Chapter 2: Literature Review

2.1 Introduction

The landscape of Artificial Intelligence (AI), particularly Natural Language Processing (NLP), was significantly reshaped by the groundbreaking work conducted at Google Brain and documented in the seminal paper *Attention Is All You Need* (Vaswani et al., 2017). This paper introduced the Transformer architecture, leveraging self-attention mechanisms, which became the foundation for modern Large Language Models (LLMs). These models have demonstrated remarkable capabilities across a wide range of tasks, including text generation, summarization, translation, and question answering, often producing outputs nearly indistinguishable from human writing (Badshah & Sajjad, 2025).

Despite these advancements, LLMs possess an inherent architectural limitation: a finite context window. This window represents the maximum amount of text (measured in tokens) that the model can process simultaneously when generating a response or performing an analysis. Consequently, if critical information or dependencies exist within a document but fall outside this fixed window—separated by a larger span of intervening text—the LLM may fail to capture the relationship or address the query accurately (Kaplan et al., 2025). This limitation poses a significant challenge when dealing with large or complex documents where understanding relies on synthesizing information across distant sections.

One promising approach to mitigate this limitation involves transforming large, unstructured documents into structured representations using Knowledge Graphs (KGs). By extracting key entities, relationships, and

attributes from the text and mapping them into a graph structure, it becomes possible to represent the document's core semantic content in a format amenable to computational analysis (Hogan et al., 2021). This allows for querying and reasoning over the entire document's scope, independent of the LLM's context window constraints, potentially enabling a more focused and comprehensive analysis for tasks such as ensuring information integrity.

This chapter reviews the pertinent literature underpinning this approach. It begins by examining the development and characteristics of Large Language Models, focusing on their capabilities and limitations, particularly the context window constraint. Subsequently, it delves into the principles, construction, and application of Knowledge Graphs as structured knowledge representations. Key techniques for populating KGs from text via Information Extraction are then discussed, followed by an exploration of the challenges associated with processing large and complex documents, especially within the legal domain. Finally, the chapter defines the critical concepts of Consistency, Completeness, and Coherence, particularly relevant for evaluating the integrity of document corpora like legal codes, and surveys related work before concluding with a summary motivating the proposed research direction.

2.2 Large Language Models

The trajectory of modern NLP took a significant turn in 2017 with the publication of *Attention Is All You Need* by Vaswani et al. (Vaswani et al., 2017). This work introduced the Transformer architecture, which uniquely relies on self-attention mechanisms to weigh the importance of different words (tokens) in the input sequence. This design enables superior handling

of long-range dependencies compared to previous dominant recurrent (like LSTMs) or convolutional architectures (Turner, 2025) and (Zhao et al., 2023), addressing critical bottlenecks present in earlier sequence models. This innovation paved the way for the development of increasingly large and powerful language models, such as Google's BERT, which introduced bidirectional pre-training (Koroteev, 2025), and OpenAI's influential Generative Pre-trained Transformer (GPT) series (Gao et al., 2025).

While research groups at numerous institutions continuously pursued improvements in model scale, training data, and architectural refinements, the public release of OpenAI's *ChatGPT* (based on the GPT-3.5 architecture) on November 30, 2022, marked a pivotal moment. This event dramatically increased public awareness and accelerated the development and deployment of advanced conversational AI systems across various sectors. It catalyzed the release and further development of competing models from major research labs, including Google's *Gemini* family of multimodal models (Team et al., 2024), Anthropic's safety-focused *Claude* series (Caruccio et al., 2024), and Meta's open-source *Llama* family, which has spurred significant community innovation (Grattafiori et al., 2024). The proliferation of models is evident on platforms like Hugging Face, a central repository for AI models and datasets, which reportedly surpassed one million hosted models by late 2024, reflecting the rapid pace of development in the field (Edwards, 2025).

Functionally, LLMs process input text (the "prompt") by first converting it into numerical representations called tokens, often using techniques like Byte Pair Encoding (BPE) or WordPiece (Schmidt et al., 2025). Using the complex patterns and linguistic knowledge learned during extensive pre-training on vast text corpora (often terabytes of data), the model then predicts

subsequent tokens autoregressively to generate a coherent and contextually relevant output. Prompts can be engineered to elicit specific behaviors or perform complex tasks, potentially including substantial amounts of text for analysis or context (in-context learning). For instance, an LLM might be prompted with a company's annual report and asked specific questions about its contents, or asked to summarize key findings. Many current LLMs can perform reasonably well on such tasks, provided the relevant information falls within their processing limits (Rzepka et al., 2023).

However, a fundamental limitation remains the context window size. This size, representing the maximum number of tokens the model can attend to simultaneously, while increasing with newer model generations (ranging from a few thousand in early models to potentially over a million tokens in recent research prototypes (Kaplan et al., 2025)), is always finite (Liu et al., 2025). If a document's length exceeds this limit, the LLM cannot process it in its entirety in a single pass. Standard techniques involve processing the document in overlapping or non-overlapping chunks (T. Chen et al., 2023), but this can sever long-distance contextual links crucial for deep understanding. For example, determining if a policy statement defined on page 1 of a lengthy legal code is adequately supported or subtly contradicted by detailed regulations presented hundreds of pages later might be impossible if the intervening text exceeds the context window. The LLM would process the sections independently, unable to synthesize the relationship between them effectively.

