

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

**Znalostné grafy pre zlepšenie vyhľadávania právnych
informácií**
Diplomová práca

2026

Bc. Matej Ščišľak

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

**Znalostné grafy pre zlepšenie vyhľadávania právnych
informácií**
Diplomová práca

Študijný program: Počítačové siete (PS_Ing_D_sk)
Študijný odbor: Informatika
Školiace pracovisko: Katedra počítačových sietí (KPS)
Školiteľ: doc. Ing. Daniel Hládek, PhD.
Konzultant: doc. Ing. Daniel Hládek, PhD.

2026 Košice

Bc. Matej Ščišľák

Abstrakt v SJ

Práca sa zaoberá návrhom a implementáciou systému na vyhľadávanie právnych informácií zo slovenskej legislatívy s využitím znalostných grafov a jazykových modelov. Motiváciou pre zvolený prístup sú obmedzenia klasických systémov Retrieval-Augmented Generation, ktoré pri práci s právnymi textami nedokážu spoľahlivo zachytiť hierarchické a sémantické vzťahy medzi právnymi pojmami a ich zdrojmi. Práca navrhuje hybridný systém Graph RAG, ktorý kombinuje grafovú databázu Neo4j obsahujúcu tezaurus slovenských právnych pojmov s vektorovou databázou FAISS indexujúcou textový korpus právnych predpisov. Znalostný graf bol zostavený z dát siedmich tematických právnych tezaurů pokrývajúcich oblasti občianskeho, obchodného, trestného, finančného, ústavného, správneho a pracovného práva, pričom spolu obsahuje 2 781 právnych pojmov s ich definíciami, vzájomnými vzťahmi a odkazmi na konkrétne právne predpisy. Systém implementuje trojvrstvovú architektúru pozostávajúcu z vrstvy retrieval, vrstvy reasoning a vrstvy generácie. Vrstva retrieval vykonáva hybridné vyhľadávanie nad grafovou databázou pomocou jazyka Cypher a nad vektorovou databázou pomocou embeddingového modelu multilingual-e5-large, pričom rozhodovacia logika na základe hodnôt istoty určuje, ktorý zdroj informácií bude pre daný dopyt použitý. Vrstva reasoning zabezpečuje orchestráciu celého procesu prostredníctvom agenta, ktorý na základe výsledkov retrieval vrstvy formuluje vstup pre jazykový model Google Gemini. Systém bol vyhodnotený na súbore 110 právnych otázok pokrývajúcich rôzne oblasti slovenského práva, pričom hodnotenie vykonala právnička. Celková úspešnosť systému dosiahla 57,3 % úplne správnych odpovedí. Pri otázkach, kde systém aktivoval grafovú cestu vyhľadávania, dosiahol 65,9 % správnych odpovedí, zatiaľ čo pri otázkach spracovaných výlučne prostredníctvom vektorovej databázy klesla úspešnosť na 32,1 %. Výsledky potvrdzujú, že integrácia znalostného grafu do RAG pipeline merateľne zlepšuje kvalitu odpovedí na právne otázky a zvyšuje spoľahlivosť citovania zdrojov.

Kľúčové slová v SJ

znalostný graf, Graph RAG, právne vyhľadávanie, Neo4j, FAISS, jazykový model, sémantické vyhľadávanie, slovenská legislatíva, tezaurus, Retrieval-Augmented Generation

Abstract

This thesis deals with the design and implementation of a system for retrieving legal information from Slovak legislation using knowledge graphs and language models. The motivation for the chosen approach stems from the limitations of classical Retrieval-Augmented Generation systems, which are unable to reliably capture the hierarchical and semantic relationships between legal concepts and their sources when working with legal texts. The thesis proposes a hybrid Graph RAG system that combines a Neo4j graph database containing a thesaurus of Slovak legal terms with a FAISS vector database indexing a text corpus of legal regulations. The knowledge graph was constructed from data of seven thematic legal thesauri covering the areas of civil, commercial, criminal, financial, constitutional, administrative and labour law, containing a total of 2,781 legal concepts with their definitions, mutual relationships and references to specific legal regulations. The system implements a three-layer architecture consisting of a retrieval layer, a reasoning layer and a generation layer. The retrieval layer performs hybrid search over the graph database using the Cypher language and over the vector database using the multilingual-e5-large embedding model, with decision logic based on confidence values determining which information source is used for a given query. The reasoning layer orchestrates the entire process through an agent that formulates the input for the Google Gemini language model based on the results of the retrieval layer. The system was evaluated on a set of 110 legal questions covering various areas of Slovak law, with evaluation performed by a legal practitioner. The overall system accuracy reached 57.3% fully correct answers. For questions where the system activated the graph retrieval path, it achieved 65.9% correct answers, while for questions processed exclusively through the vector database, accuracy dropped to 32.1%. The results confirm that integrating a knowledge graph into the RAG pipeline measurably improves the quality of answers to legal questions and increases the reliability of source citation.

Keywords

knowledge graph, Graph RAG, legal information retrieval, Neo4j, FAISS, language model, semantic search, Slovak legislation, thesaurus, Retrieval-Augmented Generation

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY
Katedra elektroniky a multimediálnych telekomunikácií

ZADANIE DIPLOMOVEJ PRÁCE

Študijný odbor: **Informatika**
Študijný program: **Počítačové siete**

Názov práce:

Znalostné grafy pre zlepšenie vyhľadávania právnych informácií
Knowledge graphs for improving legal information retrieval

Študent: **Bc. Matej Ščišľak**
Školiteľ: **doc. Ing. Daniel Hládek, PhD.**
Školiace pracovisko: **Katedra elektroniky a multimediálnych telekomunikácií**
Konzultant práce:
Pracovisko konzultanta:

Pokyny na vypracovanie diplomovej práce:

1. Vypracujte prehľad súčasných prístupov k využitiu znalostných grafov na vyhľadávanie pomocou jazykových modelov.
2. Zostavte vzorový znalostný graf z právnej oblasti.
3. Navrhnite a implementujte systém, ktorý pomocou znalostných grafov umožní vyhľadávanie v rozsudkoch, zákonoch a vyhláškach, odpovedanie na otázky a sumarizáciu dokumentov.
4. Otestujte systém na navrhnutých úlohách, vyhodnoťte jeho výkonnosť a identifikujte slabé miesta.

Jazyk, v ktorom sa práca vypracuje: **slovenský**
Termín pre odovzdanie práce: **24.04.2026**
Dátum zadania diplomovej práce: **31.10.2025**



n. z. Liberios Vokorokos
prof. Ing. Liberios Vokorokos, PhD.
dekan fakulty

Čestné vyhlásenie

Vyhlasujem, že som celú záverečnú prácu vypracoval samostatne s použitím uvedenej odbornej literatúry. Vyhlasujem, že som záverečnú prácu vypracovala samostatne s použitím uvedenej odbornej literatúry, ako aj s využitím nástrojov generatívnej umelej inteligencie v súlade s Metodickým pokynom o záverečných a kvalifikačných prácach a Etickým kódexom študenta Technickej univerzity v Košiciach. Umelá inteligencia bola použitá ako podporný nástroj bez porušenia zásad akademickej etiky a autorskej integrity.

Košice, 23. apríla 2026

.....
vlastnoručný podpis

PodĎakovanie

Ďakujem svojmu školiťovi doc. Ing. Danielovi Hládkovi, PhD. za odborné vedenie, cenné pripomienky a trpezlivosť počas celého procesu spracovania diplomovej práce. Ďakujem tiež konzultantke Mgr. Kristíne Ščišťakovej za odborné právne hodnotenie výsledkov systému a za čas venovaný posúdeniu správnosti odpovedí a citovaných zdrojov, bez čoho by vyhodnotenie práce nebolo možné realizovať na odbornej úrovni. Poďakovanie patrí aj Bc. Martinovi Šariškému za zostavenie a prípravu vektorovej databázy FAISS, ktorá tvorí dôležitú súčasť implementovaného systému.

Obsah

Zoznam obrázkov	10
Zoznam tabuliek	11
Zoznam symbolov a skratiek	12
Úvod	13
1. Formulácia úlohy a cieľ práce.....	15
2. Znalostné grafy a ich využitie pri vyhľadávaní	17
2.1. Definícia a štruktúra.....	17
2.2. Vyhľadávanie v znalostných grafoch.....	18
2.2.1. Spôsoby vyhľadávania	19
2.2.2. Dopytovacie jazyky a ich význam pre vyhľadávanie	20
2.3. Grafové databázy	23
2.3.1. Natívna grafová databáza	23
2.3.2. Indexovanie a optimalizácia vyhľadávania v grafových databázach.....	26
2.4. Tezaurus ako zdroj znalostného grafu v právnej doméne.....	27
3. Moderné prístupy k sémantickému vyhľadávaniu a GraphRAG	30
3.1. Sémantické vektorové vyhľadávanie	31
3.1.1. Vyhľadávanie a indexovanie	31
3.2. Retrieval-Augmented Generation (RAG)	32
3.3. Graph RAG systémy	35
3.3.1. Čo je to Graph RAG?.....	36
3.3.2. Výhody a obmedzenia Graph RAG	37
3.3.3. Hybridný prístup.....	39
4. Návrh a implementácia systému.....	42
4.1. Cieľ práce a motivácia návrhu.....	42
4.2. Architektúra navrhnutého systému.....	43
4.3. Využívané technológie	44
4.4. Infraštruktúra a nasadenie	45

5.	Dátová príprava.....	47
5.1.	Zdroj dát.....	47
5.2.	Schéma grafovej databázy	49
5.3.	Integrácia údajov do databázy.....	50
5.4.	Vektorová databáza FAISS	51
6.	Implementácia hybridného retrievalu	53
6.1.	Predspracovanie pre grafové vyhľadávanie.....	53
6.2.	Proces grafového vyhľadávania	54
6.3.	FAISS retriever a sémantické vyhľadávanie	55
6.4.	Orchestrácia hybridného vyhľadávania	56
7.	Reasoning a generovanie odpovede	59
7.1.	Agent a jeho rozhodovacia logika	59
7.2.	Jazykový model Google Gemini	60
7.3.	Správa konverzačnej pamäte	61
8.	Vyhodnotenie systému	62
8.1.	Metodika hodnotenia	62
8.2.	Výsledky hodnotenia	64
8.3.	Diskusia výsledkov	68
	Záver.....	71
	Zoznam použitej literatúry	72
	Prílohy	76

Zoznam obrázkov

Obr. 1 Architektúra RAG systému	33
Obr. 2 Architektúra navrhnutého systému	44
Obr. 3 Overenie dostupnosti komponentov	46
Obr. 4 Ukážka záznamu dát z grafovej databázy.....	48
Obr. 5 Odpoveď systému založená výhradne na grafovom kontexte.....	56
Obr. 6 Odpoveď systému vytvorená kombináciou grafového a vektorového vyhľadávania.....	57
Obr. 7 Vyhodnotenie úspešnosti odpovedí systému v rámci celého testovacieho súboru	65
Obr. 8 Vyhodnotenie úspešnosti identifikácie právnych zdrojov v rámci celého testovacieho súboru	66
Obr. 9 Vyhodnotenie úspešnosti odpovedí systému pri využití výlučne grafovej databázy.....	67
Obr. 10 Vyhodnotenie úspešnosti odpovedí systému pri využití výlučne vektorovej databázy.....	68

Zoznam tabuliek

Tab. 1 Prehľad hodnôt istoty a výsledného režimu vyhľadávania	57
Tab. 2 Kategorizácia správnosti odpovedí systému	63
Tab. 3 Kategorizácia presnosti identifikácie právnych zdrojov	64

Zoznam symbolov a skratiek

ANN	Approximate Nearest Neighbors
API	Application Programming Interface
BFS	Breadth-First Search
CLI	Command Line Interface
DBMS	Database Management System
DFS	Depth-First Search
ETL	Extract, Transform, Load
FAISS	Facebook AI Similarity Search
HNSW	Hierarchical Navigable Small World
IVF	Inverted File Index
JSON	JavaScript Object Notation
KOS	Knowledge Organization Systems
LLM	Large Language Model
LPG	Labeled Property Graph
NLP	Natural Language Processing
OLTP	Online Transaction Processing
PQ	Product Quantization
RAG	Retrieval-Augmented Generation
RDF	Resource Description Framework
SKOS	Simple Knowledge Organization System
SPARQL	SPARQL Protocol and RDF Query Language
SSH	Secure Shell

Úvod

Vyhľadávanie právnych informácií predstavuje náročnú úlohu, a to nielen pre bežných občanov, ale aj pre odborníkov z praxe. Slovenský právny poriadok tvorí rozsiahly súbor zákonov, vyhlášok, nariadení a iných legislatívnych dokumentov, ktoré sa navzájom odkazujú, dopĺňajú a v čase menia. Orientovať sa v tomto prostredí si vyžaduje nielen znalosť konkrétnych ustanovení, ale aj pochopenie vzájomných vzťahov medzi právnymi pojmami a ich zakotvenia v príslušných predpisoch.

Tradičné metódy vyhľadávania, založené na zhode kľúčových slov, narážajú v právnej oblasti na zásadné limity. Právny jazyk je terminologicky presný a jeden pojem môže mať odlišný význam v závislosti od právneho odvetvia, v ktorom sa vyskytuje. Navyše bežný používateľ nemusí poznať presnú právnu terminológiu a formuluje svoje otázky prirodzeným jazykom, bez použitia formálnych výrazov zakotvených v zákonoch. Výsledkom je, že klasické vyhľadávanie buď nevracia relevantné výsledky, alebo ich vracia príliš veľa bez schopnosti rozlíšiť ich skutočnú relevantnosť pre daný prípad. Rozvoj veľkých jazykových modelov otvoril nové možnosti v oblasti automatického odpovedania na otázky. Tieto modely dokážu generovať súvislé a kontextovo relevantné odpovede, no ich zásadnou slabinou je závislosť na dátach použitých počas tréningu. V právnej doméne je táto slabina obzvlášť kritická – modely môžu odpovedať na základe neaktuálnych alebo nesprávnych informácií a pri generovaní odpovedí nie sú schopné spoľahlivo uviesť konkrétny právny predpis, z ktorého čerpajú.

Prístup Retrieval-Augmented Generation čiastočne tieto problémy rieši tým, že model pri formulovaní odpovede čerpá z externej databázy dokumentov. Aj tento prístup však naráža na obmedzenia, keď je potrebné pracovať s komplexnými vzťahmi medzi právnymi pojmami. Vektorové vyhľadávanie zachytáva sémantickú podobnosť textov, no nie je schopné explicitne reprezentovať hierarchické väzby medzi pojmami ani spoľahlivo priradiť každý pojem k jeho autoritatívnemu právnemu zdroju.

Odpoveďou na tieto nedostatky je integrácia znalostných grafov do procesu vyhľadávania. Znalostný graf umožňuje modelovať právne pojmy ako uzly a vzťahy medzi nimi ako hrany, pričom každý pojem je priamo naviazaný na právny predpis, z ktorého pochádza. Takto štruktúrované znalosti poskytujú jazykovému modelu presný a vysvetliteľný kontext, ktorý vektorové vyhľadávanie nedokáže plnohodnotne nahradiť.

Táto diplomová práca sa venuje návrhu a implementácii hybridného systému Graph RAG pre oblasť slovenského práva. Systém kombinuje grafovú databázu Neo4j, v ktorej je uložený tezaurus slovenských právnych pojmov zostavený zo siedmich tematických oblastí práva, s vektorovou

databázou FAISS indexujúcou textový korpus právnych predpisov. Výsledkom je nástroj schopný odpovedať na právne otázky formulované prirodzeným jazykom a ku každej odpovedi priložiť citácie konkrétnych právnych predpisov, z ktorých informácie čerpal.

Práca je štruktúrovaná nasledovne. Prvá kapitola popisuje úlohy a ciele tejto práce, druhá kapitola systematicky opisuje znalostné grafy, spôsoby vyhľadávania nad grafovou štruktúrou a moderné prístupy spájajúce grafy s jazykovými modelmi. Kapitola číslo 3 sa venuje sémantickému vyhľadávaniu, vektorovým reprezentáciám a indexovacím štruktúram. Kapitola číslo 4 popisuje prístup Graph RAG, jeho výhody v právnej doméne a hybridný spôsob kombinácie grafového a vektorového vyhľadávania. Piata kapitola analyzuje obmedzenia Graph RAG systémov. Kapitola šiesta opisuje architektúru a motiváciu navrhnutého systému, pričom kapitoly 7, 8 a 9 detailne popisujú dátovú prípravu, implementáciu hybridného retrievalu a mechanizmus generovania odpovedí. Desiata kapitola prezentuje výsledky vyhodnotenia systému na súbore 110 právnych otázok a diskutuje o silných a slabých stránkach navrhnutého riešenia.

1. Formulácia úlohy a cieľ práce

Zadanie diplomovej práce vymedzuje štyri konkrétne úlohy, ktoré táto práca rieši. Táto kapitola opisuje, čo sa pod jednotlivými bodmi rozumie a akým spôsobom boli dosiahnuté výsledky.

Prvým bodom je vypracovanie prehľadu súčasných prístupov k využitiu znalostných grafov na vyhľadávanie pomocou jazykových modelov. Táto úloha predstavuje teoretický základ celej práce. Je potrebné analyzovať celý reťazec od štruktúry grafov cez spôsoby dopytov a grafové databázy až po ich integráciu s modernými jazykovými modelmi v rámci RAG architektúr. Prehľad zahŕňa porovnanie grafového a vektorového vyhľadávania, analýzu dopytovacích jazykov SPARQL a Cypher a opis prístupu Graph RAG ako kombinácie oboch prístupov. Táto úloha je splnená v kapitolách 2 až 4.

Druhým bodom je zostavenie vzorového znalostného grafu z právnej oblasti. Pod týmto bodom sa rozumie nielen zostavenie grafovej štruktúry, ale celý proces od výberu zdrojových dát cez ich transformáciu až po naplnenie grafovej databázy. Ako zdroj dát bol použitý tezaurus slovenských právnych pojmov, konkrétne dáta zo siedmich tematických tezaurů dostupných prostredníctvom portálu Slov-Lex pokrývajúcich oblasti občianskeho, obchodného, trestného, finančného, ústavného, správneho a pracovného práva. Zostavenie grafu zahŕňalo návrh schémy s tromi typmi uzlov a tromi typmi vzťahov, implementáciu ETL procesu a import 2 781 právnych pojmov s ich definíciami a väzbami do databázy Neo4j. Táto úloha je splnená v kapitole číslo 5.

Tretím bodom je návrh a implementácia systému umožňujúceho vyhľadávanie v rozsudkoch, zákonoch a vyhláškach, odpovedanie na otázky a sumarizáciu dokumentov. Ide o jadro celej práce. Navrhnutý systém je realizovaný ako hybridný Graph RAG právny asistent, ktorý na základe otázky používateľa vyhľadáva relevantné informácie v znalostnom grafe a v textovom korpuse právnych predpisov a následne generuje odpoveď s citáciami zdrojov. Systém implementuje trojvrstvovú architektúru – retrieval vrstvu zabezpečujúcu vyhľadávanie nad Neo4j a FAISS, reasoning vrstvu s agentom orchestrujúcim celý proces a generačnú vrstvu využívajúcu jazykový model Google Gemini. Dôraz bol kladený na vysvetliteľnosť výsledkov prostredníctvom citovania konkrétnych právnych predpisov. Táto úloha je splnená v kapitolách 4 až 7.

Štvrtým bodom je testovanie systému na navrhnutých úlohách, vyhodnotenie jeho výkonnosti a identifikácia slabých miest. Pre hodnotenie bol zostavený súbor 110 otázok pokrývajúcich rôzne oblasti slovenského práva a rôzne typy právnych dopytov. Hodnotenie vykonala právnička, ktorá pre každú odpoveď nezávisle posúdila správnosť odpovede aj správnosť citovaného zdroja podľa definovaných hodnotiacich kategórií. Výsledky boli spracované a vizualizované, pričom kľúčovým zistením je merateľný prínos znalostného grafu – pri otázkach s aktívnou grafovou cestou systém

dosiahol 65,9 % správnych odpovedí oproti 32,1 % pri čisto vektorovom vyhľadávaní. Identifikované boli aj slabé miesta systému, najmä neaktuálnosť vektorového indexu a nižšia presnosť identifikácie zdrojov. Táto úloha je splnená v kapitole číslo 8.

Podmienkou riešenia bola dostupnosť slovenských právnych dát vo strojovo spracovateľnej forme, dostupnosť vhodného embeddingového modelu podporujúceho slovenčinu a možnosť nasadenia systému na výpočtovej infraštruktúre fakulty. Všetky tieto podmienky boli splnené – dáta boli získané z verejne dostupného portálu Slov-Lex, embeddingový model intfloat/multilingual-e5-large je dostupný cez platformu Hugging Face a systém bol nasadený na serveri Quadro Fakulty elektrotechniky a informatiky Technickej univerzity v Košiciach.

2. Znalostné grafy a ich využitie pri vyhľadávaní

Táto kapitola systematicky opisuje znalostné grafy ako nástroje na reprezentáciu komplexne prepojených dát a analyzuje spôsoby vyhľadávania nad ich štruktúrou. Pozornosť venuje prechodu od tradičných dopytovacích metód k moderným prístupom, ktoré integrujú grafové znalosti s veľkými jazykovými modelmi. Práve táto synergia, často označovaná ako Graph RAG, predstavuje kľúčový mechanizmus pre zvyšovanie presnosti a kontextuálneho chápania umelej inteligencie pri práci s doménovými dátami. [1]

2.1. Definícia a štruktúra

V odbornej literatúre neexistuje jediná univerzálna definícia pojmu znalostný graf. Pre potreby informatického a vyhľadávacieho kontextu je vhodná definícia, ktorú uvádzajú Hogan a kol. (2021). Tí definujú znalostný graf ako „graf dát určený na akumuláciu a sprostredkovanie znalostí o reálnom svete, kde uzly reprezentujú entity záujmu a hrany reprezentujú (potenciálne rôzne) vzťahy medzi týmito entitami“. Táto definícia je vhodná najmä preto, že pokrýva viaceré typy realizácií, od orientovaných grafov až po komplexnejšie štruktúry s označenými hranami.

Znalostný graf je typicky **viac-relačný**. Znamená to, že hrany v grafe nie sú jednotvárne. Každá hrana má svoj konkrétny typ (predikát) a často aj smer, vďaka čomu vieme presne modelovať rôzne druhy vzťahov, ako napríklad hierarchiu, synonymiu alebo definície. Z pohľadu vyhľadávania je kľúčové, že vzťahy a cesty v grafe považujeme za rovnocenný objekt dopytovania. Systém sa tak nezameriava len na hľadanie dokumentov, ale dokáže identifikovať konkrétne entity a logické súvislosti medzi nimi.

