School of Engineering

Centre for Artificial Intelligence (CAI)

Bachelor Thesis Computer Science

Integrating a Jurivoc Topic Layer into Graph-Enhanced Retrieval over Swiss Federal Supreme Court Decisions

Author(s)		Pascal Küng Manuel Strenge
Main supervisor		Prof. Dr. Mark Cieliebak
Secondary supervisor		Manuela Hürlimann
Date		05.06.2026

Abstract

Legal research over Swiss Federal Supreme Court decisions is difficult not because case law is unavailable, but because relevant decisions are often connected by doctrine rather than by identical wording or explicit citations. Keyword search, dense embeddings and citation-graph traversal capture important retrieval signals, but they do not explicitly model legal concepts as shared entities between decisions. This thesis investigates whether Jurivoc, the legal thesaurus of the Swiss Federal Supreme Court, can close part of this gap when integrated as a topic layer in a graph-enhanced retrieval system.

Building on a predecessor system for Swiss case-law retrieval, the thesis adds Jurivoc descriptors as first-class topic nodes in a Neo4j knowledge graph and links them to rulings, facts and considerations by automatic lexical assignment. The resulting topic layer contains 10,975 topics and 3,600,344 ADDRESSES_TOPIC edges. It is evaluated on a German-language corpus of Swiss Federal Supreme Court decisions using five directly comparable retrieval settings: keyword search, embedding search, embedding plus citation graph, topic plus keyword graph and topic plus embedding graph. A benchmark of 31 queries and 84 query-decision pairs is used, structured into colloquial concept search, doctrinal legal search and statute-anchored search, and measured with Recall@10, nDCG@10, MRR and F1@10.

On this benchmark, topic-aware retrieval improves performance on every aggregate metric. The strongest configuration, topic plus embedding graph, raises Recall@10 from 0.509 to 0.592 and nDCG@10 from 0.450 to 0.551 compared with the best inherited baseline. The benefit is not uniform: it is largest for statute-anchored and colloquial queries and smallest for doctrinal queries. The topic layer also remains imperfect, recovering only between one third and two fifths of the Court's manual Jurivoc assignments. Overall, the thesis shows that a controlled legal vocabulary can improve graph-enhanced legal retrieval on this corpus and benchmark, especially where citation and lexical signals alone do not already surface the relevant decisions.

Zusammenfassung

Die juristische Recherche in Entscheiden des Schweizerischen Bundesgerichts ist nicht deshalb schwierig, weil Rechtsprechung nicht verfügbar wäre, sondern weil relevante Entscheide häufig durch ihre Lehre und nicht durch identische Formulierungen oder explizite Zitate miteinander verbunden sind. Stichwortsuche, dichte Embeddings und Citation-Graph-Traversal erfassen wichtige Retrieval-Signale, modellieren rechtliche Konzepte jedoch nicht ausdrücklich als gemeinsame Entitäten zwischen Entscheiden. Diese Arbeit untersucht, ob Jurivoc, der Rechtsthesaurus des Schweizerischen Bundesgerichts, einen Teil dieser Lücke schliessen kann, wenn er als Topic-Layer in ein graphbasiert erweitertes Retrieval-System integriert wird.

Aufbauend auf einem Vorgängersystem für das Retrieval schweizerischer Rechtsprechung fügt diese Arbeit Jurivoc-Deskriptoren als eigenständige Topic-Knoten in einen Neo4j-Wissensgraphen ein und verknüpft sie durch automatische lexikalische Zuordnung mit Urteilen, Sachverhalten und Erwägungen. Der daraus entstehende Topic-Layer umfasst 10'975 Topics und 3'600'344 ADDRESSES_TOPIC-Kanten. Evaluiert wird er auf einem deutschsprachigen Korpus von Entscheiden des Schweizerischen Bundesgerichts anhand von fünf direkt vergleichbaren Retrieval-Einstellungen: Stichwortsuche, Embedding-Suche, Embedding plus Citation Graph, Topic plus Keyword Graph und Topic plus Embedding Graph. Verwendet wird ein Benchmark mit 31 Queries und 84 Query-Decision-Paaren, gegliedert in umgangssprachliche Konzeptsuche, dogmatische juristische Suche und gesetzesverankerte Suche, gemessen mit Recall@10, nDCG@10, MRR und F1@10.

Die Ergebnisse zeigen, dass topic-basiertes Retrieval die Leistung gegenüber dem Benchmark in allen aggregierten Metriken verbessert. Die stärkste Konfiguration, Topic plus Embedding Graph, erhöht Recall@10 von 0.509 auf 0.592 und nDCG@10 von 0.450 auf 0.551 gegenüber der besten übernommenen Baseline. Der Nutzen ist nicht gleichmässig verteilt: Er ist am grössten bei gesetzesverankerten und umgangssprachlichen Queries und am kleinsten bei dogmatischen Queries. Der Topic-Layer bleibt zudem unvollkommen, da er nur etwa ein Drittel bis zwei Fünftel der manuellen Jurivoc-Zuordnungen des Gerichts reproduziert. Insgesamt zeigt die Arbeit, dass ein kontrolliertes juristisches Vokabular graphbasiert erweitertes Legal Retrieval verbessern kann, insbesondere dort, wo Zitations- und lexikalische Signale allein schwach sind.

Preface

This thesis grew out of a shared fascination with the intersection of law, language and computer science. Legal research is not only a search problem. It is the task of finding structure, authority and meaning in a body of decisions whose relevance is often hidden behind doctrine, terminology and context.

Swiss Federal Supreme Court decisions offered a particularly meaningful setting for this challenge. Relevant rulings may address the same legal concept without using the same words or citing each other directly. This thesis explores whether Jurivoc, the legal thesaurus of the Swiss Federal Supreme Court, can help make such conceptual connections visible in a graph-enhanced retrieval system.

Our aim was not to replace legal research or legal judgement, but to investigate how technology can support them more intelligently. The project showed us that retrieval quality depends not only on algorithms, but also on representation: a system can only retrieve relationships it is able to model.

We are deeply grateful to Prof. Dr. Mark Cieliebak and Manuela Hürlimann for their supervision, guidance and valuable feedback throughout this thesis. Their support helped us sharpen the research question, strengthen the methodology and keep the work focused on both technical soundness and practical relevance.

We also thank Elio Petrini, whose predecessor project provided the technical and conceptual foundation on which this thesis builds. Our thanks further go to Jonas Hertner for OpenCaseLaw, which made it possible to work with structured Swiss case-law data at this scale. We are also grateful to the legal experts and reviewers who supported the construction and validation of the benchmark and gold standard, and to the ZHAW infrastructure and support teams for providing the technical environment required for this project.

Table of Contents

List of Figures ..... **VI**

List of Tables ..... **VI**

List of Abbreviations ..... **VII**

1 Introduction **1**

1.1 Motivation and Background 1

1.2 Problem Statement 2

1.3 Predecessor Work and Research Gap 3

1.4 Objectives and Requirements 4

1.4.1 Objectives 4

1.4.2 Requirements 4

1.5 Research Questions 6

1.6 Use Cases and Target Users 7

1.7 Scope and Delimitation 7

1.8 Contributions of this Thesis 8

1.9 Overview of the Thesis 9

1.10 Use of Generative AI 10

2 Theoretical Foundations **11**

2.1 Legal Information Retrieval 11

2.2 Dense Retrieval and Embeddings 12

2.3 Retrieval-Augmented Generation (RAG) 13

2.4 Graph-Enhanced Retrieval (GraphRAG) 14

2.5 Controlled Vocabularies and Legal Thesauri 15

3 Approach and Methods **17**

3.1 System Architecture (Overview) 17

3.2 Data Sources and Preprocessing 19

3.3 Baseline System Inherited from Petrini 22

3.4 Vocabulary Selection: TERMDAT vs. Jurivoc 23

3.5 Topic Assignment 24

3.6 Retrieval Settings 27

3.6.1 Setting A: Keywords (Petrini Baseline) 27

3.6.2 Setting B: Embeddings (Petrini Baseline) 28

3.6.3 Setting C: Embeddings & Citation Graph (Petrini Baseline) 28

3.6.4 Setting D: Topic & Keywords & Graph (This Thesis) 29
3.6.5 Setting E: Topic & Embeddings & Graph (This Thesis) 31
3.7 Evaluation Design 32
3.7.1 Gold Standard Construction 32
3.7.2 Relevance Model 33
3.7.3 Metrics 34
3.7.4 Execution Protocol 34
4 Results 36
4.1 Jurivoc Topic Assignment Statistics 36
4.2 Retrieval Performance 38
4.2.1 Aggregate Metrics across Five Settings 38
4.2.2 Results by Use Case Category 40
4.3 Per-Query Outcome Analysis 42
4.4 Summary of Empirical Findings 47
5 Discussion and Outlook 48
5.1 Interpretation of the Results 48
5.2 Answering the Research Questions 49
5.3 Comparison with Petrini's Baseline 50
5.4 Limitations and Validity Threats 51
5.5 Reflection on Objectives 52
5.6 Future Work 53
6 Lists & References 55
6.1 Bibliography 55
6.2 Glossary 61
7 Appendix 64
7.1 Benchmark Queries and Gold Standard 64
7.2 Heatmap per Metric 71
7.3 User Setup Guide 74
Prerequisites 74
Phase 1 — External services (PostHog optional) 75
Phase 2 — Docker Compose & data ingestion 76
Phase 3 — Run it locally (dev) 78
Phase 4 — Production (public domain) 79

List of Figures

Figure 1: System architecture of the retrieval prototype 18
Figure 2: Ingestion pipeline 20
Figure 3: Inherited graph schema 22
Figure 4: Extended graph schema 25
Figure 5: Topic-linking pipeline 26
Figure 6: Retrieval flow of Setting A 27
Figure 7: Retrieval flow of Setting B 28
Figure 8: Retrieval flow of Setting C 29
Figure 9: Retrieval flow of Setting D 30
Figure 10: Retrieval flow of Setting E 31
Figure 11: Distribution of the 133,005 rulings by number of assigned topics 37
Figure 12: Coverage of the manual Jurivoc indexing by the automatic assignment 37
Figure 13: Aggregate retrieval metrics across the five settings 38
Figure 14: Per query recall heatmap 40
Figure 15: Metrics by use case and setting 41
Figure 16: Returned documents for query 17 43
Figure 17: Returned documents for query 13 44
Figure 18: Returned documents for query 28 45
Figure 19: Returned documents for query 24 46
Figure 20: Per query nDCG heatmap 71
Figure 21: Per query MRR heatmap 72
Figure 22: Per query F1 heatmap 73

List of Tables

Table 1: Comparison of Petrini's project thesis and this bachelor thesis 4
Table 2: Functional Requirements 5
Table 3: Non-Functional Requirements 5
Table 4: Three use cases structuring the evaluation 7
Table 5: Classification of the 31 queries by retrieval outcome 42

List of Abbreviations

Abbreviation	Meaning
Abs.	Absatz, paragraph of a statutory article
aBV	alte Bundesverfassung, former Swiss Federal Constitution
AI	Artificial Intelligence
API	Application Programming Interface
Art.	Article
ATSG	Bundesgesetz über den Allgemeinen Teil des Sozialversicherungsrechts, Swiss Federal Act on the General Part of Social Insurance Law
BERT	Bidirectional Encoder Representations from Transformers
BGE	Entscheidungen des Schweizerischen Bundesgerichts, published leading decisions of the Swiss Federal Supreme Court
BGer	Non-leading ruling of the Swiss Federal Supreme Court
BGFA	Bundesgesetz über die Freizügigkeit der Anwältinnen und Anwälte, Swiss Federal Act on the Freedom of Movement for Lawyers
BM25	Best Match 25, also known as Okapi BM25
BT	Broader Term
BV	Bundesverfassung, Swiss Federal Constitution
CC0	Creative Commons Zero
DSG	Datenschutzgesetz, Swiss Federal Act on Data Protection
EGMR	Europäischer Gerichtshof für Menschenrechte, European Court of Human Rights
EMRK	Europäische Menschenrechtskonvention, European Convention on Human Rights
F1	F1 score, harmonic mean of precision and recall
F1@10	F1 score calculated over the top ten retrieved results
GlG	Gleichstellungsgesetz, Swiss Gender Equality Act
GPT	Generative Pre-trained Transformer
GraphRAG	Graph-based or graph-enhanced Retrieval-Augmented Generation
LLM	Large Language Model
MCP	Model Context Protocol
MRR	Mean Reciprocal Rank
nDCG	Normalised Discounted Cumulative Gain
nDCG@10	Normalised Discounted Cumulative Gain calculated over the top ten retrieved results
NT	Narrower Term
OR	Obligationenrecht, Swiss Code of Obligations

Abbreviation	Meaning
RAG	Retrieval-Augmented Generation
REST	Representational State Transfer
RoBERTa	Robustly Optimized BERT Pretraining Approach
RPG	Raumplanungsgesetz, Swiss Spatial Planning Act
RQ	Research Question
RRF	Reciprocal Rank Fusion
SCD	Swiss Federal Supreme Court Dataset
SchKG	Schuldbetreibungs- und Konkursgesetz, Swiss Debt Enforcement and Bankruptcy Act
SQ	Sub-question
StGB	Strafgesetzbuch, Swiss Criminal Code
TERMDAT	Terminology database of the Swiss Federal Chancellery
TF-IDF	Term Frequency-Inverse Document Frequency
UC	Use Case
UF	Used For, relation from a descriptor to a non-preferred term in a thesaurus
USE	Use relation from a non-preferred term to its descriptor in a thesaurus
ZGB	Zivilgesetzbuch, Swiss Civil Code
ZHAW	Zurich University of Applied Sciences

1 Introduction

1.1 Motivation and Background

"The difficulty seems to be, not so much that we publish unduly in view of the extent and variety of present-day interests, but rather that publication has been extended far beyond our present ability to make real use of the record."
Vannevar Bush, "As We May Think," *The Atlantic Monthly*, July 1945

Vannevar Bush's observation [1], made eighty years ago, remains highly relevant to legal information retrieval today. Legal researchers do not face a lack of legal information. Rather, they face the difficulty of identifying, connecting and interpreting the relevant information within a large and continuously growing body of legal documents [2].

Switzerland operates under a civil-law system in which statutes form the primary source of law [3]. Nevertheless, decisions of the Swiss Federal Supreme Court (*Bundesgericht*, BGer) play an important role in the interpretation and application of statutory law. As the supreme judicial authority of the Confederation under Article 188 I of the Federal Constitution, the Federal Supreme Court has the task of supervising the uniform application of federal law and protecting individual constitutional rights [4]. Statutory provisions are framed by the legislator as abstract and general rules, leaving it to the courts to apply them concretely in each individual case [3]; many provisions deliberately contain open-textured concepts, general clauses or evaluative standards that require judicial concretisation over time [5]. Leading decisions of the Federal Supreme Court are published in the official collection (BGE) and have strong guiding authority for the interpretation and application of Swiss law [4]; they therefore define how abstract legal norms are applied across the Swiss legal system.

The scale of this body of case law makes retrieval challenging. The Federal Supreme Court provides access to its leading decisions (*Entscheidungen des Schweizerischen Bundesgerichts*, BGE) back to 1875 and to all Federal Supreme Court decisions since 2007 in anonymised form. Non-leading decisions have been available online almost completely since 2000 and completely since 2007 [6]. In 2025 alone, the court received 7,947 new cases and resolved 7,883 cases [7]. The German-language corpus used in this thesis contains approximately 133,000 Federal Supreme Court decisions in total, comprising 108,000 non-leading rulings and 25,000 leading decisions.

The practical challenge is not only the size of the corpus, but also the nature of legal relevance. A decision may be relevant because it interprets a statutory provision, develops a doctrinal concept, addresses a comparable factual constellation or belongs to a line of case law. Legal research therefore requires more than finding text that looks similar to a query, and existing legal research tools, useful when the user knows the exact provision, term or case citation, struggle when research begins with a broader doctrinal question phrased without the canonical terminology. A practitioner searching for case law on proportionality, discrimination, burden of

proof, arbitrariness or data protection must then sift the results manually, distinguishing central authorities from marginally related cases.

The bottleneck is therefore retrieval quality. If relevant decisions are not retrieved, later summarisation or question-answering components cannot reliably compensate for this omission. This is particularly important for retrieval-augmented systems, where the quality of the generated answer depends directly on the quality of the retrieved context [8]. Improving retrieval is thus a prerequisite for building reliable legal information systems.

This thesis investigates whether a controlled legal vocabulary can improve retrieval over Swiss Federal Supreme Court decisions. The central idea is to integrate Jurivoc, the legal thesaurus used by the Federal Supreme Court for indexing decisions, as a conceptual topic-layer in a graph-enhanced retrieval system. By doing so, the thesis examines whether decisions can be connected not only through textual similarity or citations, but also through the legal concepts they address.

1.2 Problem Statement

The retrieval approaches used as baselines in this thesis do not explicitly model legal concepts as connecting entities between decisions. Keyword search, embedding-based retrieval and citation-graph traversal each capture a different aspect of legal relevance, but they leave a structural gap [2].

Keyword search captures lexical overlap and fails when the same doctrinal issue is expressed through different terms. Embedding-based retrieval reduces this vocabulary problem by retrieving semantically similar passages, but semantic proximity does not necessarily correspond to doctrinal relevance. Citation-graph traversal adds legal structure by following explicit references between decisions, but it only connects decisions that actually cite each other.

The remaining gap is conceptual. Two decisions may apply the same legal doctrine without using the same terminology and without citing each other. This is especially relevant for legal concepts that appear across different areas of law, such as discrimination, proportionality, burden of proof or arbitrariness. In such cases, the relationship between decisions exists at the level of doctrine but is not directly represented by words, embeddings or citation edges.

Wage discrimination illustrates the gap. Relevant decisions appear under varied formulations, *Lohngleichheit*, *Lohndiskriminierung*, *geschlechtsspezifische Lohndifferenz* and in both private and public employment. BGE 144 II 65 develops the doctrine of relaxed standard of proof and burden-reversal under Art. 6 GlG, citing earlier rulings such as BGE 125 III 368. Yet the vocabulary diverges across these decisions: a practitioner searching with *Lohnklasse* will not necessarily reach the presumption-of-discrimination reasoning (*Vermutung der Diskriminierung*) of the earlier rulings, even where the underlying doctrine is identical.

This thesis investigates whether Jurivoc-based topics can close this gap by linking decisions through shared doctrinal concepts, even when the decisions do not cite each other or use identical terminology. The central question is whether such concept-level bridges improve retrieval compared with keyword, embedding and embedding-plus-citation baselines.

1.3 Predecessor Work and Research Gap

This bachelor thesis extends a project thesis [2] completed at the ZHAW Centre for Artificial Intelligence in 2025, which developed a hybrid retrieval system for Swiss Federal Supreme Court decisions. The predecessor system ingests rulings from *entscheidsuche.ch*, segments them into facts and considerations, extracts citations and statutory references and stores the result in a Neo4j graph alongside dense embeddings in a Qdrant vector database.

On this infrastructure, the predecessor work implemented three retrieval strategies: keyword search over extracted terms, embedding-based semantic retrieval and an embedding-plus-citation-graph variant that follows explicit references between rulings during retrieval. These three strategies form the technical baseline for the present thesis.

The predecessor work demonstrated that combining semantic retrieval with graph traversal is a promising approach for Swiss case law. Petrini explicitly identified the integration of legal concept nodes as future work: *"With sufficiently high-quality keyword extraction, the resulting terms could be represented as nodes in the graph database [...] making it possible to identify conceptual relationships"* [2]. The predecessor graph itself, however, modelled only the relationships already present in the corpus, citations between decisions and references to statutes and contained no dedicated concept layer.

This is the research gap. Two decisions may address the same legal doctrine without citing each other and without sharing terminology; in a graph that models only citations and statute references, they remain disconnected. For legal research, conceptually related but non-citing rulings may nevertheless be highly relevant.

Aspect	Petrini project thesis	This bachelor thesis
Corpus & graph	BGE and BGer decisions from *entscheidsuche.ch*; rulings, facts, considerations, citations and law-article edges in Neo4j	Swiss Federal Supreme Court decisions (BGE and BGer) sourced from OpenCaseLaw; graph rebuilt but schema and basic structure reused as baseline
Retrieval settings	A: keyword-based retrieval; B: embedding-based retrieval; C: embeddings + citation-graph traversal	A–C conceptually inherited; plus D: topic + keyword-graph retrieval; E: topic + embedding-graph retrieval
Concept layer	No explicit topic or concept nodes; keywords only used as textual features in nodes	Jurivoc-based topics added as first-class entities in the graph

Relationships used	Explicit case-to-case citations and references from considerations to statutory provisions	Citations, statute references and concept links (topic assignments and thesaurus relations)
Evaluation setup	3 settings, 10 doctrinal queries from legal textbooks, evaluated with precision, recall and F1	5 settings, 31 queries across three use-case categories, evaluated with Recall@10, nDCG@10, MRR and F1@10
Focus	Shows benefit of combining embeddings with citation-graph traversal, but leaves concept-level structure and topics for future work	Investigates whether a domain-specific concept layer (Jurivoc topics) improves graph-enhanced retrieval over Swiss Federal Supreme Court decisions

Table 1: Comparison of Petrini's project thesis and this bachelor thesis

1.4 Objectives and Requirements

1.4.1 Objectives

This thesis evaluates whether Jurivoc-based topics, integrated into the predecessor's graph-enhanced retrieval system, improve retrieval of Swiss Federal Supreme Court decisions compared with keyword, embedding and embedding-plus-citation-graph baselines. The evaluation quantifies aggregate retrieval performance, identifies the query types for which topic-aware retrieval is most beneficial and characterises its principal error patterns.

1.4.2 Requirements

The system extension and evaluation design are guided by functional and non-functional requirements. The functional requirements define what the prototype must support to compare retrieval strategies meaningfully; the non-functional requirements ensure reproducibility, transparency and a clear limitation to retrieval evaluation rather than legal advice.

Functionally, all five retrieval settings must operate within the same backend architecture, on the same benchmark queries and against the same gold standard, so that differences in performance reflect differences between retrieval strategies rather than differences in input data. Each setting returns ranked top-k results with ranking information where available. The user interface supports the selection of each setting, which allows query-related parameters to be configured where necessary and displays retrieved documents per setting and ranking position; this supports both quantitative evaluation and qualitative error analysis.

The main functional requirements are summarised in Table 2:

Requirement	Description
Support five retrieval settings	The backend must support settings A-C and the new topic-aware settings D-E.
Return ranked top-results	Each setting must return ranked documents or chunks with ranking information where available.
Use a common benchmark	All settings must be evaluated on the same queries and gold standard.
Support frontend selection	The frontend must allow users to select and run each retrieval setting.
Visualise retrieved results	Retrieved documents should be displayed per setting and ranking position.
Allow query configuration	Query-related settings should be configurable where needed for testing and analysis.

Table 2: Functional Requirements

Non-functionally, the evaluation is tied to a frozen code version, documented model versions and fixed result files, so that the results reported in Chapter 4 can be reproduced. Inherited and newly developed components are clearly distinguished so that future work can build directly on the present contribution. The system is evaluated as a research prototype for retrieval and is not intended as a legal advice tool. Raw evaluation data is accessible through the repository.

The main non-functional requirements are summarised in Table 3:

Requirement	Description
Reproducibility	Evaluation must be tied to a frozen code version, documented models and fixed result files.
Data transparency	Data sources, queries, gold standard files and results must be documented.
Component separation	Inherited and newly developed components must be clearly distinguished.
Maintainability and handover	Documentation must allow future students to continue the work.
Raw data availability	Raw evaluation data is provided in the repository.

Table 3: Non-Functional Requirements

1.5 Research Questions

The objective and requirements outlined above lead to the central research question of this thesis. The thesis investigates whether adding a domain-specific conceptual layer to an existing graph-enhanced retrieval system improves the retrieval of Swiss Federal Supreme Court decisions.

The main research question is:

RQ: To what extent does the integration of Jurivoc-based topics into a graph-enhanced retrieval system improve retrieval performance for Swiss Federal Supreme Court decisions compared with keyword, embedding and embedding-plus-citation graph baselines?

This main question is divided into three sub-questions.

SQ1: How suitable is Jurivoc as a controlled legal vocabulary for constructing topics over Swiss Federal Supreme Court decisions?

This sub-question concerns the vocabulary layer itself. Before evaluating topic-aware retrieval, it must be assessed whether Jurivoc provides a suitable basis for representing legal concepts in the graph. This includes its coverage of relevant legal concepts, its thesaurus structure and its ability to support links between decisions and topics.

SQ2: For which query types do topic-aware retrieval strategies improve, degrade or leave unchanged retrieval performance compared with the baselines?

This sub-question addresses the fact that topics are unlikely to help all queries equally. Conceptual and cross-domain queries may benefit more from topic-based traversal than queries that are already well served by citation links or explicit statutory references. The evaluation therefore considers not only aggregate performance, but also differences across query groups.

SQ3: What error patterns explain cases where topic-aware retrieval underperforms and what design choices could mitigate them?

This sub-question focuses on failure modes. Topics may improve recall by connecting conceptually related decisions, but they may also introduce noise if concepts are too broad, ambiguous or incorrectly matched. Analysing such cases is necessary to understand the limitations of the approach and to identify directions for future improvement.

Together, these questions structure the thesis: SQ1 examines the suitability of the topic layer, SQ2 evaluates its retrieval effect, and SQ3 analyses its limitations. The overall aim is not only to determine whether Jurivoc-based topics improve retrieval, but also to understand under which conditions they are useful.

1.6 Use Cases and Target Users

Three use cases structure the evaluation in this thesis. They are defined by the construction of the benchmark queries, specifically, by the linguistic register of the query and by the type of legal information it provides. They correspond to three retrieval situations that differ in how strongly the query itself points toward the relevant legal concepts or statutory references. Of the 31 active benchmark queries, 8 are assigned to UC1, 15 to UC2 and 8 to UC3; "active" refers to the queries used in the evaluation after one of Petrini's ten inherited queries was dropped. The concrete assignment of queries to use cases and their use in the evaluation are described in Section 3.7.1 and in the appendix.

Use case	Typical user perspective	Query construction	Retrieval hypothesis
UC1 Colloquial concept search	Non-specialist user	Author-formulated queries in colloquial register, without specialised legal terminology	Topics are expected to provide moderate benefit by reaching relevant decisions despite the absence of explicit doctrinal anchors
UC2 Doctrinal legal search	Legal practitioner or researcher	Legally informed queries using doctrinal terminology and legal concepts	Topics are expected to provide the strongest benefit, because the query vocabulary aligns with Jurivoc descriptors
UC3 Statute-anchored search	Student or practitioner	Queries anchored in one or more specific statutory provisions	Retrieval is expected to be well served by explicit statutory-reference and citation-based signals; topics are not expected to add substantive benefit

Table 4: Three use cases structuring the evaluation

UC1 represents colloquial concept search, UC2 doctrinal legal search and UC3 statute-anchored search, the last serving as a control category in which a finding of degradation would indicate noise introduced by the topic layer rather than added value. Together, they test an initial directional hypothesis: Jurivoc-based topics were expected to help most for doctrinal queries, less for colloquial queries and least for statute-anchored queries, where explicit statutory references already provide strong retrieval signals. This expected pattern is later used to interpret the per-use-case retrieval results in Section 4.2.2.

