

Inside CORE-KG: Evaluating Structured Prompting and Coreference Resolution for Knowledge Graphs

Dipak Meher
 Department of Computer Science
 George Mason University
 Fairfax, USA
 dmeher@gmu.edu

Carlotta Domeniconi
 Department of Computer Science
 George Mason University
 Fairfax, USA
 cdomenic@gmu.edu

Abstract—Human smuggling networks are increasingly adaptive and difficult to analyze. Legal case documents offer critical insights but are often unstructured, lexically dense, and filled with ambiguous or shifting references, which pose significant challenges for automated knowledge graph (KG) construction. While recent LLM-based approaches improve over static templates, they still generate noisy, fragmented graphs with duplicate nodes due to the absence of guided extraction and coreference resolution. The recently proposed CORE-KG framework addresses these limitations by integrating a type-aware coreference module and domain-guided structured prompts, significantly reducing node duplication and legal noise. In this work, we present a systematic ablation study of CORE-KG to quantify the individual contributions of its two key components. Our results show that removing coreference resolution results in a 28.32% increase in node duplication and a 4.32% increase in noisy nodes, while removing structured prompts leads to a 4.34% increase in node duplication and a 73.33% increase in noisy nodes. These findings offer empirical insights for designing robust LLM-based pipelines for extracting structured representations from complex legal texts.

Index Terms—Knowledge Graph Construction, Coreference Resolution, LLMs, Human Smuggling Networks

I. INTRODUCTION

Human smuggling has become a complex operation involving dynamic networks of actors, routes, vehicles, and intermediaries [1]. These networks facilitate illicit human mobility for payment, exploiting legal loopholes and adapting rapidly to enforcement efforts to avoid detection. Often linked to transnational crime, they pose serious security and humanitarian risks due to their covert, profit-driven nature. Understanding their structure is critical for shaping effective policy responses and disrupting illicit operations. However, much of the relevant intelligence remains locked within unstructured legal texts such as court rulings and case transcripts, making systematic analysis challenging.

Despite growing interest from legal and social science communities, computational approaches for analyzing such documents remain underdeveloped. Entity references in unstructured legal text are often inconsistent, appearing as varying coreferent mentions (e.g., “Defendant Lewis,” “smuggler,” “driver Lewis,” or simply “Lewis”), abbreviations, or role-based titles (e.g., “Officer Lewis” vs. “Defendant Lewis”).

All example names are fictitious and used solely to protect privacy.

These inconsistencies complicate coreference resolution, entity normalization, and downstream tasks such as information extraction and knowledge graph construction.

Prior work has shown the value of knowledge graphs in legal investigations. Mazepa et al. [2] and Shi et al. [3], for instance, used rule-based methods to build graphs from legal cases, but their reliance on static templates limits flexibility in handling aliasing and surface-level entity variation. Large language models (LLMs) have shown promise in knowledge extraction [4], yet their use in human smuggling case narratives remains limited. A major challenge is merging semantically equivalent mentions of typed entities in complex legal texts [5], which, if left unresolved, creates fragmented and redundant graphs. Additionally, LLM-based extractors often misclassify or hallucinate entities and relations [6], adding noise that hinders downstream analysis.

To address these challenges, we introduced CORE-KG [7], a modular framework that builds clean and interpretable knowledge graphs from legal case documents using large language models. CORE-KG combines two main components: (1) a type-aware coreference resolution module that sequentially resolves contextually similar mentions within each entity type, and (2) a structured prompting strategy that guides the model to extract relevant entities and relationships while filtering legal boilerplate and reducing ambiguity. Compared to a GraphRAG-based baseline, CORE-KG reduces node duplication by 33.28% and legal noise by 38.37%, showing strong improvements in graph quality.

In this paper, we conduct an ablation study on CORE-KG to quantify the individual contributions of its two key components: type-aware coreference resolution and structured prompting. In the first setting (*CoreKG-no-coref*), we disable the coreference module while keeping structured prompts to assess its role in entity consolidation. In the second (*CoreKG-no-structprompts*), we retain coreference resolution but remove structured prompts to evaluate their impact on reducing noise and improving extraction precision. We adopt the same setup as CORE-KG, using legal case documents related to human smuggling from publicly available U.S. federal and state court proceedings. Since no gold-standard graphs exist, we compare the outputs of each ablation with CORE-KG and a GraphRAG-based baseline [8], both of which generate knowledge graphs

from raw legal text using large language models.

Our ablation study reveals that removing either coreference resolution or structured prompts significantly degrades KG quality. Specifically, removing coreference resolution (*CoreKG-no-coref*) results in a 28.32% increase in node duplication and a 4.32% increase in noisy nodes. In contrast, removing structured prompts (*CoreKG-no-structprompts*) leads to a 73.33% increase in noisy nodes and a 4.34% increase in node duplication. On average, this corresponds to a 38.83% degradation in noise reduction and a 16.33% degradation in entity resolution, underscoring the necessity of both modules for accurate and interpretable knowledge graph construction.¹ To summarize, our key contributions are:

- We present the first systematic ablation study of the CORE-KG framework, isolating the effects of its two central components: coreference resolution and structured prompting.
- We quantitatively show that removing the coreference module results in a 28.32% increase in node duplication and 4.32% increase in noise, while removing structured prompting leads to a 4.34% increase in duplication and a 73.33% spike in noise, demonstrating the distinct and complementary impact of each component.
- Our findings empirically validate that both components are essential and complementary for constructing accurate, interpretable knowledge graphs from unstructured legal texts, offering design guidance for future LLM-based KG pipelines.

II. RELATED WORK

Knowledge graphs enable structured representation of unstructured text for tasks like reasoning, retrieval, and analysis [8]. They have seen wide adoption across domains including education [9], life sciences [10], and construction safety [11]. However, poorly constructed graphs often exhibit node duplication and structural fragmentation, which hinder their effectiveness [12].

A. Knowledge Graph Construction

Traditionally, knowledge graphs are constructed by extracting key entities and their relationships from text and linking them sequentially. Entity extraction relies on rule-based, statistical, and domain-specific methods [13], while relation extraction uses syntactic, lexical, and semantic features [14], ontology-based systems like HowNet [15], and semi-supervised techniques [16]. Though effective in narrow domains, these approaches often rely on hand-crafted rules and focus on maintaining close alignment with the source text.