Furthermore, the computational cost of processing information within the context window remains a significant factor. The self-attention mechanism, core to the Transformer, typically scales quadratically ($O(n^2)$) with the sequence length (n) in terms of both computation and memory require-

ments (Minsky, 1974). While various "efficient Transformer" variants aim to reduce this to near-linear complexity (Tay et al., 2023), processing very long sequences up to the maximum context window still demands substantial memory (RAM/VRAM), processing power (CPU/GPU), and energy resources. This quadratic (or near-quadratic) scaling makes analyzing very large documents prohibitively expensive or slow for many practical applications, further motivating alternative approaches, such as KG-based structuring, for achieving comprehensive and efficient analysis.

2.3 Knowledge Graphs

Knowledge Graphs (KGs) provide a structured paradigm for representing information and knowledge, evolving from concepts in semantic networks, frame systems, and earlier AI research in symbolic knowledge representation (Hogan et al., 2021). Formally, a KG typically represents knowledge as a directed labeled graph, comprising a collection of interconnected entities (nodes or vertices) and the explicitly typed relationships (edges or links) between them. Both nodes and edges can possess attributes or properties (key-value pairs) that store additional metadata, context, or provenance information (Ehrlinger & W "o ss, 2016).

The core components of a KG are:

- **Nodes (Entities):** Represent real-world objects, abstract concepts, events, or specific instances of interest (e.g., persons like 'John Doe', organizations like 'Acme Corp', locations like 'West Chester, PA', legal statutes like '15 Pa.C.S.A. § 1502', defined terms like 'nonconforming use'). Nodes are often identified by unique identifiers (URIs or IRIs in RDF-based KGs).

- **Edges (Relationships):** Represent the connections or typed relationships between pairs of nodes (e.g., 'works for', 'located in', 'cites', 'amends', 'defines', 'has requirement'). Edges are typically directed (from a subject node to an object node) and labeled with the relationship type (predicate).

- **Attributes (Properties):** Key-value pairs associated typically with nodes (though sometimes edges in Property Graphs), providing additional details or literal values (e.g., a 'Person' node might have an 'email' attribute with value 'john.doe@example.com'; a 'cites' edge might have a 'citation date' attribute).

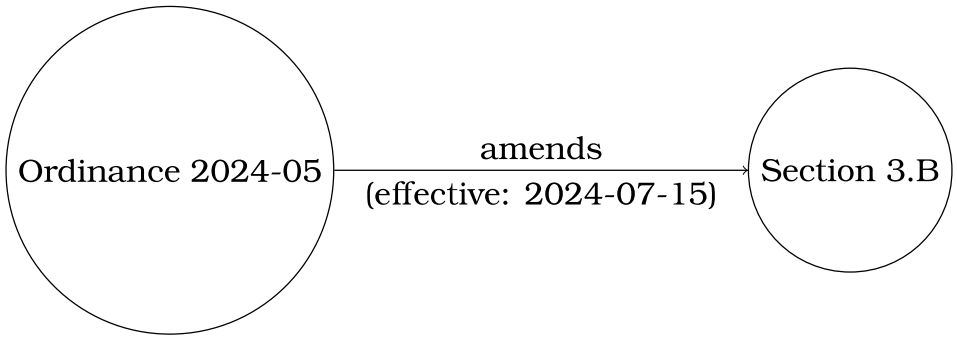

Figure 2.1: A simple knowledge graph fragment representing a legal amendment.

Pioneering work in structured knowledge includes Minsky's concept of Frames (Minsky, 1974), which represented stereotypical situations using slots (attributes) and relationships, influencing subsequent knowledge representation formalisms like description logics and semantic web ontologies.

Knowledge graphs can be implemented and stored using various technologies, each with different strengths:

- **RDF (Resource Description Framework):** A W3C standard data model based on triples (subject-predicate-object statements). RDF

graphs are often serialized in formats like Turtle, RDF/XML, or N-Triples and queried using the SPARQL protocol and query language ("RDF 1.2 Primer", 2025) and (Kumar & Kumar, July 1, 2013). RDF is foundational to the Semantic Web and facilitates data interoperability. Ontology languages like RDFS and OWL (Hitzler et al., 2009) and (Hartig et al., 2025) can be used to define schemas and enable richer reasoning over RDF KGs.

- **Property Graphs:** A flexible graph model widely adopted in native graph databases like Neo4j, Neptune, and TigerGraph. Property Graphs allow attributes (properties) to be attached to both nodes and edges, which can be convenient for certain modeling tasks. They are often queried using specialized graph query languages like Cypher or Gremlin (Fernandes & Bernardino, 2018).

- **Graph Neural Networks (GNNs):** While primarily a machine learning technique rather than a storage mechanism, GNNs operate directly on graph structures (Gupta et al., 2021) and (Scarselli et al., 2009). They learn low-dimensional vector representations (embeddings) of nodes and edges, capturing graph topology and features. These embeddings enable tasks like link prediction (inferring missing relationships), node classification, graph classification, and similarity computations within KGs (Li & Chen, Oct 26, 2021), (Feng et al., 2024), (Wang et al., 2024), and (P. Chen et al., 2021).

- **Other Formats:** KGs can also be represented or serialized using formats like JSON (e.g., JSON-LD) or XML, though these may lack the optimized querying and reasoning capabilities offered by dedicated graph databases or RDF triple stores.

KGs are employed in diverse applications, including powering Google's Knowledge Graph for semantic search, enhancing recommendation systems (e.g., Amazon, Netflix), integrating heterogeneous data sources in enterprises, bioinformatics, financial analysis, and enabling more sophisticated question answering systems **[CITE - KG Applications Survey]**. Their ability to explicitly model complex relationships and provide a structured representation of knowledge makes them potentially valuable for analyzing the internal structure, interconnections, and overall integrity of large document collections, such as legal codes.

