UDC 004.89

A Formal Model for Constructing Sensitive Data Graphs from Cyber Reports using Large Language Models

Viktor Turskyi[1]

[1] *National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine*

Abstract

Unstructured cyber threat intelligence (CTI) reports present major challenges for systematic analysis, particularly when accuracy and reliability are critical. This paper introduces a formal, four-stage mathematical model for constructing canonical knowledge graphs from sensitive textual data. The model integrates the advanced extraction and reasoning capabilities of GPT-5 with deterministic rule-based inference and network analysis to bridge the "formalization gap" between probabilistic large language model (LLM) outputs and verifiable analytical structures. Using a corpus of 204 official CERT-UA incident reports as a test case, the methodology successfully normalized thousands of raw entities, identified central threat actors and high-value targets, and revealed distinct operational ecosystems within Ukraine's cyber threat landscape. Theoretically, the study contributes a replicable and mathematically defined framework for integrating next-generation LLMs into formalized knowledge graph pipelines. Practically, it provides a scalable and reliable tool for analysts in cybersecurity, national security, and related fields, enabling the transformation of unstructured reports into actionable intelligence.

Keywords: Large Language Models (LLM), Cyber Threat Intelligence (CTI), Sensitive Data Analysis, Network Analysis, Entity Resolution, CERT-UA.

1. Introduction

Large amounts of unstructured text in important fields like national security, law, and finance present a major challenge for automatically finding and managing sensitive information. The manual analysis of this data is often intractable, creating a demand for automated solutions to extract actionable intelligence, particularly in domains like Cyber Threat Intelligence (CTI) [1]. Large Language Models (LLMs) are powerful tools for understanding text, but they are not always reliable. Their outputs can be random and difficult to verify, which is a significant risk in areas where accuracy is essential. The main problem, known as the "formalization gap," is the lack of a structured method to turn the probabilistic outputs of LLMs into a reliable model that can be formally analyzed. Without this, it is hard to check, combine, or systematically study the information LLMs provide, which limits their use in critical applications.

To solve this problem, a four-stage mathematical model is proposed for the automated construction and analysis of a canonical knowledge graph [2]. This model formalizes the entire intelligence pipeline: (i) raw entity extraction from each document is performed using GPT-5 to identify entities and assign their categories and roles; (ii) a global, GPT-5 powered normalization stage is applied to resolve synonyms and create unique, canonical entity representations; (iii) a deterministic, rule-based engine is used to infer semantic relationships (e.g., attacks, is_attributed_to) between entities based on their attributes; and (iv) a formalized analysis is conducted using targeted graph metrics, such as filtered weighted degree and ego network profiling, to generate quantitative insights.

The efficacy of this model was validated using a complete corpus of 204 official cyber incident reports published by the Computer Emergency Response Team of Ukraine (CERT-UA) since the 2022 full-scale invasion. This real-world dataset, rich with sensitive and strategically important information, serves as a robust test case. The primary contribution of this work is the formal, hybrid model itself, which provides a replicable methodology for combining the advanced reasoning of GPT-5 with the

logical consistency of a rule-based system to create a high-fidelity knowledge graph. This paper details the mathematical framework, presents the empirical results and specific insights derived from the CERT-UA dataset, and discusses the broader implications of the methodology for sensitive data analysis.

2. Related Work

This section reviews the evolution of relevant research in three key areas: information extraction using language models, the methodologies for knowledge graph construction, and the automated analysis of Cyber Threat Intelligence (CTI).

2.1. The Evolution of Language Models for Information Extraction

Information Extraction has been significantly advanced by progress in neural network architectures. Early benchmarks in Named Entity Recognition (NER) were established by models such as BiLSTM-CRF, though these required large, manually labeled datasets for training [3]. The introduction of the Transformer architecture [4] and large-scale pre-trained models like BERT [5] marked a paradigm shift, enabling high performance through fine-tuning on smaller datasets.

The subsequent generation of auto-regressive models, such as the GPT series, introduced the concept of in-context learning, which reduced the need for fine-tuning altogether [6]. However, while models in the GPT-3 and GPT-4 class demonstrated impressive zero-shot capabilities, their application in high-stakes domains was often hampered by issues of factual consistency and "hallucinations" [7]. The development of GPT-5 represents the current state of the art, engineered for enhanced multi-step reasoning, higher factuality, and a more robust adherence to complex instructions. This study is among the first to leverage these next-generation capabilities to address the specific challenges of extracting sensitive, highly contextual information from specialized texts.

