Using Large Language Models to Convert Documents to Knowledge Graphs to Check for Completeness and Consistency

by Michael Wacey

B.S. in Computer Science, June 1985, Washington University in St. Louis
M.S. in Computer Science, December 1990, Drexel University

A Praxis submitted to

The Faculty of
The School of Engineering and Applied Science
of The George Washington University
in partial satisfaction of the requirements
for the degree of Doctor of Engineering

March 8, 2026

Praxis directed by

Professor M. Elbasheer
Professorial Lecturer in Engineering and Applied Science

The School of Engineering and Applied Science of The George Washington University certifies that [Your Name] has passed the Final Examination for the degree of Doctor of Engineering as of December 18, 2025. This is the final and approved form of the Praxis.

Using Large Language Models to Convert Documents to Knowledge Graphs to Check for Completeness and Consistency

Michael Wacey

Praxis Research Committee:

XYZ, Professor of Online Engineering Programs, Chair

XYZ, Professorial Lecturer of Engineering and Applied Science, Committee Member

Dedication

In loving memory of my parents, Jack and Helen. I dedicate this work to their belief in me and the lessons they taught. From my mother, I learned what is possible through sheer determination. The memory of her returning to school to earn a master's in economics while raising our family is an inspiration that I carry with me always. From both of my parents, I received the encouragement and love of learning that made this achievement possible.

To my siblings, Gordon, Carole, and Iain, who have graciously and humorously allowed me to neglect them during this endeavor. I am deeply grateful for your patience and look forward to reconnecting now that it is complete.

To my children, Alex, Peter, and Kayla. Your unwavering support and quiet understanding have been a constant source of strength. Thank you for inspiring me to see this through.

And most importantly, to my lovely wife, Elaine. You encouraged me to start this journey, sustained me with endless coffee and patience when the mornings were early or the nights grew long, and became my partner in every sense of the word. You will now be living with the results, and I am eternally grateful. This achievement is as much yours as it is mine.

Acknowledgments

I would like to express my profound gratitude to my advisor, Dr. Elabasheer, whose mentorship was instrumental in guiding this research. He showed me the way forward when the path was unclear and challenged me to think more critically. Whenever I find myself thinking I cannot multitask or overcome a challenge, I will undoubtedly hear his voice reminding me that I must—a lesson in perseverance I will carry throughout my career.

My deepest appreciation extends to all my professors at The George Washington University who taught me during this program. Their dedication to teaching and intellectual generosity created the rich academic environment and provided the abundant knowledge that were essential for this journey.

I am also deeply grateful to my West Chester University Dean, Dr. Burns, who served as a steadfast source of support. He was always available to listen to my trials and tribulations, and with a few short, insightful words, he consistently helped me untangle complex problems and understand how to proceed. His wisdom and perspective were invaluable.

A special and heartfelt thank you is reserved for my wife, Elaine. As a lawyer, her tolerance for dense text is matched only by her sharp eye for detail—both of which were indispensable gifts to this project. She patiently read every single word at least ten times, and her contribution to this work is immeasurable.

Finally, I wish to acknowledge my students. Their intellectual curiosity and enthusiasm for learning were a constant source of motivation. It was my desire to be a more effective and knowledgeable teacher for them that provided the final impetus to see this project through to its completion. You reminded me daily of why this work matters.

Abstract

Using Large Language Models to Convert Documents to Knowledge Graphs to Check for Completeness and Consistency

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.

Table of Contents

List of Figures

List of Tables

Glossary of Terms

AI Artificial Intelligence

ANN Artificial Neural Network

API Application Programming Interface

BiLSTM Bidirectional Long Short-Term Memory Network

BPE Byte Pair Encoding

CPU Central Processing Unit

CRF Conditional Random Field

DAG Directed Acyclic Graph

GAT Graph Attention Networks

GCNN Graph Convolutional Neural Network

GNN Graph Neural Network

GPT Generative Pre-trained Transformer

GPU Graphical Processing Unit

HMM Hidden Markov Model

IE Information Extraction

IRI Internationalized Resource Identifier

JSON JavaScript Object Notation

KG Knowledge Graph

LLM Large Language Model

LSTM Long Short-Term Memory

MEMM Maximum Entropy Markov Model

ML Machine Learning

NER Named Entity Recognition

NLP Natural Language Processing

OWL Web Ontology Language

RAM Random Access Memory

RDF Resource Description Framework

RE Relationship Extraction

SHAP SHapley Additive exPlanations

SPARQL SPARQL Protocol and RDF Query Language

SVM Support Vector Machine

TN True Negative

TP True Positive

UBB User-Based Batching

UBS User-Based Sequencing

URI Uniform Resource Identifier

VRAM Video Random Access Memory

W3C World Wide Web Consortium

XML eXtensible Markup Language

Chapter 1: Introduction

1.1 Background and Research Motivation

Ensuring document quality involves verifying completeness, consistency, and correctness (Zowghi & Gervasi, 2003). While evaluating correctness often necessitates access to knowledge external to the document and understanding the document's intent, completeness and consistency can be assessed using only the document's internal content. This research focuses on developing automated methods using Large Language Models (LLMs) to address the latter two aspects. The specific focus is on converting a large document into a knowledge graph that can be used in future research to check for document consistency and completeness.

1.1.1 Background

The increasing complexity and scale of textual documents in various domains present significant challenges to ensuring consistency and completeness. Legal codes, technical documentation, and regulatory frameworks are often drafted collaboratively over extended periods, a process that can lead to inconsistencies, redundancies, and informational gaps. Traditional manual review methods, while necessary, are labor-intensive and prone to human oversight, making automated solutions an attractive alternative. Advances in Natural Language Processing (NLP) and Artificial Intelligence (AI) have introduced new methodologies for analyzing and structuring large bodies of text, with promising applications in document validation and knowledge extraction.

At the core of modern NLP advancements are Transformer-based models that rely on the Attention Mechanism to understand and generate text. LLMs, which build upon this foundation, can process and interpret vast amounts of textual data, although they are constrained by fixed context windows. To address this limitation, structured approaches such as knowledge graphs have emerged, enabling explicit representation of entities and relationships within documents. This research applies these technologies to Pennsylvania township laws, a domain where maintaining consistency is particularly critical. Given the size and complexity of municipal codes, inconsistencies in legal definitions, zoning regulations, and procedural rules can lead to legal disputes and financial losses. By leveraging AI-driven tools, this study aims to develop a framework for systematically analyzing and improving the consistency of legal documents.

Ensuring structural consistency and completeness in documents has been a longstanding challenge across various domains. Previous research has focused on methods to maintain internal coherence within documents (Laban et al., 2021), while other studies have explored domain-specific approaches to consistency checking (Tröls et al., 2022). In academic literature, the term coherence is often used interchangeably with consistency (Shen et al., 2021), reflecting the broader goal of ensuring logical and semantic alignment within textual content.

In 2017, a research team at Google introduced the Transformer model, a neural network architecture based entirely on the Attention Mechanism (Vaswani et al., 2017). Unlike previous sequential models, the Transformer processes all words within a given input simultaneously, allowing it to assess how each word influences others across the text. Using self-attention, this architecture captures long-range dependencies more effectively than earlier

models. Despite advances in scaling Transformer-based models, they remain constrained by a limited attention window due to memory and computational efficiency considerations.

Large Language Models are built upon the Transformer architecture and inherit its fundamental attention-based mechanisms. These models, however, are constrained by a fixed context window, which limits the amount of text they can analyze at once. As documents grow in length, they often exceed this window, preventing comprehensive processing in a single pass. Despite this limitation, document analysis does not necessarily require attending to an entire document simultaneously. Instead, LLMs can be employed to extract key entities and concepts across different sections, enabling a more focused and structured approach to consistency checking. Through the identification of entities and the analysis of their relationships, LLMs can effectively navigate large documents while maintaining efficiency.

Knowledge graphs provide a structured, human-readable representation of information, serving as an alternative to the implicit encoding of knowledge found in neural networks. A knowledge graph is a directed acyclic graph (DAG) in which nodes represent entities and edges define the relationships between them. Each node can possess attributes that enrich its descriptive properties. For instance, a node representing a car might include attributes such as color, model, or manufacturer. A useful way to conceptualize knowledge graphs is through the framework of frames, as described by Minsky (Minsky, 1974). In contrast to LLMs, which rely on statistical inference, knowledge graphs offer explicit, interpretable relationships that can be leveraged for consistency and completeness verification in structured documents.

Pennsylvania is home to over 1,200 townships of the second class, each

responsible for drafting and maintaining its own set of municipal laws. These laws regulate a wide range of local governance areas, including police services, fire departments, zoning, and land development. Over time, the cumulative nature of legal amendments introduces inconsistencies and gaps, which, if left unaddressed, can lead to legal ambiguities and enforcement challenges. Although legal professionals and municipal officials work diligently to identify and resolve these issues, the complexity of these documents—often spanning thousands of pages—renders manual review error-prone and inefficient.

A key source of complexity is the interdependence of different sections within municipal codes. For example, early sections may define zoning regulations, specifying minimum frontage, setbacks, and other boundary constraints for different zoning districts. Inconsistencies can arise, however, when later sections introduce or reference zoning areas that were never formally defined. Similar discrepancies can emerge across other regulatory provisions, necessitating careful synchronization of legal language and definitions. Ensuring consistency across these interconnected legal elements is a critical challenge that demands a more systematic and automated approach to legal document analysis.

1.1.2 Research Motivation

Despite extensive research on the analysis of small documents or specific document sections, a significant gap remains in addressing the challenges of comprehensive, large-scale document analysis. The need for automated consistency and completeness checks is critical in various industries where these tasks are often performed manually, requiring substantial time and resources while still potentially yielding suboptimal results. This research

aims to bridge this gap by developing an effective and efficient automated solution.

The process for publishing local regulations in Pennsylvania townships exemplifies these challenges. After a governing body enacts a law, it is sent to an organization for compilation into the township's existing legal code. This manual and intensive process involves determining if any existing laws are affected by the new one. Even with this careful review, new laws frequently render the existing legal framework incomplete or inconsistent, thereby motivating the present research.

1.2 Problem Statement

Municipal laws in Pennsylvania Townships, authored by multiple people over time, develop inconsistencies and are incomplete (Curley, 2024; Rau, 2024; Sanders, 2024), leading to annual revenue losses of hundreds of thousands of dollars. (Bosco, 2024)

The complexity of municipal laws in Pennsylvania townships arises from their incremental development. Ordinances and regulations are often drafted by different individuals, including elected officials, legal counsel, and administrative staff, each contributing to the evolving legal framework. This decentralized process can lead to inconsistencies in language, overlapping provisions, and unintended gaps in regulatory coverage. As laws are amended or new ones are introduced, prior statutes may not be adequately reconciled, further exacerbating these issues. Without a systematic approach to maintaining legal coherence, townships face challenges in enforcing their laws effectively and equitably.

The consequences of these inconsistencies extend beyond legal ambiguity. Incomplete or conflicting municipal laws can create loopholes that hinder a

township's ability to collect fees, fines, and other sources of revenue. For example, unclear zoning regulations may allow developments to proceed without appropriate permits or impact fees, and ambiguous tax ordinances can lead to disputes that reduce collections. When enforcement mechanisms are weakened by gaps in the legal framework, municipalities struggle to ensure compliance, leading to significant financial losses. These inefficiencies, compounded over time, place additional strain on local budgets and limit resources for essential public services and infrastructure improvements.

Addressing these issues requires a structured methodology for analyzing, refining, and maintaining municipal laws. Traditional legal review processes, while valuable, are labor-intensive and reactive, often identifying issues only after disputes or financial shortfalls have arisen. Advances in artificial intelligence, particularly the use of LLMs, offer a potential solution by systematically identifying inconsistencies, redundancies, and gaps within legal texts. By applying LLMs to municipal laws, townships could proactively assess their legal frameworks, thereby improving clarity, enforcement, and financial sustainability. The implementation of such an approach, however, requires careful consideration of computational constraints, document formats, and the broader applicability of AI-driven legal analysis.

1.3 Thesis Statement

An LLM-based tool to convert a document into an attributed knowledge graph can be used to check for consistency and completeness, which will allow municipal lawyers to create consistent and complete law documents, thereby preventing costly disputes and reducing revenue losses.

The application of LLMs in legal document analysis has the potential to revolutionize municipal law by providing an automated, systematic approach

to ensuring consistency and completeness. Traditional legal drafting and review processes rely heavily on human oversight and are consequently susceptible to errors, especially in laws that have evolved over time through multiple amendments and contributors. By leveraging an LLM-based tool to convert legal documents into attributed knowledge graphs, municipalities can proactively identify gaps, redundancies, and contradictions before laws are enacted or enforced. This proactive approach serves to minimize ambiguity, strengthen legal clarity, and enhance the efficiency of legal review processes.

A knowledge graph-based representation of municipal laws enables a structured, machine-readable format that facilitates logical analysis. Unlike traditional text-based legal review, which requires extensive manual effort to trace dependencies and resolve conflicts, a knowledge graph explicitly maps relationships between legal provisions, definitions, and enforcement mechanisms. This structure allows municipal lawyers to assess the interconnectivity of legal clauses and verify their consistency against established legal principles. Furthermore, an attributed knowledge graph can highlight areas where laws are incomplete or misaligned with overarching governance policies, enabling timely revisions that improve legal coherence.

Beyond improving legal clarity, the ability to create consistent and complete municipal laws has direct financial implications. Inconsistent or incomplete regulations can lead to disputes over zoning, taxation, and permitting, often resulting in costly litigation or lost revenue from unenforceable provisions. By employing an LLM-driven tool to detect and resolve these issues at the drafting stage, municipalities can reduce legal ambiguities that might otherwise be exploited. This strengthens fiscal sustainability by preventing revenue leakage and ensuring that all applicable fees, fines, and taxes are properly assessed and collected.

The integration of LLM-based tools into municipal lawmaking represents a transformative step toward modernizing local governance. As artificial intelligence continues to advance, municipalities that adopt such technologies will gain a significant advantage in maintaining legally sound and financially sustainable frameworks. Future research could extend this approach beyond municipal laws to other domains of legal and regulatory governance, demonstrating the broader impact of AI-driven knowledge representation on ensuring legal accuracy and reducing administrative burdens.