More recently, LLM-based frameworks now enable prompt-driven knowledge graph construction with minimal manual effort [6], [17]. Kommineni et al. [18] combined competency question generation, ontology design, and RAG-based triple extraction, while Zhang et al. [6] proposed a modular pipeline

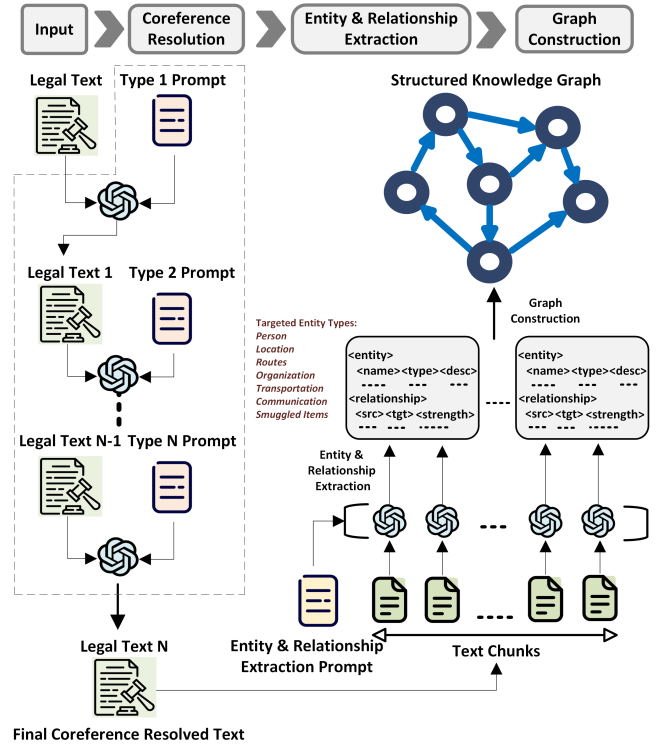


Fig. 1: Overview of the CORE-KG Pipeline. The pipeline begins with legal text inputs, processed via a type-aware coreference resolution module using sequential per-type prompting. The final resolved text is then passed through a structured prompting stage for entity and relationship extraction. The resulting triples are used to construct a coherent knowledge graph with significantly reduced duplication and noise.

with open IE, schema definition, and canonicalization. However, these methods often assume clean input and overlook challenges like reference ambiguity and entity aliasing, which are common in legal texts where entities shift across aliases, roles, and pronouns. To address these, coreference resolution is critical for maintaining graph coherence. Wang et al. [19] demonstrated that even sparse pronoun usage can impact graph completeness, underscoring the need to integrate coreference mechanisms into the construction pipeline.

Several works have explored knowledge graph construction in criminal domains. Mazepa et al. [2] used rule-based NLP pipelines and CoreNLP to build a homicide investigation graph. Shi et al. [3] employed regex-based extraction to construct a Neo4j graph for job-related crime indictments. While effective in legal contexts, both rely on static templates and lack coreference handling and modular prompting, limiting their robustness in complex narratives.

CORE-KG addresses this gap with a prompt-driven, entity-type-specific coreference resolution strategy tailored for modeling criminal networks. This design improves semantic precision and node coherence in knowledge graphs involving complex inter-entity relations such as migration routes and

¹Our code is available at <https://github.com/dipakmeher/corekg-ablation-study.git>.

procedural actors.

But to what extent do structured prompts and coreference resolution each contribute to graph quality in such complex domains? Our study aims to answer this through targeted ablations, isolating the individual and joint effects of these modules. By measuring changes in node duplication and legal noise, we empirically evaluate how each component supports accurate, interpretable knowledge graph construction.

B. Coreference Resolution

Several methods have been proposed to address node duplication and fragmentation via coreference resolution [20]. CNN-based models by Pogorilyy et al. [21] and Wu et al. [22] enhance semantic and syntactic pattern modeling.

Recent advances in large language models (LLMs) have advanced coreference resolution in low-resource settings, using prompt-based, few/zero-shot, and chain-of-thought strategies [23], [24]. However, these methods often target single entity types or general narratives without modeling type distinctions [23], [25]. While effective in focused contexts, their ability to generalize across diverse, role-shifting entity types in domain-specific texts remains underexplored.

Some progress has been made in coreference resolution for legal texts. Jia et al. [5] introduced a neural model combining ELMo, BiLSTM, and GCNs to resolve speaker-based coreference in court records. However, to the best of our knowledge, no prior work has examined coreference resolution in the legal domain using large language models. While instruction-tuned LLMs have shown competitive performance in coreference resolution [4], these studies are restricted to standard narrative data and do not address the challenges of multi-entity legal documents. This raises an open question: to what extent can LLM-based coreference resolution scale to complex, multi-entity legal texts, and how much does it contribute to improving knowledge graph quality?

III. METHOD

A. CORE-KG Overview

Figure 2a illustrates the complete CORE-KG pipeline. CORE-KG consists of two key components: (1) a *type-aware coreference resolution* module that consolidates semantically and contextually equivalent mentions within each entity type, and (2) a *knowledge graph construction* module that employs structured prompts to extract entities and relationships. These prompts incorporate domain-specific filtering instructions to suppress legal boilerplate, sequential type-wise extraction to minimize attention drift, and explicit type definitions to reduce classification ambiguity.

1) *Coreference Resolution*: We design a Coreference Resolution module that leverages the contextual reasoning capabilities of large language models (LLMs) [26], [27]. The model unifies disparate mentions under a canonical form (e.g., “Young” and “the defendant”), ensuring that all relational links converge on a single entity node. This improves structural consistency and enhances analytical clarity.

However, resolving all entity types simultaneously can dilute the model’s attention and cause type drift (incorrect category assignments) or feature entanglement (interference between features of different types) [28], [29]. For instance, the phrase “The Camp” might denote a temporary migrant holding site (*Location*) but may be misclassified as an *Organization* in ambiguous contexts. Similarly, “the van” might refer to either a transport vehicle (*Means of Transportation*) or a holding site (*Location*).

To mitigate these issues, we adopt a *type-wise sequential resolution strategy*, where the LLM resolves one entity type at a time. This reduces cross-type interference and allows more accurate coreference consolidation. The pipeline proceeds as follows:

- 1) The legal text is first passed to the LLM with a *Person*-focused prompt, linking mentions like “Young,” “the defendant,” and “the driver” to a canonical form.
- 2) The output is then processed with a *Location* prompt to resolve place-based mentions (e.g., “Laredo,” “Laredo Texas”).
- 3) This sequence continues for *Routes*, *Organization*, *Means of Transportation*, *Means of Communication*, and *Smuggled Items*.

This modular strategy enhances *semantic precision* and *structural fidelity* while aligning with criminal network modeling through type-limited resolution guided by domain expertise.

a) *Coreference Resolution Prompt Design*: Prompts are designed to generalize across diverse legal cases while capturing the variability in how entities appear in text. For instance, *Person* entities may include aliases, shortened names, or role-based mentions (e.g., “the driver”); *Location* entities may vary in granularity (e.g., “Laredo” vs. “Laredo, Texas”); *Routes* may be expressed using formal names or abbreviations (e.g., “Interstate 35” vs. “I-35”); and *Means of Transportation* often combine object references with ownership cues (e.g., “trailer” vs. “the defendant’s truck”).

To ensure consistency and precision, each prompt follows a structured format that includes a persona definition, clear task description, contextual information, entity-type-specific resolution rules, and few-shot examples. These components guide the LLM to resolve coreferences accurately without altering the input while accommodating the different ways each entity type is expressed in legal documents.

2) *Entity-Relationship Extraction and KG Construction*: The coreference-resolved text is fed into the Knowledge Graph Construction (KGC) module of the GraphRAG framework [8], which comprises: (1) a KGC component that extracts entities and relationships to form a structured graph, and (2) a retrieval module that uses this graph as an index for response generation. This work focuses on the KGC component, extended for processing legal case files on human smuggling.