Základom každého znalostného grafu je sieť zložená z dvoch hlavných prvkov: **uzlov** (reálnych objektov, ako sú zákony, osoby či dokumenty) a **hrán**, ktoré definujú konkrétne vzťahy medzi nimi. V praxi sa na tvorbu týchto grafov využívajú dva dominantné prístupy:

1. **RDF model** – dáta ukladá do logických trojíc (subjekt – predikát – objekt)
2. **Property graph** - uzly aj hrany môžu obsahovať doplňujúce informácie vo forme vlastností (atribútov)

Hogan a kol. (2021) vo svojej analýze taktiež zdôrazňujú, že hlavnou výhodou grafov nie je len samotné uloženie dát, ale sila ich dopytovacích jazykov. Tie totiž na rozdiel od bežných databáz dokážu v grafe „navigovať“ a sledovať súvislosti po ľubovoľne dlhých cestách. Aby však graf mohol čerpať dáta z rôznych zdrojov, musí vyriešiť problém s identitou entít, známy ako **zlučovanie identít (entity alignment)**. Je to proces, ktorý zabezpečuje, aby systém pochopil, že dva rôzne záznamy z rôznych databáz opisujú tú istú reálnu vec. V praxi to zahŕňa:

- **Priradovanie unikátnych identifikátorov** - Každý právny pojem (napr. „zmluva“) má priradený jedinečný identifikátor, ktorý zabezpečuje, že sa s ním pracuje jednoznačne bez ohľadu na jeho jazykové varianty alebo synonymá.
- **Mapovanie rovnakých entít naprieč zdrojmi** - Rovnaký právny pojem môže byť uvedený v rôznych zákonoch alebo dokumentoch (napr. Občiansky zákonník a obchodné právo), pričom systém musí rozpoznať, že ide o tú istú entitu a prepojiť ju v znalostnom grafe.
- **Kontrolu kvality a konzistencie dát** - Zabezpečenie, že definícia pojmu je správne priradená ku konkrétnemu zákonu alebo paragrafu a nevznikajú konflikty medzi rôznymi zdrojmi (napr. staršia verzia zákona vs. aktuálne znenie).

Tento proces je podľa odborných prehľadov absolútne kľúčový pre úspešné spájanie dát a vytváranie jednotných znalostných systémov z viacerých nejednotných zdrojov. [1]

2.2. Vyhľadávanie v znalostných grafoch

Vyhľadávanie v prostredí znalostných grafov predstavuje zásadný posun od klasického spracovania textu k sémantickému chápaniu dát. Na rozdiel od bežných systémov, tento prístup nefunguje len na báze kľúčových slov, ale kombinuje tri základné úrovne dopytovania:

1. **Vyhľadávanie entít** – Identifikácia konkrétnych objektov (napr. konkrétna osoba, právna norma alebo organizácia), ktoré tvoria základné uzly grafu.
2. **Vyhľadávanie vzťahov** – Analýza priamych prepojení medzi entitami, čo umožňuje odpovedať na otázky typu „kto je autorom tohto dokumentu“ alebo „ktorý úrad vydal toto rozhodnutie“.
3. **Vyhľadávanie štrukturovaných vzorov a ciest** - Najkomplexnejšia forma, pri ktorej systém hľadá logické reťazce naprieč viacerými uzlami. Umožňuje to odhaliť aj nepriame súvislosti, ktoré by pri bežnom čítaní textu zostali skryté.

Rozdiel oproti vektorovému vyhľadávaniu

Princíp vektorového vyhľadávania spočíva v mapovaní dotazov a fragmentov textu do spoločného vektorového priestoru, kde sa podobnosť určuje pomocou **vzdialenosti medzi bodmi**. Súčasná metóda, známa ako dense retrieval, sa opiera o modely typu dual-encoder. Tie dokážu efektívne zakódovať dopyt aj dokument do podoby embeddingov, čo umožňuje rýchlo identifikovať a vybrať pasáže, ktoré najlepšie zodpovedajú kontextu otázky. [2]

Vektorové vyhľadávanie je síce efektívne, no naráža na limity v situáciách, kde sa vyžaduje presnosť. Keďže podobnosť vo vektorovom priestore je len aproximáciou, nedokáže plnohodnotne nahradiť:

- **presné logické obmedzenia a deterministické filtrovanie podľa typov vzťahov**
- **transparentnosť**, teda jasne vysvetliteľnú cestu medzi jednotlivými pojmami
- **zložité logické podmienky**, ktoré si vyžadujú viacero krokov uvažovania.

Okrem funkčných obmedzení existujú aj systémové výzvy. Vektorové databázové systémy (VDBMS) zápasia s neurčitou sémantickou podobnosťou, vysokými nárokmi na hardvér a potrebou efektívne kombinovať štruktúrované atribúty s vektorovými dátami v rámci jedného dopytu.

Znalostné grafy v praxi zvyčajne nenahrádzajú vektorové vyhľadávanie, ale skôr ho dopĺňajú. Grafy do systému vnášajú poriadok v podobe presných vzťahov a identifikátorov, zatiaľ čo vektory vynikajú v sémantickom porovnávaní významov a synonym. Tento hybridný prístup je základom moderných RAG systémov. Tie sa snažia prepojiť vnútorné vedomosti modelu (parametrická pamäť) s externou databázou (neparametrická pamäť), čím efektívne riešia problémy s neaktuálnymi informáciami a potrebou jasne uvádzať zdroje, z ktorých model čerpal. [1] [3] [25]

2.2.1. Spôsoby vyhľadávania

Vyhľadávanie nad znalostným grafom možno chápať ako kombináciu hľadania entít, vzťahov a štruktúrnych vzorov. Vo fulltexte je základnou jednotkou informácie dokument, v znalostnom grafe je však prirodzenou jednotkou dopytu entita a jej sémantické okolie. To zahŕňa jej priamych susedov, nadradené a podradené pojmy, definície či relevantné zdrojové dokumenty. V praxi sa na prácu s týmito údajmi využívajú dva základné režimy dopytovania:

Pattern matching

Formálnym jadrom mnohých grafových dopytov je pattern matching. V tomto režime používateľ špecifikuje „vzor“ (malý graf so špecifickou štruktúrou a typmi vzťahov) a systém následne hľadá všetky jeho výskyt v celom dátovom grafe. Tento mechanizmus je základom dopytovacích jazykov ako **SPARQL** (pre RDF grafy) alebo **Cypher** (pre property grafy.) Hogan a kol. zaraďujú tieto jazyky medzi základné stavebné prvky znalostných grafov.

Najväčšou prednosťou pattern matchingu je jeho vysoká expresivita a deklaratívna povaha. Používateľ nemusí definovať presný algoritmus vyhľadávania, stačí intuitívne opísať cieľovú štruktúru vzťahov a systém automaticky nájde aj tie najkomplexnejšie prepojenia. To umožňuje:

- **Objavovanie skrytých schém** – Identifikovať zložité sémantické reťazce, ktoré by v tabuľkových databázach vyžadovali neúmerne zložité dopyty.
- **Prácu s premennými** – Hľadať celé triedy entít a ich vzájomné interakcie na základe vopred definovaných pravidiel.

- **Presnosť a sémantickú hĺbku** - Zabezpečiť, že výsledky presne zodpovedajú logickej štruktúre, ktorú doména, v našom prípade právo, vyžaduje.

Hoci je hľadanie zhody podgrafov (subgraph matching) matematicky komplexnou úlohou, moderné grafové motory využívajú pokročilé techniky optimalizácie, indexovania a heuristického prerezávania. Vďaka tomu dokáže pattern matching v reálnom čase spracovávať rozsiahle znalostné bázy a poskytovať odpovede na otázky, ktoré presahujú možnosti klasických vyhľadávacích nástrojov. V kontexte tejto práce je práve pattern matching kľúčovým nástrojom na precízne mapovanie legislatívnych súvislostí a extrakciu relevantných znalostí. [4]

Graph traversal

Druhým spôsobom vyhľadávania je graph traversal – proces prechádzania grafu. Na rozdiel od hľadania fixného vzoru, pri tomto spôsobe dopyt explicitne alebo implicitne určuje trasu, ktorou sa má systém vydať od štartovacích uzlov po jednotlivých hranách. Využívajú sa pritom algoritmy ako **BFS** – hľadanie do šírky, alebo **DFS** – hľadanie do hĺbky.

Traversal prístup je vhodný pre lokálne úlohy, ako napríklad nájdenie všetkých definícií, zdrojov či súvisiacich konceptov v okolí danej entity. Systém sa nezameriava na celý graf ale sústredí sa na jeden **konkrétny uzol** a skúma jeho bezprostredné **susedstvo**.

Taktiež je veľmi vhodný na riešenie viackrokových otázok, ktoré by v klasickom textovom vyhľadávaní vyžadovali ťažko vysvetliteľné kombinácie filtrov. Bolo by potrebné vytvoriť viacero samostatných hľadání a dokázať ich nejako logicky spojiť. V traversal prístupe je tento proces prirodzenejší, keďže algoritmus sa pohybuje priamo po hranách grafu, ktoré spájajú súvisiace entity. Namiesto manuálneho spájania výsledkov z viacerých zdrojov systém prechádza celú cestu v rámci jedného dopytu. To umožňuje efektívne spracovať aj komplexné reťazce súvislostí (tzv. multi-hop queries), pričom výsledkom nie je len izolovaný fakt, ale celá logická cesta, ktorá k nemu viedla.

2.2.2. Dopytovacie jazyky a ich význam pre vyhľadávanie

Dopytovacie jazyky sú špecializované formálne jazyky navrhnuté na komunikáciu s databázami a informačnými systémami. Ich hlavnou úlohou je umožniť používateľovi alebo aplikácii vyberať, vkladať, upravovať alebo odstraňovať údaje.

Dopytovací jazyk SPARQL

Pre RDF grafy je dominantným jazykom SPARQL (SPARQL Protocol and RDF Query Language). Ten umožňuje formulovať dopyty nad RDF dátami prostredníctvom tzv. trojicových vzorov (triple patterns), ktoré opisujú vzťahy medzi jednotlivými entitami v grafe. SPARQL je navrhnutý tak, aby

efektívne pracoval s grafovou štruktúrou dát a umožňoval vyhľadávanie komplexných vzťahov medzi zdrojmi. Podporuje viacero typov dopytov, pričom najčastejšie sa využíva typ SELECT, ktorý umožňuje vybrať konkrétne premenné z grafu na základe definovaných vzorov. Okrem toho jazyk poskytuje aj ďalšie typy operácií, napríklad ASK na zistenie, či určitý vzťah v grafe existuje, alebo CONSTRUCT, ktorý umožňuje vytvoriť nový RDF graf na základe výsledkov dopytu. Pri formulovaní dopytov je možné používať aj rôzne filtre, ktoré obmedzujú výsledky podľa zvolených podmienok. Týmto spôsobom je možné efektívne získavať relevantné informácie z rozsiahlych grafových datasetov a pracovať s nimi v rôznych aplikáciách.

Dopytovací jazyk Cypher

Jazyk Cypher je deklaratívny dopytovací jazyk pre databázy nad **property graph** modelom, kde sa pri vyhľadávaní prirodzene kombinuje **presné vyhľadávanie** pojmov podľa vlastností uzlov, vyhľadávanie cez **reťazcové predikáty** (napr. CONTAINS) a navigácia po vzťahoch (**traversal**), ktorá umožní deterministicky prejsť od právneho pojmu k definícii a k zdrojovým zákonom. Kľúčovou výhodou Cypheru pre právnu terminológiu je, že dopyt priamo vyjadruje grafový vzor (pattern), pričom formálna sémantika jadra jazyka sa dá interpretovať ako postupná transformácia tabuľky väzieb medzi premennými a prvkami grafu. Jednoduchšie povedané, dopyt najskôr identifikuje požadované vzťahy medzi entitami v grafe a následne tieto nájdené prepojenia postupne rozširuje a filtruje, až kým nevznikne výsledná množina relevantných informácií, ktorú je možné ďalej spracovať alebo analyzovať. Jeho najznámejšou implementáciou je databázový systém **Neo4j**. Tento prístup je obzvlášť vhodný v kontexte tejto práce, kde sa využíva grafová reprezentácia právnych pojmov a ich vzťahov na efektívne vyhľadávanie relevantných informácií v rámci právneho RAG systému.

Cypher sa využíva v databázach, kde sa dáta reprezentujú ako orientovaný graf. Uzly aj hrany obsahujú vlastnosti vo forme dvojíc property–value. Uzly môžu mať zároveň priradené štítky, ktoré určujú ich typ alebo význam (napr. pojmy, definície alebo právne predpisy). Tento model je vhodný najmä pre doménové znalostné grafy, pretože umožňuje priamo ukladať vlastnosti entít a zároveň explicitne zachytávať vzťahy medzi nimi. V kontexte právnych dát to umožňuje modelovať napríklad vzťahy medzi právnymi pojmami, ich definíciami a konkrétnymi právnymi predpismi, čo je kľúčové pre presné vyhľadávanie a interpretáciu právnych informácií.

Jednou z kľúčových vlastností jazyka Cypher je aj jeho orientácia na vyhľadávanie **grafových vzorov** (pattern matching), pričom dotaz priamo opisuje štruktúru vzťahov, ktoré majú byť v grafe nájdené. Dopyty sú spracovávané postupne, pričom jednotlivé časti dopytu rozširujú alebo filtrujú priebežné výsledky, čo umožňuje flexibilnú prácu s grafovými dátami. V porovnaní s RDF modelom, ktorý

reprezentuje dáta ako trojice, property graph pracuje priamo s vlastnosťami uzlov a hrán ako základnými prvkami.

Formálne je možné jazyka Cypher interpretovať ako postupnú transformáciu tabuľky výsledkov. Každá časť dopytu pracuje s tabuľkou, ktorá obsahuje viazania premenných na konkrétne prvky grafu. Túto tabuľku následne rozširuje, filtruje alebo zjednodušuje. Výsledkom je tzv. „pipeline“ spracovanie, kde jednotlivé kroky nadväzujú jeden na druhý a postupne formujú finálny výsledok. Tento princíp je dôležitý aj z praktického hľadiska, pretože umožňuje presne sledovať, ako sa mení množina výsledkov v priebehu vyhodnocovania dotazu.

Základnou operáciou v Cypher je **definovanie grafového vzoru**, ktorý má byť v dátach nájdený. Pri jeho vyhodnocovaní sa aktuálna množina výsledkov rozšíri o všetky výskyty daného vzoru v grafe. Takýmto spôsobom je možné postupne objavovať vzťahy medzi entitami a vytvárať komplexnejšie štruktúry výsledkov.

Na tento proces nadväzuje možnosť **filtrovania**, ktorá umožňuje obmedziť výsledky len na tie, ktoré spĺňajú zadané podmienky. Filtrovanie sa vyhodnocuje nad jednotlivými riadkami výslednej tabuľky a je nevyhnutné najmä pri práci s rozsiahlymi dátami, kde je potrebné získať len relevantnú podmnožinu výsledkov. Výstup dopytu je následne definovaný projekciou, teda výberom konkrétnych hodnôt alebo výrazov, ktoré majú byť vrátené používateľovi. Tento princíp je podobný relačným databázam, kde sa tiež pracuje s výberom stĺpcov z výslednej tabuľky.

Dôležitou súčasťou jazyka je aj možnosť **deliť dopyt** na viacero krokov, pričom medzi jednotlivými fázami sa explicitne určuje, ktoré premenné budú prenesené ďalej. Tento mechanizmus umožňuje lepšie riadiť rozsah dát, znižovať veľkosť medzivýsledkov a postupne skladať zložitejšie dotazy. Špecifickou vlastnosťou Cypheru je aj schopnosť pracovať s neúplnými dátami. Pri niektorých operáciách môže dôjsť k situácii, že požadovaný vzor v grafe neexistuje. V takom prípade je možné zachovať pôvodné výsledky a chýbajúce hodnoty reprezentovať ako neznáme (null). Tento prístup je obzvlášť dôležitý v doménach, ako sú právne znalostné grafy, kde nie všetky entity musia mať kompletne prepojenia alebo dostupné všetky informácie. Celkovo Cypher poskytuje konzistentný a prehľadný model spracovania dotazov, ktorý kombinuje princípy grafového vyhľadávania s tabuľkovým spracovaním výsledkov. Tento prístup umožňuje efektívne pracovať so zložitými vzťahmi medzi dátami a zároveň zachováva intuitívnosť pri formulovaní dotazov. V rámci tejto práce je Cypher využitý ako základný nástroj na dopytovanie právneho znalostného grafu, ktorý slúži ako zdroj štruktúrovaných informácií pre Graph RAG systém pri generovaní odpovedí na právne otázky.

[5][6][7]

2.3. Grafové databázy

Grafová databáza je databázový systém určený na ukladanie a správu dát reprezentovaných vo forme grafu. Pri práci s týmito dátami zachováva základné princípy databázových systémov, ako sú nezávislosť dát, integrita či konzistencia. Zároveň poskytuje efektívne operácie zamerané na spracovanie vzťahov medzi entitami a vyhľadávanie vzorov v grafe. V praxi sa ako dominantný aplikačný model presadil **labeled property graph** (LPG). Je to orientovaný multigraf, kde uzly aj hrany môžu mať typy a množiny vlastností. Tento model, ktorý využívame aj v rámci nášho riešenia, má formálne definície a jasné mapovanie na doménové entity a vzťahy, čo je kľúčové pri návrhu vyhľadávacích a analytických funkcionalít.

Kľúčový prínos grafových databáz sa typicky objavuje v dopytoch, kde je jadrom „vzťahová“ časť problému - hľadanie ciest, viac-krokových súvislostí, vzorov a kombinácií filtrov nad uzlami a hranami. Zatiaľ čo relačné systémy historicky vynikajú v agregáciách a triedení údajov, grafové systémy sú efektívnejšie pri spracovaní úloh založených na viacnásobných spojeniach, identifikácii ciest a hľadaní štrukturálnych vzorov.

V literatúre sa tento rozdiel často vysvetľuje aj na úrovni fyzickej navigácie v dátach. Zatiaľ čo v relačnom modeli sa väzby medzi entitami rekonštruujú prostredníctvom **operácií spájania** (JOIN) nad tabuľkami, v grafovom modeli sa vzťahy realizujú priamym **prechodom po hranách**.

Pri vhodnej fyzickej reprezentácii umožňuje grafová databáza realizovať prechod medzi uzlami priamo, keďže systém disponuje explicitnými referenciami na susedné prvky grafu. V relačnom modeli je však prepojenie dát realizované prostredníctvom operácií JOIN, pri ktorých je potrebné kombinovať údaje z viacerých tabuliek na základe kľúčov a indexov. Tento proces môže byť pri väčších objemoch dát výpočtovo náročnejší, keďže vyžaduje opakované vyhľadávanie a porovnávanie záznamov. V dôsledku toho sa pri dotazoch založených na vzťahoch medzi entitami môže grafový prístup ukázať ako efektívnejší, najmä ak ide o viac-krokové prechody medzi prepojenými dátami. [7][8][9]

2.3.1. Natívna grafová databáza

Pre potreby právneho znalostného grafu je podstatné odlíšiť natívne LPG databázy od systémov, ktoré graf podporujú ako nadstavbu nad iným backendom. **Neo4j** sa v systematických prehľadoch radí do kategórie LPG-based native graph databases, teda systémov budovaných priamo pre ukladanie a spracovanie grafu.

LPG má tri charakteristiky reprezentovania uzlov. Uzol je reprezentovaný uzlovým záznamom s:

1. **Označením** - V právnom kontexte môže označenie reprezentovať napríklad právny pojem, konkrétny zákon alebo verziu tezauru.
2. **Odkazom na zoznam vlastností** - V prípade právnych zdrojov môžu vlastnosti obsahovať napríklad názov zákona, číslo paragrafu alebo citáciu.
3. **Odkazom na prvú incidentnú hranu** - Umožňuje efektívne nadviazanie na susedné uzly bez potreby rekonštrukcie vzťahov

Hrana je reprezentovaná hranovým záznamom s typom, odkazmi na vlastnosti, odkazmi na susedné uzly a napojením do zoznamov susednosti.

Kľúčovým optimalizačným princípom databázy Neo4j je využitie priamych ukazovateľov na fyzické umiestnenie susedných uzlov a hrán. V dôsledku toho je pri traversal dopytoch po identifikovaní štartovacieho uzla možné prechádzať graf bez potreby opakovaného vyhľadávania cez indexy. Asymptotika potom závisí skôr od veľkosti navštíveného podgrafu než od globálnej veľkosti databázy. Tento prístup je výhodný aj v kontexte tejto práce, kde sa nad právnym znalostným grafom vykonávajú traversal dopyty s cieľom identifikovať relevantné definície pojmov a ich prepojenie na právne zdroje v rámci RAG systému.

Z hľadiska transakčného správania a konzistencie je pre aplikačnú doménu práva podstatné, že operácie nad grafom prebiehajú transakčne a systém používa mechanizmy izolácie. V akademických zdrojoch sa uvádza, že operácie prístupu ku grafu v Neo4j sú vykonávané v transakciách. Je predvolená izolácia na úrovni **read-committed**, pričom vyššie izolácie sú predmetom výskumu a rozšírení. Súčasne sa v taxonómiách grafových databáz explicitne diskutuje ACID ako referenčný rámec a uvádza sa, že Neo4j používa write-locky na ochranu modifikácií až do commitu, čo je typická implementačná stratégia pre OLTP transakcie v DBMS. Z pohľadu tejto práce je dôležité, že systém poskytuje konzistentné a transakčne bezpečné spracovanie dát, čo je v právnej doméne nevyhnutné z hľadiska spoľahlivosti a auditovateľnosti. [8][10]

V právnej doméne je vhodné modelovať pojmy a ich vzťahy už od začiatku tak, aby zodpovedali očakávaným spôsobom vyhľadávania. Ide najmä o:

- rýchlu identifikáciu pojmu podľa názvu
- navigáciu medzi príbuznými pojmi
- vysvetliteľné prepojenie pojmu na autoritatívny zdroj (napr. zákon)

Model typu LPG je na tento účel vhodný, pretože pracuje s uzlami a hranami ako s prvotriednymi objektmi, ktoré môžu obsahovať vlastnosti. Je možné ho rozdeliť na tri základné typy uzlov a tri základné typy vzťahov.

Typy uzlov

Medzi typy uzlov patrí najmä uzol **Term**, ktorý reprezentuje právny pojem. Tento uzol by mal obsahovať jedinečný identifikátor, preferovaný názov, prípadné synonymá. Taktiež môže obsahovať alternatívne označenia a tiež normalizované alebo lemmatizované formy názvu pre účely vyhľadávania. Okrem toho je vhodné uchovávať aj informácie vysvetľujúce význam pojmu, napríklad stručnú definíciu alebo poznámku. Takto navrhnutý uzol umožňuje kombinovať presné aj tolerantné vyhľadávanie. V rámci tejto práce je tento uzol kľúčový pre identifikáciu vstupného pojmu zo strany používateľa a jeho následné spracovanie v RAG pipeline.

Druhým typom je uzol **Source**, ktorý reprezentuje právny zdroj, napríklad zákon alebo iný dokument. Tento uzol by mal obsahovať identifikátor zdroja, jeho názov, citáciu a prípadne aj detailnejšie informácie, ako je konkrétna časť alebo paragraf. Vďaka tomu je možné výsledky vyhľadávania nielen zobrazit, ale aj jednoznačne odkázať na ich právny základ.

Tretím typom je uzol **Thesaurus**, ktorý reprezentuje konkrétnu verziu tezauru. Obsahuje identifikačné a metadátové informácie, ako napríklad názov, verziu alebo dátum publikovania. Tento prvok je dôležitý najmä z pohľadu správy dát, keďže umožňuje pracovať s viacerými verziami a sledovať ich vývoj v čase.

Typy vzťahov

Z hľadiska vzťahov medzi uzlami je kľúčový vzťah **DEFINED_BY**, ktorý prepája právny pojem so zdrojom, z ktorého jeho definícia pochádza. Tento vzťah zabezpečuje vysvetliteľnosť výsledkov, keďže umožňuje jednoznačne určiť, na základe akého dokumentu je pojem definovaný.

Ďalším dôležitým vzťahom je **RELATED_TO**, ktorý prepája príbuzné pojmy. Tento typ vzťahu umožňuje vytvárať sieť súvisiacich termínov, čo je užitočné napríklad pri rozširovaní dotazov alebo pri zobrazovaní kontextu pojmu.