1.7 Scope and Delimitation

This thesis is bounded by the following scope decisions, made explicit here to delineate what is and what is not claimed by the empirical results in Chapter 4 and the conclusions in Chapter 5. Validity-affecting exclusions are revisited in Section 5.4; all exclusions are taken up as future-work directions in Section 5.6.

Document corpus: The retrieval system operates exclusively on decisions of the Swiss Federal Supreme Court (*Bundesgericht*), comprising both leading decisions (BGE) and non-leading rulings (BGer), while reusing the basic graph structure of the predecessor system [2]. Decisions

of cantonal courts, federal administrative courts and the federal criminal court are not included, nor are legal commentaries, parliamentary materials, administrative guidelines or scholarly articles. The corpus is restricted to German-language decisions; the French and Italian language versions of the Federal Supreme Court's jurisprudence are not indexed and not evaluated.

Topic vocabulary: The controlled vocabulary used to construct topics is the German edition of Jurivoc, the trilingual legal thesaurus maintained by the Federal Supreme Court. The French and Italian editions are out of scope. A second candidate vocabulary, TERMDAT, was analysed during development and rejected on empirical grounds; the corresponding finding is reported in Section 3.4 and Section 4.1, but TERMDAT is not used in the evaluated system.

Statutory texts: Article-level statutory texts from the federal codes (BV, ZGB, OR) were parsed from the Fedlex Akoma Ntoso format during development but were not ingested into the production retrieval graph. Statutory references are represented through the inherited citation-extraction layer rather than through full statute-text nodes; the rationale is developed in Section 3.2.

Retrieval components not investigated : Fine-tuning of Swiss-legal embedding models is not part of this thesis; domain-adapted alternatives such as the *legal-swiss-roberta* family [9] are benchmarked separately but not deployed in the evaluated system. Graph-analytic methods that operate on aggregate structure, community detection, graph clustering and the hierarchical community-based summarisation of the Microsoft GraphRAG approach [10] are not investigated. The present work traverses explicit graph relationships (citations and topic links) rather than partitioning the graph into thematic clusters and summarising them.

Evaluation focus: The empirical evaluation measures *retrieval* performance, the system's ability to surface relevant decisions for a benchmark query. The quality of generated answer text is not evaluated; an LLM-as-judge or human-expert evaluation was scoped during development but deferred for time reasons. No user study with practitioners is conducted, no comparison with operational legal-search products is performed and no production-deployment concerns (latency at scale, authentication, anonymisation) are addressed. The system is not intended to replace professional legal research; the contribution of this thesis is the design, implementation and evaluation of a topic-enhanced retrieval layer that can serve as a foundation for further project, bachelor or master theses.

1.8 Contributions of this Thesis

This thesis makes four contributions, all extending the predecessor system documented in his project thesis [2] and jointly addressing the research questions stated in Section 1.5.

1) **Jurivoc as an explicit concept layer:** The German Jurivoc dump is parsed into descriptor-based topic entries and represented in Neo4j as a first-class topic layer. Based on the literature and project review conducted for this thesis, no prior documented implementation was found that uses Jurivoc as an explicit topic layer in a graph-enhanced retrieval system for Swiss case law.

2) **Two topic-aware retrieval strategies:** Two new retrieval settings are designed and implemented within the inherited LangGraph framework: topic-plus-keyword (Setting D) and topic-plus-embedding (Setting E). They extend the three predecessor strategies to five directly comparable configurations and isolate the contribution of the topic layer against keyword, embedding and citation-graph baselines.
3) **Benchmark for Swiss case-law retrieval:** A gold standard of 31 queries and 84 query-decision pairs is constructed, annotated by use case (UC1–UC3), and documented with sources and exclusion criteria. The benchmark is designed to probe lexical, semantic, doctrinal and statute-anchored retrieval situations within one consistent evaluation frame.
4) **Empirical evaluation of the topic layer:** All five settings are evaluated on the benchmark using Recall@10, nDCG@10, MRR and F1@10, with breakdown by use case and error analysis, where appropriate. The contribution of the topic layer relative to the baselines is quantified, the query types for which it is beneficial are identified and the principal failure modes are characterised.

1.9 Overview of the Thesis

The remainder of this thesis is structured as follows.

Chapter 2 establishes the theoretical foundations on which the work builds. It surveys legal information retrieval, dense retrieval and embeddings, retrieval-augmented generation, graph-enhanced retrieval over legal citation networks, controlled vocabularies and legal thesauri.

Chapter 3 describes the approach and methods. It positions the inherited baseline system [2], documents the vocabulary selection between TERMDAT and Jurivoc, details the integration of Jurivoc as a topic layer into the Neo4j graph, specifies the five retrieval settings under comparison (Settings A–E), and defines the evaluation design including the gold standard construction, the metric set and the execution protocol.

Chapter 4 presents the empirical results with limited interpretation; the broader discussion follows in Chapter 5. It reports the topic-assignment statistics, the aggregate retrieval performance of the five settings, the breakdown by use case (UC1–UC3) and the error analysis.

Chapter 5 discusses the results, answers the research questions stated in Section 1.5, situates the findings relative to the predecessor work, reflects on the limitations and validity threats and outlines directions for future work.

Chapter 6 lists the bibliography and glossary.

Chapter 7 contains the appendices including the full gold standard, the per-query evaluation results and a user setup guide.

1.10 Use of Generative AI

Generative AI systems were used in three distinct roles during this thesis, separated here because they correspond to different scientific accountability standards.

Writing support: Claude (Anthropic) and GPT-4 (OpenAI) were used to refine phrasing, restructure paragraphs and provide critical feedback on the argumentation. The conceptual content, the research design, the empirical analysis and the interpretation of results are the work of the authors. All AI-generated or AI-refined text was critically reviewed, edited and validated by the authors.

Methodological component: Large language models are an integral part of the retrieval pipeline evaluated in this thesis. Specifically, GPT-4.1 accessed through Microsoft Azure OpenAI is used for query expansion, topic discovery, relevance verification and response generation. This is a methodological choice that constitutes part of the system under evaluation, not a writing aid. The configuration is documented in Chapter 3.

Coding and debugging: Claude Code (Anthropic) and Codex (OpenAI) were used during system development for feature implementation, experimentation, code review, debugging and refactoring. All code produced or modified with AI assistance was critically reviewed, integration-tested and functionally validated by the authors, who retain full responsibility for the system's correctness and reproducibility.

2 Theoretical Foundations

2.1 Legal Information Retrieval

Information retrieval in the legal domain is its own research problem, recognised as such for over two decades. Legal text relies on concepts, references and structure rather than on the literal wording of the documents, which means that techniques developed for general-purpose retrieval must be reworked before they do useful work on legal material [11]. The question is also harder than it first appears, because what counts as a relevant decision is itself multi-dimensional and cannot be reduced to textual similarity alone [12]. And progress in the field, while substantial, has been uneven, with the absence of standardised test collections remaining a particular weakness on the evaluation side [13]. These observations frame part of the problem this thesis takes up.

In simple terms, legal information retrieval means finding, in a large legal collection, the documents that match what a user is looking for, usually expressed as a query or a set of keywords. Many classical systems rely primarily on lexical matching while two keyword-based methods are especially common. The first is term-weighting with TF-IDF, which gives more weight to terms that appear often in one document but are rare across the whole collection [14]. For example, if a decision contains the word *Lohndiskriminierung* several times and that word appears in only a few decisions overall, TF-IDF treats it as important. Words such as *Gericht*, *Beschwerde* or *Entscheid*, on the other hand, are far less useful, because they appear in almost every legal text. The second method, BM25, refines the same intuition within a probabilistic framework, dampening the effect of very frequent terms and normalising for document length [15], and it remains a strong baseline in legal case retrieval despite its simplicity [16]. This is not only background: the Neo4j full-text index that Setting A falls back on (Section 3.6.1), is scored with BM25, so it is one of the lexical signals the baselines actually use.

However, legal texts also have characteristics that make pure keyword matching difficult. Three aspects are especially relevant for Swiss legal retrieval:

1. **Synonymy**: a similar legal doctrine is often named differently. One example is the constitutional principle of proportionality (*Verhältnismässigkeit*) and the prohibition of excess (*Übermassverbot*). A system that matches one term can therefore easily miss the other.
2. **Polysemy**: one word can mean very different things depending on the area of law. The word *Sicherheit*, for example, can mean public safety in police law, the collateral that secures an obligation in contract law and the security a party may have to lodge with the court under civil procedure. A keyword system has no way of keeping these apart.
3. **Multilingualism**: the Federal Supreme Court decides in three different languages, which a system matching keywords in only one of them cannot follow.

The practical implication of these particularities is always the same: rulings are missed simply because they happen to be worded differently from the query. In retrieval terms, this lowers recall.

Current Tools and Databases: These limitations matter in practice, because some of the tools and databases that Swiss lawyers use to find case law today still rest largely on the keyword-based methods just described. The Federal Supreme Court runs its own official case-law database, which is operated through a search service provided by Eurospider. It works well for users who already know which rulings they are looking for, but it is built primarily on lexical and metadata search [17]. Swisslex, on the other hand, is much broader and includes commentaries and journals as well as decisions; it is well known and widely used in legal practice [18]. There are also free providers such as entscheidsuche.ch, which offers a search across the published decisions of Swiss courts and was created to improve access beyond the commercial databases [19]. A more recent resource is OpenCaseLaw, which provides an open, machine-readable dataset of Swiss case law and legislation, including published court decisions and federal and cantonal texts [20]. Its documented REST API, together with the fact that it returns the facts, considerations and Regeste of each decision in already structured form, is what made it so attractive as the main data source for this thesis.

2.2 Dense Retrieval and Embeddings

Lexical retrieval primarily relies on term overlap, whereas dense retrieval represents texts in a vector space and ranks them by semantic proximity. Two passages can therefore count as similar even when they do not share a single word, which is one way around the synonymy problem that held back lexical retrieval in Section 2.1. What happens underneath is conceptually the following: instead of scoring a decision by the terms it has in common with the query, the system turns each text into a vector, an embedding, and arranges those vectors in space so that texts about similar things end up close together. Retrieval then comes down to returning the decisions that sit nearest to the query in that space. The paper that introduced dense passage retrieval gives a good illustration of why this helps. The authors take the question "Who is the bad guy in *Lord of the Rings*?" and note that the passage which answers it describes the actor who played the villain Sauron, without ever using the words "bad guy". A keyword system has a hard time connecting the two, whereas a dense retriever can link "bad guy" to "villain" and still return the right passage [21]. The idea has since found its way into legal case retrieval, where the best results were reported for a combination of BM25, the method from Section 2.1, and dense retrieval over embeddings adapted to legal text [22].

These vectors are produced by a neural language model, and a major turning point was BERT[1] in 2019. Where earlier models read a text from left to right, BERT looks at the words on both sides of a position at the same time, so it can represent the same word differently depending on the context it appears in. Its authors call it "conceptually simple and empirically powerful" [23]. BERT on its own returns one vector per word, but comparing whole passages needs a single vector for the entire text. Sentence-BERT introduced a way of producing such a vector by training BERT in a siamese setup, so that semantically similar sentences end up close together in the vector space [24]. The closeness itself is measured by cosine similarity, which compares the direction of two vectors and not their length, so a short query and a long decision can be matched on equal footing

[1] BERT stands for Bidirectional Encoder Representations from Transformers.

[14]. The decisions whose vectors point most nearly the same way as the query are the ones returned first.

One practical obstacle remains. Comparing the query against every stored decision one by one is far too slow on a corpus of this size, so the embeddings are organised in advance to let a search reach the closest ones without inspecting them all. This is known as approximate nearest-neighbour search. The graph-based method used here, Hierarchical Navigable Small World graphs, links the vectors across connected layers so the search moves quickly from a rough to a precise match, and was shown to outperform the earlier open-source approaches [25]. It is the index used by Qdrant, the open-source vector database used in this thesis [26].

A last point matters for legal text especially. A model trained on ordinary web pages copes badly with legal wording, since expressions such as *Erwägung* or *Rechtsbegehren* barely occur there. Continuing the training on legal documents helps considerably. LEGAL-BERT reported gains over general-purpose BERT on English legal corpora [27], and for Switzerland there is Legal-Swiss-RoBERTa, pre-trained on a multilingual collection of Swiss legal texts [9].

In short, an embedding for each text, a way to measure how close two of them are and an index that keeps the search fast are what every embedding-based setting in this thesis depends on. How they are configured in practice, and how they fit together with the graph-based approach introduced next, is covered in Chapter 3.

2.3 Retrieval-Augmented Generation (RAG)

Sections 2.1 and 2.2 dealt with retrieval on its own. Retrieval-augmented generation, usually just called RAG, is what you get when that retrieval step is put in front of a language model that then writes an answer out of what was found [8]. The reason for doing this is that a language model on its own is fluent but unreliable and will happily produce a sentence that sounds right but is backed by nothing, which is what people mean when they say a model hallucinates. The basic recipe has three core components: retrieval of relevant documents for the query, augmentation of the model's input with those documents and generation of an answer conditioned on the combined input [28]. The grounding is what counts in a legal setting, since an answer a lawyer cannot trace back to a real decision is of little use to them and even products that already use RAG have been shown to give wrong or unsupported answers on a fair share of legal queries [29].

This thesis is concerned only with the first of those three components. The system does produce a final generated answer, as Chapter 3 describes, but the quality of that text is not what is measured here. A decision that is never retrieved cannot be recovered later, so the answer can only ever be as good as the documents the model was handed in the first place. One round of retrieval, however, is not always enough. Some questions can only be answered by putting several pieces together, where the first result points to what has to be looked up next, and going back and forth between retrieving and reasoning brings up more of the relevant material than asking once does [30]. This is much the same as following citations or topic links through a graph, which is what the graph-based settings of this thesis rely on. Holding these steps together is the

job of an orchestration framework. The one used here is LangGraph [31], which models the pipeline as a graph of steps; how the settings are wired into it, is again covered in Chapter 3.

2.4 Graph-Enhanced Retrieval (GraphRAG)

The word GraphRAG has come to mean two rather different things, so it is worth being clear from the start. In the narrow sense made popular by Microsoft, GraphRAG is a particular pipeline that first groups a corpus into communities of related entities and then writes hierarchical summaries of them, so that a model can answer broad questions about a whole collection at once [10]. In this thesis the term is used in the broad sense of retrieval augmented by graph-structured legal relationships, and the system is more precisely a graph-enhanced retrieval system. The difference is that the present work follows explicit links between individual decisions, rather than clustering the corpus and summarising it.

Legal case law lends itself to a graph because the documents already point to one another. Every time a decision cites an earlier one it creates a link, and the whole body of case law can be read as a network of decisions joined by these citations. The predecessor system on which this thesis builds already stores Swiss decisions and their citations as such a graph [2]. Citation links, however, are only one way two decisions can be related. Two rulings can turn on the same legal concept without sharing a citation or even the same wording, and a graph made only of citations leaves them apart. The present work therefore extends the graph with a second kind of edge, one that attaches each decision to the legal concepts it addresses, drawn from a controlled vocabulary that the next section introduces. The central question of this thesis is whether this second kind of edge improves retrieval over a graph built from citations alone.

A citation network has the property that a few decisions are cited very often, and most are cited rarely, and the pattern of citations can be used to measure how central a ruling is within a line of case law, in much the same spirit as the importance scores a search engine computes for web pages. Fowler and colleagues showed this for the entire citation network of the U.S. Supreme Court, where importance scores built from the citation structure picked out the leading precedents better than a plain count of citations could [32]. A concept-based graph has a different property: each concept node can connect multiple decisions that have been assigned to that concept, regardless of whether those decisions cite one another, so concepts act as bridges between rulings that the citation graph keeps apart.

Whichever edges the graph carries, the same retrieval strategies become possible. The simplest is neighbour expansion: once a promising decision is found, the decisions reachable from it through the graph are pulled in as well, on the assumption that a relevant ruling tends to sit among other relevant ones. Repeating this over several steps is the multi-hop idea from the previous section, now applied to graph edges rather than to reasoning. A further refinement is score propagation, where the relevance of one decision is allowed to flow along the edges to its neighbours, so that a ruling tied to several strong matches gains in score even when its own text is a weaker fit for the query. Recent work on legal case retrieval has shown that learning from this connectivity, and not from the text alone, pays off: CaseGNN encodes each case as a small graph

and learns over it with a graph neural network [33] while CaseLink builds a single graph spanning many cases and exploits the reference relations between them [34].

Representing legal concepts as explicit entities in a graph is not new, and earlier work on legal ontologies and knowledge graphs has modelled areas of law and their relations in this way [35], [36]. What this thesis adds is to draw the concept nodes from a controlled legal vocabulary, Jurivoc, and to put both kinds of edges into the same retrieval system, so that the contribution of each can be measured. Jurivoc is the subject of the next section. How the citation graph and the topic graph are wired into the retrieval settings, and how they are evaluated against one another, is described in Chapter 3.

2.5 Controlled Vocabularies and Legal Thesauri

Jurivoc is the trilingual (German, French and Italian) legal thesaurus of the Swiss Federal Supreme Court, developed with the Swiss law libraries and maintained by the Court's documentation service [37]. The service assigns its descriptors by hand, "intellectually," in the Court's own phrase, to the published leading decisions of the official collection (BGE) [38]. To our knowledge the online BGE has been Jurivoc-indexed from approximately 1990 onward, although we found no source documenting the precise start. Because Jurivoc is trilingual and, as a thesaurus, connects alternative wordings to a single preferred descriptor, one ruling can be reached across all three national languages and through different formulations of the same idea, which makes it the most directly relevant controlled vocabulary for retrieval over Swiss case law.

Such thesauri follow long-established conventions for controlled vocabularies, maintained today under the international standard ISO 25964 and documented by its working group [39]. Each concept is represented by a single descriptor, the only form used to index a decision, while variant expressions (non-descriptors) are linked to it by the UF/USE relation, which redirects an alternative wording to its official label; for example, the non-descriptor *eherecht* points to the descriptor EHE ("marriage"). Descriptors are linked to one another by narrower-term (NT) and the inverse broader-term (BT) relations, which build the hierarchy: EHE has narrower terms such as EHESCHLIESSUNG ("conclusion of marriage"), EHESCHEIDUNG ("divorce") and GÜTERRECHT ("matrimonial property law"). Jurivoc applies this model in a legal form and is largely monohierarchical: each descriptor has a single broader term, so the descriptors form a tree. Comparable vocabularies serve other domains, e.g. EuroVoc for European Union legislation [40] or AGROVOC in agriculture [41].

For this thesis, Jurivoc's value lies in supplying legally curated concept nodes. A decision linked to a descriptor becomes reachable through it even when it shares none of the query's wording and cites no other relevant decision.

Jurivoc must be distinguished from TERMDAT, the federal terminology database of the Swiss Federal Chancellery [42]. TERMDAT supports drafting, translation and consistent communication, providing terms, definitions and equivalents across Switzerland's official languages and English. Jurivoc is not primarily a translation database, but a legal thesaurus used for indexing case law, and that is why it, rather than TERMDAT, is the appropriate topic layer over

Swiss case law: its descriptors are legal indexing concepts, not translation entries. The selection between the two and Jurivoc's integration into the graph are set out in Sections 3.4 and 3.5.

3 Approach and Methods

This chapter describes how the topic-enhanced retrieval system was designed, built, and prepared for evaluation. The presentation follows the order in which the system is constructed: from the data it operates on, through the baseline inherited from Petrini, to the topic layer that constitutes the contribution of this thesis, and finally to the retrieval settings and evaluation design through which that contribution is tested.

Section 3.1 gives an architectural overview, distinguishing the components inherited from the predecessor system from those added here. Section 3.2 describes the data sources and their preprocessing: the BGer/BGE decision corpus, the OpenCaseLaw full-text reference used for ingestion and retrieval, and the statutory texts that were prepared but deferred from the evaluated pipeline. Section 3.3 documents the baseline system inherited from Petrini and states explicitly what was kept unchanged and what was modified.

The next three sections present the core contribution. Section 3.4 explains the choice of controlled vocabulary, motivating the choice of Jurivoc over TERMDAT. Section 3.5 describes how the Jurivoc thesaurus is turned into a topic layer in the knowledge graph, covering the topic nodes and thesaurus edges added to the inherited schema and the automatic lexical procedure that assigns decisions to topics. The manual Jurivoc labels curated by the Court are deliberately held out as an external reference to avoid leakage.

Section 3.6 then defines the five retrieval settings under comparison, three inherited from Petrini (A–C) and two introduced by this thesis (D, E), each running on the same query set under identical conditions so that the contribution of the topic layer can be isolated. Section 3.7 sets out the evaluation design: the construction of the gold standard, the metrics reported and the execution protocol that makes the results reproducible.

3.1 System Architecture (Overview)

The system reuses the architecture of the predecessor system by Petrini [2] and extends it at the points where the topic layer required new behaviour. Retrieval is served by a Python backend that sits between a Next.js frontend and three external services: a Neo4j knowledge graph, a Qdrant vector store and the Azure OpenAI API endpoint used for embeddings and answer generation. The backend is organised as a single path from request to result, moving through four stages: entry, dispatch, retrieval and storage. This lets a query be traced through one layer at a time. Figure 1 shows this path and the components along it.

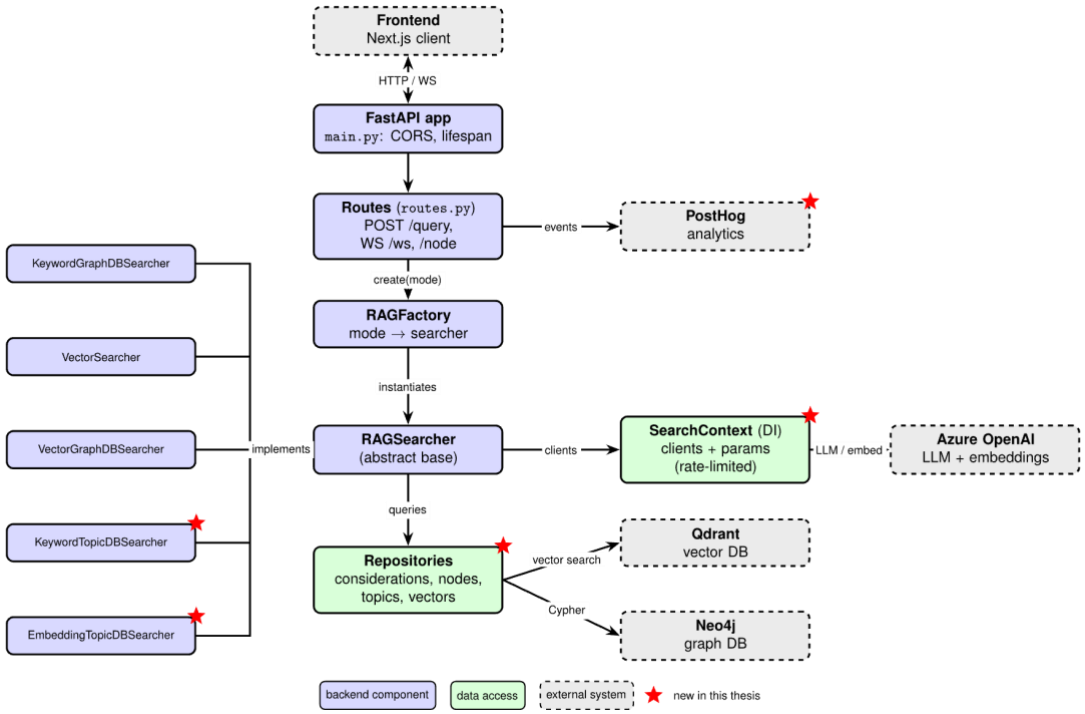

Figure 1: System architecture of the retrieval prototype

Purple components are backend retrieval components; green components are the data-access and shared-infrastructure layers; dashed boxes are external systems. Red stars mark components newly introduced or substantially extended in this thesis.

The remainder of this section follows a single request down the diagram, from the frontend at the top to the external systems at the bottom.

A request enters through the API layer, where main.py has already built the FastAPI application and a lifespan hook initialises the PostHog client and the Qdrant indexes. Routes.py exposes a synchronous POST /query endpoint and a streaming websocket /ws endpoint, alongside /node and /config helpers that serve the frontend.

From there the request is dispatched. The RAGFactory reads the requested searchMethod and instantiates the matching searcher, one of five implementations of the abstract RAGSearcher: keyword, vector, vector with citation graph, keyword with topics and embedding with topics. Routing all five behind a single abstraction is what allows them to be compared under identical conditions; each is a LangGraph state machine, detailed in Section 3.6.

The searchers do not hold their own connections. Instead, a SearchContext dependency-injection container supplies the shared infrastructure: the Neo4j driver, the Qdrant client, the LLM and the embedder, together with the tunable search parameters. Centralising these clients means every strategy runs against the same configuration, and every call to the LLM and embedder passes through one shared token-bucket rate limiter before reaching Azure OpenAI.

Retrieval itself goes through the data-access layer. The Repositories module (considerations, nodes, topics and vectors) holds the Cypher templates and the record-to-model mapping, so the

searchers express what they need rather than how Neo4j and Qdrant are queried. The external systems at the end of the path are Neo4j for the graph, Qdrant for the vectors, Azure OpenAI for the chat and embedding models and PostHog for usage analytics.

Five of these components are new in this thesis, marked with a star in Figure 1: the SearchContext container, the Repositories layer, the PostHog integration and the two topic-aware searchers (KeywordTopicDBSearcher and EmbeddingTopicDBSearcher). The FastAPI layer, the factory, the three baseline searchers and both storage systems are inherited from Petrini [2]. Because all five searchers share the same entry point, the same client configuration and the same data-access layer, they differ only in retrieval strategy, which is what let the evaluation in Section 3.7 attribute any difference in results to the topic layer rather than to the surrounding system.

3.2 Data Sources and Preprocessing

The primary corpus for this thesis consists of decisions of the Swiss Federal Supreme Court, comprising both leading decisions of the official collection (BGE) and non-leading rulings (BGer). Restricting the corpus to German-language decisions, as motivated in Section 1.7, corresponds to a publication landscape of approximately 133,000 Federal Supreme Court decisions, of which approximately 25,000 are leading decisions.

Departure from the predecessor source: In Petrini's thesis [2], entscheidsuche.ch [19] served as the source for all court decisions: the decisions were scraped as HTML and then loaded through an ingestion pipeline that reconstructed their internal structure from the markup. The initial plan for this thesis was to reuse the same pipeline to ingest BGE and BGer decisions. In practice, however, the existing document parser proved unreliable and a substantial share of decisions were parsed incorrectly. The cause is the markup itself: older decisions differ from more recent ones in structure, in section naming and in reference format, and the same variability exists across courts, so heuristics that match section headings in the running text fail on a non-negligible fraction of documents and propagate errors into the graph.

