Physical Education, Health and Social Sciences

https://journal-of-social-education.org E-ISSN: 2958-5996

P-ISSN: 2958-5988

A Comparative Analysis of Large Language Models for Extracting Judicial Entities

Amir Siddique [1], Ali Saeed [2]

[1] Applied Computing Technically Department, Faculty of Information Technology, University of Central Punjab, Lahore, Pakistan Email: amir017@hotmail.com
[2] Department of Software Engineering, Faculty of Information Technology, University of Central Punjab, Lahore, Pakistan

DOI: https://doi.org/10.63163/jpehss.v3i4.978

Abstract

Legal judgments contain rich structured information, but the content is typically embedded inside long, formal narrative text. Named Entity Recognition (NER) is therefore a practical building block for legal search, analytics, and document understanding. In Pakistan, however, there is limited comparative evidence showing how well current large language model (LLM) services extract judicial entities from Lahore High Court (LHC) judgments under consistent experimental conditions.

This thesis benchmarks four LLM families, ChatGPT, Gemini, Grok, and DeepSeek, for prompt-only judicial NER on a corpus of 500 LHC judgments. A 22-type entity schema is defined to reflect common information needs in case law, including parties, judges, citations, acts, sections, and dates. Gold labels are produced as token-level IOB tags and then converted into a gold JSON representation with reconstructed entity spans and offsets. For model inference, a batch API application submits a unified prompt template to each model and converts responses into a standardized prediction JSON format. A separate evaluation application matches prediction JSON against the gold JSON using strict span-level exact match rules and reports micro-averaged precision, recall, and F1, supported by qualitative error analysis.

Results show that the top three models perform closely under strict scoring, with Grok achieving the highest micro F1 (0.6854), followed by Gemini (0.6820) and ChatGPT (0.6783), while DeepSeek scores lower (0.5790). ChatGPT produces the highest precision (0.7366), whereas Grok and Gemini achieve higher recall (0.6628 and 0.6534), indicating different operating points that matter for deployment. The error analysis highlights recurring legal-specific challenges, particularly span boundary drift, citation formatting variability, and confusions among closely related legal labels.

Overall, the thesis provides a reproducible, schema-driven benchmarking pipeline for Pakistani legal NLP and offers practical guidance on selecting and integrating LLM-based extraction into legal information systems where precision, recall, and auditability must be balanced.

Keywords: legal NLP, judicial named entity recognition, Lahore High Court, large language models, IOB tagging, prompt-based information extraction, evaluation pipeline

Introduction

Courts produce long-form legal narratives that contain many structured facts, such as party names, judges, dates, case numbers, statutory provisions, and cited precedents. When these facts remain inside plain text, legal search and analysis becomes slower because users must manually scan documents to locate the details they need. This is a practical problem for lawyers, researchers, students, and institutions that work with large judgment collections.

In Pakistan, reported Lahore High Court (LHC) decisions are publicly accessible through official web portals [1]. While this improves access to justice, it does not automatically provide structured metadata for information retrieval. To support advanced search, citation linking, and analytics,

legal technology systems need tools that can convert judgment text into structured fields in a consistent way.

Named Entity Recognition (NER) is one of the most useful NLP tasks for this purpose. NER identifies spans in text that refer to meaningful categories, for example judges, parties, courts, case identifiers, dates, acts, law sections, and citations. In legal pipelines, NER is often a foundational component that supports downstream tasks such as relation extraction, summarization, and knowledge-based construction.

Legal NER differs from general-domain NER because judgments are long, formal, and contain citations and statute references with highly variable formatting. For this reason, legal NLP research has emphasized domain adaptation and legal-domain resources. LEGAL-BERT provides legal-domain pretrained transformer models and shows that domain-specific pretraining and fine-tuning decisions can affect legal task performance [2]. LexGLUE highlights that evaluation in legal language should cover multiple tasks and datasets because models do not always generalize across legal subdomains and document types [3].

Dataset work also shows that legal entities tend to be more fine-grained than the common news entities. In Legal NER, for example, introduces an annotated corpus of court judgments and uses legal-specific entity types beyond standard PERSON, ORGANIZATION, and LOCATION categories [4]. These directions matter for Pakistani case law because LHC judgments contain entity patterns that are uncommon in general corpora, such as compound case titles, multiple citation formats, and references to acts and sections that require careful span boundaries. However, results from widely studied jurisdictions do not automatically transfer to Pakistan. Pakistani judgments often include local naming conventions, region-specific organizations and locations, and citation patterns that differ from US and EU corpora. Earlier work on Lahore High Court decisions demonstrates that information mining from Pakistani judgments is feasible and useful, but also confirms that local patterns and document structure affect extraction quality [5, 6]. In addition, Pakistan has low-resource language realities: even though this thesis focuses on English judgments, the broader ecosystem includes Urdu and mixed-language legal text, and Urdu NER research has shown that language and resource constraints influence extraction performance and dataset design [7].

