strictly lowercase, 32-character hexadecimal strings, which suppressed false positives from non-hash hex sequences but simultaneously caused genuine MD5 values to be missed, subsequently reducing recall.

C. Comparison with Fine-Tuned Baselines

All three models in this study operate without task-specific training, so situating their performance against fine-tuned supervised systems is necessary to interpret the results. No controlled comparison is possible here because the published baselines differ from the present study and from one another, along several axes that each affect the reported score.

TABLE VI. COMPARISON WITH FINE-TUNED CTI NER

Approach	Corpus, taxonomy	Reported F1
This work	21 CTI reports, 15 entities	0.832, 0.831, 0.726 (micro)
[8]	4 harmonized corpora, 21 entities	0.723 to 0.736 (micro)
[2]	DNRTI dataset, 13 entities	0.875 to 0.883 (macro)
[4]	DNRTI dataset, 13 entities	0.900
[5]	7 entity types	0.811
[6]	5 entity types	0.945
[3]	Vulnerability-related entities only	0.968
[7]	16 IoC-related entities	0.975

The single comparison conducted on a comparable footing is with [7], the only baseline evaluated on a broad, cross-source, STIX-aligned taxonomy of 21 entity types against the 15 used in the present study, with entity-level micro F1. On that benchmark, the best fine-tuned transformer-CRF reaches micro F1 0.736, with the remaining encoders clustered between 0.723 and 0.730. The zero-shot micro F1 obtained in the present study, 0.832 for Claude Sonnet 4.6 and 0.831 for GPT-5.4, falls at or above the fine-tuned range on the most similar task, and does so without any task-specific training. The strongest configuration measured, Claude Sonnet 4.6 under few-shot prompting at 0.837, exceeds the fine-tuned range by a clear margin. Even the strongest single-source result reported in [7], an in-domain DNRTI model at 0.871, is achieved on 13 entity types with the model trained and tested on the same corpus and annotation style, a substantially easier setting than the cross-source, fifteen-type extraction evaluated here.

The remaining approaches in Table VI are not directly comparable, and their higher figures do not contradict this finding, because each is produced under conditions more favourable to a fine-tuned model than the present evaluation. References [3] and [5] report 0.875 to 0.883 and 0.900, respectively, but on an in-domain split of the 13-type DNRTI corpus, and the figures in [2] are macro-averaged rather than micro-averaged and so are not directly comparable to the micro F1 reported here. CyBERT reports 0.879 on its own seven-type dataset [5], and SecureBERT 2.0 reaches 0.945 over only five broad categories [6], both on a single in-domain corpus. These results reflect specialization to one annotation style and a narrower taxonomy, not extraction across heterogeneous sources.

The two highest scores in Table VI rest on the least comparable basis and are informative only for a subset of the taxonomy. Reference [3] reports 0.968 on structured vulnerability-description fields such as operating system, vendor, and version rather than CTI-campaign entities, and the models in [7] reach 0.975 over only sixteen indicator-of-compromise types, produced by regular-expression parsers rather than human annotation. Both are relevant to the technical-indicator categories alone, where the present models already perform near-perfectly, and not to overall extraction.

D. Few-Shot Prompting: Hallucination Vs. Cardinality Risk

Few-shot prompting reliably reduces hallucination but introduces cardinality anchoring risk when exemplar entity counts are not representative of the target corpus distribution. The DynoWiper failure demonstrates that this risk is not theoretical – it produces a 73% collapse in micro F1 on an affected report. Practitioners constructing few-shot prompts for CTI NER should 1) sample exemplars to reflect the full entity count distribution per type, 2) include at least one high-cardinality example for any entity type that can appear as a large list, and 3) evaluate the few-shot condition per-report rather than relying solely on corpus-level averages.

Taken together, these findings suggest that the principal bottleneck in prompt-based CTI NER is not model capability in the general sense, but rather the structural properties of the entity types being extracted and the degree to which the prompt communicates their formal constraints. Entities with machine-readable formats (CVE identifiers, IP addresses, cryptographic hashes) are amenable to near-perfect extraction through explicit specification alone. Entities that require operational domain knowledge to disambiguate (tool versus malware, implied sector versus named sector) resist improvement through either few-shot conditioning or format constraints, pointing toward fine-tuning on annotated CTI corpora as the necessary next step. The consistent hierarchy of difficulty across three architecturally distinct models reinforces that this is a domain-level challenge, not a model-level one.