1.4 Research Objectives

The primary objective of this research is to develop a tool capable of automatically processing documents of any size into a coherent set of entities within a knowledge graph. This tool will leverage advanced techniques to analyze document content, identify potential entities, and provide access to the resulting knowledge graph.

The created knowledge graph will then be analyzed to determine its suitability for checking the source document for inconsistencies and incompleteness. This evaluation will involve introducing targeted issues into the source documents and subsequently demonstrating the ease with which these issues can be observed and identified within the knowledge graph representation.

1.5 Research Questions

To achieve the research objectives, the following research questions will be addressed.

RQ1: Can an LLM be used to convert a large document into a knowledge

graph?

RQ2: Can an LLM be used to process multiple knowledge graphs into a typed cluster of knowledge graphs.

RQ3: Can a typed cluster of knowledge graphs be used to check the source document for consistency and completeness?

1.6 Research Hypotheses

This research will test the following hypotheses.

H1: An LLM can be used to convert a large document into a knowledge graph.

H2: An LLM can be used to process multiple knowledge graphs into a typed cluster of knowledge graphs.

H3: A typed cluster of knowledge graphs can be used to check the source document for consistency and completeness.

1.7 Research Scope and Limitations

The subsequent sections outline the scope and limitations of this study. This research employs Pennsylvania township laws as a case study to develop and evaluate an automated tool for analyzing legal documents. These publicly accessible laws, having undergone extensive manual reviews, provide a rigorous benchmark for assessing the proposed methodology. The primary focus is the construction of Knowledge Graphs that faithfully represent document structure and content, thereby laying the groundwork for future work in verifying legal consistency and completeness. This study acknowledges several inherent limitations, including computational constraints, challenges associated with specific document formats, and a

primary emphasis on textual analysis, all of which underscore the need for continued research.

1.7.1 Research Scope

This study focuses on the use of Pennsylvania township laws as a case study for developing and testing an automated tool designed to analyze legal documents. These laws, publicly available in PDF and Word formats, were selected for their complexity, length, and history of multiple authorships. Having undergone rigorous manual reviews, they serve as an ideal benchmark for evaluating the effectiveness of the proposed approach. Although the primary application is in the legal domain, the methodology is designed to be adaptable for broader use across various document types.

The core of this research centers on constructing Knowledge Graphs that accurately represent the structure and content of the documents under review. These graphs will serve as a foundation for future work in verifying legal consistency and completeness. While this study will assess the suitability of the generated Knowledge Graphs for such tasks, the actual implementation of automated consistency and completeness checks is deferred to future research. This approach ensures a focused and systematic exploration of Knowledge Graph generation while establishing a foundation for subsequent advancements in automated legal analysis.

1.7.2 Research Limitations

This study has several potential limitations. Computational constraints may affect the efficiency and scalability of processing large and complex legal documents. Challenges may also arise in handling specific document formats and linguistic intricacies, particularly in ensuring accurate inter-

pretation and structuring of legal text. Furthermore, while this research focuses on leveraging existing LLMs such as Gemini and ChatGPT, it does not involve developing specialized models tailored for knowledge graph construction. Such specialization could be an avenue for future work to reduce computational costs and energy consumption.

This research does not perform direct testing for consistency and completeness. Instead, it utilizes Pennsylvania township laws that are publicly available and have already undergone such validation, treating them as a gold standard. Future studies should explore the applicability of this approach to a broader range of legal and non-legal documents where ground-truth validation is not pre-existing.

For document handling, this research primarily uses Word documents to facilitate modifications during testing. Although the methodology is expected to be compatible with PDFs, further research is needed to confirm seamless integration and processing across different file formats.

Finally, this study is limited to textual analysis. Future work could expand upon this research by incorporating non-textual elements such as tables, formulas, images, and diagrams to achieve a more holistic document comprehension and analysis.

1.8 Praxis Organization

The remainder of this research is organized into several key chapters. Chapter 2 provides a comprehensive review of the relevant literature, focusing on the creation of knowledge graphs from documents by LLMs, the processing of multiple knowledge graphs by LLMs, and the utility of knowledge graphs in ensuring document consistency and completeness. This review also covers the process of creating and maintaining local laws in Pennsylvania. Chapter

3 delves into the statistical and machine learning methodologies employed, detailing the processes of data pre-processing, model selection, training, and evaluation. Chapter 4 presents and analyzes the results, addressing each research question and hypothesis while evaluating the performance of the proposed methodology. Finally, Chapter 5 concludes the investigation with a discussion of the key findings, contributions, and recommendations for practical applications, as well as potential avenues for future research.

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, documented in the seminal paper *Attention Is All You Need* (Vaswani et al., 2017). This paper introduced the Transformer architecture, which leverages self-attention mechanisms and serves as 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, 2024; Verma, 2024).

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 a model can process simultaneously when generating a response or performing an analysis. Consequently, if critical information or dependencies within a document 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., 2020). This limitation poses a significant challenge when analyzing large or complex documents where understanding relies on synthesizing information across distant sections.

A promising approach to mitigate this limitation involves transforming large, unstructured documents into structured representations using Knowledge Graphs (KGs). Through the extraction of key entities, relationships, and

attributes from 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). Such a structure allows for querying and reasoning over the entire document's scope, independent of an LLM's context window constraints, potentially enabling a more focused and comprehensive analysis for tasks like ensuring information integrity.

This chapter reviews the pertinent literature underpinning this approach, beginning with an examination of the development and characteristics of Large Language Models. It focuses on their capabilities and limitations, particularly the context window constraint. Subsequently, the text delves into the principles, construction, and application of KGs as structured knowledge representations. Key techniques for populating KGs from text via Information Extraction (IE) are then discussed, followed by an exploration of the challenges associated with processing large and complex documents, especially within the legal domain. The chapter concludes by defining the critical concepts of consistency, completeness, and coherence, surveying related work, and presenting a summary that motivates the proposed research direction.

2.2 Large Language Models

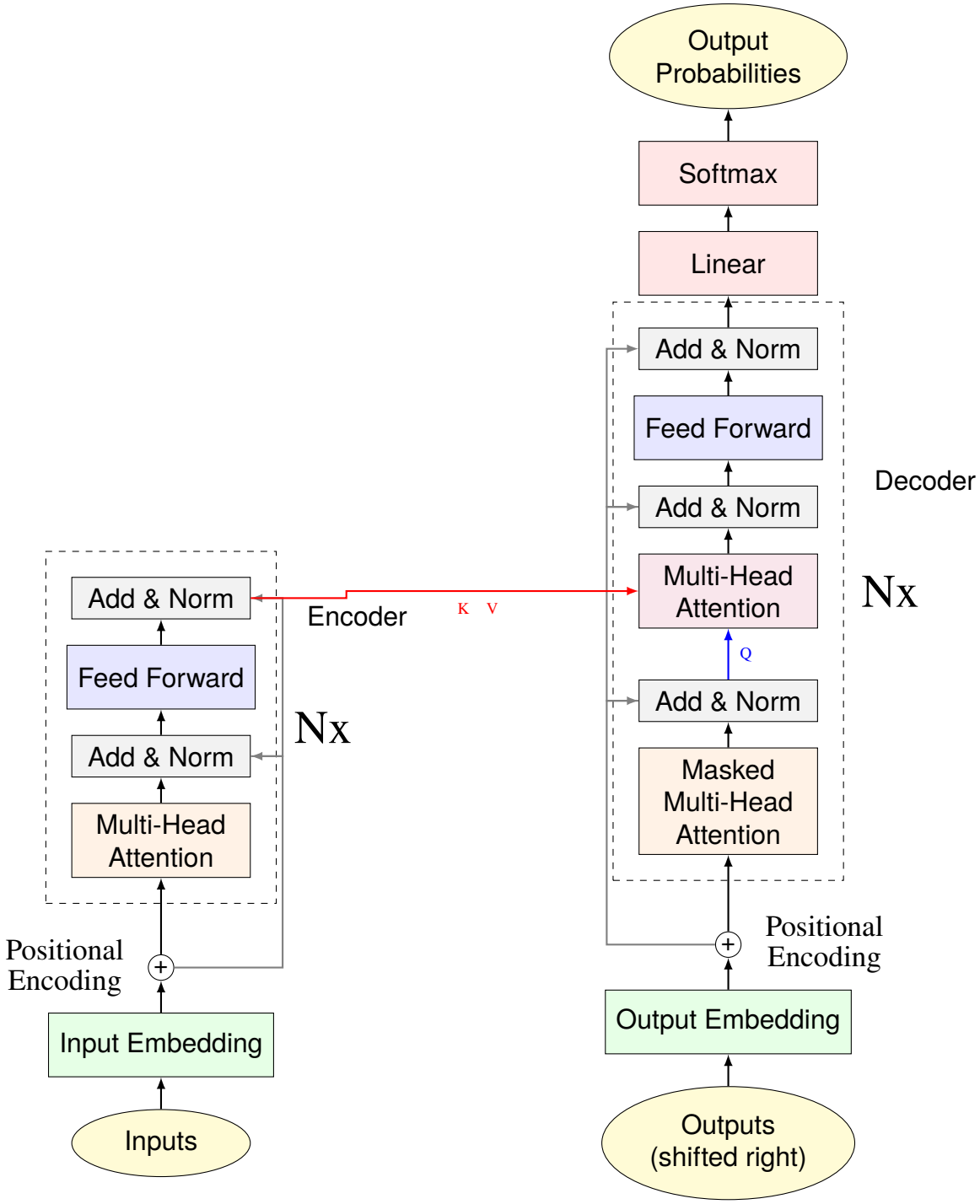
The trajectory of modern NLP took a significant turn in 2017 with the publication of *Attention Is All You Need* by Vaswani et al. (Vaswani et al., 2017). This work introduced the Transformer architecture, which relies on self-attention mechanisms to weigh the importance of different words (tokens) in an input sequence. This design enables superior handling of long-range dependencies compared to previous dominant recurrent or convolutional architectures, thereby addressing critical bottlenecks in earlier

sequence models (Turner, 2024; Zhao et al., 2023). This innovation paved the way for the development of increasingly powerful language models, such as Google's BERT, which introduced bidirectional pre-training (Gardazi et al., 2025), and OpenAI's influential Generative Pre-trained Transformer (GPT) series (Gao et al., 2023).

While research groups at numerous institutions continuously pursued improvements, the public release of OpenAI's *ChatGPT* on November 30, 2022, marked a pivotal moment. The event dramatically increased public awareness and accelerated the deployment of advanced conversational AI systems. It also catalyzed the development of competing models from major research labs, including Google's *Gemini* family, Anthropic's safety-focused *Claude* series, and Meta's open-source *Llama* family (Caruccio et al., 2024; Grattafiori et al., 2024; Team 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 (Edwards, 2025).

Functionally, LLMs process input text (a "prompt") by converting it into numerical representations called tokens, often using techniques like Byte Pair Encoding (BPE) or WordPiece (Schmidt et al., 2024). The model then uses the complex patterns learned during pre-training on vast text corpora to predict subsequent tokens autoregressively, generating a coherent and contextually relevant output. Prompts can be engineered to elicit specific behaviors, including the analysis of substantial text provided for context (in-context learning). For instance, an LLM can be prompted with a company's annual report to answer specific questions or to summarize key findings, tasks on which many current models perform reasonably well, provided the information falls within their processing limits (Rzepka et al., 2023). The

architecture of the Transformer model is depicted in Figure 2.1.



adapted from Vaswani et al. (2017)

Figure 2.1: *The Transformer - model architecture*

A fundamental limitation of LLMs, however, remains the context window size. This size, representing the maximum number of tokens the model can attend to simultaneously, is always finite, although it has increased with newer model generations (Kaplan et al., 2020; Liu et al., 2025; Ratner et al., 2022). If a document's length exceeds this limit, the LLM cannot process it in a single pass. Standard techniques involve processing the document in chunks, yet this can sever long-distance contextual links crucial for deep understanding (T. Chen et al., 2024). For example, determining if a policy defined on page one of a lengthy legal code is contradicted by regulations hundreds of pages later becomes impossible if the intervening text exceeds the context window, as the model would process these sections independently.

Furthermore, the computational cost of processing information within the context window is 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 requirements (Vaswani et al., 2017). Although various "efficient Transformer" variants aim to reduce this to near-linear complexity, processing long sequences still demands substantial resources (Tay et al., 2023). This 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, evolving from concepts in semantic networks, frame systems, and earlier AI research in symbolic knowledge representation (Hogan et al., 2021). Formally, a KG represents knowledge as a directed labeled graph,

comprising a collection of interconnected entities (nodes) and the explicitly typed relationships (edges) between them. Both nodes and edges can possess attributes or properties that store additional metadata or context (Cong, 2024; Ehrlinger & W "o ss, 2016).

The core components of a KG are as follows:

- **Nodes (Entities):** These represent real-world objects, abstract concepts, events, or specific instances of interest. Examples include persons, organizations, locations, legal statutes ('15 Pa.C.S.A. § 1502'), or defined terms ('nonconforming use'). Nodes are often identified by unique identifiers.
- **Edges (Relationships):** These represent the connections or typed relationships between pairs of nodes, such as 'works for', 'located in', 'cites', or 'amends'. Edges are typically directed from a subject node to an object node and are labeled with the relationship type.
- **Attributes (Properties):** These are key-value pairs associated with nodes or edges, providing additional details. For instance, a 'Person' node might have an 'email' attribute, or a 'cites' edge might have an 'effectiveDate' attribute.

