LEGRA: A Pipeline for Building Graph-Based Representations of Polish Court Rulings for Legal Retrieval-Augmented Generation

Szymon Dobrowolski and Waldemar Bauer *

Posted Date: 24 November 2025

doi: 10.20944/preprints202511.1742.v1

Keywords: legal knowledge graphs; retrieval-augmented generation (RAG); Polish administrative court rulings; hybrid semantic–lexical retrieval; graph databases

 doi:10.20944/preprints202511.1742.v1

Article

LEGRA: A Pipeline for Building Graph-based Representations of Polish Court Rulings for Legal Retrieval-Augmented Generation

Szymon Dobrowolski and Waldemar Bauer *

Department of Process Control, AGH University of Kraków, 30–059 Kraków, Poland
* Correspondence: bauer@agh.edu.pl

Abstract

Efficient access to similar legal cases is a crucial requirement for lawyers, judges, and researchers. Traditional text-based search systems often fail to capture both the semantic similarity and the relational context of legal documents [1]. To address this challenge, we present LEGRA, a novel graph-based dataset of Polish court rulings designed for Retrieval-Augmented Generation (RAG) and legal research support [2]. LEGRA is automatically constructed through an end-to-end pipeline: rulings are collected from public sources, converted and cleaned, chunked into passages, and enriched with TF-IDF vectors and embedding representations. The data is stored in a Neo4j graph database where documents, chunks, embeddings, judges, courts, and cited laws are modeled as nodes connected through explicit relations. This structure enables hybrid retrieval that combines semantic similarity with structural queries, allowing legal professionals to quickly identify not only textually related cases but also those linked through judges, locations, or legal references. We discuss the construction pipeline, the graph schema, and potential applications for legal practitioners. LEGRA demonstrates how graph-based datasets can open new directions for AI-powered legal research.

Keywords: legal knowledge graphs; retrieval-augmented generation (RAG); Polish administrative court rulings; hybrid semantic–lexical retrieval; graph databases

1. Introduction

The legal domain is inherently dependent on access to prior judicial decisions, forming the cornerstone of legal reasoning and interpretive consistency. Lawyers, judges, and researchers consult past rulings to identify applicable precedents, reason analogically, and ensure that individual judgments align with established legal principles. This process not only safeguards the coherence of jurisprudence but also supports the gradual evolution of case law. However, the growing volume of court rulings, increasing textual complexity, and the proliferation of overlapping jurisprudential domains have made the efficient retrieval and interpretation of relevant cases more challenging than ever. Traditional search mechanisms, which rely primarily on keyword-based matching, remain inadequate for capturing the nuanced semantic similarity between cases or the intricate relational context in which legal decisions are situated [1,3].

Recent advances in artificial intelligence (AI), especially the emergence of large language models (LLMs), have opened new prospects for improving access to, and understanding of, legal information. Retrieval-Augmented Generation (RAG) systems, which combine generative language models with structured access to external data sources, offer a means of augmenting human expertise through hybrid reasoning [2]. Instead of relying solely on memorized statistical patterns within language models, these systems dynamically retrieve external documents and condition generation on factual, contextually relevant material. In domains such as biomedicine, finance, and policy research, RAG-based frameworks have demonstrated superior retrieval quality, factual accuracy, and transparency.

 doi:10.20944/preprints202511.1742.v1

Nevertheless, their penetration into the legal domain and particularly within the Polish juridical context remains limited. Many legal professionals remain reliant on bespoke solutions or institutional databases which often do not leverage the full potential of AI or recent advances in search and retrieval.

Poland's judiciary, while rich in accessible court data following the 2012 introduction of open court portals, still lacks an integrated and openly available corpus that supports both semantic and relational querying. The existing systems maintained by governmental or institutional portals tend to focus on metadata-driven or keyword-based indexing, often without leveraging the deep semantic similarities between cases or the network of relations linking judges, legal articles, and prior citations. As a result, legal professionals often find themselves limited to repetitive, labor-intensive workflows when identifying relevant case law. The absence of richly annotated and connected legal resources creates a substantial bottleneck for both empirical legal research and the application of modern machine learning techniques, while also limiting opportunities for transparency and reproducibility in legal analytics. Building an open, graph-structured legal knowledge base would fill this gap and enable reproducible research, providing an empirical foundation for developing more explainable and context-aware legal AI systems.

Furthermore, the Polish legal system introduces unique complexities not only due to language but also because of historical and procedural specificities that make direct adoption of international AI-based legal retrieval solutions impractical. Existing datasets are often fragmented, either lacking consistent metadata or failing to capture the meaningful links between rulings, statutes, and judicial actors that underlie legal reasoning. Without a resource that reflects both the semantic content of legal texts and their relational structure, both practitioners and researchers are left without the tools needed for advanced, context-sensitive legal analysis.

To address this challenge, we introduce LEGRA a pipeline for constructing a graph-based representation of Polish court rulings designed to facilitate hybrid semantic and relational retrieval. LEGRA automatically acquires publicly available rulings, applies a multi-step preprocessing pipeline (including rigorous text cleaning, segmentation, and chunking), and generates two complementary types of representations: traditional TF–IDF vectors and neural embedding-based representations. These representations are integrated into a Neo4j graph database, where documents, text chunks, embeddings, judges, courts, and cited legal provisions are modeled as interconnected nodes. This configuration enables not only standard semantic similarity searches but also complex graph queries that incorporate relational constraints and support topological reasoning such as identifying all cases citing a specific legal article adjudicated by the same court division, tracing chains of legal influence across time, or finding all rulings semantically similar to a target judgment but delivered by a distinct jurisdiction.

