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:

Professor M. Elbasheer, Professorial Lecturer in Engineering and Applied Science, Praxis Director

XYZ, Professor of Online Engineering Programs, Chair

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

Dedication

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.

Acknowledgments

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.

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

API Bidirectional Long Short-Term Memory network

BPE Byte Pair Encoding

CPU Central Processing Unit

CRF Conditional Random Field

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 within the document itself. 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 the consistency and completeness of a document.

1.1.1 Background

The increasing complexity and scale of textual documents in various domains present significant challenges in ensuring consistency and completeness. Legal codes, technical documentation, and regulatory frameworks are often drafted collaboratively over extended periods, leading to inconsistencies, redundancies, and gaps in information. 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, though 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 in various domains. Previous research has focused on methods to maintain internal coherence within documents (Laban et al., July 7, 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 previous models. However, 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. However, LLMs are constrained by a fixed context window, limiting 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 at once. Instead, LLMs can be employed to extract key entities and concepts across different sections, enabling a more focused and structured approach to consistency checking. By identifying entities of interest and analyzing 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 in which nodes represent entities, and edges define 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. One useful way to conceptualize knowledge graphs is through the framework of frames, as described by Minsky (Minsky, 1974). Unlike 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. While legal professionals and municipal officials work diligently to identify and resolve these issues, the complexity of these documents—often spanning thousands of pages—makes 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. However, inconsistencies can arise when later sections introduce or reference zoning areas that were never formally defined. Similar discrepancies can emerge across other regulatory provisions, requiring 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 sections of documents, there is a significant gap in addressing the challenges of comprehensive, large-scale document analysis. The need for automated consistency and completeness checks is critical in various industries. Currently, 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.

Within the scope of this research, local regulations of townships in

Pennsylvania go through a time consuming and complicated process to be published. After the governing body enacts a law, it is sent to an organization to compile it into existing laws of the township. This is a manual and intensive process to determine if any of the existing laws are affected by the new law. Even with this, there are many cases of new laws that make a set of existing laws incomplete or inconsistent.

1.2 Problem Statement

Municipal laws in Pennsylvania Townships, authored by multiple people over time, develop inconsistencies and are incomplete (Curely, 2024), (Rau, 2024), and (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 over time. Ordinances and regulations are often drafted by different individuals, including elected officials, legal counsel, and administrative staff, each contributing to the evolving legal framework. However, 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 inconsistencies. 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 the 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 may lead to disputes that reduce collections. When enforcement mechanisms are weak due to 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, limiting 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 only identifying issues when disputes or financial shortfalls arise. 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, improving clarity, enforcement, and financial sustainability. However, implementing such an approach 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 will allow municipal lawyers to create consistent and complete law documents which prevent costly disputes and reduce 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, which is inherently susceptible to errors, inconsistencies, and omissions—particularly 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 minimizes ambiguity, strengthens legal clarity, and enhances 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 allows municipal lawyers to assess the interconnectivity of legal clauses and verify their consistency against established legal principles and precedents. 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 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, permitting, and compliance, often resulting in costly litigation or lost revenue due to 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, streamline enforcement mechanisms, and enhance overall regulatory efficiency. This, in turn, 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 in 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 a legally sound, financially sustainable framework. Future research can extend this approach beyond municipal laws to other domains of legal and regulatory governance, demonstrating the broader impact of AI-driven knowledge representation in ensuring legal accuracy, reducing administrative burdens, and enhancing public trust in local government operations.

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 in a knowledge graph. This tool will leverage advanced techniques to analyze document content, identify potential entities, and provide access to the knowledge graph.

The created knowledge graph will be analyzed to determine whether it is appropriate to check the document for inconsistencies and incompleteness. This will include introducing issues in the source documents and then highlighting how easy they are to observe in the knowledge graph.

1.5 Research Questions

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

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

Research will be conducted to 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, which employs Pennsylvania township laws as a case study to develop and evaluate an automated tool for the analysis of legal documents. These publicly accessible laws, having undergone extensive manual reviews for consistency and completeness, provide a rigorous benchmark for assessing the proposed methodology. The primary focus of this research is the construction of Knowledge Graphs that faithfully represent the structure and content of the documents, thereby laying the groundwork for future efforts in verifying legal consistency and completeness. Notwithstanding, this study acknowledges several inherent limitations, including computational constraints, challenges associated with specific document formats and linguistic nuances, and the primary emphasis on textual analysis. These limitations underscore the necessity for continued research to enhance and broaden the applicability of the proposed approach.

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, which are publicly available in both PDF and Word formats, were selected due to their complexity, extensive length, and the fact that they have been authored by multiple contributors over time. Additionally, they have undergone rigorous manual reviews for consistency and completeness, making them an ideal benchmark for evaluating the effectiveness of the proposed approach. While the primary application is in the legal domain, the methodology is designed to be adaptable for broader use across various document types.

The core development in 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 the study will assess the suitability of the generated Knowledge Graphs for such tasks, the actual implementation of automated consistency and completeness checks will be left for future research. This approach ensures a focused and systematic exploration of Knowledge Graph generation while laying the groundwork 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 language intricacies, particularly in ensuring accurate interpretation and structuring of legal text. Additionally, while this research focuses on leveraging LLMs such as Gemini and ChatGPT, it does not develop specialized models tailored for knowledge graph construction, consolidation, or query answering—an approach that could reduce computational costs and energy consumption.

This research does not perform direct testing for consistency and completeness. Instead, it utilizes Pennsylvania township laws, which are publicly available and have already undergone such validation. Future studies should explore the applicability of this approach to a broader range of legal and non-legal documents.