Two options were considered. The first was to extend the existing pipeline with more robust parsing logic and to validate it against the failing cases. The second was to replace the source with an alternative, already-structured dataset. The first option was implemented as an improved parser, but it produced a complex and brittle pipeline stage that still showed a significant failure rate, because the underlying variability across years and courts cannot be removed by better heuristics alone. The second option was therefore pursued, and two public datasets were evaluated against each other: OpenCaseLaw [20] and the Swiss Federal Supreme Court Dataset (SCD) [43]. OpenCaseLaw was selected. It provides the broadest coverage of the relevant decisions, its decisions are already parsed into the required sections (Regeste, Sachverhalt, Erwägungen), and the annotations needed for cross-references are preserved. Above all, integrating it, requires no scraping and no reconstruction of document structure: the structure that the predecessor pipeline had to recover heuristically is delivered directly.

The OpenCaseLaw dataset: OpenCaseLaw is a public, openly licensed dataset of Swiss case law maintained by Jonas Hertner [20]. At the time of writing it indexes more than 972,000

published court decisions from 109 courts in German, French and Italian, with coverage reaching from 1875 to the present. Each decision is fetched from its primary source, mostly the publishing courts themselves, with entscheidsuche.ch used for historical fills, and is stored together with structured metadata such as docket number, court, language, Regeste and full text. Beyond the full text, the project extracts a citation graph across the corpus: the raw layer contains roughly 8.65 million citation edges, of which about 8.09 million (93.5%) are resolved to canonical decision identifiers [20]. The corpus is published under CC0 on HuggingFace, as the dataset voilaj/swiss-caselaw, in Parquet shards with a daily delta [44], and is additionally accessible through a REST API and an MCP server at mcp.opencaselaw.ch [20]. For the decisions used in this thesis, the relevant coverage is 175,728 BGer decisions from 1986 to the present and 35,364 BGE decisions from 1875 to the present, across all three languages [20]; the German subset of this population is the corpus characterised at the start of this section.

For this thesis, the REST API was used to retrieve decisions by identifier. The pre-existing structural parsing into Regeste, Sachverhalt and Erwägungen, together with the resolved citation references, are the properties that make the dataset a structured replacement for the original entscheidsuche.ch scrape-and-parse approach.

Ingestion pipeline: The ingestion pipeline takes the cases listed in the HuggingFace dataset and produces the Ruling, Fact and Consideration nodes stored in Neo4j. As shown in Figure 2, it is organised in six stages:

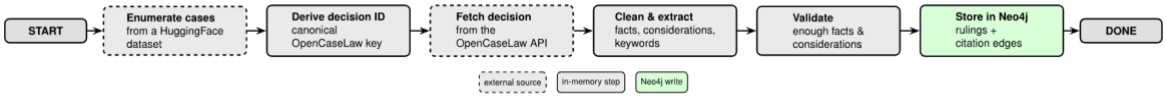

Figure 2: Ingestion pipeline

1. **Enumerate cases:** The list of Federal Supreme Court rulings (BGE and BGer) is read from the public HuggingFace dataset and filtered by language, by decision date and by a minimum full-text length of 500 characters, which removes empty or stub records.
2. **Derive decision ID:** Each case's metadata is canonicalised into a stable OpenCaseLaw identifier, for example bger_5A_350_2023. This identifier addresses the API and also deduplicates rulings across re-runs.
3. **Fetch decision:** The decision's structured sections, namely the statement of facts and the numbered considerations, are retrieved from the OpenCaseLaw REST API. Requests are rate-limited and cached on disk, so repeated runs do not re-hit the API. If the API holds no record for a given identifier, the pipeline falls back to the raw text shipped with the HuggingFace dataset.
4. **Clean and extract:** The text is normalised, and the facts, the considerations, the per-language keywords (extracted with spaCy and PyTextRank) and the cited rulings and statutes are combined into a single RulingData record. Because the API's citation field gives only ruling-level references without their position in the text, the citations are additionally recovered by regular-expression matching over the considerations, so that each citation can be attached to the consideration in which it occurs.

5. **Validate:** The record is admitted only if it carries the required structure: at least one consideration, the required metadata fields (date, language and instance identifier) present, and consistently numbered fact and consideration labels. Records that fail these checks are logged and skipped rather than written, which keeps incompletely structured decisions out of the corpus.
6. **Store in Neo4j:** The ruling, fact and consideration nodes are written in batches, together with their citation edges to other decisions.

In Figure 2, the two stages drawn with dashed borders, the HuggingFace dataset and the OpenCaseLaw API, denote external sources; the stages with solid borders denote in-memory processing; and the green stage denotes the write into Neo4j, the only stage that mutates the knowledge graph. The figure shows the decision-ingestion path only; the topic layer and the dense embeddings are added by separate steps in Section 3.5.

The pipeline runs eight worker processes in parallel and tracks batch state on disk, so a run can resume after an interruption without re-fetching decisions that were already processed. After ingestion completes, a PageRank score is computed over the citation graph and stored on the ruling and consideration nodes, giving a measure of how central each decision is within the citation network. The dense embeddings are not computed during ingestion: in a separate step, each consideration is embedded through the Azure OpenAI embedding model and the resulting vectors are transferred from Neo4j into the Qdrant vector store, so that similarity search over considerations and traversal of the citation graph can co-operate within a single query at retrieval time. Because the underlying dataset is updated daily, the figures reported in this section refer to a fixed snapshot.

Language scope: Although OpenCaseLaw and the underlying HuggingFace dataset expose French and Italian decisions as well, only the German subset is enumerated and ingested. The reason is methodological rather than technical. The thesis isolates a single variable, namely whether a controlled-vocabulary topic layer improves retrieval, and evaluating that within one language keeps the comparison between the baseline and the topic-aware settings clean. Extending across the three official languages would introduce a second, orthogonal dimension of difficulty in the form of cross-lingual matching, alignment of the trilingual Jurivoc editions and language-dependent retrieval behaviour, and would make it harder to attribute any measured difference to the topic layer alone. Multilingual topic-enhanced retrieval is therefore left as a distinct line of work, taken up in Section 5.6.

Statutory texts: Article-level statutory texts from the federal codes (BV, ZGB and OR) were parsed from the Fedlex Akoma Ntoso format during development [45], and an ingestion script was implemented, but the full statute texts were not ingested into the evaluated retrieval graph. Statutory references are instead represented through the inherited consideration-level citation extraction, which provides sufficient article-level anchoring for the retrieval evaluation. Integrating the full statute texts as first-class nodes is therefore treated as future work (Section 5.6).

3.3 Baseline System Inherited from Petrini

This thesis builds on the retrieval system developed in the preceding project thesis by Petrini [2], which provided the data pipeline, the graph and vector databases and the three retrieval strategies that serve as the comparison baseline for the topic layer. The sections below set out what was carried over without change and what was altered.

Inherited architecture: Court rulings are read, cleaned and parsed into a Neo4j graph of ruling, fact and consideration nodes, connected by citation edges and references to law articles. The considerations are embedded and the resulting vectors are held separately in Qdrant. A FastAPI backend orchestrated with LangGraph exposes the retrieval strategies and an LLM (GPT-4.1) converts the retrieved nodes into a response. The inherited graph schema, decisions decomposed into ruling, consideration and fact nodes and linked by citation edges and law-article references, appears in Figure 3. The pipeline runs in resumable batches, and once ingestion is complete a PageRank score over the citation graph captures how central each decision is. The data pipeline, the Neo4j/Qdrant split, the backend, the PageRank ranking and the batch design all remain unchanged.

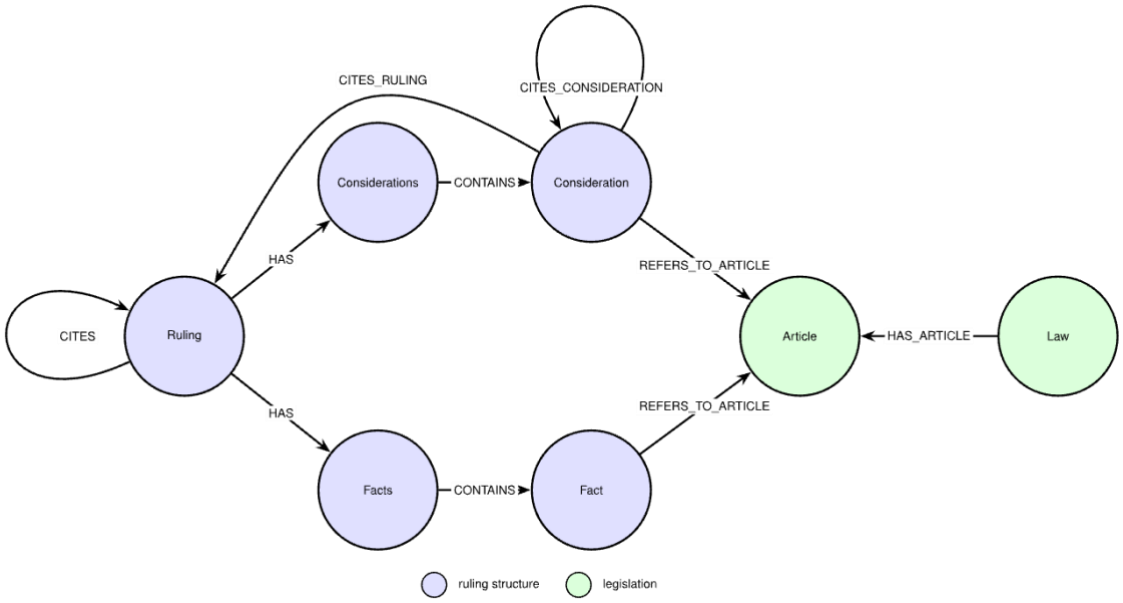

Figure 3: Inherited graph schema

Inherited retrieval strategies: Petrini implemented three retrieval strategies, keyword search, embedding search and embedding-plus-graph search, which this thesis adopts unchanged as Settings A, B and C and specifies in detail in Section 3.6. Across his ten-query evaluation, these improved monotonically (F1 0.58, 0.76, and 0.84) [2], establishing the result this thesis builds on: adding graph structure to embedding retrieval improves performance.

Modifications: Three changes are relevant to the evaluation. First, the decision source was replaced: Petrini scraped HTML from entscheidsuche.ch and reconstructed the structure heuristically, whereas this thesis ingests pre-structured decisions from OpenCaseLaw (Section

3.2). Second, the embedding model was upgraded from text-embedding-3-small to text-embedding-3-large, so the dense vectors underlying Settings B, C and E differ from Petrini's; this affects the comparability of absolute scores between the two theses, but not the within-thesis comparison across settings, which all share the same model. Third and most substantially, the graph schema is extended with a Jurivoc topic layer, and two retrieval strategies (Settings D and E) are added, the contribution of this thesis, described in Sections 3.5 to 3.7. The LLM (GPT-4.1) is unchanged.

3.4 Vocabulary Selection: TERMDAT vs. Jurivoc

Section 2.5 introduced the two candidate vocabularies and argued on conceptual grounds that Jurivoc, the Court's own legal indexing thesaurus, is in principle better suited to a topic layer than TERMDAT, the federal terminology database. This section turns that expectation into a decision grounded in the corpus: the empirical test that confirmed TERMDAT's unsuitability, and the pruning required before Jurivoc could be used. The resulting coverage and assignment statistics are reported in Section 4.1.

Testing TERMDAT against the corpus

To give TERMDAT a fair test, the comparison used not its full set of roughly 400,000 multilingual entries but its German-language, legally classified (RECHT) subset of about 9,600 terms, the portion most plausibly suited to indexing court decisions. Each term was matched against the full text of the ingested decisions by whole-word matching, counting for every term the number of distinct decisions that mention it.

The result strongly indicated that TERMDAT was unsuitable for this task. Of 9,578 terms, 7,014 (73.2 %) matched no decision at all. The 2,564 that did match were dominated by general-language expressions and abbreviations rather than legal concepts. This is a twofold failure for a topic layer: most of the vocabulary is inert against this corpus, and the part that occurs is too generic to discriminate, since a label appearing in tens of thousands of decisions cannot bridge related ones. Even TERMDAT's legal subset thus confirmed what Section 2.5 anticipated: it is a drafting and translation aid, not an instrument for indexing judicial reasoning. We therefore pivoted to Jurivoc.

Pruning Jurivoc for legal use

Jurivoc cannot be ingested whole because it is designed for legal document indexing rather than as a purely legal taxonomy. Hence, it deliberately includes terms from other domains, such as geography and medicine, wherever they help pin down a legal point (Section 2.5). Matched indiscriminately, these would create bridges that are far too broad or legally meaningless. Two complementary filters prune the vocabulary before linking.

The first is structural. Jurivoc groups its non-legal vocabulary under top-level catalogue descriptors, and each such catalogue root together with its directly subordinate descriptors is excluded: the catalogues of countries and cities, animals, personal names, languages, time

references, bodies of water, geographic units, occupations, and the "Switzerland" catalogue, under which the cantons sit.

The second is content based. Each remaining descriptor was classified as legal or non-legal, and the non-legal ones removed. This catches individually unsuitable terms scattered across otherwise-legal branches, such as AIDS, Diabetes, Allergie, Apotheke or Antenne, that the structural filter does not reach. A small hand-curated allowlist overrides the classifier where its judgement was wrong. The structural filter removes the catalogue roots and their direct children; the content filter then removes scattered non-legal descriptors, including any that sit deeper under a catalogue and are therefore not reached structurally.

The topic layer is therefore built on the pruned Jurivoc vocabulary, whose construction and topic assignment are described in Section 3.5, and resulting coverage (answering SQ1) in Section 4.1.

3.5 Topic Assignment

The vocabulary selected and pruned in Section 3.4 supplies the legal concepts, but a concept becomes useful for retrieval only once it is connected to the decisions that discuss it. This section describes how the Jurivoc thesaurus is turned into a layer of the knowledge graph and how that layer is linked to the corpus.

Extended graph schema: On top of the inherited schema (Section 3.3), this thesis adds one new node type and four new edge types, shown in Figure 4:

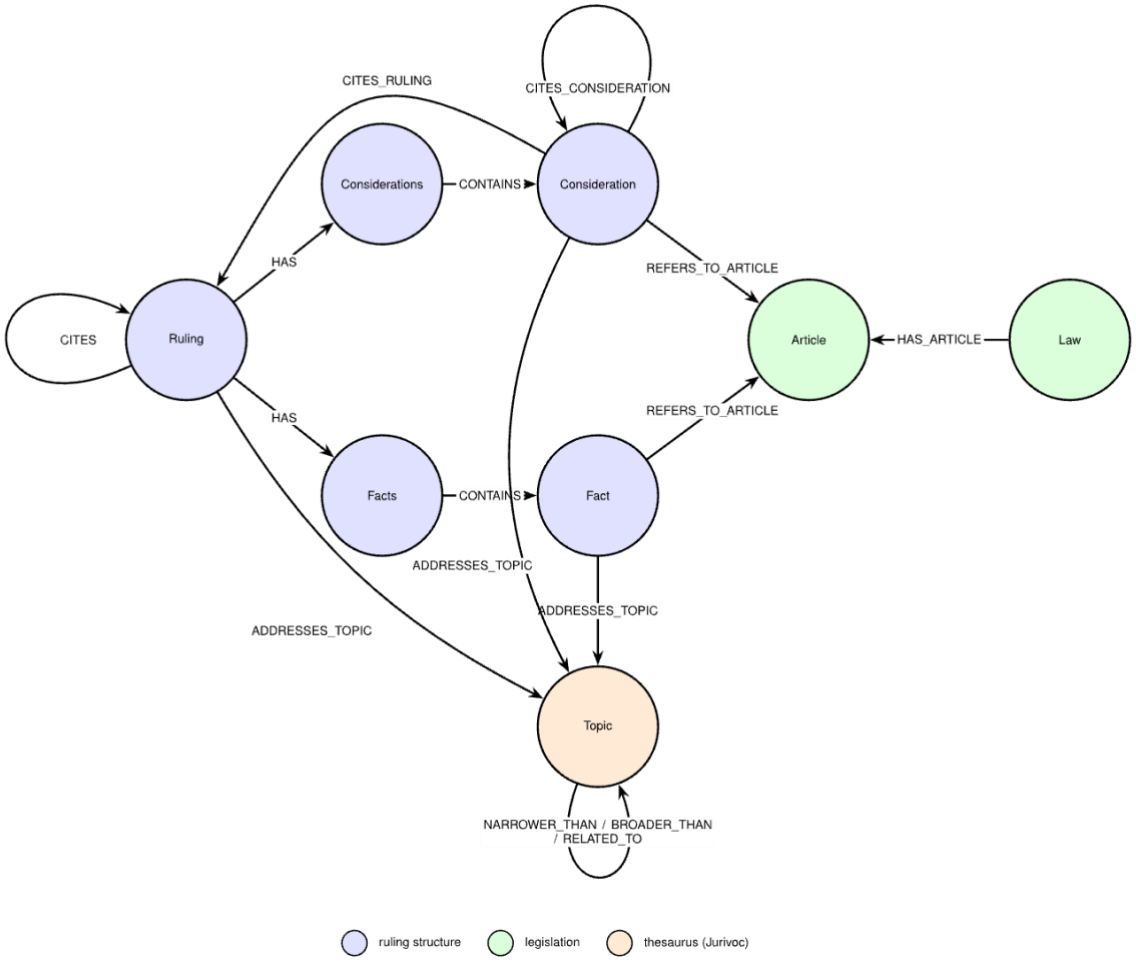

Figure 4: Extended graph schema

Each retained Jurivoc descriptor becomes a topic carrying its German, French and Italian labels and its synonyms as properties. All three language editions are imported, but because the corpus is German only (Section 1.7), in practice only the German labels match. The thesaurus relations are materialised as edges between topics: each narrower-term relation in the dump produces a NARROWER_THAN edge and its reciprocal BROADER_THAN edge, and each see-also relation produces a RELATED_TO edge. The topic layer is connected to the corpus by a single edge type, ADDRESSES_TOPIC, which links a ruling, consideration or fact node to a topic it discusses. The thesaurus edges between topics are created when the Jurivoc dump is loaded; the ADDRESSES_TOPIC edges are created by the assignment procedure described next. The resulting node and edge counts are reported in Section 4.1.

Automatic assignment: The manual Jurivoc labels that the Court's documentation service assigns to leading decisions (Section 2.5) are deliberately not imported into the retrieval graph; importing them would let human-curated assignments leak into the system whose automatic assignment is itself under evaluation, so they are held out as an external reference for the comparison reported in Section 4.1 (SQ1). Every ADDRESSES_TOPIC edge used by the retrieval settings is produced by the automatic procedure below.

The topic-linking pipeline takes the Jurivoc topic labels and produces the ADDRESSES_TOPIC edges that attach each concept to the corpus. It presupposes that the topics already exist in Neo4j, loaded from the Jurivoc dumps, and that the rulings have been ingested by the decision-corpus pipeline of Section 3.2. As shown in Figure 5, it is organised in six stages:

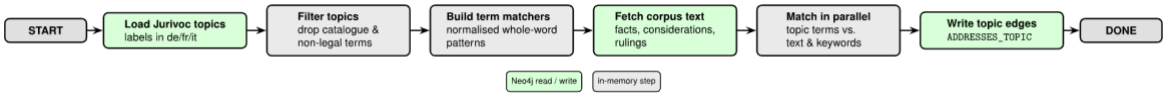

Figure 5: Topic-linking pipeline

1. **Load Jurivoc topics:** the topics and their German, French and Italian labels are read from Neo4j.
2. **Filter topics:** catalogue entries (countries, animals, and similar enumerations) and terms flagged as non-legal are dropped, so that only meaningful legal concepts are linked; this applies, at link time, the pruning motivated in Section 3.4, together with a fixed blocklist of individually noisy terms.
3. **Build term matchers:** each label is normalised (lower-cased, umlauts folded, whitespace collapsed) and compiled into a whole-word, case-insensitive pattern.
4. **Fetch corpus text:** the text and keyword lists of every fact, consideration and ruling to be searched are read from Neo4j; the matched text is the chunk content for facts, the consideration content for considerations and the Regeste for rulings.
5. **Match in parallel:** each topic's pattern is tested against the text and its terms against the keyword lists. A topic whose pattern matches more than 20,000 nodes is treated as too generic and dropped.
6. **Write topic edges:** the ADDRESSES_TOPIC relationships are created in batches, with progress recorded on disk so that the step can resume after an interruption.

The step is purely lexical: it uses neither a language model nor embeddings. A topic is linked to a node when one of its normalised labels occurs as a whole word in the node's text, or when one of its terms exactly equals one of the node's keywords. Two properties matter. First, matching uses preferred descriptor labels only, not the synonyms; the synonyms are stored on each topic but are not used for linking, so synonym-based bridging, where it occurs, happens during query-time topic discovery (Section 3.6), not during corpus assignment. Second, the ADDRESSES_TOPIC edge is binary, with no confidence score; which topics survive is governed by the filters and the 20,000-edge cap.

Assignment runs at all three node levels and depends only on the corpus snapshot and the pruned vocabulary, so it is fully reproducible. Matching surface forms of preferred labels means it inherits the limits of lexical matching discussed in Section 2.1, revisited in the error analysis (Section 4.4) and future work (Section 5.6).

3.6 Retrieval Settings

This section specifies the five retrieval settings compared in this thesis. Settings A, B, and C are conceptually inherited from Petrini and reimplemented as baselines in the present system. Settings D and E are the topic-aware strategies introduced here. All five share the same searcher interface, factory, and infrastructure (Section 3.1), and each is implemented as a LangGraph state machine, a directed graph of processing nodes through which a query flows to a final ranked answer. They are documented here at equal depth, the baselines included, so that a successor extending or replacing any one of them has the full picture of how it is wired, not only of the contribution. Where a strategy loops, the loops are bounded by configurable caps and were held fixed across the evaluation.

3.6.1 Setting A: Keywords (Petrini Baseline)

Setting A runs a graph search anchored on either the law articles or the keywords found in the query, without using embeddings. Candidate considerations are verified by an LLM, expanded with their immediate context, and passed to the LLM with the question to generate an answer. Figure 6 shows the flow.

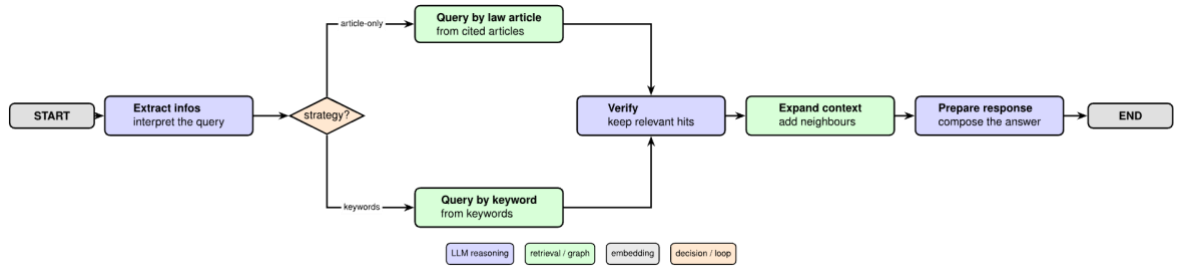

Figure 6: Retrieval flow of Setting A

Extract infos: an LLM distils the question into searchable handles, keywords, referenced laws, article numbers, and chooses a strategy: go straight to the named articles or search by keywords.

Strategy branch: routes to article-only when the user mainly asks for jurisprudence about a specific article.

Query by law article: follows graph edges from the named articles to the considerations that cite them; if nothing is linked, for example a rarely cited law, it falls back to a full-text keyword match so that plain-prose mentions are not missed.

Query by keyword*:* matches the extracted keywords against each consideration's keyword list, a wider, looser net when no provision anchors the search.

Verify: an LLM keeps only the candidates that genuinely address the question, trading the recall of the previous step for precision.

Expand context: pulls in the neighbouring considerations from the same ruling, since a single paragraph rarely carries the full reasoning.

Prepare response: an LLM composes the final answer from the retained considerations, attributing each statement to its source.

3.6.2 Setting B: Embeddings (Petrini Baseline)

Setting B rewrites the query into several sub-queries, embeds them, and retrieves the considerations whose embeddings are closest in meaning, without any graph traversal. The hits are filtered for relevance by an LLM, which then decides whether to answer or to launch another search round with refined sub-queries. Figure 7 shows the flow.

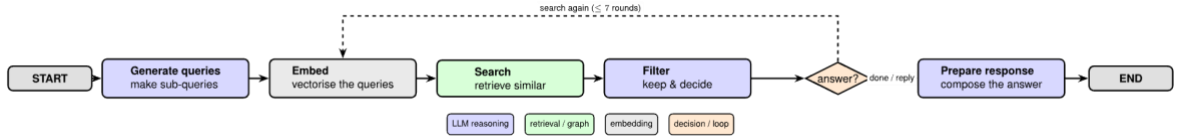

Figure 7: Retrieval flow of Setting B

Generate queries: an LLM rewrites the question into several focused sub-queries, each probing a different facet, since one phrasing rarely captures everything a question is after.

Embed: turns each sub-query into a vector, so passages can be compared by meaning rather than by shared words.

Search: for every sub-query, retrieves the most similar considerations and merges the separate rankings into a single candidate list.

Filter: an LLM keeps the passages that truly answer the question and judges whether the evidence is sufficient, deciding to answer now or to search again with refined queries — removing semantically close but irrelevant hits and steering the loop.

Decision: if the verdict is to search again and the round cap has not been reached, the refined queries are embedded and the loop repeats; otherwise the flow proceeds to the answer.

Prepare response: an LLM composes the grounded, cited answer from everything gathered across the rounds.

3.6.3 Setting C: Embeddings & Citation Graph (Petrini Baseline)

Setting C extends Setting B with citation-graph traversal. After filtering, the LLM may nominate citation neighbours of the retained considerations, which are then fetched and re-evaluated alongside the original hits, letting the search follow the legal links between rulings rather than relying on semantic similarity alone. Figure 8 shows the flow.

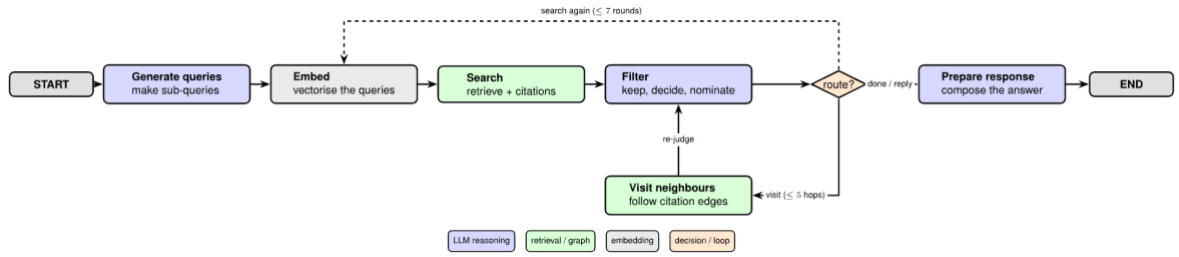

Figure 8: Retrieval flow of Setting C

Generate queries / Embed: as in Setting B, an LLM produces several sub-queries, which are turned into vectors.