Each document is split into overlapping 300-token chunks, passed to the LLM with a domain-optimized extraction prompt. The model returns entity–relationship triples, which

are aggregated across chunks. Post-processing merges entities by exact string and type match, and the graph is built using NetworkX. Outputs are saved in GraphML and Parquet formats for downstream analysis in tools like Gephi and the GraphRAG visualizer.

a) Prompt Tuning for Entity and Relationship Extraction:

The extraction prompt follows the same structured design as described in Section III-A1a, but is adapted for a single-pass setup required by GraphRAG’s KGC component. Unlike coreference resolution, which uses separate prompts per entity type, this unified prompt extracts all seven entity types and their relationships together. It includes the same key components, persona, task description, contextual framing, step-by-step instructions, output format, and few-shot examples, to ensure relevance and reduce noise.

Additional refinements are introduced to improve precision for human smuggling network analysis, as outlined below.

(1) Sequential Entity Extraction to Reduce Attention Spread:

GraphRAG’s default joint extraction can dilute model attention across entity types, leading to missed entities, misclassifications, or loss of fine-grained distinctions, issues common in legal texts with overlapping entity mentions [30]. To address this, we enforce a fixed extraction order: *Person* first, followed by *Location*, *Routes*, *Organization*, etc. The model extracts relationships only after completing all entity types in sequence. This reduces attention competition, improves entity precision, and ensures more complete and reliable knowledge graph construction.

(2) Filtering High-Frequency Irrelevant Entities: Legal texts often include frequently mentioned but non-critical entities, such as courts, juries, and appeals, typically extracted as *Organization* type. If retained, these inflate the graph with irrelevant nodes and distort relationship patterns. To prevent this, we add an explicit filtering step in the prompt. After extraction, the LLM removes all government-related entities based on predefined rules before generating the final output. This in-prompt filtering reduces noise, improves graph clarity, and eliminates the need for post-processing.

(3) Entity Type Definitions to Reduce Misclassification:

LLMs often misclassify entities due to overgeneralization e.g., labeling event names as *Location* just because they appear near geographic terms during training [31], [32]. This problem worsens in legal texts, where subtle context matters. To reduce such errors, we include clear definitions and examples for all seven entity types directly in the prompt. These definitions help the model distinguish between similar types like *Person*, *Organization*, and *Route*, leading to more accurate and context-aware extractions.

B. Ablation Study: Isolating the Impact of Coreference Resolution and Structured Prompts

To evaluate the individual contributions of the core components of CORE-KG, we design a focused ablation study that isolates the effects of (1) type-aware coreference resolution and (2) domain-specific structured prompts. While the full CORE-KG pipeline incorporates both modules, this study aims

to disentangle their respective influence on the quality of the resulting knowledge graphs and assess how each module contributes to cleaner, more complete, and more accurate representations of human smuggling networks.

We construct three experimental variants for comparison. The first is the Full CORE-KG pipeline, which includes both coreference resolution and structured prompts. The second variant, denoted as CoreKG-no-coref, removes the coreference module while retaining the structured prompt for entity and relationship extraction. In this setting, raw legal text is passed directly to the KGC module without resolving intra-document entity references. The third variant, CoreKG-no-structprompt, removes our structured extraction prompt and instead uses the default prompt from GraphRAG [8], while still applying coreference resolution. This allows us to isolate the contribution of structured prompt design while holding the input (coref-resolved text) constant.

Finally, we include the GraphRAG baseline, which uses neither coreference resolution nor structured prompts and instead relies on its default entity and relationship extraction settings, with only minimal adaptation through a few in-domain examples from human smuggling cases. Including GraphRAG allows us to anchor our ablation study to a standard LLM-based extraction pipeline. By comparing CoreKG-no-coref and CoreKG-no-structprompt against this baseline, we can directly quantify how each individual module (coreference resolution or structured prompting) improves graph quality over a non-specialized system, and how the full CORE-KG pipeline combines these gains to achieve the strongest performance.

IV. EXPERIMENTAL SETUP

We aim to answer the following research questions:

- RQ1: How much does the coreference resolution module contribute to reducing duplicate nodes and noisy entities in knowledge graphs constructed from human smuggling cases?
- RQ2: How much does the modified structured prompt contribute to reducing duplicate nodes and noisy entities in knowledge graphs?

A. Dataset

We follow the same dataset setup as the CORE-KG framework to ensure a consistent and fair comparison across system variants. The dataset consists of judicial case documents related to human smuggling, retrieved from the Nexis Uni academic search engine. These cases span U.S. federal and state court proceedings filed between 1994 and 2024, covering a diverse range of smuggling-related scenarios.

Since legal case files often contain lengthy procedural and statutory sections, we extract only the “Opinion” section from each case, which typically includes the most relevant factual narrative, such as involved individuals, routes, vehicles, and smuggled items. This section serves as the input for all graph construction experiments including GraphRAG (used as a

Case	Node Duplication												Noisy Nodes											
	GraphRAG			CoreKG-no-coref			CoreKG-no-str-prompt			CoreKG			GraphRAG			CoreKG-no-coref			CoreKG-no-str-prompt			CoreKG		
	Tot	Dup	Rate	Tot	Dup	Rate	Tot	Dup	Rate	Tot	Dup	Rate	Tot	Noisy	Rate	Tot	Noisy	Rate	Tot	Noisy	Rate	Tot	Noisy	Rate
Case 1	94	32	34.04	63	20	31.75	75	21	28	51	13	25.49	94	26	27.66	63	8	12.70	75	13	17.33	51	5	9.80
Case 2	86	24	27.91	48	11	22.92	81	15	18.52	41	5	11.9	86	28	32.56	48	2	4.17	81	30	37.04	42	0	0.00
Case 3	68	19	27.94	37	11	29.73	63	13	20.63	32	6	18.75	68	38	55.88	37	9	24.32	63	36	57.14	32	13	40.63
Case 4	99	30	30.3	71	24	33.8	84	26	30.95	63	19	30.16	99	12	12.12	71	3	4.23	84	13	15.48	63	3	4.76
Case 5	66	21	31.82	50	17	34	86	20	23.26	60	19	31.67	66	15	22.73	50	8	16.00	86	16	18.60	60	11	18.33
Case 6	83	33	39.76	59	19	32.2	76	12	15.79	32	4	12.5	83	9	10.84	59	9	15.25	76	12	15.79	32	4	12.50
Case 7	87	33	37.93	67	24	35.82	79	17	21.52	49	10	20.41	87	12	13.79	67	7	10.45	79	15	18.99	49	8	16.33
Case 8	60	19	31.67	29	7	24.14	40	10	25	22	5	22.73	60	17	28.33	29	11	37.93	40	11	27.50	22	5	22.73
Case 9	75	15	20	22	1	4.55	61	9	14.75	34	5	14.71	75	19	25.33	22	3	13.64	61	18	29.51	34	7	20.59
Case 10	76	18	23.68	29	7	24.14	65	12	18.46	23	4	17.39	76	40	52.63	29	14	48.28	65	45	69.23	23	10	43.48
Case 11	49	11	22.45	28	5	17.86	30	6	13	21	3	14.29	49	9	18.37	28	5	17.86	30	5	16.67	21	3	14.29
Case 12	68	20	29.41	46	8	17.39	63	13	20.63	57	15	26.32	68	10	14.71	46	4	8.70	63	9	14.29	57	5	8.77
Case 13	103	29	28.16	47	11	23.4	91	14	15.38	41	5	12.2	103	49	47.57	47	18	38.30	91	40	43.96	41	11	26.83
Case 14	55	13	23.64	36	7	19.44	37	4	10.81	28	1	3.57	55	10	18.18	36	4	11.11	37	5	13.51	28	3	10.71
Case 15	73	26	35.62	57	19	33.33	72	16	22.22	49	11	22.45	73	10	13.70	57	5	8.77	72	11	15.28	49	5	10.20
Case 16	77	26	33.77	42	10	23.81	95	27	28.42	51	15	29.41	77	17	22.08	42	6	4.29	95	32	33.68	51	7	13.73
Case 17	78	17	21.79	56	11	19.64	70	9	12.86	47	6	12.77	78	10	12.82	56	4	7.14	70	8	11.43	47	3	6.38
Case 18	71	19	26.76	45	9	20	57	11	19.3	33	6	18.18	71	25	35.21	45	11	24.44	57	24	42.11	33	5	15.15
Case 19	94	36	38.3	29	8	27.59	83	17	20.48	41	9	21.95	94	49	52.13	29	5	17.24	83	43	51.81	41	8	19.51
Case 20	81	37	45.68	56	25	44.64	61	22	36.07	49	22	44.9	81	26	32.10	56	7	12.50	61	17	27.87	49	9	18.37
Avg	77.15	23.9	30.53	45.85	12.4	26.01	68.45	14.7	21.15	41.25	9.05	20.27	77.15	21.55	27.43	45.85	7.15	17.37	68.45	20.15	28.86	41.25	6.25	16.65