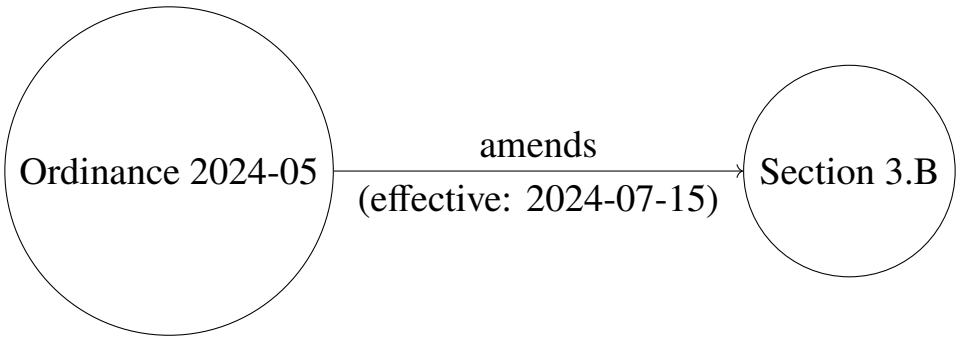


Figure 2.2: *Knowledge graph fragment of a legal amendment.*

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

Knowledge graphs are implemented using various technologies. The Resource Description Framework (RDF) is a W3C standard based on triples (subject-predicate-object) and is foundational to the Semantic Web. It is often queried using SPARQL and defined with ontology languages like OWL (Hitzler, Krotzsch, & Rudolph, 2009b; N. Kumar & Kumar, 2013; "RDF 1.2 Primer," 2025). Property Graphs, used in databases like Neo4j, offer a flexible model where both nodes and edges can have properties and are queried with languages like Cypher or Gremlin (Fernandes & Bernardino, 2018). Additionally, Graph Neural Networks (GNNs) are machine learning techniques that operate on graph structures to learn vector representations (embeddings), enabling tasks like link prediction and node classification (Gupta et al., 2021; Scarselli et al., 2009; K. Wang et al., 2024).

KGs are employed in diverse applications, including semantic search, recommendation systems, and enterprise data integration (Fensel et al., 2020; Hogan et al., 2021; Ji et al., 2022). Their ability to explicitly model complex relationships makes them valuable for analyzing the internal structure and integrity of large document collections.

2.4 Information Extraction for KG Construction

To leverage KGs for document analysis, the unstructured information within source documents must be transformed into the structured format of a graph. This process, termed KG construction, relies on Information Extraction (IE) techniques (Kolluru et al., 2020; Zhong et al., 2024). This

section covers two fundamental IE tasks: identifying nodes via Named Entity Recognition (NER) and identifying edges via Relation Extraction (RE). LLMs have shown significant promise in performing both tasks, often with minimal task-specific training data (Benjira et al., 2025).

2.4.1 Named Entity Recognition

Named Entity Recognition (NER) is a primary task in information extraction that identifies and classifies named entities in text into predefined categories (Al-Moslmi et al., 2020). These categories can include standard types like persons and organizations or be extended to domain-specific entities. For building KGs, NER serves as the main mechanism for identifying the potential **nodes** of the graph. A subsequent step, Entity Linking, is often necessary to disambiguate these mentions and link them to unique identifiers (Chaurasiya et al., 2022).

NER methods have evolved significantly. Early approaches were rule-based systems that used hand-crafted rules and dictionaries, which achieved high precision but were brittle and labor-intensive (Grishman & Sundheim, 1996; Nadeau & Sekine, 2007). These were followed by statistical models like Hidden Markov Models and Conditional Random Fields, which offered better generalization (Lafferty et al., 2001). Currently, deep learning approaches, particularly Transformer-based models like BERT, have achieved state-of-the-art performance by leveraging powerful pre-trained representations (Al-Moslmi et al., 2020; Carbonell et al., 2020).

Applying NER to the legal domain requires identifying specialized entities such as specific statutes, defined legal terms, legal roles, and explicit document references (Au et al., 2022; Kalamkar et al., 2022). The unique vocabulary and complex sentence structures of legal texts necessitate that

NER models be trained or fine-tuned on legally annotated corpora to achieve high accuracy (Chalkidis et al., 2022). A robust legal NER system provides the essential building blocks for constructing a meaningful knowledge graph from legal documents.

2.4.2 Relation Extraction

While NER identifies entities (nodes), Relation Extraction (RE) identifies the semantic relationships between them, which correspond to the edges in a knowledge graph (Carbonell et al., 2020; Ji et al., 2022). For instance, from the sentence "Acme Corp, headquartered in West Chester, acquired Beta Inc.," RE aims to identify relations such as ‘headquarteredIn(Acme Corp, West Chester)‘. This task is crucial for building the graph’s structure. Work in this area includes both Closed RE, for a predefined set of relation types, and Open Information Extraction (OpenIE), which extracts relations expressed with arbitrary text (Etzioni et al., 2008).

Approaches to RE have mirrored those in NER. Early systems were rule-based, using linguistic patterns to identify relations (Hearst, 1992). Supervised statistical models followed, using classifiers trained on annotated data or data generated through distant supervision, a technique that aligns known relations from KGs with text but can introduce noise (Kambhatla, 2004; Mintz et al., 2009). Modern deep learning approaches, especially Transformer-based models, now represent the state-of-the-art (S. Kumar, 2017; Wu & He, 2019). LLMs, through prompting techniques, offer a powerful alternative capable of extracting relations with minimal task-specific fine-tuning (Chia et al., 2022).

Key challenges in RE include handling ambiguity, extracting relations that span sentences, and adapting models to new domains. In the legal

context, extracting relations such as amendments, definitions, and obligations is critical for building a KG that accurately reflects the legal framework (Dhani et al., 2021; Tauqeer et al., 2022). The structured output from NER and RE forms the basis for the constructed knowledge graph.

2.5 Consistency, Completeness, and Coherence

When analyzing formal document corpora such as legal codes or technical standards, quality evaluation often involves assessing internal integrity. Three key aspects of this integrity are consistency, completeness, and coherence (Umar & Lano, 2024). These concepts, while sometimes overlapping, address distinct facets crucial for ensuring documents are understandable, unambiguous, and effective.

- **Consistency:** Refers to the absence of logical contradictions within a document set (Egyed, 2006; Guo et al., 2023; Heitmeyer et al., 1996; Nentwich, 2005; Tröls et al., 2022; Yang et al., 2024; Zowghi & Gervasi, 2003). A consistent document should not contain provisions that assert mutually exclusive facts or prescribe conflicting obligations under identical conditions. Detecting such inconsistencies is vital for legal certainty and predictability (Donelson, 2019; Duck-Mayr, 2022; Rossi, 2016). Formal logic and automated reasoning techniques are often employed to check consistency in formal specifications (Brucker & Wolff, 2019; Heitmeyer et al., 1996).

- **Completeness:** Pertains to whether the document set contains all necessary information relative to its intended scope (Zowghi & Gervasi, 2003). Defining completeness is inherently challenging, as it depends on a clear specification of what should be included. In a

legal context, completeness may require that all terms are adequately defined, referenced procedures are specified, and relevant scenarios are addressed. Gaps or omissions can lead to ambiguity and disputes. Assessing completeness often requires significant domain knowledge and may involve checking against predefined templates or requirements specifications (Umar & Lano, 2024; Zowghi & Gervasi, 2003). The interpretation of completeness in KGs is also affected by the "Closed World Assumption" versus the "Open World Assumption" (Hitzler, Kr otzsch, & Rudolph, 2009a; Reiter, 1978).

- **Coherence:** Relates to the overall understandability, organization, and logical flow of the presented information (Shen et al., 2021; Wang & Guo, 2014). A coherent document is well-structured, uses terminology consistently, and ensures cross-references are accurate. While related to consistency, coherence focuses more on clarity and comprehensibility for a human reader, encompassing aspects like lexical cohesion and discourse structure (Wang & Guo, 2014).

Ensuring these three qualities simultaneously in large, evolving legal codes through traditional manual review is exceptionally difficult. The volume of text, the intricate web of interdependencies, the potential for ambiguity in natural language, and the distributed nature of authorship make manual detection of subtle flaws challenging and error-prone (Beth, 2018). Computational approaches that leverage structured representations like KGs offer significant potential advantages.

A Knowledge Graph provides a structured substrate amenable to automated analysis by explicitly modeling entities and their relationships. Graph-based queries and algorithms can be designed to automatically detect potential inconsistencies, such as conflicting property values or circular

definition chains (Brucker & Wolff, 2019; Schönberg et al., 2011; Tauqeer et al., 2022; Weitl & Freitag, 2006). While perfect completeness verification is often intractable for natural language documents, KGs can help identify potential gaps by analyzing the graph's structure for missing nodes, absent relationships, or orphaned sections (Knublauch & Kontokostas, 2017; Omran et al., 2020; Rabbani et al., 2022, 2023; Umar & Lano, 2024).

LLMs can play a role throughout this pipeline, aiding in the initial interpretation of text to populate the KG, helping to formulate complex graph queries, or summarizing the findings from the analysis for human review (Benjira et al., 2025). The KG itself, however, provides the persistent, globally coherent, and computationally tractable structure necessary for systematic integrity checks. Such a structure can overcome the context window limitations and potential lack of deterministic reasoning inherent in LLMs alone. Research exploring KGs and related AI techniques for automated integrity checking provides a foundation for this approach (Aumiller et al., 2021; Dhani et al., 2021; Heitmeyer et al., 1996; Tauqeer et al., 2022; Umar & Lano, 2024). This praxis project aims to build upon such work by investigating the practical application of LLM-driven KG construction for analyzing municipal legal codes.

2.6 Challenges in Analyzing Large Documents

Research in automated document processing is extensive, covering tasks like summarization, information extraction, and question answering (D. Chen et al., 2017; Gambhir & Gupta, 2017; Zhong et al., 2024). Historically, much of this research focused on relatively small documents, as they are less computationally demanding and more feasible for the human evaluation required to establish ground truth.

Many critical real-world applications, however, involve documents that are orders of magnitude larger, such as legal contracts, technical manuals, and extensive regulatory codes. Analyzing these large documents presents distinct and significant challenges:

- **Computational Resources:** Processing large volumes of text demands substantial memory, storage, and processing time. The computational complexity often scales non-linearly with document length, making naive processing of entire large documents infeasible (Vaswani et al., 2017).
- **Long-Range Dependencies:** Comprehension frequently requires capturing semantic connections or references between sections that are far apart in the document. Models with limited context windows struggle to capture these long-distance relationships accurately (Liu et al., 2025; Zhao et al., 2023).
- **Context Fragmentation:** A common technique for handling large documents involves splitting them into smaller chunks. This method, while necessary for models with fixed inputs, risks losing critical context that spans chunk boundaries, potentially leading to fragmented understanding (T. Chen et al., 2024; Qu et al., 2024).
- **Evaluation Complexity:** Assessing the quality of automated analysis on a large document is inherently difficult for human evaluators. Establishing reliable ground truth for evaluation benchmarks remains a major challenge for large-document tasks (Shaham et al., 2022).

Techniques like Retrieval-Augmented Generation (RAG) allow LLMs to leverage information from large external corpora without processing the

entire corpus in their context window (Lewis et al., 2020). RAG retrieves relevant text snippets and provides them as context to the LLM. Although powerful for knowledge-intensive tasks, standard RAG retrieves discrete, localized chunks and may not provide the holistic, structured view of the entire document that a pre-constructed Knowledge Graph can offer.

2.7 Challenges in Analyzing Legal Documents

Legal documents, particularly statutory codes, represent a compelling yet challenging domain for advanced document analysis techniques. They possess several intrinsic characteristics that make them difficult testbeds and valuable targets for automation:

- **Complexity and Precision:** Legal language is dense, employing specialized terminology and complex sentence structures. Ambiguity must be minimized, demanding high precision in interpretation, as misinterpretations can have significant real-world consequences (Ashley, 2017; Malik et al., 2022).
- **Volume and Interconnectedness:** Legal corpora can be vast and are highly interconnected through citations, amendments, and definitions. Understanding one part often requires understanding its relationship to many others (Beth, 2018).
- **Semi-structured Format:** Legal texts mix structured elements (e.g., sections, clauses) with unstructured natural language prose, requiring sophisticated NLP techniques to handle both.
- **Critical Need for Integrity:** The consistency, completeness, and coherence of legal documents are paramount for their function. These

qualities underpin the rule of law, ensuring predictability and enforceability. Flaws can lead to disputes, costly litigation, and erosion of public trust (Donelson, 2019; Duck-Mayr, 2022; Rossi, 2016).

The specific focus on the codified ordinances of Pennsylvania townships provides a valuable and concrete dataset. These codes exhibit realistic complexity, having often been developed over decades with multiple authorships and numerous amendments. The legislative drafting and codification process, although designed to ensure quality through multi-stage human review, can still introduce errors. Inconsistencies, incompleteness, and incoherence can persist as the code grows (Rossi, 2016). The resource-intensive and fallible nature of purely manual review motivates the exploration of computational methods to assist legal professionals in maintaining the integrity of these foundational legal documents.

2.8 Related Work

Research relevant to this praxis project spans several areas: utilizing LLMs for Information Extraction, applying KGs for document integrity checking, and the application of NLP to the legal domain.

LLMs for Information Extraction and KG Construction: The advent of powerful LLMs has advanced information extraction. Studies demonstrate the ability of models like GPT-4, often via prompting, to perform NER and RE with performance rivaling or exceeding traditional fine-tuned models, especially in specialized domains (Xu et al., 2024). Researchers have explored various techniques to mitigate LLM limitations like hallucinations and to control output structure (S. Wang et al., 2023). Several works focus on constructing KGs from text using LLMs as the primary extraction engine,

developing pipelines that integrate entity identification, relation extraction, and schema mapping, sometimes with human-in-the-loop refinement (Benjira et al., 2025; Lairgi et al., 2024).

KGs for Document Analysis and Integrity Checking: Beyond construction, KGs serve as a substrate for advanced document analysis, including semantic search and complex question answering (Hogan et al., 2021; Ji et al., 2022). Directly relevant to this work is the use of KGs for consistency and completeness checking. In software requirements engineering, KGs and ontologies model requirements to detect conflicts (Umar & Lano, 2024). In the Semantic Web community, technologies like SHACL (Shapes Constraint Language) provide a standard way to validate RDF KGs against predefined constraints, effectively checking aspects of data integrity (Knublauch & Kontokostas, 2017).

AI and NLP for Legal Document Analysis: The legal domain has been a target for AI and NLP research for decades (Ashley, 2017). Recent research applies modern NLP to tasks like legal information retrieval, case outcome prediction, and contract review (Aletras et al., 2016; Chalkidis et al., 2022; Moens, 2001). Information extraction from legal texts has received significant attention, focusing on extracting citations, legal entities, and obligations (Kalamkar et al., 2022; Tauqeer et al., 2022). Some prior work has explored automated consistency checking in legal documents, often using rule-based or logic-based approaches, but these efforts have typically focused on specific types of conflicts rather than a comprehensive KG-based approach applied to municipal codes (Rossi, 2016).

