LEGALLY SPEAKING

Developing Explainable AI Approaches for Legal Decision-Making

by

Cihan Erdoğanyılmaz

Abdelrahman Zahran

Mohamad Nael Ayoubi

CSE4197 Engineering Project 1

Project Specification Document

Supervised by:

Assoc. Prof. Ali Haydar Özer

Marmara University, Faculty of Engineering

Computer Engineering Department

14.11.2024

Table of Contents

1. PROBLEM STATEMENT

In the field of natural language processing in law, large language models (LLMs) often fall short of expectations due to issues like hallucinations, lack of domain-specific training, and the "black box" nature of their reasoning processes. This project introduces a novel explainable AI and NLP approach, termed "Law of Thought Tree (LoTT)," which aims to directly guide the reasoning process of LLMs in legal decision-making. The approach not only generates legal outcomes but also provides the underlying legal reasoning tailored to the specific input provided by the user. It is designed to handle complex legal problems with high performance and transparency. Furthermore, this approach plans to create artificial legal intelligence by introducing a new set of methods to directly transfer the mind process of a legal professional to LLMs.

2. PROBLEM DESCRIPTION AND MOTIVATION

In recent years, artificial intelligence (AI) has demonstrated remarkable potential across sectors such as healthcare, finance, and law. High-performing models have been developed in lots of domains, driving innovation and efficiency. However, in fields that directly affect human life, freedom, and justice, particularly law—success cannot be measured by performance metrics alone. How a model arrives at its outputs is as critical as the accuracy of the results.

However, the black-box nature of many AI models, particularly those with billions of parameters, poses a significant challenge. As these models increasingly shape our decisions and society, their opacity can leave crucial areas like law vulnerable to unpredictable and unjust outcomes. This challenge emphasizes the need for transparency, interpretability, and trustworthiness in AI systems. Without explainable AI (XAI), we risk losing the transparency that is fundamental to the rule of law, while drifting toward a future reminiscent of dystopian science fiction.

In addition, training with datasets that are relatively rare in the pre-training corpus of large language models such as Turkish law is not sufficient and poses a problem not only for the explainability of the models but also for the models to produce high performance outputs. This is because the training set of the model does not have the knowledge to solve the problem in question or the reasoning capabilities of LLMs are not yet at a level that can directly solve complex legal cases.

The challenge, then, was to develop a new approach that could both provide a transparent legal decision-making process and combine the general reasoning capabilities of LLMs with the mind-flow process of lawyers, while achieving high performance even on complex problems. We addressed this problem with a new approach, Law of Thought Tree (LoTT), and made a bold invitation for LLMs to be used in legal systems.

We aim that this project will provide important academic outputs in the fields of natural language processing such as LLM planning, agentic Retrieval Augmented Generation, using and training domain-specific language models, and providing helpful tools to make the time spent by legal professionals highly productive.

3. MAIN GOAL AND OBJECTIVES

The primary goal of this project is to develop an Explainable AI (XAI) system specifically designed to aid legal professionals by simulating a human-like approach to reasoning in the analysis and prediction of legal case outcomes. This system, built around a decision-tree-inspired framework called the "Law of Thought" (LoT) tree, will allow users—judges, lawyers, and legal researchers—to engage with a logical, step-by-step path that leads toward potential judicial decisions in alignment with Turkish law. The interactive, structured approach embedded in the LoT trees will provide a transparent and systematic method for navigating legal reasoning, aligning each decision with established case precedents and legal principles.

To achieve this goal, the project will follow a series of objectives designed to support each phase of development and implementation:

- Collecting data through web scraping of public legal databases, including the Constitutional Court of Turkey's website, along with additional resources such as law theses from the YÖK National Thesis Centre, decisions from the Supreme Court of Appeals and Council of State, and legal journals from law faculties.
- Preprocessing and organizing the scraped data, extracting key elements (e.g., case

details, rulings, and reasoning) to build a dataset.

- Developing an intuitive graphical user interface (GUI) to assist in creating and visualizing the LoT tree structure. The GUI should allow us to define nodes and branches that represent the logical progression of legal reasoning in each case.
- To populate the LoT trees with case data, allowing us to construct training data by converting unstructured legal text into structured LoT trees manually. This GUI will help capture the hierarchical, decision-based structure of legal reasoning.
- To implement machine learning models for semantic search functionality, which can analyze input case descriptions (Legal Events) while trying to find the best match LoT tree, in addition to retrieving similar cases and their corresponding LoTs from the dataset based on meaning rather than keywords alone.
- To fine-tune large language models such as Llama 3.2 and Turkish BERT on the legal dataset to improve the understanding of domain-specific language, concepts, and decision patterns in Turkish law.
- Integrating the results of semantic search into prompts to guide the model in presenting similar cases, related laws, and judicial reasoning that align with the ongoing decision-making process within the LoT tree.
- To implement prompt engineering techniques that facilitate interaction between the LoT tree nodes and the language model, allowing each node's decision or conclusion to serve as an input for the next node in the reasoning chain.