2.4 Information Extraction for KG Construction

To leverage the benefits of KGs for document analysis, the unstructured or semi-structured information within the source documents must first be transformed into the structured format of the graph. This process, often termed KG construction or population, relies heavily on Information Extraction (IE) techniques **[CITE - IE Survey]**. This section covers two fundamental IE tasks critical for extracting the primary components of a KG: identifying the nodes (entities) using Named Entity Recognition (NER) and identifying the edges (relationships) using Relation Extraction (RE). LLMs have shown significant promise in performing both tasks, often with minimal task-specific training data **[CITE - LLMs for IE]** and (Benjira et al., 2025).

2.4.1 Named Entity Recognition

Named Entity Recognition (NER) is a fundamental task in information extraction that focuses on identifying and classifying mentions of named entities within unstructured text into pre-defined categories **[CITE - NER Definition]**.

These categories typically include standard types like persons (PER), organizations (ORG), locations (LOC), dates, and monetary values, but crucially, can be extended to domain-specific entities relevant to the application context.

In the context of building knowledge graphs from text, NER plays a crucial role. It serves as the primary mechanism for identifying the potential **nodes** (entities) that will populate the graph. By extracting key actors, locations, concepts, defined terms, document sections, or other items of interest from the source documents, NER provides the raw material for the structured representation. Disambiguating these mentions and linking them to unique identifiers in the KG (Entity Linking) is often a necessary subsequent step **[CITE - Entity Linking Survey]**.

Various methods have been developed for NER over the years:

- **Rule-based Systems:** Early approaches relied on hand-crafted grammatical rules, dictionaries (gazetteers), and regular expressions. These systems can achieve high precision when rules are well-defined but are often brittle, domain-specific, and labor-intensive to create and maintain **[CITE - Rule-Based NER]**.

- **Statistical Models:** Supervised machine learning techniques became dominant, including Hidden Markov Models (HMMs), Maximum Entropy Markov Models (MEMMs), and especially Conditional Random Fields (CRFs), which learn probabilistic sequence labeling models from large annotated datasets **[CITE - Statistical NER e.g.**; **CRF Lafferty et al. 2001]**. These models offered better generalization than purely rule-based systems.

- **Deep Learning Approaches:** More recently, deep neural networks have

achieved state-of-the-art performance. Architectures like Bidirectional Long Short-Term Memory networks (BiLSTMs), often combined with a CRF output layer (BiLSTM-CRF), effectively capture sequential context **[CITE - BiLSTM-CRF NER e.g.; Lample et al. 2016]**. Increasingly, Transformer-based models like BERT (Koroteev, 2025) and its variants, fine-tuned on NER tasks, have become standard, leveraging powerful pre-trained representations **[CITE - Deep Learning NER Survey]**. LLMs can also perform NER directly via prompting or few-shot learning **[CITE - LLM for NER]**.

Applying NER to the legal domain requires careful consideration of domain-specific entities that are critical for understanding legal texts. Beyond standard types, entities might include: specific legal statutes or section references (e.g., '15 Pa.C.S.A. § 1502'), defined legal terms (e.g., 'applicant', 'nonconforming use', 'force majeure'), legal roles (e.g., 'Township Supervisor', 'Zoning Officer', 'plaintiff'), court names, specific dates or deadlines, monetary penalties, and explicit references to other documents or sections **[CITE - Legal NER examples/papers]**. Due to the specialized vocabulary, complex sentence structures, and importance of precision, training or fine-tuning NER models on legally annotated corpora (like those from legal shared tasks or specific research projects) is often necessary to achieve high accuracy **[CITE - Legal NER Datasets/Tasks]**. The output of a robust legal NER system provides the essential entity building blocks for constructing a meaningful knowledge graph from legal texts.

2.4.2 Relation Extraction

While NER identifies the entities (nodes), **Relation Extraction (RE)** is the task of identifying semantic relationships that hold between pairs (or

sometimes n-tuples) of these entities in text **[CITE - RE Definition/Survey]**. These extracted relations typically correspond to the **edges** in the knowledge graph, connecting the nodes identified by NER and thus building the graph's structure. For instance, given the sentence "Acme Corp, headquartered in West Chester, acquired Beta Inc.", RE aims to identify relations like 'headquarteredIn(Acme Corp, West Chester)' and 'acquired(Acme Corp, Beta Inc)'.

Identifying the type of relation is crucial. While early work focused on a small set of predefined relation types (Closed RE), more recent work also tackles Open Information Extraction (OpenIE), which aims to extract relations expressed using arbitrary textual phrases **[CITE - OpenIE]**. For KG construction, typically a predefined schema or ontology dictates the target relation types (Closed RE), such as 'cites', 'amends', 'defines', 'employs', 'locatedIn', etc. Some fundamental ontological relationships often considered include 'is-a' (subclass/instance) and 'part-of' (meronymy) relations, alongside more domain-specific associative relationships **[CITE - Relation Types Ontology]**.

Similar to NER, various approaches have been developed for RE:

- **Rule-based / Pattern-based Systems:** Utilize linguistic patterns (e.g., dependency paths between entities) or hand-crafted rules over text or syntactic structures (like parse trees) to identify relations **[CITE - Rule-Based RE]**. Bootstrapping methods like DIPRE and Snowball automatically learn extraction patterns from seed examples **[CITE - Bootstrapping**. These can be effective for specific relations but suffer from similar limitations as rule-based NER.

