Secure-by-Design Re-Engineering of Anti-Mafia Processes and Investigative Workflows

Mario Varlese[1,*], Simon Pietro Romano[1], Giancarlo Sperlì[1] and Andrea Vignali[1]

[4]University of Naples Federico II, Naples, Italy

Abstract

Intelligence analysis plays a key role in investigative activities that require processing vast amounts of confidential and heterogeneous information. Often, key information is fragmented across different documents, collected over years of investigative work, and this fragmentation constitutes a significant challenge for investigators. Structuring those data and constructing a knowledge base is vital for maximizing the chances of success of an investigative activity. Relation extraction is becoming a common trend in natural language processing research. However, automatically extracting entities and relations from domain-specific documents is a complex task, which becomes even more complicated when the specific domain is as intricate as the legal domain. This research focuses on the design and the implementation of a framework to build a relational graph from unstructured documents. The domain-specific documents belong to the domain of investigations conducted by the Italian National Anti-Mafia and Counter-Terrorism Directorate. The framework is designed with strict privacy and sensitivity constraints, demonstrating that AI technologies can be safely applied in highly regulated investigative contexts. As a result of this work, suggestions are identified for future research.

Keywords

Anti-Mafia, Investigative Workflows, Secure-by-Design, Cybersecurity

1. Introduction

Intelligence analysis is a major area of interest within the field of modern investigative activities. While open-source intelligence (OSINT) focuses on publicly available information, the analysis of anti-mafia investigative documents requires the processing of confidential, sensitive, and non-public data, a context we refer to as closed-source intelligence (CSINT). The huge amount of information produced by the analysis of investigative case files, together with their intrinsically interconnected nature, offers new opportunities for investigative activities but also entails significant challenges in terms of data organization, interpretation, and correlation. It is now well established from a variety of studies that graph-based representations of entities and relations provide an effective way to structure complex and heterogeneous information, enabling analysts to identify hidden links and relevant patterns that would not be immediately visible otherwise. However, a major problem with this kind of application is the manual effort required to feed these graph-based systems, especially when dealing with large volumes of unstructured text. A considerable amount of literature has been published on Information Extraction [1],[2] (IE). Several researchers have investigated whether traditional Natural Language Processing [3] (NLP) models can effectively process unstructured text. However, these models often struggle with irregular syntax, informal language, technical jargon, and abbreviations. In recent years, there has been increasing interest in generative artificial intelligence (Gen-AI). Specifically, Large Language Models [4] (LLMs) have been studied extensively and have gained growing attention for their reasoning capabilities and text understanding. A considerable amount of literature has been published on the employment of LLMs for IE tasks [5],[6]. These studies confirm that LLMs can effectively identify entities, classify semantic roles, and extract relational structures from unstructured text. Building

Joint National Conference on Cybersecurity (ITASEC & SERICS 2026), February 09–13, 2026, Cagliari, IT

*Corresponding author.

mario.varlese@unina.it (M. Varlese); spromano@unina.it (S. P. Romano); giancarlo.sperli@unina.it (G. Sperlì); andrea.vignali@unina.it (A. Vignali)

0009-0006-1981-3548 (M. Varlese); 0000-0002-5876-0382 (S. P. Romano); 0000-0003-4033-3777 (G. Sperlì); 0000-0002-0273-1056 (A. Vignali)

on these capabilities, and given the necessity of automating the construction of investigative graphs, LLM-based pipelines are particularly suitable for OSINT/CSINT scenarios.

This work is situated within the context of the Direzione Nazionale Antimafia e Antiterrorismo (National Directorate for Anti-Mafia and Counter-Terrorism, DNAA). More specifically, it focuses on a reengineering of their National Database. The DNAA plays a fundamental role in combating organized crime and terrorism in Italy, coordinating investigative activities on a national scale. Its responsibilities extend beyond the operational management of individual investigations to encompass broader efforts against organized crime, including mafia activities, drug trafficking, money laundering, and human trafficking.

National Deputy Prosecutors, in addition to coordinating investigations across the various DDA (Direzioni Distrettuali Antimafia, or District Anti-Mafia Directorates), are tasked with monitoring these areas, collecting and analyzing information that can inform the development of new investigative strategies. A key tool supporting these activities is the National Database, which is fed by procedural documents from the DDAs. This database represents a crucial asset for the retrieval and analysis of relevant data.

The National Database, which centralizes and manages the procedural records of the DDAs, provides investigators with rapid and structured access to a vast amount of information. However, with the advancement of investigative techniques and the growing complexity of the data collected, a process of reengineering has become necessary. The aim is to improve data management and retrieval, reducing the fragmentation of information and mitigating the risks associated with the dispersion of critical data across unstructured documents.

Even though the use of LLMs for extracting entities and relations from text is a well-known field, there is limited literature analyzing the effectiveness of this technology in domain-specific legal contexts [7], particularly when stringent privacy requirements are involved.

This paper describes the design and implementation of a framework to support the investigative activities of DNAA operators, developed with careful consideration of the stringent privacy and sensitivity constraints of the information being processed. Specifically, the contributions of this work are as follows:

- Proposing a methodology to automate the transformation of heterogeneous investigative documents into a relational graph.
- Investigating the effectiveness of LLMs when applied to DNAA-specific documents, which are characterized by complex structures and domain-specific legal terminology.
- Developing and deploying a framework to support investigative activities while ensuring compliance with data privacy requirements.