The motivation behind LEGRA is to shift the focus of legal information retrieval from surface-level textual matching to contextual understanding and relational reasoning. By embedding documents in a semantic space and simultaneously preserving their contextual and structural relationships, LEGRA supports richer interpretability and higher retrieval precision. For instance, a legal scholar analyzing compensation cases under civil liability may retrieve judgments not only containing the relevant statutory references but also those exhibiting similar reasoning patterns, shared judicial panels, or analogous procedural circumstances. In practical terms, this hybrid approach not only reduces the cognitive load associated with manual document review, but also accelerates legal research and enhances the quality of argument construction within judicial practice.

Discussions with legal professionals in Poland have consistently underscored the need for AI-assisted tools that enhance transparency and efficiency without compromising judicial independence or interpretive nuance. Many practitioners emphasize that while automation has the potential to minimize administrative burdens, it must not erode trust in the rule of law or replace human ethical judgment. Studies from other jurisdictions, such as the use of AI-assisted adjudication in the Suzhou and Hangzhou Internet Courts, demonstrate tangible efficiency gains reducing trial duration by more than sixty percent while maintaining consistency and accessibility [4]. Moreover, empirical studies

 doi:10.20944/preprints202511.1742.v1

suggest that language models can achieve greater impartiality and uniformity in quasi-judgmental tasks compared to human judges, who may be influenced by cognitive biases or emotional framing [5,6]. Yet, these same studies emphasize that any use of AI in juridical contexts must preserve the principles of interpretability, due process, and accountability, supporting human decision-making rather than supplanting it. In this regard, LEGRA is conceived as a tool for augmenting, not replacing expert researchers and judges.

Within this paradigm, document analysis systems serve as assistive instruments summarizing, comparing, and linking judicial information while leaving ethical and interpretive judgment to human experts. LEGRA's graph-based architecture directly supports this goal by making reasoning chains transparent and queryable, allowing users not merely to obtain an answer but to trace the relational path that justifies it. By integrating semantic embeddings with explicit graph relations, LEGRA exemplifies a new direction in explainable legal AI, combining the rigor of structured databases with the flexibility of modern neural models.

Ultimately, the broader objective of LEGRA is to democratize access to advanced legal search technologies and foster an ecosystem of open, reproducible research in the Polish legal domain. By enabling comprehensive retrieval strategies that merge semantic similarity, relational context, and factual grounding, the system contributes to the modernization of legal analytics and lays a foundation for future RAG-based applications in law. This endeavor reflects a growing recognition that explainability and relational transparency are not ancillary features, but essential characteristics of any trustworthy AI deployed in judicial or decision-making environments.

2. Building the LEGRA Knowledge Graph

The journey from raw court documents to a navigable legal knowledge graph merges data engineering with domain insight. This section outlines the acquisition, structuring, and transformation of Polish administrative rulings, focusing on how both technical and legal constraints shaped LEGRA.

2.1. Data Sources and Legal Context

LEGRA's foundation is a comprehensive collection of Polish administrative court judgments, officially published in the Central Database of Court Rulings (CBOSA) [7]. This repository, mandated by the Act on Access to Public Information [8], offers standardized Rich Text Format (RTF) files enriched with metadata. The structure and completeness of documents varies, but the underlying sources remain uniform, supporting methodological robustness. Notably, each judgment includes a case identifier, date, judicial panel, and operative sentence, with additional domains such as legal theses or dissenting opinions appearing less frequently yet providing valuable context.

An illustration of a typical CBOSA judgment is provided in Figure 1. Here, structured metadata is linked directly to narrative sections, enabling LEGRA to transform both contextual and factual elements into graph nodes.

 doi:10.20944/preprints202511.1742.v1

I FSK 2079/24 - Wyrok NSA

Data orzeczenia	2025-05-28 *orzeczenie prawomocne*
Data wpływu	2024-12-04
Sąd	Naczelny Sąd Administracyjny
Sędziowie	Dominik Mączyński /sprawozdawca/ Roman Wiatrowski Ryszard Pęk /przewodniczący/
Symbol z opisem	6110 Podatek od towarów i usług
Hasła tematyczne	Podatek od towarów i usług Podatkowe postępowanie
Sygn. powiązane	I SA/Lu 197/24 - Wyrok WSA w Lublinie z 2024-07-26
Skarżony organ	Naczelnik Urzędu Celno-Skarbowego
Treść wyniku	Uchylono zaskarżony wyrok i decyzję II instancji
Powołane przepisy	Dz.U. 2024 poz 900 art. 70 par. 1, art. 70 par. 6 pkt 1, art. 70c *Ustawa z dnia 29 lipca 2005 r. o przekształceniu prawa użytkowania wieczystego w prawo własności nieruchomości (t. j.)*

TEZY

Dzień wszczęcia postępowania w sprawie o przestępstwo skarbowe lub wykroczenie skarbowe, w rozumieniu art. 70 § 6 pkt 1 Ordynacji podatkowej oznacza dzień, od którego istnieje związek między podejrzeniem popełnienia przestępstwa lub wykroczenia a niewykonaniem zobowiązania przez podatnika, niezależnie od tego, że postępowanie w sprawie o przestępstwo skarbowe lub wykroczenie skarbowe zostało wszczęte wcześniej, lecz w dniu wszczęcia postępowanie to nie miało związku z niewykonaniem zobowiązania przez podatnika, a dopiero na skutek rozszerzania zakresu tego postępowania związek ten został ujawniony.