2.2. Knowledge Graph Construction from Unstructured Text

The construction of Knowledge Graphs (KGs) from text is a well-established research area, with early work focusing on Open Information Extraction (OpenIE) from web-scale corpora (Banko et al., 2007). These traditional methods typically relied on complex, multi-stage NLP pipelines. More recently, LLMs have been utilized to create end-to-end KG construction pipelines [8].

However, a persistent bottleneck in these LLM-based approaches has been the quality of entity resolution and canonicalization. Previous models often struggled to consistently group different aliases for the same entity, leading to fragmented and noisy graphs. The advanced reasoning and vast internal knowledge base of GPT-5 provide a new opportunity to overcome this challenge. Our methodology was designed to specifically test the hypothesis that a state-of-the-art model like GPT-5 can perform high-accuracy entity normalization as an integral and formalized step of the KG construction process.

2.3. Automated Analysis of Cyber Threat Intelligence

The CTI domain presents unique challenges due to its specialized vocabulary and the dynamic nature of threats. While standards like STIX/TAXII exist for structured intelligence, a majority of CTI is disseminated in unstructured reports. Prior research has successfully applied NLP to this area, for example, in the extraction of adversary Tactics, Techniques, and Procedures (TTPs) [9] and in the construction of attack graphs to model threat scenarios.

These foundational works, however, were constrained by the capabilities of earlier NLP technologies, often requiring extensive feature engineering or fine-tuning. They could extract explicit indicators but often struggled with inferring the nuanced, implicit relationships that define strategic context. The advent of GPT-5 provides an opportunity to revisit these challenges with a significantly more powerful analytical engine, capable of understanding deeper context from raw text.

2.4. Research Gap and Contribution

The existing literature demonstrates a clear trajectory towards more powerful language models for data analysis. However, there remains a gap in research that formally applies a next-generation model like GPT-5 within a structured, replicable, and mathematically defined framework for CTI analysis.

This paper addresses this gap directly. Our primary contributions are:

The first (to our knowledge) empirical application of GPT-5 within a formal model for constructing a canonical knowledge graph from sensitive CTI reports.

A demonstration that the advanced capabilities of GPT-5, particularly in zero-shot extraction and entity resolution, can significantly improve the fidelity and completeness of the resulting graph compared to previous approaches.

A robust, hybrid methodology that combines the power of GPT-5 with a deterministic rule-based engine, providing a benchmark for future research into the use of state-of-the-art LLMs for sensitive data analysis.

3. Formal Mathematical Model and Methodology

The proposed model formalizes a four-stage process for transforming a corpus of unstructured text documents into a unified, consistent, and structured knowledge graph. The model's objective is to ensure that nodes in the graph represent unique, real-world entities by resolving duplicates and synonyms that inevitably arise from the automated processing of text.

The process is initiated with stage 1 “Raw Entity Extraction”, where each document is parsed by a LLM to identify key entities and their contextual roles. In stage 2 “Canonical Normalization”, all extracted entities from the entire corpus are aggregated, and synonyms are resolved to create a unified, canonical set of nodes. Subsequently, in stage 3 “Graph Construction”, a deterministic, rule-based engine is used to infer the semantic relationships between co-occurring entities, resulting in the creation of a weighted, directed graph. The final stage, stage 4 “Graph Analysis”, is where the constructed graph is subjected to a suite of network science metrics to identify key structural patterns and quantify the importance of each entity. To demonstrate our model, a full implementation was developed and applied to the dataset [1].

3.1. Stage 0: Initial Definitions

- $D = \{d_1, d_2, \dots, d_n\}$a corpus of n text documents.
- $C = \{Country, HackerGroup, \dots\}$ a finite set of predefined entity **categories**.
- $P = \{Attacker, Target, Neutral\}$ a finite set of predefined entity **roles**.

3.2. Stage 1: Raw Entity Extraction

In this stage, each document is processed independently by an extractor function $\Phi_{extract}$, implemented using LLM. The extractor function, $\Phi_{extract}$, was implemented using the state-of-the-art GPT-5 large language model. This model was specifically chosen for the initial data extraction task due to several key advantages over previous generations of models. Firstly, GPT-5 exhibits superior zero-shot performance, allowing it to accurately adhere to the complex, multi-part schema (10 entity categories and 3 roles) presented in the prompt without any task-specific fine-tuning. Secondly, it has been engineered for higher factual accuracy and a reduced rate of "hallucination," which is a critical requirement when processing sensitive data. Finally, its advanced multilingual capabilities were essential for effectively parsing the source reports written in Ukrainian. For this study, the “gpt-5-2025-08-07” version was utilized via its official API.