The rest of the paper is organized as follows. Section 2 provides an overview on the literature. Section 3 describes the main problem in adopting LLM to DNAA processes and related privacy issues. Section 4 explains the design criteria identified and the methodology used to transform unstructured investigative texts into relational graphs. Section 5 illustrates the implementation details of the framework. While section 6 explains the integration of the DevSecOps paradigm in the development process. Section 7 describes an investigation case of study. Finally, section 8 summarizes the main findings and outlines directions for future research.

2. Related Works

A considerable amount of literature has been published on domain-specific IE tasks [8],[9],[10] and [11]. To date, several studies have investigated the application of artifical intelligence to automate extraction of information from legal text. [12] designed and evaluated a hybrid classification model to automatically extract key information from legal documents, specifically from first-instance criminal verdicts for homicide in South Korea. In particular, their work was based on the use of transformer-based models fine-tuned for the specific domain. [13] proposes a framework for information extraction from

Indian legal documents using MinIE and by defining a pipeline that includes preprocessing steps based on regex, coreference resolution, and named entity recognition through spaCy NER. [14] proposed an approach for extracting elements from legal documents based on Stacked Long Short-Term Memory (SLSTM) networks to capture the complex dependencies typical of legal text. More recent attention has focused on exploring LLMs capabilities in domain specific IE tasks [15]. In their survey [16] the authors identify significant potential in LLMs for relation extraction tasks, particularly in contexts characterized by a scarcity of training examples. Furthermore, they recognize the importance of their ability to capture broader contexts and therefore detect relations across distant segments of text. Authors of [17] focus on the task of information extraction in the Brazilian legal domain, specifically comparing approaches based on traditional supervised learning and ChatGPT. Even if supervised learning models achieve slightly better results, they rely solely on prompt engineering techniques, making this solution suitable in contexts where no dataset is available or in scenarios where development time needs to be significantly reduced. Similarly, [18] evaluated performance of GPT-4 in an in-context learning setup on a dataset manually annotated from legal wills of two U.S. states. The results were satisfactory, demonstrating that GPT-4 is capable of performing the task reasonably well. Thus, interesting research on IE in the legal domain has been carried out, but to the best of our knowledge, there are no existing works addressing the investigative context of the Antimafia And Antiterrorism Directorate (DNAA). Furthermore, although many authors have explored the use of generative LLMs and noted how models such as GPT can be promising for information extraction tasks, our choice cannot rely on the use of these models. Indeed, unlike the contexts previously highlighted, the data produced during an investigation are subject to stringent constraints, as they must remain within certified secure environments. Therefore, in our work we will use open-weight models that are significantly smaller in terms of parameters compared to models such as GPT-4.

3. Problem Statement

Relation Extraction (RE) is a particular type of information extraction task. The task consists of three main NLP tasks: NER, relation identification and relation classification [16]. More formally, given an unstructured text T we want to obtain a structured representation T' that clearly expresses the semantic relations between the entities present in the text. The structured representation we propose consists of a set of triples (s, p, o), where:

- s represents the subject, i.e., the entity from which the relationship originates,
- p represents the predicate, i.e., the relationship or link between the subject and the object,
- o represents the object, i.e., the entity toward which the predicate points.

Formally, the set of triples can be defined as follows:

$$T' = \{(s_1, p_1, o_1), (s_2, p_2, o_2), \ldots, (s_n, p_n, o_n)\}$$

where each triple (s_i, p_i, o_i) represents a relationship extracted from the text T, and n indicates the number of identified relations. In more general terms, the set of triples can be formalized as:

$$T' = \{(s, p, o) \mid s \in S, p \in P, o \in O\}$$

where:

- S is the set of subjects (entities),
- P is the set of predicates (relations),
- O is the set of objects (entities).

The problem addressed in this work therefore concerns the automation of this process, with the aim of obtaining a clearer and more immediate semantic representation that highlights the information of greatest investigative relevance. However, there are several significant challenges related to the

application domain. We summarize the main obstacles that make RE task far from trivial in Table 1. The main issues range form text complexity to syntactic and semantic complexity [19]. Legal documents are characterized by a rich lexicon and semantically dense complex sentences including uniquely legal expressions that depend on the specific legal sub-area.

Obstacle	Description
Language Ambiguity	Includes different levels of ambiguity, such as lexical ambiguity (where a word can have multiple meanings), implicit information that requires contextual inference, as well as complex or ambiguous entities.
Noisy Data	Texts may contain typographical errors, grammatical mistakes, or other forms of 'noise'.
Specialized Technical Language	Technical terminology may not be immediately recognizable.
Sentence Length and Complexity	Text structured in sentences with multiple clauses and propositions.
Lack of Standardization	Unstructured texts often do not follow well-defined grammatical or formatting rules.
Data Sensitivity	The data contained in investigative case files impose stringent privacy requirements. Specifically, they must be kept in-house, and their processing cannot be entrusted to third-party platforms.

Table 1
Main obstacles in extracting relations from unstructured text.