Traditional legal NER systems typically frame the task as sequence labelling and rely on supervised training over annotated corpora. While these methods can achieve strong accuracy when enough labeled data exists, building a high-quality legal NER corpus is expensive because annotation requires clear guidelines and domain knowledge. In addition, changes in document layout or entity definitions can require re-annotation and model retraining.

Large Language Models (LLMs) offer an alternative workflow. Instead of training a task-specific model, an LLM can be prompted to perform extraction and return structured output. Recent surveys describe a growing trend in treating information extraction as a generative task, where models output structured representations under prompt constraints and are then validated by parsers or lightweight rules [8]. For NER specifically, prompt-based methods such as Prompt NER show that providing entity definitions and using structured prompting can improve cross-domain extraction performance [9]. These findings motivate evaluating LLMs for judicial NER in Pakistan where domain shift and formatting variation are common.

At the same time, legal NER places strict requirements on span boundaries and label consistency. Small boundary shifts, for example missing part of a case citation or truncating a statute title, can break citation linking and downstream indexing. LLM services also differ in training, alignment, and context handling, which can influence how they treat long legal spans and semi-structured headings. This thesis therefore evaluates four widely used LLM families, ChatGPT, Gemini, Grok, and DeepSeek, under a fixed entity schema and strict span-level scoring, using consistent prompt constraints so that comparisons are meaningful.

Although LLMs are widely used for text generation, there is limited publicly documented evidence about how current LLM services perform on Pakistani judicial entity extraction when evaluated against human-annotated references. Without careful evaluation, it is difficult for practitioners to decide which model is appropriate for legal search, citation linking, and analytics in Pakistan.

Comparative results are also often hard to reproduce because studies use different prompts, entity schemes, and scoring rules. This thesis addresses these issues by evaluating four widely used LLM families, ChatGPT, Gemini, Grok, and DeepSeek, using the same entity taxonomy, the same prompt constraints, and the same strict evaluation protocol. The goal is to quantify trade-offs between precision and recall, and to analyses error patterns that matter for real legal workflows.

This study evaluates prompt-only judicial Named Entity Recognition on Lahore High Court judgments by comparing ChatGPT, Gemini, Grok, and DeepSeek under a shared entity schema and a strict evaluation protocol. A domain-specific taxonomy of 22 judicial entity types is defined, a manually annotated reference dataset is prepared using IOB tagging, and a prompt-and-parse pipeline is designed to enforce consistent, machine-readable outputs. Model performance is assessed using strict span-level precision, recall, and F1 scores, complemented by category-level analysis and error examination to identify common failure patterns and improve reliability. The scope is limited to English-language LHC judgments and focuses solely on entity extraction, contributing practical insights for legal NLP applications such as precedent retrieval, citation analysis, and judicial metadata indexing

The structure of this synopsis is as follows: Chapter 2 reviews related work in NER, legal NLP, and LLM-based information extraction, with emphasis on how domain and jurisdiction affect extraction quality. Chapter 3 describes the dataset, annotation guidelines, entity schema, prompting strategy, post-processing, and evaluation metrics. Chapter 4 reports experimental results and provides detailed error analysis across entity types and models. Chapter 5 concludes the thesis and outlines future directions, including hybrid pipelines that combine prompting with rule-based checks, and broader evaluation across other Pakistani courts and document formats.

Literature Review

Named entity recognition is typically formulated as a sequence labeling problem where each token in a sentence (or document) is assigned a label indicating whether it begins, continues, or lies outside an entity mention. Early shared tasks in information extraction standardized entity categories (for example, person, organization, and location) and provided evaluation protocols that shaped later work in both general-domain and specialized-domain NER. The Message Understanding Conferences (MUC) helped formalize the named entity task definition, while CoNLL shared tasks popularized comparable corpora and micro-averaged precision, recall, and F1 for span-level evaluation under exact-match criteria. [14], [15]

In the legal domain, the same high-level formulation applies, but texts introduce additional complexity: frequent citations, section and clause references, nested entities (for example, a statute name containing a year), long-range dependencies across paragraphs, and jurisdiction-specific terminology. These factors motivate domain-adapted models, curated entity schemas, and careful annotation guidelines. Recent surveys of NLP and law emphasize the rapid expansion of tasks, datasets, and languages, but also note uneven coverage across jurisdictions and limited resources for many legal systems outside North America and Europe. [37], [36]

