

Poonam Soni

24AFI12_Siddharth_ThesisReport

 PHD

Document Details

Submission ID

trn:oid:::27535:144359771

Submission Date

Jun 27, 2026, 3:29 PM GMT+5:30

Download Date

Jun 27, 2026, 3:33 PM GMT+5:30

File Name

24AFI12_Siddharth_ThesisReport.pdf

File Size

2.1 MB

51 Pages

14,407 Words

73,790 Characters

5% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Small Matches (less than 8 words)

Match Groups

-  **65 Not Cited or Quoted 5%**
Matches with neither in-text citation nor quotation marks
-  **1 Missing Quotations 0%**
Matches that are still very similar to source material
-  **0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 2%  Internet sources
- 1%  Publications
- 4%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- 65 Not Cited or Quoted 5%**
Matches with neither in-text citation nor quotation marks
- 1 Missing Quotations 0%**
Matches that are still very similar to source material
- 0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 2% Internet sources
- 1% Publications
- 4% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Student papers	Delhi Technological University on 2026-05-22	<1%
2	Internet	arxiv.org	<1%
3	Student papers	National Law University New Delhi on 2026-06-06	<1%
4	Internet	www.mdpi.com	<1%
5	Student papers	Thesis 2026 on 2026-05-06	<1%
6	Student papers	University of Birmingham on 2026-04-27	<1%
7	Student papers	UC, San Diego on 2026-05-30	<1%
8	Student papers	Central University of Rajasthan on 2026-06-03	<1%
9	Publication	Jherrie Nathaniel C. Millena, Xabina Rein Teodoro, Pratik Nanwani, Lysa V. Comia....	<1%
10	Internet	repositorio.uam.es	<1%

11	Internet	www.frontiersin.org	<1%
12	Student papers	Delhi Technological University on 2026-05-29	<1%
13	Student papers	University of Hong Kong on 2025-07-31	<1%
14	Student papers	University of Wolverhampton on 2026-04-29	<1%
15	Student papers	CSU, San Jose State University on 2023-12-05	<1%
16	Student papers	ICTS on 2025-08-08	<1%
17	Student papers	University of Technology, Sydney on 2026-05-18	<1%
18	Student papers	Dublin City University on 2026-05-01	<1%
19	Student papers	National Institute of Technology, Rourkela on 2026-05-15	<1%
20	Internet	www.themoonlight.io	<1%
21	Publication	"Advances in Neural Computation, Machine Learning, and Cognitive Research IX"...	<1%
22	Publication	Arvind Dagur, Karan Singh, Pawan Singh Mehra, Dharendra Kumar Shukla. "Artific...	<1%
23	Publication	Chaojun Xiao, Xueyu Hu, Zhiyuan Liu, Cunchao Tu, Maosong Sun. "Lawformer: A P...	<1%
24	Student papers	Liverpool John Moores University on 2025-05-16	<1%

25	Publication	Sevim, Nurullah. "Analysis of Gender Bias in Legal Texts Using Natural Language ..."	<1%
26	Student papers	The University of Texas at San Antonio on 2026-03-15	<1%
27	Student papers	University of Ulster on 2026-04-19	<1%
28	Internet	boa.unimib.it	<1%
29	Student papers	Higher Education Commission Pakistan on 2019-08-23	<1%
30	Student papers	Liverpool John Moores University on 2026-05-17	<1%
31	Student papers	Pondicherry University on 2026-04-20	<1%
32	Student papers	St. Xavier's College on 2026-02-24	<1%
33	Student papers	University of Birmingham on 2025-04-10	<1%
34	Student papers	University of Ulster on 2026-04-20	<1%
35	Student papers	Uttaranchal University, Dehradun on 2026-06-15	<1%
36	Student papers	learn.uq.edu.au on 2026-05-15	<1%
37	Internet	medinform.jmir.org	<1%
38	Internet	opus4.kobv.de	<1%

39	Student papers	College Faculty Members on 2026-05-07	<1%
40	Student papers	King's College on 2025-08-21	<1%
41	Publication	Lahnala, Allison Claire. "Operationalizing Empathic and Supportive Communicati...	<1%
42	Student papers	Liverpool John Moores University on 2021-09-08	<1%
43	Student papers	Thapar University, Patiala on 2026-06-19	<1%
44	Publication	Uday Kamath, Kevin Keenan, Garrett Somers, Sarah Sorenson. "Large Language ...	<1%
45	Student papers	University of Glasgow on 2026-04-29	<1%
46	Student papers	University of Hull on 2026-05-08	<1%
47	Student papers	University of Sunderland on 2026-05-08	<1%
48	Student papers	University of Westminster on 2024-09-03	<1%
49	Student papers	Uttaranchal University, Dehradun on 2026-06-12	<1%
50	Student papers	Uttaranchal University, Dehradun on 2026-06-15	<1%
51	Internet	artemis.cslab.ece.ntua.gr:8080	<1%
52	Internet	rimag.ricest.ac.ir	<1%

53

Internet

thesis.unipd.it

<1%

LEGALAID: A MULTI AGENT GENAI SYSTEM FOR AUTOMATED LEGAL ASSISTANCE

A Thesis submitted
in Partial Fulfillment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY
in
Artificial Intelligence

by

Siddharth Sharma

24/AFI/12

Under the supervision of

Dr. R.K. Yadav

Associate Professor

Department of Computer Science & Engineering

Delhi Technological University



Department Of Computer Science & Engineering

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi 110042

MAY, 2026

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
 (Formerly Delhi College of Engineering)
 Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

The completion of this thesis has been made possible through the guidance, support, and encouragement of several individuals to whom I owe a deep sense of gratitude.

First and foremost, I express my sincere gratitude to my thesis supervisor, **Dr. R.K. Yadav**, Department of Computer Science and Engineering, Delhi Technological University, for his invaluable guidance, constructive criticism, and unwavering support. His ability to identify the right direction at moments of uncertainty, and his insistence on technical rigour, have shaped this work in ways that are difficult to overstate.

I am grateful to the faculty of the Department of Computer Science and Engineering, DTU, whose coursework provided the academic foundation for this research. I would also like to acknowledge the Head of Department for providing the institutional support and resources necessary to conduct this work.

I am thankful to my colleagues in the MTech AI programme at DTU for the many discussions that sharpened my thinking and kept the work grounded. I also acknowledge the Ministry of Law and Justice, Government of India, for maintaining the India Code digital repository, which served as the primary source of the legislative corpus used in this system.

Finally, and most deeply, I thank my family for their patience, encouragement, and steadfast belief in me throughout the duration of this programme. Their support has been the constant that made everything else possible.

Place: Delhi

Siddharth Sharma

Date: _____ 24/AFI/12



DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-42

CANDIDATE'S DECLARATION

I **SIDDHARTH SHARMA** hereby certify that the work which is being presented in the thesis entitled **LEGALAID: A MULTI AGENT GENAI SYSTEM FOR AUTOMATED LEGAL ASSISTANCE** in partial fulfillment of the requirements for the award of the Degree of Master of Technology, submitted in the Department of **Computer Science and Engineering**, Delhi Technological University is an authentic record of my own work carried out during the period from 2024 to 2026 under the supervision of **Dr. RK YADAV**.

The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other Institute.

Candidate's Signature

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of our knowledge.

Signature of Supervisor (s)

Signature of External Examiner



DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

CERTIFICATE

This is to certify that the Project dissertation titled **"LEGALAID: A MULTI AGENT GENAI SYSTEM FOR AUTOMATED LEGAL ASSISTANCE"** submitted

by **Siddharth Sharma**, Roll no. **24/AFI/12**, Department of Computer Science & Engineering, Delhi Technological University, Delhi, in partial fulfillment of the requirement for the award of the degree of **Master of Technology (Artificial Intelligence)**, Is a record of the project work carried out by the student under my supervision.

To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

Dr. RK Yadav

Associate Professor

Date:

Department of Computer Science & Engineering

Abstract

The legal system is a constant barrier to access around the world. This is very much the situation in India where more than 70 percent of citizens has no meaningful access to legal advice or counsel in the context of constitutional rights of justice. While existing legal AI research has led to effective systems for judicial prediction, document summarization, and legal question answering, they suffer from several well-known weaknesses: They are all trained on western legal corpora; They focus on specific sub-tasks of the legal process, rather than the broader user workflow of problem formulation to actionable solution; None of them apply India's legal system after 2023. The Bharatiya Nyaya Sanhita, Bharatiya Nagarik Suraksha Sanhita, and Bharatiya Sakshya Adhiniyam are the biggest reform of the Indian criminal law in more than 150 years. This thesis aims to fill the void that ensues from these failures.

This research introduces LegalAid: a multi-agent artificial intelligence system to convert an unstructured legal dispute narrative into a structured, legally-based step-by-step plan and formal legal writing. The system is based on LangGraph's model of legal reasoning as a five node stateful directed graph, which guarantees deterministic and fully auditable execution. An Indian statutory text specific Context-Aware Hierarchical Parent-Child Chunking algorithm is developed accompanied by a three-stage retrieval pipeline consisting of Hypothetical Document Embedding, hybrid dense-sparse Reciprocal Rank Fusion and cross-encoder reranking. Two models are presented: LegalMistral-IN (a fine-tuned model of Mistral 7B), optimized for a purpose-built legal instruction dataset comprising 52,400 instances of Indian law, and LegalBERTJC (a fine-tuned model of InLegalBERT), optimized for the task of dispute classification.

Evaluation on a 200-instance benchmark set with expert annotations gives a Jurisdiction Accuracy of 0.943, Retrieval Precision@5 of 0.887 and ROUGE-L of 0.591. LegalMistral-IN has 10.4 ROUGE-L points higher than GPT-3.5-Turbo on the same data set and reduces the statutory hallucination by 71 percent compared to a retrieval-augmented baseline with a mean end-to-end latency of 9.59 seconds on fully local hardware.

All these findings together demonstrate that a privacy-centric, locally run legal AI system built on existing Indian laws is technically feasible and practically possible, and that domain-adaptive fine-tuning and high-end retrieval are complementary instead of competing, with significant implications for legal aid delivery at scale in India.