- **Supervised Statistical Models:** Train classifiers (e.g., SVMs, MaxEnt) on features derived from the text snippet connecting two candidate entities (e.g., lexical features, syntactic features from dependency paths)

using annotated data **[CITE - Supervised Feature-Based RE]**. Distant supervision attempts to automatically generate training data by aligning known relations from an existing KG (like Freebase or DBpedia) with sentences mentioning the related entities, though this can introduce significant noise **[CITE - Distant Supervision RE]**.

- **Deep Learning Approaches:** Neural models, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs/LSTMs), and Graph Neural Networks (operating on dependency trees), have been applied successfully, often outperforming feature-engineered systems **[CITE - Neural RE Survey]**. Transformer-based models, pre-trained on large corpora and fine-tuned for RE, currently represent the state-of-the-art for many RE benchmarks **[CITE - Transformer for RE]**. LLMs, particularly through prompting techniques (including few-shot prompting), offer a powerful alternative, capable of extracting relations with minimal task-specific fine-tuning by leveraging their vast world knowledge and language understanding capabilities **[CITE - LLM for RE]**. Prompting strategies often involve formulating the task as question answering or fill-in-the-blank over the input text containing the entity pair.

Challenges in RE include handling ambiguity (the same text might imply different relations), extracting relations expressed across sentence boundaries, dealing with complex or n-ary relations (involving more than two entities), adapting models to new domains or relation types with limited data, and robust evaluation. In the legal domain, extracting relations like amendments between code sections, definitions of terms, obligations imposed by regulations, or citations between cases and statutes is critical for building a KG that accurately reflects the legal framework **[CITE - Legal RE]**.

The structured output of NER and RE, when combined and potentially refined (e.g., through entity linking and relation consolidation), forms the basis for the constructed knowledge graph.

2.5 Consistency, Completeness, and Coherence

When analyzing formal document corpora, particularly large and evolving ones like legal codes, software requirements, or technical standards, evaluating their quality often involves assessing their internal integrity. Three key aspects of this integrity are Consistency, Completeness, and Coherence **[CITE - Software Eng Quality Attributes e.g.**; **ISO/IEC 25010]**. These concepts, while sometimes overlapping, address distinct facets crucial for ensuring the documents are understandable, unambiguous, reliable, and effective in their intended function.

- **Consistency:** Refers primarily to the absence of logical contradictions within the document set **[CITE - Consistency Definition]**. A consistent set of statements should not allow for the derivation of both a proposition and its negation. In a legal code, this means it should not contain provisions that assert mutually exclusive facts (e.g., defining the same term in incompatible ways) or prescribe conflicting obligations or permissions under identical conditions (e.g., one section mandates an action that another prohibits for the same actor and circumstance). Detecting inconsistencies is vital for legal certainty, predictability, and avoiding disputes **[CITE - Consistency in Law]**. Formal logic and automated reasoning techniques are often employed to check consistency in formal specifications **[CITE - Formal Methods Consistency]**.

- **Completeness:** Pertains to whether the document set contains all

the necessary information required relative to its intended scope or purpose **[CITE - Completeness Definition]**. Defining completeness is inherently challenging as it depends on a clear specification of what *should* be included. In a legal context, this could mean ensuring that all terms used are adequately defined, procedures referenced are fully specified, criteria for decisions are enumerated exhaustively, exceptions are handled, and potential scenarios relevant to the scope are addressed. Gaps, omissions, or "TBD" markers can lead to ambiguity, loopholes, and disputes. Assessing completeness often requires significant domain knowledge and may involve checking against predefined templates, checklists, or requirements specifications **[CITE - Requirements Completeness]**. The "Closed World Assumption" versus "Open World Assumption" impacts how completeness might be formally interpreted in KGs **[CITE - CWA vs OWA]**.

- **Coherence:** Relates to the overall understandability, organization, and logical flow of the information presented **[CITE - Coherence Definition]**. A coherent document is well-structured, uses terminology consistently across sections, ensures cross-references are accurate and lead to relevant information, avoids unnecessary jargon or ambiguity, and maintains a clear narrative or argumentative structure. While related to consistency (an incoherent document might contain implicit contradictions), coherence focuses more on the clarity, usability, and comprehensibility for a human reader **[CITE - Text Coherence Linguistics]**. Aspects include lexical cohesion, referential clarity, and discourse structure **[CITE - Discourse Analysis Coherence]**.

Ensuring these three qualities simultaneously in large, evolving legal codes through traditional manual review processes is exceptionally difficult.

The sheer volume of text, the intricate web of interdependencies (definitions, cross-references, amendments), the potential for ambiguity in natural language, and the often distributed and lengthy nature of authorship and revision processes over time make manual detection of subtle flaws challenging and error-prone **[CITE - Challenges in Legislative Drafting]**. This is where computational approaches leveraging structured representations like KGs, potentially populated and analyzed with the aid of LLMs, offer significant potential advantages.

A Knowledge Graph, by explicitly modeling entities (like defined terms, sections, obligations, actors, conditions) and their relationships (like 'defines', 'cites', 'amends', 'requires', 'prohibits', 'conflicts with'), provides a structured substrate amenable to automated analysis. Graph-based queries (e.g., using SPARQL or Cypher) or graph algorithms can be designed to automatically detect certain classes of potential inconsistencies, such as finding terms used before they are defined, identifying conflicting property values assigned to the same entity under specific conditions, detecting circular definition chains, or finding contradictory requirements linked to the same scenario **[CITE - KG for Consistency Checks]**. While achieving perfect completeness verification is often intractable or ill-defined for natural language documents, KGs can help identify potential gaps by analyzing the graph's structure for missing nodes (e.g., undefined terms that are used), expected relationships that are absent (e.g., a procedure is mentioned but not detailed), or orphaned sections **[CITE - KG for Completeness Checks]**. Coherence might be partially assessed by analyzing the density and structure of cross-references, consistency in terminology usage (via entity linking), or detecting potentially ambiguous references.