Posledným typom je vzťah **FROM_THESAURUS**, ktorý prepája pojem s konkrétnou verziou tezauru. Tento vzťah slúži na evidenciu pôvodu dát a umožňuje pracovať s rôznymi verziami znalostného grafu. Výhodou modelu property graph je aj možnosť ukladať vlastnosti priamo na vzťahy. V praxi to znamená, že ku každému prepojeniu je možné priradiť doplňujúce informácie, napríklad mieru spoľahlivosti, zdroj extrakcie alebo čas vytvorenia. Tieto metadáta zvyšujú transparentnosť a auditovateľnosť systému, čo je v právnej oblasti obzvlášť dôležité. V rámci tejto práce takto navrhnutý model predstavuje základnú dátovú vrstvu, ktorá umožňuje efektívne vyhľadávanie, prepojenie a interpretáciu právnych informácií v spojení s generatívnym modelom. [7][8][11]

2.3.2. Indexovanie a optimalizácia vyhľadávania v grafových databázach

Výkon vyhľadávania v grafových databázach je v praxi výrazne ovplyvnený tým, ako rýchlo je možné identifikovať počiatočné uzly, z ktorých sa následne vykonáva traversal. Samotná navigácia v grafe je zvyčajne efektívna, pretože prebieha nad lokálnym podgrafom. Kritickým krokom preto zostáva fáza výberu vhodných kandidátov. Z tohto dôvodu sa v natívnych grafových databázach často kombinuje indexovo podporované vyhľadávanie s následnou navigáciou po vzťahoch. [8]

Stromové indexy pre identifikátory a selektívne podmienky

Stromové indexy typu **B-tree** alebo **B+tree** patria medzi štandardné mechanizmy databázových systémov na urýchlenie vyhľadávania podľa kľúčov a rozsahových podmienok. Ich efektivita vyplýva najmä z vyváženej štruktúry a optimalizácie pre prácu s externou pamäťou. V kontexte právneho znalostného grafu je vhodné aplikovať indexovanie najmä nad jednoznačnými identifikátormi, ako sú napríklad identifikátory pojmov a zdrojov. Význam má aj indexovanie názvov pojmov, ktoré umožňuje rýchle presné vyhľadávanie.

Využitie indexov nad vlastnosťami a typmi uzlov výrazne znižuje potrebu prechádzať celý graf. Výsledkom je rýchlejšia selekcia kandidátov, čo následne redukuje množstvo operácií potrebných v traversal fáze. V prípade databázy Neo4j sú indexy implementované nad labelmi a vlastnosťami uzlov, pričom systém udržiava mapovanie na množiny uzlov s danými charakteristikami. Indexy môžu byť vytvárané aj nad vlastnosťami vzťahov, čo ďalej rozširuje možnosti optimalizácie dotazov. Empirické pozorovania ukazujú, že najväčší prínos indexovania spočíva práve **v urýchlení výberu počiatočných uzlov**, od ktorých sa následne pokračuje v grafe.

Stromové indexy zároveň umožňujú efektívne spracovanie rozsahových dotazov a triedenie výsledkov bez potreby dodatočných operácií nad dátami. Vďaka logaritmickému zložitosti vyhľadávania poskytujú stabilný výkon aj pri veľkých objemoch dát, čo je dôležité najmä pri práci s rozsiahlymi právnymi databázami. Ďalšou výhodou je podpora unikátnych obmedzení, ktoré zabezpečujú konzistenciu dát, napríklad pri identifikátoroch pojmov alebo zdrojov. Z pohľadu implementácie v právnom znalostnom grafe tieto vlastnosti umožňujú rýchlu a spoľahlivú identifikáciu vstupných entít, ktoré následne slúžia ako východiskový bod pre ďalšie spracovanie. Stromové indexy tak predstavujú kľúčový prvok pre **optimalizáciu prvej fázy** dopytu, na ktorú nadväzuje samotná grafová navigácia a sémantické spracovanie výsledkov. [12][13]

Full-text indexy pre prácu s textovými dátami

Vyhľadávanie v právnej doméne je vo veľkej miere založené na textových údajoch, ako sú názvy pojmov, synonymá, definície alebo poznámky. Na tento účel sa využívajú full-text indexy, ktoré sú

typicky implementované pomocou invertovaného indexu. Ten mapuje jednotlivé termy (tokeny) na dokumenty alebo objekty, v ktorých sa vyskytujú. Súčasťou tohto procesu býva aj textová analýza, ktorá zahŕňa normalizáciu znakov, tokenizáciu, odstránenie stop-slov a prípadne aj stemming alebo lemmatizáciu.

Pri prepojení full-text vyhľadávania s grafovými databázami sa uplatňujú dva základné prístupy. Prvým je využitie interného full-text indexu priamo v rámci databázového systému. Druhým prístupom je integrácia externých nástrojov určených na vyhľadávanie, ktoré poskytujú pokročilé možnosti práce s textom a multiatribútovými dotazmi. V praxi sa často využívajú riešenia založené na externých systémoch, ktoré dopĺňajú grafovú databázu o efektívne textové vyhľadávanie.

V kontexte právneho znalostného grafu plní full-text index najmä úlohu generovania kandidátov. Na základe textovej zhody identifikuje množinu relevantných pojmov, ktoré následne slúžia ako vstup pre grafové spracovanie. V druhej fáze dochádza k sémantickému spresneniu výsledkov prostredníctvom navigácie v grafe, napríklad rozšírením cez vzťahy medzi pojmami alebo filtráciou podľa konkrétnych zdrojov či verzií tezaurov. Tento dvojfázový prístup, kombinujúci textové vyhľadávanie s grafovou navigáciou, umožňuje efektívne využívať výhody oboch prístupov a predstavuje základ pre moderné systémy vyhľadávania v právnej oblasti. [8][14]

2.4. Tezaurus ako zdroj znalostného grafu v právnej doméne

Tezaurus v právnej oblasti predstavuje kontrolovanú slovnú zásobu, ktorá obsahuje štandardizované názvy pojmov a explicitne definované vzťahy medzi nimi, či už hierarchické alebo asociatívne. Jeho hlavným cieľom je zlepšiť vyhľadávanie informácií tým, že zavádza riadenú synonymiu, podporuje rozširovanie dopytov a zároveň umožňuje prepojenie pojmu na jeho definíciu a relevantný právny zdroj. V praxi sa tezaurus často publikuje v štandardoch ako **ISO 25964** a **SKOS**, ktoré umožňujú transformáciu z ľudsky čitateľných formátov do strojovo spracovateľnej podoby, napríklad vo forme property graph alebo RDF databáz.

Tezaurus patrí medzi systémy organizácie znalostí (Knowledge Organization Systems – KOS), ktoré modelujú koncepty a ich jazykové reprezentácie. Dôležitým aspektom je rozlíšenie medzi konceptom a jeho názvom, pričom jeden koncept môže mať viacero označení. Štandard ISO 25964 poskytuje dátový model pre tezaury využiteľné v informačnom vyhľadávaní, tak ako v našom prípade. Definuje základné vzťahy medzi pojmami, ako sú synonymia, hierarchia a asociácie medzi pojmami. V praxi tezaurus obsahuje najmä:

- Pojmy
- Preferované názvy

- Alternatívne názvy
- Hierarchické vzťahy
- Asociatívne vzťahy

V slovenskom prostredí predstavuje významný zdroj právnych dát portál Slov-Lex.sk, ktorý obsahuje legislatívne dokumenty vrátane tezauru právnych pojmov. Tento tezaurus možno považovať za primárny dataset vhodný na budovanie znalostného grafu. Z akademického hľadiska existujú práce, ktoré využívajú tieto dáta napríklad pri automatizovanom budovaní terminologických tezaurí alebo pri spracovaní legislatívnych korpusov. Takéto prístupy ukazujú, že tezaurus nie je len zoznam pojmov, ale aj praktický zdroj pre vyhľadávanie, analýzu textov a NLP úlohy. Dôležitou súčasťou spracovania dát zo Slov-Lexu je aj čistenie, deduplikácia a prepojenie s konkrétnymi právnymi dokumentmi, čo umožňuje vytvárať komplexný znalostný graf.

Transformácia tezauru z publikačného formátu (napr. PDF) do znalostného grafu prebieha ako ETL proces. Zahŕňa to extrakciu dát, ich normalizáciu, štruktúrovanie a následné uloženie do cieľového formátu. Pri tomto procese je kľúčové priradiť každému pojmu stabilný identifikátor a zachovať jeho väzby na zdrojové informácie. Následne sa spracovávajú synonymá a rôzne jazykové varianty. Dôležitou súčasťou je aj identifikácia vzťahov medzi pojmami, ako sú hierarchické alebo asociatívne väzby. Okrem toho je potrebné zachytiť aj definície a ich legitímny kontext. V kontexte tejto práce tento proces umožňuje transformáciu tezauru zo Slov-Lexu do grafovej databázy, ktorá je následne využívaná v rámci RAG architektúr

Tezaurus predstavuje vhodný základ pre právne vyhľadávanie najmä vďaka explicitným vzťahom medzi pojmami. To umožňuje rozširovanie dopytov a presnejšie mapovanie používateľského vstupu na konkrétne koncepty. V právnej doméne je zároveň dôležitá možnosť prepojenia pojmov na ich zdroje, čo zabezpečuje vysvetliteľnosť výsledkov. Modelovanie tezauru umožňuje aj evidenciu verzií a sledovanie zmien v čase, čo je dôležité vzhľadom na dynamiku legislatívy. Skúsenosti ukazujú, že kombinácia tezauru a vyhľadávacích techník vedie k zlepšeniu kvality výsledkov a umožňuje transparentne pracovať s významom pojmov.

Obmedzenia

Používanie tezauru v právnej oblasti prináša okrem výhod aj viaceré obmedzenia, ktoré je potrebné zohľadniť pri návrhu a implementácii systému. Jedným z hlavných problémov je **neúplnosť pokrytia** terminológie, keďže tezaurus spravidla neobsahuje všetky možné varianty pojmov, najmä neformálne alebo novovznikajúce výrazy používané v praxi. Ďalším významným obmedzením je **viacvýznamovosť**, kde jeden termín môže reprezentovať viacero rôznych konceptov v závislosti od kontextu. Bez dostatočného kontextového spracovania môže dochádzať k nesprávnemu

mapovaniu používateľských dotazov na nesprávne pojmy. S tým úzko súvisí aj problém synonymie a variabilných jazykových foriem, kde rôzne výrazy môžu označovať rovnaký koncept, no nemusia byť všetky zachytené v tezaure.

Významnou výzvou je aj dynamika právnej domény, kde dochádza k častým zmenám legislatívy, terminológie a definícií. Tezaurus preto vyžaduje pravidelnú aktualizáciu a údržbu, aby zostal aktuálny a relevantný. Nedostatočne aktualizovaný tezaurus môže viesť k zastaraným alebo nepresným výsledkom vyhľadávania. Ďalším obmedzením je závislosť na kvalite zdrojových dát, najmä pri automatizovanej extrakcii z dokumentov. Chyby v extrakcii, nejednoznačná štruktúra zdrojov alebo nedostatočne konzistentné formátovanie môžu negatívne ovplyvniť kvalitu výsledného znalostného grafu. Z pohľadu modelovania môže byť problematická aj obmedzená expresivita tezauru, ktorý typicky zachytáva len základné typy vzťahov (hierarchické a asociatívne), no nemusí dostatočne reprezentovať komplexnejšie právne väzby alebo logické závislosti medzi pojmami. To môže limitovať možnosti pokročilejšej analýzy alebo inferencie. Pri využití tezauru vo vyhľadávacích systémoch sa objavuje aj problém závislosti na presnosti mapovania dopytu na koncept, ktorý môže byť ovplyvnený kvalitou spracovania prirodzeného jazyka. V prípade jazykov s bohatou morfológiou, ako je slovenčina, je nevyhnutné riešiť normalizáciu a lemmatizáciu, inak môže dochádzať k zníženiu úspešnosti vyhľadávania.

Napokon je potrebné spomenúť aj výpočtové a implementačné obmedzenia, najmä pri práci s rozsiahlymi grafmi. S rastúcim objemom dát rastie náročnosť na výpočtové zdroje, optimalizáciu dopytov a správu indexov, čo môže ovplyvniť odozvu systému a jeho škálovateľnosť. Celkovo je preto nevyhnutné pristupovať k tezauru ako k dynamickému a postupne rozvíjanému systému, ktorý si vyžaduje priebežnú údržbu, validáciu a dopĺňanie, ideálne v kombinácii automatizovaných metód a expertnej kontroly. [15][16][17]

3. Moderné prístupy k sémantickému vyhľadávaniu a GraphRAG

Táto kapitola opisuje technologické základy, na ktorých stojí navrhnuté riešenie. Najprv sú predstavené veľké jazykové modely a prístupy k sémantickému vyhľadávaniu, následne princíp RAG a jeho obmedzenia v právnej doméne. Na základe týchto obmedzení je v poslednej časti kapitoly opísaný prístup Graph RAG ako ich prirodzené rozšírenie.

Sémantické vyhľadávanie predstavuje moderný prístup k práci s textovými dátami, ktorý sa zameriava na porozumenie významu dopytu a dokumentov, nie len na ich presnú slovnú zhodu. Na rozdiel od tradičných metód založených na kľúčových slovách využíva reprezentáciu textu vo forme vektorov, ktoré zachytávajú jeho významové vlastnosti. Rozvoj týchto prístupov úzko súvisí s pokrokom v oblasti neurónových jazykových modelov, ktoré umožňujú efektívne vytváranie takýchto reprezentácií a ich následné využitie pri vyhľadávaní.

Veľké jazykové modely predstavujú pokročilé neurónové siete založené na architektúre transformer, ktoré sú trénované na rozsiahlych textových dátach. Vďaka tomu sú schopné generovať súvislý a kontextovo relevantný text na základe vstupného zadania. Medzi významné modely patria napríklad GPT, T5, ako aj moderné multimodálne modely, napríklad Google Gemini, ktoré dokážu spracovávať nielen text, ale aj obraz, audio či iné typy dát. Počas generovania model spracováva vstupný kontext a postupne predikuje nasledujúce tokeny, čím vytvára odpoveď. Tento proces je založený na pravdepodobnostnom modelovaní jazyka, kde model vyberá najpravdepodobnejšie pokračovanie textu vzhľadom na predchádzajúci kontext.

Dôležitým parametrom LLM je veľkosť kontextového okna, ktorá určuje množstvo informácií, ktoré môže model zohľadniť naraz. Zatiaľ čo staršie modely pracovali s kontextom na úrovni niekoľkých tisíc tokenov, moderné modely dosahujú výrazne väčšie rozsahy. Súčasné systémy umožňujú pracovať s kontextom v desiatkach až stovkách tisíc tokenov, pričom najnovšie modely, vrátane pokročilých verzií modelu Google Gemini, podporujú kontextové okná na úrovni stoviek tisíc až miliónov tokenov. To umožňuje spracovanie rozsiahlych dokumentov, ako sú celé právne predpisy alebo kolekcie legislatívnych textov v rámci jedného dopytu. Modely Gemini sú zároveň navrhnuté ako natívne multimodálne systémy, čo znamená, že dokážu pracovať s viacerými typmi vstupov bez potreby samostatných špecializovaných modelov.

Rozšírenie kontextového okna významne zlepšuje schopnosť modelov pracovať s dlhšími dokumentmi, analyzovať komplexné informácie a využívať externé zdroje dát. To je obzvlášť dôležité v doménach, ako je právo, kde je potrebné pracovať s rozsiahlymi textami a presne interpretovať ich obsah. Zároveň však aj moderné LLM narážajú na obmedzenia, najmä pokiaľ ide o presnosť faktických informácií a potrebu doplnenia externými zdrojmi. Práve preto sa v súčasnosti

často využíva prístup Retrieval-Augmented Generation (RAG), ktorý kombinuje schopnosti jazykových modelov s externými databázami alebo znalostnými grafmi, čím sa zvyšuje presnosť a vysvetliteľnosť generovaných odpovedí. [18][19]

3.1. Sémantické vektorové vyhľadávanie

Tradičné vyhľadávanie textu často používa kľúčové slová a metódy ako Boolean matching či BM25, ktoré hľadajú presnú zhodu slov. Sémantické vyhľadávanie však reprezentuje texty a dopyty pomocou **číselných vektorov** - embeddingov, ktoré zachytávajú význam obsahu. Technicky sa vyhľadávanie realizuje pomocou metód approximate nearest neighbors (ANN). Dopyt a dokumenty sa najskôr zakódujú do vektorov pomocou modelov ako napríklad BERT alebo špecializovaných Siamesian sietí (SBERT). Pre efektívne vyhľadávanie v miliardových korpusoch sa používajú indexy ako **FAISS (Facebook AI Similarity Search)**. Výber vhodného embedding modelu (napr. multijazyčného **multilingual-e5-large**, LaBSE, či SBERT) je kľúčový pre slovenskú legislatívnu doménu. Systém zároveň uchováva metadáta, napríklad identifikáciu zákona alebo paragrafu, ktoré slúžia na citovanie zdrojov.

Najčastejšie používané metriky pre mieru podobnosti medzi vektormi sú **cosine similarity** a skalárny súčin. Cosine similarity sa počíta ako kosínus uhla medzi dvoma vektormi a vychádza zo skóre -1 až 1 (pre normalizované vektory od 0 do 1). Vyššie skóre znamená väčšiu podobnosť. Skalárny súčin sa tiež často využíva, obzvlášť keď sú číselné vektory pred tréningom normalizované. Tieto metriky sa dajú efektívne implementovať vo vektorových indexoch. Dôležitou vlastnosťou je, že sú diferencovateľné, čo umožňuje trénovať modely tak, aby priamo maximalizovali relevantnú metriku. Výber konkrétnej metriky môže ovplyvniť výsledky. [20][21][22][23]

3.1.1. Vyhľadávanie a indexovanie

Pri vyhľadávaní sa zvyčajne snažíme nájsť top-k dokumentov (napr. $k=10$ alebo 100) s najvyššou podobnosťou k dopytu. K-najbližších susedov (k-NN) algoritmus v jeho presnej forme však pri veľkých dátach nie je praktický kvôli vysokým nákladom. Každý dopyt vyžaduje výpočet podobnosti so všetkými n dokumentmi. Preto sa používa **approximate nearest neighbors (ANN)** prístup, ktorý nájde približných susedov s výrazne nižšou časovou náročnosťou. Populárne riešenia sú založené na dátových štruktúrach ako stromové indexy či grafy. Napríklad HNSW (Hierarchical Navigable Small World) organizuje dáta do viacúrovňového grafu, čím umožňuje rýchly prístup s logaritmickou zložitnosťou. Product Quantization (PQ) s IVF (Inverted File) rozdeľuje priestor na časti a komprimuje vektory, čo dramaticky znižuje pamäťovú náročnosť a zrýchľuje vyhľadávanie (Johnson et al., 2017). Algoritmus HNSW dokáže dokonca dosahovať vysoký recall takmer ako presný k-NN, s výrazne menším počtom prepočtov

FAISS je štandardná knižnica pre vyhľadávanie veľkých vektorových súborov. Umožňuje kombinovať rôzne indexovacie schémy. Využíva paralelizmus (CPU aj GPU) pre rýchle vyhľadávanie a dokáže indexovať stovky miliónov vektorov. Prakticky to znamená, že top-k dopyty (napr. Vyhľadaj 10 najpodobnejších dokumentov) sú vrátené v reálnom čase aj pri masívnej databáze. Na zníženie latencie a spotreby pamäte sa v FAISS využíva aj kvantizácia (int8, float16), ktorá predstavuje kompromis medzi presnosťou a výkonom.

Pre veľmi veľké zbierky dokumentov je možné škálovať indexy. FAISS podporuje viacero možností:

- **GPU akcelerácia:** Urýchlenie vyhľadávania pomocou grafických kariet, čo zvládne milióny dotazov za sekundu.
- **Sharding:** Rozdelenie indexu na viaceré pod-indexy bežiacie paralelne (napr. každý GPU alebo server vlastní časť dát).
- **Disk-based indexy:** Pre prípady, keď vektory nevojdú do RAM/GPU pamäte, FAISS umožňuje udržať časť dát na disku s vyrovnávacou pamäťou v pamäti (tzv. IVF + on-disk PQ).
- **Parametrizácia ANN:** HNSW má parametre, ktoré priamo ovplyvňujú pamäť a recall. Tieto parametre treba prispôbiť veľkosti korpusu a požadovanej presnosti.

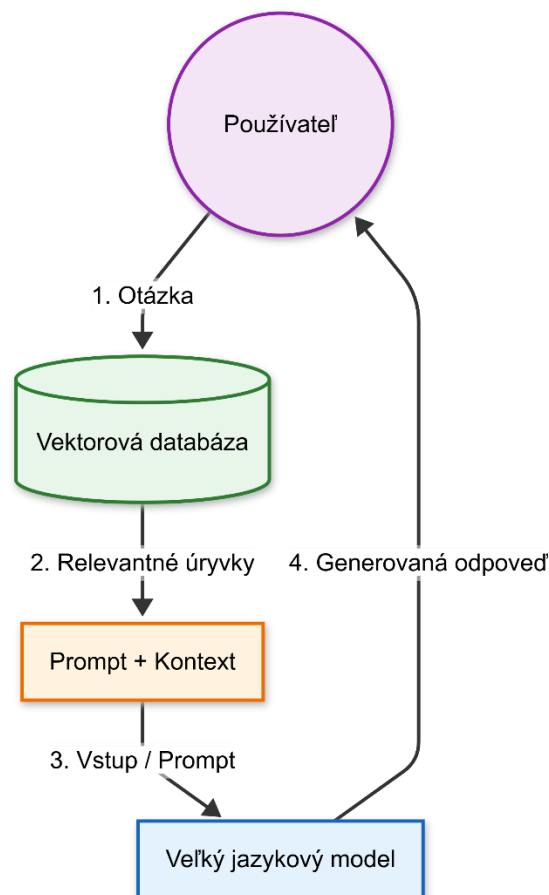
Napriek tomu, že sémantické vektorové vyhľadávanie umožňuje efektívne identifikovať relevantné dokumenty na základe významovej podobnosti, samo o sebe neposkytuje finálnu odpoveď na používateľský dopyt. Výstupom vyhľadávania je spravidla množina textových úryvkov, ktoré je potrebné ďalej interpretovať a spracovať. V praxi to znamená, že systém musí nielen nájsť relevantné informácie, ale aj ich zrozumiteľne syntetizovať do ucelenej odpovede. Práve tento nedostatok viedol k vzniku prístupu Retrieval-Augmented Generation (RAG), ktorý kombinuje vyhľadávanie s generovaním textu pomocou jazykových modelov. [21][24]

3.2. Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) predstavuje prístup, ktorý kombinuje vyhľadávanie informácií s generovaním textu pomocou veľkých jazykových modelov. Na rozdiel od klasických jazykových modelov, ktoré generujú odpovede len na základe znalostí naučených počas tréningu, RAG využíva aj externé zdroje dát. Vďaka tomu je možné generovať odpovede, ktoré sú aktuálnejšie, presnejšie a zároveň podložené konkrétnymi informáciami. Základná myšlienka RAG spočíva v tom, že pred samotným generovaním odpovede systém najprv vyhľadá relevantné informácie v externom úložisku, napríklad v databáze dokumentov. Tieto informácie sú následne použité ako kontext pre jazykový model, ktorý na ich základe vytvorí odpoveď. Takýto postup výrazne znižuje riziko tzv. halucinácií, teda generovania nepresných alebo vymyslených informácií,

a zároveň umožňuje lepšie vysvetliť pôvod odpovede prostredníctvom citácií alebo odkazov na zdroje.