Search: retrieves by similarity and, for each hit, additionally records its citation neighbours, the considerations it cites and those that cite it, making those links available to the next step.

Filter: as in Setting B, keeps the relevant passages and decides answer-now versus search-again; additionally, it nominates the citation neighbours most worth pulling in.

Decision: if neighbours were nominated and the hop cap allows, the flow visits them; otherwise, if the verdict is to search again and the round cap has not been reached, it re-embeds new queries; otherwise it answers. A hard cap on the total number of filter passes keeps the nested loops bounded.

Visit neighbours: follows the nominated citation edges to the related considerations, together with their adjacent paragraphs, and hands the result back to the filter to be vetted like any other candidate.

Prepare response: an LLM composes the grounded, cited answer.

3.6.4 Setting D: Topic & Keywords & Graph (This Thesis)

For the two topic-aware settings D and E, ranked lists from separate retrieval lanes are combined by reciprocal rank fusion (RRF). RRF rewards documents that appear near the top of one or more ranked lists and is robust when the underlying scores are not directly comparable [46]. This is important because keyword retrieval, embedding retrieval and topic-based retrieval produce different score types. Fusion is therefore performed on ranks rather than on raw scores.

Setting D runs the keyword-based retrieval of Setting A alongside a parallel topic lane and fuses their results. The topic lane uses Neo4j full-text search over Jurivoc labels and synonyms to identify candidate legal concepts, lets an LLM verify them, and can optionally walk the thesaurus hierarchy to neighbouring topics. The rulings linked to the surviving topics are fused with the Setting A results by reciprocal rank fusion, and the top-ranked considerations are then passed to answer generation in the interactive prototype. Figure 9 shows the flow.

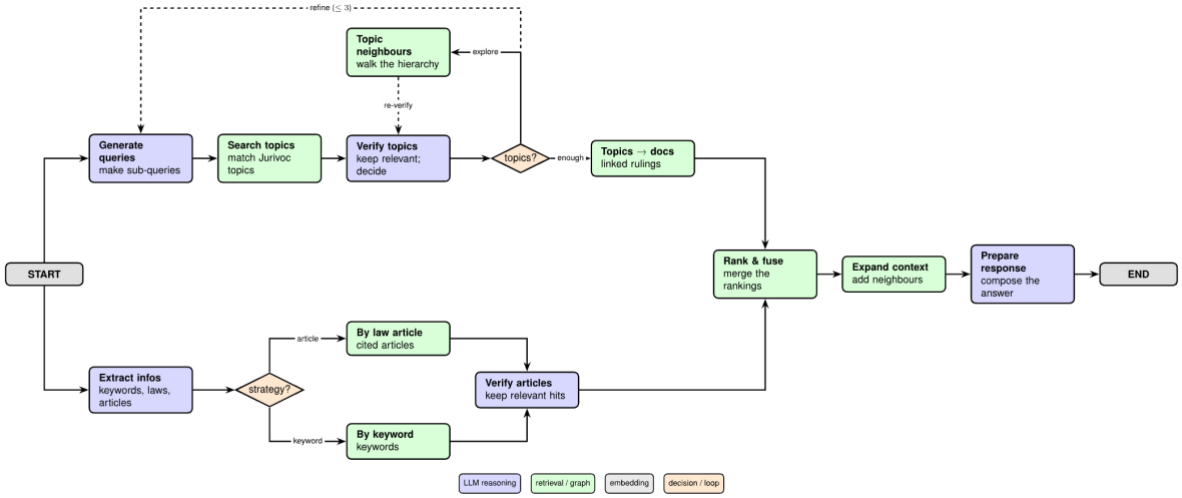

Figure 9: Retrieval flow of Setting D

Topic lane / Generate queries: an LLM expands the question into several sub-queries to match against topic labels.

Search topics: matches those sub-queries against the Jurivoc topic labels and synonyms to find candidate legal concepts. This query-time use of synonyms is where the synonym bridging deferred from corpus assignment (Section 3.5) takes effect.

Verify topics: an LLM keeps the topics that genuinely fit the question and decides what to do next, refine the query, explore neighbours or finish, with the loop bounded by a topic-iteration cap.

Topic neighbours: walks the thesaurus hierarchy (narrower, broader, related) from the kept topics to nearby concepts, which are then re-verified.

Topics → docs: follows the ADDRESSES_TOPIC links from the final topics to the rulings that address them, grounding the search in the curated legal vocabulary rather than in surface wording.

Direct lane: extract infos, query by article or by keyword, and verify, the keyword-and-graph retrieval of Setting A, run in parallel with the topic lane.

Rank and fuse: merges the topic-based and keyword-based rankings into a single ordered list by reciprocal rank fusion, so that conceptual and lexical evidence reinforce each other.

Expand context and prepare response: as in the baselines, the fused list is expanded with neighbouring considerations and an LLM composes the grounded, cited answer.

3.6.5 Setting E: Topic & Embeddings & Graph (This Thesis)

Setting E follows the same topic lane as Setting D, but fuses it with the embedding-based retrieval of Setting C rather than with keyword retrieval. It therefore tests whether Jurivoc topics add value on top of semantic retrieval, where candidate decisions are already selected by dense vector similarity.

Setting E follows the same two-lane structure as Setting D, but differs in two respects. First, topics are matched by embedding similarity rather than by full-text search over Jurivoc labels and synonyms. Second, the parallel retrieval lane uses the embedding-and-citation-graph strategy of Setting C instead of the keyword strategy of Setting A. Figure 10 shows the flow.

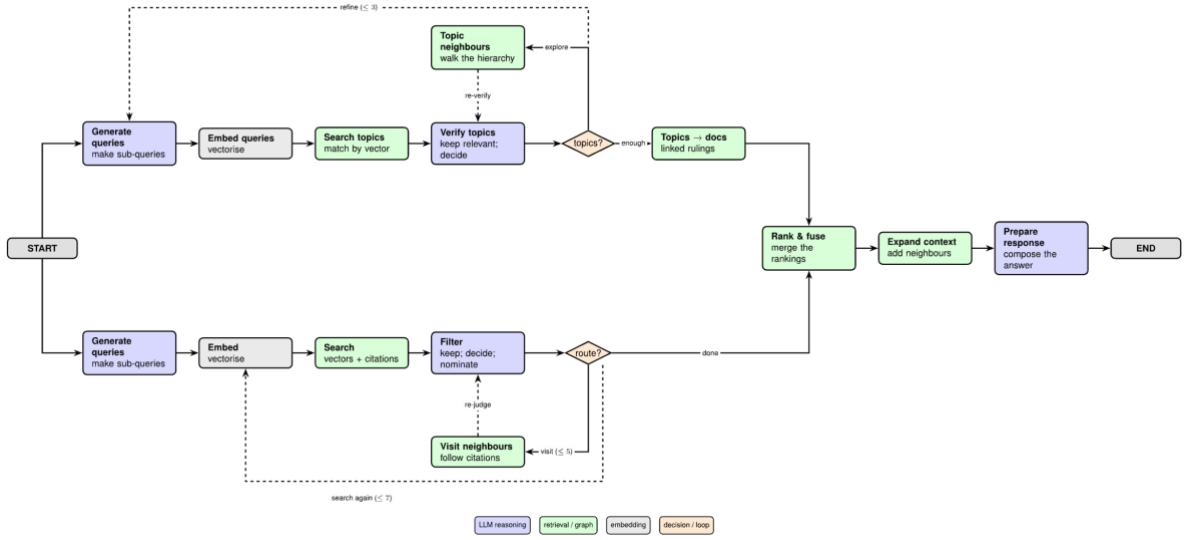

Figure 10: Retrieval flow of Setting E

Topic lane: the same shape as in Setting D, but topics are retrieved by embedding similarity. An LLM expands the question into sub-queries (*generate queries*), which are turned into vectors (*embed queries*) and matched against the Jurivoc topic embeddings (*search topics*). An LLM then *verifies* which topics fit and decides whether to refine, explore neighbours, or finish, with the loop bounded by a topic-iteration cap; the retained topics are followed along ADDRESSES_TOPIC links to the rulings that address them (*topics → docs*).

Vector lane: the embedding-and-citation-graph retrieval of Setting C. Its *search* retrieves by similarity and records citation links, and its *filter* keeps the relevant passages while routing to visit nominated citation neighbours, search again with refined queries, or finish — the nested hop and round loops of Setting C.

Rank and fuse: merges the topic ranking and the vector ranking into a single ordered list by reciprocal rank fusion, so that a ruling reachable by meaning, by topic, and by citation reinforces its own ranking across the three signals.

Expand context and prepare response: as in the baselines, the fused list is expanded with neighbouring considerations and an LLM composes the grounded, cited answer.

3.7 Evaluation Design

To compare the five settings of Section 3.6, we needed a benchmark on which differences should primarily reflect the retrieval strategy rather than changes in the corpus or evaluation setup. This section describes how we built it and the decisions we made; the resulting figures are reported in Chapter 4 and interpreted in Chapter 5. The design serves SQ2, for which query types the topic layer helps, hurts or makes no difference, and supplies the per-query material for the SQ3 error analysis. Because the benchmark is deliberately small (Section 3.7.1), we do not rest our conclusions on aggregate metrics alone but trace what happens in the graph behind them, which is why SQ3 carries as much weight as SQ2.

3.7.1 Gold Standard Construction

The gold standard decisions for a question should be the recognised landmark rulings in that field, not a sample of whatever a textbook happened to cite. We wanted each gold set to contain the decisions a specialist would name as authoritative for the legal point at issue and nothing arbitrary. Building the benchmark on that principle involved four design choices: the first concerns how the questions are written, the other three each shrank the benchmark by ruling out non-German rulings, unpublished rulings and rulings too old or too new to anchor.

Authoring the questions by use case: We made the use cases of Section 1.6 the construction principle of the questions rather than a label applied afterwards, because each exercises a different part of the system. UC3 consists of statute-anchored queries, where statutory-reference and citation signals were expected to provide strong retrieval cues, while the evaluation tests whether topic links still add additional value. A UC2 question is phrased in official doctrinal terminology and should activate the topics that bridge to the right BGE; a UC1 question is colloquial, with no doctrinal anchor. We did not engineer the query vocabulary either to match or to avoid Jurivoc descriptors. The UC2 and UC3 questions were formulated together with legal experts, who wrote with professional terminology and authoritative target decisions in mind; that terminology probably also activates the matching topics, which is the situation in which we expect the topic layer to work. The UC1 questions we wrote ourselves, in the way a non-specialist would phrase the same concern, to test whether everyday phrasing still reaches the right decisions or whether it activates unrelated or too-broad topics. We built the most questions for UC2, fifteen of the 31, against eight each for UC1 and UC3. UC2 receives the largest share of benchmark queries because it is the use case in which the topic layer was expected to provide the clearest benefit. Concentrating the benchmark there lets us verify that expectation and gives a follow-up thesis a documented basis to build on. The 31 items are identified as Q1a/Q1b, Q2a/Q2b and Q3–Q29. Q1 and Q2 each split into two variants of a similar topic. The single queries and the full assignment, with IDs, sources and gold-standard decisions, is given in Appendix 7.1.

Restricting the corpus to German: Because we build the topic layer from the German edition of Jurivoc and link a decision only when a German descriptor matches its German text (Section 3.5), the benchmark could only contain German decisions. The Federal Supreme Court writes each decision in a single language, as a rule that of the cantonal proceedings, and does not translate the full text, only the headnotes (Regesten) of leading decisions exist in all three languages [47]. A non-German decision therefore offers no German full text for our labels to match and its Regeste alone is too thin to index on. The restriction hurt more than we expected once we began assembling gold sets with the legal experts. Many landmark decisions of the Court are French and we repeatedly had to discard otherwise-good query questions because all their landmark decisions were French. Had we been able to admit French and Italian decisions, we could roughly have doubled the number of questions. We accepted the restriction as a consequence of the topic-layer design and note in Section 5.4 that it may bias the benchmark toward concepts well represented in German doctrine.

Counting only published leading decisions as gold: Here we departed from the predecessor. Petrini also admitted unpublished chamber decisions (BGer), which textbooks cite frequently, and following that would roughly have doubled our decisions per question. We restricted gold standard decisions to published leading decisions (BGE), with the exception category described below, because publication is the Court's own marker of which decisions it considers significant, making it a principled population to draw from rather than an arbitrary one, as the legal experts confirmed. The gold standard decisions also had to sit within a usable window: present in the corpus, which the very oldest are not (their format cannot always be parsed, so no usable text exists), yet established enough that commentary points to them, which the most recent are not. We therefore took neither the oldest nor the newest landmark decisions.

Inheriting nine questions and extending them: To tie this thesis to its predecessor, we reused Petrini's doctrinal questions in full, both the questions and their gold standard decisions. Of his ten we carried over nine; the tenth we dropped because its landmark decisions were unusable on our corpus, two too old (from 1931) and one French. Inheriting the gold sets required re-grounding them: we re-checked each cited decision, corrected citations that resolved to a different leading-case page and removed decisions that were French, too old to have parseable text or otherwise absent. This is where the exception category to the BGE-only rule sits: where Petrini cited an unpublished BGer decision present in the corpus, we kept it. Since dropping it would break the comparison on exactly the questions meant to bridge the two studies.

3.7.2 Relevance Model

Relevance is binary: for a question, a retrieved decision is a true positive if its case number is in the gold set and a false positive otherwise; a gold-standard decision we do not retrieve is a false negative. Our gold sets name whole decisions (for example BGE 144 II 65), but the settings retrieve and rank passages, and a single decision can come back as several passages, each with its own score. To check whether a decision was found, we collapse all of its retrieved passages into one entry and score that entry by its best-matching passage. Because the same decision can be written in several ways, we first normalise every citation to one canonical form, so a correctly found decision is not counted as a miss because its number was formatted differently. The model

is deliberately strict: it credits only the landmark decisions for a question and gives nothing for plausibly relevant decisions outside the gold set. The drawback is that a setting surfacing a sound but non-landmark decision is scored as if it had erred, and together with the small gold sets this makes the aggregate metrics blunt.

3.7.3 Metrics

We score each setting on every question with four metrics, all at retrieval depth ten.

- Recall@10 is the share of a question's gold standard decisions in the top ten, against the full gold set; we treat it as central, because completeness is what matters in legal research and a missed leading decision cannot be recovered downstream.
- F1@10 is their harmonic mean.
- nDCG@10 rewards placing relevant decisions high in the ranking, not merely within the top ten, against binary relevance.
- MRR, the reciprocal rank of the first relevant decision, tells us how quickly a user reaches the first useful result.

We report all four per question and as a mean across questions, treating Recall@10 and nDCG@10 as the primary measures. We fixed the depth at ten to reflect a realistic legal-search scenario in which users review only a short-ranked list. In practice users rarely have time to read fifty results and instead need relevant decisions near the top of a short-ranked list. So, what matters is whether a setting surfaces them within the first ten. The depth is held constant across settings.

3.7.4 Execution Protocol

A run issues one request per setting and question to the backend, choosing the setting by name so all five go through the same endpoint and configuration, and asking for ten results. We score the returned ranking against the gold set, write out the per-query scores and aggregate them to one row per setting. We hold every tunable parameter of Section 3.6 at a fixed default across all settings and questions, so the only thing varying across the five rows is the retrieval strategy. Two defaults bound how far the topic layer reaches: we traverse the topic graph one hop from a matched concept, and we fuse the topic lane with its parallel lane by reciprocal rank fusion at its standard setting (Section 3.6). One adjustment we deliberately did not make is to favour leading decisions in the ranking. Since only BGE count as gold, weighting the system toward published leading decisions over chamber decisions would have raised every metric, but it would have measured our scoring bias rather than the retrieval strategies, so we left the ranking neutral as to decision type across all five settings.

The settings are not deterministic. Settings B through E rewrite the question into sub-queries with a language model and the topic settings let it decide which concepts and citation neighbours to pursue, so the same question can return slightly different rankings across runs. We handle this by repeating each question 10 times and averaging its per-query metrics, so a reported figure reflects typical behaviour rather than one sample. Two engineering constraints shaped the rest:

the graph-aware settings can take over a minute per question, so we use a long timeout and retry with backoff before recording a question as failed, and we can re-run a failed question in isolation and merge it back rather than repeating the whole matrix.

We also attempted to tune the model against the benchmark rather than run it only at fixed defaults. The query set proved too small: the tuned configurations overfitted to the few questions they were optimised on and did not generalise, so we abandoned tuning and report the system at its fixed defaults. This is a further consequence of the benchmark's size and a direct argument for the larger query set proposed as future work.

4 Results

This chapter reports results for the five retrieval settings defined in Section 3.6. It describes the performance patterns and leaves the broader interpretation, comparison with Petrini's predecessor system, answers to the research questions and limitations to Chapter 5.

Section 4.1 reports the size, coverage and manual-indexing alignment of the Jurivoc topic layer. Section 4.2 presents retrieval performance across the five settings, first in aggregate and then by use case category. Section 4.3 examines representative per-query cases in which topic-aware retrieval improves performance or the inherited baselines remain stronger. Section 4.4 summarises the empirical findings.

Retrieval figures are reported at retrieval depth ten and averaged over repeated runs per query, as described in Section 3.7.4. The evaluation uses the binary gold standard of Section 3.7.2 and holds system parameters fixed, so that differences reflect retrieval strategy rather than corpus or setup.

In aggregate, the five settings follow one order across all reported metrics: keywords (A), embeddings (B), embeddings with citation-graph traversal (C), topic-and-keyword retrieval (D) and topic-and-embedding retrieval (E).

4.1 Jurivoc Topic Assignment Statistics

This section reports the size, coverage and manual-indexing alignment of the Jurivoc topic layer described in Section 3.5. The construction of the layer and the lexical assignment procedure are not repeated here; only the resulting statistics are reported.

After the pruning step described in Section 3.4, the topic layer contains 10,975 topics, of which 8’148 are matching descriptor topics. Most of these are legal descriptors that can be matched against decision text; the remainder are non-descriptor terms that exist only as targets of thesaurus relations. The assignment procedure produced 3,600,344 ADDRESSES_TOPIC edges over the corpus, of which 2,658,490 link topics to consideration nodes, 685,132 to fact nodes and 256,722 to ruling nodes.

Almost every decision in the corpus receives at least one topic (cf. Figure 11). Across the full corpus of 133,005 rulings, the number of topics per decision is right-skewed: the median is 13 and the mean 15.8, with the fifth and ninety-fifth percentiles at 3 and 35 and a maximum of 229. The topic layer therefore assigns multiple conceptual access points to most decisions rather than a single label.

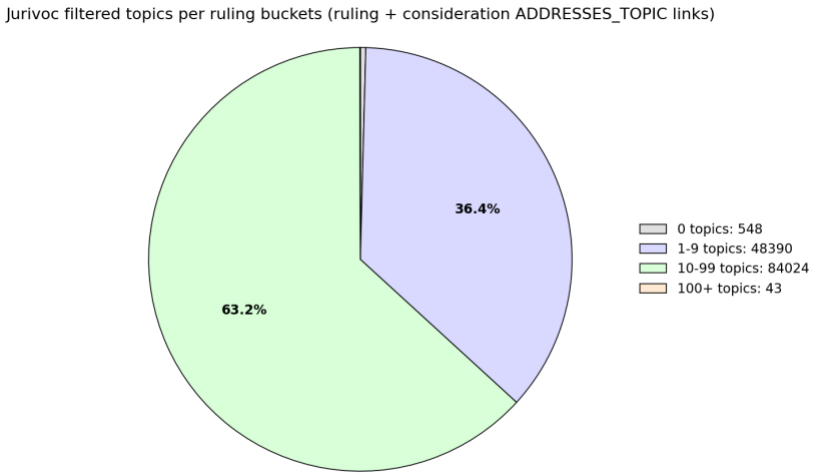

Figure 11: Distribution of the 133,005 rulings by number of assigned topics

The most heavily linked descriptors are broad procedural, institutional or substantive terms, including NEUBEURTEILUNG and ZUSTÄNDIGKEIT (procedural), REGIERUNGSRAT (institutional) and ZINS (substantive). The generic-term cap described in Section 3.5 removes the most frequent generic terms, but the resulting distribution still contains a high-frequency tail of broad descriptors, whose edges are unlikely to discriminate between relevant and irrelevant decisions.

The automatic assignment was also compared with the Federal Supreme Court's manual Jurivoc indexing, which was held out as an external reference. For each descriptor, the evaluation measured how many of the decisions manually indexed under that descriptor were also reached by an automatic ADDRESSES_TOPIC edge. Across the 4,462 descriptors with at least one manual entry, the automatic layer covers 32,467 of 91,676 manual decision-topic assignments. This corresponds to 35.4% coverage by volume (cf. Figure 12). The per-descriptor mean is 41.0% and the median is 39.4%.

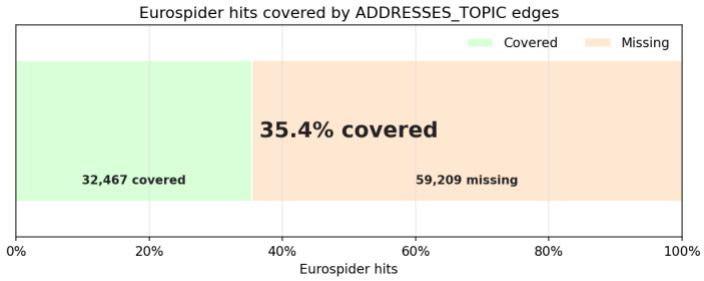

Figure 12: Coverage of the manual Jurivoc indexing by the automatic assignment

These statistics establish two properties of the topic layer. First, it has dense corpus coverage, because almost every decision receives at least one topic. Second, its alignment with manual Jurivoc indexing is partial, because the lexical assignment recovers between one third and two fifths of the Court’s manual descriptor-decision assignments. This means that the automatic topic layer should be interpreted as a broad retrieval signal rather than as a reproduction of the Court’s manual Jurivoc indexing.

4.2 Retrieval Performance

This section reports retrieval performance for the five settings, first in aggregate across all 31 queries (Section 4.2.1) and then by use case category (Section 4.2.2).

4.2.1 Aggregate Metrics across Five Settings

Figure 13 reports four metrics of Section 3.7: Recall@10, nDCG@10, MRR and F1@10, each averaged over the 31 queries and repeated runs (Section 3.7.4).

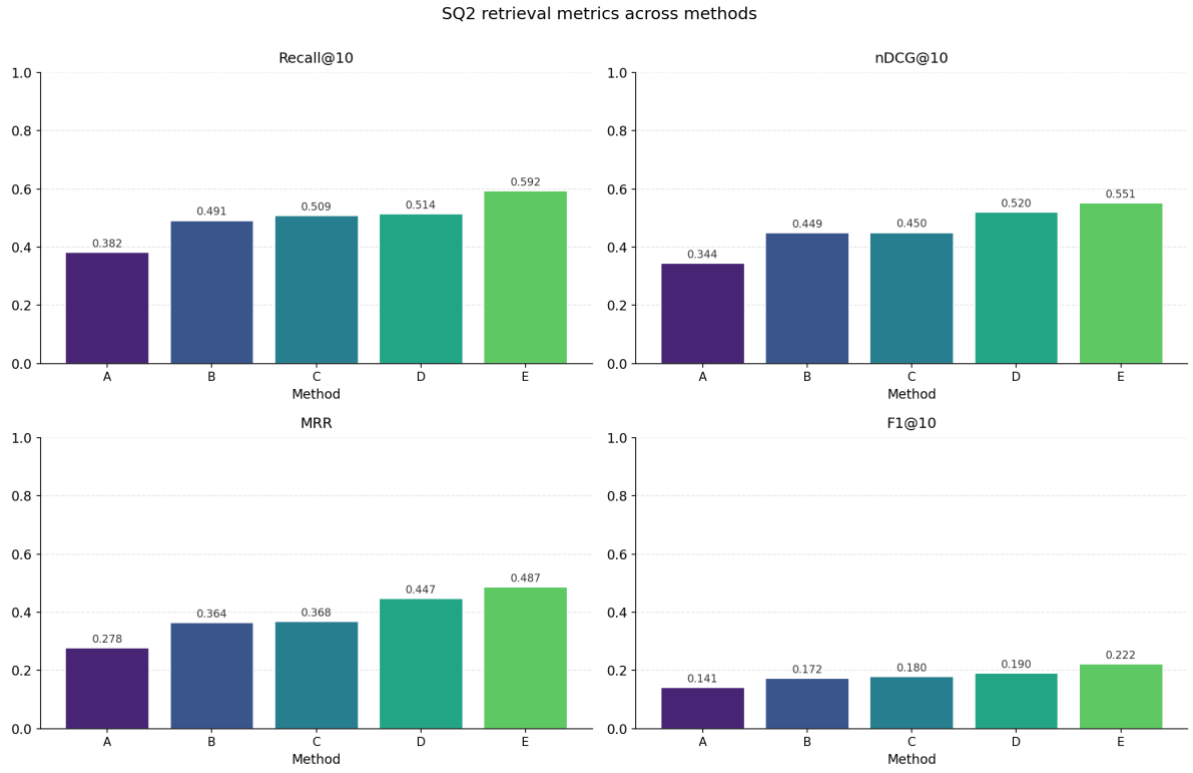

Figure 13: Aggregate retrieval metrics across the five settings

The five settings rank in one order on every metric: A below B, B below C, C below D, D below E. The two topic-aware settings (D and E) therefore outperform all three inherited baselines on every measure, and Setting E is strongest throughout, reaching Recall@10 0.592 against 0.509 for the best baseline C, and nDCG@10 0.551 against 0.450, with relative gains of 16% and 22% over C.

The separation is clearest on the rank-sensitive metrics. From C to E, Recall@10 rises by 0.083 (0.509 to 0.514 to 0.592), while nDCG@10 rises by 0.101 (0.450 to 0.520 to 0.551) and MRR by 0.119 (0.368 to 0.447 to 0.487). The topic layer thus improves where relevant decisions are placed at least as much as how many appear in the top ten.

The absolute F1@10 values are low across all settings, but this reflects the metrics' ceiling under small gold sets rather than weak retrieval. With a fixed denominator of ten (Section 3.7.3) and gold sets of one to six decisions per query, mean F1@10 cannot exceed about 0.41 even under

perfect retrieval; Setting E's 0.222 is roughly half that achievable maximum and Setting A's 0.141 about a third. Both metrics preserve the A-to-E ordering.

Within the inherited baselines, performance increases monotonically from A to B to C, so adding citation-graph traversal to embedding retrieval (C over B) reduces no aggregate metric on this benchmark.