LLMs can potentially play a role throughout this pipeline: aiding in

the initial interpretation of nuanced text to populate the KG accurately (NER/RE), helping to formulate complex graph queries based on natural language questions about integrity, or summarizing the findings from the graph analysis for human review **[CITE - LLMs assisting KG analysis]**. However, the KG itself provides the persistent, globally coherent, and computationally tractable structure necessary for systematic integrity checks that can overcome the context window limitations and potential lack of deterministic reasoning inherent in LLMs alone. Research exploring the use of KGs and related AI techniques for automated consistency and completeness checking in domains like software requirements engineering **[CITE - Automated Consistency/Completeness Checking RE]**, logical formalisms, and more recently, legal texts provides a foundation for this approach **[CITE - AI/KG for Legal Doc Analysis]**. This praxis project aims to build upon such work, investigating the practical application of LLM-driven KG construction for checking the consistency and completeness of municipal legal codes.

2.6 Challenges in Analyzing Large Documents

Research efforts in automated document processing and understanding are extensive, covering tasks like summarization **[CITE - Summarization]**, information extraction (as discussed previously) **[CITE - Info Extraction]**, document classification **[CITE - Doc Classification]**, question answering **[CITE - Document QA]**, and validating the faithfulness or factuality of generated content (like summaries) against source documents **[CITE - Summary Validation]**. Historically, much foundational research and benchmark development focused on relatively small documents (e.g., news articles, single paragraphs, short scientific abstracts) for several practical reasons. Smaller documents

are computationally less demanding to process, and crucially, human evaluation and annotation required to establish ground truth and verify system performance are significantly more feasible and reliable at smaller scales.

However, many critical real-world applications involve documents that are orders of magnitude larger – legal contracts, court proceedings, technical manuals, full-length books, extensive regulatory codes, or large scientific papers. Analyzing these large documents presents distinct and significant challenges:

- **Computational Resources:** Simply processing large volumes of text demands substantial memory (RAM and VRAM for deep learning models), storage, and processing time. The computational complexity often scales non-linearly (e.g., quadratically for standard Transformers) with document length, making naive processing infeasible **[CITE - Computational Cost Large Docs]**.

- **Long-Range Dependencies:** Understanding often requires capturing semantic connections, references (e.g., pronoun resolution, term definitions), or causal dependencies between sections that are far apart in the document. Models with limited context windows struggle to capture these long-distance relationships accurately, as discussed regarding LLMs **[CITE - Long Range Dependency Challenge]**.

- **Context Fragmentation:** Common techniques for handling large documents with fixed-input models involve splitting them into smaller chunks (e.g., fixed size, sentence-based, paragraph-based, or even semantically coherent chunks) **[CITE - Chunking Strategies Review]**. While necessary, this risks losing critical context that spans across chunk boundaries, potentially leading to fragmented understanding or

incorrect inferences when information needs to be synthesized globally. Hierarchical processing methods attempt to mitigate this but add complexity **[CITE - Hierarchical Document Models]**.

- **Evaluation Complexity:** Assessing the quality of processing (e.g., the accuracy of a summary of a 500-page report, the correctness of an answer requiring synthesis across chapters, or the completeness of consistency analysis over an entire legal code) is inherently difficult and resource-intensive for human evaluators. Establishing reliable ground truth for evaluation benchmarks remains a major challenge for large-document tasks **[CITE - Evaluation Challenges Large Docs]**.

Techniques like Retrieval-Augmented Generation (RAG) **[CITE - RAG Lewis et al. 2020]** have emerged as a popular and effective approach to allow LLMs to leverage information from large external corpora without needing to process the entire corpus within their context window. RAG typically involves retrieving relevant text snippets (often chunks) from the large document(s) based on the input query or prompt, and then providing these retrieved snippets as additional context to the LLM for generating a response. While powerful for knowledge-intensive tasks like open-domain QA, standard RAG often retrieves discrete, localized chunks. It may not provide the holistic, structured view of the entire document's content, relationships, and potential inconsistencies that a pre-constructed Knowledge Graph aims to offer, representing a potential gap for tasks requiring global analysis and integrity checking.

2.7 Challenges in Analyzing Legal Documents

Legal documents, particularly statutory or regulatory codes like the municipal ordinances central to this work, represent a compelling yet particularly challenging domain for applying and evaluating advanced document analysis techniques. They possess several intrinsic characteristics that make them both difficult testbeds and highly valuable targets for automation:

- **Complexity and Precision:** Legal language is notoriously dense, often employing specialized terminology (jargon with precise, sometimes non-intuitive, meanings), complex and nested sentence structures (long sentences with multiple subordinate clauses), and numerous explicit and implicit cross-references. Unlike much general text, ambiguity must be minimized, demanding extremely high precision in interpretation and analysis, as misinterpretations can have significant real-world consequences **[CITE - Legal Language Complexity e.g.**; **Ashley AI Law]**.

- **Volume and Interconnectedness:** Legal corpora can be vast (e.g., entire state statutes, federal regulations like the CFR, large collections of case law, or extensive municipal codes). Furthermore, documents within these corpora are rarely standalone; they are highly interconnected through explicit citations, amendments that modify prior text, definitions that apply across sections or entire codes, and implicit dependencies based on legal principles or hierarchy **[CITE - Interconnectedness Legal T**
  Understanding one part often requires understanding its relationship to many others.