TABLE I: Comparison of node duplication and legal noise across different extraction methods. The table includes two ablation settings derived from CORE-KG, and baselines (CORE-KG and GraphRAG). *Tot* indicates the total number of extracted entities; *Dup* denotes the number of duplicate or noisy entities; *Rate* represents the percentage of duplication or noise relative to the total. Bold values indicate the lowest Rate within each group of methods.

baseline in CORE-KG), the ablation variants, and the full CORE-KG pipeline.

For this study, we use the same 20 legal cases as in CORE-KG. Each case contains approximately 2000 words in the Opinion section and reflects the narrative complexity necessary for evaluating the impact of coreference resolution and structured prompting on knowledge graph quality.

B. Implementation

We use the LLaMA 3.3 70B model for both coreference resolution and knowledge graph construction, served locally via the Ollama framework with temperature set to zero to ensure deterministic outputs. All experiments are conducted on an NVIDIA A100 GPU with 80GB of memory.

The knowledge graph construction is based on the GraphRAG framework (v0.3.2), configured with overlapping 300-token chunks. Although the framework requires specifying an embedding model, such as `nomic-embed-text` in our case, we do not use embeddings in this work, as the graph construction process does not rely on them. The codebase is implemented in Python 3.12.

C. Baselines and Experimental Design

To evaluate the individual contributions of the two core modules in the CORE-KG pipeline—coreference resolution and structured prompting—we design an ablation study comparing four system variants:

- GraphRAG: A minimally adapted version of the GraphRAG framework [8], using its standard prompt. It is applied directly to the input text, without coreference resolution or modified structured prompts.

- CoreKG-no-coref: Applies the structured prompt from CORE-KG to the raw input text, without any coreference resolution. This variant isolates the impact of the modified structured prompt on knowledge graph construction.
- CoreKG-no-str-prompt: Applies the type-aware coreference resolution module but uses the standard GraphRAG prompt for entity and relationship extraction. This variant isolates the impact of coreference resolution alone.
- CORE-KG (Full): The complete pipeline with both modules enabled—type-aware coreference resolution followed by entity and relationship extraction guided by the modified structured prompt. This represents the end-to-end system evaluated in the original CORE-KG framework.

All systems use the same input documents, model (LLaMA 3.3 70B), and GraphRAG-based graph construction setup to ensure a fair comparison.

D. Evaluation Measures

Since no gold-standard annotations exist for knowledge graphs in this domain, we adopt a quantitative evaluation framework using two metrics: node duplication rate and noise rate. These metrics are computed for all system variants using the same 20 legal cases from the CORE-KG setup.

Node duplication rate measures the proportion of redundant entity nodes in the graph. Duplicate detection is performed in two stages: (1) intra-type fuzzy string matching using the `partial_ratio` function from the `RapidFuzz` library, with a similarity threshold of 75%; and (2) manual review by a domain expert to refine the clusters. Given a set of

duplicate clusters C_i , the duplication count is computed as $\sum_{C_i} (|C_i| - 1)$, and normalized by total node count.

Noise rate quantifies the percentage of extracted nodes that are irrelevant to smuggling network analysis, such as legal boilerplate terms (e.g., Court, Appeal Process, Judicial Proceedings). These are manually identified by a domain expert and reported as a proportion of total nodes.

In addition to these metrics, we perform a quality-to-analysis comparison by examining the actual graph structures across CORE-KG, CoreKG-no-coref, and CoreKG-no-str-prompt. This graph-level analysis provides qualitative insight into the nature of duplication, noise, connectivity, and semantic coherence, offering a deeper understanding of how different components impact overall graph quality.

V. RESULTS

Table I reports node duplication and noisy nodes statistics across 20 legal case graphs, comparing the outputs of four configurations: GraphRAG, CoreKG-no-coref (structured prompts without coreference resolution), CoreKG-no-str-prompt (coreference resolution with the standard GraphRAG prompt), and the full CoreKG pipeline.

A. RQ1: Impact of Coreference Resolution on Node Duplication and Noise

1) *Node Duplication*: As shown in Table II, removing the coreference resolution module leads to a notable degradation due to increased duplicate nodes. CoreKG-no-coref exhibits a duplication rate of 26.01%, compared to 20.28% in CORE-KG, reflecting a relative degradation of 28.25%. The GraphRAG baseline shows the highest duplication at 30.53%, underscoring the effectiveness of both coreference resolution and structured prompting in minimizing redundancy.

As shown in Table I, case-wise analysis reinforces this trend. In nearly all cases, CoreKG-no-coref exhibits higher duplication than CORE-KG. For example, in Case 6, the duplication rate rises from 6.25% (CORE-KG) to 35.82%, and in Case 13, from 3.57% to 19.44%. The only exceptions are Case 9, Case 12, Case 16, and Case 20, where CoreKG-no-coref reports slightly lower rates. However, these differences stem from minor fluctuations in the total number of nodes extracted by the LLM, which in turn reduce the observed duplication proportion. Overall, on average, CORE-KG maintains lower duplication rates compared to the ablation variants and GraphRAG.