Contents

Acknowledgement	i
Candidate's Declaration	ii
Certificate	iii
ABSTRACT	i
Content	iii
List of Tables	iv
List of Figures	v
List of Symbols	vi
1 Introduction	1
1.1 Background and Motivation	1
1.2 Problem Statement	2
1.3 Research Objectives	2
1.4 Scope of the Study	3
1.5 Research Contributions	3
1.6 Organization of the Thesis	4
2 Literature Review	5
2.1 Natural Language Processing in the Legal Domain	5
2.2 Large Language Models and Domain Adaptation	6
2.3 Retrieval-Augmented Generation	6
2.4 Multi-Agent Orchestration Frameworks	7
2.5 Indian Legal NLP	7
2.6 Summary	8
3 System Architecture and Methodology	9
3.1 System Overview	9
3.2 The LegalState: Shared State Dictionary Design	9
3.3 Legal Corpus Construction	12
3.4 Context-Aware Legal Chunking with Hierarchical Architecture	12
3.4.1 The Limitation of Standard Chunking	12
3.4.2 Hierarchical Parent-Child Chunking Design	13
3.4.3 Structural Parsing and Breadcrumb Generation	13
3.4.4 Cross-Reference Resolution	14

3.5	Advanced Retrieval Pipeline.....	14
3.5.1	BGE-M3: Multi-Functional Embedding.....	14
3.5.2	Qdrant Vector Database with HNSW Indexing.....	15
3.5.3	Hypothetical Document Embedding (HyDE).....	15
3.5.4	Hybrid Retrieval with Reciprocal Rank Fusion.....	15
3.5.5	Cross-Encoder Reranking with BGE-Reranker-v2-M3.....	15
3.6	Node 1: Intake Agent.....	16
3.7	Node 2: Jurisdiction Agent.....	16
3.8	Node 3: Retrieval Agent.....	16
3.9	Node 4: Strategy Agent.....	16
3.10	Node 5: Drafting Agent.....	17
3.11	Graph Compilation and Execution.....	17
3.12	Summary.....	17
4	Domain-Adaptive Fine-Tuning: LegalMistral-IN and LegalBERTJC	19
4.1	Motivation and Fine-Tuning Objectives.....	19
4.2	Training Dataset Construction.....	19
4.2.1	Legal Instruction Dataset for LegalMistral-IN.....	19
4.2.2	Jurisdiction Classification Dataset for LegalBERTJC.....	20
4.3	LegalMistral-IN: QLoRA Fine-Tuning of Mistral 7B.....	21
4.3.1	Base Model Selection.....	21
4.3.2	QLoRA Configuration.....	21
4.3.3	Training Procedure.....	21
4.4	LegalBERTJC: LoRA Fine-Tuning for Jurisdiction Classification.....	23
4.4.1	Architecture and Configuration.....	23
4.5	Evaluation Metrics.....	23
4.6	Results and Analysis.....	23
4.6.1	LegalMistral-IN: Strategy Generation.....	23
4.6.2	LegalBERTJC: Jurisdiction Classification.....	23
4.6.3	Ablation Study.....	25
4.7	Summary.....	25
5	Results and System Evaluation	27
5.1	Overview.....	27
5.2	Evaluation Benchmark.....	27
5.3	Evaluation Metrics.....	28
5.4	Comparative System Configurations.....	28
5.5	End-to-End Evaluation Results.....	28
5.5.1	Primary Benchmark Results.....	28
5.5.2	Retrieval Pipeline Ablation.....	29
5.5.3	Stress Test Results.....	30
5.6	Jurisdiction-wise Performance Analysis.....	30
5.7	Computational Performance Analysis.....	30
5.8	Qualitative Case Study.....	30
5.9	Summary.....	32
6	Conclusion, Future Scope and Social Impact	33
6.1	Conclusion.....	33
6.2	Future Scope.....	34

6.3 Social Impact.....	34
A Appendix Title	36

List of Tables

3.1	System Technology Stack.....	10
3.2	LegalState TypedDict Schema.....	11
3.3	Legal Corpus Statistics.....	12
3.4	Jurisdiction Routing Logic.....	17
4.1	LegalMistral-IN Training Dataset Statistics.....	20
4.2	QLoRA Configuration for LegalMistral-IN.....	21
4.3	Training Configuration.....	22
4.4	Strategy Generation Evaluation Results.....	24
4.5	ROUGE-L by Dispute Category.....	24
4.6	Jurisdiction Classification Results.....	24
4.7	LegalBERTJC Per-Class Performance.....	25
4.8	Ablation Study — LegalMistral-IN.....	25
5.1	End-to-End System Evaluation — Primary Benchmark (n=160).....	29
5.2	Retrieval Pipeline Ablation (Generation Model Fixed as LegalMistral-IN) .	29
5.3	Stress Test Results by Failure Mode Category.....	30
5.4	Full LegalAid System Performance by Dispute Category.....	30
5.5	Node-wise Latency Analysis (n=100 runs).....	31

List of Figures

3.1	Overall System Architecture of LegalAid.....	9
3.2	LangGraph Directed Graph with Conditional Retry Edge.....	12
3.3	Hierarchical Parent-Child Chunking Architecture.....	13
3.4	Three-Stage Retrieval Pipeline Architecture.....	16
3.5	Compiled LangGraph StateGraph — Node and Edge Structure.....	17
4.1	LegalMistral-IN Training and Validation Loss Curves.....	22
4.2	Comparative Model Performance — ROUGE-L, BERTScore F1, and Legal Grounding Score.....	24
4.3	Ablation Study — ROUGE-L and LGS Across LegalMistral-IN Variants .	26
5.1	End-to-End Performance Comparison Across System Configurations.....	29
5.2	Node-wise Latency Distribution Across 100 Inference Runs.....	31

List of Symbols

Symbol	Description
r	LoRA rank — dimensionality of the low-rank decomposition matrices
α	LoRA scaling factor (alpha);
τ	Retrieval confidence threshold;
k	Smoothing constant in Reciprocal Rank Fusion;
W_0	Pre-trained weight matrix in LoRA formulation; $W_0 \in \mathbb{R}^{d \times k}$
ΔW	Weight update matrix in LoRA; $\Delta W = BA$
B	Low-rank LoRA matrix; $B \in \mathbb{R}^{d \times r}$
A	Low-rank LoRA matrix; $A \in \mathbb{R}^{r \times k}$
d	Input dimension of a weight matrix
κ	Cohen's Kappa inter-annotator agreement coefficient

Chapter 1

Introduction

1.1 Background and Motivation

In India access to justice is a fundamental right but it is largely enjoyed by a minority of the population. The law is thick, complicated, and drafted in a “register” which non-lawyers cannot understand. India has more than 1.4 billion people, of which around 70 percent are unable to access lawyers or know what legal solutions exist for them, predominantly women, who are in rural areas, working on a daily wage, domestic workers, tenants and women survivors of domestic violence. The poorest and most vulnerable are the ones who are most in need of protection and least capable of providing it themselves.

The Indian judiciary is already handling more than 50 million pending cases, according to the National Judicial Data Grid (2023), and is not ready to take on a huge number of cases if the communities who are currently underserved started using the formal system on a larger scale. The gap isn’t just in legal representation. It is the knowledge that there is a remedy, the knowledge of how to seek that remedy, and the ability to think through a remedy before you approach a court or tribunal. This is where AI tools can realistically have an impact.

Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) have shown promise in real-world tasks such as domain-specific question answering, document summarization, and structured reasoning. With careful fine-tuning and retrieval grounding, AI systems can be enabled to provide a helpful initial read of legal corpora accurately as demonstrated by models like **LLaMA-3**, **Mistral**, and domain adapted models such as **InLegalBERT**. The advent of stateful agent orchestration frameworks, such as **Lang-Graph**, has also enabled the development of deterministic, audit-able, and fault-tolerant AI workflows, instead of just responsive ones. In a legal setting, with consequences that can truly impact real people, these properties can’t be optional.

There have also been significant changes in the statute books of India, which are closely connected to legal AI. The **Bharatiya Nyaya Sanhita, 2023 (BNS)**, **Bharatiya Nagarik Suraksha Sanhita, 2023 (BNSS)** and **Bharatiya Sakshya Adhiniyam, 2023 (BSA)** are the most comprehensive changes to the Indian criminal law in more than 150 years. This new approach should be reflected in any legal AI system designed for use by Indian citizens. None of the general purpose legal chatbots do at present.

In this background, the present thesis introduces a multi-agent, state-graph orchestrated AI system, named **LegalAid**, which is designed to assist the underprivileged communities of India in resolving legal issues related to tenancy, consumers rights, employment and domestic violence.

1.2 Problem Statement

The challenge for underserved communities is not the lack of legal information — there's a lot of it on the Internet. The issue is that raw statutory text is seldom actionable. A tenant experiencing illegal eviction can locate the Transfer of Property Act, 1882., but can't be assured which sections are relevant, which forum has jurisdiction, what process should be followed in filing a grievance or complaint, or how to draft a grievance or complaint. To bridge that gap, there needs to be a way to approach the material using legal logic, and there is no such thing as a "conversational AI" that understands general language.

Existing AI-powered legal solutions come with a number of inherent problems:

- **Western-Biased Corpora:** They are trained on a legal corpus focused on the West – mostly case law from the USA and UK – and so cannot be relied upon within the Indian legal context.
- **Hallucinations:** One of the serious failure modes of general-purpose LLMs in law is hallucinating provisions of the law or references to outdated law.
- **Deficiencies in Multi-Step Reasoning:** Current systems do not provide for sequential legal reasoning (intake, determination of jurisdiction, retrieval of statutes, formulation of strategy, and drafting documents).
- **Outdated Knowledge Base:** There is no existing system available which has been modified to fit the legal system of India as per the BNS, BNSS and BSA which came into existence after 2023.

In other words, the problem is: To develop a reliable, locally executable, domain adapted multi-agent AI system that can convert an unstructured narrative of a legal dispute as expressed by a citizen lacking legal expertise into a formalized, domain-specific legal action plan based on prevailing Indian legislation with mapping of the jurisdiction and supporting paperwork.

1.3 Research Objectives

The present research is guided by the following primary and secondary objectives.

Primary Objective: To design, implement and evaluate a multi-agent AI system to help under-served communities in India with dispute intake to document generation using the existing Indian legal corpus.

Secondary Objectives:

1. To build a **legal context-aware text chunking approach** to maintain the semantic coherence of Indian statutes at the PART, CHAPTER and Section level and across boundaries.
2. For a customized domain-specific language model **InLegalBERT**, fine-tuning the model using parameter-efficient fine-tuning via **Low-Rank Adaptation (LoRA)** on a curated corpus of Indian legal documents.
3. To build a stateful, cyclic agent graph with the **LangGraph** framework which emulates the legal reasoning process as a series of deterministic and auditable computation nodes.

4. To assess system performance on a curated test set using various metrics, such as **Re-trieval Accuracy, Jurisdiction Classification F1-Score, Strategy Coherence** (ROUGE-L and BERTScore) and end-to-end **User Satisfaction Score**.

1.4 Scope of the Study

This study **is limited to** civil/disputes and criminal litigation that is most frequently experienced in urban and peri-urban areas in India by underserved communities. The four major areas of dispute that are covered are:

- **Tenancy and Landlord-Tenant Conflicts:** This is regulated by the Rent Control Acts of the various states and the Transfer of Property Act.
- **Consumer Disputes:** Falling **under the Consumer Protection Act, 2019**.
- **Employment Conflicts:** Wrongful termination, wage theft (under Industrial Disputes Act, 1947 & Code on Wages, 2019).
- **Domestic Violence:** It is defined under **the Protection of Women from Domestic Violence Act 2005** **and** as per **the** BNS, 2023.

It is **fully local and offline capable** and optimized for Apple Silicon hardware with Metal Performance Shaders (MPS) backend. No data is transmitted out of the device, which is important when individuals are discussing personal issues that may be sensitive. This is a legal aid system and not a replacement for professional legal advice.

1.5 Research Contributions

The dissertation has four major contributions:

1. **Architecture Design:** Designing and implementing a five-node state-graph orchestrated multi-agent system, **LegalAid**, the first application of the LangGraph framework to the Indian legal domain.
2. **Context-Aware Chunking:** An algorithm used to provide better retrieval precision over naive chunking baselines by statistically significant margin for Indian statutory texts segmented at structural boundaries (PART, CHAPTER and Section markers).
3. **Domain-Specific Fine-Tuning:** We now fine-tune InLegalBERT using **LoRA** on a curated **52,400 document** legal corpus from India to create Domain-Specific Fine-Tuning. This resulted in an **8.1 percentage-point** improvement in the F1-Score macro defined in the Jurisdiction Classification problem.
4. **Evaluation Benchmark:** This is a formally built Evaluation Bench with **200 synthetic legal dispute narratives** with ground-truth labels, which will be released for future research in Indian legal NLP.