These aggregate means conceal substantial per-query variation (Figure 14). The topic-aware settings recover decisions all three baselines miss (for example Q17, where A, B and C retrieve none of the gold decisions while D and E do) and fall behind on others (for example Q24 and Q28, where a baseline retrieves more). Two counts capture the balance: Setting E retrieves at least as many gold decisions as the best baseline on 18 of the 31 queries, and the better topic setting strictly exceeds the best baseline on 16. The number of queries on which a setting retrieves no gold decision at all falls along the same ordering: 11 for A, 5 for B, 4 for C, 3 for D and 1 for E. The topic layer therefore raises the averages and reduces complete retrieval failures.

Section 4.3 examines selected mechanisms behind the per-query variation. Their broader interpretation follows in Chapter 5.

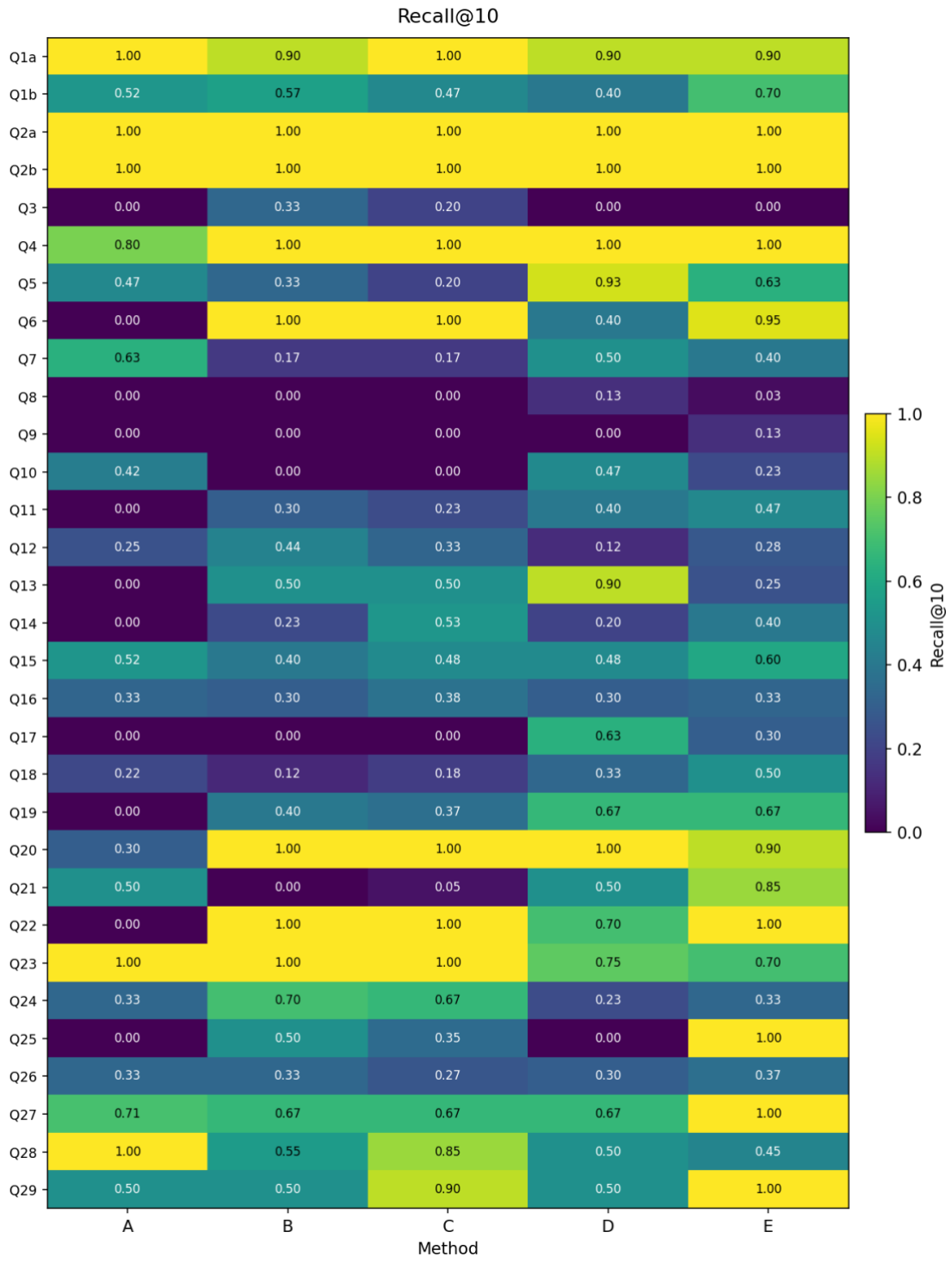

Figure 14: Per query recall heatmap

4.2.2 Results by Use Case Category

The 31 queries are grouped into the three use cases of Section 1.6: self-written colloquial concept search (UC1, 8 queries), doctrinal legal search (UC2, 15 queries) and statute-anchored search (UC3, 8 queries). The grouping reflects each query's dominant register rather than a strict

partition, so the group boundaries are approximate. Figure 15 shows Recall@10, nDCG@10, MRR and F1@10 by use case and setting.

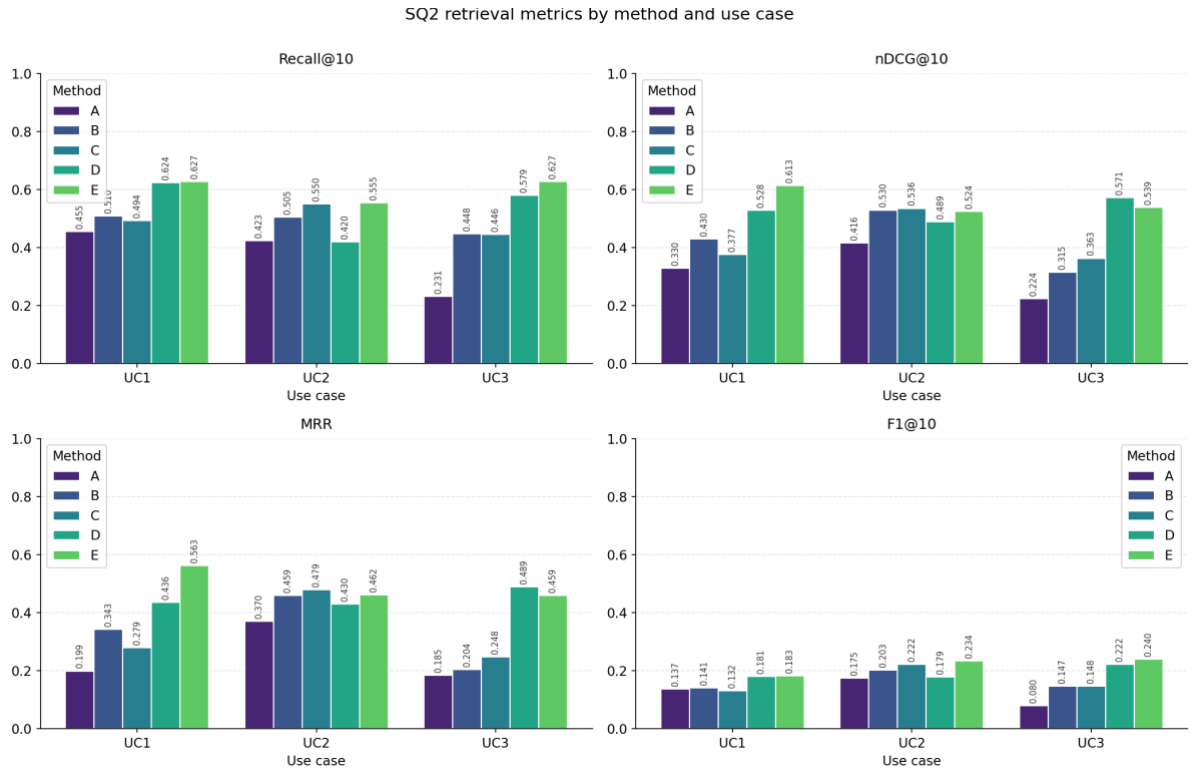

Figure 15: Metrics by use case and setting

Setting E achieves the highest Recall@10 in all three groups, but the size and pattern of the topic-aware advantage differs by use case.

On **UC3** the topic-aware settings show their largest advantage: both D and E exceed every baseline on every metric. E reaches Recall@10 0.627 against 0.448 for the best baseline B, and D and E reach nDCG@10 0.571 and 0.539 against 0.363 for C. Within the group, D and E divide the rank-sensitive metrics from the coverage metrics: D attains the higher nDCG@10 (0.571 against 0.539) and MRR (0.489 against 0.459), while E attains the higher Recall@10 (0.627 against 0.579) and F1@10 (0.240 against 0.222). D therefore places an early relevant decision higher on average, while E retrieves more relevant decisions within the top ten.

On **UC1** both topic-aware settings lead every baseline on every metric. E reaches Recall@10 0.627 and nDCG@10 0.613 against 0.510 and 0.430 for the best baseline B, and as in the aggregate the advantage on the rank-sensitive metrics (nDCG@10, MRR) is wider than on recall.

On **UC2** the pattern is different. Setting E leads on Recall@10 (0.555) and F1@10 (0.234), but both embedding baselines attain a higher nDCG@10 (C 0.536 and B 0.530 against E's 0.524) and C also attains a higher MRR (0.479 against 0.462). Setting D is weaker than both embedding baselines on recall: its Recall@10 of 0.420 is the lowest of all five settings, below B (0.505) and C (0.550). UC2 is therefore the only group in which a baseline leads Setting E on any metric, and the only one in which a topic-aware setting falls behind the baselines on recall.

The topic-aware advantage on Recall@10 is largest on UC3 (E exceeds the best baseline by 0.179) and UC1 (by 0.117), and smallest on UC2 (by 0.005). This ordering runs counter to the directional hypothesis of Section 1.6, which anticipated the largest topic benefit on the doctrinal queries (UC2). Why the pattern inverts that expectation is taken up in Section 5.2.

4.3 Per-Query Outcome Analysis

The aggregate and per-use-case means of Section 4.2 average over substantial per-query variation. This section examines that variation directly, first by classifying all 31 queries by outcome, then tracing the mechanisms behind representative cases. All counts below are hits among the ten runs, scored against the in-corpus gold sets of Section 3.7.

Each query is assigned to one of three outcomes by comparing the best topic-aware setting (D or E) against the best baseline (A, B or C) on Recall@10. The topic-aware settings retrieve strictly more gold decisions than any baseline on 16 of the 31 queries, a baseline retrieves more on 10, and the two tie on 5 (Table 5).

Outcome pattern	n
Topic-aware setting retrieves more gold decisions than any baseline	16
A baseline retrieves more than either topic setting	10
Topic-aware and baseline tie	5

Table 5: Classification of the 31 queries by retrieval outcome

Table 5 gives the distribution. The cases below trace the mechanism behind each non-trivial outcome: a topic-win where the layer recovers decisions the baselines miss (Q17), the topic-lane split that drives the use-case differences of Section 4.2.2 (Q13), and two baseline-wins of different kinds, one where strong explicit signals already retrieve the gold decisions (Q28) and one where the topic layer cannot reach a gold decision at all (Q3, Q24). Ties, where both approaches retrieve equally, add no mechanism and are not traced separately. Each case is a representative instance.

The topic layer recovers decisions the baselines miss (Q17): Query 17 concerns *in dubio pro reo* as an expression of the presumption of innocence. None of the three baselines retrieves any of the three gold decisions (BGE 127 I 38, BGE 144 IV 345, BGE 120 Ia 31) in any run (Recall@10 0.000 for A, B and C). Setting D retrieves BGE 120 Ia 31 in all ten runs, BGE 127 I 38 in five and BGE 144 IV 345 in four (Recall@10 0.633); Setting E retrieves BGE 120 Ia 31 in five runs, BGE 144 IV 345 in three and BGE 127 I 38 in one (Recall@10 0.300). The topic lane reaches these decisions where lexical and semantic matching find nothing (Figure 16). The same recovery appears on Q21, where only Setting E retrieves BGE 127 I 164, the gold decision the baselines miss, with any regularity, in eight of ten runs against at most one for the baselines.

Figure 16: Returned documents for query 17

The fused ranking determines which topic lane leads (Q13): Setting E leads UC3 on Recall@10 while Setting D leads on the rank-sensitive metrics, and Query 13 (private-life grounds for a residence permit) shows why. Setting D retrieves both gold decisions in nine of ten runs each and ranks them high (Recall@10 0.900, nDCG@10 0.695); Setting E retrieves them in only three and

two runs and ranks them low (Recall@10 0.250, nDCG@10 0.200), cf. Figure 17. Query 17 shows the same direction (D nDCG@10 0.611 against E 0.272). Section 5.1 explains why, in terms of how the two settings' rankings combine, and shows the same mechanism behind the divergences on Q25 and Q1b.

Figure 17: Returned documents for query 13

A baseline remains stronger where explicit signals already retrieve the gold decisions (Q28): Query 28 concerns the deference the Federal Supreme Court shows to the Communications Commission. The keyword baseline retrieves both gold decisions (BGE 131 II 13, BGE 132 II 257)

in all ten runs (Recall@10 1.000) and the embedding-plus-citation baseline in ten and seven runs (Recall@10 0.850), while the topic settings are weaker (D 0.500, E 0.450). The keyword and citation baselines retrieve these decisions directly through the ruling text; Section 5.1 explains why the topic settings rank them lower, in terms of how reciprocal-rank fusion combines the topic and parallel lanes (Figure 18).

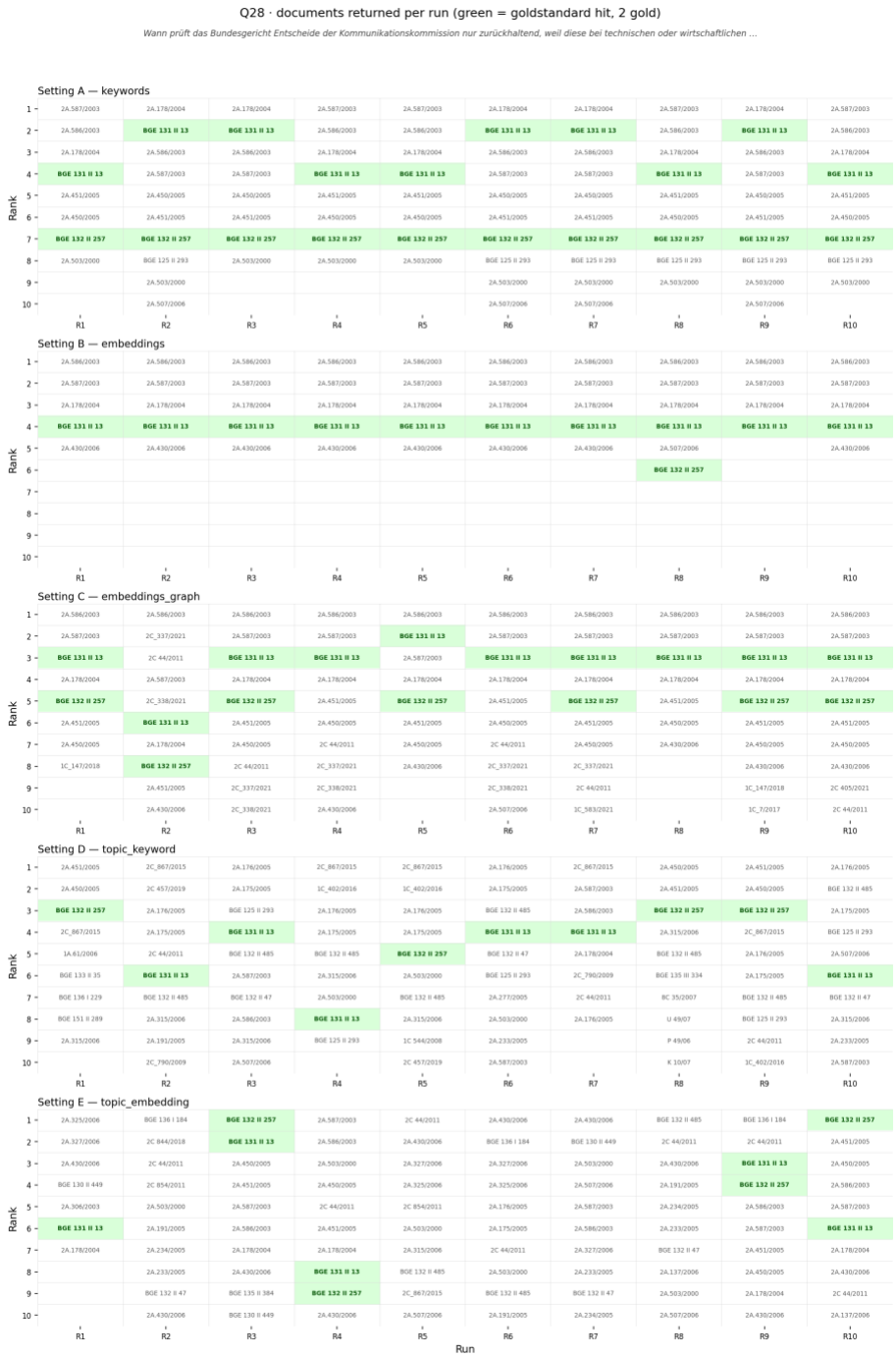

Figure 18: Returned documents for query 28

A baseline remains stronger where the topic layer cannot reach the gold decisions (Q3, Q24): On some doctrinal queries the gold decisions carry no descriptors that the query's

concepts match, so the topic lane contributes nothing and the topic settings fall back on their parallel lane alone. On Q3 only the embedding baselines retrieve a gold decision at all (Recall@10 B 0.333, C 0.200), while the keyword baseline and both topic settings retrieve none. On Q24 the embedding baselines again lead (Recall@10 B 0.700, C 0.667) over the topic settings (Recall@10 D 0.233, E 0.333), because the gold decisions are reachable by semantic similarity but not from the query's topics. These queries lie behind the doctrinal group's weak showing for the topic settings in Section 4.2.2, and Section 5.1 takes up why the topic path fails to match them.

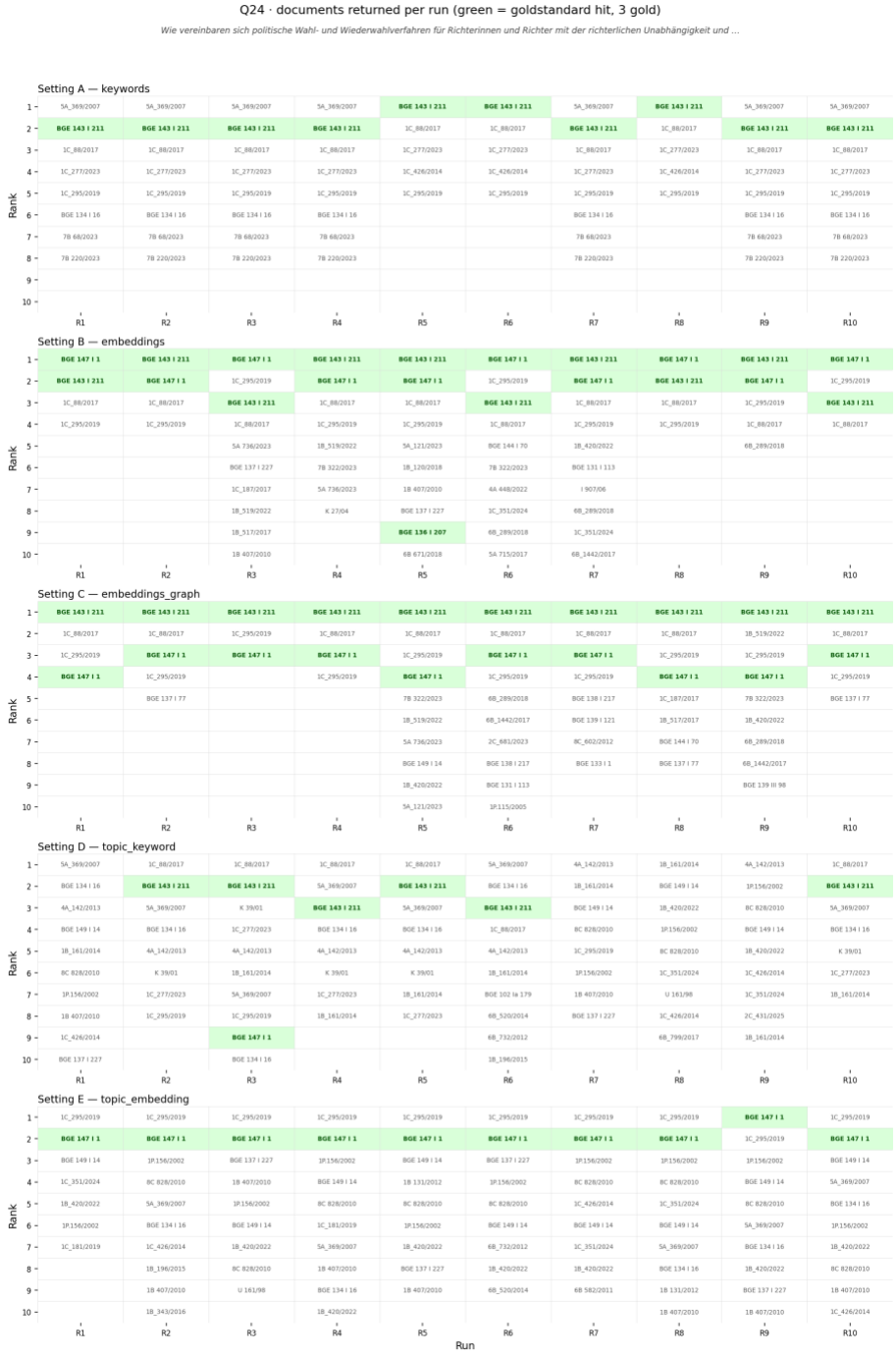

Figure 19: Returned documents for query 24

4.4 Summary of Empirical Findings

This chapter supports four findings, stated here without interpretation; Chapter 5 explains them and relates them to the research questions.

The topic layer is dense but only partially aligned with human indexing. Almost every decision in the corpus receives at least one topic, yet the automatic assignment reproduces only about a third to two-fifths of the descriptor–decision pairs the Court assigned by hand (Section 4.1). It reaches the whole corpus, but it is not a substitute for the manual index.

On this benchmark, the topic-aware settings improve retrieval, and Setting E improves it most. Both topic-aware settings outperform all three inherited baselines on every aggregate metric, ordered A < B < C < D < E throughout, with Setting E (topic + embedding) strongest on each (Section 4.2.1). The gain is larger on the rank-sensitive metrics than on recall: the topic layer changes where relevant decisions are placed more than how many are found.

The benefit is conditional, not uniform. It is largest on the statute-anchored and colloquial queries and smallest on the doctrinal ones. On the doctrinal queries only the embedding-based Setting E still leads, and only on recall by the narrowest margin in the study, while the keyword-based Setting D falls behind the embedding baselines (Section 4.2.2). This is the reverse of the directional hypothesis of Section 1.6, which expected the doctrinal queries to benefit most.

It fails in consistent ways. The topic layer is outperformed where explicit statutory and citation signals already retrieve the gold decisions, where the query's concepts match none of a relevant decision's descriptors so the topic lane cannot reach it, and through its weaker keyword-based lane relative to its embedding-based one (Section 4.3). On 16 of the 31 queries a topic-aware setting retrieves more gold decisions than any baseline; on 10 a baseline retrieves more.

The picture is therefore of a topic layer that helps where lexical and citation signals are weak, defers to them where they are strong, and depends on the path it is combined with. Chapter 5 interprets this, sets it against the predecessor system and weighs it against the limitations of the evaluation.

5 Discussion and Outlook

This chapter interprets the results of Chapter 4, answers the research questions, situates the findings against the predecessor work, and sets out the limitations and future work. Each claim is grounded in the data of Chapter 4, the design of Chapter 3 or the literature of Chapter 2; where an explanation goes beyond what the data strictly establish, it is marked as interpretation.

5.1 Interpretation of the Results

Why the topic layer helps: Its advantage (Section 4.2.1) comes from the one relation the baselines cannot represent: a link between decisions that are conceptually related but neither share query vocabulary nor cite each other (Sections 1.2, 2.4). This is the gap citation-graph retrieval, the strongest baseline, structurally cannot close, since a citation edge exists only where one decision references another. The per-query traces show the layer working in two ways. On some queries it adds gold decisions the baselines miss entirely, including ones the embedding lane cannot reach at all, which the topic and keyword signals recover instead (Q5, Q10), and on others it lifts decisions the semantic lane already retrieves into higher ranks. In aggregate the second effect dominates: the topic-aware settings gain more on nDCG@10 and MRR than on Recall@10 (Section 4.2.1). Therefore, the layer acts more as a reranking signal than a recall signal.

Why partial coverage still helps: The automatic layer reproduces only a third to two-fifths of the Court's manual indexing (Section 4.1), yet the topic settings still improve on the baselines. Hence, even this partial assignment captures enough legally relevant links to matter, and the descriptors that are matched are specific enough to discriminate where the general-purpose TERMDAT vocabulary of Section 3.4 did not. Coverage nonetheless bounds the contribution: a denser or more precise assignment could raise the ceiling the present figures set.

Why the two topic lanes diverge: Settings D and E both fuse a topic lane with a second, parallel lane by reciprocal-rank fusion, and differ in only two places (Section 3.6): E discovers topics by embedding similarity where D matches topic labels lexically, and E's parallel lane is the embedding-and-citation-graph retrieval of Setting C where D's is the keyword retrieval of Setting A. Which setting ranks the gold decisions higher turns on what that parallel lane contributes. On the queries where the gold decisions sit only on the topic lane, the parallel lane hurts E: the gold decisions are reached on the topic lane but not on E's embedding lane, so once fused they fall below on-domain decisions that appear on both lanes, while D's keyword lane returns little and lets the topic lane dominate (the pattern behind Q13 and Q17 in Section 4.3). The same asymmetry explains why D is the weakest topic setting on the doctrinal group (Section 4.2.2): where the topic lane finds the gold decisions but ranks them low, only a parallel lane that also finds them can lift them, and D's keyword lane frequently cannot, while E's embedding lane can (Q1b). That D's keyword lane is so often empty on doctrinal queries follows from its query router, which goes to the named articles only when the query pins down specific provisions and searches by keyword otherwise (Section 3.6.1), a condition the benchmark's doctrinal phrasing rarely meets. The same router behaviour explains the keyword baseline's own weakness on these queries, where Setting A is the weakest baseline (Section 4.2.2): its keyword search returns little when no specific provision is named.

Why the topic layer also adds noise: The mechanism that connects distant-but-relevant decisions can equally connect adjacent-but-irrelevant ones. Four mechanisms cause the topic settings to lose ground (two visible directly in the traces, two the design makes plausible but the present data cannot isolate), each consistent with Chapter 4.