SENTENCJA

Naczelny Sąd Administracyjny w składzie: Przewodniczący Sędzia NSA Ryszard Pęk, Sędzia NSA Roman Wiatrowski, Sędzia WSA (del.) Dominik Mączyński (spr.), Protokolant Agnieszka Plekan, po rozpoznaniu w dniu 9 maja 2025 r. na rozprawie w Izbie Finansowej skargi kasacyjnej E. sp. z o.o. w S. od wyroku Wojewódzkiego Sądu Administracyjnego w Lublinie z dnia 26 lipca 2024 r. sygn. akt I SA/Lu 197/24 w sprawie ze skarg E. sp. z o.o. w S. oraz Rzecznika Małych i Średnich Przedsiębiorców na decyzję Naczelnika Lubelskiego Urzędu Celno-Skarbowego w Białej Podlaskiej z dnia 25 stycznia 2024 r. nr 308000-COP.4103.53.2022.80 w przedmiocie podatku od towarów i usług za miesiące od czerwca do grudnia 2015 r. 1) uchyla zaskarżony wyrok w całości, 2) uchyla decyzję Naczelnika Lubelskiego Urzędu Celno-Skarbowego w Białej Podlaskiej z dnia 25 stycznia 2024 r. nr 308000-COP.4103.53.2022.80, 3) zasądza od Naczelnika Lubelskiego Urzędu Celno-Skarbowego w Białej Podlaskiej na rzecz E. sp. z o.o. w S. kwotę 193.867 (sto dziewięćdziesiąt trzy tysiące osiemset sześćdziesiąt siedem) złotych tytułem zwrotu kosztów postępowania kasacyjnego.

Figure 1. Sample Polish administrative court judgment retrieved from CBOSA, showing structured metadata and textual sections as processed in LEGRA [9].

2.2. Graph Modeling: Entities and Relationships

To bridge legal logic and technical infrastructure, LEGRA parses judgment files into granular fields, as summarized in Table 1. This mapping connects Polish legal terms to their graph representations,

 doi:10.20944/preprints202511.1742.v1

from "court" and "judges" to nuanced tags like "dissenting opinion." Each element becomes a node or attribute, reflecting both legal perspective and practical document variability.

Table 1. Comprehensive list of fields parsed and stored in LEGRA, including mandatory, optional, and rare metadata/sections.

Polish Term	English Equivalent	Node Type	Presence/Notes
	Core Metadata		
Sygnatura (id)	Case Identifier	:Dokument	Always
Typ	Judgment Type	:Dokument	Always
Data orzeczenia	Judgment Date	:Dokument	Always
Prawomocne	Final/Binding	:Dokument	Extracted from date line
Data wpływu	Filing Date	:Dokument	Optional
	Institutions and Persons		
Sąd	Court	:Sąd	Always
Sędziowie	Judges	:Sędzia	Always
Funkcja	Function	:Funkcja	If specified per judge
	Parties and Authorities		
Strony	Parties	:Dokument	Optional
Skarżony organ	Challenged Authority/Body	:Organ	Optional, multi
Pełnomocnik/Obrońca	Attorney/Representative	:Dokument	Rare (custom)
	Classification		
Symbol z opisem	Symbol (with description)	:Symbol	Optional
Hasła tematyczne	Thematic Keyword (Tag)	:Hasło	Optional
	Legal Links		
Sygn. powiązane	Related Case Signature	:Dokument + POWOŁUJE_SIĘ_NA	Optional
Powołane orzeczenia z tekstu	Case Citations (Text)	:Dokument + POWOŁUJE_SIĘ_NA	Auto-detected/NLP
Powołane przepisy	Referenced Legal Provisions	:Przepis	Optional, multi
	Decision		
Treść wyniku	Case Outcome/Result	:Wynik	Optional, multi
	Judgment Text		
Sentencja	Operative Sentence	:Sentencja	Always
Uzasadnienie	Justification/Reasoning	:Uzasadnienie	Optional
Tezy	Legal Principles/Theses	:Teza	Rare
Zdanie odrębne	Dissenting Opinion	:ZdanieOdrebne	Rare
Inne pola	Custom Fields	JSON in :Dokument.dodatkowe_dane	Dynamic
	Retrieval & Indexing		
Chunk	Text Chunk	:Chunk	Generated from texts
Embedding	Dense Vector Embedding	:Embedding	Optional
TF-IDF	TF-IDF Vector	:TFIDF	Always
Vocabulary	TF-IDF Vocabulary	:TFIDF_Vocabulary	Model meta

Key relationships - edges in the graph capture not only who made a decision but also how statutes and precedents connect cases. Table 2 illustrates the principal links, connecting courts to judgments, judges to roles, documents to legal provisions, and many more. This architecture is specifically designed for flexibility: new fields and relations can be added without breaking the schema, allowing LEGRA to adapt as legal data standards evolve.

Table 2. Principal relationships (edges) in the LEGRA graph model; mapping from document/content to legal, semantic, and technical dimensions.