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

Finally, this study is limited to textual analysis. Future research could expand upon this work by incorporating additional elements such as tables, formulas, images, and diagrams to improve 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 into a combined knowledge graph by LLMs, the utility of knowledge graphs in representing the original document to ensure consistency and completeness, the process of creating and maintaining local laws in Pennsylvania, and background information on checking documents for consistency and completeness. Chapter 3 delves into the statistical and machine learning methodologies employed in this research, detailing the processes of data pre-processing, model selection, training, and evaluation. Chapter 4 presents and analyzes the results of the data analysis, addressing each research question and hypothesis while evaluating the performance of the proposed methodology and tool. Finally, Chapter 5 concludes the investigation with a discussion of the key findings, contributions to the field, recommendations for practical applications, and 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 and documented in the seminal paper *Attention Is All You Need* (Vaswani et al., 2017). This paper introduced the Transformer architecture, leveraging self-attention mechanisms, which became the foundation for modern Large Language Models (LLMs). These models have demonstrated remarkable capabilities across a wide range of tasks, including text generation, summarization, translation, and question answering, often producing outputs nearly indistinguishable from human writing (Badshah & Sajjad, 2024) and (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 the model can process simultaneously when generating a response or performing an analysis. Consequently, if critical information or dependencies exist within a document but fall outside this fixed window, separated by a larger span of intervening text, the LLM may fail to capture the relationship or address the query accurately (Kaplan et al., 2020). This limitation poses a significant challenge when dealing with large or complex documents where understanding relies on synthesizing information across distant sections.

One promising approach to mitigate this limitation involves transforming large, unstructured documents into structured representations using Knowledge Graphs (KGs). By extracting key entities, relationships, and attributes from the text and mapping them into a graph structure, it becomes possible to represent the document's core semantic content in a format amenable to computational analysis (Hogan et al., 2021). This allows for querying and reasoning over the entire document's scope, independent of the LLM's context window constraints, potentially enabling a more focused and comprehensive analysis for tasks such as ensuring information integrity.

This chapter reviews the pertinent literature underpinning this approach. It begins by examining the development and characteristics of Large Language Models, focusing on their capabilities and limitations, particularly the context window constraint. Subsequently, it delves into the principles, construction, and application of 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. Finally, the chapter defines the critical concepts of consistency, completeness, and coherence, particularly relevant for evaluating the integrity of document corpora like legal codes, and surveys related work before concluding with a summary motivating the proposed research direction.

2.2 Large Language Models

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

Long Short Term Memories (LSTMs)) or convolutional architectures (Turner, 2024) and (Zhao et al., 2023), addressing critical bottlenecks present in earlier sequence models. This innovation paved the way for the development of increasingly large and powerful language models, such as Google's BERT, which introduced bidirectional pre-training (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 in model scale, training data, and architectural refinements, the public release of OpenAI's *ChatGPT* (based on the GPT-3.5 architecture) on November 30, 2022, marked a pivotal moment. This event dramatically increased public awareness and accelerated the development and deployment of advanced conversational AI systems across various sectors. It catalyzed the release and further development of competing models from major research labs, including Google's *Gemini* family of multimodal models (Team et al., 2024), Anthropic's safety-focused *Claude* series (Caruccio et al., 2024), and Meta's open-source *Llama* family, which have spurred significant community innovation (Grattafiori et al., 2024). The proliferation of models is evident on platforms like Hugging Face, a central repository for AI models and datasets, which reportedly surpassed one million hosted models by late 2024, reflecting the rapid pace of development in the field (Edwards, 2025).

Functionally, LLMs process input text (the "prompt") by first converting it into numerical representations called tokens, often using techniques like Byte Pair Encoding (BPE) or WordPiece (Schmidt et al., 2024). Using the complex patterns and linguistic knowledge learned during extensive pre-training on vast text corpora (often terabytes of data), the model then predicts subsequent tokens autoregressively to generate a coherent and contextually relevant output. Prompts can be engineered to elicit specific behaviors or perform complex tasks, potentially including substantial amounts of text for analysis or context (in-context learning). For instance, an LLM might be prompted with a company's annual report and asked specific questions about its contents, or asked to summarize key findings. Many current LLMs can perform reasonably well on such tasks, provided the relevant information falls within their processing limits (Rzepka et al., 2023).

However, a fundamental limitation remains the context window size. This size, representing the maximum number of tokens the model can attend to simultaneously, while increasing with newer model generations (ranging from a few thousand in early models to potentially over a million tokens in recent research prototypes (Ratner et al., Dec 21, 2022) and (Kaplan et al., 2020)), is always finite (Liu et al., 2025). If a document's length exceeds this limit, the LLM cannot process it in its entirety in a single pass. Standard techniques involve processing the document in overlapping or non-overlapping chunks (T. Chen et al., December 11, 2023), but this can sever long-distance contextual links crucial for deep understanding. For example, determining if a policy statement defined on page one of a lengthy legal code is adequately supported or subtly contradicted by detailed regulations presented hundreds of pages later might be impossible if the intervening text exceeds the context window. The LLM would process the sections independently, unable to synthesize the relationship between them effectively.

Furthermore, the computational cost of processing information within the context window remains a significant factor. The self-attention mechanism, core to the Transformer, typically scales quadratically $(O(n^2))$ with the sequence length (n) in terms of both computation and memory requirements (Vaswani et al., 2017). While various "efficient Transformer" variants aim to reduce this to near-linear complexity (Tay et al., 2023), processing very long sequences up to the maximum context window still demands substantial memory (RAM/VRAM), processing power (CPU/GPU), and energy resources. This quadratic (or near-quadratic) scaling makes analyzing very large documents prohibitively expensive or slow for many practical applications, further motivating alternative approaches, such as KG-based structuring, for achieving comprehensive and efficient analysis (Tay et al., 2023).

2.3 Knowledge Graphs

Knowledge Graphs (KGs) provide a structured paradigm for representing information and knowledge, evolving from concepts in semantic networks, frame systems, and earlier AI research in symbolic knowledge representation (Hogan et al., 2021). Formally, a KG typically represents knowledge as a directed labeled graph, comprising a collection of interconnected entities (nodes or vertices) and the explicitly typed relationships (edges or links) between them. Both nodes and edges can possess attributes or properties (key-value pairs) that store additional metadata, context, or provenance information (Ehrlinger & W "o ss, 2016) and (Cong, 2024/10/16).