Proces fungovania RAG systému možno rozdeliť do niekoľkých fáz. Najprv používateľ zadá otázku, ktorú spracuje vyhľadávací mechanizmus. Ten identifikuje najrelevantnejšie dokumenty alebo ich časti pomocou rôznych techník, ako sú fulltextové vyhľadávanie, vektorové reprezentácie alebo ich kombinácia. Následne sa získané výsledky vyhodnotia a zoradia podľa miery ich relevancie. V ďalšom kroku sa tieto informácie spoja s pôvodným dopytom a vytvoria vstup pre jazykový model. Model potom generuje odpoveď, ktorá vychádza z poskytnutého kontextu. Výsledok môže byť následne upravený, napríklad doplnený o citácie, štruktúrovaný výstup alebo validáciu voči zdrojovým dátam.



Obr. 1 Architektúra RAG systému

RAG sa v súčasnosti využíva v rôznych oblastiach, najmä tam, kde je potrebná práca s veľkým množstvom špecifických alebo často sa meniacich informácií. Typickými príkladmi sú vyhľadávacie systémy, chatboty, analytické nástroje alebo systémy na podporu rozhodovania. V právnej doméne má tento prístup mimoriadny význam, pretože umožňuje prepájať generovanie odpovedí s konkrétnymi právnymi zdrojmi, ako sú zákony, paragrafy alebo definície pojmov. V kontexte tejto

práce RAG predstavuje kľúčový mechanizmus, ktorý prepája grafovú databázu obsahujúcu právne pojmy a ich vzťahy s jazykovým modelom. Vyhľadávacia časť systému identifikuje relevantné informácie zo znalostného grafu a tieto údaje sú následne použité ako vstup pre generovanie odpovede. Takýto prístup umožňuje dosiahnuť vyššiu presnosť, lepšiu vysvetliteľnosť a praktickú využiteľnosť systému pri práci s právnymi informáciami.

Výhody RAG systému

Jednou z hlavných výhod prístupu Retrieval-Augmented Generation je jeho schopnosť pracovať s aktuálnymi informáciami. Na rozdiel od klasických jazykových modelov, ktoré sú odkázané na dáta použité počas tréningu, RAG využíva externé zdroje, ktoré je možné priebežne aktualizovať. To znamená, že pri zmene alebo doplnení informácií nie je potrebné model znovu trénovať, ale postačuje aktualizovať zdrojové dáta alebo index. Ďalšou významnou výhodou je redukcia halucinácií. Keďže generovanie odpovede je podmienené relevantnými dokumentmi získanými vo fáze vyhľadávania, model je viac viazaný na reálne existujúce informácie. V dôsledku toho dochádza k zníženiu výskytu nepresných alebo vymyslených tvrdení, ktoré sú typické pre samostatné jazykové modely bez prístupu k externým dátam. RAG zároveň umožňuje lepšiu vysvetliteľnosť výsledkov. Odpovede môžu byť priamo prepojené na konkrétne zdroje, z ktorých vychádzajú, čo umožňuje ich overenie a zvyšuje dôveryhodnosť systému. Táto vlastnosť je obzvlášť dôležitá v oblastiach, kde je potrebná transparentnosť a možnosť citovania, napríklad v právnej doméne. Významným prínosom je aj zvýšenie presnosti pri riešení faktografických otázok. Kombinácia vyhľadávania relevantných informácií a následnej generácie odpovede umožňuje dosahovať lepšie výsledky v porovnaní s modelmi, ktoré sa spoliehajú výlučne na interné znalosti. [25][26]

Obmedzenia RAG systému v právnej oblasti

V právnom kontexte sa pri použití čistých RAG systémov objavuje viacero zásadných obmedzení, ktoré vyplývajú zo špecifickej povahy právnych informácií. Jedným z hlavných problémov je neschopnosť zachytiť komplexné hierarchické a **sémantické vzťahy** medzi právnymi dokumentmi. Právne predpisy majú prirodzene viacúrovňovú štruktúru, kde ústava ovplyvňuje zákony, zákony obsahujú paragrafy a súdne rozhodnutia odkazujú na predchádzajúce prípady. Štandardný RAG prístup, založený na rozdelení dokumentov na menšie textové úseky, túto hierarchiu často nereflektuje, čo vedie k strate dôležitých väzieb medzi jednotlivými časťami dokumentu. Výsledkom je, že model nemusí správne pochopiť kontext, v ktorom sa konkrétny pojem alebo ustanovenie nachádza.

Ďalším významným obmedzením je **fragmentácia informácií**. Právne odpovede často vyžadujú syntézu viacerých častí textu, napríklad kombináciu konkrétneho ustanovenia so súvisiacou

definíciou alebo výnimkou uvedenou inde v dokumente. Čistý RAG však pracuje primárne s lokálne relevantnými úryvkami, bez explicitného mechanizmu na ich prepojenie do uceleného výkladu. Ak sa medzi vyhľadanými pasážami nenachádza kľúčová definícia, jazykový model ju môže doplniť na základe vlastných znalostí, čo zvyšuje riziko nepresností alebo halucinácií. To je v právnej oblasti kritický problém.

Významným problémom je aj otázka **citovateľnosti**. V právnej praxi je nevyhnutné presne identifikovať zdroj informácie, napríklad konkrétny zákon, paragraf alebo súdne rozhodnutie. RAG síce využíva externé dokumenty, no negarantuje, že vybrané zdroje sú skutočne správne alebo najrelevantnejšie z právneho hľadiska. V prípade nesprávneho výberu dokumentu môže byť výsledná odpoveď formálne správna, no bez dostatočnej dôkazovej opory.

Aj napriek tomu, že RAG prístup výrazne znižuje **výskyt halucinácií v** porovnaní s čistými jazykovými modelmi, tento problém úplne neeliminuje. Presnosť odpovede je totiž stále závislá od kvality vyhľadávacej fázy. Ak retriever nedokáže identifikovať relevantné pasáže alebo ak dôjde k strate kontextu pri nevhodnom rozdelení dokumentov, model môže generovať nepresné alebo neúplné odpovede. Tento jav je ešte výraznejší v právnej doméne, kde je potrebná presná interpretácia textu a zohľadnenie širšieho kontextu.

Uvedené obmedzenia naznačujú, že samotný prístup Retrieval-Augmented Generation nie je v právnej doméne postačujúci pre riešenie komplexných dotazov, ktoré vyžadujú pochopenie vzťahov medzi jednotlivými právnymi pojmami a dokumentmi. Najmä absencia explicitného modelovania hierarchických a sémantických väzieb vedie k potrebe rozšírenia klasického RAG prístupu o štruktúrovanú reprezentáciu znalostí. Jedným z riešení tohto problému je integrácia znalostných grafov do RAG pipeline, čo vedie k vzniku prístupu označovaného ako Graph RAG. Tento prístup sa snaží kombinovať výhody sémantického vyhľadávania s explicitnou reprezentáciou vzťahov medzi entitami, čím umožňuje presnejšie a kontextovo bohatšie generovanie odpovedí. [27]

3.3. Graph RAG systémy

Táto kapitola nadväzuje na poznatky uvedené v predchádzajúcich častiach práce. Ako bolo ukázané v predchádzajúcej podkapitole, samotný RAG prístup naráža v právnej doméne na viaceré obmedzenia, najmä pri práci s komplexnými vzťahmi medzi pojmami a dokumentmi. Znalostné grafy naopak umožňujú tieto vzťahy explicitne modelovať, no ich využitie je obmedzené rozsahom dostupných dát. Z tohto dôvodu je potrebné navrhnúť prístup, ktorý dokáže tieto metódy efektívne prepojiť a využiť ich komplementárne vlastnosti.

3.3.1. Čo je to Graph RAG?

V nadväznosti na uvedené obmedzenia klasického prístupu Retrieval-Augmented Generation (RAG) možno Graph RAG chápať ako jeho rozšírenie, ktoré do procesu vyhľadávania a generovania odpovede zapája aj znalostný graf ako štruktúrovaný zdroj informácií. Kým tradičný RAG je založený predovšetkým na práci s textovými úryvkami získanými prostredníctvom vektorového vyhľadávania, Graph RAG tento princíp dopĺňa o explicitne modelované entity a vzťahy medzi nimi. Jazykový model tak pri tvorbe odpovede nepracuje iba s neštruktúrovaným textovým kontextom, ale súčasne aj so štruktúrovanými znalosťami reprezentovanými vo forme grafu. Dochádza tak k prepojeniu dvoch rozdielnych, no navzájom sa dopĺňajúcich foriem reprezentácie znalostí – na jednej strane neštruktúrovaných textových dát spracovaných na základe sémantickej podobnosti a na druhej strane štruktúrovaných dát zachytávajúcich väzby medzi jednotlivými entitami v rámci domény.

Z technického hľadiska možno Graph RAG interpretovať ako rozšírenie klasickej RAG pipeline, v ktorej sa do fázy retrieval nepridáva iba ďalší dokumentový zdroj, ale samostatná vrstva reprezentovaná grafovou databázou alebo znalostným grafom. Tento zdroj môže byť dopytovaný prostredníctvom špecializovaných grafových dopytovacích jazykov a jeho výstupom nemusia byť len jednotlivé textové pasáže, ale aj podgrafy obsahujúce relevantné entity, ich atribúty a vzťahy medzi nimi. Takto získaný kontext má odlišný charakter než kontext v prípade čisto textového RAG. Neobsahuje totiž iba opisné alebo významovo podobné textové fragmenty, ale aj explicitne reprezentované väzby, ktoré umožňujú lepšie zachytiť štruktúru poznatkov v danej oblasti. Práve táto vlastnosť je dôležitá v doménach, kde výsledná odpoveď nezávisí len od obsahu jednotlivých dokumentov, ale aj od vzťahov medzi právnymi, vecnými alebo logickými jednotkami.

Pri spracovaní dopytu systém pracuje s dvoma komplementárnymi zdrojmi informácií, z ktorých každý plní odlišnú funkciu. Znalostný graf poskytuje explicitne definované vzťahy medzi entitami a umožňuje zachytiť štruktúrované väzby, ktoré by v čisto textovom priestore nemuseli byť jednoznačne identifikovateľné. V právnej oblasti môžu byť takýmito entitami napríklad právne pojmy, paragrafy, zákony, vyhlášky, súdne rozhodnutia alebo konkrétne ustanovenia legislatívnych dokumentov. Vektorové vyhľadávanie naopak umožňuje nájsť relevantné textové pasáže aj vtedy, keď určitý vzťah nie je v grafe explicitne zapísaný, ale je prítomný v texte vo forme opisného alebo kontextového vysvetlenia. Výsledkom je kombinovaný kontext, ktorý prepája presnosť štruktúrovaných dát so šírkou sémantického vyhľadávania a následne slúži ako vstup pre jazykový model pri generovaní finálnej odpovede.

Dôležitým aspektom tohto prístupu je spôsob, akým sa tieto dva typy informácií kombinujú. V praxi môže systém využívať paralelné vyhľadávanie, pri ktorom sa výsledky z grafu aj z vektorového indexu získavajú nezávisle od seba a následne sa spájajú do spoločného kontextu. Takýto prístup je vhodný najmä v situáciách, keď je potrebné rýchlo získať viacero pohľadov na rovnaký problém a následne ich zlúčiť do jednej odpovede. Alternatívou je sekvenčný prístup, pri ktorom jeden zdroj slúži na spresnenie, filtrovanie alebo doplnenie výsledkov získaných z druhého zdroja. Napríklad po identifikovaní relevantných entít v znalostnom grafe je možné následne vyhľadať konkrétne textové pasáže, ktoré tieto entity podrobnejšie vysvetľujú, alebo naopak na základe textového vyhľadávania zistiť, ktoré uzly a vzťahy v grafe sú najrelevantnejšie. Výber konkrétnej stratégie závisí od charakteru dopytu, štruktúry dát, požiadaviek na presnosť a aj od toho, do akej miery je potrebné zabezpečiť vysvetliteľnosť výsledného rozhodovania systému. [27][28]

3.3.2. Výhody a obmedzenia Graph RAG

Graph RAG prináša v právnej oblasti viacero významných výhod oproti prístupom, ktoré sú založené výlučne na vektorovom vyhľadávaní alebo na klasickom texte orientovanom RAG. Jednou z najdôležitejších výhod je schopnosť explicitne reprezentovať vzťahy medzi právnymi entitami a pracovať s nimi ako s plnohodnotnou súčasťou kontextu. Znalostný graf umožňuje modelovať nielen hierarchiu právnych predpisov, ale aj odkazy medzi paragrafmi, väzby medzi zákonom a jeho vykonávacou vyhláškou, príslušnosť konkrétnych ustanovení k právnym oblastiam či nadväznosť medzi súdnymi rozhodnutiami. Takto reprezentované znalosti umožňujú systému lepšie porozumieť tomu, v akom kontexte sa jednotlivé právne informácie nachádzajú, a zároveň poskytujú stabilnejší základ pre generovanie odpovedí pri zložitejších otázkach.

Ďalším významným prínosom je vyššia úroveň vysvetliteľnosti výsledkov. V prostredí, akým je právna doména, nestačí, aby systém poskytol odpoveď, ktorá je len pravdepodobne správna. Používateľ často očakáva aj možnosť overiť, z akých dokumentov, ustanovení alebo vzťahov bola odpoveď odvodená. Práve v tomto smere má Graph RAG výhodu, pretože odpoveď môže byť čiastočne opretá o konkrétne entity a explicitné väzby v znalostnom grafe.

V neposlednom rade tento prístup zlepšuje aj celkové pokrytie informácií. Vektorové vyhľadávanie prináša široký sémantický záber a dokáže identifikovať relevantné textové pasáže aj vtedy, keď sa používateľský dopyt nezhoduje s formuláciou v dokumentoch. Znalostný graf na druhej strane dopĺňa presnosť, štruktúru a explicitné väzby medzi dôležitými prvkami domény. Kombináciou týchto dvoch zdrojov vzniká robustnejší systém, ktorý dokáže efektívnejšie pracovať aj s neúplnými, nejednoznačnými alebo komplexnými otázkami. V právnej oblasti je takáto vlastnosť mimoriadne cenná, pretože právny dopyt môže byť formulovaný všeobecne, laicky alebo bez presného

označenia relevantného ustanovenia, pričom systém by mal byť napriek tomu schopný nájsť vecne správny a kontextovo podložený výsledok. [27][28][29][30]

Obmedzenia Graph RAG systému

Napriek tomu, že Graph RAG predstavuje perspektívne rozšírenie klasického prístupu RAG, jeho využitie so sebou prináša aj viaceré obmedzenia. Tieto obmedzenia sa týkajú nielen technickej implementácie systému, ale aj kvality reprezentovaných znalostí, spoľahlivosti retrieval mechanizmov a celkovej náročnosti nasadenia v reálnom prostredí. Súčasná literatúra poukazuje na to, že Graph RAG síce dokáže zlepšiť prácu s komplexnými vzťahmi a viac-krokovým uvažovaním, avšak za cenu vyššej systémovej zložitosti v porovnaní s tradičnými textovo orientovanými RAG systémami.

Jedným zo základných obmedzení Graph RAG je závislosť od kvality znalostného grafu. Ak graf neobsahuje dostatočne presné, úplné alebo aktuálne informácie, výsledný systém môže poskytovať nespoľahlivé odpovede aj v prípade, že samotný jazykový model a retrieval mechanizmus fungujú korektne. V právnej oblasti je táto otázka obzvlášť dôležitá, pretože právne dokumenty obsahujú zložité väzby medzi ustanoveniami, novelami, odkazmi a interpretačnými súvislosťami. Ak sú tieto väzby v grafe zachytené nepresne alebo neúplne, systém môže produkovať odpovede, ktoré síce pôsobia štruktúrovane, ale ich základ je fakticky oslabený.

Ďalším významným obmedzením je výpočtová a implementačná náročnosť. Oproti klasickému RAG, ktorý pracuje primárne s dokumentovým korpusom a vektorovým indexom, Graph RAG vyžaduje správu ďalšej vrstvy znalostí vo forme grafovej databázy alebo inej grafovej reprezentácie. To znamená potrebu navrhnuť schému grafu, zabezpečiť extrakciu a aktualizáciu uzlov a vzťahov a následne realizovať efektívne dopytovanie nad touto štruktúrou. Náročnosť rastie aj pri väčšom objeme dát, pretože retrieval nad grafom môže byť pri rozsiahlych a husto prepojených štruktúrach časovo aj pamäťovo náročný. Viaceré novšie práce zdôrazňujú, že škálovanie Graph RAG systémov zostáva otvoreným problémom a že latencia retrieval fázy môže byť jedným z limitujúcich faktorov pri praktickom nasadení. Tieto problémy sú ešte výraznejšie v prípadoch, keď sa Graph RAG kombinuje s ďalšími zložkami, ako sú vektorové indexy, reranking alebo viacstupňové pipeline spracovania.

Zhrnutím možno konštatovať, že Graph RAG prináša významné výhody najmä pri práci s komplexnými vzťahmi, doménovou štruktúrou a vysvetliteľnosťou odpovedí. Hoci je jeho praktické nasadenie spojené s viacerými otvorenými výzvami, systém ich efektívne adresuje práve tým, že spája silné stránky znalostného grafu a vektorového vyhľadávania. Táto realizácia je založená na hybridnom prístupe, ktorý určuje spôsob výberu relevantných zdrojov, kombináciu ich výsledkov a

formovanie finálneho kontextu pre generovanie odpovede. Z tohto dôvodu sa nasledujúca podkapitola zameriava na podrobný opis tohto hybridného mechanizmu ako kľúčového prvku fungovania celého systému. [31][32][33]

3.3.3. Hybridný prístup

V navrhovanom systéme je hybridný prístup realizovaný v zjednodušenej forme ako kombinácia dvoch hlavných mechanizmov získavania kontextu – vyhľadávania nad znalostným grafom a vektorového vyhľadávania nad textovým korpusom. Cieľom tohto prístupu nie je implementovať všetky pokročilé mechanizmy opisované v súčasnej literatúre, ale vytvoriť prakticky použiteľné riešenie, ktoré umožní efektívne spracovanie právnych dopytov s využitím štruktúrovaných aj neštruktúrovaných zdrojov informácií. Hybridnosť navrhovaného systému spočíva v tom, že sa pri spracovaní otázky najprv využíva grafová reprezentácia právnych znalostí a až následne, v prípade potreby, sa kontext dopĺňa pomocou vektorového vyhľadávania. Takto navrhnutý postup umožňuje spojiť presnosť a vzťahovosť znalostného grafu so sémantickou flexibilitou vektorového prístupu. Výsledkom je riešenie, ktoré je schopné pracovať s právnymi pojmami, väzbami medzi dokumentmi a zároveň aj s textovými pasážami, ktoré nemusia byť explicitne reprezentované v grafe.

Prvou fázou spracovania dopytu je analýza otázky a identifikácia relevantných pojmov. V tejto fáze sa systém snaží určiť, či sa v používateľskom dopyte nachádzajú právne termíny, názvy dokumentov, odkazy na ustanovenia alebo iné pojmy, ktoré možno prepojiť s uzlami v znalostnom grafe. Nejde pritom o plnohodnotnú implementáciu pokročilého rozpoznávania pomenovaných entít ani o sofistikovaný entity linking založený na skórovacích modeloch, ale o praktické mapovanie kľúčových pojmov na relevantné prvky grafovej reprezentácie. Tento krok je dôležitý preto, že umožňuje systému rozhodnúť, či je možné pri spracovaní otázky využiť štruktúrované vzťahy uložené v grafe. V právnej oblasti má takýto prístup význam najmä preto, že otázky používateľov často obsahujú konkrétne pojmy alebo odkazy, ktoré možno spojiť s právnymi normami, paragrafmi alebo súvisiacimi dokumentmi.

Na základe tejto počiatočnej analýzy systém uplatňuje jednoduchú rozhodovaciu logiku. Ak sa podarí identifikovať relevantné pojmy a tieto pojmy majú zodpovedajúce reprezentácie v znalostnom grafe, systém uprednostní grafové vyhľadávanie ako primárny mechanizmus získania kontextu. Ak však graf neposkytuje dostatočné pokrytie alebo nie je možné dopyt spoľahlivo priradiť ku konkrétnym uzlom a vzťahom, systém prechádza na vektorové vyhľadávanie ako doplnkový alebo náhradný zdroj informácií. Táto rozhodovacia logika nie je založená na pokročilom adaptívnom routingu ani na paralelnom vyhodnocovaní viacerých retrieval stratégií, ale na praktickom a transparentnom pravidle, ktoré je vhodné pre rozsah navrhovaného riešenia.

Výhodou takéhoto prístupu je jeho jednoduchosť, zrozumiteľnosť a zároveň možnosť jednoznačne vysvetliť, prečo systém v konkrétnom prípade použil určitý zdroj informácií.

Ak je dopyt úspešne previazaný so znalostným grafom, nasleduje grafový retrieval. V tejto fáze systém vyhľadáva relevantné uzly a vzťahy, ktoré súvisia s identifikovanými pojmami, a vytvára z nich kontext pre ďalšie spracovanie. Takto získané informácie môžu zahŕňať napríklad väzby medzi právnymi pojmami, odkazy medzi ustanoveniami, príslušnosť paragrafu ku konkrétnemu právnemu predpisu alebo iné vzťahy zachytené v grafovej databáze. Výhodou grafového prístupu je, že poskytuje explicitne štruktúrované informácie a umožňuje lepšie zachytiť súvislosti medzi jednotlivými prvkami právnej domény. To je dôležité najmä pri otázkach, kde nestačí nájsť iba textovo podobný úryvok, ale je potrebné zohľadniť aj vzťahy medzi právnymi entitami. V navrhovanom systéme preto znalostný graf predstavuje primárny zdroj kontextu všade tam, kde je možné tieto vzťahy spoľahlivo identifikovať a využiť.

V situáciách, keď grafová reprezentácia neposkytuje dostatočný kontext alebo keď sa relevantné informácie nachádzajú skôr v opisnom texte než vo forme explicitných väzieb, systém využíva vektorové vyhľadávanie ako fallback mechanizmus. Dopyt je v tomto prípade transformovaný do embeddingovej reprezentácie a následne porovnávaný s vektormi textových dokumentov alebo ich fragmentov uložených vo vektorovom indexe. Tento prístup umožňuje nájsť aj také textové pasáže, ktoré sú významovo príbuzné používateľskej otázke, aj keď neobsahujú presne tie isté pojmy alebo nie sú explicitne modelované v znalostnom grafe. Vektorové vyhľadávanie tak rozširuje pokrytie systému a zvyšuje jeho robustnosť pri spracovaní voľnejšie formulovaných alebo menej štruktúrovaných otázok. V navrhovanom riešení teda nepredstavuje konkurenčný mechanizmus voči grafu, ale doplnkovú vrstvu, ktorá pomáha v prípadoch, keď samotný graf nestačí na vytvorenie dostatočne informatívneho kontextu.

Po získaní výstupov z dostupných zdrojov nasleduje ich zlučovanie do spoločného vstupu pre jazykový model. V tejto fáze systém spája štruktúrované informácie získané zo znalostného grafu s relevantnými textovými úryvkami nájdenými pomocou vektorového vyhľadávania. Grafový kontext je pritom spravidla prevedený do textovej podoby tak, aby mohol byť spolu s textovým retrievalom odovzdaný jazykovému modelu ako jednotný vstup. Cieľom tejto fázy nie je aplikovať pokročilé re-ranking mechanizmy alebo sofistikované techniky fúzie poradia, ale vytvoriť zrozumiteľný a vecne relevantný kontext, ktorý kombinuje silné stránky oboch prístupov. Takto pripravený vstup následne slúži na generovanie finálnej odpovede. Zlučovanie kontextu má v rámci systému dôležitú úlohu, pretože umožňuje spojiť presnosť vzťahov zachytených v grafe so širším textovým pokrytím dokumentového korpusu.