The first is *chunk-level fusion splitting the evidence*. Where a gold decision is retrieved by the baseline lane through its ruling text while the topic lane matches the same decision through its Regeste, reciprocal-rank fusion scores the two chunks separately rather than crediting the decision once, so neither chunk accumulates enough weight to rank highly; the decision rises only in the runs where the Regeste is also reached through the parallel lane. This is why the Q28, which the keyword and citation baselines retrieve directly, is ranked lower by the topic settings (Section 4.3).

The second is *the topic path failing to match the decision at all*. Topic discovery only reaches a decision whose descriptors can be matched from the query's concepts; where they cannot, the topic lane contributes nothing and the topic settings fall back on their parallel lane alone. On Q3 and Q24 the query's concepts do not map to the gold decisions' descriptors, so both topic settings trail the embedding baselines, which reach the decisions by semantic similarity instead.

The third is *granularity mismatch*. The descriptor distribution of Section 4.1 is heavily skewed: a minority of descriptors are broad, high-frequency terms linking to thousands of decisions. A query activating such a descriptor inherits a large, doctrinally undifferentiated neighbourhood, and the lane returns many on-domain but non-landmark decisions that crowd out the few the gold set credits. This is the expected signature of the broad descriptors identified in Section 4.1.

The fourth is *traversal drift*, which the data cannot confirm but the design makes plausible. Topic matching is lexical (Section 3.5), so a query term can match a descriptor it shares a label with but whose legal sense differs. This is the polysemy problem of Section 2.1, where a term such as *Sicherheit* denotes unrelated doctrines across areas of law. A single mismatched match, expanded one hop along the thesaurus edges, could route the search into a neighbouring doctrine that shares the label but not the legal question. The depth-one and generic-term caps (Section 3.5) exist to contain this, and the absence of catastrophic off-domain failures in Chapter 4 suggests they largely succeed; isolating whether residual noise enters by this route rather than by one of the first three would require further trace analysis.

The overall reading: The topic layer is most valuable where a query under-specifies the doctrine or where relevant decisions are linked conceptually but not lexically or by citation, and least valuable, or counterproductive, where explicit signals already identify the gold decisions or the topic path cannot reach them.

5.2 Answering the Research Questions

RQ: To what extent does integrating Jurivoc-based topics into a graph-enhanced retrieval system improve retrieval performance, compared with the keyword, embedding and embedding-plus-citation-graph baselines? On this benchmark, it improves retrieval on every

aggregate metric. Both topic-aware settings exceed all three baselines, and the strongest, topic-and-embedding (E), raises Recall@10 from 0.509 to 0.592 and nDCG@10 from 0.450 to 0.551 over the best baseline (Section 4.2.1). The improvement is real but conditional: it holds in aggregate on every metric, yet its size depends on query type (SQ2) and it reverses on individual queries (SQ3).

SQ1: How suitable is Jurivoc as a controlled legal vocabulary for constructing topics over Swiss Federal Supreme Court decisions? Suitable, with one qualification. Structurally it yields a usable concept layer of 10,975 topic nodes carrying broader-, narrower- and related-term relations after pruning (Sections 3.4, 4.1). Against the Court's manual indexing, the automatic assignment recovers a third to two-fifths of the descriptor–decision pairs by a purely lexical procedure (Section 4.1). The qualification is that this figure is recall against the manual labels in one direction only: it does not assess the precision of the automatic assignment, so suitability is established for coverage but not for full agreement with human indexing.

SQ2: For which query types do topic-aware strategies improve, degrade or leave unchanged retrieval performance compared with the baselines? The topic layer helps most on the statute-anchored group (UC3) and the colloquial group (UC1), and least on the doctrinal group (UC2), where only Setting E still leads, by the narrowest margin in the study, while Setting D falls behind the embedding baselines and so degrades retrieval (Section 4.2.2). This inverts the directional hypothesis of Section 1.6. The group we expected to gain most (UC2) gains least, and the statute-anchored group (UC3), posed there as a control in which topics were not expected to add substantive benefit, shows the largest gain instead. Section 5.1 explains the inversion in terms of where the baselines are weakest. Given the approximate use-case boundaries, this is a direction rather than a precise ranking.

SQ3: What error patterns explain cases where topic-aware retrieval underperforms, and what design choices could mitigate them? The four failure mechanisms of Section 5.1 each suggest a mitigation. Chunk-level fusion splitting a decision's evidence points to fusing at the level of the decision rather than the chunk. The topic path failing to match a decision is addressed by the embedding-based topic assignment of Section 5.6, which finds concepts without exact label matching. Granularity mismatch points to suppressing or splitting over-broad descriptors beyond the existing cap. Traversal drift, if it could be isolated from the present data, would call for sense-disambiguated matching. The weaker keyword-based lane of Setting D is, separately, not a matter of mitigation but of configuration: Setting E already replaces that lane with embedding-and-citation retrieval, which is why E is the stronger of the two topic settings.

5.3 Comparison with Petrini's Baseline

Absolute scores are not comparable: Petrini evaluated three settings on ten doctrinal queries from legal textbooks and reported F1 values of 0.58 (keyword), 0.76 (embedding) and 0.84 (embedding-plus-graph). The present benchmark differs in nearly every dimension that affects absolute values: 31 queries across three registers rather than ten doctrinal ones; a gold standard restricted to published leading decisions with a single inherited exception (Section 3.7.1); a corpus changed to the pre-structured OpenCaseLaw dataset (Section 3.2); and an embedding

model upgraded to text-embedding-3-large (Section 3.3). All of them shift the metric values, so the two studies' figures cannot be compared.

The within-thesis comparison is the valid one: Because all five settings share one corpus, model, gold standard and protocol (Section 3.1), differences between them are attributable to the retrieval strategy, and within that frame the ordering A < B < C < D < E holds on every metric.

A divergence worth noting: In Petrini's evaluation the embedding-plus-graph setting was strongest of the three; here the same setting (C) remains strongest of the three baselines, so the finding that citation-graph traversal helps is reproduced. What changes is that two topic-aware settings now sit above it. This extends rather than contradicts the predecessor result: the two kinds of edge serve different situations. Citation traversal connects decisions that reference one another, concept traversal connects those that share a doctrine without citing each other. The per-use-case results bear this out, with C relatively strongest on the doctrinal group where citation structure is densest, while the topic layer gains most on the colloquial and statute-anchored groups Petrini's doctrinal-only set did not isolate (Section 4.2.2).

5.4 Limitations and Validity Threats

The limitations are grouped by the part of the study they affect; the scope decisions were anticipated in Section 1.7.

Gold standard: The gold sets were restricted to mid-period leading decisions (neither the oldest, whose text cannot reliably be parsed, nor the most recent, which are not yet established as landmarks in published books) and to published BGE decisions; a setting that surfaces a relevant but non-landmark or non-BGE ruling is therefore scored as if it erred. Relevance is binary rather than graded, which together with the small gold sets makes the aggregate metrics blunt. One construct-validity risk follows from how the questions were written: the doctrinal and statute-anchored questions were authored together with the consulted legal experts, who had the authoritative target decisions in mind, so their terminology is likely to activate the Jurivoc descriptors that reach those decisions; this may inflate the apparent benefit of the topic layer on those groups, and is one reason the colloquial group (UC1), authored without that knowledge, is the more conservative test. In the event, the doctrinal group showed the smallest topic benefit of the three (Section 4.2.2), so any such inflation did not drive the headline result; the concern bears mainly on the statute-anchored group, where the topic layer did gain.

Benchmark size and evaluation: The benchmark has 31 active queries with small gold sets, which limits statistical power. Parameter tuning overfit on a query set this small and was abandoned, so the reported figures are at fixed, untuned defaults (one-hop traversal and standard reciprocal-rank fusion) and tuned performance could move in either direction. Settings B–E are non-deterministic, since they rewrite queries and choose expansions with a language model; results are reported as means over repeated runs, which dampens but does not remove this variance. The evaluation measures retrieval only: the quality of the generated answers was not assessed, neither by an LLM-as-judge nor by an expert review. The F1@10 figures in particular should be read in relative terms only: because the cut-off is fixed at ten while the gold sets hold

one to six decisions, mean F1@10 cannot exceed about 0.41 even under perfect retrieval (Section 4.2.1), so with these small gold sets it tracks Recall@10 times a near-constant factor and adds little beyond what recall and nDCG already report; a cut-off tied to each query's gold-set size, such as R-precision, would be the more informative comparator. The ordering of the settings is preserved across all four metrics, so this does not affect the comparison, but the absolute F1 values understate retrieval quality.

Topic assignment: Topic assignment is purely lexical and matches preferred labels as surface forms, inheriting the synonymy and polysemy limits of Section 2.1. It reaches only a third to two-fifths of the Court's manual indexing, and that figure is recall against the manual labels in one direction only: the precision of the automatic assignment was not evaluated, so its agreement with human indexing is characterised one-sidedly.

Data and language: The corpus and benchmark are German-only, because the topic layer is built from the German Jurivoc edition. This excluded many landmark decisions that exist only in French and, by our estimate during gold-set construction, roughly halved the number of constructible queries; it may bias the benchmark toward concepts well represented in German doctrine.

Scope of the claim: The central result should be read as holding for this corpus, benchmark and metric set, and as a direction rather than a settled magnitude. The conditional pattern across use cases (SQ2) is the more robust finding, reproducing across several metrics and supported by the per-query traces; the size of the aggregate gain is the figure most sensitive to the benchmark's construction.

5.5 Reflection on Objectives

Measured against the objective and requirements of Section 1.4:

Achieved: The primary objective, to evaluate whether Jurivoc-based topics integrated into the predecessor's graph-enhanced system improve retrieval relative to the keyword, embedding and citation-graph baselines, was met. The Jurivoc topic layer was implemented as a first-class layer in the Neo4j graph, two topic-aware retrieval settings (D and E) were added within the inherited framework, and all five settings were evaluated on a common benchmark and gold standard, satisfying the functional requirements of Table 2. The non-functional requirements were met too: the evaluation is tied to a fixed corpus snapshot, documented model versions and stored result files; inherited and newly developed components are clearly separated (Section 3.1); and the raw evaluation data is provided in the repository.

Partially achieved: Two intended elements were realised only in part. The automatic topic assignment was validated against the Court's manual Jurivoc indexing, but in one direction only, as coverage of the manual labels, not as a full precision–recall agreement (Section 4.1). And the article-level statutory texts were parsed and an ingestion script written, but the full texts were not ingested as first-class nodes; statutory anchoring relies on the inherited citation-extraction layer instead (Section 3.2).

Not achieved or deferred: An evaluation of generated answer quality, either by an LLM-as-judge or by expert review, was not carried out. Embedding-based topic assignment, fine-tuned Swiss-legal embeddings, and the French and Italian Jurivoc editions were also not pursued. Benchmark-based parameter tuning was attempted but abandoned because the query set was too small for the tuned configurations to generalise reliably (Section 3.7.4). These deviations follow chiefly from the available time and the size of the benchmark and are taken up in Section 5.6. They limit the breadth of the claim, but they do not undermine the central evaluation objective, because all reported comparisons are made within the same corpus, benchmark, model configuration and execution protocol.

5.6 Future Work

The limitations above suggest a programme of future work, ordered from the most immediate extensions to the most substantial.

Improving the topic layer: The most consequential single step is to raise topic-assignment coverage beyond the current third to two-fifths and to make matching robust to vocabulary mismatch, by replacing the purely lexical assignment with an embedding-based or sense-disambiguated one for which the evaluation scripts already provide a foundation. This would directly address two of the failure modes of Section 5.1: the cases where the topic path matches none of a relevant decision's descriptors, and the granularity mismatch from over-broad descriptors. It would also strengthen Setting D's topic discovery, which is lexical where Setting E's is already embedding-based. A full precision–recall validation of the automatic assignment against the manual Jurivoc labels would also complete the one-directional SQ1 answer (Section 5.2).

Extending the evaluation: A larger gold standard with graded rather than binary relevance, a second independent annotation, and the inclusion of French and Italian decisions would strengthen validity and lift the constraint that roughly halved the number of constructible queries (Sections 3.7.1, 5.4); a larger query set would also permit the parameter tuning that overfitted here. Adding an LLM-as-judge or expert evaluation of the generated answers would extend the assessment from retrieval to end-to-end answer quality.

Extending the system: The article-level statutory texts, already parsed, should be ingested as first-class nodes and evaluated. The multilingual Jurivoc editions would enable cross-lingual retrieval; fine-tuned Swiss-legal embeddings (Section 2.2) could replace the general-purpose model; and graph-analytic methods such as community detection, out of scope here (Section 1.7), could be tested on the enriched citation-and-topic graph. Beyond this, a user study with practitioners, a factual-consistency evaluation of the generated answers, and the production concerns deliberately excluded here, namely latency at scale, anonymisation and explainability of the topic traversal, remain open.

A personal closing note: Each of these directions is a self-contained piece of work, and the system was built, with its inherited and new components deliberately separated (Section 3.1), so that they can be taken up independently. We hope they are. The central idea, layering a

controlled, domain-specific vocabulary over citation- and embedding-based retrieval, is not specific to Swiss case law: any domain that combines an established thesaurus with a corpus linked by references could host the same construction, from patent retrieval, with its classification schemes and citation networks, to the case law of other jurisdictions. We would be glad to see the approach carried forward, whether as a project, bachelor or master thesis or in another setting, and we believe it has value well beyond the corpus studied here.

6 Lists & References

6.1 Bibliography

[1] V. Bush, 'As We May Think', *The Atlantic Monthly*, vol. 176, no. 1, pp. 101–108, Jul. 1945.

[2] E. A. A. Petrini, 'Merging LLM's with Classical NLP Methods for Large Document Set Exploration: A Hybrid Approach to Exploring Swiss Case Law', Project Thesis, Zurich University of Applied Sciences (ZHAW), Winterthur, Switzerland, 2025.

[3] P. G. Picht and G. Studen, 'Civil Law Principles and Family Law', in *Introduction to Swiss Law*, 2nd edn, M. Thommen, Ed., Zurich: sui generis Verlag, 2022, pp. 229–251. doi: 10.38107/026-10.

[4] M. Thommen, 'Swiss Legal System', in *Introduction to Swiss Law*, 2nd edn, M. Thommen, Ed., Zurich: sui generis Verlag, 2022, pp. 1–26. doi: 10.38107/026-00.

[5] M. Mahlmann, 'Legal Philosophy and Legal Theory', in *Introduction to Swiss Law*, 2nd edn, M. Thommen, Ed., Zurich: sui generis Verlag, 2022, pp. 65–90. doi: 10.38107/026-03.

[6] Schweizerisches Bundesgericht, 'Wo finde ich Bundesgerichtsentscheide, die mich interessieren?' Accessed: May 27, 2026. [Online]. Available: https://www.bger.ch/index/federal/federal-inherit-template/federal-faq/federal-faq-34.htm

[7] Schweizerisches Bundesgericht, 'Gemeinsame Medienmitteilung des Bundesgerichts, des Bundesstrafgerichts, des Bundesverwaltungsgerichts und des Bundespatentgerichts zu den Geschäftsberichten 2025', Schweizerisches Bundesgericht, Lausanne, Switzerland, Medienmitteilung (Press release), Mar. 2026. Accessed: May 27, 2026. [Online]. Available: https://www.bger.ch/files/live/sites/tfl/files/pdf/de/211.2_02_2026_yyyy_mm_dd_T_d_11_50_13.pdf

[8] P. Lewis *et al.*, 'Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks', in *Advances in Neural Information Processing Systems 33*, Red Hook, NY: Curran Associates, Inc., 2020, pp. 9459–9474. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

[9] V. Rasiah *et al.*, 'One Law, Many Languages: Benchmarking Multilingual Legal Reasoning for Judicial Support', in *The Twelfth International Conference on Learning Representations*, Vienna, Austria: OpenReview, 2024. Accessed: May 28, 2026. [Online]. Available: https://openreview.net/forum?id=7vkz7cKd1X

[10] D. Edge *et al.*, 'From Local to Global: A Graph RAG Approach to Query-Focused Summarization', Feb. 19, 2025, arXiv:2404.16130. doi: 10.48550/arXiv.2404.16130.

[11] M.-F. Moens, 'Innovative techniques for legal text retrieval', *Artif. Intell. Law*, vol. 9, no. 1, pp. 29–57, Mar. 2001, doi: 10.1023/A:1011296124449.

[12] M. Van Opijnen and C. Santos, 'On the concept of relevance in legal information retrieval', *Artif. Intell. Law*, vol. 25, no. 1, pp. 65–87, Mar. 2017, doi: 10.1007/s10506-017-9195-8.

[13] D. Locke and G. Zuccon, 'Case law retrieval: problems, methods, challenges and evaluations in the last 20 years', Feb. 15, 2022, arXiv:2202.07209. doi: 10.48550/arXiv.2202.07209.

[14] C. D. Manning, P. Raghavan, and H. Schuetze, *Introduction to Information Retrieval*. Cambridge, UK: Cambridge University Press, 2008. doi: 10.1017/CBO9780511809071.

[15] S. Robertson and H. Zaragoza, 'The Probabilistic Relevance Framework: BM25 and Beyond', *Found. Trends® Inf. Retr.*, vol. 3, no. 4, pp. 333–389, 2009, doi: 10.1561/1500000019.

[16] G. M. Rosa, R. C. Rodrigues, R. Lotufo, and R. Nogueira, 'Yes, BM25 is a Strong Baseline for Legal Case Retrieval', Oct. 25, 2021, arXiv:2105.05686. doi: 10.48550/arXiv.2105.05686.

[17] Schweizerisches Bundesgericht, 'Eurospider-Suchdienst'. Schweizerisches Bundesgericht. Accessed: May 27, 2026. [Online]. Available: https://www.bger.ch/ext/eurospider/live/de/php/aza/http/index.php

[18] Swisslex AG, 'Schweizerische Juristische Datenbank'. Swisslex AG. Accessed: May 29, 2026. [Online]. Available: https://www.swisslex.ch/

[19] entscheidsuche.ch, 'Entscheidsuche - Recherche d'arrêts'. entscheidsuche.ch. Accessed: May 28, 2026. [Online]. Available: https://entscheidsuche.ch

[20] OpenCaseLaw, 'The Complete Open Dataset of Swiss Case Law'. OpenCaseLaw. Accessed: May 25, 2026. [Online]. Available: https://opencaselaw.ch

[21] V. Karpukhin *et al.*, 'Dense Passage Retrieval for Open-Domain Question Answering', in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online: Association for Computational Linguistics, Nov. 2020, pp. 6769–6781. doi: 10.18653/v1/2020.emnlp-main.550.

[22] S. Althammer, A. Askari, S. Verberne, and A. Hanbury, 'DoSSIER@COLIEE 2021: Leveraging dense retrieval and summarization-based re-ranking for case law retrieval', in *Proceedings of the COLIEE Workshop in Association with the 18th International Conference on Artificial Intelligence and Law (ICAIL)*, Online: arXiv, Jun. 2021. doi: 10.48550/arXiv.2108.03937.

[23] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding', in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423.

[24] N. Reimers and I. Gurevych, 'Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks', in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3982–3992. doi: 10.18653/v1/D19-1410.

[25] Y. A. Malkov and D. A. Yashunin, 'Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs', *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 824–836, Apr. 2020, doi: 10.1109/TPAMI.2018.2889473.

[26] Qdrant, 'Vector Database'. Qdrant. Accessed: May 27, 2026. [Online]. Available: https://qdrant.tech/

[27] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, 'LEGAL-BERT: The Muppets straight out of Law School', in *Findings of the Association for Computational Linguistics: EMNLP 2020*, Online: Association for Computational Linguistics, Nov. 2020, pp. 2898–2904. doi: 10.18653/v1/2020.findings-emnlp.261.

[28] Y. Gao *et al.*, 'Retrieval-Augmented Generation for Large Language Models: A Survey', Mar. 27, 2024, arXiv:2312.10997. doi: 10.48550/arXiv.2312.10997.

[29] V. Magesh, F. Surani, M. Dahl, M. Suzgun, C. D. Manning, and D. E. Ho, 'Hallucination-Free? Assessing the Reliability of Leading AI Legal Research Tools', *J. Empir. Leg. Stud.*, pp. 1–27, 2025, doi: 10.1111/jels.12413.

[30] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal, 'Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions', in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Toronto, Canada: Association for Computational Linguistics, 2023, pp. 10014–10037. doi: 10.18653/v1/2023.acl-long.557.

[31] LangChain Inc., 'LangGraph Overview'. 2024. Accessed: May 27, 2026. [Online]. Available: https://docs.langchain.com/oss/python/langgraph/overview

[32] J. H. Fowler, T. R. Johnson, J. F. Spriggs, S. Jeon, and P. J. Wahlbeck, 'Network Analysis and the Law: Measuring the Legal Importance of Precedents at the U.S. Supreme Court', *Polit. Anal.*, vol. 15, no. 3, pp. 324–346, 2007, doi: 10.1093/pan/mpm011.

[33] Y. Tang, R. Qiu, Y. Liu, X. Li, and Z. Huang, 'CaseGNN: Graph Neural Networks for Legal Case Retrieval with Text-Attributed Graphs', in *Advances in Information Retrieval*, in Lecture Notes in Computer Science, no. 14609. Glasgow, UK: Springer, Cham, Mar. 2024, pp. 80–95. doi: 10.1007/978-3-031-56060-6_6.

[34] Y. Tang, R. Qiu, H. Yin, X. Li, and Z. Huang, 'CaseLink: Inductive Graph Learning for Legal Case Retrieval', in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Washington, DC, USA: ACM, Jul. 2024, pp. 1–11. doi: 10.1145/3626772.3657693.

[35] R. Hoekstra, J. Breuker, M. Di Bello, and A. Boer, 'The LKIF Core Ontology of Basic Legal Concepts', in *Proceedings of the Workshop on Legal Ontologies and Artificial Intelligence Techniques (LOAIT 2007)*, in CEUR Workshop Proceedings, vol. 321. Stanford, CA, USA: CEUR-WS.org, Jun. 2007, pp. 43–63. Accessed: May 28, 2026. [Online]. Available: https://ceur-ws.org/Vol-321/paper3.pdf

[36] P. Casanovas *et al.*, 'Semantic Web: A Legal Case Study', in *Semantic Web Technologies: Trends and Research in Ontology-based Systems*, J. Davies, R. Studer, and P. Warren, Eds, Chichester, UK: John Wiley & Sons, Ltd, 2006, pp. 259–280. doi: 10.1002/047003033X.ch12.

[37] Schweizerisches Bundesgericht, 'Jurivoc - der dreisprachige juristische Thesaurus'. Schweizerisches Bundesgericht, 2024. Accessed: May 28, 2026. [Online]. Available: https://www.bger.ch/index/juridiction/jurisdiction-inherit-template/jurisdiction-jurivoc-home.htm

[38] Schweizerisches Bundesgericht, Dokumentations- und Publikationsdienst, 'Auffinden einschlägiger Urteile in den BGE auf Internet – Suchstrategien', Schweizerisches Bundesgericht, Lausanne, 12.5.8.1, Dec. 2008. Accessed: May 28, 2026. [Online]. Available: https://www.bger.ch/files/live/sites/tfl/files/pdf/Rechtsprechung/Expertensuche_Suchstrategie_2023_04_13_d.pdf

[39] National Information Standards Organization (NISO), 'ISO 25964 - The international standard for thesauri and interoperability with other vocabularies'. National Information

Standards Organization (NISO), 2013. Accessed: May 28, 2026. [Online]. Available: https://www.niso.org/schemas/iso25964

[40] Publications Office of the European Union, 'EuroVoc – multilingual thesaurus of the European Union'. Publications Office of the European Union, 2024. Accessed: May 24, 2026. [Online]. Available: https://op.europa.eu/en/web/eu-vocabularies/eurovoc

[41] Food and Agriculture Organization of the United Nations (FAO), 'AGROVOC Multilingual Thesaurus'. Food and Agriculture Organization of the United Nations (FAO), 2024. Accessed: May 23, 2026. [Online]. Available: https://agrovoc.fao.org

[42] Schweizerische Bundeskanzlei, Sektion Terminologie, 'TERMDAT – die Terminologiedatenbank der Bundesverwaltung'. Schweizerische Bundeskanzlei, 2024. Accessed: May 23, 2026. [Online]. Available: https://www.bk.admin.ch/bk/de/home/dokumentation/sprachen/termdat.html

[43] F. Geering and J. Merane, 'Swiss Federal Supreme Court Dataset'. Zenodo, Feb. 14, 2025. doi: 10.5281/zenodo.14867950.

[44] J. Hertner, 'Swiss Case Law Dataset (voilaj/swiss-caselaw)'. Hugging Face. Accessed: Mar. 15, 2026. [Online]. Available: https://huggingface.co/datasets/voilaj/swiss-caselaw

[45] Schweizerische Bundeskanzlei, 'Fedlex – Die Publikationsplattform des Bundesrechts: Systematische Rechtssammlung (SR)'. Schweizerische Bundeskanzlei. Accessed: May 29, 2026. [Online]. Available: https://www.fedlex.admin.ch

[46] G. V. Cormack, C. L. A. Clarke, and S. Buettcher, 'Reciprocal rank fusion outperforms condorcet and individual rank learning methods', in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, Boston MA USA: ACM, Jul. 2009, pp. 758–759. doi: 10.1145/1571941.1572114.

[47] Schweizerisches Bundesgericht, 'FAQ: "Werden die Urteile des Bundesgerichts übersetzt?"' Schweizerisches Bundesgericht. Accessed: May 29, 2026. [Online]. Available: https://www.bger.ch/index/federal/federal-inherit-template/federal-faq/federal-faq-38.htm

[48] W. Portmann, I. Wildhaber, and R. Rudolph, *Schweizerisches Arbeitsrecht*, 5th edn. Zürich/St. Gallen: Dike Verlag, 2024.

[49] A. Abegg, G. Seferovic, and N. Vögeli Galli, Eds, *Personalrecht in der Schweiz: Praxisorientierter Grundriss mit Fragen, Antworten, Checklisten und Mustern*, 1st edn. Zurich: Dike Verlag, 2026. doi: 10.3256/978-3-03929-093-2.

[50] K. A. Vallender and P. Hettich, 'St. Galler Kommentar zu Art. 28 BV', in *Die schweizerische Bundesverfassung: Kommentar*, 2nd edn, Zurich/St. Gallen: Dike / Schulthess, 2008, pp. 562–575. Accessed: Mar. 28, 2026. [Online]. Available: https://alexandria.unisg.ch/entities/publication/87885d35-0c49-4e59-833d-8a91ad1a4852

[51] C. Leuenberger and B. Uffer-Tobler, *Schweizerisches Zivilprozessrecht*, 2nd edn. in Stämpflis juristische Lehrbücher. Bern: Stämpfli Verlag, 2016.

[52] J. Kren Kostkiewicz, *Schuldbetreibungs- & Konkursrecht*, 3rd edn. Zürich Basel Genf: Schulthess, 2023.