The core components of a KG are:

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

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

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

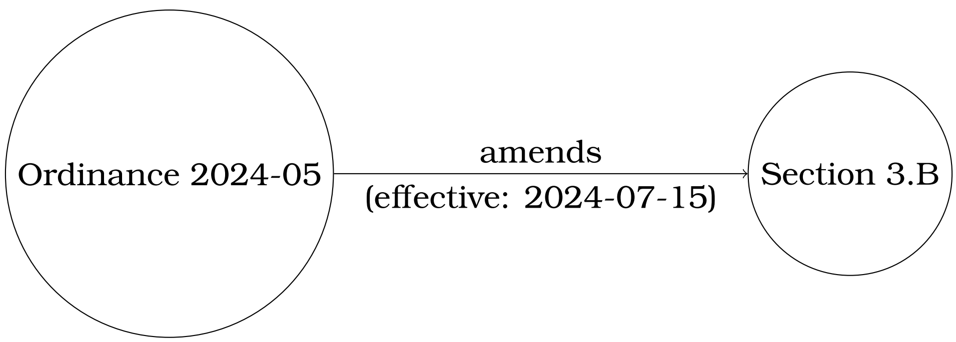

Figure 2.1: *Knowledge graph fragment of a legal amendment.*

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

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

- **RDF (Resource Description Framework):** A W3C standard data model based on triples (subject-predicate-object statements). RDF graphs are often serialized in formats like Turtle, RDF/XML, or N-Triples and queried using the SPARQL protocol and query language ("RDF 1.2 Primer", 2025) and (N. Kumar & Kumar, July 1, 2013). RDF is foundational to the Semantic Web and facilitates data interoperability. Ontology languages like RDFS and OWL (Hitzler, Krotzsch, & Rudolph, 2009b) and (Hartig et al., 2025) can be used to define schemas and enable richer reasoning over RDF KGs.

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

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

- **Other Formats:** KGs can also be represented or serialized using formats like JSON (e.g., JSON-LD) (Sporny et al., 2025) or XML, though these may lack the optimized querying and reasoning capabilities offered by dedicated graph databases or RDF triple stores.

KGs are employed in diverse applications, including powering Google's

Knowledge Graph for semantic search, enhancing recommendation systems (e.g., Amazon, Netflix), integrating heterogeneous data sources in enterprises, bioinformatics, financial analysis, and enabling more sophisticated question answering systems (Hogan et al., 2021), (Ji et al., 2022), (Fensel et al., 2020), and (Tian et al., 2022). Their ability to explicitly model complex relationships and provide a structured representation of knowledge makes them potentially valuable for analyzing the internal structure, interconnections, and overall integrity of large document collections, such as legal codes.

2.4 Information Extraction for KG Construction

To leverage the benefits of KGs for document analysis, the unstructured or semi-structured information within the source documents must first be transformed into the structured format of the graph. This process, often termed KG construction or population, relies heavily on Information Extraction (IE) techniques (Zhong et al., 2024) and (Kolluru et al., July 1, 2020). This section covers two fundamental IE tasks critical for extracting the primary components of a KG: identifying the nodes (entities) using Named Entity Recognition (NER) and identifying the edges (relationships) using Relation Extraction (RE). LLMs have shown significant promise in performing both tasks, often with minimal task-specific training data (Benjira et al., 2025).

2.4.1 Named Entity Recognition

Named Entity Recognition (NER) is a fundamental task in information extraction that focuses on identifying and classifying mentions of named entities within unstructured text into pre-defined categories (Al-Moslmi et al., 2020). These categories typically include standard types like persons (PER), organizations (ORG), locations (LOC), dates, and monetary values, but crucially, can be extended to domain-specific entities relevant to the application context.

In the context of building knowledge graphs from text, NER plays a crucial role. It serves as the primary mechanism for identifying the potential **nodes** (entities) that will populate the graph. By extracting key actors, locations, concepts, defined terms, document sections, or other items of interest from the source documents, NER provides the raw material for the structured representation. Disambiguating these mentions and linking them to unique identifiers in the KG (Entity Linking) is often a necessary subsequent step (Chaurasiya et al., 2022).

Various methods have been developed for NER over the years:

- **Rule-based Systems:** Early approaches relied on hand-crafted grammatical rules, dictionaries (gazetteers), and regular expressions. These systems can achieve high precision when rules are well-defined but are often brittle, domain-specific, and labor-intensive to create and maintain (Nadeau & Sekine, 2007) and (Grishman & Sundheim, 1996).

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

- **Deep Learning Approaches:** More recently, deep neural networks have achieved state-of-the-art performance. Architectures like Bidirectional

Long Short-Term Memory networks (BiLSTMs), often combined with a CRF output layer (BiLSTM-CRF), effectively capture sequential context (Lin et al., September 1, 2017), (Lample et al., April 7, 2016), and (Luo et al., 2018). Increasingly, Transformer-based models like BERT (Gardazi et al., 2025) and its variants, fine-tuned on NER tasks, have become standard, leveraging powerful pre-trained representations (Al-Moslmi et al., 2020) and (Carbonell et al., Jan 10, 2021). LLMs can also perform NER directly via prompting or few-shot learning (S. Wang et al., 2023) and (Ye et al., 2024).

Applying NER to the legal domain requires careful consideration of domain-specific entities that are critical for understanding legal texts. Beyond standard types, entities might include: specific legal statutes or section references (e.g., '15 Pa.C.S.A. § 1502'), defined legal terms (e.g., 'applicant', 'nonconforming use', 'force majeure'), legal roles (e.g., 'Township Supervisor', 'Zoning Officer', 'plaintiff'), court names, specific dates or deadlines, monetary penalties, and explicit references to other documents or sections (Au et al., December 19, 2022) and (Kalamkar et al., Nov 7, 2022). Due to the specialized vocabulary, complex sentence structures, and importance of precision, training or fine-tuning NER models on legally annotated corpora (like those from legal shared tasks or specific research projects) is often necessary to achieve high accuracy (Chalkidis et al., 2022). The output of a robust legal NER system provides the essential entity building blocks for constructing a meaningful knowledge graph from legal texts.

2.4.2 Relation Extraction

While NER identifies the entities (nodes), Relation Extraction (RE) is the task of identifying semantic relationships that hold between pairs (or sometimes n-tuples) of these entities in text (Ji et al., 2022) and (Carbonell et al., Jan 10, 2021). These extracted relations typically correspond to the edges in the knowledge graph, connecting the nodes identified by NER and thus building the graph's structure. For instance, given the sentence "Acme Corp, headquartered in West Chester, acquired Beta Inc.", RE aims to identify relations like ‘headquarteredIn(Acme Corp, West Chester)‘ and ‘acquired(Acme Corp, Beta Inc)‘.