4. RELATED WORK

4.1 Natural Language Processing Studies with Legal Corpus

Natural language processing studies with Turkish legal corpus have generally focused on the classification of court decisions. There are many studies that classify the decisions for predicting outcomes of the courts [1,2,3] with fundamental machine learning algorithms such as Decision Tree, Naive Bayes, Support Vector Machine, Logistic Regression and deep learning algorithms such as Feed Forward Neural Network, LSTM [4], GRU, Transformer [5]. Although there are relatively few studies that examine solutions to different problems such as semantic search [6], legal text generation [7], gender bias prevention in legal corpus [8], many studies in the field of natural language processing in law have focused on the classification problem. This is also the common trend in studies that are not conducted in Turkish legal corpus. There have been many studies on predicting the decisions of the European Court of Human Rights [9,10] or the supreme courts of countries [11,12] through classification studies. In our study, however, the focus is not only on predicting the class to which a court decision or any legal text belongs, but also on the process of generating a complex legal text, i.e. the justification activity performed directly by the judge. In addition, in order to support the LoTT approach we have developed, new approaches have been designed to solve many sub-problems such as semantic search and classification.

4.2 Explainable Artificial Intelligence in Natural Language Processing with Legal Corpus

The previously mentioned studies have focused on classification performance with F1, precision, recall or accuracy metrics, but not on model explainability. In the Turkish legal natural language processing literature, the only study on the numerical calculation of the explainability of models is a study by Erdoğanyılmaz et al. [2] on predicting the outcomes of Constitutional Court decisions with a binary classification problem.

In this study, the authors identified the word groups that encoder-based Transformer models, pre-trained on Turkish corpora, focused on when classifying court decisions. Using SHAP values, they analyzed these word groups and calculated their similarity to the word groups that lawyers emphasized. They then proposed a new metric, which they called "explainability score." There is also a study in the literature on the classification of Constitutional Court decisions that extracts the parts of the decision text that affect the label predicted by the model through SHAP [3]. In our approach to explainability, not only the word groups that affect the model's decision, but also the relationship between these word groups can be examined; in addition, unlike the solutions of ready-available libraries such as SHAP and LIME, a new explainable artificial intelligence approach is introduced to the literature of natural language processing in law.

4.3 Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

The Chain of Thought (CoT) prompting approach [13] by Google Research's Brain Team improves reasoning in large language models (LLMs) by breaking down complex tasks into sequential steps, enhancing performance and transparency across tasks like arithmetic and commonsense reasoning. However, CoT has key limitations: (1) it lacks the ability to explore multiple reasoning paths, and (2) it does not incorporate global planning, meaning it cannot look ahead or revise decisions based on alternative paths. Our project introduces the "Law of Thought" (LoT) tree, a novel, structured framework explicitly designed for the demands of legal reasoning within Turkish law. Unlike CoT's linear approach, the LoT tree models legal analysis as a decision tree, where each node represents a specific legal principle or a ruling for the subproblem. This setup enables structured, legally grounded paths that users can follow transparently, supporting domain-specific reasoning with both depth and clarity.

4.4 Tree of Thoughts: Deliberate Problem Solving with Large Language Models

The Tree of Thoughts (ToT) framework [14] enhances general problem-solving by allowing large language models (LLMs) to explore diverse paths through a structured, tree-based system, generating and evaluating multiple thought steps at each node to select the most promising branch as shown in Figure 1. ToT's internal processing relies on heuristics and state evaluations to backtrack and optimize reasoning paths based on feasibility. In contrast, our "Law of Thought" (LoT) tree is specifically designed for legal reasoning within Turkish law, prioritizing transparency and legal solidarity. Each LoT node represents legal constructs such as case precedents or procedural rules, sequentially guiding the model through a decision-making process grounded in legal logic. Instead of heuristics, our project utilizes semantic search to retrieve legally relevant LoTs and principles.

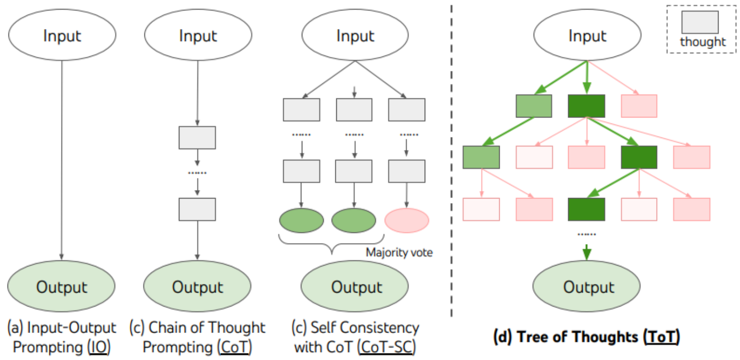

Figure 1: Schematic illustrating various approaches to problem solving with LLMs. Each rectangle box represents a thought, which is a coherent language sequence that serves as an intermediate step toward problem solving [14].