Relation	From → To	Purpose/Description
`[:WYDAŁ]`	`Sąd` → `Dokument`	Court issued judgment
`[:BRAŁ_UDZIAŁ]`	`Sędzia` → `Dokument`	Judge participated
`[:PEŁNI_FUNKCJĘ]`	`Sędzia` → `Funkcja`	Judicial role assignment
`[:DOTYCZY]`	`Dokument` → `Organ`	Judgment concerns authority
`[:ZAWIERA_SYMBOL]`	`Dokument` → `Symbol`	Legal classification
`[:ZAWIERA_HASLO]`	`Dokument` → `Hasło`	Thematic tag/data category
`[:ZAKOŃCZONY]`	`Dokument` → `Wynik`	Final result link
`[:POWOŁUJE_PRZEPIS]`	`Dokument` → `Przepis`	Cites legal provision
`[:MA_SENTENCJĘ]`	`Dokument` → `Sentencja`	Sentence section link
`[:MA_UZASADNIENIE]`	`Dokument` → `Uzasadnienie`	Reasoning section link
`[:MA_TEZY]`	`Dokument` → `Teza`	Legal principle link
`[:MA_ZDANIE_ODREBNE]`	`Dokument` → `ZdanieOdrebne`	Dissenting opinion
`[:POWOŁUJE_SIĘ_NA]`	`Dokument` → `Dokument`	Precedent/related case
`[:MA_CHUNK]`	`Dokument` → `Chunk`	Chunked text section
`[:MA_EMBEDDING]`	`Chunk` → `Embedding`	Embedding for search
`[:MA_TFIDF]`	`Chunk` → `TFIDF`	TF-IDF representation

 doi:10.20944/preprints202511.1742.v1

2.3. Processing Pipeline

The LEGRA pipeline is designed to be robust, replicable, and modular—reflecting both the technical challenges and best practices of modern legal AI research. Below, each stage is described as it unfolds in the journey from raw court records to a structured, queryable legal knowledge graph.

Ingesting Data

The acquisition process begins with automated high-integrity retrieval of the source files. Using a Selenium-controlled headless browser [10], the pipeline simulates the navigation of Poland's CBOSA court judgments portal, bypassing CAPTCHAs and dynamic web forms to ensure broad and reproducible coverage. Detailed logs record every download, and filenames encode court, signature, and timestamp, offering full traceability and auditability—crucial to compliance and reproducibility in legal analytics pipelines.

Cleaning and Structuring

Raw court documents are supplied as RTFs, often containing inconsistent formatting or legacy code-pages. LEGRA uses industrial-grade libraries like Spire.Doc [11] and BeautifulSoup [12] to convert, sanitize, and structure texts. The pipeline restores Polish diacritics, standardizes line breaks, and harmonizes section titles, ensuring that all documents—irrespective of court or year—are made uniform in structure. This harmonization not only improves downstream parsing, but also future-proofs the system as evolving court formats are quickly accommodated through logic modularization.

Parsing Metadata and Relations

With input consistency achieved, custom extraction scripts traverse each document, identifying and standardizing both the core- and edge-case metadata fields (see Table 1). This includes essential features such as court, signature, type, and parties, as well as rare or nested elements such as legal theses and dissenting opinions. Entity recognition algorithms establish links among cases, judges, cited statutes, and references—explicit citations are joined with inferred precedent relations to populate a rich, navigable graph structure. Consequently, each document becomes a node in a dynamic, multi-relational network, enabling complex queries spanning years or jurisdictions.

Chunking and Indexing

Recognizing the complexity and length of legal texts, pipeline segments (chunks) judgements into semantically coherent passages using state-of-the-art NLP tokenizers [13]. Each fragment is simultaneously indexed with a sparse TF–IDF representation (through scikit-learn [14]) for keyword-based retrieval, and a dense neural embedding using the nomic-embed-text-v2-moe model [15,16]. The Nomic model, with over 1.6 billion multilingual sentence pairs (including 63 million in Polish), leverages a Mixture-of-Experts architecture and Matryoshka adaptation for high accuracy and efficiency. This hybrid approach empowers LEGRA to deliver both precise keyword search and semantic similarity retrieval, ensuring that no relevant precedent is overlooked, even when described using a varied legal language.

Compliance and Versioning

Reproducibility and regulatory compliance underpin every facet of LEGRA's design. All scripts are version-controlled, containerized with Docker [17], and parametrized with detailed configuration files, simplifying deployment in research clusters or cloud environments. The modular architecture supports dynamic schema extensions—new fields or node types can be easily added without disrupting prior data. As an additional advantage, LEGRA manages its Python dependencies using the modern `uv` [18] package manager, which can deliver installation speeds 10 to 100 times faster than legacy tools such as `pip` [19,20]. Full audit trails and data deletion options maintain strict legal anonymity and GDPR compliance, making the LEGRA pipeline suitable for both experimental research and operational deployments.

 doi:10.20944/preprints202511.1742.v1

2.4. Example Case: A Judgment's Journey

For illustration, consider the sample shown in Figure 2. Here, the core metadata (date, court, judges), narrative sections (operative sentence, justification) and citations are parsed and mapped into LEGRA nodes and edges instantly searchable by meaning, legal connection, or procedural pathway.

In general, LEGRA transforms static legal documents into a living knowledge graph: searchable, explainable, and ready for semantic exploration.

