UDC 004.93(045)
DOI:10.18372/1990-5548.88.20970

Illia Savenko

COMPARATIVE ANALYSIS OF LLM-BASED GRAPH REPRESENATION CONSTRUCTION FOR DOMAIN-SPECIFIC DOCUMENTS

Department of Artificial Intelligence, Institute for Applied System Analysis, National Technical University of Ukraine "Sikorsky Kyiv Polytechnic Institute," Kyiv, Ukraine
E-mail: savenko.ilya@lll.kpi.ua

Abstract*—Recent advances in large language models have substantially improved natural language understanding and enabled their application across a wide range of domains. However, highly specialized fields such as law and medicine remain challenging because their documents often contain complex structures, domain-specific terminology, and dense logical dependencies. In such settings, large language models may produce errors when important structural information is not explicitly preserved in the document representation. To address this limitation, we propose a novel approach for document decomposition into graph-based representations that better capture the structural and semantic relationships within complex texts. We develop a method for processing raw legal documents from the Ukrainian domain using an LLM-based decomposition pipeline, transforming them into structured graph representations that can reinforce contextual retrieval and support retrieval-augmented generation. The proposed method improves document understanding by preserving key contextual dependencies and enhancing the representation of legal knowledge in downstream tasks.*

Keywords—Intellectual text analysis, natural language processing, text embeddings, graph representation, machine learning, LLM, RAG.

I. INTRODUCTION

Graph-based representation is an important area of data analysis that complements classical machine learning methods. In many applications, data cannot be adequately described as isolated observations and instead must be modeled as a set of interconnected entities and the relationships among them. Such a representation makes it possible to capture structural dependencies, contextual interactions, and higher-level organization within the data. By explicitly separating entities and linking them through meaningful relations, graph models provide a more expressive framework for analyzing complex domains where structure plays a central role [8].

As demonstrated by Perozzi et al. in DeepWalk [1], social relations can be effectively modeled through network representations, where vertices and their connections encode important structural information. Rather than relying solely on sparse graph topology, the authors show that learning latent representations of vertices in a low-dimensional continuous space provides a more expressive basis for statistical modeling. These embeddings capture neighborhood similarity, community structure, and other relational patterns that are difficult to exploit directly from raw graph data. This is particularly important in settings with sparse labels or missing information, where latent vertex representations support stronger generalization and improve downstream classification performance. In this sense, graph representation learning offers a principled way to transform incomplete and sparse relational data into a compact semantic structure that can be more effectively used by machine learning models [1].

Subsequent research significantly extended the foundations of graph representation learning by introducing models with stronger generalization capabilities and richer mechanisms for capturing structural dependencies. Graph Convolutional Networks (GCNs) [3] proposed an efficient framework for semi-supervised learning on graph-structured data, where node representations are updated through localized propagation over graph neighborhoods, allowing the model to jointly exploit node features and graph topology. Node2Vec [2] further advanced this direction by introducing a scalable method for learning continuous low-dimensional representations of nodes through biased random walks, thereby enabling more flexible exploration of network neighborhoods and improving the quality of learned embeddings for downstream tasks. More recently, Transformer-based architectures have been adapted to graph data. In particular, the Heterogeneous Graph Transformer (HGT) [7] extended attention mechanisms to heterogeneous graphs by incorporating node-type- and edge-type-dependent parameters, making it possible to model complex relational systems with

multiple entity and relation types at scale. Together, these developments demonstrate the evolution of graph representation learning from latent vertex embeddings toward expressive neural architectures capable of modeling increasingly complex and heterogeneous graph structures [2], [3], [7].