To ensure consistency and accuracy, a detailed, structured prompt was designed. The LLM was instructed to act as an expert cybersecurity analyst and to return its findings in a strict JSON format. The prompt provided a closed set of possible entity categories (e.g., HackerGroup, Software, Government Body) and contextual roles (Attacker, Target, Neutral), which the model was required to assign to each identified entity.

$\Phi_{extract}: d \to E'$, where $d \in D$, and E' is the set of "raw" entities extracted from document d.

Each raw entity $e' \in E'$ is a tuple of three elements:

$$e' = (s, c, p)$$

where:

- $s \in$ Strings is the textual representation (surface form) of the entity as it appeared in the document (e.g., "Fancy Bear", "SBU").
- $c \in C$ is the entity's category.
- $p \in P$ is the entity's role within the context of the given document.

The result of this stage is a collection of sets of raw entities for each document: $\{E'_1, E'_2, \dots, E'_n\}$, where $E'_i = \Phi_{extract}(d_i)$. These entities are considered "raw" because they

[1]The source code for the model and the analysis pipeline is publicly available at: https://github.com/koorchik/llm-analysis-of-text-data

have not yet been de-duplicated or canonicalized across the entire corpus, a process which is addressed in the next stage of the model.

3.3. Stage 2: Global Aggregation and Canonical Normalization

This stage is critical for ensuring data consistency across the entire corpus.

1. Aggregation**:** First, all raw entities from all documents are aggregated into a single global set, V_{raw}

$V_{raw} = E'_1 \cup E'_2 \cup \ldots \cup E'_n$

V_{raw} contains all unique tuples (s, c, p) found in the corpus.

2. LLM-based Normalization**:** An entity resolution function, Ψ_{norm}, is introduced. This function uses an LLM to identify and group synonymous entity names. It takes the set of all unique textual names,

$S_{names} = \{s \mid \exists c, p: (s, c, p) \in V_{raw}\}$

as input.

$\Psi_{norm}: S_{names} \to \{S_1, S_2, \ldots, S_m\}$

where $\{S_j\}_{j=1}^{m}$ is a partition of the set S_{names} into m disjoint clusters. Each cluster S_j contains surface forms that refer to the same real-world entity(e.g., $S_j = \{$"APT28","Fancy Bear"$\}$).

3. Canonical Representative Selection: For each cluster S_j, a single canonical name s_j^* is selected.

$s_j^* = \text{select\_canonical}(S_j)$

The "select_canonical" function can be designed to choose the most frequent, shortest, or another representative name from the cluster.

4. Canonical Map Creation: The output of this stage is a mapping function, $\text{canon}(s) \to s^*$, which, for any raw name $s \in S_j$, returns its canonical representative s_j^*. Upon completion of this stage, a definitive mapping from any raw entity name to its unique, canonical identity is established, ensuring that the subsequent graph construction is based on a clean and consistent set of entities.

3.4. Stage 3: Canonical Graph Construction

Following the normalization of entities, the third stage of the model is focused on the construction of the final canonical knowledge graph, denoted as $G = (V, E)$. In this stage, the per-document sets of raw entities and the global canonical map, produced in the previous stages, are transformed into a single, unified graph structure. This process involves three key steps: defining the canonical nodes, inferring relationships between them, and aggregating these relationships into a final set of weighted edges.

3.4.1. Node Definition and Population

The nodes in the graph G are defined to represent the unique, canonical entities identified in Stage 2. To avoid ambiguity between entities with the same name but different categories (e.g., a country versus an organization with the same name), a node $v \in V$ is formally represented by a tuple containing its canonical name and its category.

Let V_{raw} be the global set of all raw entity tuples (s, c, p) extracted from the corpus, and let $\text{canon}(s) \to s^*$ be the canonical mapping function. The final set of unique nodes V is defined as:

$$V = (\text{canon}(s), c) \mid \exists p: (s, c, p) \in V_{raw}$$

For practical implementation, each unique node $v \in V$ is assigned a unique integer ID. Attributes such as a human-readable `label` (the canonical name) and the `category` are stored with each node.