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, the specific

contribution here lies in the integration and practical application of modern LLMs to construct KGs from municipal legal codes for consistency and completeness analysis. The project addresses LLM context window limitations by leveraging the KG structure for global reasoning. Unlike prior legal AI work focusing on case law or contracts, this project targets foundational legislative texts at the local government level. The "praxis" aspect emphasizes developing and evaluating a practical methodology to assist municipal professionals in maintaining the quality of their codified laws.

2.9 Conclusions

This chapter surveyed key literature relevant to utilizing LLMs and KGs for analyzing legal documents. The review traced the rise of LLMs from the Transformer architecture, noting their capabilities alongside their critical context window limitations (Liu et al., 2025; Vaswani et al., 2017). Knowledge Graphs were introduced as a structured paradigm capable of overcoming these limitations for global analysis (Hogan et al., 2021). The discussion covered Information Extraction as the bridge from unstructured text to structured KGs, detailing the roles of NER and RE (Xu et al., 2024).

The application was framed by defining the quality attributes of consistency, completeness, and coherence (Zowghi & Gervasi, 2003). The challenges of maintaining these qualities manually in complex legal codes motivate the need for computational assistance (Beth, 2018). This project is situated within research on LLM-driven IE, KG-based analysis, and legal AI, but is distinguished by its novel integration and practical focus on consistency and completeness checking for municipal ordinances.

The limitations of LLMs for global document understanding and the

structure offered by KGs, combined with the critical need for ensuring legal code integrity, motivate the methodology of this praxis project. By leveraging LLMs to extract 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. The following chapter will detail the specific methodology employed to achieve this objective.

Chapter 3: Methodology

3.1 Introduction

This chapter details the systematic methodology for developing and evaluating a system that leverages Large Language Models (LLMs) to convert extensive legal documents into attributed knowledge graphs (KGs) (Fensel et al., 2020; Hogan et al., 2021). As established in Chapter 1, these KGs are intended to serve as a foundation for subsequent analysis of document completeness and consistency (Umar & Lano, 2024; Zowghi & Gervasi, 2003). The challenge of managing the complexity and scale of modern textual documents, particularly in legal codification, necessitates automated solutions (Bhattacharya et al., 2019). Traditional manual review is often insufficient for ensuring comprehensive consistency in large, evolving document sets. This research posits that KGs, by providing structured representations of entities and their relationships, offer a viable approach to address these challenges (Zhong et al., 2024).

The methodology herein is designed to be rigorous and reproducible. It outlines the overall research approach, system architecture, and data sources—specifically, Pennsylvania township laws. The chapter then details the critical hyperparameters governing the system, the experimental plan for their tuning, the software tools utilized, and the multi-faceted strategy for evaluating the quality of the generated KGs in the absence of a definitive ground truth. Finally, the inherent limitations of the chosen methods and relevant ethical considerations are examined, concluding with a restatement of the research questions and hypotheses this framework aims to address.

3.2 Approach

This research introduces a novel, multi-stage pipeline for transforming large, complex legislative documents into queryable Knowledge Graphs (KGs). By leveraging Large Language Models (LLMs) (Benjira et al., 2025; Lairgi et al., 2024), our system is designed to facilitate automated analysis of legal texts for internal consistency and completeness. The architecture prioritizes two critical features: **usability**, requiring minimal specialized skill for operation, and **traceability**, ensuring every piece of information in the KG can be precisely linked to its source sentence. While developed using legal corpora, the modular design is generalizable to other domains characterized by large, structured documents (Shaham et al., 2022).

3.2.1 Architectural Overview

The system employs a modular pipeline that synthesizes established KG construction practices (Ji et al., 2022) with a novel, two-tiered memory architecture adapted for the legal domain. This design manages the inherent complexity and context window limitations of modern LLMs (Liu et al., 2025) by building the KG iteratively. The data flows through a sequence of components, from initial document processing to advanced semantic enrichment. This modularity permits targeted optimization and evaluation at each stage. Figure 3.1 provides a high-level depiction of this data flow.

3.2.1.1 Document Ingestion and Chunking (P1)

The pipeline begins by ingesting source documents and segmenting them into semantically coherent units, or *chunks*. This chunking process is a critical preprocessing step to overcome the fixed context window of

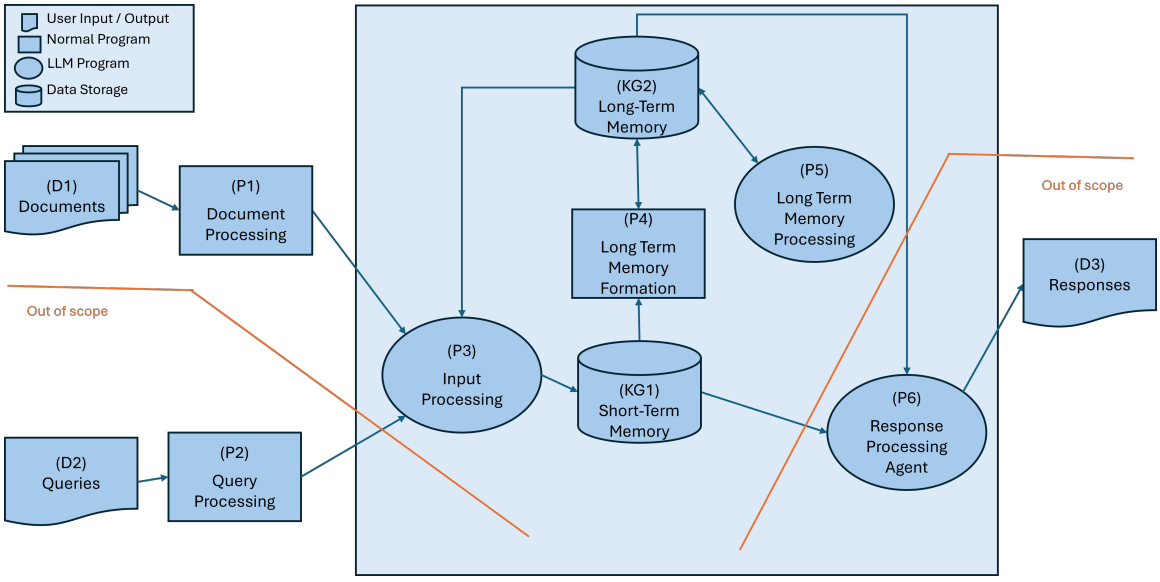


Figure 3.1: *High-Level Overview of the Document-to-KG Pipeline*

LLMs. A naive split can sever key semantic links; therefore, this component implements and evaluates several intelligent strategies, including fixed-size token windows, paragraph-based segmentation, and overlapping chunks. For hierarchically structured legal documents, a structure-aware strategy leverages the heading styles in DOCX files to ensure chunks do not violate logical boundaries (e.g., articles, sections). Each chunk is meticulously annotated with location metadata (e.g., page, section number) to guarantee end-to-end traceability.

3.2.1.2 LLM-Based Information Extraction (P3)

This component serves as the core information extraction engine. For each text chunk, a carefully engineered prompt instructs an LLM to perform two coordinated tasks: Named Entity Recognition (NER) and Relation Extraction (RE). The model identifies entities corresponding to a predefined legal ontology (e.g., `Ordinance`, `DefinedTerm`, `ZoningDistrict`) and the semantic relationships between them (e.g., `AMENDS`, `DEFINES`, `PERMITS_IN`).

To ensure structured, reliable output, the LLM's generation is constrained to a predefined JSON schema. The result for each chunk is a self-contained KG fragment, a design that enables parallel processing for enhanced throughput.

3.2.1.3 Short-Term Memory (STM): Batch Aggregation (KG1)

As KG fragments are generated, they are temporarily held in the Short-Term Memory (STM), an in-memory buffer. The STM functions as an efficient staging area, accumulating fragments until a predefined batch size is reached (the "STM fullness threshold" hyperparameter). This batching strategy optimizes performance by reducing the I/O overhead associated with numerous small writes to the persistent database. It also creates an opportunity for low-cost, preliminary consolidation operations before committing the data to long-term storage.

3.2.1.4 LTM Ingestion and Entity Resolution (P4 & KG2)

Once the STM reaches capacity, the Long-Term Memory (LTM) Ingestion process is triggered. This blocking operation transfers the batch of KG fragments from the STM into the persistent graph database, which constitutes the LTM. The LTM is implemented in Neo4j, a database optimized for highly interconnected data. A crucial step in this process is *entity resolution*, which de-duplicates nodes by merging new entities with existing ones based on unique identifiers or high lexical similarity. For example, multiple mentions of "Ordinance 2024-05" are resolved into a single canonical node. This ensures the global KG remains consistent and serves as the definitive, structured representation of the source documents.

3.2.1.5 Asynchronous KG Refinement and Enrichment (P5)

The final stage is an asynchronous engine that performs computationally intensive refinement operations to enhance the global KG's semantic richness. This process operates iteratively on small, random subsets of the graph, allowing the KG to improve gradually without blocking ingestion or requiring complete reprocessing. This stage is triggered only when the primary ingestion pipeline is idle. Key operations include:

- **Ontological Classification:** Formalizing type hierarchies by asserting `ISA` relationships (e.g., "Ordinance_123" `ISA` "Ordinance"), essential for semantic reasoning (Minsky, 1974; Noy & McGuinness, 2001).
- **Meronymic Relationship Modeling:** Establishing the document's structure within the graph by adding part-whole (`PARTOF`) relationships (e.g., "Section_3.B" `PARTOF` "Article_III").
- **Type System Refinement:** Identifying and merging semantically equivalent entity types (e.g., consolidating "Regulation" and "Rule") by calculating the similarity of their LLM-generated embeddings (Gardazi et al., 2025).
- **Ontology Organization:** Structuring entity types into a coherent subsumption hierarchy using a combination of LLM-based classification and domain heuristics (Tian et al., 2022).
- **Instance Type Correction:** Re-classifying entity instances based on new evidence or a more appropriate type within the evolving ontology.

This two-tiered memory architecture effectively decouples initial data aggregation from deeper semantic enrichment, creating a robust and scalable framework for KG construction.

3.3 Document Processing Workflow

The journey of a single document through the system is a multi-stage process designed to incrementally build and refine a knowledge graph (KG). This workflow is illustrated in the sequence of diagrams from Figure 3.2 to Figure 3.5. Each figure captures a snapshot of the system's state, showing how data is transformed at each step.

The process begins with the Document Processing stage, where an incoming document is deconstructed into a series of smaller, manageable chunks. These chunks are then processed sequentially. As shown in Figure 3.2, the first chunk is passed to the Input Processor. This component analyzes the text to extract key entities and their relationships, generating a KG fragment. This initial fragment is then stored in the Short-Term Memory (STM), which, as the diagram illustrates, was empty prior to this operation.

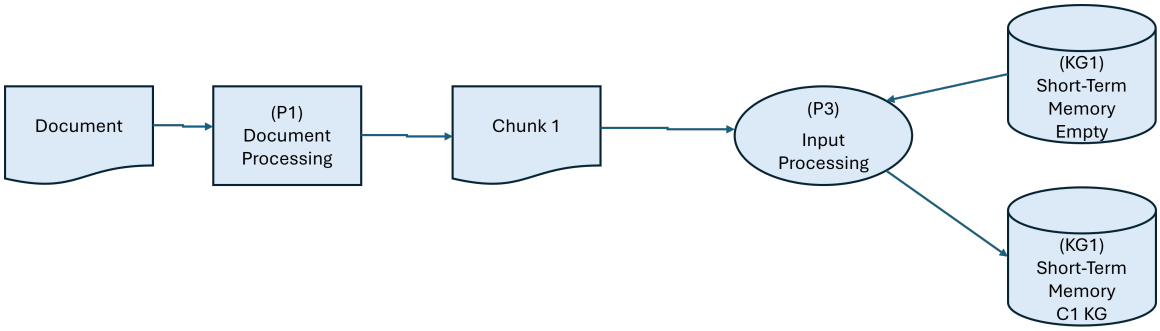


Figure 3.2: *Chunk 1 Processing*

Following the successful processing of the first chunk, the system proceeds to the next one. Figure 3.3 depicts this subsequent step. The Input Processor analyzes "Chunk 2" and generates a new KG fragment. This new fragment is then added to the Short-Term Memory. At this stage, the STM contains the distinct knowledge graphs from both the first and second chunks. While basic duplicate detection may occur, the primary goal is rapid ingestion, so

the STM accumulates these fragments as largely separate graphs.

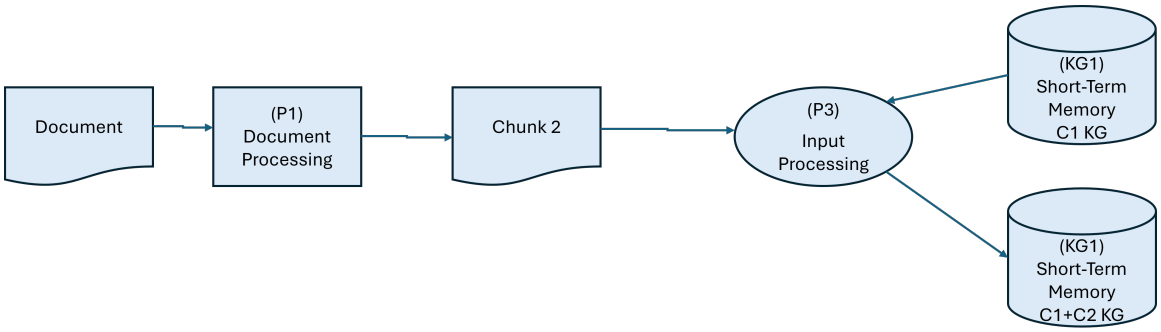


Figure 3.3: *Chunk 2 Processing*

This process of chunk-by-chunk analysis continues until the STM reaches its designated capacity or the document processing is complete. At this point, the LTM Formation process is triggered, as illustrated in Figure 3.4. The entire collection of KG fragments held in the STM is transferred to be consolidated. This consolidation phase merges the individual fragments, resolves redundancies, and integrates the new information into the persistent Long-Term Memory (LTM). Once the transfer and consolidation are complete, the STM is cleared, making it ready to process the next chunk or document.