From the perspective of model training, GraphSAGE introduced an important advance in inductive graph representation learning by learning an aggregation function over a node's local neighborhood, rather than learning a separate embedding for each individual node. This formulation allows the model to generate representations for previously unseen nodes by sampling and aggregating features from their neighbors, which improves both scalability and generalization in evolving graphs. Building on this direction, PinSage[6] extended neighborhood aggregation to web-scale graph learning by combining graph convolution with efficient random-walk-based neighborhood sampling. In addition, PinSage introduced a training strategy based on progressively harder examples and an inference procedure suitable for extremely large graphs, making it possible to learn high-quality node embeddings on graphs containing billions of nodes and edges. Together, these works show how graph representation learning evolved from general inductive neighborhood aggregation toward scalable architectures capable of learning robust representations on very large and complex graph structures [4], [6].

II. Related Works

The structural representation of knowledge is essential in domains where documents are long, semantically dense, and logically complex [16]. In such settings, modern language-processing systems face several persistent challenges, including limited context focus over long documents, domain-specific vocabulary, temporal dynamics, confusing or highly similar statements, and broader concerns related to bias, ethics, and interpretability. These challenges are especially pronounced in high-stakes domains such as law and medicine, where automated systems must operate under strong requirements for precision, reliability, and explainability [11], [12].

To address these difficulties, recent research has increasingly relied on structured and graph-based representations that can preserve relations, dependencies, and domain-specific semantics more explicitly than flat text representations. In the legal domain, Xu et al. proposed the LADAN model for legal judgment prediction, where the central objective is to distinguish confusing law articles that are semantically close yet legally different. Their approach introduces a graph-based mechanism to learn subtle distinctions between related law articles and combines it with an attention mechanism to extract discriminative information from case facts more effectively [9].

Other work in legal AI has investigated the construction of knowledge graphs from legal texts in order to organize legal knowledge and improve the extraction of entities, attributes, and relations from unstructured documents. Such approaches are motivated by the need to transform complex legal language into structured representations that are more suitable for downstream reasoning and retrieval. At the same time, broader surveys of Legal NLP emphasize that the field still faces substantial open research challenges, including bias and fairness, privacy preservation, multilinguality, interpretability, explainability, and the efficient adaptation of language models to specialized legal settings [10], [11].

A similar tendency can be observed in the medical domain [15], where graph-based representations are used to integrate heterogeneous clinical information. Patient-Centric Knowledge Graphs (PCKGs) provide a framework for combining structured, semi-structured, and unstructured medical data into a unified representation of patient state, thereby supporting a more holistic view of diagnosis, treatment, and longitudinal care. The literature highlights that constructing such graphs requires solving difficult problems of ontology design, data integration, knowledge extraction, and reasoning over heterogeneous evidence sources [12].

In addition, Liu et al. introduced the Heterogeneous Similarity Graph Neural Network (HSGNN) for electronic health records, motivated by the fact that EHR data naturally form heterogeneous graphs but are difficult to process directly with standard heterogeneous GNNs because of issues such as hub nodes. Their framework uses a preprocessing stage that normalizes edges and decomposes the original EHR graph into multiple homogeneous graphs, which are then fused by an end-to-end graph neural network for diagnosis prediction. This design improves representation quality while mitigating limitations of direct graph modeling in complex clinical data [13], [14].

III. Problem Statement

A. General Description

Let a corpus of documents be denoted by:

$$D = \{d_1, d_2, \ldots, d_N\},$$

where each document d_i is a sequence of textual units,

$$d_i = \left(t_{i,1}, t_{i,2}, \ldots, t_{i,L}\right),$$

where each unit $t_{i,j}$ may correspond to a sentence, paragraph, section, article, clause, or any other structural segment. In complex domains, the sematic meaning of a document is determined not only by the local content of its units, but also by the relations between them, including hierarchy, references, temporal dependency, contradictions, specializations and attributions. Therefore, representing a document only as a flat sequence of tokens could lead to information loss.

The objective is to construct for each document d_i a graph representation:

$$G_i = \left(V_i, E_i, \varphi_i, \psi_i\right),$$

where V_i is the set of vertices; E_i denotes edges; φ_i assigns a type to each vertex, and ψ_i assign type to each edge. The vertex set may include entities, concepts, legal norms, temporal expressions, structural blocks or claims, where the edge set may encode semantic or logical relations.