3.4.2. Rule-Based Relational Inference

In this model, relationships between entities are not extracted directly from the text but are inferred programmatically. This is done to ensure consistency and to create a fully connected graph for all co-occurring entities within a document, as per the project requirements. This process is formalized by a relational inference function, denoted as Ω_{infer}

The function takes a pair of raw entity tuples that co-occur within the same document as its input. Based on the attributes of these entities (such as their assigned `role` and `category`), a relationship between them is inferred. The output of this function is a "raw relation tuple" that specifies the source, target, and type of the relationship.

Formally, for any pair of raw entities $\{e'_a, e'_b\}$ from a single document's entity set E'_i:

$$\Omega_{infer}(e'_a, e'_b) \to$$

$$r'_{ab} = (\text{source: } e'_s, \text{target: } e'_t, \text{type: } \rho)$$

where $e'_s, e'_t \in \{e'_a, e'_b\}$ defines the directionality, and ρ is the inferred edge type

from a predefined set of relations R. In the context of this study, the function Ω_{infer} implements a set of deterministic rules to infer edge types such as “attacks”, “is_attributed_to”, or “uses_infrastructure”.

3.4.3. Edge Aggregation and Weighting

The final step in the graph construction stage is the aggregation of all inferred relationships from across the corpus into a single, final set of weighted edges, denoted as E. This process ensures that the frequency and prevalence of each relationship are quantitatively captured.

First, the raw relation tuples (r'_{ab}) inferred for each document are canonicalized. The source and target entities within each tuple are mapped to their canonical representations using the $canon$ function established in Stage 2. This produces a set of "canonical edge tuples" for each document.

Next, all these canonical edge tuples from all documents are collected into a single global list, which can be denoted as L_{edges}. This list contains every instance of every relationship inferred across the entire corpus, allowing for duplicates.

The final edge set E is then formed by creating one unique, directed edge e for each unique canonical tuple found in the list L_{edges}. The attributes of this edge are then defined. Most importantly, the edge weight, $w(e)$, is defined as its frequency, or total number of occurrences, within the global list L_{edges}. This can be expressed as:

$$w(e) = \text{Frequency}(r^*, L_{edges})$$

where r^* is the unique canonical tuple corresponding to the edge e. In essence, the weight represents the number of separate times a specific relationship was observed in the source documents. Other attributes, such as the earliest date the relationship was observed, are also finalized during this aggregation step. This results in a comprehensive, weighted knowledge graph ready for formal analysis.

3.5. Stage 4: Formalized Graph Analysis

The final stage of the model is the quantitative analysis of the constructed canonical knowledge graph, $G = (V, E)$. This stage is focused on applying a series of formalized aggregations and ranking functions to the graph's nodes and edges. The objective is to transform the static topological structure into a set of interpretable metrics that reveal significant entities and dominant relational patterns.

3.5.1. Node Centrality and Ranking

To quantitatively assess the importance of each node, a set of adapted network centrality metrics was utilized. The foundation for this analysis is Degree Centrality. For a directed graph, this is separated into:

- In-Degree: The count of incoming edges to a node.
- Out-Degree: The count of outgoing edges from a node.

A more nuanced version, Weighted Degree, considers the Weight of each edge rather than just the count.

For the specific analytical goals of this study, a Filtered Weighted Degree was calculated. This approach adapts the standard Weighted Degree by limiting the calculation to only include edges of specific, predefined types. This allows for a more context-aware ranking of nodes based on their specific roles in the network.

Formally, for a given node v and a subset of edge types of interest $R_{filter} \subseteq R$, the Filtered Weighted Degree scores are defined as:

Filtered Weighted In-Degree: This score measures the total weighted interaction a node receives from a specific class of relationships.

$$S_{\text{in}}(v, \text{R}_{filter}) = \sum_{e=(u,v,\rho)\in E \text{where} \rho\in \text{R}_{filter}} w(e)$$

This score was used for the **Target Prioritization** analysis, where the ranking was based on S_{in} with the subset R_{filter}defined as {“$attacks$”}.

Filtered Weighted Out-Degree: This score measures the total weighted activity of a node for a specific class of outgoing relationships.

$$S_{out}(v, \text{R}_{filter}) = \sum_{e=(v,u,\rho)\in E \text{where} \rho\in \text{R}_{filter}} w(e)$$

This was used for the **Actor Activity Analysis**, where “HackerGroup” nodes were ranked based on S_{out} with the subset R_{filter} defined as {"attacks","uses_infrastructure"}

3.5.2. Relational Pattern and Profile Analysis

Analysis was also performed at the edge and local network level to understand relationship patterns and create detailed entity profiles.