4. Methodology

This section details the methodology that led to the development of the framework presented in Section 5. In order to clearly define the types of information that will populate our knowledge base, the entities and relations of interest were identified through two complementary taxonomies. For the identification of entities, we initially relied on OntoNotes5, a well-established resource for entity recognition and coreference in general-domain text. However, due to the specific characteristics of DNAA investigative documents, such as legal terminology, domain-specific abbreviations, and highly structured procedural information, it was necessary to extend OntoNotes5 to include additional entity types relevant to the DNAA context, as summarized in table 2.

Entity	Description
LegalType	Type of legal act
LegalTypeEntity	Type of legal entity (e.g., registry office)
LegalInfo	Type of legal information (e.g., hearing)
ArticleNumber	Number of the article cited in the text
LegalNumber	Number of the legal act cited in the text
LegalRole	Legal role of the people mentioned in the text
Crime	Type of crime that was committed or mentioned
ProductID	Product serial number
Status	Deceased/alive
Clan	Family or mafia association not covered by previous cases
Phone	Telephone number associated with a person or company
Progressive	Sequence number related to the phone call

Table 2
The proposed extension of the Ontonotes5 ontology, detailing the added entities along with their descriptions.

Similarly, for the definition of relations of interest, we started from a POLE model (Person, Object, Location, Event), which we adapted to the specifications of the DNAA based on the previously identified entities. As an example, table 3 summarizes the allowed relations for the entity type *PERSON*.

Table 3: Relations between PERSON and PERSON

Relation Type	Description
KNOWS	A generic acquaintance relationship between two people.
FAMILY_RELATION	A generic family relationship between two people.
KNOWS_IN_PERSON	An acquaintance relationship between two people who have met physically.
KNOWS_ONLINE	An acquaintance relationship between two people who have communicated electronically.

To tackle the obstacles identified in Section 3, a workflow has been defined that employs LLMs in several of the defined steps. Thanks to their text comprehension capabilities [20], LLMs are indeed particularly suitable for addressing issues related to the lack of standardization and the presence of noisy data. Regarding the management of ambiguities and specialized technical language, several methods currently exist. In this regard, one of the most effective approaches consists of adopting fine-tuning techniques to specialize the model on a specific task. In this way, the model begins to learn domain-specific terminology and context, improving its ability to correctly identify entities and relations. However, even if this strategy is indeed effective, it is not always feasible. In fact, in order to train a model, it is necessary to have access to a large dataset that is characteristic of the target domain. However, when fine-tuning is not feasible, one can leverage the knowledge that the model has already acquired during pre-training by applying prompting techniques to elicit the desired behavior. [21] provides a systematic survey of prompt engineering and shows how different prompt techniques can lead large language models to solve downstream tasks.

The implemented framework relies on both approaches adopted in the information extraction workflow. Specifically, for the entity extraction task (NER), a model specialized on a manually annotated dataset of 1,500 samples was employed, whereas the relation extraction (RE) step heavily relies on prompting techniques such as Chain of Thought (CoT) and dynamic few-shot learning.

The defined workflow also introduces a step for splitting the document into smaller chunks. This is not only related to the limitation imposed by the context window of LLMs, but also, most importantly, as noted by [22], due to their difficulty in effectively utilizing all the information provided in the input.

Finally, the last design criterion of paramount importance concerns data privacy. Indeed, it limits the possibility of using models such as GPT, cloud-based services, or similar, as their use would entail a violation of privacy regulations. Therefore, the feasible choice is constrained to using small, open-weight models that can operate offline.

Figure 1 illustrates the main steps included in the implemented workflow. The overall process can be structured into the following macro-phases.

Entity Identification. As shown, the starting point for the extraction process is the document itself. First, it is reduced into smaller text chunks, and then the relevant entities are extracted. Finally, the extracted entities are located within the text, producing a version of the document enriched with the recognized entities. Figure 2 shows an example of a chunk of text enriched with the extracted entities.

Text Chunking. The entity-enriched document text is split into smaller blocks, referred to as chunks. *Text Normalization.* Each chunk of the entity-enriched document is transformed through a set of static rules aimed at replacing abbreviations and domain-specific terms with their full and more comprehensible forms. The objective of this step is therefore to transform the text in such a way as to make explicit these sector-specific abbreviations, which can otherwise hinder text comprehension. This is achieved through the application of static rules based on regular expressions. *Coreference Resolution.* The original text is converted into an unambiguous version by identifying and replacing implicit or ambiguous expressions with their primary entities, ensuring that every reference is clear and unique. Following this step, a more concise representation of the original text is generated, in which the entities of interest (persons, locations, events, etc.) are expressed in a “key”:“value” format.