Thus, the graph is intended to preserve the structural and sematic organization of the original text. Let the decomposition procedure be defined as a mapping

$$f_\theta = D \to \mathcal{G},$$

where $\mathcal{G}$ is the space of types attributed graph and θ denotes the parameters of model. For each document d_i

$$G_i = f_\theta\left(d_i\right).$$

B. Document-to-Graph Transformation

The first problem is to convert the documents to the graph structure. Given a raw set of domain documents d_i, infer graph G_i such that graph represents structural and contextual dependencies for downstream reasoning and retrieval. After graph construction, the corpus of graph space is

$$G = \left(G_1, G_2, \ldots, G_N\right).$$

For retrieval-augmented generation, let a user query be denoted by $q \in \mathcal{Q}$.

C. Relevance Defintion

The goal also includes retrieval alongside with a set of chunks, a set of graph elements that are most relevant to the query. Let the retrieval function be:

$$R : \mathcal{Q} \times \mathcal{G}_D \to 2^S,$$

where S is the space of candidate evidence structures and 2^S denotes the set of retrieved subsets. $\mathcal{Q}$ is the set of user queries and q is the user query entry from the set. Retrieved objects is defined as subgraphs:

$$S_q = R\left(q, \mathcal{G}_D\right).$$

The retrieved subgraph S_q should maximize relevance to the query while preserving relational evidence. Defining the objective more specifically:

$$S_q^* = \arg\max Rel\left(q, S\right) - \lambda Cost\left(S\right),$$

where $Rel\left(q, S\right)$ measures semantic and structural relevance; $Cost\left(S\right)$ penalizes excessive retrieval size or noise, and $\lambda > 0$ controls the trade-off.

IV. METHOD DESCRIPTION

This section describes the proposed architectural approach for constructing a knowledge graph aimed at linking complex entities in legal documents. The data used in this study consist of Ukrainian court documents from the legal domain. The source documents are provided in RTF format, and the main objective is to extract their structural topology and convert it into a graph representation. The extraction process is performed with the assistance of a large language model (LLM), which is used as the core component for structured information extraction.

Given the source legal documents, the first stage of the pipeline is pre-processing. At this stage, the documents are normalized and segmented into structurally meaningful units. The pre-processing procedure includes paragraph segmentation, identification of section numbers, and preservation of stable character offsets for each text span. This step is necessary to ensure that the extracted graph elements remain traceable to the original document. The source documents generally follow a relatively stable semantic structure, which includes a header containing legal-process metadata, such as the court name, decision date, and case number, followed by sections describing the facts and procedural history, legal reasoning, and the final ruling.

The main focus of this work is the LLM-based structure extraction stage. Modern language models make it possible to produce outputs in a constrained and structured format, which is essential for reliable graph construction. The extraction process is

organized into multiple passes. In the first pass, the system identifies the primary entities and relations within a single document. Specifically, it extracts the parties involved in the case, procedural events, the decision outcome, references to legal norms, and evidence pointers represented as paragraph identifiers together with character offsets. This stage produces an initial set of graph candidates grounded in the original text.

The second pass performs document-level consolidation. At this stage, the extracted elements are normalized and merged into a coherent per-document graph. This includes deduplication of parties mentioned multiple times within the same case, merging repeated descriptions of the same procedural events, and resolving overlaps among extracted references. The result of this stage is the final graph object for each document, where nodes correspond to normalized legal entities or structural components, and edges represent their semantic or procedural relations.

The third pass introduces cross-document linking. In this stage, top-level references between documents are identified and connected, allowing the system to extend the graph beyond a single case. This makes it possible to capture broader legal dependencies, such as references to related proceedings, cited decisions, or other linked legal materials. As a result, the proposed pipeline produces not only a document-level graph representation, but also a higher-level interconnected graph structure suitable for downstream retrieval and reasoning tasks.

D. Graph Construction

To support the graph construction process, we define a fixed set of node and edge types that represent the main structural and semantic components of a legal document. This schema serves as the basis for transforming unstructured text into a typed graph representation.