Z implementačného hľadiska predstavuje tento prístup kompromis medzi jednoduchosťou riešenia a kvalitou výsledkov. Navrhovaný systém neimplementuje všetky pokročilé mechanizmy uvádzané v modernej literatúre, ako sú paralelné retrieval stratégie, dekompozícia komplexných otázok na viacero podotázok, skórovacie mechanizmy entity linking-u alebo pokročilé techniky re-rankingu. Napriek tomu však poskytuje funkčný základ hybridného Graph RAG prístupu, ktorý je pre potreby práce dostatočný a zároveň dobre obhájiteľný. Jeho hlavný prínos spočíva v tom, že kombinuje výhody dvoch rozdielnych spôsobov získavania informácií bez nadmerného zvyšovania implementačnej zložitosti. V právnej doméne je takýto prístup vhodný najmä preto, že umožňuje využívať explicitné vzťahy medzi právnymi entitami, no zároveň si zachováva schopnosť pracovať aj s textovými dokumentmi v prípadoch, keď štruktúrované znalosti nie sú postačujúce. Navrhnuté riešenie tak predstavuje praktickú realizáciu hybridného Graph RAG systému, ktorá zodpovedá rozsahu diplomovej práce a vytvára priestor pre prípadné budúce rozšírenie o pokročilejšie mechanizmy. [31][32][33][34][35]

4. Návrh a implementácia systému

Predchádzajúce kapitoly tejto práce sa venovali teoretickým základom, na ktorých stojí navrhnuté riešenie. Boli opísané znalostné grafy a ich využitie pri vyhľadávaní, grafové databázy a ich výhody oproti relačným systémom, moderné prístupy k sémantickému vyhľadávaniu, princíp RAG a jeho rozšírenie v podobe Graph RAG, ako aj obmedzenia, s ktorými sa tieto systémy v praxi stretávajú. Táto kapitola prechádza od teórie k praxi a opisuje konkrétny systém, ktorý bol v rámci tejto diplomovej práce navrhnutý a implementovaný. Ide o právneho asistenta schopného odpovedať na otázky z oblasti slovenského práva, pričom ku každej odpovedi prikladá citácie zdrojov, z ktorých pri formulovaní odpovede čerpal. Systém teda nie je len generatívnym modelom odpovedajúcim na základe naučených znalostí, ale aktívne pracuje s externou znalostnou bázou a textovým korpusom právnych predpisov.

4.1. Cieľ práce a motivácia návrhu

Primárnym cieľom práce je navrhnuť nástroj, ktorý dokáže zodpovedať otázky týkajúce sa slovenského práva spôsobom, ktorý je presný, vysvetliteľný a zdrojmi podložený. Motiváciou pre zvolený prístup boli konkrétne nedostatky, ktoré boli identifikované pri analýze klasického RAG systému v právnej doméne. Klasický RAG, postavený výlučne na vektorovom vyhľadávaní, nedokáže spoľahlivo pracovať s terminologickou presnosťou typickou pre právny jazyk. Právne pojmy majú presné definície zakotvené v konkrétnych zákonoch a vzájomné vzťahy medzi nimi sú explicitne dané právnym poriadkom. Vektorové vyhľadávanie tieto vzťahy zachytáva len implicitne, prostredníctvom štatistickej blízkosti v embeddingom priestore, čo pri terminologicky presných otázkach nemusí byť dostatočné. Navyše pri čisto vektorovom prístupe nie je garantované, že citovaný zdroj skutočne obsahuje hľadaný pojem ako definovanú právnu kategóriu – môže ísť len o dokument, ktorý daný výraz zmieňuje v inom kontexte.

Z týchto dôvodov bol zvolený hybridný prístup, v ktorom grafová databáza **Neo4j** plní úlohu primárneho zdroja terminologických znalostí. Táto databáza obsahuje tezaurus slovenských právnych pojmov s ich definíciami, vzájomnými vzťahmi a odkazmi na konkrétne právne predpisy. Keď sa v otázke používateľa podarí identifikovať pojem zastúpený v tezaure, systém uprednostní práve tento štruktúrovaný zdroj. Ak grafová databáza neposkytuje dostatočný kontext, systém sa obracia na vektorovú databázu FAISS, ktorá indexuje rozsiahlejší textový korpus právnych predpisov. Výsledný systém teda nie je ani čistý RAG, ani čistý grafový systém, ale hybridné riešenie, ktoré sa snaží využiť silné stránky oboch prístupov a vzájomne kompenzovať ich slabiny.

4.2. Architektúra navrhnutého systému

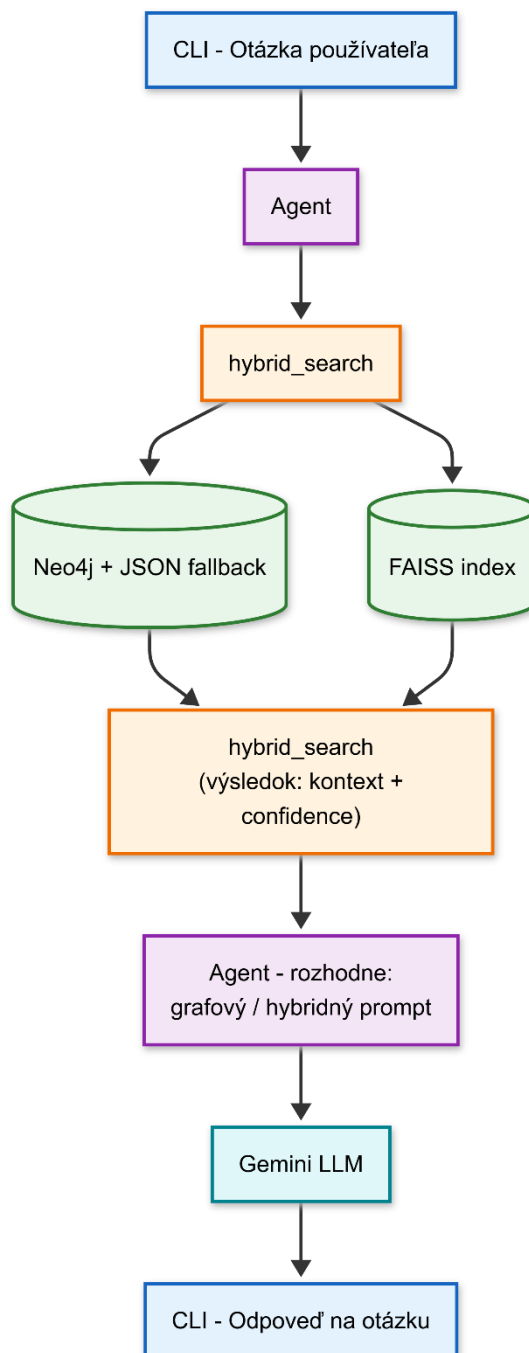
Z architektonického hľadiska je systém rozdelený do troch hlavných funkčných vrstiev, ktoré na seba logicky nadväzujú a spoločne tvoria kompletnú pipeline od vstupu používateľa až po vygenerovanú odpoveď.

Prvou vrstvou je vrstva **retrieval**, ktorej úlohou je na základe používateľskej otázky získať relevantný kontext z dostupných dátových zdrojov. Táto vrstva komunikuje s grafovou databázou Neo4j prostredníctvom jazyka Cypher a s vektorovou databázou FAISS prostredníctvom embeddingového vyhľadávania. Kľúčovou súčasťou tejto vrstvy je rozhodovacia logika, ktorá na základe vypočítanej hodnoty istoty určuje, či bude pre danú otázku použitý výlučne grafový kontext, alebo či bude doplnený o výsledky z vektorovej databázy. Vrstva retrieval taktiež implementuje záložný mechanizmus pre prípad, keď Neo4j nie je dostupný – v takom prípade sa vyhľadávanie realizuje priamo nad lokálne uloženým JSON súborom s dátami tezauru.

Druhou vrstvou je vrstva **reasoning**, kde agent vyhodnocuje získaný kontext a rozhoduje o spôsobe generovania odpovede. Táto vrstva obsahuje logiku, ktorá na základe výsledkov z vrstvy retrieval určuje, aký typ promptu bude odovzdaný jazykovému modelu, a zároveň rieši situácie, keď model signalizuje nedostatočnosť grafového kontextu a je potrebné aktivovať doplnkové vyhľadávanie.

Tretou vrstvou je vrstva **generácie**, v ktorej jazykový model Google Gemini na základe pripraveného kontextu a prislúchajúceho promptu vytvára finálnu textovú odpoveď. Model v závislosti od zdroja kontextu pracuje buď v striktnom režime, kde je viazaný výlučne na obsah znalostného grafu, alebo v kombinovanom režime, kde môže pracovať aj s textovými fragmentmi z korpusu právnych predpisov.

Tieto tri vrstvy sú prepojené nasledovne: Používateľ zadá otázku cez rozhranie príkazového riadku, tá je odovzdaná agentovi reasoning vrstvy, agent spustí hybridné vyhľadávanie v retrieval vrstve, získaný kontext spolu s otázkou skonštruuje do promptu a odošle do generačnej vrstvy, ktorá vráti finálnu odpoveď. Odpoveď je zobrazená používateľovi spolu s informáciou o zdrojoch. Pre lepšiu predstavu o fungovaní znázorňuje obrázok č. 1 kompletný priebeh interakcie medzi používateľom a systémom, od prijatia otázky cez rozhranie CLI až po doručenie finálnej odpovede vygenerovanej modelom Gemini LLM.



Obr. 2 Architektúra navrhnutého systému

4.3. Využitie technológií

Implementácia systému je postavená na niekoľkých kľúčových technológiách, ktoré boli zvolené na základe ich vhodnosti pre danú doménu, dostupnosti a kompatibility.

Základným programovacím jazykom je Python, ktorý poskytuje rozsiahly ekosystém knižníc pre prácu s jazykovými modelmi, vektorovými databázami aj grafovými systémami. Pre správu dátových modelov je využívaná knižnica Pydantic, ktorá zabezpečuje typovú bezpečnosť pri práci so štruktúrovanými dátami reprezentujúcimi správy konverzácie a výstupy retrieval mechanizmov.

Grafová databáza je postavená na systéme Neo4j, ktorý bol bližšie opísaný v teoretickej časti tejto práce. Pre dopytovanie grafovej databázy je využívaný jazyk Cypher, ktorý umožňuje efektívne prechádzanie vzťahov medzi právnymi pojmami a ich zdrojmi. Pripojenie k Neo4j je realizované prostredníctvom oficiálneho Python drivera neo4j, ktorý podporuje transakčné operácie a overenie spojenia.

Vektorová databáza je implementovaná pomocou knižnice FAISS (Facebook AI Similarity Search). Samotný proces vytvorenia a naplnenia tejto databázy realizoval Bc. Martin Šarišský, ktorému touto cestou vyjadrujem poďakovanie za jeho prínos a spoluprácu. Pre tvorbu vektorových reprezentácií textov je využívaný model intfloat/multilingual-e5-large dostupný cez platformu Hugging Face. Tento model bol zvolený predovšetkým preto, že je trénovaný na viacjazyčných dátach vrátane slovenčiny a češtiny a dosahuje vysokú kvalitu embeddingov aj pre doménovo špecifické texty. Na načítanie a správu FAISS indexu je využívaná knižnica langchain-community, konkrétne triedy InMemoryDocstore a príslušné utilitné funkcie pre formát indexu generovaný knižnicou LangChain.

Jazykový model pre generáciu odpovedí je Google Gemini, k dispozícii je verzia gemini-2.5-flash-lite. Prístup k modelu je realizovaný cez knižnicu google-generativeai, ktorá poskytuje Python rozhranie pre Gemini API. Konfigurácia modelu a API kľúča je načítaná zo súboru .env pomocou knižnice python-dotenv, čím sú citlivé prístupové údaje oddelené od zdrojového kódu.

Pre dosiahnutie robustného vyhľadávania v grafovej databáze je do procesu spracovania textu zaradená lemmatizácia. Lemmatizácia je proces lingvistickej analýzy, ktorého cieľom je určiť základný, slovníkový tvar slova (tzv. lemu). Jej využitie je v systéme nevyhnutné predovšetkým kvôli povahe slovenského a českého jazyka, kde jedno slovo môže nadobúdať desiatky rôznych tvarov v závislosti od pádu, čísla alebo rodu. Bez tohto kroku by vyhľadávanie v grafovej databáze záviselo od presnej zhody reťazcov, čo by viedlo k neúspešnému nadeniu informácií v prípadoch, kedy používateľ použije iný gramatický tvar slova, než aký je uložený v databáze (napríklad rozdiel medzi „študentovi“ a „študent“).

4.4. Infaštruktúra a nasadenie

Zdrojový kód systému vrátane FAISS indexu je uložený a spúšťaný na serveri **Quadro** Fakulty elektrotechniky a informatiky Technickej univerzity v Košiciach. Server disponuje dostatočnými výpočtovými zdrojmi pre načítanie a prevádzku FAISS indexu aj embeddingového modelu, čo by na bežnom osobnom počítači mohlo predstavovať výkonnostné obmedzenie najmä pri inicializácii. Pripojenie k serveru prebieha cez SSH protokol, pričom vzdialený prístup umožňuje interaktívnu prácu so systémom v reálnom čase.

Grafová databáza Neo4j beží lokálne na počítači vývojára, teda mojom, a nie na serveri Quadro. Toto riešenie bolo zvolené z dôvodu jednoduchšej správy dát a nevyhnutnosti mať Neo4j dostupné aj počas vývojovej fázy. Aby k tejto lokálnej databáze mohol pristupovať systém bežiaci na vzdialenom serveri Quadro, je využívaný mechanizmus SSH reverzného tunela. Reverzný tunel sa vytvorí príkazom `ssh -R 17687:127.0.0.1:7687 scislak@quadro.kemt.fei.tuke.sk`, ktorý spôsobí, že port 17687 na serveri Quadro je presmerovaný na port 7687 na lokálnom počítači, kde beží Neo4j. Systém bežiaci na Quadre sa teda pripája k adrese `localhost:17687` a komunikuje v skutočnosti s lokálnou inštanciou Neo4j, pričom všetka komunikácia prebieha cez zabezpečený SSH kanál.

Pri každom štarte systém automaticky overí dostupnosť všetkých komponentov. Kontrola Neo4j spočíva v pokuse o pripojenie a vykonaní jednoduchého testovacieho dopytu. Kontrola FAISS indexu overuje existenciu súborov `index.faiss` a `index.pkl` v očakávanom adresári. Kontrola Gemini API overuje prítomnosť a základný formát API kľúča. Výsledky týchto kontrol sú vypisované na štandardný výstup, čo umožňuje operátorovi systému okamžite identifikovať prípadné problémy s pripojením alebo konfiguráciou ešte pred prvým dopytom používateľa.

```
python -c "  
import sys  
sys.path.insert(0, '.')  
from retrieval.hybrid_retriever import verify_all_systems  
verify_all_systems()  
"  
  
--- KONTROLA SYSTÉMOV PRI ŠTARTE ---  
✅ Neo4j: OK (pripojené k tezaurus-db)  
✅ FAISS: OK (index nájdený v /mnt/sharedhome/sarissky/faiss_index_full_merged)  
✅ GEMINI: OK (kľúč nastavený)  
-----
```

Obr. 3 Overenie dostupnosti komponentov

5. Dátová príprava

Kvalita a štruktúra vstupných dát má v systémoch typu Graph RAG zásadný vplyv na správnosť a spoľahlivosť výsledkov. Na rozdiel od klasických RAG systémov, kde stačí indexovať textové dokumenty do vektorovej databázy, hybridný prístup využívaný v tomto systéme vyžaduje prípravu dvoch vzájomne sa dopĺňujúcich dátových zdrojov. Prvým je grafová databáza obsahujúca štruktúrovanú reprezentáciu právnych pojmov a ich vzájomných vzťahov. Druhým je vektorová databáza obsahujúca textové fragmenty z právnych predpisov vhodné na sémantické vyhľadávanie. Táto kapitola opisuje, aké dátové zdroje boli použité, akým spôsobom boli spracované a ako sú uložené v príslušných databázach.

5.1. Zdroj dát

Primárnym zdrojom štruktúrovaných dát pre grafovú databázu je tezaurus právnych pojmov. Tezaurus je kontrolovaná slovná zásoba, ktorá obsahuje preferované názvy pojmov, ich definície a explicitne definované vzťahy medzi nimi – hierarchické aj asociatívne. V právnej doméne má tezaurus osobitný význam, pretože právne pojmy nie sú len slovami prirodzeného jazyka, ale termínmi s presne vymedzeným obsahom zakotveným v konkrétnych právnych predpisoch. Použitie tezauru ako základu znalostného grafu zaručuje, že systém pracuje s autoritatívnymi definíciami, a nie s aproximáciami odvodenými zo štatistického spracovania textu.

Pre potreby tohto systému boli použité dáta zo siedmich právnych tezaurov pokrývajúcich rôzne oblasti slovenského práva, tezaurus:

- Občianskeho práva
- Obchodného práva
- Trestného práva
- Finančného práva
- Ústavného práva
- Správneho práva
- Pracovného práva

Tieto tezaury obsahujú spolu 2 781 záznamov reprezentujúcich právne pojmy z rôznych oblastí slovenského právneho poriadku. Každý záznam obsahuje identifikátor pojmu, jeho textový obsah vrátane definície a metadáta opisujúce vzťahy pojmu k iným entitám.

Dáta boli uložené v súbore **neo4j_data.json**, ktorý predstavuje vstupný dataset pre import do grafovej databázy. Každý záznam v tomto súbore má štruktúru tvorenú tromi hlavnými poliami. Pole id obsahuje jednoznačný identifikátor pojmu, ktorý zodpovedá jeho preferovanému názvu v

tezaure, napríklad „absolútny obchod“ alebo „blízka osoba“. Pole text obsahuje textový obsah záznamu vrátane definície pojmu v prirodzenom jazyku. Pole metadata obsahuje štruktúrované informácie o vzťahoch pojmu k iným entitám. Konkrétne ide o tri typy informácií: zoznam zdrojov, v ktorých je pojem definovaný (defined_by), identifikátor tezauru, z ktorého pojem pochádza (from_thesaurus), a zoznam súvisiacich pojmov (related_to). Táto štruktúra umožňuje pri importe priamo vytvoriť uzly, vzťahy aj väzby na zdroje bez potreby ďalšieho mapovania alebo transformácie.

```
{
  "id": "absolútny obchod",
  "text": "Pojem: absolútny obchod\nDefinícia: Termín pre obchodný záväzkový vzťah, ktorý sa riadi so zreteľom na povahu vecí OBZ, a nie je rozhodujúce ani to, či ho uzatvárajú podnikatelia alebo iné osoby, ani to, či sa plnenie použije na podnikateľské účely. OBZ a zák. o cenných papieroch obsahuje taxatívny výpočet absolútnych obchodov.\nTezaurus: Tezaurus obchodného práva",
  "metadata": {
    "related_to": [
      "relatívny obchod"
    ],
    "defined_by": [
      "91/2019 Z. z.",
      "91/2019 Z. z. - Zákon o neprimeraných podmienkach v obchode s potravinami a o zmene a doplnení niektorých zákonov"
    ],
    "from_thesaurus": "Tezaurus obchodného práva"
  }
}
```

Obr. 4 Ukážka záznamu dát z grafovej databázy

Príklad záznamu z datasetu zobrazený na obrázku č. 3 ilustruje typickú štruktúru právneho pojmu. Pojem „absolútny obchod“ z Tezauru obchodného práva má definíciu opisujúcu, že ide o obchodný záväzkový vzťah riadený Obchodným zákonníkom bez ohľadu na to, či ho uzatvárajú podnikatelia alebo iné osoby. Zdroje (defined_by) uvádzajú konkrétne publikácie, z ktorých definícia pochádza. Tezaurus (from_thesaurus) identifikuje príslušný tematický tezaurus. Príbuzné pojmy (related_to) odkazujú na iné termíny, s ktorými daný pojem súvisí.

Každý z použitých siedmich tezaurov je tematicky ohraničený na jednu oblasť práva. Tezaurus občianskeho práva pokrýva pojmy z oblasti osobného práva, záväzkového práva, vlastníctva a dedičstva. Tezaurus obchodného práva zahŕňa terminológiu týkajúcu sa obchodných spoločností, záväzkových vzťahov medzi podnikateľmi a obchodného registra. Tezaurus pracovného práva pokrýva vzťahy medzi zamestnávateľom a zamestnancom, pracovné podmienky a pracovnoprávne spory. Tezaurus trestného práva zahŕňa definície trestných činov, tresty a procesné pojmy. Tezaurus finančného práva pokrýva dane, poplatky a verejné financie. Tezaurus ústavného práva zahŕňa základné práva a slobody, štátnu správu a samosprávu. Tezaurus správneho práva pokrýva vzťahy medzi orgánmi verejnej moci a fyzickými alebo právnickými osobami. Najväčším tezaustom z

hľadiska počtu pojmov je tezaurus občianskeho práva, čo zodpovedá skutočnosti, že občianske právo tvorí základ slovenského súkromného práva a jeho terminológia je najrozvetvenejšia. Celkový počet takmer 2 800 pojmov nepredstavuje úplné pokrytie slovenského právneho poriadku, no pre účely prototypu pokrýva bežné otázky z každodennej právnej praxe. Práve táto skutočnosť sa odráža aj vo výsledkoch hodnotenia – systém aktivoval grafovú cestu vyhľadávania pri 74,5 % testovacích otázok, čo potvrdzuje, že zvolený rozsah tezauru zodpovedá typickým právnym dopytom bežného používateľa.

5.2. Schéma grafovej databázy

Grafová databáza Neo4j využíva model labeled property graph. Pre potreby systému bola navrhnutá schéma s tromi typmi uzlov a tromi typmi vzťahov, ktoré priamo zodpovedajú štruktúre vstupného datasetu.

Prvým typom uzla je uzol s označením **Term**, ktorý reprezentuje právny pojem. Každý uzol Term obsahuje atribút name, v ktorom je uložený identifikátor pojmu zodpovedajúci poľu id zo vstupného JSON súboru, a atribút definition, v ktorom je uložená definícia pojmu extrahovaná z textového obsahu záznamu. Uzol Term je základným stavebným prvkom celého grafu – reprezentuje entitu, na ktorú sa bude systém dopytovať pri spracovaní právnych otázok.

Druhým typom uzla je uzol s označením **Source**, ktorý reprezentuje právny zdroj alebo autoritu, z ktorej definícia pojmu pochádza. Môže ísť o zákon, zbierku zákonov, odbornú publikáciu alebo inú relevantnú autoritu. Uzol Source obsahuje atribút name s názvom zdroja tak, ako je uvedený v poli defined_by vstupného záznamu.

Tretím typom uzla je uzol s označením **Thesaurus**, ktorý reprezentuje konkrétny tematický tezaurus, z ktorého pojem pochádza. Systém pracuje so siedmimi tezaurami pokrývajúcimi rôzne oblasti práva. Uzol Thesaurus obsahuje atribút name s názvom tezauru.

Z hľadiska vzťahov medzi uzlami sú v grafe definované tri typy. Vzťah **DEFINED_BY** spája uzol Term s príslušným uzlom Source a vyjadruje, v ktorom zdroji je daný pojem autoritatívne definovaný. Tento vzťah je kľúčový pre funkciu citovania zdrojov – keď systém nájde v grafe relevantný pojem, prostredníctvom vzťahu DEFINED_BY získa aj informáciu o tom, na základe akého právneho predpisu alebo publikácie je pojem definovaný, a túto informáciu prezentuje používateľovi. Vzťah **RELATED_TO** spája navzájom súvisiace pojmy a umožňuje navigáciu po asociatívnych väzbách v grafe. Tento vzťah rozširuje kontext, ktorý systém získa pri identifikácii pojmu v otázke – okrem samotnej definície nájdeného pojmu môže systém poskytnúť aj zoznam príbuzných termínov, čo

obohacuje kontext odovzdávaný jazykovému modelu. Vzťah **FROM_THESAURUS** prepája uzol Term s uzlom Thesaurus a zaznamenáva, z ktorého tematického tezauru pojem pochádza.