Named Entity Recognition: Task Definitions and Evaluation

NER aims to locate entity mentions in text and assign each mention a semantic type. In classic formulations, the set of types is fixed and small, while later work expanded to fine-grained or hierarchical taxonomies. In MUC-7, named entities were defined as proper names and numeric expressions of interest, with guidelines describing what should be annotated and how boundaries should be marked. [14]

Most modern NER datasets adopt token-level tagging schemes such as IOB or BIO, in which the first token of an entity is tagged as B-TYPE and subsequent tokens are tagged as I-TYPE, while non-entity tokens are tagged as O. CoNLL-2003 established a widely reused benchmark and scoring practice based on exact span match, which remains common for evaluating entity extraction systems. [15]

Evaluation is usually reported using precision, recall, and F1 at the entity-span level. Exact-match scoring provides a strict view of system quality because a predicted entity must match both the label and boundary of the gold entity. In practical deployments, partial matches and near-miss errors can still be useful, so many applied studies complement strict F1 with qualitative error analysis or task-specific utility measures. [15], [36]

A major early direction in NER treated the task as structured prediction over label sequences. Conditional Random Fields (CRFs) became a standard approach because they model dependencies between neighboring labels while incorporating a rich set of hand-crafted features derived from tokens, orthography, and context windows. [16]

Neural approaches reduced reliance on manual features by learning representations directly from data. BiLSTM-CRF models combined contextual token encodings from bidirectional recurrent networks with a CRF output layer, improving accuracy on standard benchmarks while remaining compatible with BIO labeling constraints. [17]

Transformers shifted sequence modeling to attention mechanisms that can connect tokens over long ranges. The Transformer architecture introduced self-attention as the central operation for representing sequences, enabling parallel computation and richer contextual interactions. [18]

BERT extended this idea through large-scale pretraining with masked language modeling and next-sentence objectives, producing contextual representations that transfer effectively to NER after task-specific fine-tuning. For many domains, BERT-style encoders became strong baselines, and later work explored domain-adaptive pretraining to reduce the mismatch between pretraining corpora and target text. [19]

Legal NLP and Legal Named Entity Recognition

Legal texts differ from news or web data in both structure and purpose. Judicial opinions and judgments include legal reasoning, citations to statutes and precedents, formal party roles, procedural timelines, and jurisdiction-specific abbreviations. These properties create extraction targets that do not always align with general NER categories. As a result, legal NER often defines entity types such as case name, citation, statute, provision, judge, lawyer, and court. [37], [23]

Research in legal NLP has expanded across tasks including retrieval, classification, summarization, outcome prediction, and information extraction. Surveys emphasize that progress depends heavily on dataset availability and annotation choices, and they document a growing number of public benchmarks designed to standardize evaluation. [37], [36].

Resource	Text type	Main task(s)	Why it matters for this thesis
LexGLUE	Mixed (EU/US legal)	Benchmark suite for legal language understanding	Provides task diversity and encourages standardized evaluation [21]
Legal-BERT corpora	Legal corpora (EU/US)	Domain-adaptive pretraining resources	Shows benefits of legal-domain pretraining for downstream tasks [20]
CUAD	Contracts (EDGAR)	Clause and span extraction	Highlights long-document extraction

			challenges and expert annotation costs [24]
LEDGAR	Contracts (EDGAR)	Provision classification (multi-label)	Illustrates large label spaces and noisy real-world legal text [25]
E-NER	Corporate filings (EDGAR)	Legal NER corpus	Demonstrates degradation when general NER models are applied to legal text [26]
German Legal NER dataset (Lynx)	Court decisions (DE)	Fine-grained legal NER	Shows multi-role legal entities and fine-grained classes [27]
InLegalNER	Indian court judgments	Legal NER	Demonstrates legal NER in a South Asian common-law context [28]

Table 0-1: Representative Datasets and benchmarks used in legal NLP and extraction research

Domain-adaptive pretraining is a widely used strategy for handling legal language. Legal-BERT demonstrated that pretraining transformer encoders on large collections of legal text can improve performance on legal downstream tasks compared to general-purpose encoders. LexGLUE further promoted comparability by aggregating multiple legal tasks into a single benchmark suite. [20], [21].Practical legal NLP systems often rely on open-source pipelines that encode domain knowledge. LexNLP provides utilities for segmenting legal documents, extracting structured attributes such as dates and money, and performing legal-domain NER and other analyses. Blackstone offers a spaCy-based pipeline for English case law that includes NER labels tailored to legal documents, such as case names, citations, instruments, and provisions. These toolkits illustrate how legal information extraction typically combines learned models with rule-based normalization and post-processing. [22], [23]