- **Semi-structured Format:** While often exhibiting some structure (e.g., organized into titles, chapters, articles, sections, clauses, lists), legal

texts contain significant amounts of unstructured natural language prose within these structures. This mix requires sophisticated NLP techniques capable of handling both the explicit structure and the dense prose content.

- **Critical Need for Integrity:** Perhaps most importantly, the consistency, completeness, and coherence of legal documents are paramount for their function in society. These qualities underpin the rule of law, ensuring predictability, fairness in application, and enforceability. Flaws such as contradictions, ambiguities resulting from omissions, or confusing structure can lead to uncertainty, disputes, costly litigation, and erosion of public trust **[CITE - Importance of Legal Doc Integrity]**.

In this context, the specific focus on the codified ordinances (local laws) of townships within the Commonwealth of Pennsylvania, particularly places like West Chester and surrounding areas in Chester County, provides a valuable and concrete dataset for praxis-oriented research. With potentially hundreds of such municipalities in the state, each with its own evolving code (often compiled by third-party services like General Code or Municode), there exists a substantial body of relevant material. These codes exhibit realistic complexity, having often been developed over many decades, involving multiple authorships (different councils, solicitors), numerous amendments, and periodic recodification efforts.

The legislative drafting and codification process itself, while designed to ensure quality through multiple layers of review (staff, legal counsel/solicitor, planning commissions, public hearings, compiler checks), highlights the potential for introducing errors. When a new ordinance is proposed—whether initiated by elected officials, staff, or residents—it typically undergoes review by township staff and legal counsel who drafts the formal language.

A professional compiler may later be engaged to integrate the new law into the existing code and perform some validation checks. Despite this multi-stage human review process involving various stakeholders with legal or domain expertise, inconsistencies (e.g., contradictions with existing ordinances, conflicts with state preemptions), incompleteness (e.g., missing definitions for newly introduced terms, undefined procedures), and incoherence (e.g., unclear scope, confusing structure, inaccurate cross-references) can still arise and persist, especially as the code grows in size and complexity over time **[CITE - Challenges in Legislative Drafting/Codification]**. The resource-intensive, time-consuming, and inherently fallible nature of purely manual review motivates the exploration of computational methods, like the one proposed herein, to assist legal professionals and municipal staff in maintaining the integrity of these foundational legal documents.

2.8 Related Work

Research relevant to this praxis project spans several areas: utilizing Large Language Models for Information Extraction and Knowledge Graph construction, applying Knowledge Graphs for document analysis and integrity checking, and the specific application of AI and NLP techniques to the legal domain.

LLMs for Information Extraction and KG Construction: The advent of powerful LLMs has revolutionized information extraction. Numerous studies demonstrate the ability of LLMs like GPT-3/4, Claude, and Llama, often via prompting (zero-shot or few-shot), to perform NER and RE with performance rivaling or exceeding traditional fine-tuned models, especially in low-data or specialized domains **[CITE - LLM for IE Survey]**. Researchers have explored various prompting strategies, output parsing techniques, and meth-

ods for mitigating LLM limitations like hallucinations or inconsistencies in extraction **[CITE - Prompting Strategies for IE]**. Several works have specifically focused on constructing KGs from text using LLMs as the primary extraction engine, developing pipelines that integrate entity identification, relation extraction, entity linking, and schema mapping, sometimes incorporating human-in-the-loop refinement **[CITE - LLM-based KG Construction Pipeline 1]**, **[CITE - LLM-based KG Construction Pipeline 2]**. Challenges remain in scalability, controlling the output structure effectively, ensuring factual accuracy, and handling complex, long-form documents during extraction **[CITE - Challenges LLM KG Construction]**.

KGs for Document Analysis and Integrity Checking: Beyond construction, KGs serve as a substrate for advanced document analysis. They have been used to enhance semantic search, enabling queries based on relationships rather than just keywords **[CITE - KG Semantic Search]**. KGs facilitate complex question answering by allowing reasoning over extracted facts **[CITE - KG QA Systems]**. Directly relevant to this work is the use of KGs for consistency and completeness checking. In software requirements engineering, KGs and ontologies have been used to model requirements and detect conflicts or missing elements **[CITE - KG for Requirements Consistency]**. Formal methods often leverage graph-based representations for model checking **[CITE - Formal Methods Graphs]**. In the Semantic Web community, technologies like SHACL (Shapes Constraint Language) provide a standard way to validate RDF KGs against predefined constraints or schemas, effectively checking aspects of consistency and completeness relative to the schema **[CITE - SHACL]**.

AI and NLP for Legal Document Analysis: The legal domain has been a target for AI and NLP research for decades **[CITE - AI Law Survey e.g.**;

Ashley]. Early work focused on rule-based systems for legal reasoning and expert systems. More recent research applies modern NLP to tasks like legal information retrieval **[CITE - Legal IR]**, case outcome prediction **[CITE - Case Outcome Prediction]**, document summarization **[CITE - Legal Summarizat** contract review (clause identification, risk analysis) **[CITE - Contract Analysis AI]**, argument mining **[CITE - Legal Argument Mining]**, and e-discovery. Information extraction (NER and RE) from legal texts has received significant attention, focusing on extracting citations, legal entities, obligations, definitions, and relationships relevant for legal analysis **[CITE - Legal IE Work 1]**, **[CITE - Legal IE Work 2]**. Some prior work has explored automated consistency checking in legal documents, often using rule-based approaches, deontic logic, or domain-specific heuristics, but typically focused on specific types of conflicts rather than a comprehensive KG-based approach applied to municipal codes **[CITE - Prior Legal Consistency Check]**.