Identifying the type of relation is crucial. While early work focused on a small set of predefined relation types (Closed RE), more recent work also tackles Open Information Extraction (OpenIE), which aims to extract relations expressed using arbitrary textual phrases (Etzioni et al., 2008). For KG construction, typically a predefined schema or ontology dictates the target relation types (Closed RE), such as cites, amends, defines, employs, locatedIn, etc. Some fundamental ontological relationships often considered include is-a (subclass/instance) and part-of (meronymy) relations, alongside more domain-specific associative relationships (Noy & McGuinness, 2001).

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

- **Rule-based / Pattern-based Systems:** Utilize linguistic patterns (e.g., dependency paths between entities) or hand-crafted rules over text or syntactic structures (like parse trees) to identify relations (Hearst, 1992). Bootstrapping methods like DIPRE and Snowball automatically learn extraction patterns from seed examples (Brin, 1998) and (Agichtein & Gravano, June 1, 2000). These can be effective for specific relations but suffer from similar limitations as rule-based NER.

- **Supervised Statistical Models:** One approach involves training classifiers, such as Support Vector Machines (SVMs) or Maximum Entropy (MaxEnt) models, using features derived from the text connecting candidate entities. These features often include lexical information and syntactic patterns from dependency paths, and this method relies on manually annotated data for supervised learning (Kambhatla, 2004). Alternatively, distant supervision attempts to automate the generation of training data by aligning known relations from knowledge graphs (like Freebase or DBpedia) with sentences mentioning the corresponding entities. While this avoids manual annotation, the automated alignment process can introduce significant noise into the training set (Mintz et al., 2009).

- **Deep Learning Approaches:** Neural models, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs/LSTMs), and Graph Neural Networks (GNNs), operating on dependency trees, have been applied successfully, often outperforming feature-engineered systems (S. Kumar, 2017). Transformer-based models, pre-trained on large corpora and fine-tuned for RE, currently represent the state-of-the-art for many RE benchmarks (Wu & He, 2019). LLMs, particularly through prompting techniques (including few-shot prompting), offer a powerful alternative, capable of extracting relations with minimal task-specific fine-tuning by leveraging their vast world knowledge and language understanding capabilities (Chia et al., 2022). Prompting strategies often involve formulating the task as question answering or fill-in-the-blank over the input text containing the entity pair.

Challenges in RE include handling ambiguity (the same text might imply different relations), extracting relations expressed across sentence boundaries, dealing with complex or n-ary relations (involving more than two entities), adapting models to new domains or relation types with limited data, and robust evaluation. In the legal domain, extracting relations like amendments between code sections, definitions of terms, obligations imposed by regulations, or citations between cases and statutes is critical for building a KG that accurately reflects the legal framework (Tauqeer et al., 2022) and (Dhani et al., 2021). The structured output of NER and RE, when combined and potentially refined (e.g., through entity linking and relation consolidation), forms the basis for the constructed knowledge graph.

2.5 Consistency, Completeness, and Coherence

When analyzing formal document corpora, particularly large and evolving ones like legal codes, software requirements, or technical standards, evaluating their quality often involves assessing their internal integrity. Three key aspects of this integrity are consistency, completeness, and coherence (Umar & Lano, 2024). These concepts, while sometimes overlapping, address distinct facets crucial for ensuring the documents are understandable, unambiguous, reliable, and effective in their intended function.

- **Consistency:** Refers primarily to the absence of logical contradictions within the document set (Zowghi & Gervasi, 2003), (Heitmeyer et al., 1996), (Nentwich, 2005), (Egyed, May 28, 2006), (Tröls et al., 2022), (Yang et al., Apr 17, 2024), and (Guo et al., 2023). A consistent set of statements should not allow for the derivation of both a proposition and its negation. In a legal code, this means it should not contain provisions that assert mutually exclusive facts (e.g., defining the same term in incompatible ways) or prescribe conflicting obligations or permissions under identical conditions (e.g., one section mandates an action that another prohibits for the same actor and circumstance). Detecting inconsistencies is vital for legal certainty, predictability, and avoiding disputes (Donelson, 2019), (Duck-Mayr, 2022), and (Rossi, 2016). Formal logic and automated reasoning techniques are often employed to check consistency in formal specifications (Heitmeyer et al., 1996) and (Brucker & Wolff, 2019).

- **Completeness:** Pertains to whether the document set contains all the necessary information required relative to its intended scope or purpose (Zowghi & Gervasi, 2003). Defining completeness is inherently challenging as it depends on a clear specification of what *should* be included. In a legal context, this could mean ensuring that all terms used are adequately defined, procedures referenced are fully specified, criteria for decisions are enumerated exhaustively, exceptions are handled, and potential scenarios relevant to the scope are addressed. Gaps, omissions, or "TBD" markers can lead to ambiguity, loopholes, and disputes. Assessing completeness often requires significant domain knowledge and may involve checking against predefined templates, checklists, or requirements specifications (Zowghi & Gervasi, 2003) and (Umar & Lano, 2024). The "Closed World Assumption" versus "Open World Assumption" impacts how completeness might be formally interpreted in KGs (Reiter, 1978) and (Hitzler, Kr otzsch, & Rudolph, 2009a).

- **Coherence:** Relates to the overall understandability, organization, and logical flow of the information presented (Wang & Guo, 2014) and (Shen et al., 2021). A coherent document is well-structured, uses terminology consistently across sections, ensures cross-references are accurate and lead to relevant information, avoids unnecessary jargon or ambiguity, and maintains a clear narrative or argumentative structure. While related to consistency (an incoherent document might contain implicit contradictions), coherence focuses more on the clarity, usability, and comprehensibility for a human reader (Wang & Guo, 2014). Aspects include lexical cohesion, referential clarity, and discourse structure (Wang & Guo, 2014).