5. PROJECT SCOPE

This project aims to design and implement an Explainable AI (XAI) system specifically to enhance the decision-making capabilities of legal professionals in Turkey by providing transparent and interpretable insights into legal case outcomes. The system will simulate human-like reasoning and offer clear, step-by-step explanations for each decision suggestion. Focused on the Turkish legal system, where transparency and alignment with legal precedents are critical, the system addresses the complexity of legal decisions that involve stable law interpretations and case-specific factors. It bridges the gap between AI-driven insights and human legal expertise by ensuring that every prediction is accompanied by a logical, explainable path. The project is built on three core pillars: first, **Transparency**, through the use of decision-tree structures, known as the "Law of Thought" (LoT) trees, which break down legal reasoning into logical nodes and offer clear decision pathways; second, **Structured Reasoning**, where the LoT tree framework mimics the deductive reasoning of legal professionals, ensuring AI decisions align with established legal principles in Turkey; and third, **Relevance to Turkish Legal Precedents**, with the system trained on Turkish legal data, including court rulings, statutes, and judicial reasoning patterns, to ensure AI-generated outcomes reflect Turkish legal practices.

5.1 *Out of Scope*

- **Cross-Jurisdictional Application**: The AI model will be customized to Turkish law and is not intended for other jurisdictions. It is highly relevant to Turkish law while excluding the complexities associated with multi-jurisdictional support.
- **Decision-Making Authority**: The system will support, not replace, legal decision-making by offering structured insights and offers semi-final or final rulings.

5.2 *Dependencies on Other Projects*

This project does not depend on other active projects but requires access to publicly available Turkish legal data for training.

5.3 *Constraints & Limitations*

- **Data Access and Quality**: The quality and availability of Turkish legal data may impact the model's performance. Incomplete or unstructured data could affect predictions and reasoning quality.
- **Computational Resources**: The project requires significant computational power for training language models and semantic search, with limited resources potentially affecting performance and scalability.
- **Legal and Ethical Constraints**: The project must comply with Turkish data protection laws, ensuring the secure handling of sensitive legal data.
- **Complexity of Legal Reasoning**: The LoT tree framework simplifies legal reasoning, which may not fully capture the depth of complex cases with subjective elements.
- **Explainability Challenges**: While decision trees enhance transparency, large language models often exhibit "black-box" characteristics. Some model behaviors and reasoning may remain opaque in some reasoning processes, which might reduce interpretability.

5.4 *Project Assumptions – It's Assumed –:*

- **Availability of Legal Data**: Sufficient and high-quality Turkish legal data, including case rulings and legal documents, will be available for training & fine-tuning.
- **Computational Resources**: Adequate computational resources will be available for

training and real-time operation of LLMs and semantic search functions.

- **Legal Compliance**: All data sources will comply with Turkish legal and ethical standards.
- **Compatibility with Legal Procedures**: The LoTT framework will align with Turkish legal procedures, enabling seamless integration into current legal workflows.

6. METHODOLOGY AND TECHNICAL APPROACH

In this project we are going to develop an Explainable AI (XAI) system, using a structure inspired by decision trees, called the "Law of Thought" (LoT) tree, to simulate legal reasoning and predict judicial outcomes in Turkish law. By applying data scraping, preprocessing, machine learning, agentic retrieval-augmented generation and fine-tuning, our methodology comprises the following steps:

6.1 *Data Collection & Preprocessing*

We will systematically collect Turkish legal data from public sources such as the Constitutional Court, Supreme Court, and Council of State websites, as well as law theses and journals. Tools like Selenium, BeautifulSoup, and Requests will facilitate data extraction, focusing on diverse case types and judicial reasoning to ensure the model's generalizability. The gathered data will undergo preprocessing to structure it for LoT tree integration. Key steps include parsing case details, tokenizing and cleaning text, and organizing information in JSON format by subfields (e.g., case type, event, and outcome) to maintain consistency.

6.2 *Development of Law of Thought (LoT) Tree GUI and LoTT Framework*

This phase is dedicated to creating a tool that visualizes and populates the LoTTs, which structures pathways in legal decision-making as shown in the example in Figure 2. The methodology is divided into two key components: implementing the graphical user interface (GUI) and developing the LoTT framework.