Relationship Strength Analysis: To identify the most prevalent individual relationships, the entire set of edges E was ranked in descending order based on the weight function $w(e)$. This ranking was performed on both the global edge set and on subsets of edges, $E_\rho \subseteq E$, where all edges in the subset share the same type ρ.

Ego Network Profiling: To generate a detailed profile for a specific node of interest, v_i, its 1-hop ego network, G_{v_i}, was analyzed. The ego network is defined as the subgraph consisting of the central node v_i, the set of all its adjacent nodes (its neighborhood, $N(v_i)$), and all edges connecting these nodes. A profile was then constructed by aggregating and summarizing the attributes of the nodes and edges within this local network. This method was used to generate the detailed "Active Hacker Groups" and "Software Under Attack" reports.

4. Results

The four-stage model was applied to the corpus of 204 official CERT-UA reports. The process culminated in the creation of a canonical knowledge graph, which was then subjected to a series of quantitative analyses. This section presents the empirical results, starting with the overall structure of the graph, followed by detailed findings on key actors, targets, and operational patterns.

4.1. The Constructed Knowledge Graph

The execution of the data processing pipeline resulted in the construction of a comprehensive, canonical knowledge graph. A key step in this process was the GPT-5 powered normalization, which consolidated raw entity mentions into unique, canonical nodes. This normalization was critical for data accuracy, reducing the number of Government Body entities by 38.5% and HackerGroup entities by 18.3% by merging aliases and synonyms. The performance of this normalization stage is detailed in Table 1.

Table 1
Entities normalization effectiveness

Entity type	Original count	Reduction rate
Government Body	96	38.5%
Country	32	25.0%
Domain	1929	24.5%
Sector	121	22.3%
Hacker Group	93	18.3%
Software	882	15.4%
Infrastructure	15	13.3%
Individual	19	10.5%
Organization	174	8.6%
Device	31	3.2%

The final graph was constructed from 2,674 canonical nodes and 81,755 aggregated, directed edges. The graph's composition showed a high density of infrastructural elements, with Domain and Software nodes being the most numerous. The analysis of edge types revealed that the graph contains 8 unique relationship types, with a maximum edge weight of 12, indicating that the most frequent relationship was observed across 12 separate incidents.

4.2. Key Actor and Target Identification

The quantitative analysis of network graph centrality and attack relationships reveals a highly structured and persistent pattern of targeting against Ukrainian state and civil society entities. The analysis processed a total of 2,674 nodes and 4,528 attack relationships, identifying the most frequent and heavily weighted targets across eight distinct entity types. The findings underscore a multi-pronged cyber campaign focused on government, military, critical infrastructure, and the public information sphere.

The broadest targets were national and societal sectors. As detailed in the source data, "Ukrainian citizens" was the most frequently attacked sector (aggregated weight: 205.0), followed closely by "Government bodies of Ukraine" (aggregated weight: 118.0). This indicates a widespread campaign aimed at both the general populace and the state apparatus. Reinforcing this, when analyzed by country, Ukraine was the overwhelmingly primary target, with an aggregated attack weight of 496.0, an order of magnitude greater than any other nation.

A more granular analysis of specific entities highlights the campaign's strategic priorities. Table 2 summarizes the top-targeted entity within the most significant categories. The

"Сили оборони України" (Defense Forces of Ukraine) was identified as the single most critical target, absorbing an aggregated attack weight of 129.0. This focus on the unified defense command structure points to sophisticated intelligence-gathering and disruption efforts aimed at military operations. In the private sector, the national email service "UKR.NET" was the most prominent target (aggregated weight: 149.0), alongside a clear pattern of attacks against major media organizations such as "Україна 24" (32.0), "TSN" (30.0), and "Ukrinform" (20.0). This demonstrates a parallel effort to compromise civilian communications and disrupt the national information space.