Ensuring these three qualities simultaneously in large, evolving legal codes through traditional manual review processes is exceptionally difficult. The sheer volume of text, the intricate web of interdependencies (definitions, cross-references, amendments), the potential for ambiguity in natural language, and the often distributed and lengthy nature of authorship and revision processes over time make manual detection of subtle flaws challenging and error-prone (Beth, 2018). This is where computational approaches leveraging structured representations like KGs, potentially populated and analyzed with the aid of LLMs, offer significant potential advantages.

A Knowledge Graph, by explicitly modeling entities (like defined terms, sections, obligations, actors, conditions) and their relationships (like 'defines', 'cites', 'amends', 'requires', 'prohibits', 'conflicts with'), provides a structured substrate amenable to automated analysis. Graph-based queries (e.g., using SPARQL or Cypher) or graph algorithms can be designed to automatically detect certain classes of potential inconsistencies, such as finding terms used before they are defined, identifying conflicting property values assigned to the same entity under specific conditions, detecting circular definition chains, or finding contradictory requirements linked to the same scenario (Tauqeer et al., 2022), (Brucker & Wolff, 2019), (Schönberg et al., 2011), and (Weitl & Freitag, 2006). While achieving perfect completeness verification is often intractable or ill-defined for natural language documents, KGs can help identify potential gaps by analyzing the graph's structure for missing nodes (e.g., undefined terms that are used), expected relationships that are absent (e.g., a procedure is mentioned but not detailed), or orphaned sections (Rabbani et al., 2023), (Rabbani et al., Apr 25, 2022), (Omran et al., 2020), (Knublauch & Kontokostas, 2017), and (Umar & Lano, 2024). Coherence might be partially assessed by analyzing the density and structure of cross-references, consistency in terminology usage (via entity linking), or detecting potentially ambiguous references.

LLMs can potentially play a role throughout this pipeline: aiding in the initial interpretation of nuanced text to populate the KG accurately (NER/RE), helping to formulate complex graph queries based on natural language questions about integrity, or summarizing the findings from the graph analysis for human review (Benjira et al., 2025). However, the KG itself provides the persistent, globally coherent, and computationally tractable structure necessary for systematic integrity checks that can overcome the context window limitations and potential lack of deterministic reasoning inherent in LLMs alone. Research exploring the use of KGs and related AI techniques for automated consistency and completeness checking in domains like software requirements engineering (Umar & Lano, 2024) and (Heitmeyer et al., 1996), logical formalisms, and more recently, legal texts provides a foundation for this approach (Tauqeer et al., 2022), (Dhani et al., 2021), and (Aumiller et al., Jun 21, 2021). This praxis project aims to build upon such work, investigating the practical application of LLM-driven KG construction for checking the consistency and completeness of municipal legal codes.

2.6 Challenges in Analyzing Large Documents

Research efforts in automated document processing and understanding are extensive, covering tasks like summarization (Gambhir & Gupta, 2017), information extraction (as discussed previously) (Zhong et al., 2024), document classification (K. Wang et al., 2024), question answering (D. Chen et al., 2017), and validating the faithfulness or factuality of generated content (like summaries) against source documents (Guo et al., 2023) and (Yang et al., Apr 17, 2024). Historically, much foundational research and benchmark development focused on relatively small documents (e.g., news articles, single paragraphs, short scientific abstracts) for several practical reasons. Smaller documents are computationally less demanding to process, and crucially, the human evaluation and annotation required to establish ground truth and verify system performance are significantly more feasible and reliable at smaller scales.

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

- **Computational Resources:** Simply processing large volumes of text demands substantial memory (RAM and VRAM for deep learning models), storage, and processing time. The computational complexity often scales non-linearly (e.g., quadratically for standard Transformers) with document length, making naive processing infeasible (Vaswani et al., 2017).

- **Long-Range Dependencies:** Understanding often requires capturing semantic connections, references (e.g., pronoun resolution, term definitions), or causal dependencies between sections that are far apart in the document. Models with limited context windows struggle to capture these long-distance relationships accurately, as discussed regarding LLMs (Liu et al., 2025) and (Zhao et al., 2023).

- **Context Fragmentation:** Common techniques for handling large documents with fixed-input models involve splitting them into smaller chunks (e.g., fixed size, sentence-based, paragraph-based, or even semantically coherent chunks) (T. Chen et al., December 11, 2023) and (Qu et al., 2024). While necessary, this risks losing critical context that spans across chunk boundaries, potentially leading to fragmented understanding or incorrect inferences when information needs to be synthesized globally. Hierarchical processing methods attempt to mitigate this but add complexity (Liu et al., 2025).

- **Evaluation Complexity:** Assessing the quality of processing (e.g., the accuracy of a summary of a 500-page report, the correctness of an answer requiring synthesis across chapters, or the completeness of consistency analysis over an entire legal code) is inherently difficult and resource-intensive for human evaluators. Establishing reliable ground truth for evaluation benchmarks remains a major challenge for large-document tasks (Shaham et al., Jan 10, 2022).

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

2.7 Challenges in Analyzing Legal Documents

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

- **Complexity and Precision:** Legal language is notoriously dense, often employing specialized terminology (jargon with precise, sometimes non-intuitive, meanings), complex and nested sentence structures (long sentences with multiple subordinate clauses), and numerous explicit and implicit cross-references. Unlike much general text, ambiguity must be minimized, demanding extremely high precision in interpretation and analysis, as misinterpretations can have significant real-world consequences (Ashley, 2017) and (Malik et al., 2022).

- **Volume and Interconnectedness:** Legal corpora can be vast (e.g., entire state statutes, federal regulations like the CFR, large collections of case law, or extensive municipal codes). Furthermore, documents within these corpora are rarely standalone; they are highly interconnected through explicit citations, amendments that modify prior text, definitions that apply across sections or entire codes, and implicit dependencies based on legal principles or hierarchy (Beth, 2018). Understanding one part often requires understanding its relationship to many others.

- **Semi-structured Format:** While often exhibiting some structure (e.g., organized into titles, chapters, articles, sections, clauses, lists), legal texts contain significant amounts of unstructured natural language prose within these structures. This mix requires sophisticated NLP techniques capable of handling both the explicit structure and the dense prose content.