Positioning of this Work: This praxis project builds upon these converging lines of research. While previous work has explored LLMs for KG construction and KGs for consistency checking separately, and AI has been applied to legal texts, the specific contribution here lies in the **integration and practical application of modern LLMs to construct KGs *specifically* from municipal legal codes (ordinances) for the explicit purpose of assisting in consistency and completeness analysis**. It addresses the limitations of LLMs (context window) by leveraging the KG structure for global analysis and reasoning. Unlike some prior legal AI work focusing on case law or contracts, this project targets the foundational legislative texts at the local government level. Compared to general KG construction methods, it focuses on the specific entities, relations, and integrity rules pertinent to municipal ordinances. The "praxis" aspect emphasizes the development and evaluation

of a practical methodology and potential tool tailored to assist municipal staff and legal professionals in the challenging task of maintaining the quality of their codified laws, leveraging the latest advancements in LLMs and KG technologies. The evaluation will focus on the effectiveness of this integrated approach in identifying realistic inconsistencies and omissions within this specific legal domain.

2.9 Conclusions

This chapter has surveyed the key bodies of literature relevant to the proposed praxis project on utilizing Large Language Models and Knowledge Graphs for analyzing the consistency and completeness of legal documents, specifically municipal codes. We began by tracing the rise of LLMs, driven by the Transformer architecture (Vaswani et al., 2017), acknowledging their remarkable language processing capabilities but also highlighting their critical limitations concerning finite context windows **[CITE - Context Window Limitations]** and computational scaling (**RefWorks:RefID:97**).

Knowledge Graphs were then introduced as a powerful paradigm for representing structured knowledge, capable of explicitly modeling entities and their relationships **[CITE - KG Overview/History]**. The potential of KGs to serve as a structured substrate for analysis, overcoming LLM context limits, was established. Bridging the gap between unstructured text and structured KGs necessitates Information Extraction, and we reviewed the core tasks of Named Entity Recognition and Relation Extraction, noting the increasing role of deep learning and LLMs in achieving state-of-the-art performance **[CITE - LLMs for IE]**.

The target application was framed by defining the crucial quality attributes of Consistency, Completeness, and Coherence **[CITE - Software Eng Quality Attri**

ISO/IEC 25010], which are essential for the integrity and utility of formal documents. The significant challenges in maintaining these qualities manually, particularly in large and complex legal document corpora like municipal codes **[CITE - Challenges in Legislative Drafting/Codification]**, were underscored, motivating the need for computational assistance. Finally, a review of related work situated this project within the context of ongoing research in LLM-driven IE and KG construction, KG-based analysis, and AI applications in the legal domain, highlighting the novel integration and practical focus on consistency and completeness checking for municipal ordinances.

The limitations of LLMs for global document understanding and the inherent structure offered by KGs, combined with the critical need for ensuring the integrity of legal codes, strongly motivate the methodology proposed in this praxis project. By leveraging LLMs for the nuanced task of extracting information from complex legal text and mapping it into a queryable KG, this work aims to develop and evaluate a practical approach to assist in identifying potential inconsistencies and omissions that might otherwise persist undetected. The following chapter will detail the specific methodology employed to achieve this objective.

Chapter 5: Discussion and Conclusions

5.1 Conclusion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

5.2 Contribution to the Body of Knowledge

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel

leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

5.3 Recommendations for Future Research

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

References

Badshah, S., & Sajjad, H. (2025, March). *Quantifying the capabilities of llms across scale and precision*. http://arxiv.org/abs/2405.03146

Benjira, W., Atigui, F., Bucher, B., Grim-Yefsah, M., & Travers, N. (2025). Automated mapping between sdg indicators and open data: An llm-augmented knowledge graph approach. *Data knowledge engineering*, *156*, 102405. https://doi.org/10.1016/j.datak.2024.102405

Caruccio, L., Cirillo, S., Polese, G., Solimando, G., Sundaramurthy, S., & Tortora, G. (2024). Claude 2.0 large language model: Tackling a real-world classification problem with a new iterative prompt engineering approach. *Intelligent systems with applications*, *21*, 200336. https://doi.org/10.1016/j.iswa.2024.200336

Chen, P., Ding, H., Araki, J., & Huang, R. (2021). Explicitly capturing relations between entity mentions via graph neural networks for domain-specific named entity recognition. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 735–742. https://doi.org/10.18653/v1/2021.acl-short.93

Chen, T., Wang, H., Chen, S., Yu, W., Ma, K., Zhao, X., Zhang, H., & Yu, D. (2023). Dense x retrieval: What retrieval granularity should we use? *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. https://doi.org/10.48550/arxiv.2312.06648

Edwards, B. (2025, March). *Exponential growth brews 1 million ai models on hugging face*. https://arstechnica.com/information-technology/

2024/09/ai-hosting-platform-surpasses-1-million-models-for-the-first-time/

Ehrlinger, L., & W "o ss, W. (2016). Towards a definition of knowledge graphs. *SEMANTiCS (Posters, Demos, SuCCESS)*, *48*(1-4), 2.