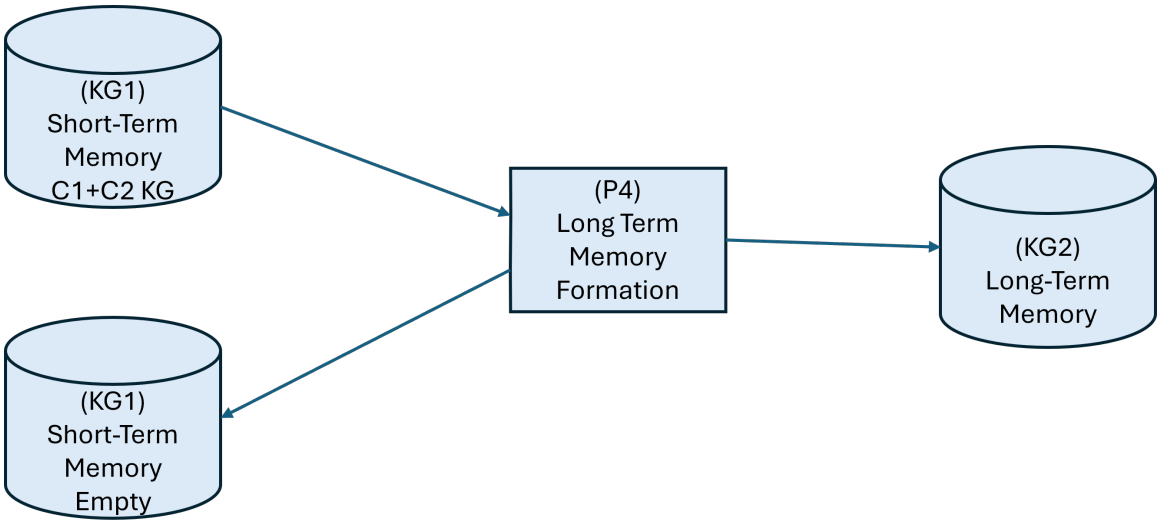


Figure 3.4: *Full Short Term Memory Processing*

Finally, the LTM is not a static repository. As depicted in Figure 3.5, an asynchronous LTM Processing engine operates continuously in the background. This engine performs computationally intensive refinement tasks on the global KG stored in the LTM. These tasks include advanced entity disambiguation, inference of new relationships, and overall structural optimization of the graph. This continuous refinement ensures that the knowledge base becomes more accurate, coherent, and semantically rich over time.

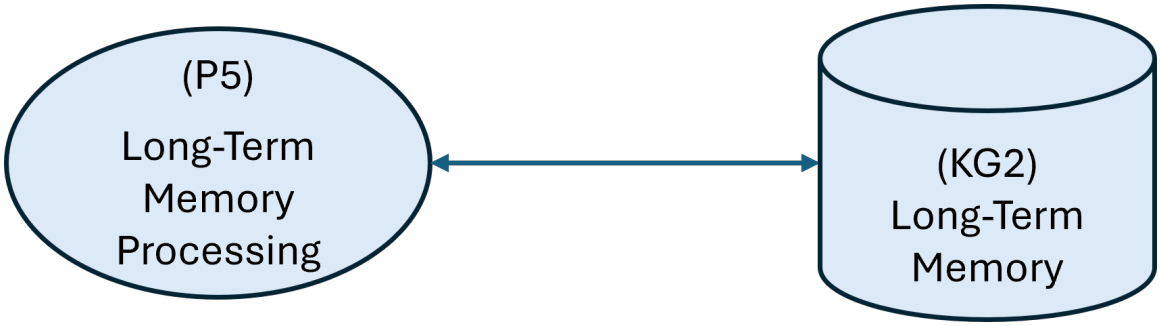


Figure 3.5: *Long Term Memory Processing*

In summary, this entire workflow represents a cyclical and scalable approach to knowledge extraction and management. The division between a fast, transient Short-Term Memory and a persistent, refined Long-Term Memory allows the system to process information efficiently without sacrificing the quality and coherence of the final knowledge graph. The initial, rapid ingestion into STM ensures that incoming data is captured without delay, while the subsequent consolidation and asynchronous refinement processes guarantee that the LTM evolves into a comprehensive and accurate representation of the knowledge contained within the source documents. This dual-memory architecture is key to balancing the demands of real-time processing with the need for deep, semantic integration.

3.4 Knowledge Graph Data Model for Legal Text Analysis

A robust data model is foundational to the development of a Knowledge Graph (KG) capable of representing complex legal corpora and facilitating sophisticated analytical queries. This section details the proposed ontological framework for a Neo4j-based Legal TTP (LTM), designed to capture the semantic structure of municipal legal codes. The model's schema is engineered for both representational fidelity to the source text and the computational efficiency required for automated consistency and completeness validation.

3.4.1 Ontological Framework: Node Schema

The core entities within the legal documents are modeled as nodes, each assigned one or more labels that denote its classification. Nodes possess a set of properties that store their specific attributes, extracted directly from the text. The primary node labels and their associated properties are enumerated in Table 3.1.

3.4.2 Semantic Relations: Edge Schema

The semantic connections and interactions between nodes are represented by directed, typed edges, formally known as relationships. The type of each relationship specifies the precise nature of the connection between the source and target nodes. Table 3.2 presents the defined relationship types.

3.4.3 Model Application: Inconsistency Detection

This ontological structure provides the necessary foundation for executing complex graph-based queries to automatically identify potential inconsistencies within the legal corpus. For instance, a critical validation is to

Table 3.1: *Node Schema and Property Definitions.*

Node Label	Description	Key Properties
`DocumentSection`	A structural component of the legal code, such as a chapter, article, or section, serving as a container for legal rules.	`id`, `title`, `textContent`, `sourceLocation`
`DefinedTerm`	An explicit definition of a specialized term provided within the legal text.	`term`, `definitionText`, `sourceLocation`
`Ordinance`	A formal legislative act, such as a law or an amendment, that creates, modifies, or repeals portions of the code.	`ordinanceNumber`, `enactmentDate`, `title`
`ZoningDistrict`	A geographically delineated area subject to a specific set of land-use regulations.	`districtName`, `districtCode`
`PermittedUse`	A specific activity, function, or land use that is legally sanctioned within a given zoning district, potentially under certain conditions.	`useName`, `conditions`
`Obligation`	A deontic expression imposing a legal duty or permission on a specific actor (e.g., a requirement to obtain a permit).	`actor`, `action`, `modality` (e.g., MUST, MAY, NOT_PERMITTED)
`Reference`	A citation pointing to another legal provision, either internal or external to the current code.	`targetIdentifier`, `referenceType` (e.g., INTERNAL, EXTERNAL), `sourceText`

Table 3.2: *Relationship Schema and Semantic Descriptions.*

Relationship Type	**Description (Source → Target)**	**Properties**
CONTAINS	DocumentSection → DocumentSection. Models the hierarchical structure of the code (e.g., a chapter contains a section).	–
DEFINES	DocumentSection → DefinedTerm. Links a section of text to a term it formally defines.	–
AMENDS	Ordinance → DocumentSection. Signifies that a law modifies a specific part of the code.	effectiveDate
CITES	DocumentSection → Reference. Indicates that a section of the code makes a citation.	–
PERMITS	ZoningDistrict → PermittedUse. Specifies that a land use is allowed within a particular district.	–
IMPOSES	DocumentSection → Obligation. Connects a legal rule to the specific duty or permission it creates.	–
USES_TERM	DocumentSection → DefinedTerm. Denotes the usage of a previously defined term within a section.	–

detect the use of terms that have not been formally defined. This scenario corresponds to a `DocumentSection` node that has an outgoing `USES_TERM` relationship to a `DefinedTerm` node, but where that `DefinedTerm` node lacks an incoming `DEFINES` relationship from any `DocumentSection`.

Such a query can be expressed declaratively using a graph pattern matching language like Cypher. The pattern to identify an undefined term (`dt`) being used in a section (`ds`) would be:

```
MATCH (ds:DocumentSection)-[:USES_TERM]->(dt:DefinedTerm)
WHERE NOT EXISTS ((:DocumentSection)-[:DEFINES]->(dt))
RETURN ds.id, dt.term
```

The execution of this and similar queries across the KG enables a systematic and scalable methodology for auditing the logical coherence of the legal code, thereby demonstrating the analytical utility of the proposed data model.

3.5 Dataset and Pre-processing

This research requires a corpus of large, complex documents to rigorously test the proposed methodology. The dataset must be readily accessible and available in a format that permits modification for experimental purposes, such as error injection tests.

3.5.1 Dataset Selection and Rationale

The primary dataset comprises the codified ordinances of various townships within the Commonwealth of Pennsylvania. This choice is predicated on several factors aligned with the research objectives. These documents are voluminous and linguistically complex, presenting a suitable challenge. Their

public availability simplifies acquisition and supports open research principles. Legal codes contain rich interrelationships—such as cross-references, definitions, and amendments—that are well-suited for KG representation. The dataset directly addresses the problem of inconsistency in municipal laws identified in Chapter 1 (Curley, 2024; Rau, 2024). Finally, the researcher's domain familiarity aids in the interpretation of legal nuances.

3.5.2 Data Acquisition and Pre-processing

The township law documents will be acquired from online public sources, primarily the legal code aggregator eCode360 (Sanders, 2024). The documents will be obtained in DOCX format, which is preferred over PDF because it preserves structural information (e.g., heading styles, lists, tables) that is advantageous for developing semantically aware chunking strategies.

An initial Exploratory Data Analysis (EDA) will be conducted on a representative sample of 5-10 municipal codes. This EDA will serve to:

- Characterize the typical document structure, length, and complexity.
- Identify the most common entity types and relationship patterns.
- Analyze linguistic features, such as the prevalence of legal jargon and complex sentence structures.

The findings from this EDA will directly inform the refinement of the KG data model, the design of chunking strategies, and the formulation of effective LLM prompts. Following the EDA, the pre-processing pipeline will be finalized. It will involve extracting the core textual content, removing irrelevant artifacts (e.g., headers, footers, page numbers), and normalizing the text (e.g., handling special characters, standardizing case) to create a clean corpus for processing.

3.6 Hyperparameters

The performance and behavior of the pipeline are governed by several critical hyperparameters. The systematic tuning of these parameters is essential for optimizing the system for accuracy and efficiency. The experimental plan involves varying these parameters and evaluating their impact using the metrics defined in Section 3.7. A summary of the parameters to be tuned is presented in Table 3.3.

The foundational choice is the **LLM Model**. Various models, including those from the Google Gemini, OpenAI GPT, and Anthropic Claude families, will be evaluated to find the optimal balance of extraction accuracy, adherence to structured output formats, and operational cost within the specialized legal domain. The **LLM Temperature** will be tuned within a low range (e.g., 0.0 to 0.7) to favor deterministic, factual extraction over creative but potentially inaccurate outputs, with the goal of minimizing hallucinations (Rzepka et al., 2023). Furthermore, **Prompt Design** will be an iterative process of refinement to improve clarity, specificity, and the inclusion of few-shot examples to better guide the model's NER and RE tasks.

Document Chunking strategies are vital for managing the context window limitations of LLMs (Ratner et al., 2022). Different strategies will be tested to determine which best preserves semantic context across chunk boundaries (Qu et al., 2024). For paragraph-based chunking, the number of **Paragraphs per Chunk** will be varied to find the ideal semantic unit size. For overlapping strategies, the **Overlap Percentage** will be adjusted to manage the trade-off between context continuity and redundant processing. The absolute **Chunk Size** in tokens will also be tuned to maximize the local context available to the selected LLM.

Finally, **Processing** parameters control the asynchronous consolidation and refinement of the KG. The **STM Fullness Threshold** dictates how many KG fragments are batched before being moved to the LTM, balancing data freshness with processing overhead. The **LTM Formation Batch Size** controls the resource intensity of these consolidation cycles. Similarly, the **LTM Processing Batch Size** determines the number of nodes to be refined in each asynchronous cycle, balancing processing depth with computational cost. The **Frequency of LTM Processing** controls the overall rate of iterative refinement of the global KG.

Table 3.3: *Hyperparameter Categories and Specific Parameters.*

Category	Parameter
LLM Used	LLM Model LLM Temperature Prompt Design
Document Chunking	Strategy Paragraphs per Chunk Overlap Percentage Chunk Size (Tokens)
Processing	STM Fullness Threshold (Number of KGs) LTM Formation Batch Size LTM Processing Batch Size (Number of Nodes) Frequency of LTM Processing

3.7 System Implementation and Environment

The methodology will be implemented using a standard, robust set of tools for AI and NLP research, ensuring reproducibility and leveraging the strengths of the current software ecosystem.

- **Core Language:** Python (version 3.8+) will be the primary program-

ming language due to its extensive libraries for data science, NLP, and API integration.

- **Development Environment:** Google Colaboratory will be used for development and experimentation, providing access to necessary computational resources, including GPUs for potentially hosting smaller, open-source LLMs or for embedding generation.
- **LLM Interaction:** The LangChain framework (Zhao et al., 2023) will be utilized to manage interactions with various LLM APIs. LangChain provides useful abstractions for prompt management, output parsing, and chaining together different components of the pipeline.
- **Graph Database:** The Long-Term Memory (LTM) will be implemented using Neo4j (version 5.x), a native graph database. Its Cypher query language is highly expressive and well-suited for the complex pattern matching required for KG analysis and integrity checking (N. Kumar & Kumar, 2013).
- **Document Parsing:** The ‘python-docx‘ library will be used to parse the input DOCX files, allowing for the extraction of both textual content and structural information like headings.
- **Version Control:** All code and configuration files will be managed using Git and hosted on GitHub. This practice ensures version control, facilitates collaboration, and guarantees the reproducibility of the research.

3.8 Evaluation Measurements

Evaluating the quality of the generated KG is a non-trivial task, primarily due to the difficulty of establishing a "gold standard" ground truth for large, complex legal documents (Dhani et al., 2021). Therefore, the evaluation strategy is multi-faceted, combining quantitative, qualitative, intrinsic, and extrinsic measures to provide a holistic assessment of the KG's fidelity and utility. This strategy is summarized in Table Table 3.4.

3.8.1 Knowledge Graph Fit Assessment

This area of evaluation addresses how well the generated KG represents the source documents.