TABLE II: Average node duplication and noise rates along with relative degradation compared to CORE-KG, computed over 20 legal cases

Method	Node Duplication (%)	Noisy Nodes (%)
GraphRAG	30.53% (+50.61%)	27.43% (+64.77%)
CoreKG-no-coref	26.01% (+28.25%)	17.37% (+4.32%)
CoreKG-no-structprompts	21.15% (+4.34%)	28.86% (+73.33%)
CORE-KG	20.28% (-)	16.65% (-)

In CoreKG-no-coref, the use of structured prompts alone reduces duplication compared to GraphRAG, as shown in

Table I. Specifically, the average node duplication rate decreases from 30.53% in GraphRAG to 26.01% in CoreKG-no-coref. This improvement stems from in-prompt filtering, sequential extraction steps, and the explicit definition of the seven entity types described in the Method section relevant to human smuggling analysis.

However, structured prompts cannot resolve referential variations on their own. Coreference resolution remains essential for merging such mentions. For instance, in Case 6 (Figure 2b), “Interstate Highway 35,” “Interstate,” and “I-35” all refer to the same route. Without coreference resolution, these are treated as distinct entities, as shown in Figure 2b. This results in a higher node duplication rate in CoreKG-no-coref (22.92%) compared to CORE-KG (11.90%).

2) *Noisy Nodes*: The removal of coreference resolution leads to a moderate increase in noisy nodes, rising from 16.65% in CORE-KG to 17.36% in CoreKG-no-coref, corresponding to a relative degradation of 4.26% (Table II). This shows that coreference resolution helps reduce noise, though its effect is smaller than on duplication. It refines the input text by consolidating scattered or semantically equivalent mentions that might otherwise increase text density, diffuse the LLM’s attention, and result in noisier extractions or misclassifications. While the improvement is modest, it contributes to overall graph cleanliness.

Even without coreference resolution, CoreKG-no-coref achieves comparatively low noise levels. This indicates that the structured prompt itself plays a key role in reducing noise by guiding the LLM to extract only relevant entity and relationship types. Through in-prompt filtering, sequential extraction, and predefined entity types tailored to human smuggling analysis, the structured prompt suppresses irrelevant or overly generic mentions. The sharp rise in noise from CoreKG-no-coref to GraphRAG (57.84%) further underscores the value of these structured instructions.

In 5 out of 20 cases, CoreKG-no-coref achieves noise rates comparable to or even lower than CORE-KG, suggesting that structured prompts alone can sometimes perform competitively. Nonetheless, the full CORE-KG pipeline consistently yields the lowest noise levels, reaffirming that both components are complementary and jointly essential for high-precision knowledge graph construction.

B. RQ2: Impact of CORE-KG’s Structured Prompt on Node Duplication and Noise

1) *Node Duplication*: Table I presents a comparison between CORE-KG, CoreKG-no-str-prompt, and the GraphRAG baseline. As shown in Table II, the removal of our structured prompt in CoreKG-no-str-prompt increases the node duplication rate from 20.27% (CORE-KG) to 21.15%, reflecting a relative degradation of approximately 4.3%. This highlights the role of CORE-KG’s structured prompt in guiding the LLM toward precise, domain-relevant entity extraction and in mitigating node duplication, even when coreference resolution is still present.

The structured prompt in CORE-KG employs sequential extraction and explicit type definitions to mitigate attention drift. This design prevents the LLM from attending to multiple entity and relationship types simultaneously, which can lead to over-extraction or conflated outputs. Without such structured guidance, as in CoreKG-no-str-prompt, the model may produce longer or imprecise spans that inflate the number of nodes. For example, in the sentence “The driver and the smuggler entered Texas with the group,” the model may extract relevant entities such as “the group” and “Texas,” but also over-extract the phrase “entered Texas with the group” as a separate entity. Although this phrase refers to the same underlying entities, its inclusion introduces duplication due to a loss of precision in entity boundaries, driven by attention drift.

Case-wise analysis further supports this conclusion. CoreKG-no-str-prompt shows higher duplication than CORE-KG in most cases. Notably, Case 06 demonstrates the largest increase, with node duplication rising from 6.25% in CORE-KG to 15.79% in CoreKG-no-str-prompt. This indicates that both structured prompting and coreference resolution are jointly necessary for achieving high-impact reductions in duplication.

However, there are six cases (Case 06, Case 12, Case 15, Case 16, Case 19, and Case 20) where CoreKG-no-str-prompt shows lower duplication than CORE-KG. This can be explained by two factors. First, even without structured prompting, the coreference module still mitigates some duplication by merging exact or near-exact mentions. Second, the absence of structured prompts leads to a surge in extracted nodes, many of which are noisy or semantically conflated. This broader and less focused extraction dilutes the proportion of detected duplicates, giving the appearance of a lower duplication rate despite the underlying graph being noisier. As shown in Table I, the average number of extracted nodes increases from 41.25 in CORE-KG to 68.45 in CoreKG-no-str-prompt, contributing to the observed variations in duplication percentages.

2) *Noisy Nodes*: CoreKG-no-str-prompt exhibits a substantial increase in the proportion of noisy nodes, rising from 16.65% in CORE-KG to 28.86%, a relative degradation of 73.33%, as shown in Table II. This spike is primarily due to the absence of structured prompting, which normally provides in-prompt filtering, sequential extraction, and explicit type definitions, mechanisms that help suppress noise during entity and relationship extraction. While the coreference resolution module in CoreKG-no-str-prompt continues to unify repeated or ambiguous legal mentions, its effectiveness is largely over-ridden by the surge in over-extracted and conflated nodes.

Interestingly, the average noisy node count in CoreKG-no-str-prompt (28.86) is nearly identical to that of GraphRAG (27.43), indicating that removing structured prompts nullifies many of the precision gains achieved through coreference resolution. This can be attributed to the fact that although CoreKG-no-str-prompt extracts fewer nodes overall (68.45 on average) compared to GraphRAG (77.15), it still results in a similar number of noisy extractions (20.15 vs. 21.55).

This suggests that while the coreference module helps reduce duplication and ambiguity, the lack of structured guidance leads to attention drift and over-extraction, thereby pushing the overall noise ratio close to the baseline.

In a head-to-head comparison, CoreKG-no-str-prompt shows a consistent rise in noisy nodes across all 20 legal cases compared to CORE-KG, clearly demonstrating the critical role of modified structured prompts in noise reduction within the CORE-KG pipeline.

C. Qualitative Analysis of Extracted Graphs

Figure 2 presents the generated graphs from CORE-KG, CoreKG-no-coref, and CoreKG-no-str-prompt for Case 6. This section analyzes the structural gains achieved through the incorporation of coreference resolution and structured prompts in the CORE-KG pipeline. For clarity, relationship labels are omitted in the visualizations, and node sizes are scaled based on their degree (i.e., the number of connected relationships).

1) *Analysis of Relationship-to-Node Ratio*: Table III presents the node count, relationship count, and relationship-to-node ratio (R/N) for graphs generated by CORE-KG, CoreKG-no-coref, and CoreKG-no-str-prompt. To assess the structural quality of these graphs, we compute the R/N ratio, which reflects how densely the nodes are interconnected.