Pre každý typ uzla bol vytvorený B-stromový index nad atribútom name. Tieto indexy sú kľúčové pre výkon vyhľadávania, pretože umožňujú databáze rýchlo lokalizovať uzly podľa ich názvu bez potreby prehľadávať celý graf. V praxi to znamená, že pri hľadaní podreťazca pomocou porovnávania normalizovaného textu s názvom v databáze sa namiesto zdĺhavého prechádzania každého jedného záznamu aktivuje vopred vytvorený index. Databáza tak vďaka predpripravenej štruktúre okamžite prejde na relevantné uzly, čo radikálne zrýchľuje odozvu, najmä pri veľkom objeme dát.

5.3. Integrácia údajov do databázy

Import dát z JSON súboru do grafovej databázy Neo4j bol realizovaný prostredníctvom skriptu `data/import_neo4j.py`, ktorý implementuje ETL proces (Extract, Transform, Load) rozdelený do štyroch sekvenčných krokov. Každý krok spracováva jeden typ entít alebo vzťahov a pracuje s dávkami po 500 záznamov, čím sa predchádza pamäťovým problémom pri spracovaní väčšieho objemu dát.

V prvom kroku sú importované uzly Term. Pre každý záznam zo vstupného JSON súboru je z textového obsahu extrahovaná definícia pojmu pomocou regulárneho výrazu, ktorý vyhľadá text nachádzajúci sa za kľúčovým slovom „Definícia:“ a pred prípadným začiatkom ďalšej sekcie. Uzly sú do databázy zapisované príkazom MERGE, ktorý zabezpečuje idempotentnosť operácie – ak uzol s daným názvom už existuje, príkaz ho aktualizuje namiesto vytvorenia duplicitného záznamu. Tento prístup umožňuje bezpečné opakované spúšťanie importného skriptu bez rizika vzniku duplicitných uzlov.

V druhom kroku sú importované uzly Source a vzťahy DEFINED_BY. Pre každý záznam sú spracované všetky položky zo zoznamu `defined_by` v metadátach. Pre každú položku je vytvorený uzol Source (ak ešte neexistuje) a vzťah DEFINED_BY medzi príslušným uzlom Term a týmto uzlom Source. Aj tu je využívaný príkaz MERGE pre zaistenie deduplikácie.

V treťom kroku sú importované uzly Thesaurus a vzťahy FROM_THESAURUS. Pre každý záznam, ktorý má v metadátach vyplnené pole `from_thesaurus`, je vytvorený uzol Thesaurus (ak ešte neexistuje) a vzťah FROM_THESAURUS medzi uzlom Term a príslušným uzlom Thesaurus.

Vo štvrtom kroku sú importované vzťahy RELATED_TO medzi pojmami. Pre každý záznam sú spracované všetky položky zo zoznamu `related_to` v metadátach. Pre každú dvojicu pojmov je vytvorený vzťah RELATED_TO, pričom cieľový uzol Term je taktiež vytvorený príkazom MERGE, ak

ešte neexistuje. Toto zabezpečuje konzistentnosť grafu aj v prípadoch, keď cieľový pojem nie je explicitne definovaný vo vstupnom datasete, ale je len odkazovaný z iného záznamu.

Dávkové spracovanie je implementované tak, že záznamy sú postupne pridávané do dočasného zoznamu (batch), a keď jeho veľkosť dosiahne 500 položiek, celý dávkový zoznam je odoslaný do databázy v rámci jednej transakcie pomocou príkazu UNWIND. Tento prístup je výrazne efektívnejší ako individuálne spracovanie každého záznamu v samostatnej transakcii, pretože minimalizuje réžiu spojenú so správou transakcií a sieťovou komunikáciou.

5.4. Vektorová databáza FAISS

Druhým dátovým zdrojom systému je vektorová databáza obsahujúca textové fragmenty z právnych predpisov. Jej úlohou je pokryť tú časť právnych otázok, pri ktorých sa v texte otázky nenachádza presný termín zastúpený v tezaure, no relevantné informácie existujú v textovom korpuse zákonov. Zatiaľ čo grafová databáza poskytuje terminologickú presnosť a explicitné vzťahy, vektorová databáza dopĺňa systém o sémantickú flexibilitu umožňujúcu nájsť relevantný obsah aj pri voľnejšie formulovaných otázkach.

Samotný proces zostavenia a naplnenia vektorovej databázy realizoval Bc. Martin Šarišský, ktorému za jeho prínos a spoluprácu patrí poďakovanie. Výsledkom jeho práce je FAISS index uložený na serveri Quadro, ktorý tento systém využíva ako externý dátový zdroj. Pre integráciu tohto indexu do systému bolo potrebné pochopiť jeho štruktúru, formát uloženia a spôsob práce s ním, čo je predmetom tejto podkapitoly.

Pre tvorbu vektorových reprezentácií textov bol použitý model **intfloat/multilingual-e5-large** dostupný cez platformu **Hugging Face**. Tento model bol zvolený predovšetkým pre jeho schopnosť pracovať s viacjazyčnými textami vrátane slovenčiny a češtiny, čo je pre doménu slovenského práva nevyhnutné. Model produkuje vektory dimenzie 1024 a podporuje rozlíšenie medzi dopytovacími a dokumentovými vektormi prostredníctvom prefixov. Pre dopyty sa používa prefix „query: ” a pre dokumenty prefix „passage: ”. Toto rozlíšenie je zabudované do architektúry modelu a zlepšuje presnosť vyhľadávania tým, že dopyt a dokumenty sú zakódované do mierne odlišných reprezentácií, ktoré sú navzájom porovnateľné, no optimalizované pre odlišné účely.

Index bol zostavený s využitím indexového typu IVF (Inverted File Index). Tento typ indexu rozdeľuje vektorový priestor do zhlukov, pričom každý vektor je priradený k najbližšiemu zhuku. Pri vyhľadávaní sa namiesto prehľadávania celého indexu prechádzajú len tie zhluky, ktoré sú najbližšie k vektoru dopytu. Počet prehľadávaných zhlukov je riadený parametrom *nprobe*. V tomto systéme je *nprobe* nastavený na hodnotu 64. Táto hodnota bola zvolená s ohľadom na kvalitu výsledkov –

predvolená hodnota, ktorá je typicky 8, by pri rozsahu tohto indexu spôsobovala, že pri každom dopyte zostane väčšina databázy neprehľadaná, čo by výrazne znižovalo recall systému. Hodnota 64 predstavuje rozumný kompromis medzi výpočtovou náročnosťou a úplnosťou vyhľadávania.

Index je fyzicky uložený v dvoch súboroch na serveri Quadro. Súbor `index.faiss` obsahuje samotnú FAISS dátovú štruktúru s vektormi dokumentov. Súbor `index.pkl` obsahuje docstore vo formáte kompatibilnom s knižnicou LangChain – konkrétne dvojicu tvorenú objektom `InMemoryDocstore`, ktorý mapuje identifikátory dokumentov na ich obsah a metadáta, a slovníkom mapujúcim interné FAISS indexy na tieto identifikátory. Každý dokument v docstore obsahuje textový obsah fragmentu a metadáta zahŕňajúce číslo zákona, jeho názov, číslo paragrafu alebo článku, číslo odseku, rok vydania a názov zdrojového súboru. Tieto metadáta systém využíva pri zostavovaní citácií, ktoré sú súčasťou odpovede prezentovanej používateľovi.

Pri integrácii FAISS indexu do hybridného retrieval mechanizmu bolo potrebné zohľadniť skutočnosť, že index obsahuje aj záznamy pochádzajúce z tezauru právnych pojmov, ktoré sú v metadátach označené príznakom `from_thesaurus`. Keďže obsah tezaurových pojmov je v systéme pokrytý grafovou cestou cez Neo4j, ich opätovné zahrnutie cez vektorovú databázu by viedlo k duplicite kontextu – rovnaká informácia by bola jazykovému modelu odovzdaná dvakrát. Toto by zbytočne zaberalo miesto v obmedzenom kontextovom okne a znižovalo by podiel nových informácií v odpovedi. Z tohto dôvodu sú tezaurové záznamy pri každom vyhľadávaní automaticky preskočené. Aby po filtrácii zostal dostatočný počet skutočných právnych dokumentov, systém načítava osemnásobok požadovaného počtu výsledkov a postupne ich prechádza až kým nezobiera požadovaný počet relevantných, netezaurových záznamov.

6. Implementácia hybridného retrievalu

Hybridný retrieval je jadrom celého nášho systému. Jeho úlohou je na základe prirodzeného jazyka používateľskej otázky získať relevantný kontext z grafovej aj vektorovej databázy a zároveň určiť, do akej miery je grafová databáza pre danú otázku spoľahlivým zdrojom. Výsledok tejto fázy priamo ovplyvňuje, aký kontext dostane jazykový model a akým typom promptu bude vyzvaný k odpovedi. Celá logika hybridného vyhľadávania je sústredená v module `retrieval/hybrid_retriever.py` a kombinuje tri čiastkové komponenty:

1. **Neo4j retriever** zabezpečujúci grafové vyhľadávanie
2. **FAISS retriever** zabezpečujúci sémantické vyhľadávanie
3. **Rozhodovaciú logiku**, ktorá tieto dve cesty orchestruje

6.1. Predspracovanie pre grafové vyhľadávanie

Slovenčina patrí medzi morfológicky bohaté jazyky, čo znamená, že každé slovo môže mať desiatky tvarov v závislosti od pádu, čísla, rodu alebo slovesného času. Táto vlastnosť predstavuje pri vyhľadávaní v grafovej databáze výzvu. Pojem uložený v grafe pod názvom „blízka osoba“ môže byť v otázke použitý v podobe „blízkou osobou“, „blízkej osoby“ alebo „blízkych osôb“, pričom všetky tieto tvary odkazujú na ten istý právny pojem. Jednoduché textové porovnávanie by tieto varianty nerozpoznalo ako zhodu.

Na riešenie tohto problému je implementovaná **lemmatizácia**. Je to proces prevodu rôznych tvarov slov na ich základný slovníkový tvar, nazývaný lemma. Lemmatizácia je realizovaná prostredníctvom knižnice **simplemma**, ktorá podporuje slovenčinu ako primárny jazyk a češtinu ako záložný jazyk pre tvary, ktoré sa v slovenčine nenájdu. Napríklad slovo „blízkou“ je lematizované na „blízky“ a slovo „osobou“ na „osoba“. Vďaka tomu sa oba varianty frázy „blízka osoba“ aj „blízkou osobou“ po lematizácii zhodujú na spoločnej reprezentácii „blízky osoba“.

Implementácia lemmatizácie je sústredená v module `retrieval/lemmatizer.py`. Funkcia `lemmatize_word` prevádza jednotlivé slová na ich základný tvar, funkcia `lemmatize_text` tokenizuje celý text, lematizuje každý token a výsledné lemmy spojí medzerami do reťazca vhodného na ďalšie porovnávanie. Pre prípad, keď **simplemma** nie je dostupná, je implementovaný bezpečnostný mechanizmus, ktorý vráti pôvodný tvar slova bez chyby – systém teda funguje aj bez lemmatizácie, len s nižšou schopnosťou zachytiť rôzne tvary.

Osobitným prípadom, na ktorý lemmatizácia sama nestačí, je situácia, keď **simplemma** vracia pre gramaticky príbuzné tvary toho istého adjektíva rôzne lemmy. Napríklad tvar „blízka“ môže byť lematizovaný na „blízko“ a tvar „blízkou“ na „blízky“, hoci oba pochádzajú z toho istého adjektíva.

Tento problém rieši záložný mechanizmus prefixového stemuovania. Funkcia `stem` oreže slovo na prvých päť znakov, čím vytvorí hrubý spoločný základ. Keďže „blízko" aj „blízky" zdieľajú prefix „blízk", stemovanie ich správne identifikuje ako príbuzné. Funkcia `lemma_or_stem_match` kombinuje oba prístupy - najprv skúsi presnú zhodu lemiem a ak tá zlyhá, vyhodnotí zhodu na úrovni prefixových stemov.

6.2. Proces grafového vyhľadávania

Trieda `Neo4jRetriever` implementovaná v module `retrieval/neo4j_retriever.py` zabezpečuje grafové vyhľadávania. Jej hlavná metóda prijíma text používateľskej otázky a snaží sa v grafovej databáze nájsť právny pojem, ktorý je v tejto otázke obsiahnutý. Vyhľadávania prebieha v troch krokoch vykonávaných postupne, pričom každý nasledujúci krok sa aktivuje len vtedy, keď predchádzajúci nevráti žiaden výsledok.

Prvým krokom je priamy **substring match**. Systém odošle do Neo4j Cypher dopyt, ktorý prehľadá všetky uzly a pre každý z nich overí, či je jeho atribút obsiahnutý v texte otázky. Výsledky sú zoradené podľa dĺžky názvu termu zostupne a obmedzené na tri najlepšie zhody. Uprednostňovanie dlhších termov je zámerné – pri otázke obsahujúcej napríklad frázu „obchodný záväzkový vzťah" je žiadanejšie nájsť tento viacslovný term ako len jednoslovný „záväzkový vzťah", pretože dlhší term je špecifickejší a pravdepodobne presnejšie zodpovedá tomu, na čo sa používateľ pýta.

Druhým krokom je **lemmatizovaný match**. Tento krok sa aktivuje v prípade, keď priamy substring match zlyhá, čo typicky nastáva pri skloňovaných tvaroch slov. Otázka je najprv lemmatizovaná a z výsledku sú extrahované kľúčové slová po odstránení stopwords. Ide o bežné slovné druhy (najmä častice, predložky a zámená), ktoré sú v procese spracovania prirodzeného jazyka odstraňované, aby sa zvýšila váha sémanticky významných pojmov. Tieto lemmatizované kľúčové slová sú odoslané do Neo4j ako zoznam, pričom databáza vráti termy, ktorých názov obsahuje aspoň jedno z týchto slov. Tým vznikne množina kandidátov, ktorú systém ďalej spracúva v Pythone - pre každý kandidátny term sú jeho slová lemmatizované a funkciou `lemma_or_stem_match` sa overuje, či sa všetky lemmy daného termu vyskytujú v lemmách otázky. Len termy, pre ktoré táto podmienka platí, sú zaradené do výsledku. Príkladom fungovania tohto kroku je otázka „Kto sa považuje za blízkou osobou?", kde priamy match zlyhá, no po lemmatizácii sú obe slová termu „blízka osoba" – teda „blízky" a „osoba" – prítomné v lemmách otázky, čo vedie k úspešnej identifikácii termu.

Tretím krokom je **keyword fallback**. Ak ani lemmatizovaný match nevrátil výsledok, systém extrahuje z otázky kľúčové slová funkciou `_extract_keywords`. Táto funkcia tokenizuje otázku, každý token lemmatizuje a odfiltruje stopwords. Vyskytovali by sa v takmer každom terme a tak by znižovali presnosť vyhľadávania, ako napríklad „zákon", „zmluva" alebo „povinnosť". Z takto

získaných kľúčových slov sa odošle dopyt do Neo4j, ktorý vráti termy obsahujúce aspoň dve z týchto lemmatizovaných kľúčových slov. Podmienka dvoch zhôd slúži ako filter relevantnosti – term, ktorý zdieľa iba jedno kľúčové slovo s otázkou, je pravdepodobne príliš všeobecný.

Bez ohľadu na to, ktorý krok vyhľadávania bol úspešný, výstup metódy `find_in_question` má jednotnú štruktúru. Vracia slovník obsahujúci nájdený term, jeho definíciu, zoznam zdrojových právnych predpisov získaných cez vzťah `DEFINED_BY`, zoznam súvisiacich pojmov získaných cez vzťah `RELATED_TO` a informáciu o type zhody s hodnotami „exact“ alebo „keyword“. Ak bolo úspešných viacero krokov alebo ak prvý krok vrátil viac výsledkov, zoznamy sú zlúčené z všetkých nájdených termov, pričom sú zachované len unikátne hodnoty a poradie prvého výskytu.

6.3. FAISS retriever a sémentické vyhľadávanie

Trieda `FaissRetriever` implementovaná v module `retrieval/faiss_retriever.py` zabezpečuje vektorové sémantické vyhľadávanie v textovom korpuse právnych predpisov. Jej práca prebieha v troch fázach:

1. Zakódovanie otázky
2. Vyhľadávanie podobných dokumentov
3. Zostavenie výsledku

V prvej fáze je text otázky **zakódovaný do číselného vektora** pomocou embeddingového modelu `multilingual-e5-large`. Pred text otázky je automaticky pridaný prefix „query: “, ktorý signalizuje modelu, že ide o vyhľadávací dopyt. Vďaka tomu model otázku zakóduje spôsobom optimalizovaným na porovnávanie s dokumentmi uloženými v indexe.

V druhej fáze je vektor otázky **porovnaný s vektormi** dokumentov vo FAISS indexe a systém vráti zoznam najpodobnejších dokumentov. Ako bolo opísané v kapitole 7.4 – Vektorová databáza FAISS, index obsahuje aj tezaurové záznamy, ktoré je pri vyhľadávaní potrebné preskočiť – ich obsah pokrýva grafová cesta cez Neo4j. Aby po odfiltrovaní týchto záznamov zostal dostatočný počet výsledkov, systém načíta osemnásobok požadovaného počtu kandidátov. Predvolený počet výsledkov je 15, teda systém načíta až 120 kandidátov a z nich vyberie prvých 15 skutočných právnych dokumentov.

V tretej fáze sú z každého nájdeného dokumentu **zostavené citácie** zdrojov. Každá citácia obsahuje číslo zákona, jeho názov, číslo paragrafu alebo článku a číslo odseku, napríklad „40/1964 Zb. – Občiansky zákonník (§ 116, ods. 1)“. Pred odovzdaním výsledkov sú dokumenty zoradené podľa roku vydania od najnovších, čím sa uprednostňujú aktuálnejšie právne predpisy. Zostavené citácie sú následne prezentované používateľovi spolu s odpoveďou.

6.4. Orchestrácia hybridného vyhľadávania

Prepojenie grafového a vektorového vyhľadávania zabezpečuje funkcia `hybrid_search`. Tá dostane otázku používateľa, spustí Neo4j retriever aj FAISS retriever a vráti štyri veci:

- textový kontext z grafovej databázy
- textový kontext z FAISS
- hodnotu istoty
- slovník so zdrojmi

Hodnota istoty vyjadruje, nakoľko si je systém istý, že nájdený term v grafe skutočne zodpovedá tomu, na čo sa používateľ pýta. Nejde o hodnotu vypočítanú nmodelom – sú to ručne nastavené prahy, ktoré vznikli počas vývoja systému pozorovaním toho, ako sa vyhľadávanie správa pri rôznych typoch otázok. Rozhodovacím bodom je porovnanie výslednej hodnoty istoty s pevným **prahom 0.85**. Ak je dosiahnutý alebo prekročený, systém použije výlučne grafový kontext. Ak je nižšia, spustí sa aj FAISS vyhľadávanie. Nasledujúci popis jednotlivých hodnôt vysvetľuje, kedy a prečo systém konkrétnu mieru istoty priradí.

Najvyššia istota 0.9 sa priradí vtedy, keď je nájdený term priamou textovou zhodou a zároveň tvorí podstatnú časť otázky – buď viac ako 40 percent jej textu, alebo je celá otázka kratšia ako 50 znakov. Príkladom je otázka „Čo je blízka osoba?“. Táto otázka má 22 znakov a term „blízka osoba“ tvorí väčšinu jej obsahu. Je takmer isté, že používateľ sa pýta priamo na definíciu tohto pojmu – grafový kontext teda postačí a nie je nutné spúšťať FAISS databázu. Na obrázku č. 5 vidíme, že systém našiel presnú definíciu v grafovej databáze. Overiť si to môžeme priamo z tezauru slovenského jazyka, ktorý je dostupný na tomto odkaze: <https://www.slov-lex.sk/tezaury/terminy/6d7d20ac-3c1e-49eb-9ac0-27299a725e09>.

Ty: Čo je blízka osoba?

Systém:

Blízkou osobou je príbuzný v priamom rade, súrodenec a manžel. Iné osoby v pomere rodinnom alebo obdobnom sa pokladajú za osoby sebe navzájom blízke, ak by ujmu, ktorú utrpela jedna z nich, druhá dôvodne pociťovala ako vlastnú ujmu. Stupeň príbuzenstva dvoch osôb sa určuje podľa počtu zrodení, ktorými v priamom rade pochádza jedna od druhej a v pobočnom rade obidve od najbližšieho spoločného predka.

Zdroj (znalostný graf): blízka osoba

Právne predpisy: 40/1964 Zb. § 116 – Občiansky zákonník | 40/1964 Zb. § 117 – Občiansky zákonník

Obr. 5 Odpoveď systému založená výhradne na grafovom kontexte

Istota 0.6 sa priradí pri presnej zhode, keď je však otázka dlhšia a term tvorí len jej malú časť. Napríklad pri otázke „Za akých podmienok môže blízka osoba napadnúť darovaciu zmluvu?“ systém

nájde term „blízka osoba“. Otázka sa v skutočnosti pýta na podmienky a darovaciu zmluvu, nie len na definíciu pojmu. Samotná definícia z grafu nestačí, preto sa doplní aj FAISS vyhľadávanie. Na obrázku č. 6 vidíme, že na to aby systém zodpovedal používateľovu otázku použil grafovú databázu ako zdroj. To však nebolo postačujúce, a preto použil aj ďalšie dokumenty. Na základe kombinácie týchto zdrojov dokázal odpovedať na otázku.

```
Ty: Za akých podmienok môže blízka osoba napadnúť darovaciu zmluvu?

Systém:
Blízka osoba môže napadnúť darovaciu zmluvu, ak ju uzatvorila osoba, ktorá nemá plnú spôsobilosť na právne
úkony, alebo ak bola zmluva uzavretá pod vplyvom násillia, hrozby či podvodu. V takýchto prípadoch môže blízka
osoba konať v mene darcu alebo uplatniť jeho práva. Dôležité je tiež, aby sa blízka osoba, ktorá by inak mohla
byť poškodená darovaním, priamo nedopustila konania, ktoré by znemožňovalo napadnutie zmluvy. Občiansky zákonník
bližšie špecifikuje, kto je považovaný za blízku osobu.

Zdroj (znalostný graf): blízka osoba
Pravne predpisy:
40/1964 Zb. § 116 - Občiansky zákonník | 40/1964 Zb. § 117 - Občiansky zákonník

Dokumenty: 18/2018 Z. z. - Zákon o ochrane osobných údajov a o zmene a doplnení niektorých zákonov | 122/2013 Z.
z. - Zákon o ochrane osobných údajov a o zmene a doplnení niektorých zákonov | 329/2000 Z. z. - Zákon o Úrade
pre finančný trh a o zmene a doplnení niektorých zákonov
```

Obr. 6 Odpoveď systému vytvorená kombináciou grafového a vektorového vyhľadávania

Istota 0.4 sa nastaví pri lemmatizovanej zhode alebo pri zhode kľúčových slov. Tu si systém nie je úplne istý, či skutočne našiel správny term, zhoda je nepriama. Preto sa FAISS doplní vždy, bez ohľadu na dĺžku otázky.