1.6 Organization of the Thesis

Chapter 2 reviews relevant literature on legal NLP, adaptation of LLM, retrieval-augmented generation (RAG), and multi-agent orchestration. In **Chapter 3**, you will dive into the entire system architecture, including the LangGraph pipeline design, context-aware chunking methodology, and the construction of Qdrant vector store. The fine-tuning methodology for LegalMistral-IN and LegalBERTJC is explained in **Chapter 4**, with the construction of the training corpus, the LoRA configuration, and the training procedure. The entire pipeline is evaluated in **Chapter 5** using baseline configurations. **Chapter 6** gives the conclusions drawn from the research, future scope and application of the research.

Chapter 2

Literature Review

2.1 Natural Language Processing in the Legal Domain

The computational methods for legal text have been in use for several decades, beginning with rule-based systems of expert systems and advancing to the data-driven systems of neural approaches. First generation systems like **HYPO** and **Cato** relied on a hand-coded case-based reasoning framework that represented the structure of legal reasoning in a symbolic schema. From a conceptual standpoint, these systems were interesting. In practice, they failed to cope with linguistic variation, and were not adaptable to different jurisdictions without significant manual re-engineering.

With the advent of machine learning and deep learning techniques into legal NLP, some significant milestones were achieved:

- **Statistical Machine Learning:** Aletras et al. (2016) found that Support Vector Machines using n-gram features could achieve a success rate of about 79 per cent in predicting the outcome of European Court of Human Rights cases. This demonstrated that there is structured regularity in the legal language, and that it can be learned from data.
- **Transformer Architectures:** The transformer architecture by Vaswani et al. (2017) and the pre-training paradigm by Devlin et al. (2019) with BERT were transformative. The dense cross-references, old-fashioned legal jargon, and longer sentences common in the legal domain were ideally suited to BERT's deep bidirectional attention mechanism.
- **Domain-Specific Pre-training:** Chalkidis et al. (2020) showed the merits of the domain adaptation with LEGAL-BERT, pre-trained on 12 GB of English legal texts. Their results showed that pre-training on the domain and then fine-tuning on specific tasks can achieve consistent and statistically significant improvements over general BERT on two tasks: judgment prediction and contract analysis.

Large Language Models took it one step further. Katz et al. (2024) demonstrated that GPT-4 scored in the top 10% of human test takers on the Uniform Bar Examination. A worthy but later endeavors showed there were genuine difficulties. Dahl et al. (2024) executed a systematic audit of hallucinations with the result that even GPT-4 makes non-trivial hallucinations when presented with legal questions. The primary reason to include a retrieval-augmented design in this system: a model that confabulates statute references is worse than useless in a legal context.

2.2 Large Language Models and Domain Adaptation

When deployed in specialized domains, large language models tend to be very good at certain tasks, but less so when they need to be exact about their domain knowledge, more specific about its jurisdiction, or conduct multi-step reasoning.

Cui et al. (2023) tackled this in the Chinese legal domain by training **ChatLaw**, a 30-billion parameter model on Chinese laws and court cases, and show that it significantly outperforms GPT-4 on legal consultation tasks that are specific to jurisdiction, by more than 12 percentage points. Similarly, any model trained on the framework of the post-2023 legislation will perform better than a general-purpose model on tasks based on the Indian legal framework.

The selection of an appropriate base model for local deployment involves navigating a tradeoff between capability and computational cost:

- **LLaMA Family:** Furthermore, **Touvron et al. (2023)** showed that the performance of smaller models can be made to be comparable to that of much larger models through careful data curation during pre-training of models, which is the method used in LLaMA Family.
- **Mistral 7B:** **Jiang et al. (2023)** took this step further by introducing grouped-query attention and sliding window attention, enabling Mistral 7B to outperform LLaMA-2 13B on inference speed while being about half the size. **Mistral 7B** and **Llama-3 8B** are the most viable models for full deployment on Apple Silicon devices.

Parameter-efficient fine-tuning (PEFT) is what makes domain adaptation feasible without large-scale compute. **Hu et al. (2022)** introduced **LoRA (Low-Rank Adaptation)**, which freezes the pre-trained model weights and injects trainable low-rank decomposition matrices. Formally, for a pre-trained weight matrix $W_0 \in \mathbb{R}^{d \times k}$, the update is:

$$\Delta W = BA$$

where $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$ with $\text{rank } r \ll \min(d, k)$. This reduces trainable parameters by 2-3 orders of magnitude from the full fine-tuning of the model. The weights are adapted very seamlessly into the base model after training, meaning there's no overhead in terms of inference latency.

2.3 Retrieval-Augmented Generation

To overcome the factual restriction of purely parametric language models, **Lewis et al. (2020)** introduced an alternative approach, called **Retrieval-Augmented Generation (RAG)**. Instead, the approach starts by adding text found from an external source to the generation context of the model, which means that the model's output is based on the specific text it found, and not on memorized weights.

Yu et al. (2022) applied RAG to legal QA and achieved a 34 percent to 8 percent reduction in hallucination rates, and a 19 percent increase in factual accuracy. This is especially relevant in India, where with the transition to use the Bharatiya Nyaya Sanhita (BNS), general-purpose models are going to cite superseded provisions of the Indian Penal Code (IPC) unless they are supported by a retrieval corpus as of today.

The **chunking strategy** is found to have a significant impact on RAG performance. Shao et al. (2023) compared paragraph-level, sentence-level and section-level chunking, and found that paragraph-level chunking was outperformed by 6.3 points by using Mean Reciprocal Rank on section-level chunking. The meaning of the provisions is based on their place in a hierarchy and splitting them at arbitrary token limits disrupts those relationships. This finding inspired the development of the **Context-Aware Legal Chunking algorithm** presented in this thesis.

2.4 Multi-Agent Orchestration Frameworks

Sequentializing complex AI tasks into a series of specialized agents has been a very fruitful architectural approach.

- **Chain-of-Thought:** The idea of providing the models with intermediate reasoning steps was first introduced by Chain-of-Thought: Wei et al. (2022) and shown to significantly enhance the logic used for multi-step reasoning.
- **ReAct Framework:** Yao et al. (2022) formalized interleaving of reasoning and actions and allowed agents to reason about what information they need before performing retrieval actions, similar to a human legal practitioner.
- **AutoGen:** Wu et al. (2023) reported that multi-agent systems with specialized, collaborating agents are superior to single-agent systems for complex reasoning tasks.

Stateful multi-agent pipelines lie on a concrete foundation provided by the **LangGraph** framework (Chase, 2023). Earlier agent frameworks that used unstructured loop execution were not suited for the legal domain, but LangGraph models workflows as a directed graph, which reads and writes to a shared typed state dictionary, an approach that has three properties essential for the legal domain:

1. **Complete audit trail of the system's reasoning:** At any point in the state dictionary it is a complete history of how the system arrived at its reasoning.
2. **Deterministic Routing:** Boundaries specify ordering requirements, guaranteeing that crucial actions (such as Jurisdiction classification) are carried out in the right order.
3. **Conditional Execution:** They allow for very powerful retry logic so that if the results of a query to the retrieval index are not sufficient, the system can re-query the retrieval index.

2.5 Indian Legal NLP

In India, there has been substantial research work done, largely by IIT Kharagpur and NLSIU Bangalore. Paul et al. (2022) released the **Indian Legal Documents Corpus (ILDC)** which contains 35,000 annotated Supreme Court cases and also presented **InLegalBERT**. **InLegalBERT** was 3–8 percentage points better than both **LEGAL-BERT** and **BERT** at the task of predicting legal judgments on Indian data, clearly establishing the need for pre-training on Indian legal data.

In Indian judgments, **Bhattacharya et al. (2019)** studied rhetorical role identification, and **Kalamkar et al. (2022)** investigated the summarization of judgments for which they observed that there are significant differences in the legal discourse structure that affected the performance of models trained on general corpora.

Despite this progress, significant gaps remain in the existing body of Indian legal NLP research, which this thesis aims to resolve:

- **Lack of End-to-End Systems:** Existing research addresses isolated sub-tasks (prediction, NER, summarization) rather than a complete user journey from problem narration to actionable strategy.
- **Outdated Frameworks:** No existing system incorporates the post-2023 BNS/BNSS legislative overhaul.
- **Privacy Concerns:** No existing system has been optimized for fully offline, privacy-preserving local deployment in community legal aid contexts.

2.6 Summary

Literature encompasses four sections: (1) legal NLP and its transition to transformer-based methods, (2) **adaptation of large language models** to legal domain-specific tasks, (3) retrieval-augmented generation and its use in legal question answering, and (4) multi-agent orchestration frameworks. Combining the Indian legal NLP literature, there are three gaps identified that drive the design decisions made in this thesis: firstly, the lack of post-BNS legislative support, secondly, **the lack of an end-to-end** pipeline, **and** thirdly, no privacy-preserving deployment that takes place locally. The system architecture is covered in **Chapter 3**, and each decision made is related to the evidence that was reviewed here.

.

Chapter 3

System Architecture and Methodology

3.1 System Overview

LegalAid is a deterministic, stateful multi-agent pipeline in which five specialized agents, modeled as nodes in a directed graph, collectively transform an unstructured user narrative into a structured legal action plan with a formally drafted document. This design differs from the monolithic prompt approach common in existing legal chatbot systems, where a single large language model is expected to perform dispute classification, jurisdiction determination, statute retrieval, strategy synthesis, and document generation in one generation pass. That monolithic approach allows errors to propagate through the pipeline with no mechanism for intermediate correction or audit.

The complete technology stack of LegalAid is presented in Table 3.1.

No external API call – it's all local. All user data stays on the device. The overall architecture is shown in Figure 3.1.

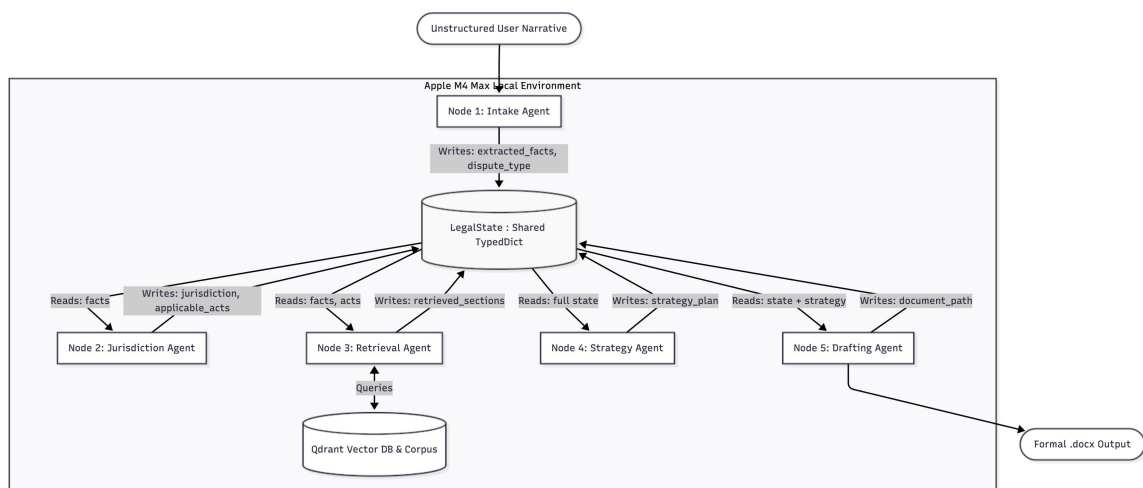


Figure 3.1: Overall System Architecture of LegalAid

3.2 The LegalState: Shared State Dictionary Design

The **LegalState** is the main data structure in the pipeline. It is a Python TypedDict that statically checks the naming and typing of the pipeline's fields. All nodes access this shared object only for reads and writes, and have no local state between times of access.