[53] M. Pieth and M. Simmler, *Strafrecht Besonderer Teil*, 3. vollständig überarbeitete Auflage. Basel: Helbing Lichtenhahn, 2024.

[54] F. Uhlmann, *Das Willkürverbot (Art. 9 BV)*. Bern: Stämpfli, 2005.

[55] H. Rey and I. Wildhaber, *Ausservertragliches Haftpflichtrecht*, 6th edn. Zürich Genf: Schulthess, 2024.

[56] D. Staehelin, P. Grolimund, A. Staehelin, D. Ammann, O. L. Mosimann, and L. Bopp, *Zivilprozessrecht: unter Einbezug des Anwaltsrechts und des internationalen Zivilprozessrechts*, 4th edn. Zürich Genf: Schulthess Juristische Medien, 2024.

[57] S. Schlegel, 'Kommentierung zu Art. 26 BV', in *Onlinekommentar zur Bundesverfassung*, Onlinekommentar.ch, 2025, pp. 1–56. Accessed: Jun. 04, 2026. [Online]. Available: https://onlinekommentar.ch/de/kommentare/bv26

[58] E. Zysset, 'Kommentierung zu Art. 34 DSG', in *Onlinekommentar zum Bundesgesetz über den Datenschutz*, Onlinekommentar.ch, 2023. Accessed: May 01, 2026. [Online]. Available: https://onlinekommentar.ch/de/kommentare/dsg34

[59] J. Drittenbass, 'Kommentierung zu Art. 13 BV', in *Onlinekommentar zur Bundesverfassung*, Bern: Onlinekommentar.ch. Accessed: May 02, 2026. [Online]. Available: https://onlinekommentar.ch/de/kommentare/bv13

[60] L. Mouttet, 'Observationen in der Sozialversicherung: Eine kritische Analyse von Art. 43a und Art. 43b ATSG', Dissertation, Universität Zürich, Rechtswissenschaftliche Fakultät, Zurich, 2022. Accessed: May 10, 2026. [Online]. Available: https://eizpublishing.ch/publikationen/observationen-in-der-sozialversicherung/

[61] M. Müller, 'Polizeiliche Generalklausel: klärende Worte des Bundesgerichts', *Z. Bernischen Juristenvereins*, vol. 148, pp. 220–228, 2012.

[62] R. Kiener, B. Schindler, and A. Brunner, *Polizeirecht: Rechtliche Grundlagen der polizeilichen Tätigkeit in der Schweiz*, 1st edn. Bern: Stämpfli Verlag AG, 2026.

[63] L. Staffler, 'Kommentierung zu Art. 32 BV', in *Onlinekommentar zur Bundesverfassung*, in Onlinekommentar — Der frei zugängliche Rechtskommentar. , Spiegel b. Bern: Stiftung Onlinekommentar, 2024, pp. 1–45. Accessed: Mar. 25, 2026. [Online]. Available: https://onlinekommentar.ch/de/kommentare/bv32

[64] R. Cueni, 'Kommentierung zu Art. 16 BV', in *Onlinekommentar zur Bundesverfassung*, Onlinekommentar.ch, 2024. Accessed: Mar. 25, 2026. [Online]. Available: https://onlinekommentar.ch/de/kommentare/bv16

[65] C. Heri, 'Kommentierung zu Art. 10 BV', in *Onlinekommentar zur Bundesverfassung*, Onlinekommentar.ch, 2023. Accessed: May 10, 2026. [Online]. Available: https://onlinekommentar.ch/de/kommentare/bv10

[66] P. M. Zumsteg, 'Kommentierung zu Art. 22 BV', in *Onlinekommentar zur Bundesverfassung*, Onlinekommentar.ch, 2023. Accessed: May 10, 2026. [Online]. Available: https://onlinekommentar.ch/de/kommentare/bv22

[67] J. Reich, 'Wirtschaftsverfassung zwischen Sogwirkung des Faktischen und ordnungspolitischer Verklärung – Wirtschaftliche Tätigkeiten des Gemeinwesens in Konkurrenz zu Privaten als methodologische Herausforderung der Verfassungsrechtswissenschaft', *Z. Für Schweiz. Recht*, vol. 140, no. I, pp. 131–153, 2021.

[68] Eidgenössische Kommission gegen Rassismus (EKR), 'Diskriminierung bei der Einbürgerung: Stellungnahme der EKR zur aktuellen Situation', Eidgenössische Kommission gegen Rassismus (EKR), Bern, Stellungnahme, 2007. Accessed: May 04, 2026. [Online]. Available: https://www.ekr.admin.ch/pdf/Einbuergerung_D_version_webbf72.pdf

[69] R. Walther, 'Kommentierung zu Art. 30 BV', in *Onlinekommentar zur Bundesverfassung*, 2024. Accessed: May 10, 2026. [Online]. Available: https://onlinekommentar.ch/de/kommentare/bv30

[70] M. Piolino and R. Cueni, 'Kommentierung zu Art. 17 BV', in *Onlinekommentar zur Bundesverfassung*, Onlinekommentar, 2023. Accessed: May 10, 2026. [Online]. Available: https://onlinekommentar.ch/de/kommentare/bv17

[71] S. Forster, *Die gewaltenteilig motivierte Kognitionspraxis des Bundesgerichts*. in Zürcher Studien zum öffentlichen Recht, no. 299. Zurich: Schulthess, 2024.

[72] D. R. Buschor, 'Managing Partnerinnen und Managing Partner in Anwaltsunternehmen der DACH-Region: Rollen – Typen – Anforderungen', Dissertation (Dr. iur.), Universität St. Gallen (HSG), St. Gallen, Switzerland, 2022. Accessed: May 11, 2026. [Online]. Available: https://alexandria.unisg.ch/entities/publication/f4eb5abd-f606-497b-932d-fe347d12f88b

6.2 Glossary

Term	Definition
ADDRESSES_TOPIC	Graph relationship linking a corpus node, such as a Ruling, Fact or Consideration, to a Jurivoc topic node when the text is assigned to that legal concept.
Akoma Ntoso	XML standard for representing legislative and judicial documents. In this thesis, it is relevant because statutory texts from Fedlex were parsed from the Akoma Ntoso format during development.
Baseline	Reference retrieval setting against which the new topic-aware settings are compared. In this thesis, Settings A to C serve as baselines.
BGE	Published leading decision of the Swiss Federal Supreme Court in the official collection. BGE decisions are used as the main benchmark population in this thesis.
BGer	Non-leading ruling of the Swiss Federal Supreme Court. In the benchmark, non-leading BGer decisions are included only where inherited from the predecessor benchmark.
Bundesgericht	Swiss Federal Supreme Court, the highest judicial authority of the Swiss Confederation.
Citation graph	Network of decisions connected by references they make to one another. It forms part of the inherited graph structure used for retrieval.
Citation-graph traversal	Retrieval step that follows citation links between court decisions in order to find legally connected rulings.
Concept layer	Additional graph layer consisting of legal topic nodes. In this thesis, the concept layer is derived from Jurivoc.
Consideration	Reasoning part of a Swiss Federal Supreme Court decision. In German decisions, this corresponds to the Erwägungen.
Controlled vocabulary	Curated set of terms used to describe concepts consistently. Jurivoc is used as such a vocabulary in this thesis.
Dense retrieval	Retrieval method that represents queries and documents as embeddings and ranks them by vector similarity.
Descriptor	Preferred term in a thesaurus. In Jurivoc, descriptors represent legal concepts and are used as the basis for topic nodes.
Embedding	Numerical vector representation of text, used to compare semantic similarity between queries, passages, decisions or topics.
Erwägungen	German term for the considerations or reasoning section of a court decision. This is the part in which the court explains its legal reasoning.

Term	Definition
Fedlex	Official Swiss federal law publication platform. It provides legal texts and was used during development for statutory text parsing.
Gold standard	Manually constructed benchmark of queries and relevant decisions used to evaluate retrieval performance.
Graph database	Database that stores information as nodes and relationships. This thesis uses Neo4j for the legal graph.
Graph-enhanced retrieval	Retrieval approach that combines text-based search with graph relationships such as citations, statutory references or topic links.
Jurivoc	Legal thesaurus used in the documentation practice of the Swiss Federal Supreme Court. In this thesis, Jurivoc provides the source for the topic layer.
Keyword retrieval	Retrieval method based primarily on lexical term matching.
Leading decision	Decision selected for publication in the official collection of the Swiss Federal Supreme Court. In this thesis, leading decisions correspond to BGE decisions.
Leitentscheid	German term for leading decision. It refers to a decision with broader doctrinal significance.
Lexical retrieval	Retrieval based on matching words or terms rather than semantic vector similarity.
MRR (Mean Reciprocal Rank)	Ranking metric that measures how highly the first relevant result appears in the returned result list.
Multi-hop retrieval	Retrieval in which the system follows several linked steps, for example from a query to a decision and then through citations or topics to further decisions.
Neo4j	Graph database used in this thesis to store decisions, facts, considerations, citations, statute references and topic nodes.
Non-descriptor	Variant term or synonym in a thesaurus that points to a preferred descriptor through a UF or USE relation.
OpenCaseLaw	Open Swiss legal data source used as the main source for structured decision content in this thesis.
PageRank	Graph-centrality algorithm used to estimate the importance of nodes in a network, for example decisions in a citation graph.
Qdrant	Vector database used in this thesis to store and search embeddings.
Recall@10	Share of all relevant gold standard decisions that appear in the top ten retrieved results.
Reciprocal Rank Fusion	Rank-fusion method that combines multiple ranked result lists by assigning higher weight to documents that appear near the top of one or more lists.

Term	Definition
Regeste	Headnote of a leading Swiss Federal Supreme Court decision. It summarises the legally relevant points of the decision.
Retrieval-Augmented Generation	Architecture in which a retrieval system first selects relevant documents and a language model then generates an answer based on the retrieved context.
Sachverhalt	Statement of facts section of a Swiss court decision.
Setting A	Keyword-based retrieval baseline.
Setting B	Embedding-based retrieval baseline.
Setting C	Embedding-based retrieval with citation-graph traversal.
Setting D	Topic plus keyword graph retrieval.
Setting E	Topic plus embedding graph retrieval.
Statute reference	Link between a decision, consideration or other corpus node and a cited legal provision.
Swiss Federal Supreme Court	Highest judicial authority of the Swiss Confederation. In German, Bundesgericht.
TERMDAT	Swiss federal terminology database considered during vocabulary selection but not used in the evaluated retrieval system.
Thesaurus relation	Structured relation between terms in a thesaurus, such as broader term, narrower term, related term or used-for relation.
Topic layer	Graph layer consisting of topic nodes derived from Jurivoc and linked to the decision corpus.
Topic node	Graph node representing a legal concept derived from Jurivoc.
Topic-aware retrieval	Retrieval strategy that uses Jurivoc topic nodes in addition to keyword, embedding or citation signals.
Vector database	Database designed to store and search embeddings efficiently.

7 Appendix

7.1 Benchmark Queries and Gold Standard

This appendix lists all 31 active benchmark queries used in the evaluation reported in Chapter 4, together with their use-case category (UC1/UC2/UC3, see Section 1.6), their gold-standard decisions and their sources. The set comprises 8 UC1 (colloquial), 15 UC2 (doctrinal) and 8 UC3 (statute-anchored) queries, yielding 84 active query–decision pairs.

Inheritance from Petrini: Nine of Petrini's ten original queries are carried over (Q1a, Q2a, Q3–Q7, Q9, Q10). One query, originally on liability reduction for concurrent causation, was dropped because two target decisions from 1931 were too old for the structured corpus and one target decision was French-language and therefore outside the German-language evaluation corpus.

Language exclusions: French-language target decisions are listed in italics under each query but are excluded from the evaluation because the corpus used in this thesis is the German-language subset (see Section 3.2). The 84 active query–decision pairs reflect this exclusion.

ID	UC	Topic & Query (DE)	Gold-Standard Decisions	Source
Q1a	**UC3**	**Diskriminierung — Glaubhaftmachung & Beweislast (Art. 6 GlG)** Gem. Art. 6 GlG muss eine betroffene Person die Diskriminierung glaubhaft machen. Was bedeutet das? Wen trifft die Beweislast?	BGE 144 II 65 *Ausgeschlossen: BGE 130 III 145 (fr)*	Portmann/Wildhaber/Rudolph, Schweizerisches Arbeitsrecht, 2024 (Petrini Q1) [48]
Q1b	**UC2**	**Diskriminierung — Geschlechtsbedingte Lohndiskriminierung** Wie kann eine Arbeitnehmerin im privatrechtlichen oder öffentlich-rechtlichen Arbeitsverhältnis eine geschlechtsbedingte Lohndiskriminierung glaubhaft machen, und wann ist ein Lohnunterschied sachlich gerechtfertigt?	BGE 125 III 368 BGE 142 II 49 BGE 144 II 65	Abegg/Seferovic/Vögeli Galli (Hrsg.), Personalrecht in der Schweiz, Dike 2026 [49]
Q2a	**UC1**	**Streikrecht — Voraussetzungen rechtmässiger Arbeitskampf** Was sind die Voraussetzungen für einen rechtmässigen Arbeitskampf?	BGE 125 III 277 *Ausgeschlossen: BGE 132 III 122 (fr)*	Portmann/Wildhaber/Rudolph, Schweizerisches Arbeitsrecht, 2024 (Petrini Q2) [48]

Q2b	**UC2**	**Streikrecht — Voraussetzungen und arbeitsvertragliche Folgen** Unter welchen Voraussetzungen ist ein kollektiver Streik im privatrechtlichen Arbeitsverhältnis rechtmässig, und welche Folgen hat die Teilnahme an einem rechtmässigen Streik für den Arbeitsvertrag?	BGE 111 II 245 BGE 125 III 277	Abegg/Seferovic/Vögeli Galli (Hrsg.), Personalrecht in der Schweiz, Dike 2026 [49]; Vallender/Hettich, St. Galler Kommentar zu Art. 28 BV [50]
Q3	**UC2**	**Rechtliches Gehör zu Rechtsfragen — Ausnahmen von Iura novit curia** Im schweizerischen Zivilprozessrecht, wann sind die Parteien zu Rechtsfragen anzuhören (entgegen dem Grundsatz „Iura novit curia")?	BGE 127 V 431 BGer 4A_178/2015 *Ausgeschlossen:* *BGE 130 III 35 (fr)* *BGE 129 II 497 (fr)*	Leuenberger/Uffer-Tobler, Schweizerisches Zivilprozessrecht, 2. Aufl. 2016 (Petrini Q3) [51]
Q4	**UC1**	**Scheidungs-Kostenverteilung bei Rückzug** Wer trägt die Kosten für eine Scheidungsklage im Falle eines Rückzuges?	BGE 139 III 358	Leuenberger/Uffer-Tobler, Schweizerisches Zivilprozessrecht, 2. Aufl. 2016 (Petrini Q4) [51]
Q5	**UC1**	**Verwertungserlös (SchKG) — Grundstückgewinnsteuer als Verwertungskosten** Im Rahmen des Verwertungserlöses, gehören die Grundstückgewinnsteuern zu den Verwertungskosten?	BGE 122 III 246 BGer 5A_229/2009 BGer 5A_989/2016 *Ausgeschlossen:* *BGer 5A_651/2015 (fr)*	Kren Kostkiewicz, Schuldbetreibungs- und Konkursrecht, 2023 (Petrini Q5) [52]
Q6	**UC2**	**Rechtsöffnung — Fortsetzung der Betreibung bei fehlender Zustellung** Kann man eine Betreibung fortführen, wenn der Rechtsvorschlag definitiv beseitigt worden ist, aber der Schuldner weder eine Vorladung zu einer Rechtsöffnungsverhandlung noch den Rechtsöffnungsentscheid erhalten hat?	BGE 130 III 396 BGer 5A_738/2010	Kren Kostkiewicz, Schuldbetreibungs- und Konkursrecht, 2023 (Petrini Q6) [52]
Q7	**UC3**	**Verbreiten gefährlicher Krankheiten (Art. 231 StGB)** In Art. 231 StGB, wie muss man den Begriff „verbreiten" interpretieren?	BGE 125 IV 242 BGE 131 IV 1 BGE 134 IV 193 *Ausgeschlossen:* *BGE 116 IV 125 (fr)*	Pieth/Simmler, Strafrecht BT, 2024 (Petrini Q7) [53]

Q8	**UC1**	**Willkür in der Rechtssetzung (Art. 9 BV)** Welche Leitentscheide gibt es zur Willkür in der Rechtssetzung?	BGE 122 I 18 BGE 123 II 16 BGE 129 I 1	Uhlmann (2005), §2 — Erste Annäherungen an Begriff und Inhalt des Willkürverbots / I. Willkürformel des Bundesgerichts [54]
Q9	**UC2**	**Schranken der Eigentumsausübung — Nachbarrecht & öffentlich-rechtliche Vorschriften** Welche Schranken der Eigentumsausübung ergeben sich aus dem Nachbarrecht und öffentlich-rechtlichen Vorschriften?	BGE 107 II 134 BGE 82 II 397 BGE 88 II 331 *Ausgeschlossen: BGE 91 II 183 (fr)*	Wildhaber/Rey, Ausservertragliches Haftpflichtrecht, 2024 (Petrini Q9) [55]
Q10	**UC2**	**Übung und Ortsgebrauch — Beweispflicht der Parteien** Muss das Bestehen einer Übung oder eines Ortsgebrauchs bewiesen werden?	BGE 86 II 256 BGE 90 II 92 BGE 91 II 356 BGE 94 II 157	Staehelin/Grolimund, Zivilprozessrecht: Unter Einbezug des Anwaltsrechts und des internationalen Zivilprozessrechts, 2024 (Petrini Q10) [56]
Q11	**UC2**	**Materielle Enteignung / Auszonung (Art. 26 BV)** Ein Grundstück wird bei der erstmaligen Schaffung einer RPG-konformen Zonenordnung nicht in die Bauzone aufgenommen, obwohl es nach altem Recht bebaubar war. Besteht ein Entschädigungsanspruch?	BGE 121 II 417 BGE 122 II 326 BGE 131 II 728	Schlegel, Onlinekommentar zur BV, Art. 26 BV (OK-Schlegel) [57]
Q12	**UC1**	**Datenschutz — Informationelle Selbstbestimmung (Art. 13 BV)** Darf der Staat einfach so Daten über mich sammeln, auch wenn ich gar nichts Verbotenes gemacht habe?	BGE 136 I 87 BGE 143 I 253 BGE 146 I 11 BGE 147 I 346	Zysset-Kommentar zu Art. 34 DSG [58]
Q13	**UC3**	**Aufenthaltsrecht — Schutz des Privatlebens (Art. 8 EMRK / Art. 13 BV)** Unter welchen Voraussetzungen vermittelt das Recht auf Achtung des Privatlebens nach Art. 8 EMRK bzw. Art. 13 Abs. 1 BV einer ausländischen Person nach langjährigem Aufenthalt einen Anspruch auf Erteilung oder Verlängerung einer Aufenthaltsbewilligung in der Schweiz?	BGE 144 I 266 BGE 147 I 268	Drittenbass, Onlinekommentar zur BV, Art. 13 BV (OK-Drittenbass) [59]

Q14	**UC2**	**Observation im Sozialversicherungsrecht** Welche bundesgerichtlichen Leitentscheide regeln die Zulässigkeit und Verwertbarkeit von Observationen versicherter Personen durch Sozialversicherungen (Unfall- und Invalidenversicherung), und wie hat sich die Rechtsprechung nach dem EGMR-Urteil Vukota-Bojić gegen die Schweiz entwickelt?	BGE 132 V 241 BGE 135 I 169 BGE 137 I 327 BGE 143 I 377	Mouttet, Observationen in der Sozialversicherung — Eine kritische Analyse von Art. 43a und Art. 43b ATSG, Diss. Universität Zürich, 2022 [60]
Q15	**UC2**	**Polizeiliche Generalklausel** Unter welchen Voraussetzungen darf sich eine Behörde trotz fehlender gesetzlicher Grundlage auf die polizeiliche Generalklausel stützen, und welche Bedeutung hat das frühere Erfordernis der Unvorhersehbarkeit nach der neueren Rechtsprechung noch?	BGE 126 I 112 BGE 128 I 327 BGE 130 I 369 BGE 137 II 431 BGE 147 I 161	Müller, ZBJV 148/2012 [61]
Q16	**UC1**	**Polizeirecht — verdeckte Vorermittlung** Wie ist die verdeckte Ermittlung in der Schweiz geregelt?	BGE 134 IV 266 BGE 140 I 353 BGE 143 IV 27 BGE 149 I 218	Polizeieinsatzrecht, Vierter Teil § 14 „Präventive Überwachungsmassnahmen", IV.–V. [62]
Q17	**UC3**	**Unschuldsvermutung / in dubio pro reo (Art. 32 Abs. 1 BV) — Kern der Doktrin** Welche Bedeutung hat die Maxime in dubio pro reo als Ausfluss der Unschuldsvermutung nach Art. 6 Ziff. 2 EMRK, Art. 4 aBV und Art. 32 Abs. 1 BV im Strafverfahren als Beweislastregel und Beweiswürdigungsregel?	BGE 120 Ia 31 BGE 127 I 38 BGE 144 IV 345	Staffler, Onlinekommentar zur BV, Art. 32 BV (OK-Staffler) [63]
Q18	**UC3**	**Allgemein zugängliche Quellen (Art. 16 Abs. 3 BV)** Welche Bedeutung hat der Begriff der „allgemein zugänglichen Quellen" in Art. 16 Abs. 3 BV für den Zugang zu staatlichen Informationen?	BGE 113 Ia 309 BGE 124 I 176 BGE 127 I 145 BGE 137 I 16 BGE 137 I 8 BGE 139 I 129	Cueni, Kommentar zu Art. 16 BV [64]

Q19	**UC3**	**Medikamentöse Zwangsbehandlung im Rahmen der fürsorgerischen Unterbringung (Art. 10 Abs. 2 BV)** Welche Anforderungen stellt das Bundesgericht an schwere Eingriffe in die persönliche Freiheit nach Art. 10 Abs. 2 BV bei medikamentösen Zwangsbehandlungen im Rahmen der fürsorgerischen Unterbringung, und welche Rolle spielen die Einwilligung urteilsfähiger Betroffener sowie das Verbot unmenschlicher Behandlung?	BGE 126 I 112 BGE 127 I 6 BGE 130 I 16	Heri, Kommentar zu Art. 10 BV [65]
Q20	**UC1**	**Polizeikosten bei Demonstrationen (Versammlungsfreiheit Art. 22 BV)** Dürfen Demonstrierenden Kosten für Polizeieinsätze auferlegt werden?	BGE 143 I 147	Zumsteg, Onlinekommentar zur BV, Art. 22 BV (OK-Zumsteg) [66]
Q21	**UC1**	**Bewilligungspflicht und Interessenabwägung bei Demonstrationen (Art. 22 BV)** Darf eine Demonstration auf öffentlichem Grund einer Bewilligungspflicht unterworfen werden, und welche Interessen muss die Behörde abwägen?	BGE 127 I 164 BGE 132 I 256	Zumsteg, Onlinekommentar zur BV, Art. 22 BV (OK-Zumsteg) [66]
Q22	**UC3**	**Wirtschaftliche Tätigkeit des Gemeinwesens in Konkurrenz zu Privaten (Art. 27 / Art. 94 BV)** Unter welchen verfassungsrechtlichen Voraussetzungen darf das Gemeinwesen (Bund, Kantone, Gemeinden) wirtschaftliche Tätigkeiten in Konkurrenz zu privaten Unternehmen ausüben, und wie beurteilt sich dies nach Art. 27 und Art. 94 BV (Wirtschaftsfreiheit, Wettbewerbsneutralität, öffentliches Interesse, Fiskalinteressen, staatliche Sachversicherung)?	BGE 138 I 378 BGE 143 II 425	Reich, ZSR 2021/I [67]

Q23	**UC2**	**Diskriminierungsverbot bei Einbürgerungen** Warum sind Urnenabstimmungen über Einbürgerungsgesuche verfassungsrechtlich problematisch, wenn Bewerber aufgrund ihrer Herkunft benachteiligt werden oder ein negativer Einbürgerungsentscheid nicht begründet werden kann?	BGE 129 I 217 BGE 129 I 232	EKR-Stellungnahme 2007 (Einbürgerung) [68]
Q24	**UC2**	**Richterwahl und Wiederwahl (Art. 30 Abs. 1 BV)** Wie vereinbaren sich politische Wahl- und Wiederwahlverfahren für Richterinnen und Richter mit der richterlichen Unabhängigkeit und Unparteilichkeit, wenn bisherige Amtsinhaber bei der Wiederwahl privilegiert werden oder Richter wegen Alters- und Eignungskriterien nicht wiedergewählt werden?	BGE 136 I 207 BGE 143 I 211 BGE 147 I 1	Walther, Onlinekommentar zur BV, Art. 30 BV [69]
Q25	**UC3**	**Justizöffentlichkeit — Verhandlungs- und Urteilsöffentlichkeit (Art. 30 Abs. 3 BV)** Welche Anforderungen ergeben sich aus Art. 30 Abs. 3 BV an die Öffentlichkeit von Gerichtsverhandlungen und Urteilsverkündungen, und in welchem Umfang sind diese Garantien einschränkbar?	BGE 137 I 16 BGE 139 I 129	Walther, Onlinekommentar zur BV, Art. 30 Abs. 3 BV [69]
Q26	**UC2**	**Grundrechtsbindung staatsnaher Unternehmen im privatrechtlichen Handeln** Inwieweit sind staatlich konzessionierte oder mit öffentlichen Aufgaben betraute Unternehmen wie die SRG und die SBB in ihrem privatrechtlichen Handeln — etwa beim Aushängen von Plakaten, bei der Auswahl von Werbung oder beim Umgang mit Nutzerkommentaren — an die Meinungsäusserungs- und Pressefreiheit Dritter gebunden?	BGE 137 I 8 BGE 138 I 274 BGE 139 I 306	Piolino/Cueni, Onlinekommentar zur BV, Art. 17 BV (OK-Piolino/Cueni) [70]

Q27	**UC2**	**Quellenschutz und Redaktionsgeheimnis im Strafverfahren (Art. 17 Abs. 3 BV)** Unter welchen Voraussetzungen darf das Bundesgericht Medienschaffende in einem Strafverfahren zur Offenlegung ihrer Informationsquellen verpflichten, eine Beschlagnahme von Recherchematerial oder eine Überwachung des Fernmeldeverkehrs anordnen, und welche Rolle spielen dabei das Redaktionsgeheimnis und die Pressefreiheit?	BGE 123 IV 236 BGE 132 I 181 BGE 140 IV 108	Piolino/Cueni, Onlinekommentar zur BV, Art. 17 BV (OK-Piolino/Cueni) [70]
Q28	**UC2**	**Zurückhaltung des Bundesgerichts bei Entscheiden der Kommunikationskommission (ComCom)** Wann prüft das Bundesgericht Entscheide der Kommunikationskommission nur zurückhaltend, weil diese bei technischen oder wirtschaftlichen Fragen mehr Fachwissen hat?	BGE 131 II 13 BGE 132 II 257	Forster (2024), Teil IV–V (Kognition und Zurückhaltung des Bundesgerichts) [71]
Q29	**UC2**	**Anwalts-AG — Institutionelle Unabhängigkeit (Art. 8 BGFA)** Welche Anforderungen stellt das Bundesgericht an die institutionelle Unabhängigkeit einer Anwaltskanzlei in der Rechtsform der Aktiengesellschaft hinsichtlich Verwaltungsrat und Aktionariat?	BGE 130 II 87 BGE 138 II 440	Buschor, Managing Partner in Anwaltsunternehmen der DACH-Region, Dissertation Nr. 5214, Universität St. Gallen, 2022 [72]