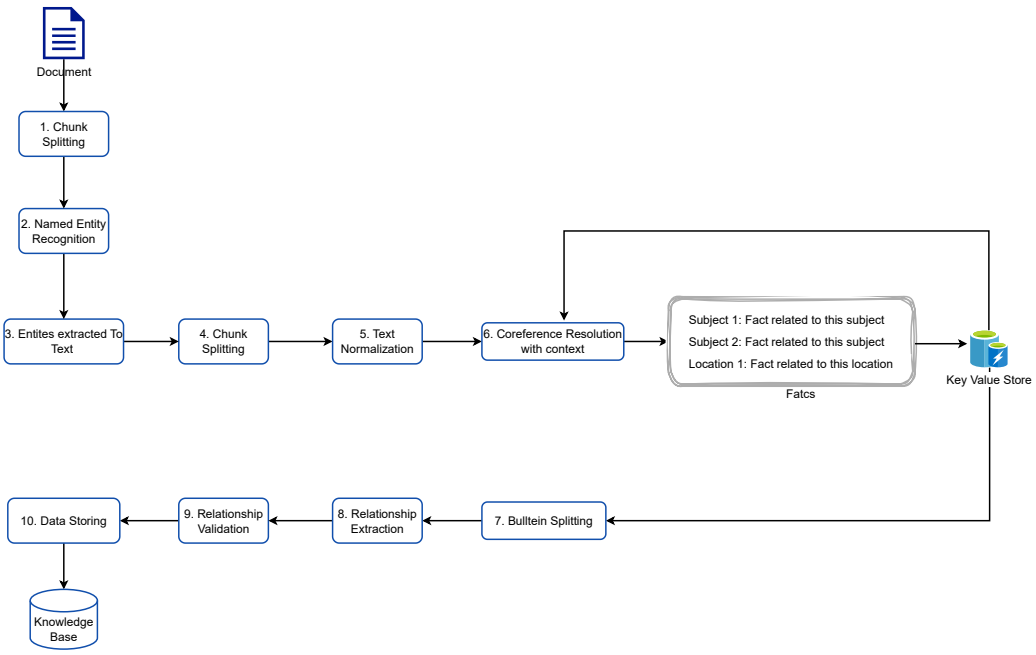

Figure 1: Workflow of the implemented framework for investigative information extraction

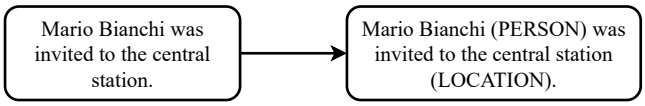

Figure 2: Chunk of text enriched with the recognized entites

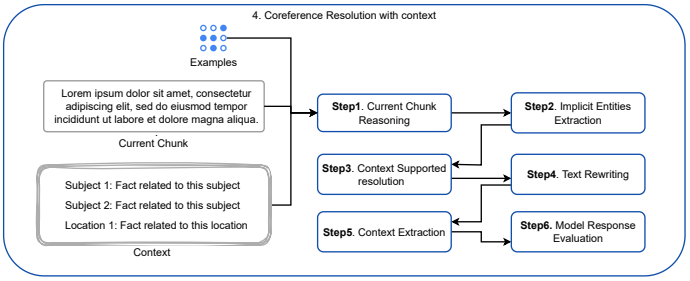

Figure 3: Coreference Resolution Pipeline

As depicted in figure 3, the Large Language Model responsible for coreference resolution receives the following elements as input:

1. A set of examples that *"explain"* to the model the reasoning process;
2. The current chunk of text to be analyzed;
3. The context extracted from the analysis of previously processed chunks.

Among the strategies commonly grouped under prompt engineering and recognized as most effective in improving LLM performance, we find the Chain of Thought (CoT) mechanism, as well as so-called Few-Shot learning. The first technique leverages the strategy used in the training of these models: their ability to predict the next token given an input text. CoT is a prompting technique that encourages the model to generate the reasoning process that led it to a particular conclusion. The second technique allows providing the model, in addition to the context and task-related information, with a set of

examples that illustrate how to perform a specific task. The Few-Shot learning technique enables conditioning the model's response, improving its performance, and making it possible to apply the model even when there is insufficient data to fine-tune it for the specific use case.

As anticipated, the model follows a step-by-step reasoning process (*CoT*), as detailed below:

1. **Current Chunk Reasoning**. The model analyzes the current text in order to locate named entities (persons, locations, dates) and pronouns (such as "he", "she"). It then annotates anaphoric references, linking pronouns and abbreviated names to their respective entities, thereby preparing the ground for coreference resolution.
2. **Implicit Entities Extraction**. The model analyzes the current text to identify implicit entities, recognizing pronouns or references that lack an explicitly named entity but can be associated with a specific already-known entity thanks to the model's prior knowledge. Pronouns and generic terms are then transformed into proper names or detailed descriptions.
3. **Context Supported Resolution**. The context provided to the model (generated from the execution of previous chunks) is used to link pronouns and anaphoric references to the correct entities, ensuring consistency with the available information. In case of ambiguity, the context helps to disambiguate and determine the correct association between pronouns and entities.
4. **Text Rewriting**. The text is rewritten in an explicit form, using the information obtained from the previous steps.
5. **Context Extraction**. The original context is updated by integrating the new information obtained from the explicit text. Details about entities and their relations, based on the coreference resolution, are added. This information is preserved as defined in the initial context, but enriched with the new information.
6. **Model Response Evaluation**. The model receives its own response as input and is tasked with further processing and refining it, if necessary.

Key-Value Element Selection. The elements represented in key-value format are split into smaller chunks to be analyzed separately. *Relation Extraction.* The elements within the chunk under examination are analyzed, and the relations of interest are extracted. As with the coreference resolution task, the relation extraction task also relies on Large Language Models. Following the same considerations made for the coreference resolution task, the mechanisms applied for performing this task are the CoT strategy, Few-Shot learning, and self-critique for reviewing the response provided by the model.