- **Critical Need for Integrity:** Perhaps most importantly, the consistency, completeness, and coherence of legal documents are paramount for their function in society. These qualities underpin the rule of law, ensuring predictability, fairness in application, and enforceability. Flaws such as contradictions, ambiguities resulting from omissions, or confusing structure can lead to uncertainty, disputes, costly litigation, and erosion of public trust (Donelson, 2019), (Duck-Mayr, 2022), and (Rossi, 2016).

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

The legislative drafting and codification process itself, while designed to ensure quality through multiple layers of review (staff, legal counsel/solicitor, planning commissions, public hearings, compiler checks), highlights the potential for introducing errors. When a new ordinance is proposed—whether initiated by elected officials, staff, or residents—it typically undergoes review by township staff and legal counsel who then drafts the formal language. A professional compiler may later be engaged to integrate the new law into the existing code and perform some validation checks. Despite this multi-stage human review process involving various stakeholders with legal or domain expertise, inconsistencies (e.g., contradictions with existing ordinances, conflicts with state preemptions), incompleteness (e.g., missing definitions for newly introduced terms, undefined procedures), and incoherence (e.g., unclear scope, confusing structure, inaccurate cross-references) can still arise and persist, especially as the code grows in size and complexity over time (Rossi, 2016). The resource-intensive, time-consuming, and inherently fallible nature of purely manual review motivates the exploration of computational methods, like the one proposed herein, to assist legal professionals and municipal staff in maintaining the integrity of these foundational legal documents.

2.8 Related Work

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

LLMs for Information Extraction and KG Construction: The advent of powerful LLMs has revolutionized information extraction. Numerous studies demonstrate the ability of LLMs like GPT-3/4, Claude, and Llama, often via prompting (zero-shot or few-shot), to perform NER and RE with performance rivaling or exceeding traditional fine-tuned models, especially in low-data or specialized domains (Xu et al., 2024). Researchers have explored various prompting strategies, output parsing techniques, and methods for mitigating LLM limitations like hallucinations or inconsistencies in extraction (S. Wang et al., 2023). Several works have specifically focused on constructing KGs from text using LLMs as the primary extraction engine, developing pipelines that integrate entity identification, relation extraction, entity linking, and schema mapping, sometimes incorporating human-in-the-loop refinement (Benjira et al., 2025), and (Lairgi et al., 2024). Challenges remain in scalability, controlling the output structure effectively, ensuring factual accuracy, and handling complex, long-form documents during extraction (Zhong et al., 2024).

KGs for Document Analysis and Integrity Checking: Beyond construction, KGs serve as a substrate for advanced document analysis. They have been used to enhance semantic search, enabling queries based on relationships rather than just keywords (Hogan et al., 2021). KGs facilitate complex question answering by allowing reasoning over extracted facts (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 have been used to model requirements and detect conflicts or missing elements (Umar & Lano, 2024). Formal methods often leverage graph-based representations for model checking (Brucker & Wolff, 2019). In the Semantic Web community, technologies like SHACL (Shapes Constraint

Language) provide a standard way to validate RDF KGs against predefined constraints or schemas, effectively checking aspects of consistency and completeness relative to the schema (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). Early work focused on rule-based systems for legal reasoning and expert systems. More recent research applies modern NLP to tasks like legal information retrieval (Moens, 2001), case outcome prediction (Aletras et al., 2016), document summarization (Bhattacharya et al., 2019), contract review (clause identification, risk analysis) (Chalkidis et al., 2022), argument mining (Mochales Palau & Moens, 2009), and e-discovery. Information extraction (NER and RE) from legal texts has received significant attention, focusing on extracting citations, legal entities, obligations, definitions, and relationships relevant for legal analysis (Kalamkar et al., Nov 7, 2022) and (Tauqeer et al., 2022). Some prior work has explored automated consistency checking in legal documents, often using rule-based approaches, deontic logic, or domain-specific heuristics, but typically focused on specific types of conflicts rather than a comprehensive KG-based approach applied to municipal codes (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, and AI has been applied to legal texts, the specific contribution here lies in the **integration and practical application of modern LLMs to construct KGs *specifically* from municipal legal codes (ordinances) for the explicit purpose of assisting in consistency and completeness analysis**. It addresses the limitations of LLMs (context window) by leveraging the KG structure for global analysis and reasoning. Unlike some prior legal AI work focusing on case law or contracts, this project targets the foundational legislative texts at the local government level. Compared to general KG construction methods, it focuses on the specific entities, relations, and integrity rules pertinent to municipal ordinances. The "praxis" aspect emphasizes the development and evaluation of a practical methodology and potential tool tailored to assist municipal staff and legal professionals in the challenging task of maintaining the quality of their codified laws, leveraging the latest advancements in LLMs and KG technologies. The evaluation will focus on the effectiveness of this integrated approach in identifying realistic inconsistencies and omissions within this specific legal domain.

2.9 Conclusions

This chapter has surveyed the key bodies of literature relevant to the proposed praxis project on utilizing Large Language Models and Knowledge Graphs for analyzing the consistency and completeness of legal documents, specifically municipal codes. First tracing the rise of LLMs, driven by the Transformer architecture (Vaswani et al., 2017), acknowledging their remarkable language processing capabilities but also highlighting their critical limitations concerning finite context windows (Liu et al., 2025) and computational scaling (Cong, 2024/10/16).

Knowledge Graphs were then introduced as a powerful paradigm for representing structured knowledge, capable of explicitly modeling entities and their relationships (Hogan et al., 2021). The potential of KGs to serve as a structured substrate for analysis, overcoming LLM context limits, was established. Bridging the gap between unstructured text and structured KGs necessitates Information Extraction, and we reviewed the core tasks of

Named Entity Recognition and Relation Extraction, noting the increasing role of deep learning and LLMs in achieving state-of-the-art performance (Xu et al., 2024).

The target application was framed by defining the crucial quality attributes of consistency, completeness and coherence (Zowghi & Gervasi, 2003), which are essential for the integrity and utility of formal documents. The significant challenges in maintaining these qualities manually, particularly in large and complex legal document corpora like municipal codes (Beth, 2018), were underscored, motivating the need for computational assistance. Finally, a review of related work situated this project within the context of ongoing research in LLM-driven IE and KG construction, KG-based analysis, and AI applications in the legal domain, highlighting the novel integration and practical focus on consistency and completeness checking for municipal ordinances.