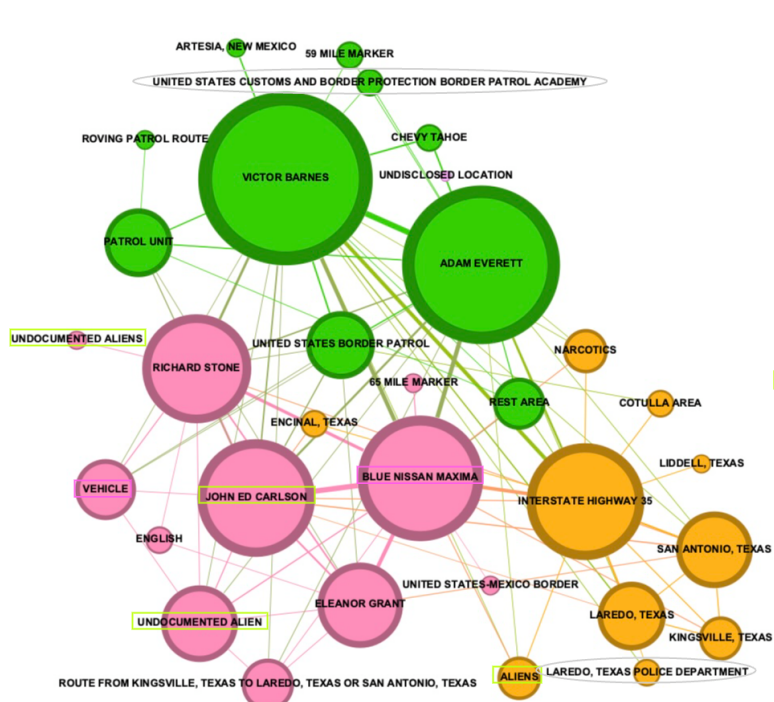
TABLE III: Graph Statistics for Case 6

Method	#Nodes	#Relationships	R/N Ratio
CORE-KG	32	92	2.88
CoreKG-no-coref	59	107	1.81
CoreKG-no-str-prompt	76	127	1.67

The graph generated by CoreKG-no-str-prompt contains the highest number of nodes and relationships, with 76 nodes and 127 edges. CoreKG-no-coref generates a moderately sized graph with 59 nodes and 107 edges, while CORE-KG produces a more compact structure containing 32 nodes and 92 edges. CORE-KG achieves the highest R/N ratio of 2.88, indicating a high degree of structural coherence. In contrast, CoreKG-no-coref and CoreKG-no-str-prompt yield lower ratios of 1.81 and 1.67, respectively. Although CoreKG-no-str-prompt produces more nodes, many appear weakly connected or isolated, suggesting increased noise. The higher density of the CORE-KG graph demonstrates the combined benefit of coreference resolution and structured prompting in producing a more informative and compact representation.

2) *Per-Case Graph Comparison and Structural Gains*: We further analyze the graph outputs by visually inspecting the impact of duplicate and noisy node generation. In Figure 2, duplicate nodes are marked using rectangles, while noisy nodes are enclosed in ovals. Duplicate nodes are colored identically to indicate that they represent the same entity but appear in multiple textual variations.

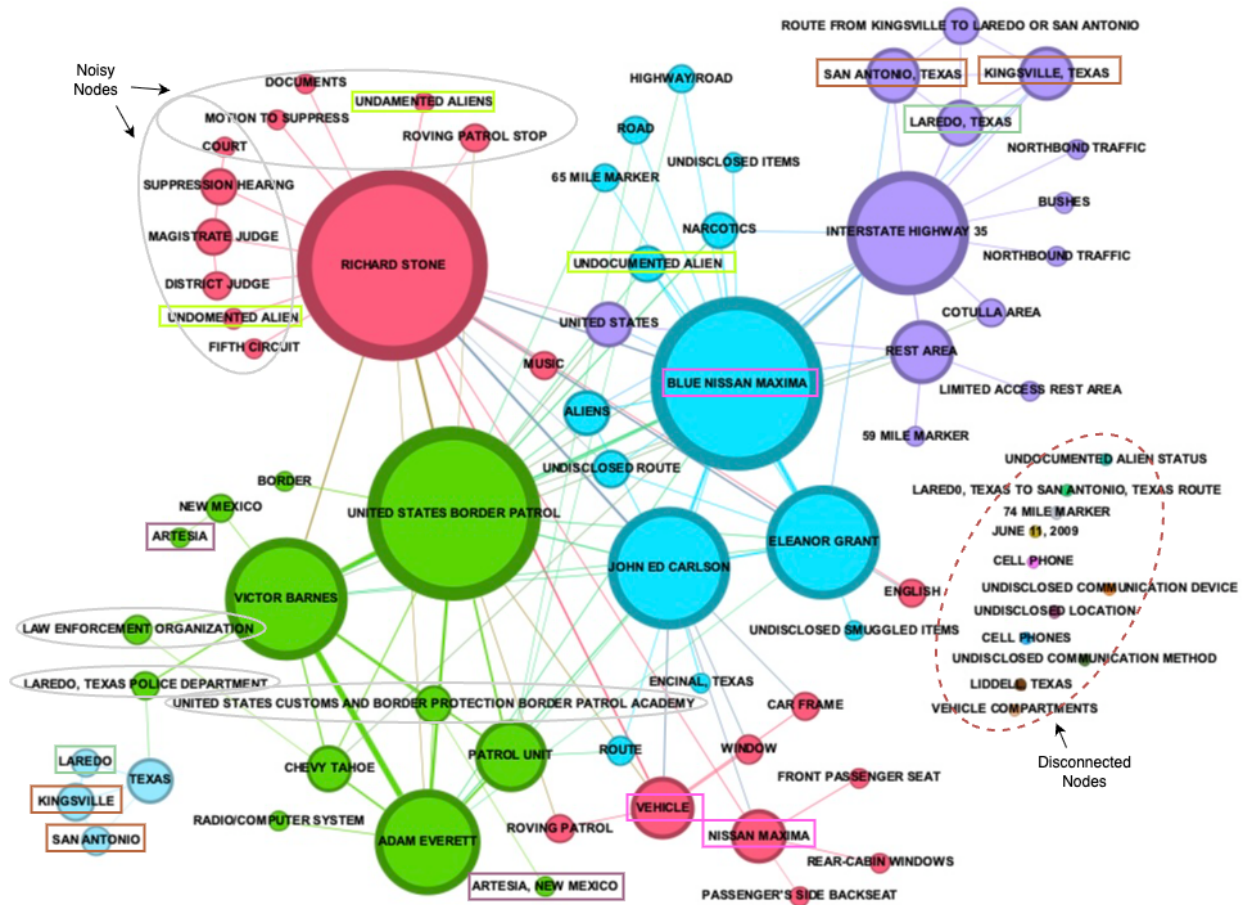
a) *Analysis of CoreKG-no-coref Graph*: Figure 2b shows the graph extracted using CoreKG-no-coref. As presented in Table I, CoreKG-no-coref produces a total of 45.85 nodes,



(a) CORE-KG Graph



(b) CoreKG-no-coref Graph



(c) CoreKG-no-str-prompt Graph

Fig. 2: Comparison across CORE-KG, CoreKG-no-coref, and CoreKG-no-str-prompt graphs on a sample legal case. Duplicate entities are highlighted with rectangles, noisy entities with ovals, and disconnected nodes with dashed ovals.

which is slightly higher than CORE-KG (41.25) and significantly lower than CoreKG-no-str-prompt (68.45). However, it exhibits a noticeably higher degree of duplication (26.01) compared to CORE-KG (20.27) and CoreKG-no-str-prompt (21.15). This is primarily due to the absence of the coreference resolution module, which fails to unify different surface forms of the same entity and thus causes the model to treat them as distinct nodes.