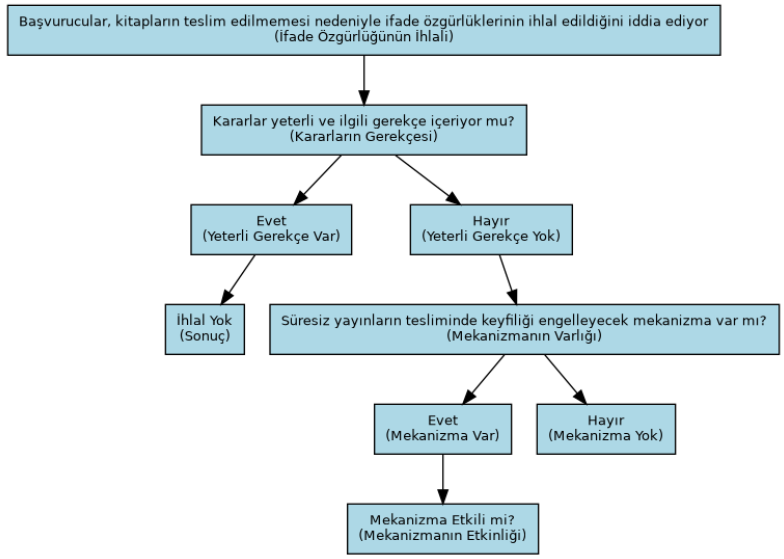

Figure 2: A Law of Thoughts Tree Example

The GUI is designed for intuitive navigation, using user-centered design principles with prototyping and wireframing to ensure accessibility. Through D3.js or similar libraries, the interface visualizes the LoTT with color-coded nodes representing legal concepts, arrows for logical dependencies, and interactive features like node expansion, modification, and drag-and-drop rearrangement. A built-in LoT Population Tool enables users to add new legal cases or rules to the tree, categorizing nodes by logical type and legal domain.

The LoTT framework organizes legal decision paths in a directed acyclic graph (DAG) structure. Each node represents a subproblem with its ruling, connected through logical dependencies to form a hierarchical pathway. This setup allows legal professionals to add and explore LoTTs, segmenting cases into subproblems and rulings. Integrated with semantic search and large language model (LLM) modules, the framework supports context-aware analysis and reasoning-based explanations. A dedicated database ensures efficient storage and retrieval, keeping LoT Trees accessible and adaptable.

6.3 *Semantic Search Implementation*

After populating the LoT trees, we will implement a hybrid semantic search that combines structured LoT tree reasoning with context-based case retrieval. Initially, the data will be indexed and vectorized in a high-performance vector database like FAISS [15]. Each legal case and its LoT tree will be converted into embeddings (Figure 3) using models such as Turkish BERT [16], fine-tuned with sentence-transformers [17] training objective with legal contexts. This vectorization captures both semantic and logical relationships, enabling similarity-based matching beyond simple keywords.

For the search functionality, two approaches may be explored: (1) classifying the input case into legal categories before searching, or (2) directly applying semantic search to identify the most similar LoTTs. We will use approximate nearest neighbor (ANN) algorithms for fast retrieval, initially extracting the top matches, which are then refined by a cross-encoder for enhanced relevance. This hybrid approach of ANN with cross-encoding balances speed with accuracy, providing results that reflect both semantic similarity and logical reasoning within the LoT trees.

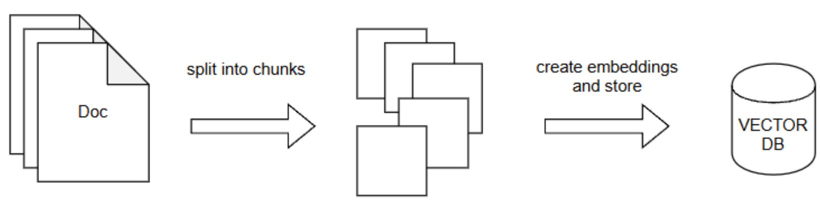

Figure 3: An illustration of splitting training data and LoTs into chunks and storing it in a vectorized database such as FAISS.

6.3.1 Bidirectional Encoding and Cross-Encoder Ranking

The search pipeline leverages a two-stage ranking: a bidirectional encoder and a cross-encoder. The bidirectional encoder model generates dense vector embeddings of the query and LoT trees as shown in its architecture in Figure 4, allowing efficient retrieval through ANN search. The cross-encoder then jointly analyzes the query with the top results to refine relevance, assigning similarity scores to rank the LoTTs precisely by contextual fit. This two-step process ensures that the results capture not only surface-level similarity but also the deeper reasoning paths that are essential in legal contexts.