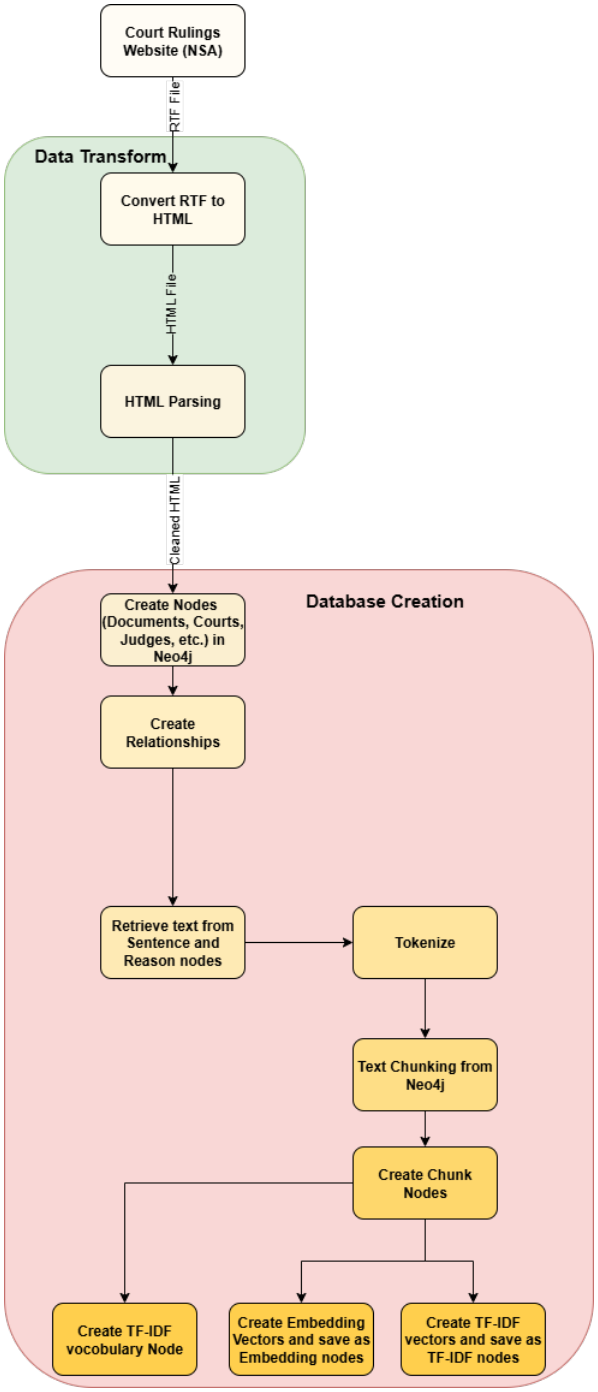

Figure 2. Overview of the LEGRA pipeline for legal knowledge graph construction. Court rulings are programmatically retrieved from the official CBOSA portal in RTF format, then converted and cleaned into structured HTML. Metadata and legal entities are extracted and modeled as graph nodes in Neo4j, while relationships are created to reflect case structure and citations. Judgment texts are tokenized, chunked, and indexed using both TF–IDF and dense vector embeddings, supporting hybrid semantic and lexical retrieval throughout the knowledge graph.

 doi:10.20944/preprints202511.1742.v1

3. Results

This section provides a comprehensive analysis of LEGRA's batch processing efficiency, disk and memory usage, and retrieval quality, all based on 15 standardized runs of 50 Polish legal judgments per batch. The results reflect the operation on the CPU alone and highlight the dependence on the external internet bandwidth during the essential document download phase, which remains the main bottleneck.

3.1. Experimental Hardware

All experiments were performed on a notebook with the following specifications:

- **Processor:** 12th Gen Intel Core i7-12700H (14 cores/20 threads, 2.3–4.7 GHz)
- **RAM:** 16 GB (15.6 GB available)
- **System:** 64-bit Windows OS, x64 architecture
- **ML Acceleration:** CPU only, no GPU utilized

While LEGRA benchmarks here used only CPU, significant acceleration, especially for text chunking and embedding generation, can be realized using modern NVIDIA/CUDA GPUs, thanks to their superior parallelization capabilities [21].

3.2. Pipeline Throughput and Storage Efficiency

e LEGRA processes 50-document batches with a mean runtime of 3.6 seconds per document (see Table 4), though this figure varies substantially with internet connection quality and host hardware. Only a fraction of disk usage per document remains in persistent storage; input and intermediate files can be safely deleted after graph population. Table 3 summarizes per-document disk requirements. Batch processing amortizes setup and model-loading overhead, resulting in improved efficiency as scale grows.

Table 3. Per-document disk usage before and after LEGRA processing. All core content is in Neo4j; temporary files can be removed post-processing.

Data Type	Mean Size per Doc [KB]	Removable?	Notes
Input RTF	16	Yes	Downloaded, parsed then disposable
Intermediate HTML	5.3	Yes	Temporary conversion file
Persistent Graph (Neo4j)	39	No	Net increase in DB after batch import
Total/document	60.3	Input/int. removable	15 runs, 50 docs each
Neo4j Baseline	516,000	No	Pre-allocated graph logs/schema

Table 4. Mean LEGRA pipeline stage processing time per document. Network and hardware parameters heavily influence total batch time.