Table 2
Top Targeted Entities by Category

Category	Most Targeted Entity	Aggregated Attack Weight
Government Body	Defense Forces of Ukraine	129.0
Organization	UKR.NET (National Email Service)	149.0
Sector	Ukrainian citizens	205.0
Country	Ukraine	496.0

Furthermore, an analysis of the software targeted by adversaries reveals the specific vectors used in these campaigns (Table 3). The attackers prioritized both ubiquitous communication platforms and specialized military systems. Public messaging applications like “Telegram” (110.0), “WhatsApp” (87.0), and “Signal” (65.0) were heavily targeted. These platforms were not typically attacked by exploiting software vulnerabilities, but were rather used as a vector for social engineering and phishing campaigns, where attackers leverage the public's trust in these applications to deliver malware or steal credentials. This pattern highlights a strategic focus on compromising trusted communication channels as a primary means of initial access. Concurrently, highly specialized Ukrainian military “C4ISR” (Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance) software, such as “DELTA”, “ТЕНЕТА”, and “Кропива” (each with a weight of 54.0), were targeted with equal intensity. This dual focus indicates a sophisticated adversary capable of running both large-scale phishing and social engineering campaigns against the general public and highly tailored technical operations against hardened military targets.

Table 3
Prominent Software Targets by Type

Software Category	Examples	Aggregated Attack Weight
Communication	Telegram, WhatsApp, Signal	110.0, 87.0, 65.0
Military C4ISR	DELTA, ТЕНЕТА, Кропива	54.0 (each)
Corporate / Email Systems	Microsoft Outlook, Roundcube	33.0, 29.0

The analysis of attacker activity, presented in Table 4, revealed that a small number of highly active groups are responsible for a large portion of the observed attacks. The top three most active actors by aggregated attack weight were identified as APT28 (Attack Weight: 31), Sandworm (Attack Weight: 25), and Armageddon (Attack Weight: 17). The data confirms that these key actors are consistently attributed to the "Russian Federation" and primarily target Ukraine and its governmental and defense sectors.

Table 4
Hacker Groups Attack Weight

HackerGroup	Attack weight
APT28	31
Sandworm	25
Armageddon	17
UAC-0050	15
UAC-0133	14
UAC-0002	14
UAC-0063	10
Turla	9
UNC4221	9
Seashell Blizzard	9

4.3. Threat Ecosystems and Tooling

The dense relationships within the graph allow for the clear identification of distinct threat ecosystems based on actor-tool-target connections. The analysis of the most frequently used tools, detailed in Table 5, shows that general-purpose software like PowerShell is the most widely adopted tool, used by at least 29 distinct hacker groups. However, more specialized malware is often closely associated with specific actors, forming clear operational ecosystems. For example, the Remcos Remote Access Trojan (RAT) was predominantly used by the group UAC-0050 (usage weight of 8), while the SmokeLoader malware was

exclusively linked to UAC-0006 in this dataset. Similarly, the GammaLoad malware is a key component in the arsenal of the Armageddon group.

Table 5
Tools Used by Hacker Groups

Tool	Usage	Users
PowerShell	38	29
Remcos	18	10
MSHTA	17	12
Windows Script Host	14	11
Cobalt Strike Beacon	14	8
Remote Utilities	13	7
Lumma Stealer	10	8
Python	9	7
ngrok	7	7
KAZUAR	7	6
Quasar RAT	7	6
SmokeLoader	6	1
Venom RAT	6	6
PEAKLIGHT	6	6
DarkCrystal RAT	6	5

The geopolitical dimension of these ecosystems is revealed by the “is_attributed_to” relationships. Key threat actors like Armageddon, APT28, Sandworm, and Turla were all formally linked to the "Russian Federation", with specific connections to government bodies such as the Federal Security Service (FSB) and the Main Directorate of the General Staff (GRU). This confirms the state-sponsored nature of the primary threat ecosystems operating against Ukraine.

5. Discussion

The results presented in the previous section serve as an empirical validation of the proposed formal model. This section is intended to discuss the broader analytical capabilities that the model enables, the general implications of this methodology for the field of sensitive data analysis, and the inherent limitations of the approach.

5.1. Interpretation of the Model's Analytical Capabilities

The application of the model to the CERT-UA corpus demonstrates its capacity to transform a large volume of unstructured text into a structured, interpretable map of a complex domain. Several key analytical capabilities were revealed.

First, the model excels at identifying the central actors and structural cornerstones within a complex system. Through the use of network centrality metrics, the model moves beyond simple frequency counts to quantify the topological importance of each entity. In the case study, this allowed for the immediate identification of the most influential threat actors and the most critical targets, demonstrating the model's utility in prioritizing focus within any large-scale dataset.