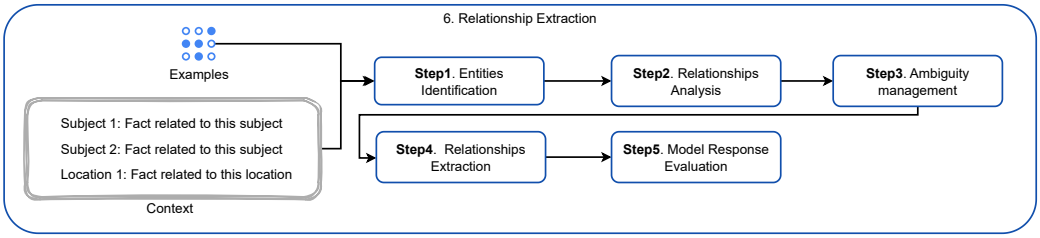

Figure 4: Relationship extraction step

Figure 4 illustrates the reasoning chain followed by the model to arrive at an appropriate set of extracted relations.

The reasoning steps, whose ultimate goal is to extract the relevant relations, are detailed below:

1. **Entities Identification**. The objective of this step is to extract the entities present in the text, as well as associating them with pertinent details;
2. **Relations Analysis**. Identify the relations between entities, ensuring that they are contextually appropriate and verified within the text, ignoring irrelevant information;
3. **Ambiguity Management**. Clarify any remaining ambiguous or indirect references;

4. **Relations Extraction**. The relations between extracted entities are represented using the YAML format;
5. **Model Response Evaluation**. The model is invoked, receiving its previous response and the task, with the aim of reviewing and refining the provided output.

Following this chain of steps illustrated in Figure 4, a set of relations in YAML format is produced, as shown below:

```yaml
- entity_from:
    entity_name: "Entity Name A"
    entity_type: "Entity Type A"
  entity_to:
    entity_name: "Entity Name B"
    entity_type: "Entity Type B"
  relationship:
    relationship_name: "Relationship Name"
    relationship_type: "Relationship Type"
    text_from: "Text that led the model to extract
    the relationship"
```

To explain this structure, it is necessary to imagine a relationship as a directional graph, where the nodes are the entities and the relationship (the verb) is the edge connecting the two entities. Thus, under the key *"entity_from"* there is the entity from which the relationship originates, characterized by a *name* (i.e., the textual value extracted from the text) and a *type* among those allowed. Under the key *"entity_to"* is the entity where the relationship ends. Finally, under the key *"relationship"* is the extracted relationship itself, also characterized by a *name* (i.e., the verb, among those allowed in the domain, inferred from the analyzed text) and a *type*, represented in the format *"Type of first entity"* - *"Type of second entity"*.

Relation Validation. The extracted relations are validated and, if necessary, "corrected". Since the ultimate goal of the relationship extraction pipeline is to populate a knowledge base, it is essential to standardize the set of relations of interest. Natural language, on which LLMs rely, offers multiple ways to express the same concept. Therefore, it is crucial to ensure that the extracted relations are not only syntactically correct, but also compliant with this standardized set.

To perform validation, a set of validation rules has been defined:

1. the structure must adhere to the defined format;
2. the type of the entity under the *entity_from* key must be among those allowed;
3. the type of the entity under the *entity_from* key must be consistent with the type defined under the *relationship_type* key;
4. the type of the entity under the *entity_to* key must be among those allowed;
5. the type of the entity under the *entity_to* key must be consistent with the type defined under the *relationship_type* key;
6. the relationship type defined under the *relationship_type* key must be among those allowed;
7. the relationship name contained under the *relationship_name* key must be among those allowed for the relationship type defined under the *relationship_type* key.

The rule-based module analyzes the structure of the relationship and evaluates its correctness. If the relationship is found to be incorrect, instead of discarding it immediately, a repair module is activated. This module uses an LLM, invoked using the same strategies described in the previous sections, relying on CoT and in-context learning. The model receives the relationship as input and attempts to correct it by leveraging the text contained in the *text_from* field of the relationship itself.

Knowledge Base Population. The validated relations are inserted into the knowledge base.

5. Implementation

The implemented framework makes extensive use of the LangChain framework to implement the chain of steps described, to properly format prompts, and to split the text into chunks. The Large Language Model used for tasks such as named entity recognition, coreference resolution, relation extraction, and relation correction is LLaMA. Finally, for representing the knowledge base extracted from the documents, Neo4j was chosen as the database.

5.1. LLaMAv3

LLaMA 3 (Language Learning and Model Adaptation) is an advanced language model developed with particular attention to adaptability and the ability to learn efficiently from large amounts of data. It employs a deep neural network structure that better captures long-range dependencies in text, improving the model's ability to identify entities even in complex and ambiguous contexts. LLaMAv3 was trained on a vast and heterogeneous text corpus, providing it with extensive and detailed knowledge of various entities and their relations.

5.2. LangChain

LangChain is a framework designed to build applications based on large language models (LLMs), such as GPT or LLaMA, combining them with other external data sources and structured chains of operations. Its main purpose is to simplify the development of complex pipelines and processes involving LLMs, enhancing the ability of these language models to work effectively with information from diverse sources.

5.3. Neo4j

Neo4j is a graph database specialized in storing, managing, and querying data that naturally lends itself to a graph representation. Unlike traditional relational databases, which use tables and key-based relations, Neo4j employs nodes, relations, and properties to represent and manage data. This makes it particularly effective in scenarios where the connections between data are complex and crucial for analysis.