The set of node types is defined at the document level and includes the following categories:

- *Document*, representing the source legal document as a whole;
- *Party*, including persons, organizations, and public authorities involved in the case;
- *Event*, representing procedural or factual actions such as hearings, payments, deliveries, signings, breaches, or notifications;
- *Claim/Statement*, corresponding to assertions, positions, or arguments made by the parties;
- *Evidence Span*, representing textual fragments that provide evidential support for claims or events;
- *Legal Reference*, including references to laws, legal articles, contracts, or other normative sources;
- *Metadata*, covering additional contextual entities such as assets, locations, courts, and case identifiers.

The set of edge types specifies the relations between these node categories and defines the topology of the resulting graph. The main edge types are as follows:

- Party – Participation – Event, indicating that a party is involved in a particular event;
- Event – References – Document, linking an event to the document in which it is described;
- Claim – supported by – evidence span, representing evidential grounding for a claim;
- Claim – asserted by – party, identifying the source of a statement or argument;
- Claim – about – party, indicating the target or subject of the claim;
- Event – precedes – event, encoding temporal or procedural order between events;
- Party – related to – party, capturing relations between parties;
- Document – cities – legal reference, linking the source document to the legal norms or contracts it mentions;
- Claim – contradicts – claim, representing conflicts between statements.

In addition to their semantic type, all extracted entities and relations are associated with provenance and extraction metadata. Each graph element stores the source document identifier, the evidence span text, its positioning information within the source document, a confidence score, and the extraction run identifier. This metadata ensures traceability, supports error analysis, and allows the resulting graph to be audited with respect to the original legal text.

V. RESULTS

For the experimental evaluation, we created a test set to measure the accuracy of hierarchical structure extraction from legal documents across six different large language model variants. The selected models represent different scales and architectural configurations, enabling a comparative analysis of their extraction capabilities. The evaluated models included ChatGPT GPT-5.4, GPT-5.4-mini, GPT-5.4-nano, Gemini 3.1 Pro, Gemini 3 Flash, and Gemini 3.1 Flash-Lite. This comparison provides insight into the effect of model size and design on the extraction of hierarchical structures from complex domain-specific documents.

To evaluate the quality of graph extraction, we used the Jaccard similarity metric, computed separately for vertex sets and edge sets. This measure reflects the degree of overlap between the predicted graph structure and the reference annotation, thereby indicating how well the language models capture the internal entities and relations of the document.

Based on the results presented in Table I, it can be concluded that the task of hierarchical structure identification is feasible and can be quantitatively evaluated. The experimental results indicate that extraction quality is strongly influenced by model scale, with larger models consistently achieving higher similarity scores. In particular, the identification of graph nodes, which corresponds to entity recognition within the document structure, is generally more accurate than the extraction of graph edges, which requires correct identification of relations between entities. Among the evaluated models, Gemini 3.1 Pro achieved the best performance, obtaining the highest Jaccard similarity for both vertices and edges, which suggests the strongest overall capability for structured reasoning and graph construction. At the same time, smaller models demonstrated substantially weaker performance, indicating reduced capacity for reliable structure extraction in complex legal documents.

TABLE I. COMPARISON BETWEEN TEST SET AND LLM EXTRACTION FOR NODES AND EDGES

Model	J_e	J_v
GPT-5.4	0.75	0.89
GPT-5.4-mini	0.66	0.67
GPT-5.4-nano	0.54	0.53
Gemini 3.1 Pro	0.87	0.96
Gemini 3 Flash	0.67	0.71
Gemini 3.1 Flash-Lite	0.52	0.55

VI. CONCLUSIONS

This paper investigated the ability of modern large language models to decompose legal-domain documents into graph representations. As part of the research, we developed a complete extraction and validation pipeline for processing raw legal texts and generating structured graph outputs. For evaluation, a dedicated test set was created with the participation of domain experts, providing a reference standard for assessing model performance. This enabled a systematic comparison of the models in terms of how accurately their graph-based representations of raw documents matched the expert-defined structure.