- **Entity/Relation Extraction Quality:** The core extraction quality will be measured using standard metrics of Precision, Recall, and F1-score. A sample of 100-200 document chunks will be manually annotated to create a ground-truth dataset. The system's output on this sample will be compared against the manual annotations. A target F1-score above 0.8 is set for key legal entity and relation types.
- **Content Coverage and Plausibility (Expert Review):** A qualitative review will be conducted by one or more individuals with legal domain expertise (ideally the researcher and a peer). They will inspect randomly selected subgraphs of the KG and compare them against the corresponding source text. The goal is to achieve high inter-rater agreement that key legal concepts are present, accurately represented, and logically connected.
- **Traceability:** This will be measured by randomly sampling 100 nodes

and relationships from the KG and verifying that their ‘source_location‘ property correctly links back to the precise text in the source document from which they were extracted. The target for this metric is over 95% accuracy to ensure verifiability.

- **KG Structural Integrity:** This will be analyzed by examining graph-level statistics before and after LTM Processing. Metrics will include node and edge distributions, graph density, and connectivity. The goal is to demonstrate that the LTM processing steps lead to a more coherent and well-formed graph structure (e.g., fewer disconnected components, more logical type hierarchies).

3.8.2 Suitability for Consistency and Completeness (C&C) Analysis

This assessment determines if the KG is structured in a way that supports the end goal of C&C analysis, even though the full implementation of C&C algorithms is designated as future work.

- **Presence of Requisite Information Types:** A checklist-based evaluation will be performed to ensure that the information types necessary for C&C checks (e.g., definitions, obligations, cross-references, deadlines) are being successfully extracted and represented in the KG.

- **Queryability for Integrity Patterns:** A set of predefined Cypher queries will be executed against the KG to test for common integrity patterns. Examples include queries to find: (1) terms used in the text but never defined, (2) sections referenced but not present, and (3) conflicting obligations applied to the same entity. The goal is to confirm that the KG structure supports such queries and returns plausible candidates for inconsistencies.

- **KG Schema/Ontology Quality:** The emergent ontology of entity types will be qualitatively assessed for its clarity, coherence, and appropriateness for the legal domain. This ensures the schema is robust enough to support downstream legal C&C applications.

3.8.3 Controlled Error Injection Experiment

To provide a more objective measure of the system's ability to facilitate error detection, a controlled experiment will be conducted. A "gold standard" clean legal document will be seeded with a variety of common legal document errors to create a synthetic ground truth (Mintz et al., 2009). A typology of errors to be injected includes:

- **Type 1: Contradictory Definition:** Defining the same term in two different ways in separate sections.
- **Type 2: Undefined Term:** Using a specialized legal term throughout a section without providing an explicit definition.
- **Type 3: Broken Reference:** Including a cross-reference to a section number that does not exist (e.g., "as described in Section 99.9").
- **Type 4: Conflicting Obligation:** Stating in one section that a permit "is required" for an activity and in another that it "is not required" under the same conditions.
- **Type 5: Incomplete Specification:** Mentioning that a fee is required but failing to specify the amount or the calculation method.

The primary metric will be the **Error Detection Rate (EDR)**, defined as the percentage of these injected errors that are identifiable through specific queries on the generated KG. The **Traceability of Error Indicators** will

also be measured to ensure that the KG elements flagging an error can be accurately traced back to the specific erroneous text in the source document.

Table 3.4: *Summary of Measurement Metrics.*

Category	Metric
Knowledge Graph Fit	Entity/Relation Extraction (Precision, Recall, F1-score) Content Coverage & Plausibility Traceability KG Structural Integrity
Suitability for C&C Analysis	Presence of Requisite Information Types Queryability for Integrity Patterns KG Schema/Ontology Quality
Error Injection	Error Detection Rate (EDR) Traceability of Error Indicators

3.9 Methodological Limitations

This research acknowledges several limitations. The system's accuracy is dependent on the capabilities of the underlying LLM, which can hallucinate or misinterpret nuanced text (Zhao et al., 2023). Document chunking inherently risks losing context that spans across chunks (Qu et al., 2024). KG evaluation involves subjective judgment in the absence of a comprehensive ground truth, and errors in early pipeline stages can propagate (Zhong et al., 2024). The scope of this praxis is limited to textual content (excluding complex tables, figures, and formulas), and the system is developed and evaluated on Pennsylvania township laws; performance on other legal document types may vary. Finally, this work focuses on generating a KG *suitable* for C&C checks; the implementation and evaluation of a fully automated C&C system is designated as future work (Zowghi & Gervasi, 2003).

3.10 Ethical Considerations

The research is guided by several ethical principles. The data consists of public records, and no private or sensitive information is processed. The system is intended as an assistive tool for legal experts, not a replacement for human judgment; the emphasis on traceability mitigates the risk of over-reliance and promotes transparency. The potential for LLMs to perpetuate biases present in their training data is acknowledged, although the narrow domain of legal texts may limit the impact of broader societal biases (Zhao et al., 2023). All intellectual property will be handled according to university policy, and the principles of responsible innovation will be followed throughout the research.

3.11 Conclusion

This chapter has presented a detailed, multi-faceted methodology for developing and evaluating an LLM-based system to convert large legal documents into KGs. The objective is to create a robust and verifiable foundation for the future analysis of document consistency and completeness. The approach specifies a modular multi-stage pipeline, a comprehensive data model, a systematic plan for hyperparameter tuning, and a rigorous evaluation strategy that combines quantitative metrics, qualitative reviews, and a controlled error injection experiment. By addressing inherent limitations and adhering to ethical considerations, this methodology provides a structured and defensible plan to investigate the research questions and test the hypotheses set forth in Chapter 1.

3.12 Research Questions

This methodology is designed to address the following research questions:

RQ1: Can an LLM be used to convert a large document into a knowledge graph?

RQ2: Can an LLM be used to process multiple knowledge graphs into a typed cluster of knowledge graphs.

RQ3: Can a typed cluster of knowledge graphs be used to check the source document for consistency and completeness?

3.13 Research Hypotheses

The research will test the following hypotheses:

H1: An LLM can be used to convert a large document into a knowledge graph.

H2: An LLM can be used to process multiple knowledge graphs into a typed cluster of knowledge graphs.

H3: A typed cluster of knowledge graphs can be used to check the source document for consistency and completeness.

Chapter 4: Results and Analysis

4.1 Introduction

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.

This is how you build and format the table. The caption is centered and italized and in the top. The table number is bolded. Please remove the red coloring i use to illutrate this.

Table 4.1 presents an overview of xyz and zkm and xyz xyz. 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

Table 4.1: *Title of the table every first letter capitalized*

Factor1	Test 1	Test 2
Something here	123	123
Something here	123	123
Something here	123	1123
Something here	16	123
Something here	123	123
Something here	123	123

This is the correct way to add and format an equation. Ensure the formatting is consistent, and remove any red coloring used for illustration.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

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..

This is the correct way to add and format an equation. Ensure the formatting is consistent, and remove any red coloring used for illustration.

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

This is the correct way to build and format a table. The caption should be centered, italicized, and placed at the top. The table number should be bolded. In this example, the table is long, so ensure the table font matches the font of your document. It is not allowed to significantly reduce the table font size just to make it fit. In this case, I used a slightly smaller font, depending on the situation—only if necessary. However, try to avoid having too many columns to maintain readability. Make sure to remove the red coloring used for illustration.

Table 4.2 depicts the xxxx.

Table 4.2: *Test-2: Transformer vs. AutoTAB Performance Metrics*

Model	Method	A	P	R	F1	AUC	FNR	FPR
Mod1	Sub1	**0.0123**	**0.0123**	**0.0123**	**0.0123**	**0.0123**	**0.0123**	**0.0123**
	Sub2	0.0123	0.0123	0.0123	0.0123	0.0123	0.0123	0.0123
	Sub3	0.0123	0.0123	0.0123	0.0123	0.0123	0.0123	0.0123
Mod2	Sun1	0.0123	0.0123	0.0123	0.0123	0.0123	0.0123	0.0123
	Sub2	0.0123	0.0123	0.0123	0.0123	0.0123	0.0123	0.0123
	Sub3	0.0123	0.0123	0.0123	0.0123	0.0123	0.0123	0.0123

4.1.1 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.

This is how you insert an image by referencing it like this: Figure 4.1 shows XYZ.

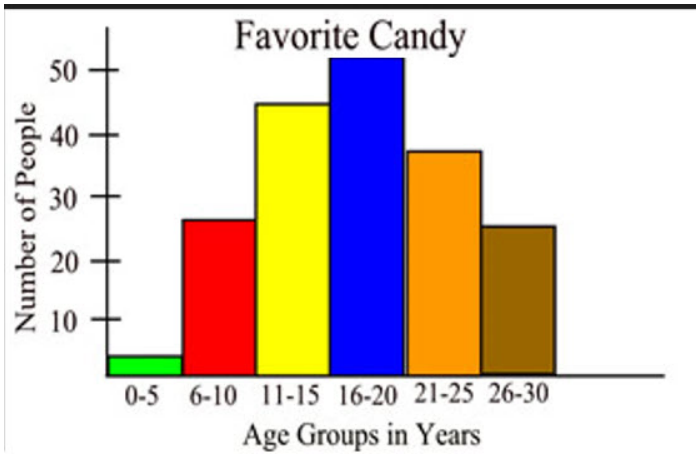


Figure 4.1: *Histogram of XYZ*

This is how to enter an equation and reference it. Equation 4.3 and 4.3 show how XYZ is implemented.

$$TE_{(pos,2i)} = sin(pos/23^{2i/Lm}) \tag{4.3}$$

$$KN_{(pos,2i+1)} = cos(pos/453^{2i/Lm}) \tag{4.4}$$

This is the correct format for a table, including the caption and proper alignment (centered and italicized).

Table 4.3: *My Table About Something*

Reference	Technique
something here	Technique here
something here	Technique here
something here	Technique here
something here	Technique here
something here	Technique here

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

Run the processes asynchronously Create specific LLM for each task Build the Query Processing Build the response processing agent Build the ability to validate documents Prepossessing the copra to identify initial Node Labels and Relationship Types

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.

Chapter 6: Test References

Reference 167 (Gardazi et al., 2025)

Reference 166 (Mochales Palau & Moens, 2009)

Reference 165 (Bhattacharya et al., 2019)

Reference 164 (Aletras et al., 2016)

Reference 162 (Lairgi et al., 2024)

Reference 160 (Xu et al., 2024)

Reference 159 (Ashley, 2017)

Reference 158 (Lewis et al., 2020)

Reference 157 (D. Chen et al., 2017)

Reference 156 (Gambhir & Gupta, 2017)

Reference 154 (Knublauch & Kontokostas, 2017)

Reference 153 (Omran et al., 2020)

Reference 152 (Rabbani et al., 2022)

Reference 151 (Rabbani et al., 2023)

Reference 149 (Hitzler, Kr otzsch, & Rudolph, 2009a)

Reference 148 (Reiter, 1978)

Reference 147 (Bosco, 2024)

Reference 146 (Sanders, 2024)

Reference 145 (Rau, 2024)

Reference 144 (Curley, 2024)

Reference 143 (Chia et al., 2022)

Reference 142 (Wu & He, 2019)

Reference 141 (S. Kumar, 2017)

Reference 140 (Mintz et al., 2009)

Reference 139 (Kambhatla, 2004)

Reference 138 (Agichtein & Gravano, 2000)

Reference 137 (Brin, 1998)

Reference 136 (Hearst, 1992)

Reference 135 (Noy & McGuinness, 2001)

Reference 134 (Etzioni et al., 2008)

Reference 132 (Luo et al., 2018)

Reference 131 (Lample et al., 2016)

Reference 130 (Lin et al., 2017)

Reference 128 (Lafferty et al., 2001)

Reference 127 (Grishman & Sundheim, 1996)

Reference 126 (Nadeau & Sekine, 2007)

Reference 125 (Kalamkar et al., 2022)

Reference 124 (Au et al., 2022)

Reference 122 (Kolluru et al., 2020)

Reference 121 (Zhong et al., 2024)

Reference 120 (Fensel et al., 2020)

Reference 118 (Ji et al., 2022)

Reference 117 (Sporny et al., 2025)

Reference 116 (Shaham et al., 2022)

Reference 115 (Ratner et al., 2022)

Reference 114 (Fernandes & Bernardino, 2018)

Reference 112 (Hartig et al., 2025)

Reference 111 (N. Kumar & Kumar, 2013)

Reference 110 (Hitzler, Krotzsch, & Rudolph, 2009b)

Reference 109 (“RDF 1.2 Primer,” 2025)

Reference 108 (Ehrlinger & W "o ss, 2016)

Reference 107 (Benjira et al., 2025)

Reference 106 (Tay et al., 2023)

Reference 105 (T. Chen et al., 2024)

Reference 104 (Qu et al., 2024)

Reference 103 (Schmidt et al., 2024)

Reference 102 (Hogan et al., 2021)

Reference 101 (Zhao et al., 2023)

Reference 100 (Kaplan et al., 2020)

Reference 99 (Liu et al., 2025)

Reference 97 (Cong, 2024)

Reference 96 (Rzepka et al., 2023)

Reference 95 (Grattafiori et al., 2024)

Reference 94 (Caruccio et al., 2024)

Reference 93 (Team et al., 2024)

Reference 92 (Gao et al., 2023)

Reference 90 (Turner, 2024)

Reference 89 (Badshah & Sajjad, 2024)

Reference 88 (Ye et al., 2024)

Reference 87 (S. Wang et al., 2023)

Reference 84 (Edwards, 2025)

Reference 81 (Vaswani et al., 2017)

Reference 78 (Minsky, 1974)

Reference 77 (Tauqeer et al., 2022)

Reference 76 (Dhani et al., 2021)

Reference 68 (Beth, 2018)

Reference 64 (Chalkidis et al., 2022)

Reference 62 (Malik et al., 2022)

Reference 57 (Carbonell et al., 2020)

Reference 54 (Rossi, 2016)

Reference 53 (Duck-Mayr, 2022)

Reference 52 (Donelson, 2019)

Reference 51 (P. Chen et al., 2021)

Reference 49 (K. Wang et al., 2024)

Reference 48 (Feng et al., 2024)

Reference 47 (Scarselli et al., 2009)

Reference 46 (Li & Chen, 2021)