For example, the main defendant in this case, “Richard Stone,” appears in several variations such as “Defendant Stone” and “Stone.” Similarly, other person-type entities like “Eleanor Grant,” “Victor Barnes,” and “Adam Everett” are each represented as multiple nodes due to their duplicate variations. Additionally, “John Ed Carlson,” listed as an “Undocumented Alien,” also appears in a shortened form as “Carlson,” further contributing to node duplication. This issue is not limited to person-type entities. Route-type entities such as “Interstate Highway 35” are inconsistently represented as “I-35” and “Interstate.” Location-type entities like “San Antonio” appear both as “San Antonio, Texas” and simply “San Antonio.” Likewise, means of transportation such as “Nissan Maxima” are scattered across forms like “Maxima,” “Car,” and “Vehicle.” These duplications—among many others—reduce clarity and introduce structural redundancy in the graph, emphasizing the importance of applying coreference resolution prior to knowledge graph construction.

Regarding noise, CoreKG-no-coref performs better than CoreKG-no-str-prompt and is nearly comparable to CORE-KG. The use of modified structured prompts in CoreKG-no-coref helps filter out irrelevant entities and ensures more precise entity and relationship extraction. However, some entities remain disconnected from the main graph, as shown in Figure 2b. This is primarily due to the presence of multiple surface forms of the same entity in the text, which hinders proper merging during post-extraction alignment. Entities and relationships are typically merged based on matching names and types; however, when a location such as “Antonio” appears as a variation of “San Antonio,” the system fails to unify them, resulting in disconnected nodes.

Another contributing factor is attention drift caused by these multiple surface variations. Legal texts are inherently dense and contain numerous entities within short spans of text. The presence of variant forms dilutes the model’s attention, impacting the quality of entity and relationship extraction and increasing the likelihood of unconnected or missed nodes. For instance, the “65 mile marker” is properly linked to the main graph in CORE-KG (Figure 2a) and CoreKG-no-str-prompt (Figure 2c), but is absent in CoreKG-no-coref (Figure 2b).

b) Analysis of CoreKG-no-str-prompt Graph: Analyzing the CoreKG-no-str-prompt graph, we observe that it performs better than CoreKG-no-coref in terms of duplication. This improvement can be attributed to the presence of the coreference resolution module, which effectively unified surface variations early in the pipeline. However, a few duplicate entities still persist, such as “Artesia, New Mexico” and “Artesia,” “San Antonio” and “San Antonio, Texas,” “Undocumented

Alien” and “Undocumented Aliens,” as well as “Blue Nissan Maxima,” “Nissan Maxima,” and “Vehicle.” As discussed in Section V-B, this duplication is likely due to the use of the standard GraphRAG prompt, which suffers from attention drift and consequently leads to the extraction of irrelevant entities, further contributing to duplication.

However, CoreKG-no-str-prompt suffered significantly from noisy entity extraction due to the use of GraphRAG’s standard prompt, which lacks the structured guidance implemented in CORE-KG. As shown in Figure 2c, multiple government-related entities are unnecessarily extracted around nodes like Richard Stone” and Victor Barnes.” Additionally, several fine-grained but irrelevant entities appear, such as Window” and Car Frame,” both referring to parts of the “Nissan Maxima.” These cases illustrate the importance of in-prompt filtering, sequential extraction, and clearly defined entity types, all key components of CORE-KG’s structured prompt.

The standard GraphRAG prompt, by contrast, lacks structured mechanisms and therefore struggles with precise extraction. As previously discussed, this leads to issues such as attention drift, misallocation of focus, and noisy or incomplete outputs, especially in densely packed legal texts. These problems manifest in the CoreKG-no-str-prompt graph through unconnected and irrelevant nodes. CORE-KG’s structured prompt effectively addresses these challenges, resulting in cleaner and more coherent graphs.

One notable observation is that the CoreKG-no-str-prompt graph captures several fine-grained entities that provide valuable context for analyzing the operational nuances of human smuggling networks. For example, entities like “Passenger Side Backseat,” where the undocumented alien Carlson was seated, and “Bushes,” which concealed Border Patrol Agents along Interstate 35, offer spatial and situational details that enhance the understanding of smuggling tactics. This suggests the need to balance filtering out irrelevant fine-grained entities while retaining those that contribute to deeper analysis.

c) Analysis of CORE-KG Graph: Analyzing the CORE-KG graph reveals a well-connected structure with minimal duplication and noise, enabled by the integration of coreference resolution and modified structured prompts. Although some duplicate nodes persist, such as “Blue Nissan Maxima” and “Vehicle” or “Undocumented Alien” and “Aliens,” their frequency is significantly lower compared to CoreKG-no-coref and CoreKG-no-str-prompt. Similarly, a few government-related noisy nodes, including “United States Customs and Border Protection Border Patrol Academy,” are present but considerably fewer than in the two ablation variants. As a result, CORE-KG produces cleaner and more interpretable graphs that better support downstream analysis.

VI. CONCLUSION

In this work, we conducted a systematic ablation study of CORE-KG to quantify the individual contributions of its two key components: type-aware coreference resolution and domain-guided structured prompts. By selectively disabling

each module, we measured how their absence affects knowledge graph quality in terms of node duplication and legal noise. Removing coreference resolution increases node duplication by 28.32% and noisy nodes by 4.32%, while removing structured prompts increases node duplication by 4.34% and noisy nodes by 73.33%. These findings demonstrate that both modules are essential and complementary. Structured prompts play a dominant role in improving extraction precision, while coreference resolution enhances entity consolidation by addressing referential ambiguity. This clear separation in impact highlights the value of modular LLM pipelines that explicitly target different error modes. Future work may explore dynamic prompt adaptation, examine how cleaner KGs improve retrieval accuracy and answer coherence in downstream RAG systems, and investigate cross-case meta-coreference to construct unified super-graphs for tracking actors and patterns across legal documents.