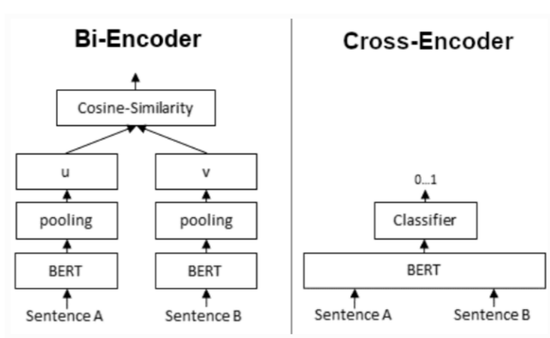

Figure 4: The architecture of Bi-Encoder and Cross-Encoder [17].

6.3.2 Passing through Internal Nodes and Similarity Scores

In our search pipeline, queries first match the root nodes of the LoT trees to identify the most similar overall reasoning paths. As the query passes through internal nodes, it retrieves details aligned with specific case details, ensuring results reflect both high-level legal LoTT structure and relevant case-specific information. For similar scoring, we employ metrics like

- cosine similarity (measuring vector alignment for semantic closeness):

$$\text{Cosine Similarity}(V_1, V_2) = \frac{V_1 \cdot V_2}{\|V_1\| \|V_2\|}$$

- Euclidean distance (calculating straight-line similarity):

$$\text{Euclidean Distance}(V_1, V_2) = \sqrt{\sum_{i=1}^{n} (V_{1,i} - V_{2,i})^2}$$

- dot product (effective with normalized vectors):

$$\text{Dot Product}(V_1, V_2) = V_1 \cdot V_2 = \sum_{i=1}^{n} V_{1,i} V_{2,i}$$

Vector databases such as FAISS apply these in the first stage of ANN search to quickly filter top matches. In the second stage, a cross-encoder then refines these results by ranking them for precise contextual alignment, balancing speed and depth in retrieving relevant LoT paths and case details.

6.4 *Fine-tuning Language Models*

To enhance the model's understanding of Turkish legal terminology and reasoning, we will fine-tune large pre-trained models, such as Turkish BERT [16] and LLaMA 3.2 [18], using a custom legal dataset. Fine-tuning enables these models to adapt to complicated legal reasoning and specialized terminology, improving their accuracy in responding to legal queries. To manage the high computational and memory requirements of fine-tuning large models, we plan to use various Parameter-Efficient Fine-Tuning (PEFT) techniques. By updating only a targeted subset of parameters, PEFT methods allow us to maintain computational efficiency without sacrificing the model’s adaptability. Techniques like Low-Rank Adaptation (LoRA) [19], for example, use low-rank matrices to adjust only specific model components, such as attention layers, while leaving the base model weights unchanged. This approach is particularly useful for capturing legal reasoning patterns with limited computational resources.

In addition to LoRA, we may evaluate other PEFT approaches, such as Adapters [20] (which add small task-specific modules within the model layers). These methods provide flexibility in tuning various aspects of the model, and their performance may be assessed to identify the most effective approach for capturing the complexity of Turkish legal contexts. Our dataset, which includes structured data on case outcomes, reasoning (LoTTs), and legal references, will support the model's ability to learn nuanced, context-specific decision-making logic.

6.5 *Fine-tuning Embedding Model*

In the training phase of embedding models, we may utilize the Triplet Objective Function. This method introduces an anchor sentence s_a, a positive sentence (closely related sentence) s_p, and a negative sentence (less relevant sentence) n, aiming to make the distance between a and p smaller than the distance between a and n. This is represented mathematically by minimizing the following triplet loss function:

$$\text{Triplet Loss} = \max\left(\|s_a - s_p\| - \|s_a - s_n\| + \epsilon, 0\right)$$

In addition, we may use alternative functions such as Multiple Negative Ranking Loss

$$\text{MN Ranking Loss} = -\log\left(\frac{\exp(\text{Cosine Similarity}(s_a, s_p))}{\sum_{i=1}^{N}\exp(\text{Cosine Similarity}(s_a, s_{n_i}))}\right)$$

For scenarios with many irrelevant candidates, Matryoshka Loss for scalable embeddings at different levels of similarity

$$\text{Matryoshka Loss} = \sum_i \max\left(0, d_i(s_a, s_p) - d_i(s_a, s_n) + \epsilon_i\right)$$

Or Cached Multiple Negatives Ranking Loss to efficiently store and access negative samples, improving computational efficiency.

6.6 *Integration of Semantic Search Results with Model Prompts*

Our initial plan for text generation and node processing is to use commercial LLM APIs with advanced reasoning capabilities, and later our plan is to develop custom models tailored to our needs. We employ a Retrieval-Augmented Generation (RAG) approach to facilitate smooth transitions between subproblems and internal nodes within the LoT trees. RAG dynamically retrieves details of similar legal cases based on the user query, leveraging in-context learning (few-shot learning) to embed relevant case information directly into model prompts. This approach ensures that each step in the reasoning chain remains contextually grounded and logically consistent. Additionally, we may incorporate an agentic processing approach to enhance the model's autonomy within each node, particularly in complex reasoning paths. Through frameworks like Microsoft's MultiGen [28], agentic processing allows the model to independently assess subproblems, verify logical consistency, and choose the most relevant next node based on the evolving context. This enables the model to re-evaluate decisions and dynamically retrieve necessary information, ensuring reliable progression through the LoTT.