Reference 45 (Moens, 2001)

Reference 44 (Wang & Guo, 2014)

Reference 35 (Verma, 2024)

Reference 30 (Guo et al., 2023)

Reference 29 (Umar & Lano, 2024)

Reference 28 (Yang et al., 2024)

Reference 27 (Tröls et al., 2022)

Reference 26 (Egyed, 2006)

Reference 25 (Nentwich, 2005)

Reference 24 (Brucker & Wolff, 2019)

Reference 23 (Weitl & Freitag, 2006)

Reference 21 (Heitmeyer et al., 1996)

Reference 19 (Schönberg et al., 2011)

Reference 14 (Shen et al., 2021)

Reference 13 (Laban et al., 2021)

Reference 11 (Aumiller et al., 2021)

Reference 10 (Zowghi & Gervasi, 2003)

Reference 7 (Gupta et al., 2021)

Reference 6 (Tian et al., 2022)

Reference 5 (Chaurasiya et al., 2022)

Reference 4 (Al-Moslmi et al., 2020)

References

Agichtein, E., & Gravano, L. Snowball : Extracting relations from large plain-text collections. In: In *Proceedings of the fifth acm conference on digital libraries (dl '00) (pp. 85–94)*. New York, New York, USA: Association for Computing Machinery, 2000, 85–94. https://doi.org/https://doi.org/10.1145/336597.336644

Aletras, N., Tsarapatsanis, D., Preotiuc-Pietro, D., & Lampos, V. (2016). Predicting judicial decisions of the european court of human rights: A natural language processing perspective. *PeerJ Computer Science*, *2*, e93. https://doi.org/10.7717/peerj-cs.93

Al-Moslmi, T., Gallofre Ocana, M., L. Opdahl, A., & Veres, C. (2020). Named entity extraction for knowledge graphs: A literature overview. *IEEE access*, *8*, 32862–32881. https://doi.org/10.1109/ACCESS.2020.2973928

Ashley, K. D. (2017). *Artificial intelligence and legal analytics* (Anonymous, Trans.). Cambridge University Press. https://doi.org/10.1017/9781316761380

Au, T. W. T., Cox, I. J., & Lampos, V. E-ner – an annotated named entity recognition corpus of legal text. In: In *Natural legal language processing workshop*. Association for Computational Linguistics, 2022, 246–255. http://arxiv.org/abs/2212.09306

Aumiller, D., Almasian, S., Lackner, S., & Gertz, M. Structural text segmentation of legal documents. In: In *Proceedings of the eighteenth international conference on artificial intelligence and law*. New York, NY, USA: ACM, 2021, 2–11. https://doi.org/10.1145/3462757.3466085

Badshah, S., & Sajjad, H. (2024). *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

Beth, R. S. (2018). *How bills amend statutes* (Explains some of the issues that can arise when a new law is in conflict with existing laws.). https://purl.fdlp.gov/GPO/gpo126602

Bhattacharya, P., Hiware, K., Rajgaria, S., Pochhi, N., Ghosh, K., & Ghosh, S. (2019). A comparative study of summarization algorithms applied to legal case judgments (Anonymous, Trans.). In, *Advances in information retrieval* (pp. 413–428, Vol. 11437). Springer International Publishing AG. https://doi.org/10.1007/978-3-030-15712-8_27

Bosco, A. (2024). *Leading to annual revenue losses of hundreds of thousands of dollars.* [I met with Alex Bosco, a Supervisor for Easttown Township, to discuss a recent Zoning Hearing Board (ZHB) decision. The ZHB had granted a waiver to a resident, exempting them from paying the established fee-in-lieu for sidewalk construction. Our discussion centered on the specifics of this case and its direct implications for the township. We examined the immediate financial cost to the municipality resulting from this single decision and the precedent it might set. Supervisor Bosco provided insight into the legislative and procedural challenges the Board of Supervisors faces when such variances are granted. We explored potential legislative remedies that the township could consider to prevent future fiscal strain and ensure consistent application of the law. This conversation is a pertinent case

study for my research, as it highlights the critical need for clarity and robustness in municipal ordinances. It underscores how an ambiguous or easily challengeable legal framework can lead to ad-hoc decisions that create financial and governance challenges, demonstrating the real-world impact of precisely engineered legislation.].

Brin, S. Extracting patterns and relations from the world wide web. In: In *International workshop on the world wide web and databases*. Springer International Publishing, 1998, 172–183.

Brucker, A. D., & Wolff, B. (2019). Using ontologies in formal developments targeting certification (Anonymous, Trans.). In, *Integrated formal methods* (pp. 65–82, Vol. 11918). Springer International Publishing. https://doi.org/10.1007/978-3-030-34968-4_4

Carbonell, M., Riba, P., Villegas, M., Fornes, A., & Llados, J. Named entity recognition and relation extraction with graph neural networks in semi structured documents. In: In *International conference on pattern recognition*. Piscataway: IEEE, 2020, 9622–9627. https://doi.org/10.1109/ICPR48806.2021.9412669

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

Chalkidis, I., Jana, A., Hartung, D., Bommarito, M., Androutsopoulos, I., Katz, D., & Aletras, N. Lexglue: A benchmark dataset for legal language understanding in english. In: In *60th annual meeting of the association for computational linguistics*. Association for Computational Linguistics, 2022. https://doi.org/10.18653/v1/2022.acl-long.297

Chaurasiya, D., Surisetty, A., Kumar, N., Singh, A., Dey, V., Malhotra, A., Dhama, G., & Arora, A. (2022). *Entity alignment for knowledge graphs: Progress, challenges, and empirical studies*. https://doi.org/10.48550/arxiv.2205.08777

Chen, D., Fisch, A., Weston, J., & Bordes, A. (2017). Reading wikipedia to answer open-domain questions. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. https://doi.org/10.18653/v1/p17-1171

Chen, P., Ding, H., Araki, J., & Huang, R. Explicitly capturing relations between entity mentions via graph neural networks for domain-specific named entity recognition (C. Zong, F. Xia, W. Li, & R. Navigli, Eds.). In: *59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing* (C. Zong, F. Xia, W. Li, & R. Navigli, Eds.). Ed. by Zong, C., Xia, F., Li, W., & Navigli, R. *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)*. Association for Computational Linguistics, 2021, 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. Dense x retrieval: What retrieval granularity should we use? In: In *2024 conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2024, 15159–15177. https://doi.org/10.48550/arxiv.2312.06648

Chia, Y. K., Bing, L., de Lichron, V., Lee, K., & Wong, K.-F. Relation extraction as open-book question answering: Evaluation on a comprehensive assessment dataset. In: In *Association for computational linguistics:*

Emnlp 2022. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, 2022, 6305–6319.

Cong, Y. Research for enhancing processing and computational efficiency in llm. In: In *Proceedings of the 2024 2nd international conference on image, algorithms and artificial intelligence (iciaai 2024)*. Atlantis Press, 2024, 970–980. https://doi.org/10.2991/978-94-6463-540-9_97

Curley, D. (2024). *Municipal laws in pennsylvania townships, authored by multiple people over time, develop inconsistencies and are incomplete* [While my capacity as an Easttown Township Supervisor affords me frequent interaction with Township Manager Don Curley, a specific discussion served as a key point of validation for my doctoral research. This personal communication focused on the systemic challenges encountered in maintaining the logical integrity and referential coherence of the municipal code. Drawing from his executive perspective as the individual responsible for implementing the Township's ordinances, Mr. Curley confirmed the operational difficulties that arise from ambiguities or contradictions within the legislative text. He articulated the significant administrative burden required to navigate these issues, which can impact staff resources, public clarity, and enforcement consistency. This discussion was pivotal, as it provided expert confirmation from a senior administrative officer regarding the real-world consequences of imperfect legislative drafting. It affirmed the practical exigency for the development of a computational framework, such as the one proposed in my research, to automate the detection of flaws and enhance the structural quality of municipal law. The conversation effectively bridged the gap between the theoretical

underpinnings of my research and the tangible governance challenges faced by municipalities.].

Dhani, J. S., Bhatt, R., Ganesan, B., Sirohi, P., & Bhatnagar, V. (2021). *Similar cases recommendation using legal knowledge graphs*. https://doi.org/10.48550/arxiv.2107.04771

Donelson, R. (2019). Legal inconsistencies. *Tulsa Law Review*, *55*(1), 16–44. https://digitalcommons.law.utulsa.edu/tlr/vol55/iss1/14

Duck-Mayr, J. (2022). Explaining legal inconsistency. *Journal of theoretical politics*, *34*(1), 107–126. https://doi.org/10.1177/09516298211061159

Edwards, B. (2025). *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/

Egyed, A. Instant consistency checking for the uml. In: In *Proceedings of the 28th international conference on software engineering*. New York, NY, USA: ACM, 2006, 381–390. https://doi.org/10.1145/1134285.1134339

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

Etzioni, O., Banko, M., Soderland, S., & Weld, D. S. (2008). Acm: Digital library: Communications of the acm. *Communications of the ACM*, *51*(12), 68–74. https://dl.acm.org/doi/fullHtml/10.1145/1409360.1409378

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*. https://doi.org/10.48550/arxiv.2405.00476

Fensel, D., Şimşek, U., Angele, K., Huaman, E., Kärle, E., Panasiuk, O., Toma, I., Umbrich, J., & Wahler, A. (2020). *Knowledge graphs : Methodology, tools and selected use cases* (Anonymous, Trans.; 1st ed.) [It is not available online. So, I ordered it from the library.]. Springer International Publishing. https://doi.org/10.1007/978-3-030-37439-6

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, 373–380. https://doi.org/10.5220/0006910203730380

Gambhir, M., & Gupta, V. (2017). Recent automatic text summarization techniques: A survey. *The Artificial intelligence review*, *47*(1), 1–66. https://doi.org/10.1007/s10462-016-9475-9

Gao, K., He, S., He, Z., Lin, J., Pei, Q., Shao, J., & Zhang, W. (2023). *Examining user-friendly and open-sourced large gpt models: A survey on language, multimodal, and scientific gpt models*. http://arxiv.org/abs/2308.14149

Gardazi, N. M., Daud, A., Malik, M. K., Bukhari, A., Alsahfi, T., & Alshemaimri, B. (2025). Bert applications in natural language processing: A review [Replaces 91.]. *The Artificial intelligence review*, *58*(6), 166. https://doi.org/10.1007/s10462-025-11162-5

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

Grishman, R., & Sundheim, B. Message-understanding conference-6: A brief history [COLING 1996 volume 1: The 16th international conference on computational linguistics]. In: In *The 16th international conference*

on computational linguistics. COLING 1996 volume 1: The 16th international conference on computational linguistics. 1996.

Guo, B., Feng, C., Liu, F., Li, X., & Wang, X. (2023). Joint contrastive learning for factual consistency evaluation of cross-lingual abstract summarization (Anonymous, Trans.). In, *Machine translation* (pp. 116–127, Vol. 1922). Springer Nature Singapore. https://doi.org/10.1007/978-981-99-7894-6_11

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). *Sparql 1.2 query language*. https://www.w3.org/TR/sparql12-query/

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. *Proceedings of the Fourteenth International Conference on Computational Linguistics, Nantes, Frans*.

Heitmeyer, C. L., Jeffords, R. D., & Labaw, B. G. (1996). Automated consistency checking of requirements specifications. *ACM transactions on software engineering and methodology*, *5*(3), 231–261. https://doi.org/10.1145/234426.234431

Hitzler, P., Kr otzsch, M., & Rudolph, S. (2009a). *Foundations of semantic web technologies* (Anonymous, Trans.). Chapman & Hall / CRC.

Hitzler, P., Krotzsch, M., & Rudolph, S. (2009b). *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

Ji, S., Pan, S., Cambria, E., Marttinen, P., & Yu, P. S. (2022). A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transaction on neural networks and learning systems*, *33*(2), 494–514. https://doi.org/10.1109/TNNLS.2021.3070843

Kalamkar, P., Agarwal, A., Tiwari, A., Gupta, S., Karn, S., & Raghavan, V. Named entity recognition in indian court judgments. In: In *Natural legal language processing workshop 2022*. Association for Computational Linguistics, 2022, 184–193. https://aclanthology.org/2022.nllp-1.pdf#page=199

Kambhatla, N. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In: In *Acl 2004 workshop on relation extraction*. Barcelona, Spain: Association for Computational Linguistics, 2004, 178–181.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020). *Scaling laws for neural language models*. http://arxiv.org/abs/2001.08361

Knublauch, H., & Kontokostas, D. (2017). *Shapes constraint language (shacl)* (Recommendation). https://www.w3.org/TR/shacl/

Kolluru, K., Aggarwal, S., Rathore, V., & Chakrabarti, S. Imojie: Iterative memory-based joint open information extraction (D. Jurafsky, J. Chai, N. Schluter, & J. Tetreault, Eds.). In: *58th annual meeting of the*

association for computational linguistics (D. Jurafsky, J. Chai, N. Schluter, & J. Tetreault, Eds.). Ed. by Jurafsky, D., Chai, J., Schluter, N., & Tetreault, J. Online: Association for Computational Linguistics, 2020, 5871–5886. https://doi.org/10.18653/v1/2020.acl-main.521

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, 2013, 1–6. https://doi.org/10.1109/ICCCNT.2013.6726698

Kumar, S. (2017). *A survey of deep learning methods for relation extraction*. https://doi.org/10.48550/arxiv.1705.03645

Laban, P., Dai, L., Bandarkar, L., & Hearst, M. A. Can transformer models measure coherence in text? re-thinking the shuffle test (C. Zong, F. Xia, W. Li, & R. Navigli, Eds.) [I had to change percent to % for LaTeX.]. In: *59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing* (C. Zong, F. Xia, W. Li, & R. Navigli, Eds.). Ed. by Zong, C., Xia, F., Li, W., & Navigli, R. I had to change percent to % for LaTeX. Online: Association for Computational Linguistics, 2021, 1058–1064. https://doi.org/10.18653/v1/2021.acl-short.134

Lafferty, J., Mccallum, A., & Pereira, F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: In *Icml*. 2001, 282–289.