E. Limitations

Several limitations of this study should be acknowledged. The evaluation corpus comprises 21 manually annotated reports. This is a modest scale, and the per-entity-type estimates for low-frequency categories such as goal and tool are therefore subject

to wider variance than the corpus-level figures. The corpus was constructed manually because, to the best of the author's knowledge, no existing public CTI NER dataset covers the 15-type taxonomy evaluated here. Prior corpora target narrower or differently structured schemas, such as the 13 types of DNRTI or the 5 of CyNER, and none of these combine threat-actor attributes, campaign context, and the indicator-of-compromise granularity used in this study within a single annotation scheme. Building a corpus aligned to this taxonomy, therefore, required manual annotation, which bounded its size. A larger corpus spanning more report authors, threat-actor origins, and publication periods would yield more stable per-type estimates and reduce the influence of individual outlier reports on the aggregate metrics, and remains a direction for future work.

All extractions are scored by exact string matching after lowercasing and whitespace normalization, which is a conservative criterion. As documented in Section V, extractions that identify the correct entity with an imprecise boundary, such as "cobalt strike beacon" for "cobalt strike", "atera agent" for "atera", or "fast reverse proxy server" for "fast reverse proxy server (frps)", are counted as both a false positive and a false negative under exact matching, even though they would be operationally useful to an analyst. The reported scores, therefore, represent a lower bound on extraction quality, and a partial-match or soft-F1 metric that credits boundary-imperfect but semantically correct extractions would yield higher figures, particularly for the tool and malware categories where boundary ambiguity is most frequent. Reporting such a metric alongside an exact match is a natural refinement for future evaluation.

A further consideration is potential contamination of the evaluation by data seen during pre-training. Because part of the corpus was published in 2025 and 2026, after the evaluated models' pre-training cutoffs, full memorization is unlikely for those documents, though the underlying campaigns and indicators may still appear in pre-training corpora. This effect is conservative for the baseline comparison in Section V: a prompt-based model may benefit from prior exposure to a report, whereas a fine-tuned encoder trained on a disjoint corpus cannot, so any contamination present would understate rather than inflate the gap. A per-report F1 stratification by publication date is left to future work.

Beyond these immediate constraints, the scope of the evaluation is intentionally limited to named entity identification. Extending the framework to STIX 2.1 relationship extraction and moving beyond the identification of individual entities to the structured representation of relationships between them, such as threat actor-to-malware attribution or campaign-to-infrastructure associations, represents a natural and high-value next step toward fully automated CTI report-to-STIX conversion pipelines.

VI. Conclusion

This study evaluated Claude Sonnet 4.6, GPT-5.4, and LLaMA 4 Scout on CTI named entity recognition across 21 reports, 15 entity types, two prompting conditions, and two prompt versions. Commercial LLMs achieve strong zero-shot CTI NER performance (~0.83 micro F1) without task-specific fine-tuning. A single prompt revision (adding explicit character-length constraints for hash entities) produced the largest F1 improvements in the study, raising SHA-1 F1 by +0.442 (GPT-5.4) and +0.373 (LLaMA), demonstrating the outsized leverage of format specification for technically structured entities.

Few-shot prompting consistently reduces hallucination rates across all models but introduces cardinality anchoring risk, demonstrated most severely by the GPT-5.4's DynoWiper failure. All three models show neutral-to-negative few-shot F1 deltas, with hallucination reduction as the primary measurable benefit of the few-shot condition. The tool entity type is the most challenging across all models due to systematic malware/tool boundary ambiguity, system utility over-extraction, and name normalization failures that exact-match evaluation does not credit.

Collectively, these findings provide both a practical guide for deploying LLM-based CTI NER and an empirical basis for future research in prompt engineering, exemplar selection strategy, and evaluation methodology for the CTI domain.

References

[1] P. Alaeifar, S. Pal, Z. Jadidi, M. Hussain, and E. Foo, "Current approaches and future directions for Cyber Threat Intelligence sharing: A survey," Journal of Information Security and Applications, vol. 83, p. 103786, Jun. 2024, doi: 10.1016/j.jisa.2024.103786.

[2] P. Evangelatos et al., "Named Entity Recognition in Cyber Threat Intelligence Using Transformer-based Models," in 2021 IEEE International Conference on Cyber Security and Resilience (CSR), Rhodes, Greece: IEEE, Jul. 2021, pp. 348–353. doi: 10.1109/CSR51186.2021.9527981.