LLMs for Information Extraction and NER

In the last few years, instruction-tuned LLMs have enabled a complementary approach to extraction: instead of training a dedicated token classifier, the model is prompted to return entities in a specified format (for example, a JSON list of spans). This reframes NER as a generation or structured prediction problem. The approach is attractive in low-resource settings because it can leverage general capabilities learned during pretraining and instruction tuning. [34], [10]

Prompt-based extraction can be organized in several ways. One line of work designs prompts that include entity definitions and a few examples, encouraging the model to follow a labeling scheme or output a structured list. Prompt NER is an example of a method that targets few-shot and cross-domain NER by carefully formatting the prompt and using entity definitions to guide extraction. [9]

Another line of work treats extraction as an interactive process. ChatIE demonstrates a multi-turn prompting strategy that decomposes information extraction into a conversation where the model is asked for different fields and then refined, allowing correction of earlier steps and improving controllability. [29]

A practical challenge in LLM-based extraction is output reliability. Free-form generation can produce invalid formats or hallucinated entities, especially when the prompt is underspecified or when the document is long. Recent research on constrained decoding and schema-based

generation suggests that enforcing structure during decoding can improve compliance with expected formats and reduce post-processing failures. [32]
Surveys of legal information extraction highlight that LLMs can reduce annotation effort, but they also warn that legal settings require careful validation due to high stakes and sensitivity to errors. In particular, entity boundaries, role labels (for example, petitioner vs respondent), and citation extraction often require domain rules and normalization beyond what a general model learns from broad web text. [36]

Error source	Common manifestation in legal judgments	Mitigation idea	Example technique
Boundary drift	Entity spans include extra words or miss parts	Explicit span rules; post-process spans	Exact-match validation with BIO conversion
Role confusion	Mixing party roles (petitioner, respondent) or actors (judge, lawyer)	Provide role definitions and examples in prompts	Few-shot prompting with definitions [9]
Citation variability	Multiple citation styles and abbreviations	Normalize via regex and legal citation rules	Hybrid pipeline with rule-based normalization [22], [23]
Hallucination	Generating entities not present in text	Restrict outputs to extracted spans; verification step	Multi-turn confirmation [29]
Long-context loss	Missing entities mentioned far apart	Chunking and overlap; retrieval of relevant parts	Chunked prompting with aggregation
Format non-compliance	Invalid JSON or inconsistent labeling	Constrained decoding; schema checks	Schema-guided generation [32]

Table 0-2: Typical error sources in LLM-based legal NER and mitigation strategies

Model Families Considered in This Thesis

This thesis compares four widely discussed LLM families: ChatGPT (OpenAI), Gemini (Google), DeepSeek (DeepSeek-AI), and Grok (xAI). These model lines differ in training data mixtures, alignment approaches, availability (open-weight vs API access), and optimization priorities. Because public reporting varies across providers, empirical evaluation on a fixed dataset is an important complement to specification-level descriptions. [10], [11], [13], [35]
The GPT-4 technical report documents strong reasoning and instruction-following performance and has influenced how subsequent work positions LLMs as general-purpose engines for downstream tasks via prompting and tool use. Gemini introduced a family of multimodal models aimed at strong performance across text and other modalities. DeepSeek-V2 reported an efficient mixture-of-experts design intended to reduce training and inference costs while maintaining strong capability. Grok-1 was later released in an open-weight form, increasing accessibility for controlled experimentation. [10], [11], [13], [35]
For judicial NER, these differences matter because extraction tasks stress both language understanding and disciplined output formatting. In addition, legal judgments can be lengthy, which can amplify differences in context handling strategies (for example, chunking policies, summarization, or retrieval-assisted prompting) used in an evaluation pipeline.

Judicial Text Mining in Pakistan and the Lahore High Court Context

Research on automated analysis of Pakistani judicial text is comparatively limited, despite growing public availability of judgments. The Lahore High Court publishes judgments and case information through its online portal, enabling the creation of corpora for research and development. [40]

Within Pakistan-specific work, Iftikhar and colleagues proposed information mining from criminal judgments of the Lahore High Court, highlighting the value of extracting structured fields from unstructured decisions for search and analytics. Related studies have explored data mining for smart legal systems and emphasized the need for indigenous datasets and tools tailored to local drafting styles. These works motivate continued investment in annotation guidelines, entity taxonomies, and evaluation resources for Pakistani case law. [38], [39]

A further complication for the Pakistani context is multilinguality. Even when judgments are primarily in English, they often include Urdu names, transliterations, and local terminology. Work on Urdu NER datasets and models demonstrates both progress and ongoing challenges in tokenization, orthographic variation, and limited labeled resources, which can indirectly affect legal-domain extraction when multilingual names appear in the text. [41], [42].