Lairgi, Y., Moncla, L., Cazabet, R., Benabdeslem, K., & Cl'eau, P. Knowledge graph construction using large language models. In: In *Journee nationale sur la fouille de textes*. Lyon, France, 2024. https://hal.science/hal-04607294

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. Neural architectures for named entity recognition. In: In *Naacl-hlt*. arXiv, 2016, 260–270. http://arxiv.org/abs/1603.01360

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-T., Rocktäschel, T., Riedel, S., & Kiela, D. Retrieval-augmented generation for knowledge-intensive nlp tasks. In: In *Advances in neural information processing systems*. Curran Associates, Inc., 2020, 9459–9474. https://discovery.ucl.ac.uk/id/eprint/10100504

Li, H., & Chen, L. Cache-based gnn system for dynamic graphs. In: In *30th acm international conference on information & knowledge management*. Virtual Event, Queensland, Australia: ACM, 2021, 937–946. ISBN: 9781450384469. https://doi.org/10.1145/3459637.3482237

Lin, B. Y., Xu, F., Luo, Z., & Zhu, K. Multi-channel bilstm-crf model for emerging named entity recognition in social media (L. Derczynski, W. Xu, A. Ritter, & T. Baldwin, Eds.). In: *3rd workshop on noisy user-generated text* (L. Derczynski, W. Xu, A. Ritter, & T. Baldwin, Eds.). Ed. by Derczynski, L., Xu, W., Ritter, A., & Baldwin, T. Copenhagen, Denmark: Association for Computational Linguistics, 2017, 160–165. https://doi.org/10.18653/v1/W17-4421

Liu, J., Zhu, D., Bai, Z., He, Y., Liao, H., Que, H., Wang, Z., Zhang, C., & Zhang, e. a., Ge. (2025). *A comprehensive survey on long context language modeling*. http://arxiv.org/abs/2503.17407

Luo, L., Yang, Z., Yang, P., Zhang, Y., Wang, L., Lin, H., & Wang, J. (2018). An attention-based bilstm-crf approach to document-level chemical named entity recognition. *Bioinformatics*, *34*(8), 1381–1388. https://doi.org/10.1093/bioinformatics/btx761

Malik, V., Sanjay, R., Guha, S. K., Hazarika, A., Nigam, S., Bhattacharya, A., & Modi, A. (2022). *Semantic segmentation of legal documents via rhetorical roles*. https://www.proquest.com/docview/2607084336

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

Mintz, M., Bills, S., Snow, R., & Jurafsky, D. Distant supervision for relation extraction without labeled data. In: In *Joint conference of the 47th annual meeting of the acl and the 4th international joint conference on natural language processing of the afnlp*. Suntec, Singapore: Association for Computational Linguistics, 2009, 1003–1011.

Mochales Palau, R., & Moens, M.-F. Argumentation mining: The detection, classification and structure of arguments in text. In: In *Twelfth international conference on artificial intelligence and law (icail 2009)*. ACM, 2009, 98–109. https://doi.org/10.1145/1568234.1568246

Moens, M. F. (2001). Innovative techniques for legal text retrieval. *Artificial intelligence and law*, *9*(1), 29–57. https://doi.org/10.1023/A:1011297104922

Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, *30*(1), 3–26. https://doi.org/10.1075/li.30.1.03nad

Nentwich, C. (2005). *Managing the consistency of distributed documents*. http://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.416649

Noy, N. F., & McGuinness, D. L. (2001). Ontology development 101 : A guide to creating your first ontology. *Stanford Knowledge Systems Laboratory Technical Report*.

Omran, P. G., Taylor, K., M 'endez, S. J. '. R. '. i., & Haller, A. (2020). Towards shacl learning from knowledge graphs. [journal: ISWC (Demos/Industry)]. *ISWC (Demos/Industry)*, *2721*, 94–99.

Qu, R., Tu, R., & Bao, F. (2024). *Is semantic chunking worth the computational cost?* http://arxiv.org/abs/2410.13070

Rabbani, K., Lissandrini, M., & Hose, K. Shacl and shex in the wild: A community survey on validating shapes generation and adoption. In: In *Web conference 2022*. New York, NY, USA: ACM, 2022, 260–263. https://doi.org/10.1145/3487553.3524253

Rabbani, K., Lissandrini, M., & Hose, K. (2023). Extraction of validating shapes from very large knowledge graphs. *Proceedings of the VLDB Endowment*, *16*(5), 1023–1032. https://doi.org/10.14778/3579075.3579078

Ratner, N., Levine, Y., Belinkov, Y., Ram, O., Magar, I., Abend, O., Karpas, E., Shashua, A., Leyton-Brown, K., & Shoham, Y. Parallel context windows for large language models. In: In *The 61st annual meeting of the association for computational linguistics*. Association for Computational Linguistics, 2022, 6383–6402. https://doi.org/10.48550/arxiv.2212.10947

Rau, A. (2024). *Municipal laws in pennsylvania townships, authored by multiple people over time, develop inconsistencies and are incomplete* [A personal communication was conducted with Andrew D. Rau, Esq., who serves as the Township Solicitor for Easttown Township. The objective of this discussion was to gain an expert legal perspective on the practical need for my research into the automated verification of municipal codes. I presented my research on developing a computational tool for analyzing legislative texts to ensure their logical consistency

and completeness. Solicitor Rau, drawing upon over two decades of experience in Pennsylvania municipal law, confirmed the inherent limitations of the current multi-stakeholder review process. He affirmed that despite rigorous oversight from Township staff, the solicitor's office, and legislative codifiers, logical lacunae and contradictions can still persist within the legal code. Solicitor Rau's assessment provides a crucial external validation for the central problem statement of this dissertation. He confirmed that a formal, automated system capable of auditing legal frameworks for structural integrity would represent a significant and highly valuable contribution to the field of municipal governance and law. This conversation underscores the real-world demand for the technological solution my research aims to develop.].

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

Reiter, R. (1978). On closed world data bases. *Logic and Data Bases*, 55–76.

Rossi, M. (2016). Inconsistent legislation (Anonymous, Trans.). In, *Legisprudence library* (pp. 189–208). Springer International Publishing. https://doi.org/10.1007/978-3-319-33217-8_8

Rzepka, R., Muraji, S., & Obayashi, A. Expert evaluation of export control-related question answering capabilities of llms [ID: cdi ieee primary 10487735]. In: In *Ieee asia-pacific conference on computer science and data engineering (csde)*. ID: cdi ieee primary 10487735. IEEE, 2023, 1–6. ISBN: 9798350341072. https://doi.org/10.1109/CSDE59766.2023.10487735

Sanders, J. (2024). *Municipal laws in pennsylvania townships, authored by multiple people over time, develop inconsistencies and are incomplete* [A telephone interview was conducted with Jeanie Sanders, an administrator at eCode360, the digital repository for numerous Pennsylvania

municipal legal codes. The purpose of the communication was twofold: to ascertain the protocols for programmatic access to their legal corpus and to understand their current data validation methodologies. Ms. Sanders described their existing process for ensuring the integrity of the codes, which relies primarily on manual editorial review and version control. Following this, I presented my doctoral research, which focuses on the application of formal methods to analyze and verify legal frameworks. I outlined my methodology for modeling municipal ordinances as a formal system to programmatically audit the code for internal consistency, completeness, and logical contradictions. The discussion confirmed the potential utility of such computational tools in the field of legal informatics. Ms. Sanders acknowledged the inherent challenges of maintaining integrity across a large-scale, text-based legal corpus through manual processes alone. This communication was valuable in validating the practical need and real-world applicability for my research into automated legislative analysis and what can be described as 'computational jurisprudence'.].

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. (2024). *Tokenization is more than compression*. http://arxiv.org/abs/2402.18376

Schönberg, C., Weitl, F., & Freitag, B. (2011). Verifying the consistency of web-based technical documentations. *Journal of symbolic computation*, *46*(2), 183–206. https://doi.org/10.1016/j.jsc.2010.08.007

Shaham, U., Segal, E., Ivgi, M., Efrat, A., Yoran, O., Haviv, A., Gupta, A., Xiong, W., Geva, M., Berant, J., & Levy, O. Scrolls: Standardized comparison over long language sequences. In: In *Conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2022, 12007–12021. https://doi.org/10.48550/arXiv.2201.03533

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

Sporny, M., Longley, D., Kellogg, G., Lanthaler, M., Champin, P.-A., & Lindström, N. (2025). *Json-ld 1.1*. https://www.w3.org/TR/json-ld11/

Tauqeer, A., Kurteva, A., Chhetri, T. R., Ahmeti, A., & Fensel, A. (2022). Automated gdpr contract compliance verification using knowledge graphs. *Information (Basel)*, *13*(10), 447. https://doi.org/10.3390/info13100447

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). *Gemini: A family of highly capable multimodal models* (I tried uploading the PDF but it would not upload. It is in my Reference file as 2312.11805v4 (1).pdf.). http://arxiv.org/abs/2312.11805

Tian, L., Zhou, X., Wu, Y.-P., Zhou, W.-T., Zhang, J.-H., & Zhang, T.-S. (2022). Knowledge graph and knowledge reasoning: A systematic

review. *Journal of Electronic Science and Technology*, *20*(2), 100159. https://doaj.org/article/fc808085fb154c6c9b032ac617e9f233

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. (2024). *An introduction to transformers*. http://arxiv.org/abs/2304.10557

Umar, M. A., & Lano, K. (2024). Advances in automated support for requirements engineering: A systematic literature review. *Requirements engineering*, *29*(2), 177–207. https://doi.org/10.1007/s00766-023-00411-0

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*.

Verma, U. (2024). A journey from ai to gen-ai. *Spectrum of Emerging Sciences*, *4*(1), 74–78. https://doi.org/10.55878/SES2024-4-1-14

Wang & Guo. (2014). A short analysis of discourse coherence. *Journal of Language Teaching and Research*, *5*(2), 460. https://doi.org/10.4304/jltr.5.2.460-465

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

Wang, S., Sun, X., Li, X., Ouyang, R., Wu, F., Zhang, T., Li, J., & Wang, G. (2023). *Gpt-ner: Named entity recognition via large language models*. https://doi.org/10.48550/arxiv.2304.10428

Weitl, F., & Freitag, B. (2006). Checking content consistency of integrated web documents. *Journal of computer science and technology*, *21*(3), 418–429. https://doi.org/10.1007/s11390-006-0418-9

Wu, S., & He, Y. Enriching pre-trained language model with entity information for relation classification. In: In *28th acm international conference on information and knowledge management (cikm '19)*. Beijing, China: Association for Computing Machinery, 2019, 2361–2364. https://doi.org/10.1145/3357384.3358039

Xu, D., Chen, W., Peng, W., Zhang, C., Xu, T., Zhao, X., Wu, X., Zheng, Y., Wang, Y., & Chen, E. (2024). Large language models for generative information extraction: A survey. *Frontiers of Computer Science*, *18*(6), 186357. https://doi.org/10.1007/s11704-024-40555-y

Yang, J., Yoon, S., Kim, B., & Lee, H. Fizz: Factual inconsistency detection by zoom-in summary and zoom-out document (Y. Al-Onaizan, M. Bansal, & Y.-N. Chen, Eds.). In: *2024 conference on empirical methods in natural language processing* (Y. Al-Onaizan, M. Bansal, & Y.-N. Chen, Eds.). Ed. by Al-Onaizan, Y., Bansal, M., & Chen, Y.-N. Miami, Florida, USA: Association for Computational Linguistics, 2024, 30–45. https://doi.org/10.18653/v1/2024.emnlp-main.3

Ye, J., Xu, N., Wang, Y., Zhou, J., Zhang, Q., Gui, T., & Huang, X. (2024). *Llm-da: Data augmentation via large language models for few-shot named entity recognition*. http://arxiv.org/abs/2402.14568

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* (No. 2). https://doi.org/10.48550/arXiv.2303.18223

Zhong, L., Wu, J., Li, Q., Peng, H., & Wu, X. (2024). A comprehensive survey on automatic knowledge graph construction. *ACM computing surveys*, *56*(4), 1–62. https://doi.org/10.1145/3618295

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

Chapter 7: Source Code

This appendix contains the complete source code for the project. The code is organized by file and includes comments to explain the functionality of key sections. This allows for a thorough review of the implementation and facilitates future development or replication of the work. For larger projects, a link to a code repository (e.g., GitHub) may be more appropriate and can be included here.

Chapter 8: Class Structure

This appendix provides a detailed description of the class structure implemented in the program. For each class, it includes the attributes, methods, and their respective purposes. A UML (Unified Modeling Language) diagram is also provided to visually represent the relationships and inheritance between the classes, offering a clear overview of the program's architecture.

Chapter 9: JSON Structure

This appendix describes the JSON (JavaScript Object Notation) structure utilized to constrain the large language models (LLMs). It outlines the schema, including the key-value pairs, data types, and nested structures. Examples of the JSON objects are provided to illustrate how specific constraints are defined and passed to the LLMs to guide their output.

Chapter 10: Cypher Queries

This appendix contains the Cypher queries used for data creation, manipulation, and retrieval from the Neo4j graph database. Each query is presented with a brief description of its function and the context in which it was executed. This provides a transparent and replicable account of all database interactions central to this project.

Chapter 11: Prompts

This appendix contains a comprehensive list of all prompts used to interact with the LLMs. Each prompt is presented verbatim, accompanied by a description of its purpose, the context in which it was used, and the expected format of the response. This section is intended to ensure the replicability of the research by detailing the exact inputs given to the models.

Chapter 12: Evaluation Metrics

This appendix details the metrics used to evaluate the performance of the system. It provides the mathematical formulas for each metric, explains why each was chosen, and describes the methodology used for its calculation. This ensures that the evaluation process is transparent and reproducible.

Chapter 13: Problems Encountered

This appendix serves as a log of the significant problems encountered during the project's development and execution. For each issue, it details the nature of the problem, the steps taken to diagnose it, the attempted solutions, and the final resolution. This section aims to provide a transparent account of the research process and to assist others who might face similar challenges.

Entities needed IDs but the LLM has no realtime access so no GUID or time based IDs. They had to be nique accros chunks. So, I cam up with an approaach passing in the date, time, chunk number and document name. It is not space efficent.

Constantly having to reset colab. After running for several minutes, in somecases colab neeeds to be restarted. this was particularly problematic dring development where each change wold reqire a restart.

neo4j shuts down

I had to pt in several rety sections because the LLM wold sometimes produce bad results.