6.7 *Evaluation and Performance Testing*

The evaluation of the XAI system will involve both human expert evaluation and standard metrics for retrieval and model fine-tuning. For human evaluation, we plan to use a system like Elo-rating. We will assess how well the LoT tree's predictions align with real-world legal decisions and the clarity of its explanations. For text generation and explanations, we will use BLEU and ROUGE to evaluate the quality of the generated content. For retrieval, we will apply precision, recall, and F1 score, along with cosine similarity thresholds to measure retrieval performance. Additionally, we will use Spearman, Pearson, Manhattan, and dot product correlations to evaluate the alignment and relationships between query and case – LoTTs embeddings.

7. PROFESSIONAL CONSIDERATIONS

7.1 Methodological Considerations / Engineering Standards

- Project Planning: Gantt charts will guide timelines and milestones, ensuring clear task management.

- System Design: UML diagrams, including class and sequence diagrams, will represent the architecture, especially the LoTT and similar cases integration with RAG and semantic search.
- Development Tools: Our primary development will use Python with deep learning libraries like PyTorch and Hugging Face Transformers. Google Colab will provide cloud-based GPU and TPU resources for efficient model training and testing. We will use Git and GitHub for version control to manage and collaborate on code. For fast similarity searches within our legal dataset, FAISS will serve as the primary vector database, though alternatives like Pinecone may also be evaluated. Finally, we'll build the web interface with Flask, leveraging its REST API support to seamlessly integrate model outputs and retrieval functionalities.

7.2 Realistic Constraints

- Economic: Costs will be managed by optimizing API usage, data processing, and cloud resources.
- Environmental: Resource-efficient models and sustainable cloud practices will reduce environmental impact.
- Ethical: The LoT Tree will maintain unbiased, transparent legal reasoning to avoid misleading users and uphold ethical standards.
- Health and Safety: Thorough testing will ensure reliability due to the project's influence on legal interpretation.
- Sustainability: Scalable and adaptable design will ensure long-term usability with minimal upkeep.
- Social Impact: The project aims to improve public access to clear legal reasoning, building trust in AI-based legal tools.

7.3 Legal Considerations

- Licensing and API Compliance: We will ensure adherence to licensing and API requirements for commercial LLMs and third-party databases.
- Data Permissions: Access to legal databases will be secured with proper permission and regulatory compliance.
- Ethical Compliance: Ethical guidelines will be upheld, focusing on privacy and

unbiased data handling to support responsible AI in legal contexts.

8. MANAGEMENT PLAN

8.1 Project Phases, Sub-tasks, and Durations

1. Data Collection (few weeks or a month)

- Use web scraping tools (Selenium, BeautifulSoup, Requests) to access Turkish legal databases (e.g. Constitutional and Supreme Courts) and extract case data like summaries, rulings, and reasoning.

2. Data Preprocessing (approximately 1 month)

- Parse, clean, and tokenize case elements, standardizing terms for consistency, then structure data in JSON format for LoT tree compatibility, with quality checks.

3. Development of GUI and LoT Tree Framework (around 2 months)

- Design a user-centered GUI using JavaScript libraries for interactive LoTT visualization, supported by a Flask backend for data processing and APIs, and implement the LoTT framework as a directed acyclic graph (DAG) with features like node editing, ruling branches, and drag-and-drop functionality.

4. LoT Tree Population (couple weeks)

- Populate LoT trees with structured case data, defining nodes and branches to map reasoning paths, followed by testing and manual review to ensure accuracy in case decision logic.

5. Semantic Search Development (1 month)

- Use vector database such as FAISS for vector indexing of case data and LoTT, with bidirectional encoding and cross-encoder ranking for accurate query matching.

6. Fine-Tuning Language Models (3 weeks)

- Prepare legal dataset and fine-tune LLMs such as Turkish BERT and LLaMA 3.2 models using PEFT techniques.
- Test, evaluate, and refine models to enhance accuracy in legal terminology and reasoning patterns.

7. Integration of Semantic Search with Model Prompts (1 month)

- Implement RAG and design prompt engineering strategies for smooth transitions between LoTT nodes.

- Develop and test agentic processing for autonomous node selection and logical consistency.

8. Evaluation and Performance Testing (1 month)

- Conduct accuracy testing of case retrieval and LoT path alignment with queries, evaluate reasoning transparency, and compile a final report with performance metrics and recommendations.

8.2 Division of Responsibilities and Duties

Division of responsibilities and duties are given below in Figure 5.