Second, the methodology allows for the automated discovery of latent thematic ecosystems. The community detection analysis showed that the model can automatically cluster entities into coherent groups based on the density of their inferred relationships. These clusters represent meaningful, real-world structures—in the CTI case study, they corresponded to distinct "theaters of operation." This capability is generic and could be applied to uncover hidden communities in other domains, such as identifying research clusters from academic papers or interconnected corporate networks from financial reports.

Finally, the model enables the analysis of complex, multi-faceted relational patterns. The results highlighted the dual role of software as both a weapon and a target. This type of nuanced insight is made possible by the rich, typed-edge graph structure, which allows analysts to move beyond simple co-occurrence analysis and explore the specific nature of interactions between entities.

5.2. Broader Implications for Sensitive Data Analysis

The implications of this work extend beyond the specific domain of cybersecurity. From an academic perspective, the primary contribution is the formalized framework for making LLM outputs more reliable and analytically useful. By integrating a state-of-the-art LLM (GPT-5) into a structured pipeline with deterministic normalization and inference stages, this model offers a replicable blueprint for conducting rigorous research with unstructured text. This addresses the "formalization gap" and provides a path for applying LLMs in other sensitive fields like legal text analysis, intelligence reporting, and financial compliance monitoring.

From a practical standpoint, the methodology offers a significant acceleration of the knowledge

discovery process. For domain experts and analysts in any field, the model can automate the laborious task of processing and structuring vast quantities of documents. This creates a queryable knowledge base from what was previously an inert archive of text, freeing human experts to focus on high-level strategic interpretation. The ability to generate a data-driven, macroscopic view of a domain and then drill down into specific entities provides a powerful tool for any intelligence-driven workflow.

6. General Model Limitations

Despite the capabilities of the proposed model, several inherent limitations should be considered. These limitations help to contextualize the findings and identify areas for future research.

Dependency on the Input Corpus. A primary limitation is that the model's output is fundamentally a reflection of its input corpus. The insights derived from the analysis represent the world as described in the source documents, not necessarily the absolute ground truth. Consequently, the model is subject to any biases, gaps, or specific perspectives present in the data it processes.

Dependency on Component Performance. The overall accuracy of the model is dependent on the performance of its core components: the LLM and the rule-based inference engine. While GPT-5 represents the state of the art, it is not infallible. Errors made during the initial extraction or normalization stages can be propagated through the system. Similarly, the inference rules, while deterministic, are based on a specific logical model of the domain and may not capture all relational nuances.

Static and Aggregated Representation. The methodology produces a static, aggregated representation of what are often dynamic events. By collapsing temporal information into a single graph, the sequencing and evolution of relationships over time are lost. While this aggregated view is powerful for identifying the overall structure of the threat landscape, a temporal analysis is required to understand the dynamics of the system. This remains a key area for future work.

7. Conclusion

This study has proposed and validated a four-stage formal mathematical model for transforming unstructured cyber incident reports into a canonical knowledge graph. By integrating the semantic extraction capabilities of GPT-5 with deterministic rule-based inference and graph-theoretic analysis, the model bridges the "formalization gap" between probabilistic LLM outputs and reliable, analyzable structures. Applied to a corpus of 204 CERT-UA reports, the approach successfully identified central threat actors, critical targets, and distinct operational ecosystems, offering a macroscopic yet actionable view of Ukraine's cyber threat landscape.

From a **theoretical perspective**, the research contributes a replicable framework that demonstrates how next-generation LLMs can be embedded into a mathematically defined pipeline. This advances the academic discourse on sensitive data analysis by showing that hybrid systems where probabilistic reasoning is tempered by formal normalization and deterministic rules can mitigate risks of inconsistency and hallucination. More broadly, the model extends knowledge graph construction methodologies and provides a basis for future exploration of dynamic, temporal, and multi-source data integration.

From a **practical perspective**, the model delivers tangible value for analysts and decision-makers in national security, cybersecurity operations, and related fields. By automating the structuring of vast archives of text, it reduces reliance on manual review and enables faster, data-driven insights. The ability to highlight high-value actors, reveal latent ecosystems, and prioritize targets makes the framework directly relevant to threat intelligence workflows, incident response, and strategic planning. Beyond cybersecurity, the pipeline can be adapted to other sensitive domains such as legal compliance, financial monitoring, and intelligence reporting—anywhere structured knowledge must be distilled from unstructured narratives.