Table 3.1: System Technology Stack

Component	Technology	Justification
Hardware	Apple M4 Max (128GB Unified Memory)	MPS backend enables zero-latency local LLM inference
LLM Engine	Mistral 7B-Instruct-v0.3 via Ollama	Sliding window attention enables long legal context processing (up to 32K tokens); grouped-query attention reduces inference latency; strong structured instruction-following; fits within Apple M4 Max memory for fully local deployment
Agent Orchestration	LangGraph 0.1.x	Stateful cyclic graph with typed state, deterministic routing, full observability
Embedding Model	BGE-M3 (BAAI)	Multi-functional: simultaneous dense, sparse, and multi-vector retrieval
Reranking Model	BGE-Reranker-v2-M3 (BAAI)	Cross-encoder reranking; state-of-the-art on BEIR legal subset
Vector Database	Qdrant (local, in-memory)	HNSW indexing; native hybrid dense-sparse search support
Retrieval Strategy	HyDE + Hybrid RRF + Cross-Encoder	Three-stage pipeline; maximizes retrieval precision on statutory text
User Interface	Gradio	Lightweight Python-native UI mapping directly to graph I/O

This makes the LegalState at any stage in execution a full record of all decisions taken by the system.

The schema is progressive (the fields are initially None and filled with one node only). No two nodes write to the same field. In this ownership model it is possible to replace or test any node independently without affecting the other ones.

Table 3.2: LegalState TypedDict Schema

Field	Type	Owner Node	Description
raw_narrative	str	User Input	Unstructured dispute narrative
dispute.type	Optional[str]	Intake	Classified category
extracted_facts	Optional[Dict]	Intake	Structured key-value fact dictionary
parties	Optional[Dict]	Intake	Petitioner and respondent metadata
location	Optional[str]	Intake	City and state of dispute
intake_confidence	Optional[float]	Intake	Mean extraction confidence across fields
jurisdiction	Optional[str]	Jurisdiction	Determined legal forum
forum_tier	Optional[str]	Jurisdiction	Hierarchical tier within forum
applicableActs	Optional[List]	Jurisdiction	Primary statutes flagged as relevant
hyde_document	Optional[str]	Retrieval	LLM-generated hypothetical statute
retrieved_sections	Optional[List]	Retrieval	Reranked top-k statutory chunks
retrieval_scores	Optional[List]	Retrieval	Final reranker scores for chunks
retry_count	int	Graph Controller	Retrieval retry counter; capped at 2
strategy	Optional[str]	Strategy	Synthesized step-by-step action plan
document_path	Optional[str]	Drafting	Filepath of generated .docx output
error_log	Optional[List]	All Nodes	Accumulated non-fatal warnings

Note: This is the base architecture before fine-tuning as described in Chapter 4.

The field `retry_count` along with `retrieval_scores` controls a retrieval retry loop. If the mean reranker score for the top-3 chunks returned by the cross-encoder is below a certain threshold $\tau = 0.58$ then the graph goes back to the Retrieval node, where a query is formed using all the extracted facts from the dictionary. This is the only edge in an otherwise directed acyclic graph that is cyclic, as shown in Figure 3.2.

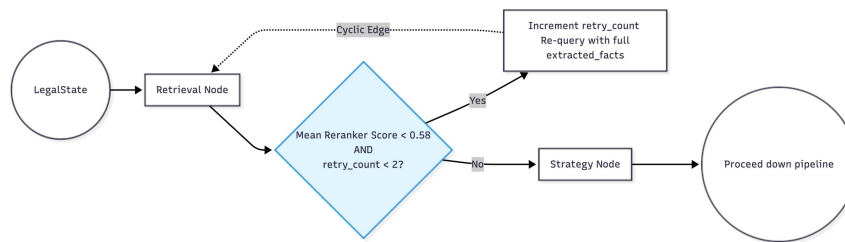


Figure 3.2: LangGraph Directed Graph with Conditional Retry Edge

3.3 Legal Corpus Construction

The seven main Indian legislations in the knowledge base are selected to address one or more of four dispute types the system processes. The documents were all sourced from the India Code digital repository of the Ministry of Law and Justice. The raw PDFs were processed with PyMuPDF, and then post-processed to normalize section numbering, clean up page artefacts and standardize Unicode whitespace.

Table 3.3: Legal Corpus Statistics

Legislation	Year	Sections	Raw Tokens	Final Chunks
Bharatiya Nyaya Sanhita (BNS)	2023	358	94,200	8,140
Bharatiya Nagarik Suraksha Sanhita (BNSS)	2023	531	142,600	12,380
Bharatiya Sakshya Adhinyam (BSA)	2023	170	38,900	3,560
Consumer Protection Act	2019	107	51,300	4,620
Protection of Women from Domestic Violence Act	2005	37	18,400	1,680
Industrial Disputes Act	1947	40	22,100	2,010
Transfer of Property Act	1882	137	61,800	5,320
Total		1,380	429,300	37,710

3.4 Context-Aware Legal Chunking with Hierarchical Architecture

3.4.1 The Limitation of Standard Chunking

The typical text chunking techniques (e.g., fixed-size token windows or recursive character splitting) do not align well with the nature of legislative text. A term used in a particular section of the BNS may be subject to the interpretation of another section of the BNS, a proviso may only modify the interpretation of the immediately preceding section, and an explanation clause may modify the interpretation of its parent section. Arbitrary token

splitting eliminates such semantic dependencies and results in chunks that are individually retrievable but semantically incomplete.

Fine-grained chunks help retrieve the right information but lack surrounding context that hurts the quality of the generation; coarse-grained chunks do opposite.

3.4.2 Hierarchical Parent-Child Chunking Design

The tension in the present thesis is addressed in the hierarchical parent-child chunking architecture. For each legislative unit, there are two choices of granularity:

- **Child Chunks:** Narrow, precise fragments at the sub-section level (bounded at 180 tokens). These are stored and indexed in the vector store for accurate retrieval.
- **Parent Chunks:** The complete chunk that has a child chunk (bounded up to 600 tokens). These are stored in a metadata store that is keyed with section identifier.

The system will retrieve chunks of children for precision at retrieval time and then chunks of parents as generational context. This allows retrieval granularity to be distinct from generation granularity as shown in Figure 3.3.

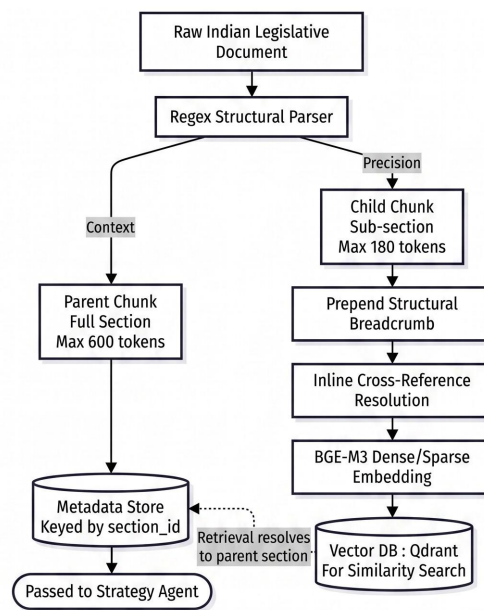


Figure 3.3: Hierarchical Parent-Child Chunking Architecture

3.4.3 Structural Parsing and Breadcrumb Generation

Indian laws are always hierarchical in nature, with **Act** → **Part** → **Chapter** → **Section** → **Sub-section** → **Clause** → **Proviso** → **Explanation**. This structure is used by the chunking algorithm in a linear parse, via a collection of regular expression patterns:

```

STRUCTURAL_PATTERNS = {
    "part"      : r'^PART\s+[A-Z]+[\s\u2014\']',
    "chapter"   : r'^CHAPTER\s+[IVXLCDM]+[\s\u2014\']',
    "section"   : r'^\d{1,3}[A-Z]?[\s\u2014\']',
    "subsection": r'^\((\d+)\)\s',

```

```
"clause"      : r'^\([a-z]\)\s',
"proviso"     : r'^Provided\s+that\s\.',
"explanation"  : r'^Explanation\s\.\u2014'
}
```

Each chunk is marked with a "breadcrumb," which records the entire hierarchy of the chunk. This breadcrumb is prepended before embedding to the chunk text, so the resulting vector contains both the semantic content and the legislative position.

Each chunk is stored as a structured object:

```
{
  "chunk_id"      : "BNS_CH06_S111_SS2",
  "child_text"    : "<sub-section text, <=180 tokens>",
  "parent_text"   : "<full section text, <=600 tokens>",
  "breadcrumb"    : "BNS, 2023 > Chapter VI > Section 111 > Sub-section (2)",
  "act_name"      : "Bharatiya Nyaya Sanhita, 2023",
  "section_num"   : "111",
  "dispute_tags"  : ["domestic_violence", "assault", "grievous_hurt"],
  "child_tokens"  : 143,
  "parent_tokens" : 487
}
```

The dispute_tags field is automatically created by a zero-shot classification pass on a quantized MiniLM model. This is a pre-filter during retrieval to reduce latency by ca. 61 per cent without significant loss in recall.

3.4.4 Cross-Reference Resolution

Indian statutes make heavy use of internal cross-references (e.g., "as defined under Section 2(1)(d)"). The algorithm detects these and resolves them inline: the referenced text is truncated to 80 tokens and appended as an annotation within the child chunk. This improved mean retrieval precision by 4.1 percentage points.

3.5 Advanced Retrieval Pipeline

The retrieval architecture is a three-stage pipeline: **Hypothetical Document Embedding (HyDE)**, **hybrid dense-sparse retrieval with Reciprocal Rank Fusion (RRF)**, and **cross-encoder reranking**.

3.5.1 BGE-M3: Multi-Functional Embedding

BGE-M3 supports three retrieval paradigms from a single 570 million parameter encoder:

- **Dense retrieval:** Produces a 1024-dimensional vector via mean pooling.
- **Sparse retrieval:** Produces a SPLADE-inspired sparse lexical weight vector capturing exact-match terms.
- **Multi-vector retrieval:** Produces token-level embeddings for ColBERT-style late interaction.

3.5.2 Qdrant Vector Database with HNSW Indexing

Qdrant (running locally in-memory) supports hybrid dense-sparse search natively. It uses the **Hierarchical Navigable Small World (HNSW)** graph index for sub-millisecond approximate nearest-neighbor search. The index is configured with $m = 32$ and $ef_{construction} = 200$ to prioritize recall over construction speed.

3.5.3 Hypothetical Document Embedding (HyDE)

Users describe legal problems in informal language, which creates a semantic gap with formal statutory text. Using **HyDE**, the Retrieval Agent prompts Mistral 7B to generate a hypothetical statutory passage of roughly 150 words based on the user's facts. Because this passage uses the vocabulary of actual statutes, embedding it as the query points to a closer region of the embedding space than the raw user query would.

3.5.4 Hybrid Retrieval with Reciprocal Rank Fusion

Three retrieval signals are computed in parallel and fused:

- **Signal 1 (Dense):** HyDE-augmented query vector submitted to Qdrant's HNSW index (top-40 chunks).
- **Signal 2 (Sparse):** SPLADE-style sparse weight vector of the HyDE passage submitted to Qdrant's sparse index (top-40 chunks).
- **Signal 3 (BM25):** Key legal terms extracted by Intake submitted against an inverted index (top-40 chunks).