5.4. Memory-Efficient LLM Finetuning

For the NER task, we performed supervised fine-tuning of the LLaMA model. Given the available hardware, equipped with a 16-core CPU, an NVIDIA GPU, and 64 GB of RAM, Parameter-Efficient Fine-Tuning (PEFT) [23] techniques were employed to optimize training efficiency while reducing memory and computational requirements. PEFT is a set of techniques designed to fine-tune LLMs by updating only a small subset of the model's parameters, while keeping the pre-trained knowledge of the model intact. Among the most common techniques included in PEFT is LoRA (Low Rank Adaptation) [24]. LoRA approximates weight updates as a low-rank matrix, drastically reducing the number of parameters that need to be updated.

5.5. Pipeline Configuration

The information extraction pipeline described in the previous sections uses text chunking with the *RecursiveCharacterTextSplitter* method. This approach allows the text to be divided into manageable segments while preserving local semantic context. Recursive splitting avoids breaking the text at critical points, such as sentence endings or keywords, ensuring that entities and relations are accurately extracted without losing the logical flow across segments. The chosen chunk_size is set to 1200 characters, while the *chunk_overlap* parameter is set to 0, as the model performing coreference resolution will also receive contextual information.

LLaMA, in its 3.1 version, is used with a model size of 8 billion parameters (8B). This version represents a significant advancement over previous ones, thanks to improvements in capturing long-term dependencies in text and handling complex contexts. The model is loaded with specific configurations that employ 4-bit quantization (NF4) along with float16 precision for computations, aiming to achieve an optimal balance between performance and resource consumption.

Regarding the Bulletin Splitting step, pairs of bulletins are provided as input to the relation extraction stage. Indeed, even in this case, providing a large number of bulletins to the model carries the risk of some information being overlooked. Therefore, this choice was considered a good compromise between performance and effectiveness.

6. Security Considerations

The Continuous Integration (CI) process consists of frequently integrating software code into a shared repository, enabling automated build and test mechanisms. This process must be complemented by the concept of Continuous Delivery / Continuous Deployment (CD), which encompasses the set of mechanisms that automate the software release phase. Built upon this concept, the DevSecOps paradigm integrates security into this process by introducing automated security tests in the CI phase, enabling SAST, secret scanning, and dependency scanning, and in the CD phase through the execution of DAST analyzes and security checks on the deployment environment.

To ensure security requirements, the software was developed following the DevSecOps paradigm using GitLab. Specifically, given the sensitivity of the data being processed, the GitLab environment was configured locally, adopting an on-premises approach that prevents the exposure of confidential information on public cloud infrastructures.

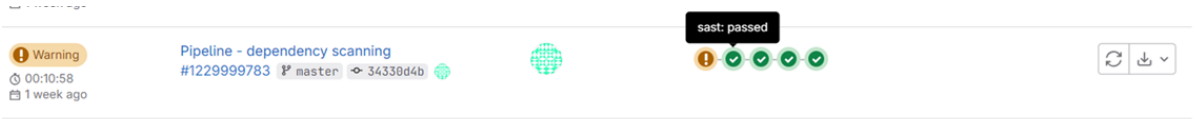

Figure 5: DevSecOps Pipeline

The implemented pipeline of figure 5 consists of five stages:

- Test: this stage performs dependency scanning on the projects that compose the framework, in order to identify vulnerable or outdated libraries.
- SAST: in this phase, static code analysis tools are executed to detect vulnerabilities directly in the source code before the build phase.
- Build: this phase compiles the various components of the framework and pushes them to a private repository.
- DAST: this stage uses dynamic analysis tools to identify vulnerabilities that can be detected during software execution.
- Container Scan: it performs a scan of the generated container image, which will host one or more components, with the goal of reducing the attack surface by detecting insecure configurations or vulnerable dependencies.

7. Upon an Investigative Application of the Framework

A case study approach was adopted to evaluate the effectiveness of the developed framework. The pipeline described was provided as input with the interrogation report No. 107/1998 R.g.n.r., a document of 10 pages of unstructured text, made available by the National Anti-Mafia and Anti-Terrorism Directorate and produced on January 22, 1998 by the District Prosecutor's Office at the Court of Catania.