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

Chapter 3: Methodology

3.1 Introduction

This praxis is intended to take large documents and convert them into a knowledge graph that can be used to check for completeness and consistency.

This chapter will detail how the goal of this praxis is accomplished.

The following sections cover the Approach, the data used, the hyper parameters defined, the measurements tracked to show success or failure, a summary, and a restatement of the research questions.

3.2 Approach

My goal for this research is to convert large legislation documents into Knowledge Graphs using an LLM in a way that it can be used to determine Consistency and Completeness. The check for consistency and completeness is not included in this work. It is important that doing this does not require any specialized technical skills. The approach should be able to detect errors and notify the user of them in a human usable way. That is, any error identified should be able to be traced back to the location or locations where it was found in the input documents. Ultimately, this approach should work with any large document or document set so it should not be focused solely on legal documents.

To accomplish all of this I have developed an architecture based on prior work. On the left is a shape to represent a set of input documents. Next is a box that represents the processing of the documents into chunks. The output of the chunks go through Input Processing by an LLM to create a an in memory JSON KG based on each chunk. These KGs are stored in Short Term Memory. Next is a process that runs occasionally to consolidate Short Term Memory into Long Term Memory. While a more robust solution would rn this consolodation as a separate process. For this praxis I will run the consolodation whenever Short Term Memory is full. It collapses duplicates and ensures that everything has a type. Next there is a process that runs agianst Long Term Memory on a periodic bases. Insert a tikz drawing of the architecture.

Long Term Memory Consolidation It analyzes a randomly selected subset of the Long Term Memory KG to look for. There are six steps:

1. Mark entities with DoNotChange label
2. Assign ISA relationships to entities
3. Assign PARTOF relationships to entities
4. Merge Type Entities
5. Arrange Type Hierarchy
6. Change Entity Type

3.3 Data

For this praxis a set of large documents is needed for both development and testing. The laws of townships of Pennsylvania serve this purpose well. They are large documents, publicly available, and have lots of interrelationships within them. As a Supervisor for the Township of Easttown, I have the ability to download any township laws from the General Code repository.

I can download it as a PDF or DOCX file. For now, I have selected to use DOCX as more information is stored in these files.

EDA of documents

Pictures, statistical tables

3.4 Hyper Parameters

There are many hyper parameters that can be tuned in this process. This section describes the critical ones and how they will be varied.

3.4.1 LLM Used

The facility will be set up to selectively use Gemini, ChatGPT, Claude, or oLlama for all of the LLM work. Test will be run with each one. The goal is to find out which one provides the best results with the fewest mistakes.

3.4.2 Document Chunking

Three methods of chunking will be used. One will be to divide the document in equal sized chunks of paragraphs. One will be to run through the documents two or more times using the using the same size chnk but varying the size of the first chunk. The third is to set a chunk size but overlap the chunks by a percentage. Various percentages such as 10%, 20%, and 50% will be tried. The goal is to address the entities that are at a chunk boundary.

3.4.3 Processing

There are several asynchronous processes that can run. These are the Short Term Memory Consolidation and Long Term Memory Processing. Each can be started with a limit. However, the limit can only occur on well defined boundaries. Determining how often to run then and how long to let them run are hyper-parameters that can be used. There is a Hyper Paramater to indicate when Short Term Memory is full. It is the number of KGs stored in STM. When started, the Short Term memory consolidation will be given the hyper parameter of a number of Knowlege Graphs to process. If there are this many or fewer, all of short term memory will be processed. Otherwise, some portion of Short Term memory will be left for later. Long term memory processing is started with a number of nodes to process. That set of nodes is randomly selected from all the nodes. All six steps defined are run against all of these nodes.

3.5 Tooling

This praxis uses Python, Google Collab, neo4j, ...

3.6 Measurements

This section describes what will be measured and how it will be measured. For much of this work, there is not well defined ground truth. The goal will be to determine the best fit.

3.6.1 Knowledge Graph Fit

How well does the generated knowledge graph fit the input documents.

3.6.2 Use for Consistency and Completeness

How well will the generated knowledge graph be useful for detecting inconsistencies and incompleteness.

3.6.3 Error Introduction

Carefully constructed errors will be interjected into the documents being used.

3.7 Methodological Limitations

Discuss limitations inherent *to your chosen methods*. For example, the limitations of LLMs in accurately extracting complex relations, the potential information loss due to chunking (even with overlap), the subjectivity in measuring "fit" or "usefulness".

3.8 Ethical Considerations

Since you're using publicly available legal documents, this is likely straightforward. A brief statement confirming no private data or human subjects are involved might be useful for completeness.

3.9 Conclusion

TBD

3.10 Research Questions

This methodological approach addresses the research questions and hypotheses outlined in Chapter 1 and repeated below:

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

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?

3.11 Research Hypotheses

Research will be conducted to 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.

Component	Description
Documents	The input documents
Document Processing	Converts each document into chunks
Input Processing	Convert each chunk to a KG in JSON format using an LLM.
Short Term Memory	In memory storage of KG in JSON format.
Consolidate Short Term Memory	Convert all the KGs in Short Term Memory into the

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.

4.2 Some Subsection

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.

4.3 Some Subsection

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

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.

4.4 Some Subsection

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.

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

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.

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

4.5 Some Subsection

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 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

4.5.0.1 Sub Subsection

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.

4.5.1 Subsection

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.

4.6 Some Section

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.

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.

4.6.1 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.

At the end of Chapter 3, you must restate your research questions and hypotheses exactly as they are. The format should follow the structure shown below. Please Ensure all text coloring is removed.:

RQ1:

RQ2:

RQ3:

H1:

H2:

H3:

Example:

The results from this Chapter address the research questions and hypotheses outlined in Chapter 1 and repeated below:

RQ1: How do Transformer encoders compare to Autoencoders in terms of accuracy, precision, and recall when detecting malicious insider threats?

RQ1: You will repeat your research question1 here?

RQ2: You will repeat your research question2 here?