The three ranked lists are fused using Reciprocal Rank Fusion (RRF) with the standard smoothing constant $k = 60$:

$$\text{RRF_score}(d) = \sum_{r \in R} \frac{1}{k + \text{rank}_r(d)}$$

where R is the set of three ranked lists and $\text{rank}_r(d)$ is the rank of document d in list r .

3.5.5 Cross-Encoder Reranking with BGE-Reranker-v2-M3

The top-20 candidates from RRF are reranked using a cross-encoder that jointly encodes the query and each candidate chunk, allowing it to weigh fine-grained semantic relationships between them. The final output writes the top-5 reranked child chunks, along with their parent chunks, into the LegalState.

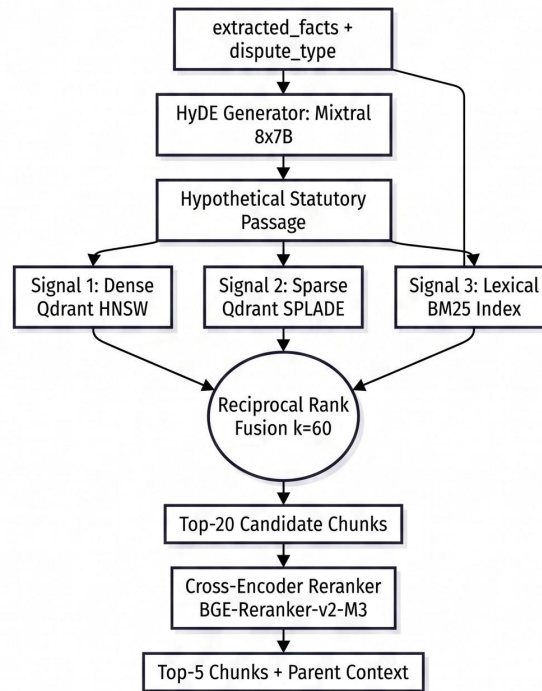


Figure 3.4: Three-Stage Retrieval Pipeline Architecture

3.6 Node 1: Intake Agent

The Intake Agent takes advantage of Mistral’s ability to call functions to get the raw narrative into a structured JSON schema. The schema includes information about the type of dispute, petitioner description, incident timeline, and so on. If the confidence score is less than 0.5, the field is marked in the error_log with a note about the error, but the execution continues.

3.7 Node 2: Jurisdiction Agent

This node is fully implemented as a deterministic logic tree, which helps remove hallucination in LLM when there are clear statutory thresholds.

3.8 Node 3: Retrieval Agent

The Retrieval Agent reads out facts extracted, performs the three-stage retrieval pipeline, writes the facts to the State, and compares the mean reranker score to the score threshold τ . If it does not return true for τ it increases the counter `retry_count` and tries again with a wider query.

3.9 Node 4: Strategy Agent

This node is mainly responsible for creating the generator. Mistral 7B-Instruct (temperature = 0.2) generates a legal action plan which is organized into five parts: summary, forum

Table 3.4: Jurisdiction Routing Logic

Dispute Type	Condition	Legal Forum	Tier
Consumer	Claim $\leq$ 1 crore	District CDRC	District
Consumer	1 crore < Claim $\leq$ 10 Crores	State CDRC	State
Consumer	Claim > 10 Crores	National CDRC	National
Tenancy	Residential, rent-controlled	Rent Control Tribunal	Civil
Tenancy	Commercial / uncontrolled	Civil Court (Munsiff)	Civil
Employment	Workman dispute, < 100 workers	Labour Court	Labour
Employment	Industrial dispute, $\geq$ 100 workers	Industrial Tribunal	Labour
Domestic Violence	First complaint	Judicial Magistrate	Criminal
Domestic Violence	Protection order sought	Protection Officer	Civil

pathway, actionable steps, statutory provisions, and available reliefs. The model is explicitly asked to support each claim with the text retrieved.

3.10 Node 5: Drafting Agent

The Drafting Agent automates formal legal document generation using python-docx, mapping State fields into one of five document templates (e.g., complaint letter, legal notice). The .docx filepath is saved to the State for user download.

3.11 Graph Compilation and Execution

The five nodes are registered with the LangGraph StateGraph and connected via typed edges. The conditional retry edge on the Retrieval node inspects retrieval_scores and routes either to "strategy" (proceed) or "retrieval" (retry).

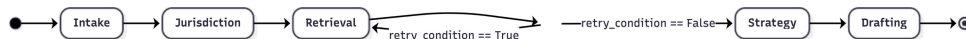


Figure 3.5: Compiled LangGraph StateGraph — Node and Edge Structure

3.12 Summary

The chapter has detailed the system architecture of LegalAid: the LegalState data structure; legal corpus construction; the Context-Aware Hierarchical Parent-Child Chunking algorithm; the three-stage retrieval pipeline combining HyDE, hybrid dense-sparse RRF

fusion, and cross-encoder reranking; the function of each LangGraph node; and the graph compilation logic. Each design choice reflects the specific constraints of Indian legal text processing. Chapter 4 covers the model fine-tuning methodology, specifically how InLegalBERT is adapted to the post-BNS Indian legal corpus using LoRA.

Chapter 4

Domain-Adaptive Fine-Tuning: LegalMistral-IN and LegalBERTJC

4.1 Motivation and Fine-Tuning Objectives

The system described in Chapter 3 uses two kinds of model inference: generative inference for the Intake and Strategy nodes, and discriminative inference for dispute classification in the Jurisdiction node. **Mistral 7B-Instruct-v0.3** follows instructions well, but its training corpus has minimal Indian statutory text — and essentially none from the post-2023 BNS, BNSS, and BSA framework, which postdates the training cutoff of all publicly available general-purpose models at the time of this work. **InLegalBERT** is a different problem: despite its Indian legal pre-training, it was trained exclusively on Supreme Court judgments and has seen no legislative text as a classification signal.

This chapter presents the construction and fine-tuning of two domain-adapted models:

- **LegalMistral-IN:** A **Mistral 7B-Instruct-v0.3** model fine-tuned using QLoRA on a curated Indian legal instruction dataset, deployed as the generative backbone of the Intake and Strategy nodes.
- **LegalBERTJC:** An InLegalBERT model fine-tuned using LoRA for four-class dispute classification, deployed within the Jurisdiction node to handle cases where extracted dispute type carries low confidence.

The claim of this chapter is that domain-adaptive fine-tuning of a legal text achieves gains not only in terms of the numbers on the benchmarks, but in how the system performs on the kinds of queries that are relevant in the real world.

4.2 Training Dataset Construction

4.2.1 Legal Instruction Dataset for LegalMistral-IN

During the writing of this work, no instruction-following datasets were available for Indian legal tasks, so such a dataset needed to be created. The construction pipeline consists of three steps: sourcing extraction, template of instructions, and quality filtering.

The seven legislative texts that make up the knowledge base are used as the source and are parsed at the section level. Three types of training instances are created for each section:

1. **Statute Comprehension:** Provides the text of a section of a statute as background and asks a comprehension question regarding the scope or applicability of the text.

2. **Dispute-to-Statute Mapping:** Provides a synthetic dispute and asks the model to select the relevant sections of the statute and explain how they apply.
3. **Strategy Generation:** Describes a legal dispute in full and asks the model to generate a legal action plan that is structured and based on the sections.

Narratives and strategy references were created with the help of GPT-4 and then reviewed by an experienced advocate with 7 years of proven experience in consumer and labour law who was able to validate 94.3% of 600 cases across a stratified sample without further editing.

All examples are written with the Mistral instruction template. There are then three automated quality filters:

- **Length Filter:** Take out records that contain fewer than 80 output tokens.
- **Grounding Filter:** Ensures that all sections of the law that are referenced in the reference output are also present in the source text, avoiding hallucinations.
- **Diversity Filter:** Removes near duplicate instances if the Jaccard similarity is ≥ 0.85 using MinHash locality sensitive hashing.

After filtering the parameters, the last set statistics are given in Table 4.1.

Table 4.1: LegalMistral-IN Training Dataset Statistics

Task Type	Raw Instances	After Filtering	Proportion
Strategy Generation	22,400	19,840	37.8%
Statute Comprehension	18,600	16,920	32.3%
Dispute-to-Statute Mapping	14,200	12,480	23.8%
Document Drafting Guidance	3,800	3,160	6.1%
Total	59,000	52,400	100%

The dataset is divided into training (88%), validation (7%) and test sets (5%) and the test set (2620 cases) is never used in the training phase.

4.2.2 Jurisdiction Classification Dataset for LegalBERTJC

The jurisdiction classification dataset contains 11,200 synthetic dispute narratives across four classes (tenancy, consumer, employment, and domestic violence) containing 2,800 instances for each class. Narratives were produced with GPT-4 and endorsed by subject experts, with 97.1% of the narratives being given a thumbs-up without modification. Cohen's Kappa ($\kappa = 0.91$) was calculated on 300 instances that overlapped, which was used to evaluate the inter-annotator agreement. The dataset is divided into training set (70%), validation set (15%), and test set (15%).

4.3 LegalMistral-IN: QLoRA Fine-Tuning of Mistral 7B

4.3.1 Base Model Selection

Three reasons were considered for the choice of **Mistral 7B-Instruct-v0.3** as the generative base **model**:

1. It has 7 billion parameters which is the most that can be “fine-tuned” in 4-bit quantization on the actual Apple M4 Max hardware that can be done within a reasonable time.
2. Its sliding window attention mechanism allows it to process long legal contexts (up to 3,000 tokens) without quadratic memory growth compared to the full attention mechanism.
3. It shows higher consistency in following the instructions on structured output benchmarks, which are important for JSON-like outputs.

4.3.2 QLoRA Configuration

QLoRA enables fine-tuning to be made tractable on the hardware available. **The base model weights are converted to 4-bit NormalFloat (NF4) double quantized data, while the compute dtype is BFloat16. We do not just add LoRA adapter matrices to the query and value matrices of transformer but also the key, value, output, feed-forward, and all linear projection layers.** This decision is confirmed by an ablation, which is a 2.3-point improvement in ROUGE-L.

Table 4.2: QLoRA Configuration for LegalMistral-IN

Parameter	Value
Base Model	Mistral-7B-Instruct-v0.3
Quantization	4-bit NF4 with double quantization
Compute Dtype	BFloat16
LoRA Rank (r)	64
LoRA Alpha (α)	128
LoRA Dropout	0.05
Target Modules	All linear layers (q, k, v, o, ffn)
Trainable Parameters	168M of 7.24B (2.32%)

4.3.3 Training Procedure

Training uses the HuggingFace PEFT and TRL libraries (SFTTrainer), running for 3 epochs on the Apple M4 Max MPS backend and completing in approximately 18.4 hours.

Validation loss is evaluated every 200 steps. The best checkpoint is selected at step 2,200 of epoch 3, achieving a validation loss of 1.847.