ACKNOWLEDGMENT

This material is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number 17STCIN00001-08-00. Disclaimer: The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

REFERENCES

- [1] S. Carrasco-Granger, R. Bermejo-Casado, and I. Bazaga Fernández, “Scapegoating human smugglers: How migrants’ accounts challenge the eu’s dominant narrative,” *Journal of Immigrant & Refugee Studies*, pp. 1–13, 2025.
- [2] S. Mazepa, V. Vysotska, D. Ivanchyshyn, L. Chyrun, V. Schuchmann, and Y. Ryshkovets, “Relationships knowledge graphs construction between evidence based on crime reports,” in *2022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT)*. IEEE, 2022, pp. 165–171.
- [3] Y. Shi, W. An, J. Xue, and Y. Qu, “A knowledge graph constructed for job-related crimes,” *Procedia Computer Science*, vol. 199, pp. 540–547, 2022.
- [4] N. T. Le and A. Ritter, “Are large language models robust coreference resolvers?” *arXiv preprint arXiv:2305.14489*, 2023.
- [5] D. Ji, J. Gao, H. Fei, C. Teng, and Y. Ren, “A deep neural network model for speakers coreference resolution in legal texts,” *Information Processing & Management*, vol. 57, no. 6, p. 102365, 2020.
- [6] B. Zhang and H. Soh, “Extract, define, canonicalize: An llm-based framework for knowledge graph construction,” *arXiv preprint arXiv:2404.03868*, 2024.
- [7] D. Meher, C. Domeniconi, and G. Correa-Cabrera, “LLM-driven knowledge graph construction for human smuggling networks,” in *The First Structured Knowledge for Large Language Models Workshop*, 2025.
- [8] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, D. Metropolitan, R. O. Ness, and J. Larson, “From local to global: A graph rag approach to query-focused summarization,” *arXiv preprint arXiv:2404.16130*, 2024.
- [9] P. Chen, Y. Lu, V. W. Zheng, X. Chen, and B. Yang, “Knowedu: A system to construct knowledge graph for education,” *Ieee Access*, vol. 6, pp. 31 553–31 563, 2018.
- [10] T. J. Callahan, I. J. Tripodi, A. L. Stefanski, L. Cappelletti, S. B. Taneja, J. M. Wyrwa, E. Casiraghi, N. A. Matentzoglou, J. Reese, J. C. Silverstein *et al.*, “An open source knowledge graph ecosystem for the life sciences,” *Scientific Data*, vol. 11, no. 1, p. 363, 2024.
- [11] W. Fang, L. Ma, P. E. Love, H. Luo, L. Ding, and A. Zhou, “Knowledge graph for identifying hazards on construction sites: Integrating computer vision with ontology,” *Automation in Construction*, vol. 119, p. 103310, 2020.
- [12] E. Huaman, E. Kärle, and D. Fensel, “Duplication detection in knowledge graphs: Literature and tools,” *arXiv preprint arXiv:2004.08257*, 2020.
- [13] P. Sun, X. Yang, X. Zhao, and Z. Wang, “An overview of named entity recognition,” in *2018 International Conference on Asian Language Processing (IALP)*. IEEE, 2018, pp. 273–278.
- [14] N. Kambhatla, “Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction,” in *Proceedings of the ACL interactive poster and demonstration sessions*, 2004, pp. 178–181.
- [15] K. Liu, F. Li, L. Liu, and Y. Han, “Implementation of a kernel-based chinese relation extraction system,” *Jisuanji Yanjiu yu Fazhan (Computer Research and Development)*, vol. 44, no. 8, pp. 1406–1411, 2007.
- [16] A. Carlson, J. Betteridge, R. C. Wang, E. R. Hruschka Jr, and T. M. Mitchell, “Coupled semi-supervised learning for information extraction,” in *Proceedings of the third ACM international conference on Web search and data mining*, 2010, pp. 101–110.
- [17] J. Vizcarra, S. Haruta, and M. Kurokawa, “Representing the interaction between users and products via llm-assisted knowledge graph construction,” in *2024 IEEE 18th International Conference on Semantic Computing (ICSC)*. IEEE, 2024, pp. 231–232.
- [18] V. K. Kommineni, B. König-Ries, and S. Samuel, “From human experts to machines: An llm supported approach to ontology and knowledge graph construction,” *arXiv preprint arXiv:2403.08345*, 2024.
- [19] T. Wang and H. Li, “Coreference resolution improves educational knowledge graph construction,” in *2020 IEEE International Conference on Knowledge Graph (ICKG)*. IEEE, 2020, pp. 629–634.
- [20] R. Liu, R. Mao, A. T. Luu, and E. Cambria, “A brief survey on recent advances in coreference resolution,” *Artificial Intelligence Review*, vol. 56, no. 12, pp. 14 439–14 481, 2023.
- [21] S. Pogorilyy and A. Kramov, “Coreference resolution method using a convolutional neural network,” in *2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT)*. IEEE, 2019, pp. 397–401.
- [22] J.-L. Wu and W.-Y. Ma, “A deep learning framework for coreference resolution based on convolutional neural network,” in *2017 IEEE 11th International Conference on Semantic Computing (ICSC)*. IEEE, 2017, pp. 61–64.
- [23] M. Das and A. Senapati, “Co-reference resolution in prompt engineering,” *Procedia Computer Science*, vol. 244, pp. 194–201, 2024.
- [24] Y. Gan, J. Yu, and M. Poesio, “Assessing the capabilities of large language models in coreference: An evaluation,” in *Joint 30th International Conference on Computational Linguistics and 14th International Conference on Language Resources and Evaluation, LREC-COLING 2024*. European Language Resources Association (ELRA), 2024, pp. 1645–1665.
- [25] H.-D. Tran, D.-V. Nguyen, and N. L.-T. Nguyen, “Coreference resolution for vietnamese narrative texts,” *arXiv preprint arXiv:2504.19606*, 2025.
- [26] B. Wang, X. Yue, and H. Sun, “Can chatgpt defend its belief in truth? evaluating llm reasoning via debate,” *arXiv preprint arXiv:2305.13160*, 2023.
- [27] A. Havrilla, S. Raparthy, C. Nalmpantis, J. Dwivedi-Yu, M. Zhuravinskyi, E. Hambro, and R. Raileanu, “Glore: When, where, and how to improve llm reasoning via global and local refinements,” *arXiv preprint arXiv:2402.10963*, 2024.
- [28] W. Zhou, S. Zhang, Y. Gu, M. Chen, and H. Poon, “Universalner: Targeted distillation from large language models for open named entity recognition,” *arXiv preprint arXiv:2308.03279*, 2023.
- [29] S. Abdelnabi, A. Fay, G. Cherubin, A. Salem, M. Fritz, and A. Pavard, “Are you still on track!? catching llm task drift with activations,” *arXiv preprint arXiv:2406.00799*, 2024.
- [30] —, “Get my drift? catching llm task drift with activation deltas,” in *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*. IEEE, 2025, pp. 43–67.
- [31] U. Peters and B. Chin-Yee, “Generalization bias in large language model summarization of scientific research,” *Royal Society Open Science*, vol. 12, no. 4, p. 241776, 2025.
- [32] S. Dai, C. Xu, S. Xu, L. Pang, Z. Dong, and J. Xu, “Bias and unfairness in information retrieval systems: New challenges in the llm era,” in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 6437–6447.