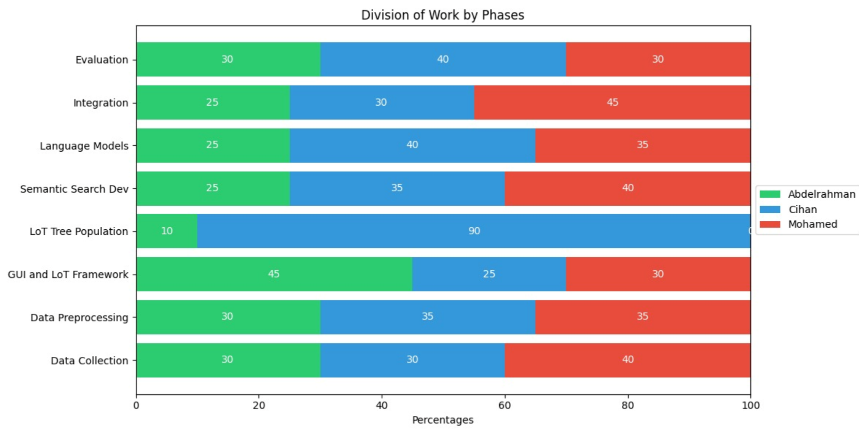

Figure 5: Division of Responsibilities.

8.3 Timeline with Milestones (Gantt Chart Outline)

Timelines with milestones are given below in Figure 6.

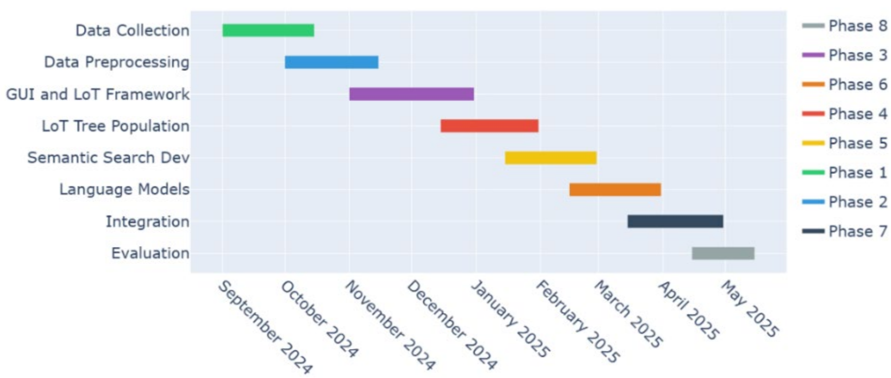

Figure 6: Timeline with Milestones.

9. SUCCESS FACTORS AND RISK MANAGEMENT

9.1 Measurability/Measuring Success

The success of this project will be measured using key performance indicators (KPIs) for each objective. Below are the success factors for each objective mentioned in Section 3.

1. **Success Factor:** Gather at least 10,000 diverse legal documents with comprehensive case details, balancing databases to support accurate legal decision-making.
2. **Success Factor:** Achieve an 85% success rate in extracting case details and reasoning, with fewer than 5% incomplete or misclassified documents, verified by random sampling.
3. **Success Factor:** The interface should support LoT tree creation with zero crashes or critical errors over a minimum of 100 trials.
4. **Success Factor:** Create manually 100 LoT trees with 90% structural accuracy, verified by cross-checks with legal professionals.
5. **Success Factor:** Ensure around 80% recall and 85% accuracy in similar case retrieval, validated on a 1,000-case test set.
6. **Success Factor:** Fine-tune models to improve Turkish legal concept comprehension by 10%, measured by domain-specific benchmarks.
7. **Success Factor:** Integration of semantic search results into prompts should enable accurate case recommendations with a precision rate of 80% a balanced F1 score in returning the most similar LoTT in root node and similar case details in internal nodes and related laws. This will be verified using a diverse dataset of legal scenarios.
8. **Success Factor:** Maintain around 90% logical consistency in passing through correct LoTT internal nodes, reviewed by legal experts over 100 test cases.

9.2 Risk Management

Here is a list of potential risks, the associated phases of the project, and resolution strategies (B-plans) to manage them.

Risk – Phase 2: Difficulty in data extraction due to complex legal document formats

- **Resolution (B-Plan)**: Implement additional preprocessing steps, such as natural language processing (NLP) parsing and manual annotation where needed.

Risk – Phase 4: Manual LoT tree construction may be time-consuming and error-prone

- **Resolution (B-Plan): Introduce** automated or semi-automated LoTT generation techniques to support and speed up manual tree construction.

Risk – Phase 1 and 5: Semantic search models might underperform due to limited domain-specific data

- **Resolution (B-Plan)**: Consider domain adaptation techniques, such as additional fine-tuning on larger or synthetic datasets, to improve performance.