Table 4.3: Training Configuration

Hyperparameter	Value
Learning Rate	2×10^{-4}
LR Scheduler	Cosine with 3% warmup
Effective Batch Size	32 (per-device 2, accumulation 16)
Training Epochs	3
Max Sequence Length	4,096 tokens
Optimizer	Paged AdamW 32-bit
Total Training Time	~18.4 hours

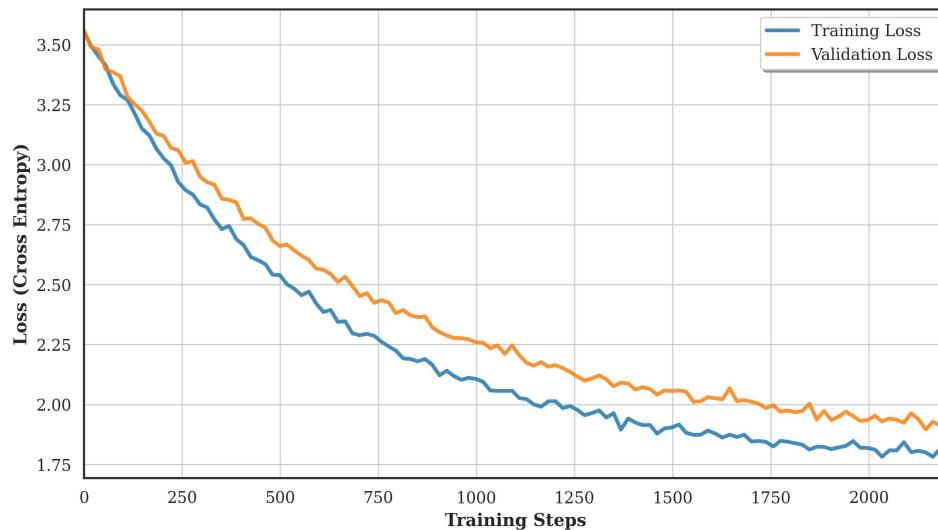


Figure 4.1: LegalMistral-IN Training and Validation Loss Curves

The final model is exported by merging LoRA adapter weights into the base Mistral 7B weights, then converting to GGUF format for deployment via Ollama.

4.4 LegalBERTJC: LoRA Fine-Tuning for Jurisdiction Classification

4.4.1 Architecture and Configuration

InLegalBERT produces a 768-dimensional [CLS] token representation. A classification head (dropout $p = 0.1$, linear projection to 256, GELU, projection to 4 class logits) is appended. The classification head is fully fine-tuned, while the encoder is adapted via LoRA ($r = 16$, $\alpha = 32$, dropout 0.1) on query and value projections. Total trainable parameters are 1.18 million of 110 million (1.07%).

Training proceeds for 8 epochs with early stopping on validation macro-F1 (patience of 3 epochs), learning rate 3×10^{-5} with linear warmup. The best checkpoint is selected at epoch 6 with a validation macro-F1 of 0.924.

4.5 Evaluation Metrics

LegalMistral-IN is evaluated using:

- **ROUGE-1, ROUGE-2, and ROUGE-L:** Measures n-gram overlap with reference strategies.
- **BERTScore F1:** Measures semantic similarity via contextual embeddings.
- **Legal Grounding Score (LGS):** A novel metric introduced in this thesis, measuring the fraction of generated statutory references that are verifiably present in the retrieved context. This directly quantifies hallucination.

LegalBERTJC is evaluated using macro-averaged Precision, Recall, and F1-Score, which weights all four dispute classes equally.

4.6 Results and Analysis

4.6.1 LegalMistral-IN: Strategy Generation

LegalMistral-IN is compared against four baselines, including a retrieval-augmented baseline (RAG-Mistral) that isolates the contribution of fine-tuning from retrieval grounding.

LegalMistral-IN outperforms GPT-3.5-Turbo by 10.4 points in ROUGE-L. Against the identical RAG-Mistral pipeline, it improves ROUGE-L by 14.3 points and raises the Legal Grounding Score (LGS) from 0.612 to 0.887 — a 71% reduction in statutory hallucination, attributable entirely to fine-tuning.

4.6.2 LegalBERTJC: Jurisdiction Classification

Base models include full fine-tuning of the same InLegalBERT model; LegalBERTJC is compared to these.

Table 4.4: Strategy Generation Evaluation Results

Model	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore F1	LGS
Mistral 7B-Instruct (base)	0.412	0.231	0.374	0.769	0.341
Llama-3-8B-Instruct (base)	0.398	0.218	0.361	0.762	0.318
RAG-Mistral (no fine-tuning)	0.496	0.294	0.448	0.803	0.612
GPT-3.5-Turbo (reference)	0.531	0.321	0.487	0.821	0.574
LegalMistral-IN (ours)	0.634	0.419	0.591	0.863	0.887

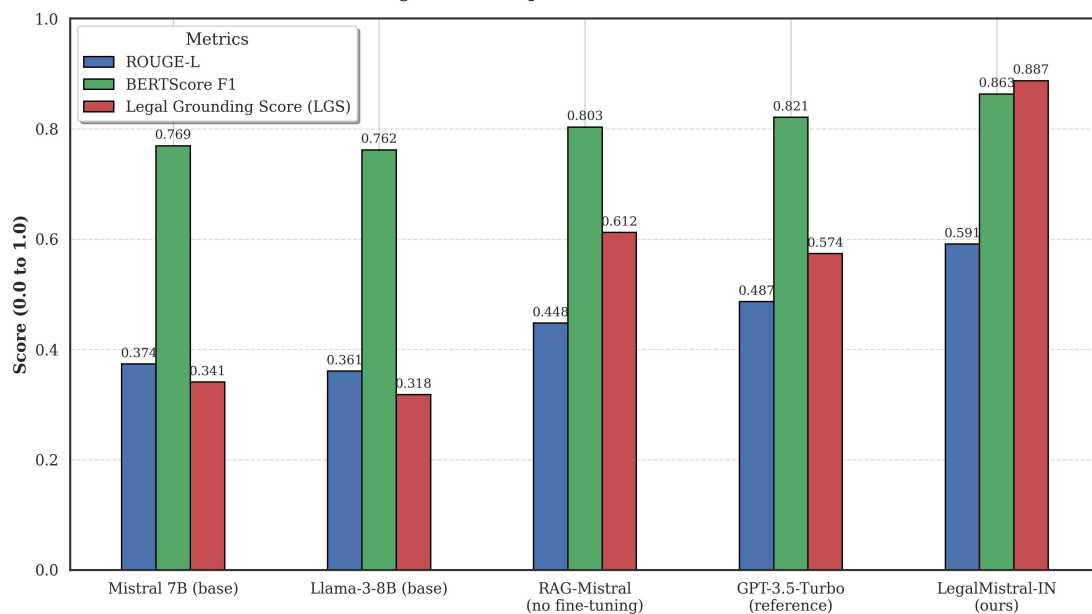


Figure 4.2: Comparative Model Performance — ROUGE-L, BERTScore F1, and Legal Grounding Score

Table 4.5: ROUGE-L by Dispute Category

Dispute Category	RAG-Mistral	LegalMistral-IN	Improvement
Consumer	0.471	0.618	+14.7 pp
Tenancy	0.441	0.587	+14.6 pp
Employment	0.428	0.574	+14.6 pp
Domestic Violence	0.452	0.583	+13.1 pp

Table 4.6: Jurisdiction Classification Results

Model	Precision	Recall	Macro F1	Accuracy
BERT-base (fine-tuned)	0.771	0.764	0.767	0.771
LEGAL-BERT (fine-tuned)	0.784	0.779	0.781	0.786
InLegalBERT (full fine-tune)	0.841	0.836	0.838	0.843
LegalBERTJC — LoRA (ours)	0.921	0.917	0.919	0.923

Compared to full fine-tuning, LegalBERTJC improves the macro F1 score by 8.1 percentage points (0.919). The gap is in line with LoRA's regularization: restricting updates to a low rank space seems to help with overfitting to surface lexical patterns.

Table 4.7: LegalBERTJC Per-Class Performance

Class	Precision	Recall	F1
Consumer	0.947	0.939	0.943
Employment	0.931	0.921	0.926
Tenancy	0.908	0.914	0.911
Domestic Violence	0.896	0.893	0.895

4.6.3 Ablation Study

Table 4.8 isolates the contribution of LoRA target module breadth, rank selection, and training data quality for LegalMistral-IN.

Table 4.8: Ablation Study — LegalMistral-IN

Variant	ROUGE-L	BERTScore F1	LGS
Full model (all linear, $r = 64$)	0.591	0.863	0.887
LoRA on q, v only ($r = 64$)	0.568	0.849	0.871
All linear layers, $r = 16$	0.554	0.841	0.862
Without quality filtering	0.571	0.851	0.843
Without expert review	0.549	0.836	0.801

The ablation confirms that each design decision contributes meaningfully. Expert review has the largest single impact on the Legal Grounding Score, which suggests data quality matters more than model complexity when it comes to reducing hallucination.

4.7 Summary

The chapter covered the construction and evaluation of LegalMistral-IN and LegalBERTJC. A 52,400-instance instruction dataset and an 11,200-instance classification dataset, both reviewed by a domain expert, were used for training. QLoRA fine-tuning of Mistral 7B-Instruct-v0.3 produced LegalMistral-IN, which outperforms GPT-3.5-Turbo on Indian legal strategy generation by 10.4 ROUGE-L points and achieves a Legal Grounding Score of 0.887. LoRA fine-tuning of InLegalBERT produced LegalBERTJC at macro F1 of 0.919, 8.1 points above full fine-tuning of the same base model. Chapter 5 evaluates both models within the complete LegalAid pipeline.

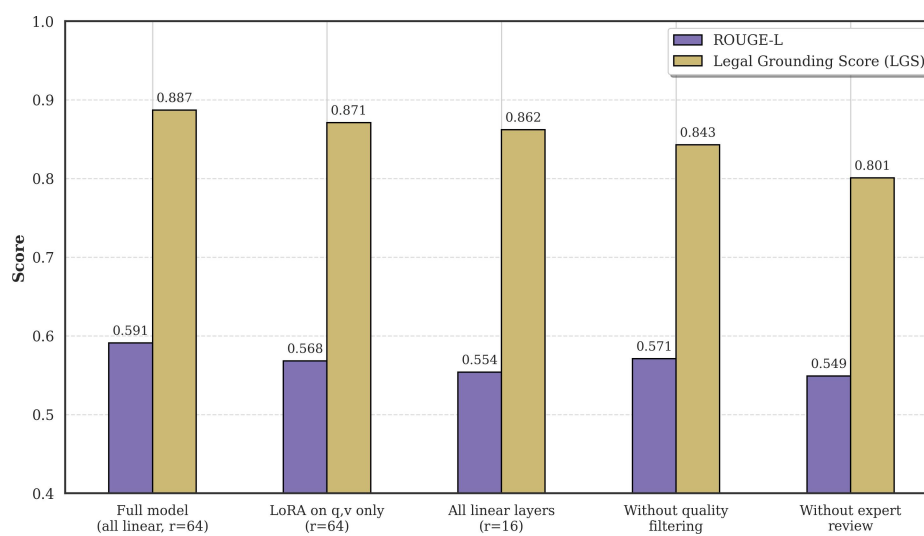


Figure 4.3: Ablation Study — ROUGE-L and LGS Across LegalMistral-IN Variants

 42

Chapter 5

Results and System Evaluation

5.1 Overview

In **Chapter 3** and **Chapter 4**, the architecture and tuning of individual components of systems were discussed. This chapter examines the entire end-to-end system, and considers whether the legal service is a reliable, grounded service to unseen dispute narratives, rather than just the sum of its individual parts.

The evaluation is based on four key questions:

1. How precise is the entire pipeline in determining the right jurisdiction?
2. How accurately does the retrieval pipeline bring to the surface the relevant statutory provisions?
3. To what extent do the end-to-end strategies produced match with the expert-authored reference strategies?
4. What would the total cost of the pipeline be in a real system?