Pipeline Stage	Mean Time [s]	Variance	Notes
RTF Download	1.74	High	Strongly dependent on internet speed
RTF→HTML Convert	0.12	Very low	Stable, local CPU operation
HTML Parse+Import	0.11	Low	Fast parsing, structure-dependent
Chunking/Indexing	1.61	Moderate	Can be accelerated with GPU
Total/document	3.6	–	Across n=15, 50-document batches

3.3. Processing Time per Pipeline Stage

Processing duration for each stage varies with the environment: document downloads may become the largest contributor under slower internet, while local hardware determines chunking and ML performance. Increased batch sizes further amortize setup and model initializations across more documents, yielding sublinear scaling benefits. Input and intermediate files (RTF, HTML) can always be safely deleted after graph import, keeping long-run storage demands exceptionally low.

 doi:10.20944/preprints202511.1742.v1

3.4. Metadata Generation and Graph Data Model

Each judgment is decomposed into a rich set of metadata: individual nodes for cases, courts, judges, legal provisions, thematic keywords, as well as chunked and embedded text indices. The resulting Neo4j knowledge graph structure, illustrated in Figure 3, allows advanced queries by text similarity or relational context. For example, queries may target all cases with a given judge, or filter for specific legal citations, leveraging both node attributes and the graph's connectivity. Chunking and embedding using hybrid TF-IDF and neural representations further enable semantic retrieval.

Figure 3. Sample visualization of the LEGRA knowledge graph: court, case, judge, and provision nodes connected via explicit legal and procedural relationships.

3.5. Chunking, Indexing, and Retrieval Performance

3.5.1. Chunking and Indexing

Text from each ruling is split into semantically coherent chunks (typically by paragraph or argument) using HuggingFace tokenization. These chunks are indexed by both sparse (TF-IDF) and dense (embedding) vectors, stored as nodes and relationships in Neo4j.

3.5.2. Retrieval Performance

Hybrid retrieval combines precise lexical matching (TF-IDF) with semantic generalization (embeddings), a strength in the legal domain due to its reliance on technical terminology. Evaluation of retrieval performance is shown in Figures 4, 5, and 6, comparing top result scores across methods.

Cosine similarity, a measure ranging from 0 (orthogonal) to 1 (identical), is used to assess document similarity for both vector types. The hybrid method consistently minimized the risk of missing relevant documents, placing the correct chunk at the top of results (example scores: 0.6839 for embeddings and the hybrid, 0.1691 for TF-IDF alone) [22].

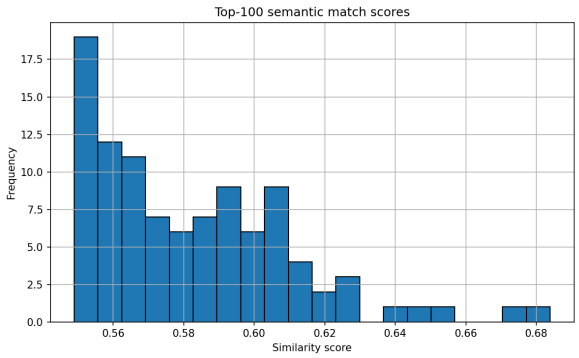

Figure 4. Semantic (embedding) search top-100 scores for test query on the evaluation corpus.

 doi:10.20944/preprints202511.1742.v1

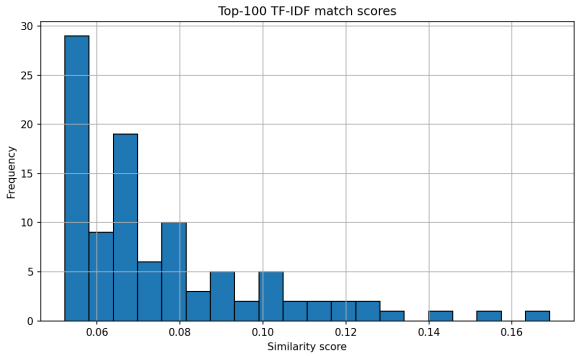

Figure 5. TF-IDF search top-100 scores for the same query.

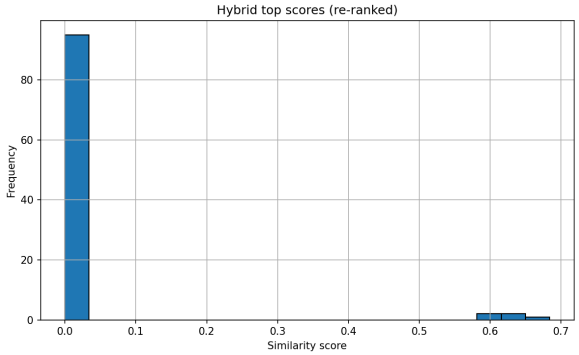

Figure 6. Hybrid search (TF-IDF + embedding) achieves best recall and precision in the test scenario.

Example evaluation query:
"Jaka była kwota zasądzona przez Wojewódzki Sąd Administracyjny w Gliwicach na rzecz strony skarżącej w sprawie ze skargi F. sp. z o.o. w W.?"
(*"What amount was awarded by the Administrative Court in Gliwice to the complaining party in the case involving F. Ltd. from W.?"*)
For this query, the relevant judgment (ID: II SA/Gl 636/25) was retrieved at the top of the result list in all tested scenarios, demonstrating the robustness of the hybrid approach.