Okrem týchto základných pravidiel je implementovaná dodatočná rozhodovacia logika. Ak je nájdený term jednoslovný, no otázka obsahuje tri alebo viac dlhších obsahových slov, istota sa zníži na maximálne 0.8. Príkladom je otázka „Za akých podmienok vzniká nárok na náhradu škody?“ Systém môže nájsť term „škoda“, no otázka sa pýta na komplexnejšiu problematiku. Jednoduchá definícia pojmu „škoda“ by nestačila, preto sa kontext doplní z FAISS.

Istota	Typ zhody	Režim vyhľadávania
0.9	Presná zhoda, term dominuje otázke	Znalostný graf
0.6	Presná zhoda, term je malou časťou otázky	Znalostný graf + FAISS
0.4	Lemmantizovaná zhoda / zhoda kľúčových slov	Znalostný graf + FAISS
< 0.8	Jednoslovný term, komplexná otázka	Znalostný graf + FAISS

Tab. 1 Prehľad hodnôt istoty a výsledného režimu vyhľadávania

Výsledok celého procesu je zachytený v hodnote `mode`, ktorá hovorí, aký typ vyhľadávania bol nakoniec použitý. Hodnota `neo4j` znamená, že stačil grafový kontext. Hodnota `faiss` znamená, že grafové vyhľadávanie zlyhalo a použil sa len FAISS. Hodnota `hybrid` znamená, že boli použité oba zdroje. Hodnota `json_fallback` znamená, že Neo4j nebola dostupná a vyhľadávanie prebehlo nad lokálnym JSON súborom. Hodnota `none` znamená, že žiaden zdroj neposkytol relevantný výsledok. Táto informácia je neskôr zobrazená používateľovi ako súčasť zdrojov a zároveň ju využíva agent v ďalšej fáze spracovania.

7. Reasoning a generovanie odpovede

Po tom, čo retrieval vrstva získa kontext z grafovej a vektorovej databázy, nastupuje fáza reasoning a generácie. V tejto fáze agent rozhoduje, čo s kontextom urobí a akým spôsobom ho odovzdá jazykovému modelu. Jazykový model následne na základe tohto kontextu vygeneruje finálnu odpoveď.

7.1. Agent a jeho rozhodovacia logika

Trieda ReActAgent implementovaná v module reasoning/react_agent.py plní úlohu orchestrátora celého systému. Dostane otázku používateľa, zavolá hybridné vyhľadávanie, vyhodnotí výsledok a rozhodne, ako formulovať vstup pre jazykový model. Rozhodovacia logika agenta má tri vetvy, pričom agent vždy vykoná prvú vetvu, ktorej podmienka platí.

Prvá vetva nastáva vtedy, keď retrieval vrátil mode s hodnotou **none** – žiadny zdroj nenašiel pre danú otázku relevantný obsah. V takom prípade agent okamžite vráti správu „Nemám k tejto otázke relevantné zdroje.“ bez toho, aby vôbec zavolať jazykový model. Ak systém nemá žiadny kontext, je lepšie neodpovedať, ako nechať model odpovedať len na základe vlastných naučených znalostí a riskovať nepresnosti alebo halucinácie.

Druhá vetva nastáva pri vysokej istote grafovej cesty, konkrétne keď je istota **rovná alebo vyššia ako 0.85** a grafový kontext nie je prázdny. Agent v tomto prípade odovzdá jazykovému modelu výlučne grafový kontext spolu so striktným promptom, ktorý modelu zakazuje použiť vlastné znalosti. Model musí odpovedať len na základe toho, čo je v kontexte, a ak hľadaná informácia nie je prítomná, musí odpovedať presnou frázou „Informácia nie je obsiahnutá v znalostnom grafe.“ Táto konkrétna fráza je kľúčová – agent ju sleduje v odpovedi a ak ju nájde, vyhodnotí to ako signál, že grafový kontext pre danú otázku nestačil. Automaticky spustí FAISS vyhľadávanie a pokračuje treťou vetvou.

Tretia vetva, **hybridný režim**, nastáva pri nízkej hodnote istoty alebo ako pokračovanie po FAISS fallbacku z druhej vetvy. Agent zlúči grafový a vektorový kontext do jedného textového bloku a odovzdá ho jazykovému modelu s voľnejším promptom, ktorý modelu umožňuje syntetizovať odpoveď z oboch zdrojov.

Výsledná odpoveď je vždy doplnená o blok so zdrojmi. Funkcia `_format_sources` zostaví tento blok na základe hodnoty mode a obsahu slovníka sources. Obsahuje identifikátor nájdeného termu zo znalostného grafu, zoznam citovaných právnych predpisov získaných cez vzťah `DEFINED_BY` a zoznam dokumentov z FAISS. Táto informácia umožňuje používateľovi overiť, z akých zdrojov odpoveď vychádza, čo je v právnej doméne osobitne dôležité.

7.2. Jazykový model Google Gemini

Komunikáciu s jazykovým modelom zabezpečuje trieda GeminiClient implementovaná v module llm/gemini_client.py. Systém využíva model **gemini-2.5-flash-lite**. Tento model bol zvolený ako vyvážený kompromis medzi rýchlosťou odozvy a kvalitou generovaných odpovedí, čo je pri interaktívnom konverzačnom systéme dôležité.

Každé volanie modelu pozostáva z troch vstupných prvkov:

- Otázka používateľa
- Kontext získaný z retrieval vrstvy
- Pravidlá formulované v systémovej časti promptu

Systémová časť promptu sa líši v závislosti od toho, z akého zdroja kontext pochádza. Keď je zdrojom výlučne znalostný graf, prompt obsahuje striktné pravidlá: model musí odpovedať výhradne na základe poskytnutého kontextu, nesmie použiť vlastné znalosti a ak informácia v kontexte chýba, musí odpovedať presnou frázou „Informácia nie je obsiahnutá v znalostnom grafe.“ Táto konkrétna fráza je kľúčová – agent ju sleduje v odpovedi a ak ju nájde, spustí FAISS fallback opísaný v predchádzajúcej podkapitole.

Keď je zdrojom kombinovaný kontext z grafovej databázy a FAISS, prompt je voľnejší. Model je vyzvaný odpovedať stručne, vecne a prehľadne v rozsahu päť až sedem viet. V tomto režime môže model syntetizovať odpoveď z oboch typov kontextu.

Pred odovzdaním kontextu modelu je jeho text orezaný na maximálne 3 000 znakov. Toto obmedzenie je pevné a zabraňuje prekročeniu limitov API. V praxi to znamená, že ak by bol kombinovaný kontext príliš dlhý, jeho prebytočná časť sa odreže. Z tohto dôvodu sú kontexty z grafovej databázy a FAISS obmedzované už na úrovni retrieval vrstvy – grafový kontext na 1 500 znakov a FAISS kontext na 3 000 znakov. Celkový limit 3 000 znakov v GeminiClient tak slúži ako posledná záchranná sieť pre prípad, keď by zlúčený kontext napriek tomu presiahol bezpečnú hranicu.

Implementácia ošetruje chybový stav ResourceExhausted, ku ktorému dochádza pri prekročení kvótového limitu API. V takom prípade model vráti používateľovi zrozumiteľnú správu, že odpoveď momentálne nie je dostupná a má skúsiť znova o chvíľu. Systém tým predchádza nekontrolovaným výnimkám, ktoré by mohli spôsobiť pád aplikácie počas behu.

7.3. Správa konverzačnej pamäte

Systém podporuje viacotázkový dialóg – používateľ môže klásť nadväzujúce otázky a systém si pamätá históriu konverzácie. Správu pamäte zabezpečujú dva moduly v balíku `memory/`.

Trieda `Conversation` v module `memory/conversation_state.py` uchováva zoznam všetkých správ konverzácie. Každá správa má rolu (`user` alebo `assistant`) a textový obsah. Trieda poskytuje metódy na pridávanie správ, čítanie celej histórie a získanie poslednej otázky používateľa. Okrem zoznamu správ môže objekt `Conversation` uchovávať aj súhrnný text staršej histórie, ktorý vzniká pri sumarizácii.

Trieda `MemoryManager` v module `memory/memory_manager.py` sleduje, ako rastie zoznam správ, a zabezpečuje, aby kontext odosielaný do jazykového modelu nenarástol do neúnosnej veľkosti. V systéme je `MemoryManager` inicializovaný s hodnotou `max_messages` rovnou 5. Prah pre spustenie sumarizácie je nastavený na dvojnásobok tejto hodnoty, teda na 10 správ. Keď počet nahromadených správ presiahne túto hranicu, `MemoryManager` zavolá Gemini s promptom, ktorý ho požiada o stručné a vecné zhrnutie staršej časti konverzácie so zameraním na právne pojmy, definície a závery. Výsledné zhrnutie sa uloží do objektu `Conversation` a pri ďalšom spracovaní otázky je prepojené do kontextu pred aktuálne správy. Vďaka tomu systém „nezabúda“ na dôležité informácie z predchádzajúcich otázok, no zároveň neposiela do každého volania plnú históriu všetkých správ – do kontextu vstupuje vždy len posledných 5 správ spolu s prípadným súhrnom.

8. Vyhodnotenie systému

Navrhnutý a implementovaný systém bol podrobený praktickému hodnoteniu, ktorého cieľom bolo zistiť, nakoľko správne odpovedá na právne otázky a nakoľko spoľahlivo identifikuje a cituje relevantné právne predpisy. Táto kapitola opisuje metodiku hodnotenia, prezentuje jeho výsledky a diskutuje o silných a slabých stránkach systému v kontexte obmedzení opísaných v teoretickej časti práce.

8.1. Metodika hodnotenia

Pre hodnotenie systému bol zostavený súbor 110 otázok pokrývajúcich rôzne oblasti slovenského práva. Otázky boli formulované tak, aby reprezentovali bežné typy právnych dopytov – od jednoduchých terminologických otázok (napr. „Čo je blízka osoba?“) cez otázky o konkrétnych právach a povinnostiach (napr. „Môže vykonávať mladistvý zamestnanec nočnú prácu?“) až po otázky vyžadujúce znalosť viacerých ustanovení (napr. „Môže mi súd uložiť pokutu, ak som bol predvolaný na súd no nedostavil som sa tam a svoju neúčasť som neospravednil?“). Otázky pokrývali oblasti občianskeho, obchodného, pracovného, trestného, správneho a finančného práva, čo zodpovedá tematickému rozsahu tezaurov použitých pri budovaní grafovej databázy.

Pri zostavovaní súboru otázok bol kladený dôraz na rozmanitosť z hľadiska formulácie aj právnej náročnosti. Niektoré otázky boli zámerne formulované laicky, bez použitia presnej právnej terminológie, aby sa overila schopnosť systému porozumieť otázke aj vtedy, keď používateľ neovláda právny jazyk. Iné otázky naopak obsahovali priamo právne termíny z tezauru, čím sa testovala grafová cesta vyhľadávania. Zaradené boli aj otázky týkajúce sa oblastí, kde dochádza k častým zmenám legislatívy, a to predovšetkým s cieľom overiť, ako sa systém vysporiada s potenciálnou neaktuálnosťou vektorového indexu.

Hodnotenie prebiehalo prostredníctvom štruktúrovanej tabuľky zostavenej v prostredí Microsoft Excel. Tabuľka obsahuje 110 riadkov a 10 stĺpcov, ktoré zachytávajú reťazec od zadania po hodnotenie. Každý riadok zodpovedá jednej testovacej otázke.

Prvý stĺpec obsahuje **poradové číslo** záznamu. Druhý stĺpec, označený ako Výrok, obsahuje samotnú **otázku položenú systému**. Tretí stĺpec obsahuje **odpoveď chatbota**. Štvrtý stĺpec obsahuje **správnu odpoveď** – znenie zostavené právničkou na základe platnej legislatívy. Piaty stĺpec zachytáva **vyhodnotenie správnosti odpovede** pomocou jednej zo šiestich kategórií. Šiesty stĺpec obsahuje **zdroje citované systémom** – identifikátor termu zo znalostného grafu a zoznam právnych predpisov. Siedmy a ôsmy stĺpec obsahujú **správny zákon** a **správny paragraf** vzťahujúci sa na danú

otázku. Deviaty stĺpec zachytáva vyhodnotenie správnosti citovaného zdroja pomocou jednej zo štyroch kategórií.

Hodnotenie vykonala právnička Mgr. Kristína Ščišľáková, ktorá pre každý záznam nezávisle doplnila referenčné odpovede, správne zákony a paragrafy a priradila každej odpovedi aj každému zdroju príslušnú kategóriu. Zapojenie odborníčky z právnej praxe bolo nevyhnutné, keďže posúdenie správnosti právnych odpovedí si vyžaduje odbornú znalosť platného právneho poriadku a schopnosť rozlíšiť úplnú správnosť od čiastočnej. Automatizované hodnotenie by takúto nuansnú analýzu nedokázalo spoľahlivo poskytnúť.

Pre hodnotenie správnosti odpovede bolo zavedených šesť kategórií, tie sú bližšie opísané v tabuľke č. 2.

Číslo kategórie	Názov kategórie	Popis kategórie
1	Správna	Odpoveď je úplná a fakticky presná.
2	Správna s nepresným doplnkom	Jadro odpovede je správne, no doplňujúce informácie sú neúplné alebo obsahujú chyby.
3	Čiastočne správna (neúplná)	Odpoveď je pravdivá, ale neuvádza všetky zákonom stanovené podmienky či možnosti.
4	Čiastočne správna (s chybou)	Odpoveď obsahuje kombináciu správnych a nesprávnych tvrdení.
5	Neaktuálna	Odpoveď vychádza z už neúčinnnej právnej úpravy.
6	Nesprávna	Odpoveď je fakticky chybná.

Tab. 2 Kategórizácia správnosti odpovedí systému

Pre hodnotenie presnosti citovaného zdroja boli zavedené štyri kategórie, sú popísané v tabuľke číslo 3.

Číslo kategórie	Názov kategórie	Popis kategórie
1	Relevantný zdroj	Citácia správneho a v čase testovania účinného právneho predpisu.
2	Irelevantný zdroj	Citácia predpisu, ktorý nesúvisí s danou problematikou.
3	Neúčinný zdroj	Citácia staršieho, už neplatného znenia zákona.
4	Chýbajúci zdroj	Systém neposkytol žiadnu referenciu na právny predpis.

Tab. 3 Kategorizácia presnosti identifikácie právnych zdrojov

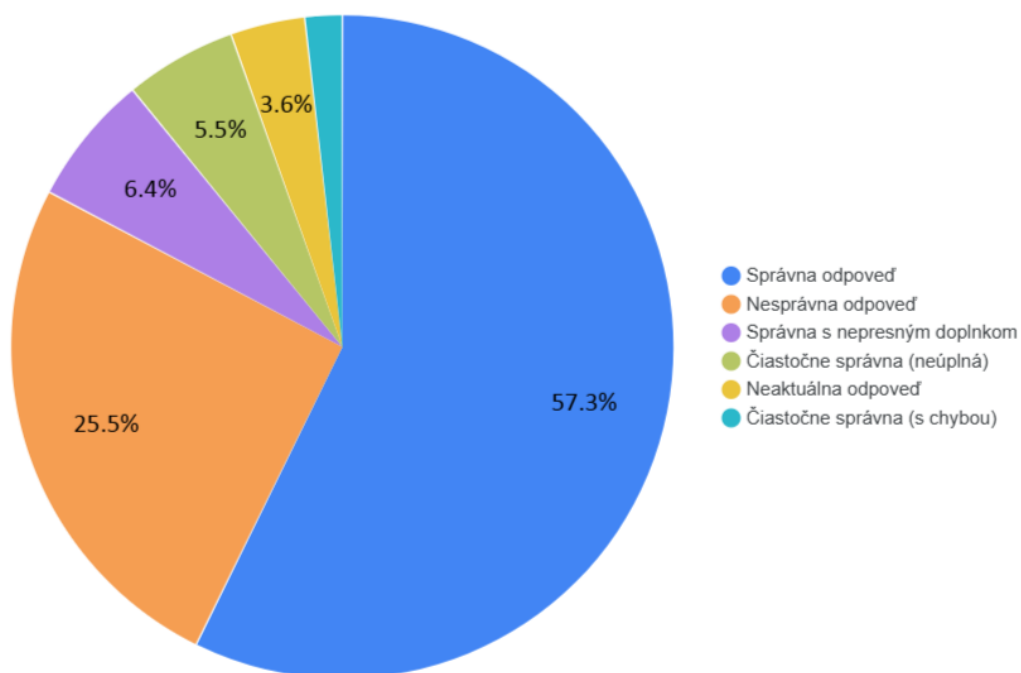
Použitie dvoch samostatných kritérií pomáha odhaliť prípady, kedy systém síce odpovie vecne správne, ale odkáže na zlý zákon. V práve je takýto nesúlad medzi odpoveďou a citáciou zásadným problémom.

Výsledky hodnotenia boli spracované v nástroji **Looker Studio** a vizualizované vo forme štyroch koláčových grafov. Prvé dva grafy hodnotia systém ako celok na všetkých 110 otázkach. Tretí a štvrtý graf rozdeľujú hodnotené otázky podľa toho, či bol pri vyhľadávaní použitý znalostný graf. Tretí graf pokrýva 82 otázok, pri ktorých grafová cesta aktívna bola, štvrtý graf pokrýva zvyšných 28 otázok, pri ktorých systém pracoval výlučne s vektorovou databázou. Keďže Graf č. 3 a Graf č. 4 sú navzájom doplnkové podmnožiny Grafu č. 1, ich priame porovnanie tvorí základ pre empirické posúdenie prínosu znalostného grafu a je predmetom diskusie v podkapitole 10.3.

8.2. Výsledky hodnotenia

Prvý graf zobrazuje rozdelenie všetkých 110 hodnotených odpovedí do šiestich kategórií správnosti a poskytuje základný prehľad o celkovej kvalite systému. Najväčší podiel tvorí kategória správnych odpovedí s 63 prípadmi, čo predstavuje 57,3 % všetkých hodnotených otázok. Do tejto kategórie patria odpovede, ktoré právnik vyhodnotil ako vecne správne a úplné. Druhý najvyšší podiel má kategória nesprávnych odpovedí s 28 prípadmi, teda 25,5 %. Zvyšných 17,3 % je rozdelených medzi čiastočne správne odpovede rôznych typov: 7 odpovedí (6,4 %) bolo správnych, no s faktickými nepresnosťami v doplňujúcom texte, 6 odpovedí (5,5 %) bolo čiastočne správnych, pričom zákon uvádza ďalšie podmienky alebo možnosti, ktoré systém nezmienil, 4 odpovede (3,6 %) neboli

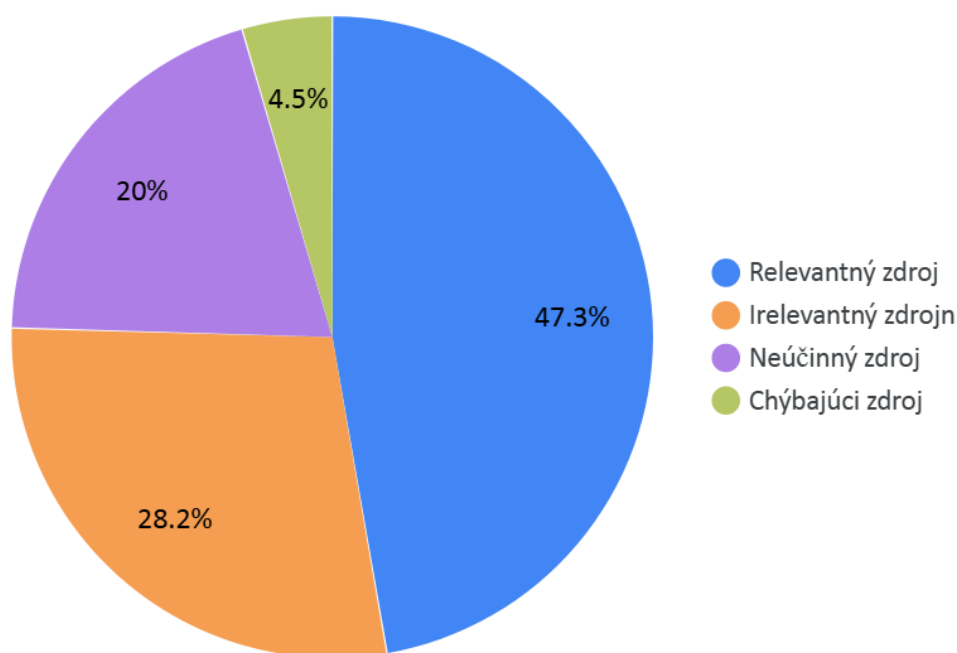
aktuálne a vychádzali z neaktuálnej právnej úpravy a 2 odpovede (1,8 %) boli čiastočne správne, no časť ich obsahu bola nesprávna.



Obr. 7 Vyhodnotenie úspešnosti odpovedí systému v rámci celého testovacieho súboru

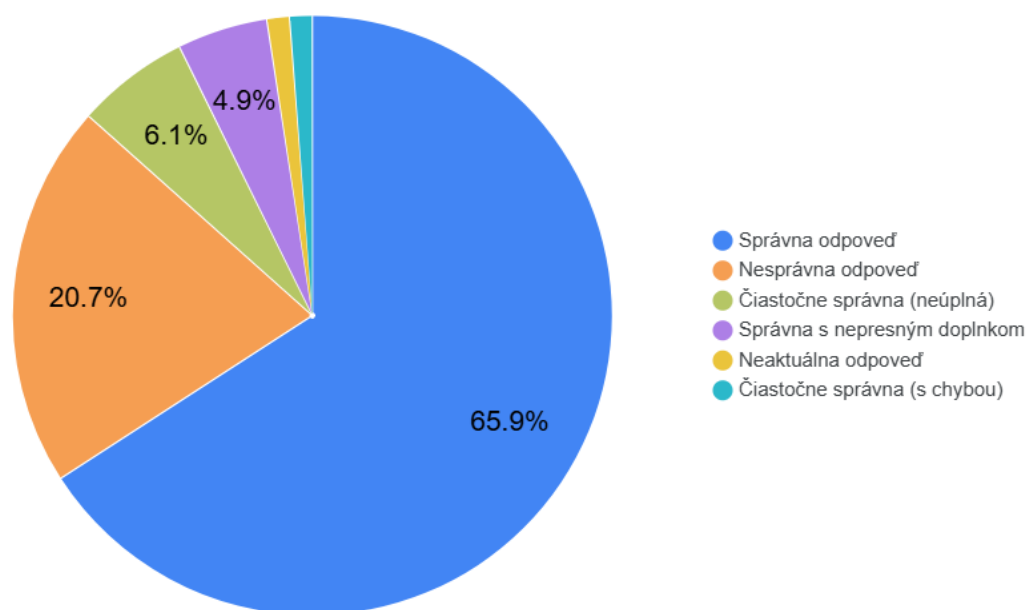
Ak zjednodušíme pohľad a zlúčime kategórie 2, 3 a 4 do skupiny čiastočne správnych odpovedí a kategórie 5 a 6 do skupiny nesprávnych alebo neaktuálnych, dostaneme nasledujúci obraz: **57,3 % odpovedí bolo úplne správnych**, 13,7 % čiastočne správnych a 29,1 % nesprávnych alebo neaktuálnych. Tento zjednodušený pohľad ukazuje, že systém **správne odpovedá na viac ako polovicu otázok**, pričom pri ďalšej šestine odpovedí obsahuje odpoveď aspoň jadro správnej informácie.