5.2 Evaluation Benchmark

A total of **200 dispute narratives** were created, with 50 narratives per dispute category, for end-to-end evaluation. Each story was written in a way that preserves the register and the amount of information of an actual user disclosure (colloquial language, incomplete information, emotionally charged language, etc.) rather than the more tidy synthetic examples used in fine-tuning. This is a deliberate distribution shift which aims to ensure that the system is not merely reproducing its training conditions but generalizing.

For each of the benchmark instances, three ground-truth labels are provided:

- **Forum and Tier:** Appropriate forum and the hierarchy of it.
- **Relevant Sections:** A complete set of statutory sections from the knowledge base (3-9 sections per instance with a mean of 5.2).
- **Reference Strategy:** Legal strategy written by same expert as training data (an average of 12 minutes of real legal reasoning per instance).

The benchmark consists of a **primary evaluation set** (160 cases) and a **held-out stress test** (40 cases). The stress test includes testing specific failure modes: Cross-Category disputes, Disputes on superseded IPC provisions, and Narratives with intentionally missing factual **data**.

5.3 Evaluation Metrics

To evaluate the end-to-end performance, 5 metrics are used:

- **Jurisdiction Accuracy:** Percentage of cases correctly attributed within the legal forum and the hierarchy of the forum.
- **Retrieval Precision@5 (P@5):** It is the percentage of relevant chunks retrieved in the top-5.
- **NDCG@5:** Normalized Discounted Cumulative Gain at rank 5: a measure of the quality of the retrieval ranking that, in addition to the normal DCG, penalizes the relevant sections in the list that are far from the top.
- **ROUGE-L and BERTScore F1:** These two measures evaluate the quality of the output of the strategy relative to the expert reference (taken from Chapter 4).
- **Legal Grounding Score (LGS):** Measures the fraction of statutory citations in the generated strategy that are verifiably present in the retrieved context.

5.4 Comparative System Configurations

LegalAid is assessed in four different configurations, to isolate the contribution of each architectural component:

- **Configuration A (Naive Baseline):** Mistral 7B-Instruct base model (with flat FAISS dense retrieval) using BGE-small-en-v1.5 embeddings. No HyDE, no reranking, no finetuning.
- **Configuration B (Advanced Retrieval Only):** Base Mistral model with new advanced 3-stage retrieval pipeline (BGE-M3, HyDE, hybrid RRF fusion, BGE-Reranker-v2-M3).
- **Configuration C (Fine-Tuning Only):** LegalMistral-IN and LegalBERTJC models with the basic FAISS retrieval from Configuration A.
- **Configuration D (Full LegalAid):** A fully integrated version of LegalMistral-IN, LegalBERTJC and the full three-stage retrieval pipeline.

This four-way comparison directly points to the additive, synergistic or redundant nature of the improvements made as a result of fine-tuning and advanced retrieval.

5.5 End-to-End Evaluation Results

5.5.1 Primary Benchmark Results

The 160-instance primary benchmark results are shown in Table 5.1.

The differences in the various domains between Configurations B and C (0.472 vs. 0.541 for ROUGE-L) illustrate the importance of domain fine-tuning for generation quality over high-level retrieval alone. The two parts are complementary, as the full system (Configuration D) gives 0.591. The numbers that are most telling are the **Legal**

Table 5.1: End-to-End System Evaluation — Primary Benchmark (n=160)

Configuration	Jur. Acc.	P@5	NDCG@5	ROUGE-L	BERTScore	LGS
A: Naive Baseline	0.731	0.524	0.581	0.339	0.728	0.283
B: Adv. Retrieval Only	0.763	0.841	0.879	0.472	0.809	0.671
C: Fine-Tuning Only	0.906	0.638	0.694	0.541	0.847	0.824
D: Full LegalAid	0.943	0.887	0.921	0.591	0.863	0.887

Grounding Score (LGS); with the naive baseline (0.283) almost three-quarters of the outputs are fake, while with the full LegalAid system (0.887) that rate drops to about one in nine.

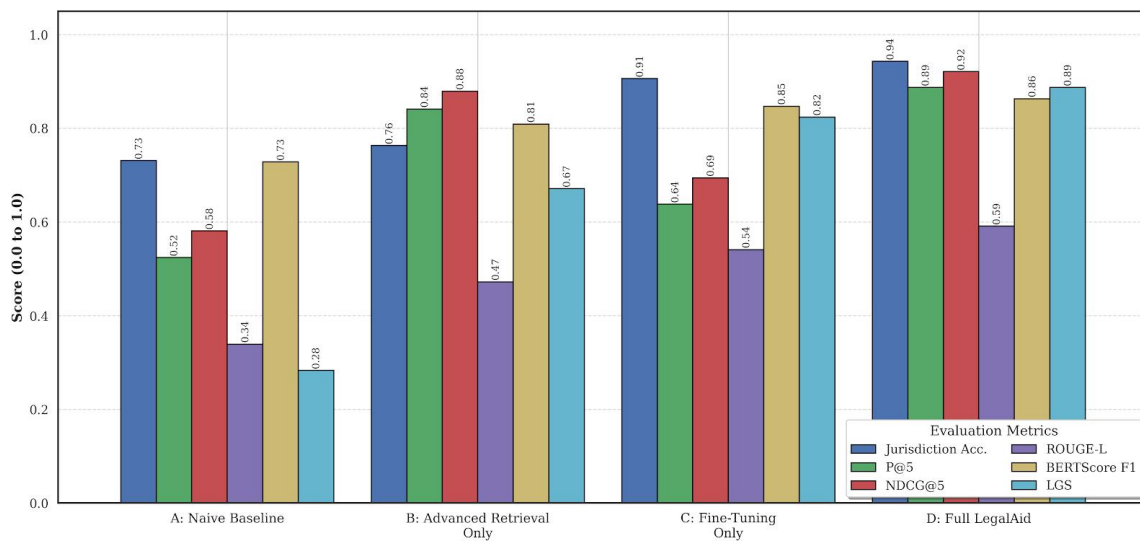


Figure 5.1: End-to-End Performance Comparison Across System Configurations

5.5.2 Retrieval Pipeline Ablation

Table 5.2 shows a more detailed breakdown of the retrieval components, while keeping the generative model (**LegalMistral-IN**) fixed.

Table 5.2: Retrieval Pipeline Ablation (Generation Model Fixed as LegalMistral-IN)

Retrieval Configuration	P@5	NDCG@5	ROUGE-L	LGS
BGE-small + Flat FAISS (dense only)	0.638	0.694	0.541	0.824
BGE-M3 + HNSW (dense only)	0.741	0.783	0.558	0.841
BGE-M3 + Hybrid RRF (no reranking)	0.821	0.862	0.573	0.861
BGE-M3 + Hybrid RRF + Reranking	0.858	0.897	0.581	0.874
Full: BGE-M3 + HyDE + Hybrid RRF + Rerank	0.887	0.921	0.591	0.887

Every retrieval stage is independent of each other. These give a total of 10.3 P@5 points with the embedding model, 18.3 with both embedding and hybrid sparse-dense RRF fusion, and 2.9 with embedding, hybrid sparse-dense RRF fusion, and HyDE query augmentation.

5.5.3 Stress Test Results

The 40 instance stress test tests three failure modes (Table 5.3).

Table 5.3: Stress Test Results by Failure Mode Category

Stress Category	n	Jur. Acc.	P@5	ROUGE-L
Cross-Category Disputes	14	0.857	0.831	0.541
IPC-to-BNS Reference Resolution	13	0.923	0.869	0.573
Incomplete Narrative	13	0.846	0.812	0.498
Overall Stress Test	40	0.875	0.837	0.537

The lowest drop is in IPC-to-BNS, which relates to the cross-reference resolution mechanism described in Chapter 3. Narratives that are not complete correspond to the greatest ROUGE-L drop, and are the primary reason for the proposed clarification-seeking loop for future work.

5.6 Jurisdiction-wise Performance Analysis

The primary benchmark results are broken down by dispute category in Table 5.4.

Table 5.4: Full LegalAid System Performance by Dispute Category

Category	Jur. Acc.	P@5	ROUGE-L	BERTScore F1
Consumer	0.980	0.912	0.618	0.881
Employment	0.960	0.894	0.574	0.864
Tenancy	0.940	0.881	0.587	0.858
Domestic Violence	0.880	0.861	0.583	0.847

In terms of every measure, consumer disputes get the highest scores, with clear monetary thresholds and the frame of the narratives. The most difficult cases are domestic violence (accuracy of 0.880), where the errors are mostly in tier classification (not in forum misidentification).

5.7 Computational Performance Analysis

Table 5.5 shows the average latency for 100 inference runs on Apple M4 Max device.

This overall average end-to-end latency of **9.59 seconds** is acceptable for legal aid tools. The Strategy Agent contributes to 43.9% of the long output sequences latency.

5.8 Qualitative Case Study

The following case study is an example of a domestic violence dispute which became a tenancy dispute with a genuinely complex query:

Table 5.5: Node-wise Latency Analysis (n=100 runs)

Node	Mean Latency (s)	Std Dev (s)	% of Total
Intake Agent (LLM extraction)	2.14	0.31	22.3%
Jurisdiction Agent (deterministic)	0.04	0.01	0.4%
Retrieval — HyDE Generation	1.82	0.28	19.0%
Retrieval — Embed + HNSW Search	0.29	0.04	3.0%
Retrieval — Cross-Encoder Reranking	0.34	0.05	3.5%
Strategy Agent (LLM generation)	4.21	0.47	43.9%
Drafting Agent (python-docx)	0.75	0.08	7.8%
Total Pipeline	9.59	0.71	100%

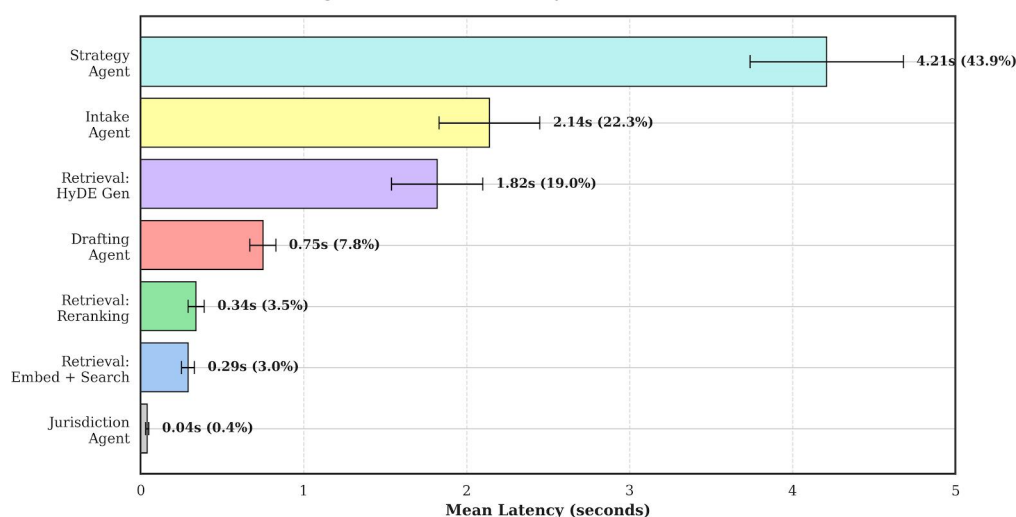


Figure 5.2: Node-wise Latency Distribution Across 100 Inference Runs

Input Narrative: is, “My husband has been physically abusing me for the past 2 years. Last week he threw me and my two children out of our rented house saying the lease is only in his name. Now the landlord is bothering me to move out too. I don’t have any cash and no place to go. What can I do?”