Feng, Z., Wang, R., Wang, T., Song, M., Wu, S., & He, S. (2024). A comprehensive survey of dynamic graph neural networks: Models, frameworks, benchmarks, experiments and challenges. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2405.00476

Fernandes, D., & Bernardino, J. Graph databases comparison: Allegrograph, arangodb, infinitegraph, neo4j, and orientdb. In: In *7th international conference on data science, technology and applications (data 2018)*. 2018, July, 373–380. https://doi.org/10.5220/0006910203730380

Gao, He, He, Lin, Pei, Shao, & Zhang. (2025, March). *Examining user-friendly and open-sourced large gpt models: A survey on language, multimodal, and scientific gpt models*. http://arxiv.org/abs/2308.14149

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., & Schelten, e. a., Alan. (2024, November). *The llama 3 herd of models*. http://arxiv.org/abs/2407.21783

Gupta, A., Matta, P., & Pant, B. (2021). Graph neural network: Current state of art, challenges and applications. *Materials today : proceedings*, *46*, 10927–10932. https://doi.org/10.1016/j.matpr.2021.01.950

Hartig, O., Seaborne, A., Taelman, R., Williams, W., & Tanon, T. (2025, April). *Sparql 1.2 query language*. https://www.w3.org/TR/sparql12-query/

Hitzler, P., Krotzsch, M., & Rudolph, S. (2009, August). *Foundations of semantic web technologies* (Anonymous, Trans.; 1st). Chapman; Hall/CRC. https://www.taylorfrancis.com/books/mono/10.1201/

9781420090512/foundations-semantic-web-technologies-pascal-hitzler-markus-krotzsch-sebastian-rudolph

Hogan, A., Blomqvist, E., Cochez, M., D'amato, C., De Melo, G., Gutierrez, C., Kirrane, S., Gayo, J. E. L., Navigli, R., Neumaier, S., Ngomo, A.-C. N., Polleres, A., Rashid, S. M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., & Zimmermann, A. (2021). Knowledge graphs. *ACM Computing Surveys*, *54*(4), 1–37. https://doi.org/10.1145/3447772

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2025, April). *Scaling laws for neural language models*. http://arxiv.org/abs/2001.08361

Koroteev, M. V. (2025, March). *Bert: A review of applications in natural language processing and understanding*. http://arxiv.org/abs/2103.11943

Kumar, N., & Kumar, S. Querying rdf and owl data source using sparql. In: In *Fourth international conference on computing, communications and networking technologies (icccnt)*. IEEE, July 1, 2013, 2013, 1–6. https://doi.org/10.1109/ICCCNT.2013.6726698

Li, H., & Chen, L. Cache-based gnn system for dynamic graphs. In: New York, NY, USA: ACM, Oct 26, 2021, 937–946. https://doi.org/10.1145/3459637.3482237

Liu, J., Zhu, D., Bai, Z., He, Y., Liao, H., Que, H., Wang, Z., Zhang, C., Zhang, G., Zhang, J., Zhang, Y., Chen, Z., Guo, H., Li, S., Liu, Z., Shan, Y., Song, Y., Tian, J., Wu, W., ... Zhang, Z. (2025, April). *A comprehensive survey on long context language modeling*. http://arxiv.org/abs/2503.17407

Minsky, M. (1974, June). *A framework for representing knowledge*. http://hdl.handle.net/1721.1/6089

Rdf 1.2 primer. (2025, April). https://www.w3.org/TR/rdf12-primer/

Rzepka, R., Muraji, S., & Obayashi, A. Expert evaluation of export control-related question answering capabilities of llms [ID: cdi$_{i}eee_{p}rimary_{1}$0487735]. In: In *Ieee asia-pacific conference on computer science and data engineering (csde)*. ID: cdi$_{i}eee_{p}rimary_{1}$0487735. IEEE, 2023, December, 1–6. ISBN: 9798350341072. https://doi.org/10.1109/CSDE59766.2023.10487735

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2009). The graph neural network model. *IEEE transaction on neural networks and learning systems*, *20*(1), 61–80. https://doi.org/10.1109/TNN.2008.2005605

Schmidt, C. W., Reddy, V., Zhang, H., Alameddine, A., Uzan, O., Pinter, Y., & Tanner, C. (2025, April). *Tokenization is more than compression*. http://arxiv.org/abs/2402.18376

Shen, A., Mistica, M., Salehi, B., Li, H., Baldwin, T., & Qi, J. (2021). Evaluating document coherence modeling. *Transactions of the Association for Computational Linguistics*, *9*, 621–640. https://doi.org/10.1162/tacl_a_00388

Tay, Y., Dehghani, M., Bahri, D., & Metzler, D. (2023). Efficient transformers: A survey. *ACM computing surveys*, *55*(6), 1–28. https://doi.org/10.1145/3530811

Team, G., Anil, R., Borgeaud, S., Alayrac, J.-B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., & Hauth, e. a., Anja. (2024, -06-17). *Gemini: A family of highly capable multimodal models* (I tried uploading the PDF but it would not upload. It is in my Reference file asnbsp;2312.11805v4 (1).pdf.). http://arxiv.org/abs/2312.11805

Tröls, M. A., Marchezan, L., Mashkoor, A., & Egyed, A. (2022). Instant and global consistency checking during collaborative engineering. *Software and systems modeling*, *21*(6), 2489–2515. https://doi.org/10.1007/s10270-022-00984-4

Turner, R. E. (2025, March). *An introduction to transformers*. http://arxiv.org/abs/2304.10557

Vaswani, A., Shazeer, N., Brain, G., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., & Kaiser, Ł. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

Wang, K., Ding, Y., & Han, S. C. (2024). Graph neural networks for text classification: A survey. *The Artificial intelligence review*, *57*(8), 190. https://doi.org/10.1007/s10462-024-10808-0

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., & Zhang, e. a., Junjie. (2023). A survey of large language models. *arXiv preprint arXiv:2303.18223*, *1*(2). https://doi.org/10.48550/arXiv.2303.18223

Zowghi, D., & Gervasi, V. (2003). On the interplay between consistency, completeness, and correctness in requirements evolution. *Information and Software Technology*, *45*(14), 993–1009. https://doi.org/10.1016/S0950-5849(03)00100-9