In sum, this research demonstrates both the scientific significance and the practical utility of a formalized, hybrid approach to sensitive data processing. It establishes a foundation for scalable, explainable, and domain-agnostic applications of LLMs, thereby contributing to both the academic theory of automated text analysis and the operational practice of intelligence-driven decision support.

8. Future Work

While this study provides a robust foundation, several avenues for future research can be pursued to extend and generalize the proposed model.

Enhancing Robustness with a Swarm of Virtual Experts: The current model's dependency on a single LLM instance can be mitigated. Future work could implement a "Swarm of Virtual Experts" [10] methodology to improve the accuracy of the foundational extraction and normalization stages. This approach involves querying multiple, diverse LLM agents for the same task and aggregating their outputs via a consensus mechanism, thereby reducing the impact of individual model biases and leading to a higher-fidelity knowledge graph.

Multi-Source Data Fusion: The model was validated on a homogenous corpus. Future work should focus on its application to fusing data from a wider variety of text sources, such as legal documents, financial filings, intelligence briefings, or open-source news reports. This would test the model's ability to create a comprehensive knowledge graph from diverse and potentially conflicting information.

Development of Advanced Analytical Models: The current analysis, based on metrics such as Filtered Weighted Degree, proved effective for identifying key entities. Future research could significantly expand these analytical capabilities. One direction is the formalization of a Multi-Dimensional Node Scoring framework, which would involve designing new, domain-specific metrics to create richer, more comprehensive profiles of entities like threat actors and their tools. Furthermore, to synthesize these multi-dimensional profiles into a single, actionable ranking, a Composite Node Ranking Model could be developed. Future work in this area could focus on: (a) creating flexible, goal-oriented scoring functions for diverse analytical tasks (e.g. ranking malware by threat level); and (b) exploring methods for dynamic calibration of the model's weights using machine learning to adapt to the evolving threat landscape.

Development of an Interactive Analytical Dashboard: The model presented in this paper can serve as the backend for a powerful, interactive tool for analysts in any domain dealing with large volumes of text. Future efforts could be directed towards developing a user interface with capabilities for dynamic filtering, drill-down analysis of specific entities, and visual exploration of relationships, thereby empowering human experts to validate hypotheses more efficiently.

References

[1] M. Arazzi et al., “NLP-Based Techniques for Cyber Threat Intelligence,” Nov. 15, 2023, arXiv: arXiv:2311.08807. doi: 10.48550/arXiv.2311.08807.

[2] A. Hogan et al., “Knowledge Graphs,” ACM Comput. Surv., vol. 54, no. 4, pp. 1–37, May 2022, doi: 10.1145/3447772.

[3] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, “Neural Architectures for Named Entity Recognition,” Apr. 07, 2016, arXiv: arXiv:1603.01360. doi: 10.48550/arXiv.1603.01360.

[4] A. Vaswani et al., “Attention Is All You Need,” 2017, arXiv. doi: 10.48550/ARXIV.1706.03762.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” May 24, 2019, arXiv: arXiv:1810.04805. doi: 10.48550/arXiv.1810.04805.

[6] T. B. Brown et al., “Language Models are Few-Shot Learners,” July 22, 2020, arXiv: arXiv:2005.14165. doi: 10.48550/arXiv.2005.14165.

[7] Z. Ji et al., “Survey of Hallucination in Natural Language Generation,” ACM Comput. Surv., vol. 55, no. 12, pp. 1–38, Dec. 2023, doi: 10.1145/3571730.

[8] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu, “Unifying Large Language Models and Knowledge Graphs: A Roadmap,” IEEE Trans. Knowl. Data Eng., vol. 36, no. 7, pp. 3580–3599, July 2024, doi: 10.1109/TKDE.2024.3352100.

[9] G. Husari, E. Al-Shaer, M. Ahmed, B. Chu, and X. Niu, “TTPDrill: Automatic and Accurate Extraction of Threat Actions from Unstructured Text of CTI Sources,” in Proceedings of the 33rd Annual Computer Security Applications Conference, Orlando FL USA: ACM, Dec. 2017, pp. 103–115. doi: 10.1145/3134600.3134646.

[10] D. Lande and L. Strashnoy, “Swarm of Virtual Experts in the Implementation of Semantic Networking,” 2024, doi: 10.13140/RG.2.2.16686.11845.