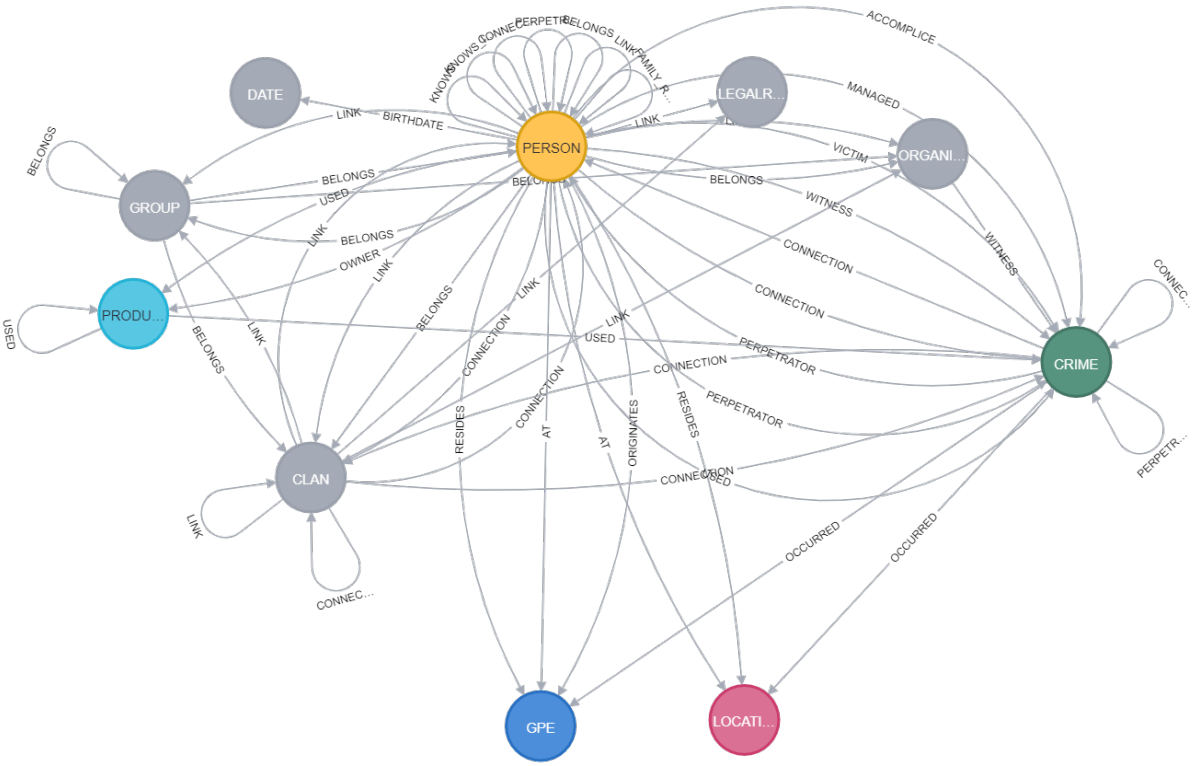

Figure 6: Graph view of the generated nodes in neo4j

Crime
Robbery at a fuel station
Investigation
Ambush
Suppression
Extortion
Crime
Robbery at a vestro postal market
Cocaine trafficking
Robbery
Homicide
Crime

Table 4
List of crimes extracted from Neo4J

The processing of this document resulted in the extraction of 120 nodes and 393 relations. Figure 6 shows the schema extracted from Neo4j.

As an example, table 4 shows the list of crimes recognized in the document under analysis. The advantage of using graph databases such as Neo4j is mainly related to the ecosystem built around them. In fact, by using the Apoc library, for instance, it is possible to leverage node-distance metrics to obtain a clearer understanding of nodes that could potentially be merged into a single super-node.

Suppose we want to search the knowledge base for all nodes belonging to a set of types (for example, Persons, Locations, and Criminal Clans) that are connected by a relationship to a node of type Crime with the value “Robbery”. The following Cypher query[1] is shown:

[1]https://neo4j.com/docs/cypher-manual/

```
MATCH (p:PERSON|LOCATION|GPE|CLAN|GROUP)-[r]-(c:CRIME {name: 'RAPINA'})
RETURN p, r, c;
```

The use of Neo4J in these investigative scenarios offers many advantages, both for the intuitive graph-based representation when dealing with complex relations between entities, as is the case for the DNAA, and for enabling investigators to identify hidden patterns that would be difficult to detect using relational databases.

8. Conclusions

This study set out to develop a framework for supporting the investigative activities of the Direzione Nazionale Antimafia e Antiterrorismo (National Directorate for Anti-Mafia and Counter-Terrorism, DNAA). This work once again confirms the applicability of LLMs for information extraction tasks, particularly in complex, domain-specific contexts such as the legal domain. While many studies have explored the use of large closed-weight GPT-like models for these tasks, our results demonstrate that even smaller models can effectively process complex, heterogeneous texts and transform them into structured relational graphs. The framework was designed with stringent privacy constraints in mind, showing that advanced AI techniques can be applied without compromising data confidentiality. The findings highlight the potential of LLMs to enhance intelligence analysis in closed-source intelligence (CSINT) scenarios. Despite its exploratory nature, this work offers some insight into the application of open-weights LLMs to extract structured information from unstructured and domain-specific text. Several questions still remain to be answered. Further research is required to better understand the limitations of these models, with particular attention to the phenomenon of hallucinations. Another important aspect that needs to be addressed is the ability of LLMs to fully leverage the provided context. Future work should also investigate the effects of different prompting strategies on context utilization. Moreover, more specialized open-weights LLMs should be made available to researchers, especially for domains with similar complexity and privacy requirements like the investigative legal domain in which the DNAA operates.

Acknowledgments

This work has been funded by the project NextGenerationEU via PNRR - DM 352 (CUP: E66G22000400009). Furthermore, the author Mario Varlese holds a PhD scholarship assigned through Ministerial Decree no. 118 of 2 March 2023, within the framework of the National Recovery and Resilience Plan (NRRP) funded by the European Union — NextGenerationEU. His PhD belongs to the "Public Administration PhD" category and is supported by *"Direzione Nazionale Antimafia e Antiterrorismo"* (DNAA) (CUP: E66E23001050002).