Risk – Phase 6: Insufficient Turkish legal language understanding by pre-trained language models (e.g., BERT)

- **Resolution (B-Plan)**: Experiment with specialized Turkish legal language datasets and model adjustments to enhance domain comprehension.

10. BENEFITS AND IMPACT OF THE PROJECT

This project holds transformative potential for the legal and AI fields. It directly benefits legal professionals, including judges, lawyers, and legal researchers—by providing a decision support tool offering transparent, machine-guided reasoning that mirrors human legal analysis.

i. **Scientific Impact:**

The project aims to advance explainable AI (XAI) and NLP by embedding structured legal reasoning into AI, enhancing transparency. We plan to publish our findings, introducing new methods for domain-specific explainability.

ii. **Economic/Commercial/Social Impact:**

This project has strong commercial potential in legal decision support and addresses the need for transparent, ethical AI in law, reducing concerns over "black-box" models.

iii. **Potential Impact on New Projects:**

The LoTT model's approach may inspire future projects in areas that require clear, domain-specific AI reasoning, such as healthcare and finance. This project could serve as a model for creating explainable AI.

iv. **Impact on National Security:**

While national security is not the primary focus, this project contributes to a fairer and more transparent justice system.

11. REFERENCES

[1] Mumcuoğlu, E., Öztürk, C. E., Ozaktas, H. M., & Koç, A. (2021). Natural language processing in law: Prediction of outcomes in the higher courts of Turkey. *Information Processing & Management*, *58*(5), 102684.,

[2] Erdoğanyılmaz, C., Mengünoğul, B., & Balci, M. (2023, September). Unveiling the Black Box: Investigating the Interplay between AI Technologies, Explainability, and Legal Implications. In *2023 8th International Conference on Computer Science and Engineering (UBMK)* (pp. 569-574). IEEE.

[3] Turan, T., Küçüksille, E., & Alagöz, N. K. (2023). Prediction of Turkish Constitutional Court Decisions with Explainable Artificial Intelligence. *Bilge International Journal of Science and Technology Research*, *7*(2), 128-141.

[4] Hochreiter, S. (1997). Long Short-term Memory. *Neural Computation MIT-Press*.

[5] Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

[6] Öztürk, C. E., ÖzçelıK, Ş. B., & Koç, A. (2023, July). A Transformer-Based Prior Legal Case Retrieval Method. In *2023 31st Signal Processing and Communications Applications Conference (SIU)* (pp. 1-4). IEEE.

[7] Erdoğanyılmaz, C., & Mengünoğul, B. (2022, September). An Original Natural Language Processing Approach to Language Modeling in Turkish Legal Corpus: Improving Model Performance with Domain Classification by Using Recurrent Neural Networks. In *2022 Innovations in Intelligent Systems and Applications Conference (ASYU)* (pp. 1-6). IEEE.

[8] Sevim, N., Şahinuç, F., & Koç, A. (2023). Gender bias in legal corpora and debiasing it. *Natural Language Engineering*, *29*(2), 449-482.

[9] Aletras, N., Tsarapatsanis, D., Preoţiuc-Pietro, D., & Lampos, V. (2016). Predicting judicial decisions of the European Court of Human Rights: A natural language processing perspective. *PeerJ computer science*, *2*, e93.

[10] Kaur, A., & Bozic, B. (2019, July). Convolutional Neural Network-based Automatic Prediction of Judgments of the European Court of Human Rights. In *AICS* (pp. 458-469).

[11] Sulea, O. M., Zampieri, M., Vela, M., & Van Genabith, J. (2017). Predicting the law area and decisions of french supreme court cases. *arXiv preprint arXiv:1708.01681*.

[12] Virtucio, M. B. L., Aborot, J. A., Abonita, J. K. C., Avinante, R. S., Copino, R. J. B., Neverida, M. P., ... & Tan, G. B. A. (2018, July). Predicting decisions of the philippine supreme court using natural language processing and machine learning. In *2018 IEEE 42nd annual computer software and applications conference (COMPSAC)* (Vol. 2, pp. 130-135). IEEE.

[13] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., ... & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, *35*, 24824-24837.

[14] Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., & Narasimhan, K. (2024). Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, *36*.

[15] Douze, M., Guzhva, A., Deng, C., Johnson, J., Szilvasy, G., Mazaré, P. E., ... & Jégou, H. (2024). The faiss library. *arXiv preprint arXiv:2401.08281*.

[16] Devlin, J. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[17] Reimers, N. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv preprint arXiv:1908.10084*.

[18] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M. A., Lacroix, T., ... & Lample, G. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

[19] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... & Chen, W. (2021). Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

[20] Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., ... & Gelly, S. (2019, May). Parameter-efficient transfer learning for NLP. In *International conference on machine learning* (pp. 2790-2799). PMLR.