RQ3: RQ2: You will repeat your research question3 here?

H1: RQ2: You will repeat your research Hyothesis1 here.

H2: You will repeat your research Hyothesis2 here.

H3: You will repeat your research Hyothesis2 here.

Leftover from chapter 2 - holding here

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.

4.7 Some Section

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.

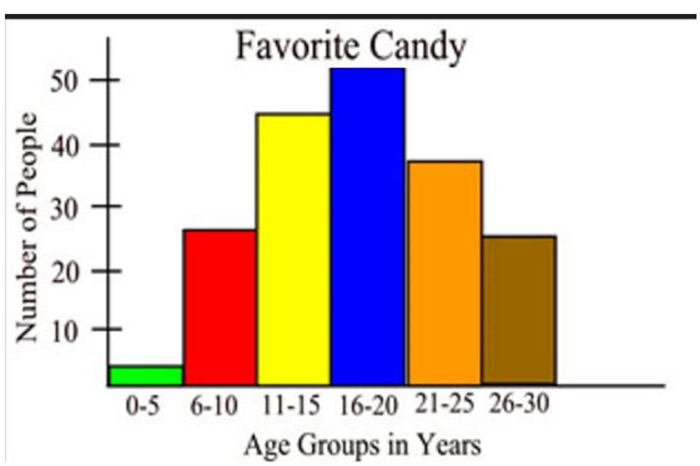

Figure 4.1: *Histogram of XYZ*

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.

4.7.1 Subsection 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 how to enter an equation and reference it. Equation 4.3 and 4.3

show how XYZ is implemented.

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

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

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.

4.8 Another Section

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 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

4.8.1 Subsection

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.

4.8.2 Subsection

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.

4.8.2.1 Sub subsection

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.

4.9 Summary

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 5: Discussion and Conclusions

5.1 Conclusion

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

5.2 Contribution to the Body of Knowledge

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

5.3 Recommendations for Future Research

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

References

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, June 1, 2000, June, 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, July). *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, December 19, 2022, 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, Jun 21, 2021, 2021, 2–11. https://doi.org/10.1145/3462757.3466085

Badshah, S., & Sajjad, H. (2024, May). *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, September). *Leading to annual revenue losses of hundreds of thousands of dollars.*

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, 1998, 172–183.

Brucker, A. D., & Wolff, B. (2019, November). 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, Jan 10, 2021, 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, 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, May). *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, August 2021, August, 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, December 11, 2023, 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, 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, 2024/10/16, June, 970–980. https://doi.org/10.2991/978-94-6463-540-9_97

Curely, D. (2024, September). *Municipal laws in pennsylvania townships, authored by multiple people over time, develop inconsistencies and are incomplete*.

Dhani, J. S., Bhatt, R., Ganesan, B., Sirohi, P., & Bhatnagar, V. (2021, July). *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, March). *Exponential growth brews 1 million ai models on hugging face*. https://arstechnica.com/information-technology/2024/09/ai-hosting-platform-surpasses-1-million-models-for-the-first-time/

Egyed, A. Instant consistency checking for the uml. In: In *Proceedings of the 28th international conference on software engineering*. New York, NY, USA: ACM, May 28, 2006, 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, Oren, Banko, Michele, Soderland, Stephen, Weld, & S., D. (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, May). *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, January). *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, July, 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, August). *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, November). *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, 1996.

Guo, B., Feng, C., Liu, F., Li, X., & Wang, X. (2023, January). 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, April). *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, August). *Foundations of semantic web technologies* (Anonymous, Trans.; 1st). Chapman; Hall/CRC. https://www.taylorfrancis.com/books/mono/10.1201/9781420090512/foundations-semantic-web-technologies-pascal-hitzler-markus-krotzsch-sebastian-rudolph

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

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, Nov 7, 2022, 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, 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, January). *Scaling laws for neural language models*. http://arxiv.org/abs/2001.08361

Knublauch, H., & Kontokostas, D. (2017, July). *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, July

1, 2020, 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, July 1, 2013, 2013, 1–6. https://doi.org/10.1109/ICCCNT.2013.6726698

Kumar, S. (2017, May). *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, July 7, 2021, 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, 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, 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, April 7, 2016, 260–270, 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, 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, Oct 26, 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, September 1, 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, March). *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, November). *Semantic segmentation of legal documents via rhetorical roles*. https://www.proquest.com/docview/2607084336

Minsky, M. (1974, June). *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, 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, 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, October). *Is semantic chunking worth the computational cost?* http://arxiv.org/abs/2410.13070

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

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, Apr 25, 2022, 2022, 260–263. https://doi.org/10.1145/3487553.3524253

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, Dec 21, 2022, 2023, 6383–6402. https://doi.org/10.48550/arxiv.2212.10947

Rau, A. (2024, September). *Municipal laws in pennsylvania townships, authored by multiple people over time, develop inconsistencies and are incomplete.*

Rdf 1.2 primer. (2025, April). 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_1}$0487735]. In: In *Ieee asia-pacific conference on computer science and data engineering (csde)*. ID: cdi$_{ieee_primary_1}$0487735. IEEE, 2023, December, 1–6. ISBN: 9798350341072. https://doi.org/10.1109/CSDE59766.2023.10487735

Sanders, J. (2024, October). *Municipal laws in pennsylvania townships, authored by multiple people over time, develop inconsistencies and are incomplete*.

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, October). *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, Jan 10, 2022, 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, April). *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, -06-17). *Gemini: A family of highly capable multimodal models* (I tried uploading the PDF but it would not upload. It is in my Reference file asnbsp;2312.11805v4 (1).pdf.). http://arxiv.org/abs/2312.11805

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, February). *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, April). *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, 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 (. Al-Onaizan, Yaser, Bansal, Mohit, Chen, & Yun-Nung, Eds.). In: *2024 conference on empirical methods in natural language processing* (. Al-Onaizan, Yaser, Bansal, Mohit, Chen, & Yun-Nung, Eds.). Ed. by Al-Onaizan, Yaser, Bansal, Mohit, Chen, & Yun-Nung. Miami, Florida, USA: Association for Computational Linguistics, Apr 17, 2024, 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, February). *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, March). *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