7.2 Heatmap per Metric

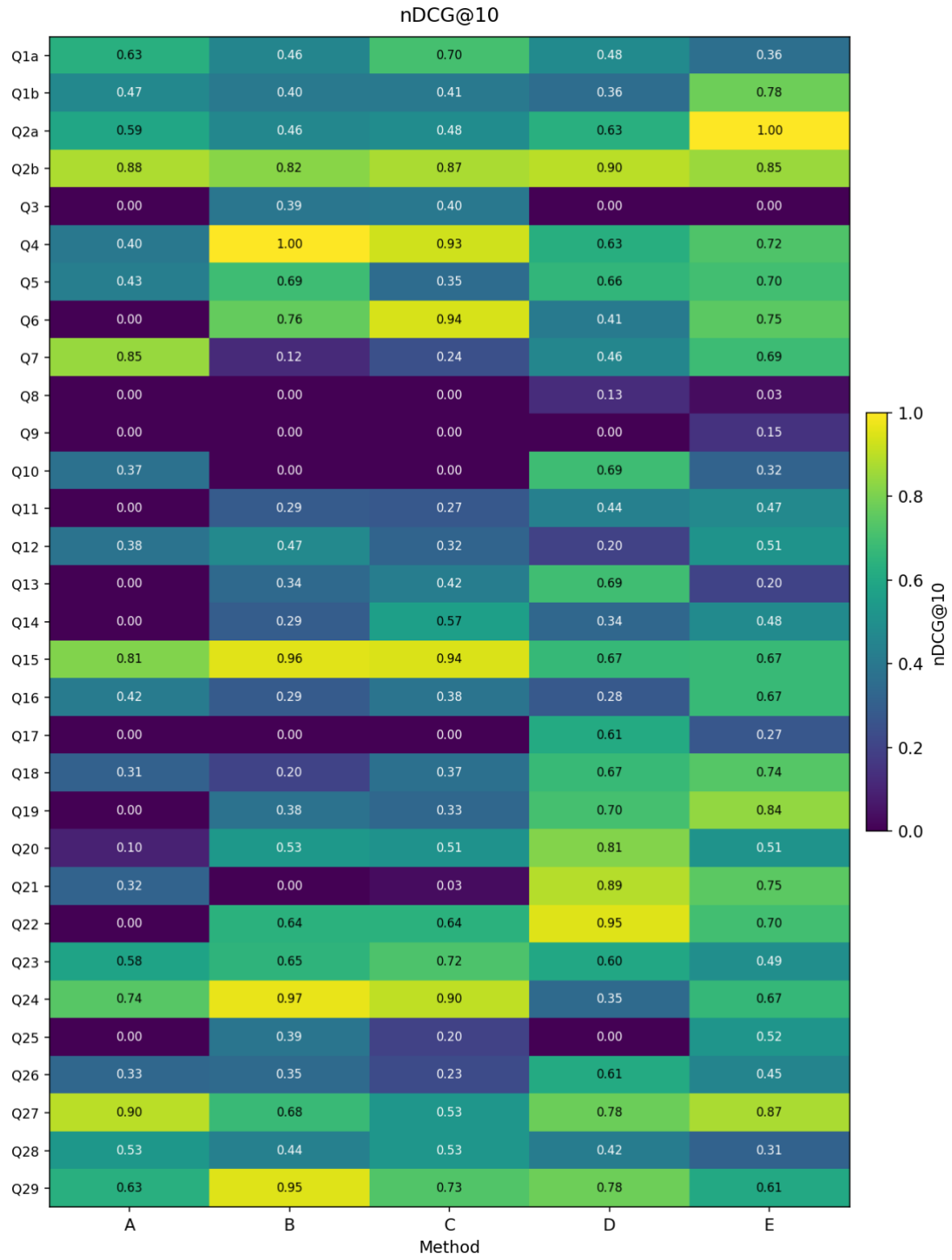

Figure 20: Per query nDCG heatmap

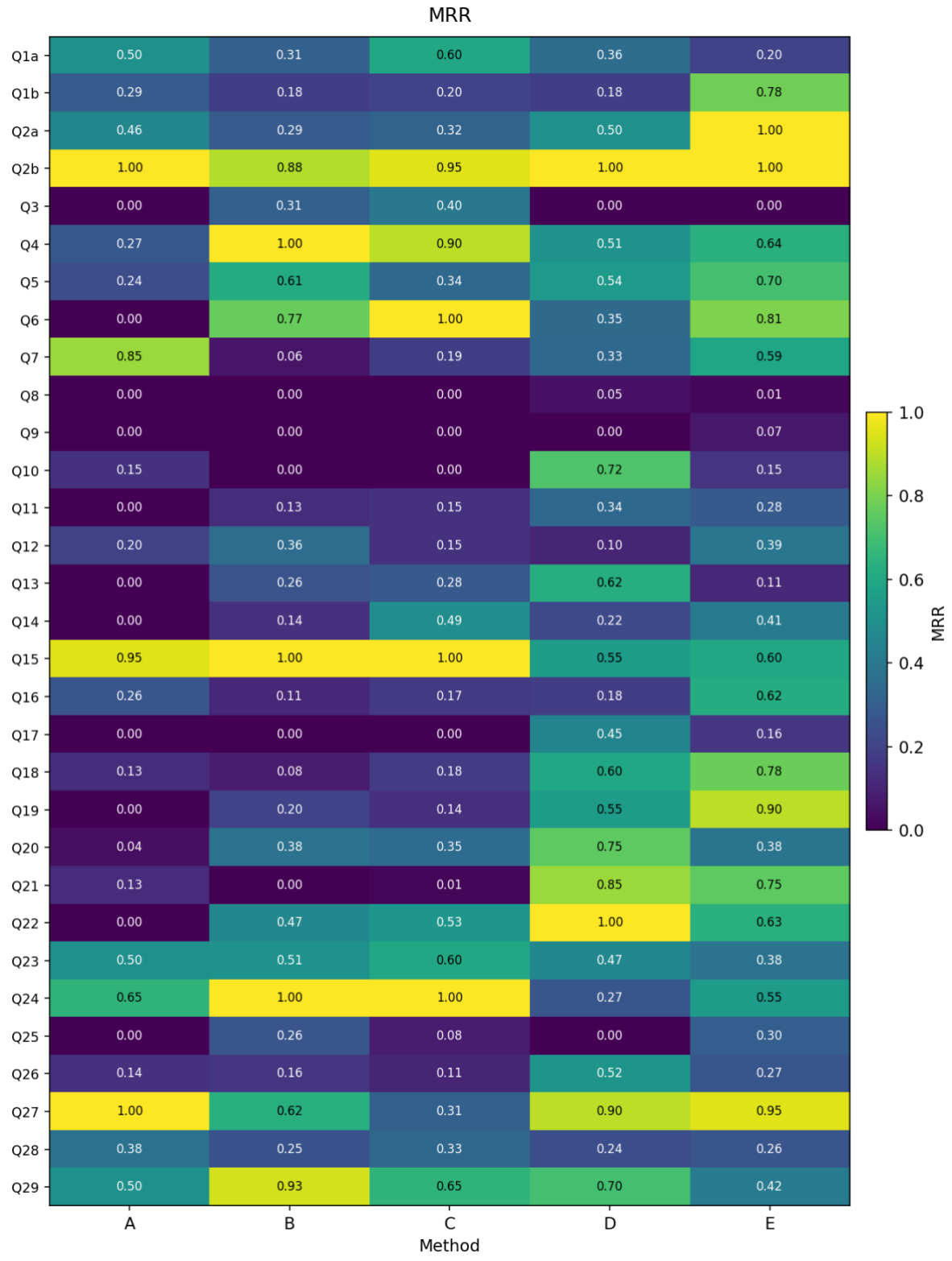

Figure 21: Per query MRR heatmap

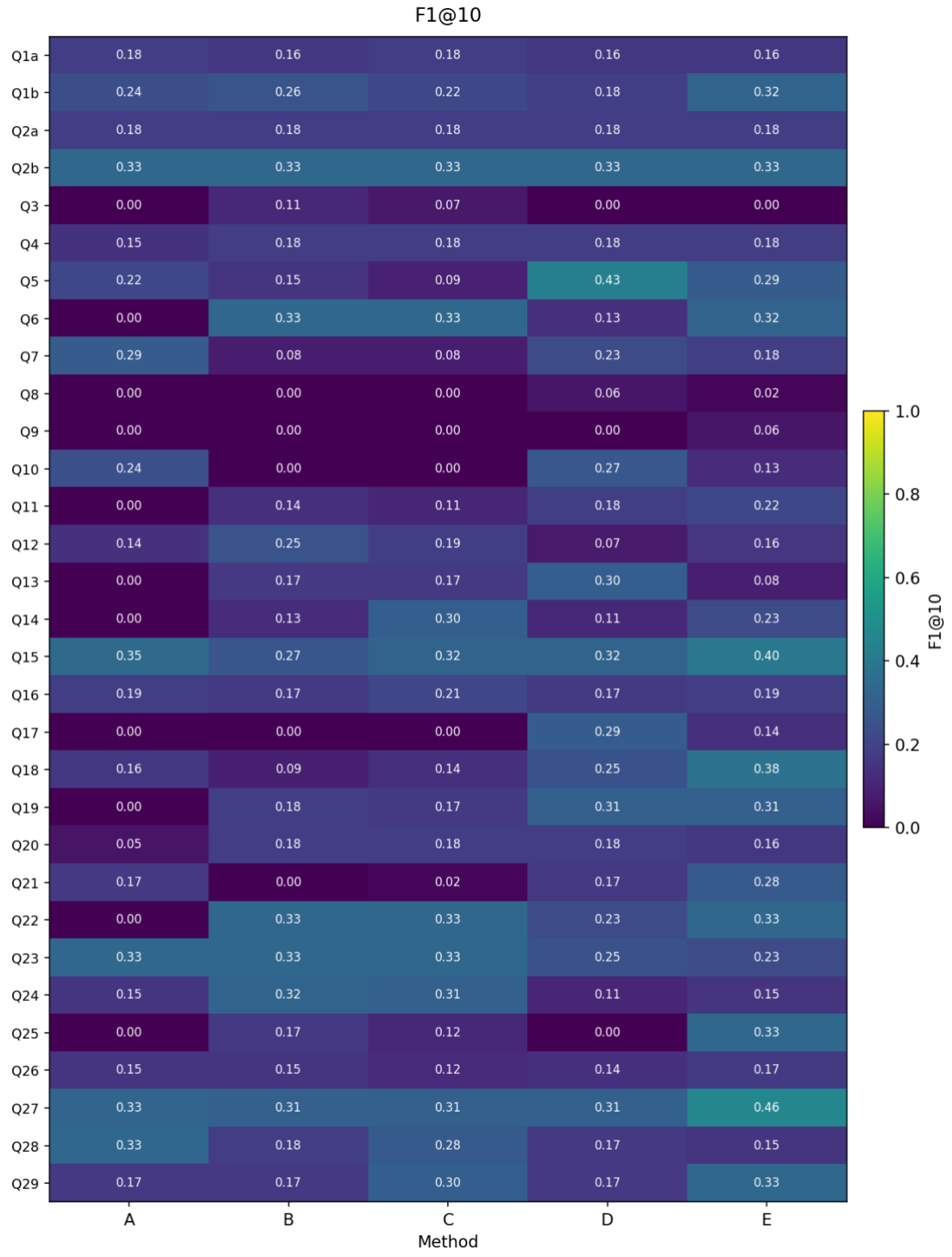

Figure 22: Per query F1 heatmap

7.3 User Setup Guide

This guide is a zero-to-production walkthrough for the Swiss Legal Graph RAG system. It takes you from an empty machine to a deployment that is publicly reachable by domain, in four phases:

1. **Phase 1 — External services**: Azure OpenAI (required), PostHog (optional), and your `.env`.
2. **Phase 2 — Docker Compose & data ingestion**: start the databases and populate the graph + vector index.
3. **Phase 3 — Run it locally**: backend + frontend dev servers.
4. **Phase 4 — Production**: TLS, a public domain, and running as a service.

The system has three moving parts: an offline **pipeline** that builds the knowledge graph (Neo4j) and vector index (Qdrant), a FastAPI **backend** that answers queries, and a Next.js **frontend**. Neo4j, Qdrant, backend, frontend and a Caddy reverse proxy all run from a single `docker compose` stack.

Prerequisites

Install on the host:

Tool	Used for	Notes
Docker Engine + Compose plugin	the whole stack	`docker compose version` should work
uv	pipeline + backend (Python 3.12)	manages the Python workspace
Bun	frontend	only needed for local frontend dev
git	cloning	

Hardware: the compose stack reserves **12 GB for Neo4j** and **8 GB for Qdrant**, so plan for a host with **~24–32 GB RAM**. Adjust the limits in `app/docker-compose.yaml` if you have less.

> **Platform:** verified on **macOS** and **Linux**. **Windows is not supported,** run it under WSL2.

Clone and install the Python workspace (pipeline + backend):

```
git clone <repo-url> ba-thesis-swiss-legal-graph-rag
cd ba-thesis-swiss-legal-graph-rag
uv sync          # installs the uv workspace: pipeline + app/backend
```

Phase 1 — External services (PostHog optional)

The only **mandatory** external service is **Azure OpenAI**. Neo4j and Qdrant are *not* external — they run as containers in Phase 2. PostHog and Azure Blob Storage are optional.

1.1 Azure OpenAI (required)

Create an Azure OpenAI resource and deploy **two** models:

Model	Purpose	Default deployment name expected	env vars
`text-embedding-3-large` (3072-dim)	embeds rulings for vector search	`text-embedding-3-large-high-limit`	`AZURE_AI_EMBEDDINGS_*`
`gpt-4.1` (or a GPT-4-class chat model)	query interpretation + answers	`gpt-4.1-18m`	`AZURE_AI_LLM_*` + `LLM_DEPLOYMENT`

From the Azure portal collect, for each deployment: the **endpoint**, an **API key**, the **deployment name**, and the **API version**. These map to:

```
# Embeddings (used by the pipeline AND the backend)
AZURE_AI_EMBEDDINGS_ENDPOINT=https://<resource>.cognitiveservices.azure.com/openai/v1/
AZURE_AI_EMBEDDINGS_KEY=...
AZURE_AI_EMBEDDINGS_DEPLOYMENT=text-embedding-3-large-high-limit
AZURE_AI_EMBEDDINGS_API_VERSION=2024-12-01-preview

# LLM (used by the backend)
AZURE_AI_LLM_ENDPOINT=https://<resource>.cognitiveservices.azure.com/openai/v1/
AZURE_AI_LLM_KEY=...
LLM_DEPLOYMENT=gpt-4.1-18m
LLM_API_VERSION=2024-12-01-preview
```

> The deployment names above are the defaults baked into `app/backend/app/config.py`. If you name your Azure deployments differently, set the env vars to match.

1.2 PostHog analytics (optional)

Skip this and leave the vars blank if you don't want analytics. Otherwise create a PostHog project and set:

```
# backend (server-side events)
POSTHOG_API_KEY=...
POSTHOG_HOST=https://eu.i.posthog.com

# frontend (baked in at build time)
NEXT_PUBLIC_POSTHOG_PROJECT_TOKEN=...
NEXT_PUBLIC_POSTHOG_HOST=https://eu.i.posthog.com
```

1.3 Managed Neo4j / Qdrant (optional alternative)

By default Neo4j and Qdrant run as containers (Phase 2). If you'd rather use **Neo4j Aura** or **Qdrant Cloud**, point `NEO4J_URI` / `QDRANT_ENDPOINT` / `QDRANT_API_KEY` at the managed instances and skip starting those two containers. The rest of the guide assumes the self-hosted containers.

1.4 Create your `.env`

All subprojects load a single `.env` at the **repo root** (via `find_dotenv()`). Start from the template:

```
cp .env.example .env
```

Fill in at least these **required** values:

Variable	Required for	Notes
`NEO4J_PASSWORD`	everything	choose any password; Neo4j user is `neo4j`
`QDRANT_API_KEY`	everything	choose any key; enforced by Qdrant
`AZURE_AI_EMBEDDINGS_*`	pipeline + backend	from §1.1
`AZURE_AI_LLM_*`, `LLM_DEPLOYMENT`	backend	from §1.1
`METADATA_FOLDER`, `PROCESSED_FOLDER`, `LOG_FOLDER`	pipeline	**must be real, existing directories** (see Phase 2)

`NEO4J_URI` and `QDRANT_ENDPOINT` can stay as in the template for local use — the compose file overrides them to the docker-internal hostnames (`bolt://neo4j:7687`, `http://qdrant:6333`) for the backend container.

Phase 2 — Docker Compose & data ingestion

The pipeline writes into Neo4j and Qdrant, so start those two containers first, then run the pipeline from the host against them.

2.1 Start the databases

```
cd app
docker compose --env-file ../.env up -d neo4j qdrant
docker compose ps          # wait until neo4j is "healthy"
```

Verify:

- **Neo4j browser:** http://localhost:7474 (log in as `neo4j` / `NEO4J_PASSWORD`).
- **Qdrant:** `curl -H "api-key: $QDRANT_API_KEY" http://localhost:6334/collections`

> By default both bind to `127.0.0.1` only (`PRIVATE_SERVICE_BIND_IP`), so they're reachable from the host but not the public internet.

2.2 Prepare pipeline folders

The pipeline needs three working directories that **must exist** before the first run. Set them in `.env` and create them:

```
# in .env (paths are relative to where you run the pipeline, i.e. pipeline/)
METADATA_FOLDER=.data/metadata
PROCESSED_FOLDER=.data/processed
```

```
LOG_FOLDER=.data/logs
```

```
mkdir -p pipeline/.data/metadata pipeline/.data/processed pipeline/.data/logs
```

2.3 Get the data

Court rulings — automatic. BGE/BGer ingestion streams directly from the public HuggingFace dataset `voilaj/swiss-caselaw`. **No manual download is required** — `process` fetches and filters the rows for you (see §2.4).

Jurivoc thesaurus — bundled. The `.dmp` dumps the topic loader needs are already committed under `dataset/jurivoc/` (`jurivoc_ger.dmp`, `jurivoc_ger_fre.dmp`, `jurivoc_ger_ita.dmp`). Nothing to download.

Optional supplementary raw data. Terminology (TERMDAT) and the Eurospider ground-truth used for *evaluation* are not needed to run the system. To (re)generate them from scratch, use the scrapers under `scripts/`:

```
uv run scripts/entscheide/scraper.py            # BGer rulings -> dataset/raw/
uv run scripts/termdat/termdat_scraper.py       # TERMDAT -> termdat/entries/
uv run scripts/eurospider/eurospider_scraper.py # eval ground truth (needs
EUROSPIDER_COOKIE)
```

See `scripts/README.md` for details (Eurospider requires a logged-in cookie).

2.4 Run the ingestion pipeline (in order)

Run from `pipeline/`. **The first `process` run must pass `--reprocess-all`** — that's what stages the batches from HuggingFace; without it the command logs "Skipping … batches preparation" and does nothing.

```
cd pipeline

# 1. Rulings (auto-fetched from HuggingFace). Supported --type: BGE, BGer,
#    BGE_API, BGer_API. Repeat per language you want (de / fr / it).
uv run main.py process --type BGE  --language de --reprocess-all --batch-size 100
uv run main.py process --type BGer --language de --reprocess-all --batch-size 100

# 2. Topics: load the Jurivoc thesaurus, then link topics to rulings.
uv run main.py jurivoc-load ../dataset/jurivoc/jurivoc_ger.dmp \
  --fre-dmp ../dataset/jurivoc/jurivoc_ger_fre.dmp \
  --ita-dmp ../dataset/jurivoc/jurivoc_ger_ita.dmp
uv run main.py jurivoc-link --batch-size 500

# 3. Enrichment: law/article references, citation edges, full-text indexes.
uv run main.py law-ref-link   --batch-size 1000
uv run main.py citations-link --batch-size 1000
uv run main.py create-indexes --node Consideration --fields consideration_content

# 4. Embeddings: generate via Azure OpenAI and mirror into Qdrant (run last).
#    4a. Considerations — the main corpus -> neo4j-consideration-embeddings.
#        Slow + Azure-credit-consuming.
uv run main.py transfer-embeddings \
  --node Consideration --field consideration_content \
  --vector-collection neo4j-consideration-embeddings --batch-size 200

#    4b. Topics — Jurivoc labels -> neo4j-topic-embeddings. Labels are short,
```

```
#        so lower --min-text-length or the default (20) skips most of them.
uv run main.py transfer-embeddings \
  --node Topic --field label_de \
  --vector-collection neo4j-topic-embeddings --batch-size 200 --min-text-length 1

# 5. Qdrant indexes (run from pipeline/). Add a payload index on embedding_id
#    to the consideration collection so the backend's MatchAny filters don't
#    hit server-side timeouts. apply_qdrant_collection_config.py is optional
#    tuning (INT8 quantization + segment merge).
uv run python create_qdrant_payload_index.py
uv run python apply_qdrant_collection_config.py   # optional
```

Ordering rules: rulings (`process`) before everything else; `jurivoc-load` before `jurivoc-link`; embeddings and Qdrant indexing last (the index is created on the collection that step 4 populates). `BGE_API` / `BGer_API` are richer alternatives that pull structured sections from the OpenCaseLaw API instead of parsing raw HTML.

2.5 Verify the data

```
# Neo4j: node counts should be > 0
#   open http://localhost:7474 and run:
#   MATCH (n) RETURN labels(n)[0] AS label, count(*) ORDER BY count(*) DESC;

# Qdrant: both collections exist with 3072-dim vectors
curl -H "api-key: $QDRANT_API_KEY" \
  http://localhost:6334/collections/neo4j-consideration-embeddings
curl -H "api-key: $QDRANT_API_KEY" \
  http://localhost:6334/collections/neo4j-topic-embeddings
```

Phase 3 — Run it locally (dev)

For local development, keep the **Neo4j + Qdrant containers from Phase 2 running** and start the backend and frontend as dev servers (hot reload).

> Why not the full compose stack locally? In the stack, the backend and frontend are only `exposed` internally and reached through Caddy, which expects a real domain + TLS. So local = dev servers + the two database containers. The full stack is for Phase 4.

Point the frontend at the local backend. For dev, set in `.env` (you'll switch this back to `/api` for production):

```
NEXT_PUBLIC_API_URL=http://localhost:8000
```

Backend (FastAPI on :8000):

```
cd app/backend
uv run uvicorn main:app --reload
```

Frontend (Next.js on :3000):

```
cd app/frontend
bun install
bun run dev
```

Smoke test:

- Open http://localhost:3000 and run a query in each search mode (`keywords`, `embeddings`, `embeddings_graph`, ...).
- Or hit the API directly: `curl http://localhost:8000/config/defaults`.

Phase 4 — Production (public domain)

In production the **full compose stack** runs, fronted by **Caddy**, which terminates TLS (auto-issued via Let's Encrypt), rate-limits the API, and reverse-proxies to the frontend and backend over the internal docker network. Caddy is the only service bound to the host's public interfaces; Neo4j and Qdrant stay on `127.0.0.1`.

4.1 DNS & firewall

1. Point an `A` **record** at the server's public IP (e.g. `app.example.com`).
2. Open `80/tcp`, `443/tcp`, and `443/udp` (HTTP/3) on the firewall. ACME (certificate issuance) requires port 80.

4.2 Production `.env`

Set these in addition to the Phase 1 values:

```
DOMAIN=app.example.com
LETSENCRYPT_EMAIL=you@example.com
NEXT_PUBLIC_API_URL=/api          # same-origin: the frontend calls /api behind Caddy
```

> `NEXT_PUBLIC_*` values are **baked into the frontend at build time**. If you change `NEXT_PUBLIC_API_URL` (or the PostHog vars), rebuild the frontend image (`docker compose ... up --build`).

Optional, for reaching the private databases over a tailnet instead of only `localhost`: set `PRIVATE_SERVICE_BIND_IP` to the server's Tailscale IP and `NEO4J_ADVERTISED_ADDRESS` to its tailnet hostname.

4.3 Launch the full stack

```
cd app
docker compose --env-file ../.env up -d --build
```

Caddy issues the certificate on first request. Routing:

Public path	Goes to	Notes
`/api/*`	`backend:8000` (path stripped)	rate-limited to 30 req/min per client IP
`/api/ws`	`backend:8000/ws`	WebSocket upgrade handled by Caddy

Public path	Goes to	Notes
`/*`	`frontend:3000`	Next.js

Adjust the rate limit (`events` / `window`) in `app/caddy/Caddyfile`.

4.4 Recommended hardening — CORS

The backend currently allows all origins (`allow_origins=["*"]` in `app/backend/main.py`). Since the frontend is served same-origin behind Caddy, tighten this to your domain for production:

```
allow_origins=[f"https://{os.getenv('DOMAIN')}"]
```

4.5 Run as a service (systemd)

So the stack restarts on boot, install the provided unit (it just runs `docker compose … up --build`). The service user must be in the `docker` group.

```
# from repo root, on the server
sed -e "s|\${SERVICE_USER}|$USER|g" \
    -e "s|\${REPO_PATH}|$(pwd)|g" \
    deploy/swiss-legal-graph-rag.service \
  | sudo tee /etc/systemd/system/swiss-legal-graph-rag.service > /dev/null

sudo systemctl daemon-reload
sudo systemctl enable --now swiss-legal-graph-rag.service

# operate
sudo systemctl status  swiss-legal-graph-rag
journalctl -u swiss-legal-graph-rag.service -f
```

The first start runs `--build` and can take many minutes (the unit sets `TimeoutStartSec=0` to allow it). See `deploy/README.md`.

4.6 Reaching the private databases in production

Neo4j (`:7474`/`:7687`) and Qdrant (`:6333`/`:6334`) are bound to `127.0.0.1`. To open the Neo4j browser from your laptop, use SSH port-forwarding:

```
ssh -L 7474:localhost:7474 -L 7687:localhost:7687 user@server
# then open http://localhost:7474
```

4.7 Verify

- `https://app.example.com` serves the app with a valid certificate.
- A query runs end-to-end (the WebSocket at `/api/ws` upgrades successfully).