To support the extraction process, we designed a structured data transformation pipeline composed of three stages for processing raw domain-specific documents. The first stage performs document segmentation into paragraph-level chunks with associated metadata. The second stage constructs graph representations from the extracted entities and relations. The third stage integrates the resulting components into a final unified graph. Such a multi-stage design provides a more detailed and reliable framework for analyzing graph extraction from complex textual data.

Graph construction with large language models represents one of the most promising directions in contemporary document intelligence because it combines the expressive reasoning capabilities of LLMs with the formal structure of graph-based knowledge representation. Unlike traditional information extraction pipelines, which often rely on separately engineered components for entity recognition, relation extraction, and schema alignment, LLM-based graph construction enables these tasks to be addressed within a unified framework. This is particularly important for complex domains, where meaning is distributed across long textual spans and depends on hierarchical, temporal, and semantic relations. In this setting, the LLM can serve not only as a parser of isolated facts, but also as a mechanism for reconstructing the latent structure of the document in the form of interconnected entities and relations. As a result, graph construction with LLMs can be viewed as a cutting-edge approach that bridges unstructured language understanding and structured knowledge modeling, thereby opening new opportunities for retrieval, reasoning, and downstream decision-support systems.

An important direction for future work is the application of model distillation to graph relation extraction. The goal is to develop smaller and more computationally efficient models that retain the performance of larger architectures in constructing graph-based relational structures. Such an approach would reduce resource requirements while

maintaining the practical effectiveness of the proposed system.

REFERENCES

[1] Bryan Perozzi, Rami Al-Rfou, Steven Skiena, DeepWalk: Online Learning of Social Representations, in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 701–710, 2014. https://doi.org/10.1145/2623330.2623732

[2] Aditya Grover, Jure Leskovec, node2vec: Scalable Feature Learning for Networks, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 855–864, 2016. https://doi.org/10.1145/2939672.2939754

[3] Thomas N. Kipf, Max Welling, Semi-Supervised Classification with Graph Convolutional Networks, in *International Conference on Learning Representations (ICLR)*, 2017.

[4] William L. Hamilton, Rex Ying, Jure Leskovec, Inductive Representation Learning on Large Graphs, in *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, 2017.

[5] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, Yoshua Bengio, Graph Attention Networks, in *International Conference on Learning Representations (ICLR)*, 2018.

[6] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, Jure Leskovec, Graph Convolutional Neural Networks for Web-Scale Recommender Systems, in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2018. https://doi.org/10.1145/3219819.3219890

[7] Ziniu Hu, Yuxiao Dong, Kuansan Wang, Yizhou Sun, Heterogeneous Graph Transformer, in *Proceedings of The Web Conference 2020*, pp. 2704–2710, 2020. https://doi.org/10.1145/3366423.3380027

[8] Fenxiao Chen, Yun-Cheng Wang, Bin Wang, C.-C. Jay Kuo, Graph Representation Learning: A Survey, *IEEE Access*, vol. 8, pp. 211799–211823, 2020.

[9] Nuo Xu, Pinghui Wang, Long Chen, Li Pan, Xiaoyan Wang, Junzhou Zhao, Distinguish Confusing Law Articles for Legal Judgment Prediction, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3086–3095, 2020. https://doi.org/10.18653/v1/2020.acl-main.280

[10] Qian Zhao, Tong Gao, Shanshan Zhou, Dongping Li, Yanyan Wen, Legal Judgment Prediction via Heterogeneous Graphs and Knowledge of Law Articles, *Applied Sciences*, vol. 12, no. 5, article 2531, 2022. https://doi.org/10.3390/app12052531

[11] Farid Ariai, Gianluca Demartini, Natural Language Processing for the Legal Domain: A Survey of Tasks, Datasets, Models, and Challenges, *ACM Computing Surveys*, 2024.