Druhý graf hodnotí schopnosť systému správne identifikovať a citovať relevantný právny predpis. Správny zákon uviedol systém v 52 prípadoch, čo predstavuje 47,3 % všetkých otázok. Nesprávny zákon bol citovaný v 31 prípadoch (28,2 %) a neaktuálne znenie zákona v 22 prípadoch (20,0 %). V 5 prípadoch (4,5 %) systém zdroj vôbec neuviedol. Z týchto výsledkov vyplýva, že správnosť identifikácie zdroja je nižšia ako správnosť samotnej odpovede. To naznačuje, že systém dokáže poskytnúť vecne správnu informáciu aj v situáciách, keď nie je schopný ju presne priradiť k správne právnomu predpisu – napríklad keď čerpá z FAISS kontextu, kde nie sú metadáta dokumentov vždy dostatočne presné.



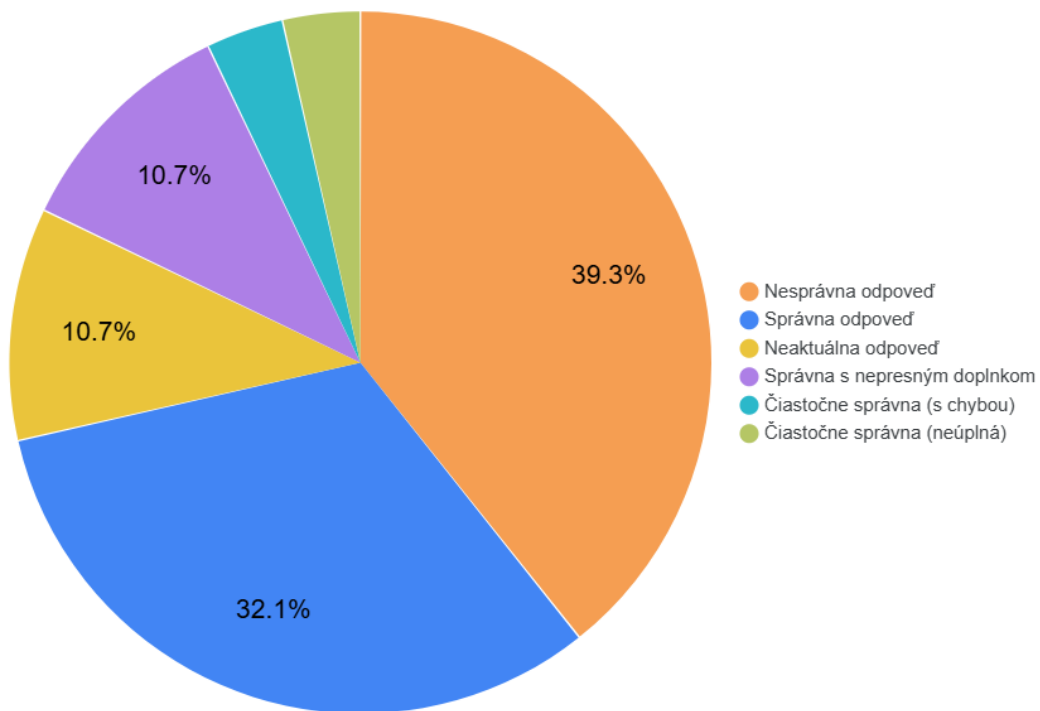
Obr. 8 Vyhodnotenie úspešnosti identifikácie právnych zdrojov v rámci celého testovacieho súboru

Tretí graf sa zameriava výlučne na 82 otázok, pri ktorých systém pri spracovaní využil znalostný graf – či už v čistom grafovom alebo hybridnom režime. Z celkových 110 otázok teda pri 82 (74,5 %) systém aktivoval grafovú cestu vyhľadávania, čo potvrdzuje, že tezaurus právnych pojmov pokrýva väčšinu typicky kladených právnych otázok. Kategória správnych odpovedí dosiahla 54 prípadov, čo predstavuje 65,9 %. Nesprávne odpovede tvoria 20,7 %. Čiastočne správne odpovede tvoria zvyšných 13,4 %: 4 odpovede (4,9 %) boli správne s nie celkom pravdivými doplňujúcimi informáciami, 5 odpovedí (6,1 %) bolo čiastočne správnych s neuvedenými ďalšími podmienkami, 1 odpoveď (1,2 %) bola čiastočne správna s chybnou časťou a 1 odpoveď (1,2 %) nebola aktuálna.



Obr. 9 Vyhodnotenie úspešnosti odpovedí systému pri využití výlučne grafovej databázy

Štvrtý graf zobrazuje výsledky pre zvyšných 28 otázok, pri ktorých systém nenašiel v tezaure žiadny relevantný pojem a spracoval otázku výlučne prostredníctvom vektorovej databázy FAISS. Táto skupina tvorí 25,5 % všetkých hodnotených otázok. Kategória správnych odpovedí dosiahla len 9 prípadov, čo predstavuje 32,1 %. Nesprávne odpovede tvoria 39,3 % (11 prípadov) – ide o výrazne vyšší podiel ako v skupine s aktívnou grafovou cestou. Čiastočne správne a neaktuálne odpovede tvoria zvyšných 28,6 %: 3 odpovede (10,7 %) boli správne s nepresným doplnkovým obsahom, 1 odpoveď (3,6 %) bola čiastočne správna, pričom zákon uvádza ďalšie podmienky, 1 odpoveď (3,6 %) bola čiastočne správna s chybnou časťou a 3 odpovede (10,7 %) neboli aktuálne. Pozoruhodný je najmä podiel neaktuálnych odpovedí, ktorý v tejto skupine dosiahol 10,7 % – takmer trojnásobok hodnoty z Grafu 3 (1,2 %). Tento rozdiel priamo odráža skutočnosť, že pri absencii grafovej cesty je systém plne závislý od FAISS indexu, ktorého obsah odzrkadľuje stav legislatívy v čase jeho zostavenia.



Obr. 10 Vyhodnotenie úspešnosti odpovedí systému pri využití výlučne vektorovej databázy

8.3. Diskusia výsledkov

Výsledky hodnotenia umožňujú identifikovať situácie, v ktorých systém funguje spoľahlivo, a situácie, kde jeho výkon zaostáva za očakávaniami.

Kľúčovým zistením je porovnanie Grafu č. 3 a Grafu č. 4, teda skupiny otázok spracovaných s grafovou cestou oproti skupine otázok spracovaných bez nej. Pri 82 otázkach s aktívnou grafovou cestou dosiahol systém 65,9 % správnych odpovedí a 20,7 % nesprávnych. Pri 28 otázkach spracovaných výlučne cez vektorovú databázu klesol podiel správnych odpovedí na 32,1 % a podiel nesprávnych vzrástol na 39,3 %. Rozdiel je výrazný – pri otázkach bez grafovej podpory systém odpovedal správne takmer o tretinu menej často a nesprávne takmer dvojnásobne častejšie. Grafová databáza obsahuje autoritatívne definície právnych pojmov priamo naviazané na konkrétne zákony, čo modelu poskytuje presne štruktúrovaný kontext, ktorý vektorové vyhľadávanie nedokáže plnohodnotne nahradiť. Výsledky teda potvrdzujú, že integrácia znalostného grafu do RAG pipeline zlepšuje kvalitu odpovedí pri právnych otázkach, a to merateľným spôsobom.

Silnou stránkou systému je jeho schopnosť presne odpovedať na terminologické otázky, kde sa v texte otázky nachádza právny pojem zastúpený v tezaure. V takýchto prípadoch grafová cesta

identifikuje pojem s vysokou hodnotou istoty, poskytne jeho presnú definíciu a zároveň cituje príslušný právny predpis priamo z grafu. Práve tieto otázky tvoria jadro prípadov zahrnutých do Grafu č. 3 a vysvetľujú, prečo sú výsledky v tejto podskupine výrazne lepšie ako v Grafe č. 4. Grafová databáza tu plní funkciu, ktorú opísala teoretická časť práce – poskytuje explicitne štruktúrované znalosti, ktoré jazykovému modelu umožňujú odpovedať presnejšie a s jasne identifikovateľným zdrojom.

Slabou stránkou systému je neaktuálnosť časti citovaných zdrojov, čo ilustruje najmä tretia kategória v Grafe č. 2 (20,0 % zdrojov). Porovnanie Grafu č. 3 a Grafu č. 4 tento problém ďalej špecifikuje. Kým v skupine s aktívnou grafovou cestou bola neaktuálna len 1,2 % odpovedí, v skupine bez grafu to bolo 10,7 %. Tento problém priamo vyplýva zo statickej povahy FAISS indexu – akonáhle je zostavený, neodráža následné zmeny v legislatíve. Slovenské právo sa pritom dynamicky vyvíja a zákony sú pravidelne novelizované. Systém v 22 prípadoch citoval staršie znenie zákona, ktoré síce pokrývalo danú problematiku, no bolo medzičasom nahradené. Toto obmedzenie zodpovedá problematike neaktuálnosti dát, ktorá bola identifikovaná ako typická slabina RAG systémov v právnej oblasti. Riešením by bola pravidelná aktualizácia vektorového indexu, ideálne automatizovaná na základe sledovania zmien v Zbierke zákonov.

Ďalšou identifikovanou slabinou je tendencia systému k čiastočne správnym odpovediam pri komplexných otázkach. Zákon pri určitom práve alebo povinnosti uvádza viacero podmienok, výnimiek alebo alternatív, pričom systém niekedy pokryje len ich časť. Príčinou je obmedzená veľkosť kontextu odovzdávaného jazykovému modelu – relevantné ustanovenia môžu byť rozptýlené v rôznych častiach dokumentu a nie všetky sa vmestia do limitovaného kontextového okna. Spolu tretia a štvrtá kategória odpovedí v Grafe č. 1 tvoria 7,3 % prípadov, čo naznačuje, že ide o opakujúci sa, hoci nie dominantný problém.

Výsledky hodnotenia zdrojov (Graf č. 2) ukazujú, že správnosť identifikácie zdroja (47,3 %) je nižšia ako správnosť samotnej odpovede (57,3 %). Systém teda v určitých prípadoch dokáže poskytnúť vecne správnu informáciu, no nepriradí k nej správny zákon. To sa typicky stáva pri otázkach spracovaných výlučne cez FAISS, kde sú zdroje odvodené z metadát textových fragmentov a nie z explicitných väzieb v grafe. Táto nezrovnalosť zdôrazňuje dôležitosť grafovej cesty nielen pre kvalitu odpovede, ale aj pre spoľahlivosť citovania zdrojov. Zároveň poukazuje na to, že rozšírenie grafovej databázy o väčší počet pojmov by mohlo priniesť ďalšie zlepšenie nielen v správnosti odpovedí, ale aj v presnosti identifikácie zdrojov.

Výsledky tiež naznačujú, že podiel 28,2 % nesprávnych zdrojov nie je zanedbateľný a v reálnom nasadení systému by mohol viesť k situáciám, keď používateľ dostane vecne správnu informáciu

podloženú nesprávnym zákonom. To je z hľadiska právnej praxe problematické – právne poradenstvo musí byť nielen vecne správne, ale aj podložené konkrétnym a platným právnym predpisom. Tento aspekt potvrdzuje, že navrhnutý systém je vhodný predovšetkým ako pomocný nástroj na orientáciu v právnych otázkach, nie ako náhrada odborného právneho poradenstva.

Celkovo možno konštatovať, že systém dosahuje uspokojivú mieru správnosti – viac ako polovica všetkých odpovedí bola vyhodnotená ako úplne správna a pri zapojení znalostného grafu toto číslo rastie na takmer dve tretiny. Zároveň výsledky poukazujú na priestor pre zlepšenie, najmä v oblasti aktuálnosti vektorového indexu, presnosti identifikácie zdrojov a schopnosti syntetizovať odpovede vyžadujúce zohľadnenie viacerých právnych ustanovení naraz.

Záver

Cieľom tejto diplomovej práce bolo navrhnúť a implementovať systém na vyhľadávanie právnych informácií zo slovenskej legislatívy s využitím znalostných grafov a jazykových modelov. Navrhnutý systém predstavuje hybridného právneho asistenta postaveného na architektúre Graph RAG, ktorý kombinuje grafovú databázu Neo4j s vektorovou databázou FAISS a jazykovým modelom Google Gemini.

V rámci práce bol zostavený znalostný graf zo siedmich tematických právnych tezaurov pokrývajúcich hlavné oblasti slovenského práva. Graf obsahuje 2 781 právnych pojmov s ich definíciami, vzájomnými vzťahmi a odkazmi na konkrétne právne predpisy a predstavuje štruktúrovanú vedomostnú bázu, ktorú systém využíva ako primárny zdroj kontextu pri odpovedaní na právne otázky.

Hlavným prínosom práce je empirické potvrdenie hodnoty znalostného grafu v kontexte právneho RAG systému. Vyhodnotenie na súbore 110 právnych otázok, ktoré nezávisle posúdila právnička, ukázalo, že pri otázkach spracovaných s využitím grafovej cesty dosiahol systém 65,9 % úspešnosť správnych odpovedí. Pri otázkach spracovaných výlučne prostredníctvom vektorovej databázy klesla úspešnosť na 32,1 %. Tento rozdiel jednoznačne potvrdzuje, že integrácia explicitne štruktúrovaných znalostí do RAG pipeline má merateľný a prakticky významný prínos. Systém zároveň potvrdzuje, že znalostný graf zvyšuje spoľahlivosť citovania zdrojov. Grafová cesta umožňuje priamo priradiť pojem k príslušnému právnomu predpisu prostredníctvom explicitnej väzby, čo vektorové vyhľadávanie, odkázané na metadáta textových fragmentov, nedokáže rovnako spoľahlivo zabezpečiť.

Medzi identifikované slabé miesta patrí predovšetkým neaktuálnosť vektorového indexu, ktorá sa prejavuje citovaním starších znení zákonov. Tento problém je typický pre RAG systémy pracujúce so statickými indexmi a jeho riešením by bola pravidelná, ideálne automatizovaná aktualizácia indexu sledujúca zmeny v Zbierke zákonov. Ďalšou slabinou je nižšia schopnosť syntetizovať odpovede na komplexné otázky vyžadujúce zohľadnenie viacerých právnych ustanovení súčasne, čo súvisí s obmedzenou veľkosťou kontextového okna odovzdávaného jazykovému modelu.

Do budúcnosti sa ako sľubné rozšírenie javí rozšírenie znalostného grafu o ďalšie právne oblasti a priame prepojenie pojmov s aktuálnymi plnými zneniami právnych predpisov, čím by sa znížila závislosť od vektorového indexu. Zaujímavou oblasťou je aj automatizovaná extrakcia nových vzťahov z legislatívnych textov pomocou metód spracovania prirodzeného jazyka, ktorá by umožnila udržiavať graf aktuálny bez manuálneho zásahu. Navrhnutý systém predstavuje funkčný základ hybridného Graph RAG prístupu v právnej doméne a vytvára priestor pre tieto ďalšie rozšírenia.

Zoznam použitej literatúry

- [1]. HOGAN, Aidan et al. : Knowledge Graphs. [online]. ACM Computing Surveys, Vol. 54, No. 4, Article 71, 2021. [cit. 2025-10-03]. Dostupné na internete: <https://aidanhogan.com/docs/knowledge-graphs-computing-surveys.pdf>
- [2]. WANG, Chenguang et al. : Language Models are Open Knowledge Graphs. [online]. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020. [cit. 2025-10-04]. Dostupné na internete: <https://aclanthology.org/2020.emnlp-main.550.pdf>
- [3]. PAN, James et al. : Vector Database Management Systems: Fundamental Concepts, State-of-the-art Systems, and Future Directions. [online]. The VLDB Journal, 2024. [cit. 2025-10-04]. Dostupné na internete: <https://dbgroup.cs.tsinghua.edu.cn/lvgl/papers/vldb2024-vectordb.pdf>
- [4]. WEI, Haixun et al. : A Survey on Large Language Models for Knowledge Graphs. [online]. ACM Transactions on Knowledge Discovery from Data, 2025. [cit. 2025-10-04]. Dostupné na internete: <https://dl.acm.org/doi/epdf/10.1145/3771791>
- [5]. ANGLES, Renzo et al. : Foundations of Modern Query Languages for Graph Databases. [online]. ACM Computing Surveys, 2018. [cit. 2025-10-10]. Dostupné na internete: <https://homepages.inf.ed.ac.uk/pguaglia/papers/sigmod18.pdf>
- [6]. VELIČKOVIĆ, Petar et al. : Graph Attention Networks. [online]. ICLR, 2018. [cit. 2025-10-10]. Dostupné na internete: <https://arxiv.org/pdf/1802.09984>
- [7]. GUPTA, S. et al. : Facilitating Knowledge Graph Management with gStore. [online]. CEUR Workshop Proceedings, Vol-2100, 2018. [cit. 2025-10-10]. Dostupné na internete: <https://ceur-ws.org/Vol-2100/paper26.pdf>
- [8]. BESTA, Maciej et al. : Graph Databases: A Survey. [online]. ETH Zurich, 2022. [cit. 2025-10-17]. Dostupné na internete: <https://htor.inf.ethz.ch/publications/img/graph-databases.pdf>
- [9]. CHENG, Yijian et al. : Which Category Is Better: Benchmarking Relational and Graph Database Management Systems. [online]. Data Science and Engineering, Vol. 4, No. 4, s. 309-322, 2019. [cit. 2025-10-17]. Dostupné na internete: <https://doi.org/10.1007/s41019-019-00110-3>
- [10]. ZOU, Lei et al. : gStore: A Graph-based SPARQL Query Engine. [online]. EDBT, 2016. [cit. 2025-11-02]. Dostupné na internete: <https://openproceedings.org/2016/conf/edbt/paper-333.pdf>

-
- [11]. ARENAS, Marcelo et al. : Querying Semantic Web Data with SPARQL. [online]. ACM Computing Surveys, 2017. [cit. 2025-11-02]. Dostupné na internete: <https://marceloarenas.cl/publications/csur17.pdf>
- [12]. BRIN, Sergey a Lawrence PAGE : The Anatomy of a Large-Scale Hypertextual Web Search Engine. [online]. Stanford University, 1998. [cit. 2025-11-20]. Dostupné na internete: <https://dl.acm.org/doi/pdf/10.1145/356770.356776>
- [13]. DAI, Y. et al. : Knowledge Graph Embedding: A Survey from the Perspective of Representation Learning. [online]. Applied Sciences (MDPI), 12(13), 6490, 2022. [cit. 2025-11-20]. Dostupné na internete: <https://www.mdpi.com/2076-3417/12/13/6490>
- [14]. BIALECKI, Andrzej et al. : Apache Lucene 4. [online]. ResearchGate, 2012. [cit. 2025-11-20]. Dostupné na internete: https://www.researchgate.net/profile/Andrzej-Bialecki/publication/260282732_Apache_Lucene_4/links/5ede6a2545851516e65f1fa6/Apache-Lucene-4.pdf
- [15]. BERNERS-LEE, Tim : Linked Data. [online]. W3C Design Issues, 2006 (aktualizované 2012). [cit. 2025-11-24]. Dostupné na internete: <https://asistdl.onlinelibrary.wiley.com/doi/pdf/10.1002/bult.2012.1720380413>
- [16]. PETERS, Matthew et al. : Deep contextualized word representations. [online]. Proc. of NAACL-HLT, 2018. [cit. 2025-11-24]. Dostupné na internete: <https://aclanthology.org/K17-3009.pdf>
- [17]. MILES, Alistair a Sean BECHHOFFER : SKOS Simple Knowledge Organization System Reference. [online]. W3C Recommendation, 2009. [cit. 2025-11-24]. Dostupné na internete: <https://www.w3.org/TR/skos-reference/>
- [18]. GOOGLE : Gemini API documentation: Long context. [online]. Google AI for Developers, 2024. [cit. 2025-12-03]. Dostupné na internete: <https://ai.google.dev/gemini-api/docs/long-context>
- [19]. REID, M team et al. : Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. [online]. arXiv:2403.05530 [cs.CL], 2024. [cit. 2025-12-03]. Dostupné na internete: <https://arxiv.org/pdf/2403.05530>
- [20]. ZAHEER, Manzil et al. : Big Bird: Transformers for Longer Sequences. [online]. NeurIPS, 2020. [cit. 2025-12-08]. Dostupné na internete: <https://arxiv.org/pdf/2004.04906>
- [21]. SHAZEER, Noam et al. : Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. [online]. arXiv:1702.08734 [cs.LG], 2017. [cit. 2025-12-08]. Dostupné na internete: <https://arxiv.org/pdf/1702.08734>
-

- [22]. DING, Yushuo et al. : LongRoPE: Extending LLM Context Window to 2M Tokens. [online]. arXiv:2402.05672 [cs.CL], 2024. [cit. 2025-12-08]. Dostupné na internete: <https://arxiv.org/pdf/2402.05672>
- [23]. VASWANI, Ashish et al. : Attention Is All You Need. [online]. NeurIPS, 2017. [cit. 2025-12-08]. Dostupné na internete: <https://arxiv.org/pdf/1706.03762>
- [24]. HE, Kaiming et al. : Deep Residual Learning for Image Recognition. [online]. CVPR, 2016. [cit. 2025-12-08]. Dostupné na internete: <https://arxiv.org/pdf/1603.09320>
- [25]. LEWIS, Patrick et al. : Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. [online]. Advances in Neural Information Processing Systems (NeurIPS), 2020. [cit. 2025-10-04]. Dostupné na internete: <https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf>
- [26]. ZHANG, J. et al. : Advanced Hybrid RAG Architectures. [online]. arXiv:2507.18910 [cs.CL], 2025. [cit. 2025-12-20]. Dostupné na internete: <https://arxiv.org/html/2507.18910v1>
- [27]. LIU, Y. et al. : Graph-Enhanced Large Language Models. [online]. arXiv:2505.00039v2 [cs.CL], 2025. [cit. 2025-12-20]. Dostupné na internete: <https://arxiv.org/html/2505.00039v2>
- [28]. CHEN, X. et al. : Optimizing Long-Context RAG. [online]. arXiv:2507.03226v3 [cs.CL], 2025. [cit. 2026-01-10]. Dostupné na internete: <https://arxiv.org/html/2507.03226v3>
- [29]. SMITH, A. et al. : The Future of Knowledge Graphs in 2026. [online]. arXiv:2509.21336v1 [cs.CL], 2025. [cit. 2026-01-10]. Dostupné na internete: <https://arxiv.org/html/2509.21336v1>
- [30]. EDGE, Darren et al. : From Local to Global: A GraphRAG Approach to Query-Focused Summarization. [online]. arXiv:2408.04948v1 [cs.CL], 2024. [cit. 2026-01-10]. Dostupné na internete: <https://arxiv.org/html/2408.04948v1>
- [31]. TAN, M. et al. : Scalable Knowledge Retrieval. [online]. ACM Transactions on Intelligent Systems and Technology, 2025. [cit. 2026-01-10]. Dostupné na internete: <https://dl.acm.org/doi/epdf/10.1145/3777378>
- [32]. YANG, H. et al. : GraphRAG: Unlocking LLM Potential for Complex Reasoning and Deep Knowledge Retrieval. [online]. Neural Computing and Applications, Springer, 2025. [cit. 2026-01-10]. Dostupné na internete: <https://link.springer.com/article/10.1007/s10489-025-06885-5>
- [33]. LIU, Y. et al. : Graph-Enhanced Large Language Models. [online]. arXiv:2505.00039 [cs.CL], 2025. [cit. 2026-01-10]. Dostupné na internete: <https://arxiv.org/pdf/2505.00039>

- [34]. WANG, L. et al. : Federated Knowledge Graph Learning. [online]. Proceedings of the 30th ACM SIGKDD, 2024. [cit. 2026-01-19]. Dostupné na internete: <https://dl.acm.org/doi/epdf/10.1145/3677052.3698671>
- [35]. ZHAO, K. et al. : Massive Context Window Optimization. [online]. arXiv:2507.03608 [cs.CL], 2025. [cit. 2026-01-19]. Dostupné na internete: <https://arxiv.org/pdf/2507.03608>

Prílohy

Príloha A: CD médium