3.6. Graph Versus SQL Databases: Query Power and Design Rationale

The LEGRA system's architecture, grounded in a property graph model, enables native support for multi-hop queries, deep path traversal, and explicable hybrid retrieval. Operations such as "find all cases judged by the same panel and citing article X" are efficient and natural in Neo4j, whereas the equivalent SQL operations would require multiple expensive JOINs and intricate schema management. LEGRA's graph-centric approach also makes it possible to evolve the schema seamlessly as additional entity types or relations are modeled essential for AI-augmented legal research workflows [23].

While SQL systems do excel at transactional integrity and bulk analytic aggregate queries, the legal case retrieval challenge in an LLM-driven environment is increasingly recognized as best served by graph databases, due to their flexibility, explainability, and hybrid semantic-structural querying capability [24,25].

3.7. Scalability, Operational Notes, and Best Practices

- **N**eo4j's 516 MB baseline is a fixed overhead; actual graph database growth per document is 39 KB.
- **A**ll temporary files can be deleted after graph construction, allowing LEGRA to efficiently manage very large corpora.

 doi:10.20944/preprints202511.1742.v1

- **O**verall memory usage remains well within 16 GB even at tested batch sizes; GPU/parallelization can raise future throughput ceiling.
- **B**oth disk and runtime are, however, variable network and local CPU/GPU being the principal moderating factors.

3.8. Summary

LEGRA delivers robust throughput and efficient disk/memory use for legal case retrieval at scale. Hybrid chunking and retrieval, explainable graph modeling, and careful operational architecture support both modern AI-driven analysis and high transparency for legal professionals.

4. Discussion

In this section, we analyze the results and discuss the advantages and challenges of the LEGRA system in the context of legal research.

4.1. Advantages of Graph Databases for Legal Retrieval

The graph-based representation of legal cases offers several advantages over traditional relational or text-based systems. Unlike relational databases, which require complex joins to combine related entities, graph databases such as Neo4j provide an intuitive way to model and traverse relationships between legal entities. This allows advanced analytical queries that can uncover patterns and connections between cases, courts, judges, and cited legal provisions revealing insights that would be difficult to extract from purely textual or tabular data.

Moreover, the Cypher query language designed for expressive and readable pattern matching and natively used in Neo4j is a rapidly evolving. Recent advances in large language models (LLMs) have shown growing proficiency in understanding and generating Cypher queries, paving the way for natural-language-driven interaction with legal knowledge graphs. This convergence between graph-based retrieval and AI-driven query generation opens new possibilities for interactive, explainable, and semantically rich legal information systems

4.2. Hybrid Retrieval Mechanism

The LEGRA hybrid retrieval mechanism, which combines TF-IDF and semantic embeddings, significantly improves the precision and relevance of search results. TF-IDF ensures that crucial legal terms are identified, while semantic embeddings capture contextual similarities, improving the system's ability to identify relevant cases even when the exact terms are not used. This combination reduces the risk of missing important documents and improves the interpretability of search results.

The results shown in Figure 6 demonstrate the performance of the hybrid retrieval approach. Combining embedding-based retrieval with TF-IDF allowed the system to balance lexical matching (through TF-IDF) with semantic similarity (via embeddings). This dual approach helps ensure that relevant rulings are retrieved based on both the specific terminology used in the text and the broader semantic context, improving the robustness of the search results.

4.3. Scalability and Performance Challenges

One of the primary challenges encountered during the project is scalability. As the size of the legal dataset increases (i.e., when including all court rulings throughout Poland), both the storage and computational requirements for embeddings and TF-IDF vectors increase. Storing of embeddings for each chunk of every case, along with full-text indexing, presents a significant storage and performance bottleneck.

To address this challenge, we propose restricting the initial search space using metadata-based filters available directly on the data acquisition website. These filters include the type of judgment (e.g., "Wyrok", "Postanowienie"), court level, date range, and keywords associated with the case. By applying such constraints before vectorization and indexing, the system can significantly reduce processing time and storage requirements while maintaining high retrieval precision. Additionally,

 doi:10.20944/preprints202511.1742.v1

this approach improves user interactivity by allowing domain experts to focus on specific subsets of the legal corpus rather than the entire database.

4.4. Limitations of Multilingual Embeddings

Although the use of the multilingual Mixture of Experts (MoE) embedding model (nomic-embed-text-v2-moe) provided strong performance in capturing semantic relationships in Polish legal texts, there are still challenges related to the quality of embeddings for the Polish language. Despite improvements in multilingual models, some domain-specific nuances in legal language may not be fully captured, particularly when compared to embeddings trained on English-language datasets.

Future improvements in multilingual embeddings are expected to address this gap, enhancing the system's ability to handle legal documents in languages other than English.

4.5. Future Directions

Future work will focus on extending the LEGRA pipeline into a complete Retrieval-Augmented Generation (RAG) system. This extension will integrate the Neo4j-based retrieval mechanism with large language models, allowing interactive exploration of legal knowledge that combines both semantic similarity and structural legal context.

In addition, a dedicated graphical user interface (GUI) is planned to enable users to interact with the system more intuitively. Through the GUI, users will be able to apply advanced filtering options (e.g., by court type, date range, or legal keyword) before triggering document retrieval or question-answering queries. This interface will also serve as a foundation for building a standalone web application, allowing legal professionals, researchers, and students to query the knowledge graph and visualize relationships between court rulings in real time.