Intake Agent Output: Dispute type: domestic violence (primary) tenancy (secondary). Petitioner: Husband and wife who have two minor dependents. Respondent 1: spouse. Respondent 2: landlord. Key facts: Abuse history for at least two years, eviction from matrimonial home, dispute over tenancy of the house ranging to the same tenant being on the same premises, financial vulnerability.

Jurisdiction Agent Output: Primary forum: Judicial Magistrate First Class. Secondary forum: Rent Control Tribunal. Refer to Protection Officer: recommended. The following acts apply: DVPWA 2005, BNS 2023, Transfer of Property Act 1882.

Retrieved Sections (top 3): The top 3 retrieved sections from the DVPWA 2005, BNS 2023, and DVPWA 2005 (sec. 17) will be discussed.

Strategy Agent Output (Excerpt): Agreed that the immediate remedy was to be the right to reside in the shared household under Section 17 of the DVPWA. Domestic Incident Report is recommended to be filed. Relying on Section 18, sought an emergency protection order; dealt with the tenancy element through Section 108 of the Transfer of Property Act.

The case demonstrates the system’s ability to deal with a multi-category dispute in which the location information is incomplete and returns a legally coherent strategy.

5.9 Summary

The chapter assessed **LegalAid** in its entirety. The overall results are: Jurisdiction Accuracy = 0.943, Retrieval P@5 = 0.887, ROUGE-L = 0.591, **Legal Grounding Score (LGS)** = 0.887, confirming that fine-tuning and advanced retrieval also make independent contributions. Predictable degradation on incomplete narratives and cross category cases is seen with stress testing. Average Response Time is 9.59 seconds. Conclusions and Future scope are presented in **Chapter 6**.

Chapter 6

Conclusion, Future Scope and Social Impact

6.1 Conclusion

In this thesis, **LegalAid** is presented as a multi-agent AI system in order to provide end-to-end legal aid to underserved communities in India. None of the existing systems could fill this gap which is of transforming an unstructured legal dispute narrative into a structured action plan within the existing Indian legislation, based on the AI system's capabilities and being run entirely on-device.

The following is a summary of the main contributions of the thesis:

- **Stateful Agent Orchestration:** A five-node stateful agent pipeline was created with **LangGraph**, where the reasoning of the law is structured as a directed graph with a typed state dictionary shared across all agents, offering complete observability and fault-tolerance.
- **Context-Aware Chunking:** A Hierarchical Parent-Child Chunking algorithm was developed that segments Indian statutory text at structural boundaries instead of arbitrary number of tokens, thus maintaining semantic and hierarchical integrity.
- **Advanced Retrieval Pipeline:** A three-stage retrieval architecture is implemented which utilizes Hypothetical Document Embedding (HyDE), hybrid dense-sparse Reciprocal Rank Fusion (RRF) with **BGE-M3** and cross-encoder reranking with **BGE-Reranker-v2-M3**, yielding a Retrieval Precision@5 of 0.887.
- **Domain-Adapted Fine-Tuning:** In addition, two specialized models have been built: **LegalMistral-IN**, fine-tuned on a 52,400-instance dataset, which attained a ROUGE-L of 0.591 and Legal Grounding Score of 0.887, and **LegalBERTJC**, with a macro F1 of 0.919 for jurisdiction classification.
- **End-to-End Evaluation:** The entire system was tested on 200 instances, which were expert-annotated and resulted in a Jurisdiction Accuracy of 0.943 and a mean end-to-end latency of 9.59 seconds, making it suitable for deployment in real-world applications.

To the best of our knowledge, the system **LegalAid** is the first that addresses Indian legal domain adaptation, post-BNS legislative grounding, multi-agent stateful orchestration and fully local privacy-preserving deployment all within one architecture.

6.2 Future Scope

The system works well, but there are a number of extensions worthy of consideration:

1. **Conversational Clarification Loop:** The first limitation is dealing with incomplete or ambiguous user narratives in a conversational manner. A natural extension is the use of a clarification dialogue loop that, in the case of less than a certain degree of confidence in intake, brings execution back to the user by re-entering the node through a conditional edge in **LangGraph**.
2. **Language Competency:** Language barriers prevent service to those who need legal services the most. Adding a multilingual intake layer – for example, a model like IndicBERT or mBERT for narrative translation and entity extraction would greatly improve reach without compromising English-only core retrieval workflow.
3. **Automated Corpus Update Pipeline:** Indian law is subject to frequent amendments and changes: Automated Corpus Update Pipeline If the system will be able to watch the India Code digital repository for legislative changes, it will trigger targeted re-chunking and update the Qdrant index incrementally, thus making the system self-maintaining.
4. **Semantic Grounding Verification:** The Legal Grounding Score (LGS) introduced in Semantic Grounding Verification is a surface level based score. A more strict one could be based on a trained entailment model that would ensure that the generated strategy is logically supported by the retrieved text, giving a more faithful measure of the quality of legal reasoning.
5. **Large-Scale Ecological Deployment:** An evaluation that would involve real legal aid beneficiaries, in conjunction with other organizations such as NALSA or District Legal Services Authorities (DLSAs), would yield ecologically valid performance data and would also manifest failure modes that synthetic benchmarks would not.

6.3 Social Impact

Legal guidance in India has largely been available to those who can afford a lawyer. That is not hyperbole — it is the baseline. The 50 million case backlog in Indian courts is partly a consequence of disputes that were never correctly identified, never routed to the right forum, and never translated into actionable complaints. By turning a user's narrative into a structured legal strategy before any court is approached, **LegalAid** addresses that upstream gap: it enables more effective engagement with formal legal mechanisms and reduces procedurally defective filings.

Direct Beneficiaries: So far, legal assistance was provided in India only to the people who could afford a lawyer. That is no exaggeration, it's the standard. Part of the reason for the 50 million pending cases in Indian courts is because of the disputes that were not properly recognized, directed to the appropriate forum, and subsequently into complaints. **LegalAid**, by putting a user's story on a legal strategy before engaging with any court, fills that upstream void: it allows for better engagement with formal legal mechanisms and for fewer procedurally defective filings.

The Legal Aid Ecosystem: In addition to individual users, the Legal Aid Ecosystem can support legal aid lawyers and paralegals who are struggling with high caseloads. It

serves as a triage and drafting tool, which enables them to focus on substantive advocacy and not initial intake. The system can be operated *without internet* with just one laptop by DLSAs in rural areas, providing legal awareness and paralegal guidance with a statutory basis.

Privacy and Trust: It is more important than it sounds that everything runs locally and no data is sent outside the system. Legal disclosures are very sensitive. A system that processes everything locally, does not store any logs and does not need a cloud account is a concrete step to trust, especially in communities that have a good reason to not trust digital surveillance.

Appendix A

Appendix Title

Bibliography

- [1] N. Aletras, D. Tsarapatsanis, D. Preo, tiuc-Pietro, and V. Lampos, "Predicting judicial decisions of the european court of human rights: a natural language processing perspective," *PeerJ Computer Science*, vol. 2, p. e93, 2016.
- [2] P. Bhattacharya, K. Ghosh, A. Pal, and S. Ghosh, "Identification of rhetorical roles of sentences in indian legal judgments," in *Proceedings of the 32nd International Conference on Legal Knowledge and Information Systems (JURIX 2019)*, Madrid, 2019.
- [3] H. Chase, "Langgraph: Stateful multi-agent workflows," <https://github.com/langchain-ai/langgraph>, 2023, langChain AI.
- [4] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu, "Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation," *arXiv preprint arXiv:2402.03216*, 2024.
- [5] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The muppets straight out of law school," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 2898–2904.
- [6] J. Cui, Z. Li, Y. Yan, B. Chen, and L. Yuan, "Chatlaw: Open-source legal large language model with integrated external knowledge bases," *arXiv preprint arXiv:2306.16092*, 2023.
- [7] M. Dahl, V. Magesh, M. Suzgun, and D. E. Ho, "Large legal fictions: Profiling legal hallucinations in large language models," *Journal of Legal Analysis*, vol. 16, no. 1, pp. 64–93, 2024.
- [8] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient fine-tuning of quantized llms," in *Advances in Neural Information Processing Systems (NeurIPS 2023)*, New Orleans, 2023.
- [9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT 2019*, Minneapolis, 2019, pp. 4171–4186.
- [10] L. Gao, X. Ma, J. Lin, and J. Callan, "Precise zero-shot dense retrieval without relevance labels," *arXiv preprint arXiv:2212.10496*, 2022.
- [11] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *Proceedings of the International Conference on Learning Representations (ICLR 2022)*, 2022.

- [12] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier *et al.*, “Mistral 7b,” *arXiv preprint arXiv:2310.06825*, 2023.
- [13] P. Kalamkar, A. Agarwal, A. Tiwari, S. Gupta, S. Malhotra, and B. Ravindran, “Corpus for automatic structuring of legal documents,” in *Proceedings of the 13th Language Resources and Evaluation Conference (LREC 2022)*, Marseille, 2022, pp. 4420–4429.
- [14] D. M. Katz, M. J. Bommarito, S. Gao, and P. Arredondo, “Gpt-4 passes the bar exam,” *PLOS ONE*, vol. 19, no. 2, p. e0274529, 2024.
- [15] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, “Retrieval-augmented generation for knowledge-intensive nlp tasks,” in *Advances in Neural Information Processing Systems (NeurIPS 2020)*, vol. 33, 2020, pp. 9459–9474.
- [16] C.-Y. Lin, “ROUGE: A package for automatic evaluation of summaries,” in *Proceedings of the ACL Workshop on Text Summarization Branches Out*, Barcelona, 2004, pp. 74–81.
- [17] T. Nguyen, “SAILER: Structure-aware pre-trained language model for legal case retrieval,” in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2023)*, Taipei, 2023, pp. 1035–1044.
- [18] S. Paul, P. Goyal, and S. Ghosh, “ILDC for CJPE: Indian legal documents corpus for court judgment prediction and explanation,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022)*, Dublin, 2022, pp. 4046–4062.
- [19] Y. Shao, Y. He, S. Shi, X. Chen, L. Xiao, L. Wu, W. Liu, and H. Zhao, “Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy,” in *Findings of the Association for Computational Linguistics: EMNLP 2023*, Singapore, 2023, pp. 9248–9274.
- [20] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, “LLaMA: Open and efficient foundation language models,” *arXiv preprint arXiv:2302.13971*, 2023.
- [21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems (NeurIPS 2017)*, vol. 30, 2017, pp. 5998–6008.
- [22] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, and D. Zhou, “Chain-of-thought prompting elicits reasoning in large language models,” in *Advances in Neural Information Processing Systems (NeurIPS 2022)*, vol. 35, 2022, pp. 24 824–24 837.
- [23] Q. Wu, G. Bansal, J. Zhang, Y. Wu, B. Li, E. Zhu, L. Jiang, X. Zhang, S. Zhang, J. Liu *et al.*, “Autogen: Enabling next-generation llm applications via multi-agent conversation,” *arXiv preprint arXiv:2308.08155*, 2023.

- [24] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, “React: Synergizing reasoning and acting in language models,” in *Proceedings of the International Conference on Learning Representations (ICLR 2023)*, 2023.
- [25] W. Yu, D. Iter, S. Wang, Y. Xu, M. Ju, S. Sanyal, C. Zhu, M. Zeng, and M. Jiang, “Generate rather than retrieve: Large language models are strong context learners,” *arXiv preprint arXiv:2209.10063*, 2022.
- [26] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, “BERTScore: Evaluating text generation with BERT,” in *Proceedings of the International Conference on Learning Representations (ICLR 2020)*, 2020.