Declaration on Generative AI

During the preparation of this work, the author(s) used Gemini to: Grammar and spelling check, Paraphrase and reword. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] J. Cowie, W. Lehnert, Information extraction, Communications of the ACM 39 (1996) 80–91.
[2] R. Grishman, Information extraction, IEEE Intelligent Systems 30 (2015) 8–15.
[3] M. A. Hedderich, L. Lange, H. Adel, J. Strötgen, D. Klakow, A survey on recent approaches for natural language processing in low-resource scenarios, in: Proceedings of the 2021 Conference of

the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 2545–2568.

[4] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, et al., A survey of large language models, arXiv preprint arXiv:2303.18223 1 (2023).

[5] Z. Zhang, W. You, T. Wu, X. Wang, J. Li, M. Zhang, A survey of generative information extraction, in: O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, S. Schockaert (Eds.), Proceedings of the 31st International Conference on Computational Linguistics, Association for Computational Linguistics, Abu Dhabi, UAE, 2025, pp. 4840–4870. URL: https://aclanthology.org/2025.coling-main.324/.

[6] S. Deng, Y. Ma, N. Zhang, Y. Cao, B. Hooi, Information extraction in low-resource scenarios: Survey and perspective, in: 2024 IEEE International Conference on Knowledge Graph (ICKG), 2024, pp. 33–49. doi:10.1109/ICKG63256.2024.00013.

[7] S. Ramaswamy, R. Sreelekshmi, G. Veena, Complexity analysis of legal documents, in: International Conference on Artificial Intelligence on Textile and Apparel, Springer, 2023, pp. 141–154.

[8] F. Shahid, M.-H. Hsu, Y.-C. Chang, W.-S. Jian, Using generative ai to extract structured information from free text pathology reports, Journal of Medical Systems 49 (2025) 36.

[9] L. Wang, J. Chou, A. Tien, X. Zhou, D. Baumgartner, Aviationgpt: A large language model for the aviation domain, in: AIAA AVIATION FORUM AND ASCEND 2024, 2024, p. 4250.

[10] A. Zhukova, F. Hamborg, B. Gipp, Anea: Automated (named) entity annotation for german domain-specific texts, arXiv preprint arXiv:2112.06724 (2021).

[11] D. Xu, W. Chen, W. Peng, C. Zhang, T. Xu, X. Zhao, X. Wu, Y. Zheng, Y. Wang, E. Chen, Large language models for generative information extraction: A survey, Frontiers of Computer Science 18 (2024) 186357.

[12] Y. Park, R. S. Park, H. Kim, Key information extraction for crime investigation by hybrid classification model, Electronics 13 (2024) 1525.

[13] N. Raj, S. Thomas, et al., Open information extraction system for extracting relations in legal documents, in: 2022 IEEE 3rd global conference for advancement in technology (GCAT), IEEE, 2022, pp. 1–8.

[14] Y. Chen, X. Yuan, Z. Zhang, Extraction of legal document elements based on lstm, in: 2025 3rd International Conference on Data Science and Network Security (ICDSNS), 2025, pp. 1–5. doi:10.1109/ICDSNS65743.2025.11168670.

[15] J. Li, Y. Yang, Y. Bai, X. Zhou, Y. Li, H. Sun, Y. Liu, X. Si, Y. Ye, Y. Wu, Y. Lin, B. Xu, B. Ren, C. Feng, Y. Gao, H. Huang, Fundamental capabilities of large language models and their applications in domain scenarios: A survey, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 11116–11141. URL: https://aclanthology.org/2024.acl-long.599/. doi:10.18653/v1/2024.acl-long.599.

[16] J. A. Diaz-Garcia, J. A. D. Lopez, A survey on cutting-edge relation extraction techniques based on language models, Artificial Intelligence Review 58 (2025) 287.

[17] G. Coelho, A. Celecia, J. de Sousa, M. Lemos, M. Lima, A. Mangeth, I. Frajhof, M. Casanova, Information extraction in the legal domain: Traditional supervised learning vs. chatgpt, in: Proceedings of the 26th International Conference on Enterprise Information Systems, volume 1, 2024, pp. 579–586.

[18] A. Kwak, C. Jeong, G. Forte, D. Bambauer, C. Morrison, M. Surdeanu, Information extraction from legal wills: How well does GPT-4 do?, in: H. Bouamor, J. Pino, K. Bali (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2023, Association for Computational Linguistics, Singapore, 2023, pp. 4336–4353. URL: https://aclanthology.org/2023.findings-emnlp.287/. doi:10.18653/v1/2023.findings-emnlp.287.

[19] A. Nazarenko, A. Wyner, Legal nlp introduction, Traitement automatique des langues 58 (2017) 7–19.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[21] J. Gu, Z. Han, S. Chen, A. Beirami, B. He, G. Zhang, R. Liao, Y. Qin, V. Tresp, P. Torr, A systematic survey of prompt engineering on vision-language foundation models, arXiv preprint arXiv:2307.12980 (2023). URL: https://arxiv.org/abs/2307.12980.

[22] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, P. Liang, Lost in the middle: How language models use long contexts, Transactions of the Association for Computational Linguistics 12 (2024) 157–173. URL: https://aclanthology.org/2024.tacl-1.9/. doi:10.1162/tacl_a_00638.

[23] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, S. Gelly, Parameter-efficient transfer learning for nlp, 2019. arXiv:1902.00751.

[24] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al., Lora: Low-rank adaptation of large language models., ICLR 1 (2022) 3.