Furthermore, future work will explore the use of BM25 (Best Matching 25) as an alternative to the traditional TF-IDF model for document retrieval. BM25 is a probabilistic-based ranking function that has been shown to perform better in information retrieval tasks, especially in cases where term frequency and document length vary. By incorporating BM25 into the pipeline, we aim to improve retrieval performance, particularly in cases involving longer or more complex legal texts, where TF-IDF might fall short in capturing nuanced relevance.

References

1. Malve, A.; Chawan, P. A Comparative Study of Keyword and Semantic based Search Engine **2015**. *4*. https://doi.org/10.15680/IJIRSET.2015.0411039.
2. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.t.; Rocktäschel, T.; et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *arXiv* **2020**. https://doi.org/https://doi.org/10.48550/arXiv.2005.11401.
3. Zeng, Z.; Bao, Z.; Lee, M.L.; Ling, T.W. A Semantic Approach to Keyword Search over Relational Databases **2013**. pp. 241–254. https://doi.org/https://doi.org/10.1007/978-3-642-41924-9_21.
4. Tahura, U.; Selvadurai, N. The use of artificial intelligence in judicial decision-making: The example of China. *International Journal of Law, Ethics and Technology* **2022**, *3*, 1–20. https://doi.org/10.55574/PYEB5374.
5. Watamura, E.; Liu, Y.; Ioku, T. Judges versus artificial intelligence in juror decision-making in criminal trials: Evidence from two pre-registered experiments. *PLOS ONE* **2025**, *20*, e0318486. https://doi.org/https://doi.org/10.1371/journal.pone.0318486.
6. Posner, E.A.; Saran, S. Judge AI: Assessing Large Language Models in Judicial Decision-Making **2025**. https://doi.org/https://dx.doi.org/10.2139/ssrn.5098708.
7. Centralna Baza Orzeczeń Administracyjnych — orzeczenia.nsa.gov.pl. https://orzeczenia.nsa.gov.pl/cbo/query. [Accessed 18-10-2025].
8. Ustawa z dnia 6 września 2001 r. o dostępie do informacji publicznej, 2001. Journal of Laws 2001 No. 112, item 1198. Consolidated version.
9. Administracyjny, N.S. Wyrok NSA I FSK 2079/24 z dnia 28 maja 2025. Centralna Baza Orzeczeń Sądów Administracyjnych (CBOSA), 2025. Accessed 2025-11-14.
10. Selenium Project. *Selenium WebDriver*, 2024. Version 4.35.

 doi:10.20944/preprints202511.1742.v1

11. E-iceblue Developers. *Spire.Doc for Python*, 2024. RTF/Word Document Conversion Library.
12. Richardson, L. *Beautiful Soup Documentation*, 2023. Beautiful Soup 4.12 Documentation.
13. Tokenizers — huggingface.co. https://huggingface.co/docs/tokenizers/index. [Accessed 19-11-2025].
14. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. *TF-IDF Vectorizer (scikit-learn)*, 2024. scikit-learn v1.4.2.
15. AI, N. nomic-ai/nomic-embed-text-v2-moe. https://huggingface.co/nomic-ai/nomic-embed-text-v2-moe, 2025. 475M parameters, 1.6B multilingual pairs (incl. 63M Polish), Mixture of Experts (MoE), Matryoshka representation learning, base model: FacebookAI/xlm-roberta-base, supports >40 languages.
16. Nussbaum, Z.; Duderstadt, B. Training Sparse Mixture Of Experts Text Embedding Models. *arXiv* **2025**. https://doi.org/10.48550/arXiv.2502.07972.
17. Docker Inc.. *Docker: Accelerated Container Application Development*, 2025. Version information and documentation available online.
18. charliermarsh. uv — docs.astral.sh. https://docs.astral.sh/uv/.
19. Python UV: The Ultimate Guide to the Fastest Python Package Manager — datacamp.com. https://www.datacamp.com/tutorial/python-uv.
20. uv: The Fastest Python Package Manager | DigitalOcean — digitalocean.com. https://www.digitalocean.com/community/conceptual-articles/uv-python-package-manager.
21. Chan, D.M.; Rao, R.; Huang, F.; Canny, J.F. t-SNE-CUDA: GPU-Accelerated t-SNE and its Applications to Modern Data **2018**. https://doi.org/https://doi.org/10.48550/arXiv.1807.11824.
22. Anthropic. Contextual Retrieval in AI Systems, 2024. Combining semantic and keyword search reduces retrieval failure rate by nearly half.
23. Rathle, P. The GraphRAG Manifesto: Adding Knowledge to GenAI. Blog post on Neo4j website, 2024.
24. Moghaddam, Z.Z.S.; Dehghani, Z.; Rani, M.; Aslansefat, K.; Mishra, B.K.; Kureshi, R.R.; Thakker, D. Explainable Knowledge Graph Retrieval-Augmented Generation (KG-RAG) with KG-SMILE **2025**. https://doi.org/https://doi.org/10.48550/arXiv.2509.03626.
25. Tiddi, I.; Schlobach, S. Knowledge graphs as tools for explainable machine learning: A survey. *Artificial Intelligence* **2022**, *302*, 103627. https://doi.org/https://doi.org/10.1016/j.artint.2021.103627.