[12] Hassan S. Al Khatib, Subash Neupane, Harish Kumar Manchukonda, Noorbakhsh Amiri Golilarz, Sudip Mittal, Amin Amirlatifi, Shahram Rahimi, Patient-Centric Knowledge Graphs: A Survey of Current Methods, Challenges, and Applications, *Frontiers in Artificial Intelligence*, vol. 7, 2024. https://doi.org/10.3389/frai.2024.1388479

[13] Zheng Liu, Xiaohan Li, Hao Peng, Lifang He, Philip S. Yu, Heterogeneous Similarity Graph Neural Network on Electronic Health Records, 2021. https://doi.org/10.1109/BigData50022.2020.9377795

[14] Maya Rotmensch, Yoni Halpern, Amr Tlimat, Steven Horng, David Sontag, Learning a Health Knowledge Graph from Electronic Medical Records, *Scientific Reports*, vol. 7, article 5994, 2017. https://doi.org/10.1038/s41598-017-05778-z

[15] Hejie Cui, Jiaying Lu, Ran Xu, Shiyu Wang, Wenjing Ma, Yue Yu, Shaojun Yu, Xuan Kan, Chen Ling, Liang Zhao, Zhaohui S. Qin, Joyce C. Ho, Tianfan Fu, Jing Ma, Mengdi Huai, Fei Wang, Carl Yang, A Review on Knowledge Graphs for Healthcare: Resources, Applications, and Promises, *Journal of Biomedical Informatics*, 2025. https://doi.org/10.1016/j.jbi.2025.104861

[16] Yanjun Gao, Ruizhe Li, John Caskey, Dmitriy Dligach, Timothy Miller, Matthew M. Churpek, Majid Afshar, Leveraging Medical Knowledge Graphs Into Large Language Models for Diagnosis Prediction: Design and Application Study, *JMIR AI*, vol. 4, article e58670, 2025. https://doi.org/10.2196/58670

Received: March 01, 2026
Accepted: March 19, 2026
Published: April 19, 2026

Savenko Illia. Postgraduate Student.
Artificial Intelligence Department, Institute for Applied System Analysis, National Technical University of Ukraine "Ihor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine, (2023).
Research Interests: artificial neural networks, artificial intelligence, programming.
Publications: 2.
E-mail: savenko.ilya@lll.kpi.ua

І. М. Савенко. Порівняльний аналіз побудови графових представлень на основі великих мовних моделей для предметно-специфічних документів

Останні досягнення у сфері великих мовних моделей суттєво покращили розуміння природної мови та розширили можливості їх застосування в широкому спектрі предметних областей. Проте вузькоспеціалізовані галузі, зокрема право та медицина, і надалі залишаються складними для опрацювання, оскільки їхні документи часто характеризуються складною структурою, предметно-специфічною термінологією та щільними логічними залежностями. За таких умов великі мовні моделі можуть припускатися помилок, якщо важлива структурна інформація не зберігається явно в поданні документа. Для подолання цього обмеження запропоновано новий підхід до декомпозиції документів у графові представлення, що дає змогу точніше відображати структурні та семантичні зв'язки в межах складних текстів. Розроблено метод опрацювання неструктурованих юридичних документів українського домену з використанням конвеєра декомпозиції на основі великої мовної моделі, який перетворює їх на структуровані графові представлення, здатні посилювати контекстний пошук і підтримувати retrieval-augmented generation. Запропонований метод покращує розуміння документів завдяки збереженню ключових контекстуальних залежностей і підвищенню якості представлення юридичних знань у подальших прикладних завданнях.

Ключові слова: інтелектуальний аналіз тексту, обробка природної мови, текстові ембединг-подання, графове представлення, машинне навчання, великі мовні моделі, RAG.

Савенко Ілля Михайлович. Аспірант.
Кафедра штучного інтелекту, Інститут прикладного системного аналізу, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна.
Освіта: Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна, (2023).
Напрям наукової діяльності: штучний інтелект, машинне навчання, штучні нейронні мережі, програмування.
Кількість публікацій: 2.
E-mail: savenko.ilya@lll.kpi.ua