[3] P. Ranade, A. Piplai, A. Joshi, and T. Finin, "CyBERT: Contextualized Embeddings for the Cybersecurity Domain," in 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA: IEEE, Dec. 2021, pp. 3334–3342. doi: 10.1109/BigData52589.2021.9671824.

[4] S. Zhou, J. Liu, X. Zhong, and W. Zhao, "Named Entity Recognition Using BERT with Whole World Masking in Cybersecurity Domain," in 2021 IEEE 6th International Conference on Big Data Analytics (ICBDA), Xiamen, China: IEEE, Mar. 2021, pp. 316–320. doi: 10.1109/ICBDA51983.2021.9403180.

[5] S.-S. Chen, R.-H. Hwang, C.-Y. Sun, Y.-D. Lin, and T.-W. Pai, "Enhancing Cyber Threat Intelligence with Named Entity Recognition Using BERT-CRF," in GLOBECOM 2023 - 2023 IEEE Global Communications Conference, Kuala Lumpur, Malaysia: IEEE, Dec. 2023, pp. 7532–7537. doi: 10.1109/GLOBECOM54140.2023.10436853.

[6] E. Aghaei, S. Jain, P. Arun, and A. Sambamoorthy, "Securebert 2.0: Advanced Language Model for Cybersecurity Intelligence," in 2025 Annual Computer Security Applications Conference Workshops (ACSAC Workshops), Honolulu, HI, USA: IEEE, Dec. 2025, pp. 382–388. doi: 10.1109/ACSACW69556.2025.00071.

[7] H.-J. Chan, C.-Y. Hsu, C.-C. Chien, J.-J. Wu, and H.-L. Ku, "FeedRef2022: A Named Entity Recognition Dataset for Extracting Indicators of Compromise," in 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan: IEEE, Dec. 2022, pp. 2578–2584. doi: 10.1109/BigData55660.2022.10020985.

[8] Y. Ech-Chammakhy, A. Motii, A. Rabii, O. Azrara, and J. Chbili, "CyberNER: A Harmonized STIX Corpus for Cybersecurity Named Entity Recognition," in 2025 IEEE 24th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guiyang, China: IEEE, Nov. 2025, pp. 2190–2197. doi: 10.1109/Trustcom66490.2025.00255.

[9] S. Della Penna, R. Natella, V. Orbinato, L. Parracino, and L. Pianese, "CTI-HAL: A Human-Annotated Dataset for Cyber Threat Intelligence Analysis," in 2025 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Venice, Italy: IEEE, Jun. 2025, pp. 69–78. doi: 10.1109/EuroSPW67616.2025.00014.

[10] K. Guru, R. J. Moss, and M. J. Kochenderfer, "On Technique Identification and Threat-Actor Attribution using LLMs and Embedding Models," in 2025 International Conference on Cybersecurity and AI-Based Systems (Cyber-AI), Varna, Bulgaria: IEEE, Sep. 2025, pp. 332–339. doi: 10.1109/Cyber-AI66431.2025.11233715.

[11] H. Alturkistani, A. G. Jaafar, and S. Chuprat, "Automating Cyber Threat Intelligence Workflows with LLMs: Processing, Analyzing, and Defending in One Model," in 2025 3rd International Conference on Cyber Resilience (ICCR), Dubai, United Arab Emirates: IEEE, Jul. 2025, pp. 1–6. doi: 10.1109/ICCR67387.2025.11292095.

[12] K. Bandla, S. Castro "APTnotes," Source: [Online]. Available: https://github.com/aptnotes/

[13] D. Plohmann, M. Clauß, S. Enders, and E. Padilla, "Malpedia: A Collaborative Effort to Inventorize the Malware Landscape," The Journal on Cybercrime and Digital Investigations, vol. 3, no. 1, pp. 1–19, Dec. 2017, doi: 10.18464/cybin.v3i1.17.

[14] A. Papoutsis et al., "CTI-GEN: A Framework for Generating STIX 2.1 Compliant CTI Using Generative AI," in 2025 IEEE International Conference on Cyber Security and Resilience (CSR), Chania, Crete, Greece: IEEE, Aug. 2025, pp. 334–341. doi: 10.1109/CSR64739.2025.11130126.