Summary of Research Gaps

The literature reviewed above highlights several gaps that motivate this thesis. First, most public legal NER resources target jurisdictions and document types outside Pakistan, so performance transfer to Pakistani judgments is uncertain and requires empirical validation. Second, even within legal NLP, many benchmarks focus on classification or contract clause extraction rather than role-sensitive judicial entity schemas common in court rulings. Third, while LLM prompting has emerged as a label-efficient alternative to supervised tagging, comparatively fewer studies evaluate multiple modern LLM families side by side on a consistent judicial dataset and a consistent evaluation protocol, especially for South Asian case law. Finally, practical legal extraction requires reliable output structure and low hallucination rates, suggesting that careful prompt design, post-processing, and verification should be evaluated together with core extraction accuracy. [36], [37]

Based on these gaps, the next chapter presents the thesis methodology, including dataset construction for LHC judgments, the entity schema and annotation guidelines, and the experimental design used to compare ChatGPT, Gemini, DeepSeek, and Grok under a common evaluation framework.

Materials and Methods

Materials: Data Source and Corpus Construction

The corpus used in this thesis consists of English-language Lahore High Court (LHC) judgments that are publicly available through official LHC reporting portals. LHC judgments are typically semi-structured: they contain headings, coram and bench information, party names and counsel listings, followed by narrative reasoning and references to statutes, case law, and procedural history. This mixture of semi-structured and free-form segments creates realistic challenges for entity boundary detection and legal citation extraction.

Document acquisition and selection criteria

Judgments were collected from the LHC reporting portal(s). To keep the dataset consistent and suitable for judicial NER, the following inclusion criteria were applied:

A. The document is a judgment PDF (not a notice, cause list, or administrative circular).
B. The judgment is primarily in English (short Urdu names or transliterations may still appear).
C. The judgment contains standard judicial metadata patterns (for example case number, parties, dates, or cited provisions)
D. The extracted text is readable and not severely corrupted by formatting issues.

Exclusion criteria included documents with incomplete text, duplicated entries, and items where the content could not be reliably tokenized due to heavy layout artifacts.

Corpus characteristics and dataset splits

The annotated dataset is treated as a gold standard for evaluation. Because this work does not fine-tune models, a classical train-test split is not required for learning.

Text extraction and preprocessing

Preprocessing aims to preserve the legal content while reducing noise that harms tagging consistency. The pipeline applies the following steps:

A. Convert judgment content into plain text while preserving paragraph boundaries where possible.
B. Normalize whitespace, line breaks, and repeated headers or footers.
C. Standardize quotation marks and remove control characters.
D. Tokenize text into sequences suitable for BIO or IOB tagging.
E. For long judgments, split into chunks with overlap to reduce context loss at chunk boundaries.

All preprocessing is deterministic to ensure repeatability across models.

Entity Schema and Annotation Methodology

A domain-specific entity taxonomy was defined to reflect judicial information needs for Pakistani case law. The schema targets entities that support search, filtering, citation linking, and structured legal analytics. Entities are annotated using an IOB tagging format at token level, with tags B-TYPE, I-TYPE, and O.

The entity schema contains 22 types. These include judicial metadata (case number, case name, court, judge), procedural and participant roles (petitioner, respondent, advocate), and legal references (act, law section, citation, precedent case), along with supporting types such as organization, location, and money where relevant.

Annotation guidelines and span rules

Annotation guidelines define what should be marked as an entity and how boundaries are selected. Key span rules include:

A. Exact span selection: only tokens that belong to the entity are included, excluding trailing punctuation unless it is part of the formal reference.
B. Long-form legal references: statute titles and provisions are captured as separate spans when the text supports both (for example ACT as the full statute name and AW_SECTION as section identifiers).
C. Citations and precedent cases: citations are marked based on explicit citation markers or reporter formats, while precedent case names are marked when the case title is present.
D. Party roles: petitioner and respondent spans are assigned using contextual cues from headings and the narrative.
E. **These rules are designed to support strict span-level evaluation**.

Annotation workflow and quality control

Annotation was carried out using an annotation utility that presents judgment text and allows span marking with the predefined label set. To reduce inconsistency, annotation proceeded in iterative passes:

A. Pilot annotation on a small subset to refine label definitions and span rules.
B. Full annotation using the locked schema and guidelines.
C. Consistency review to detect common errors such as missing B-tags, label drift, and inconsistent boundary selection.
D. Final conversion into token-level IOB tags for evaluation.

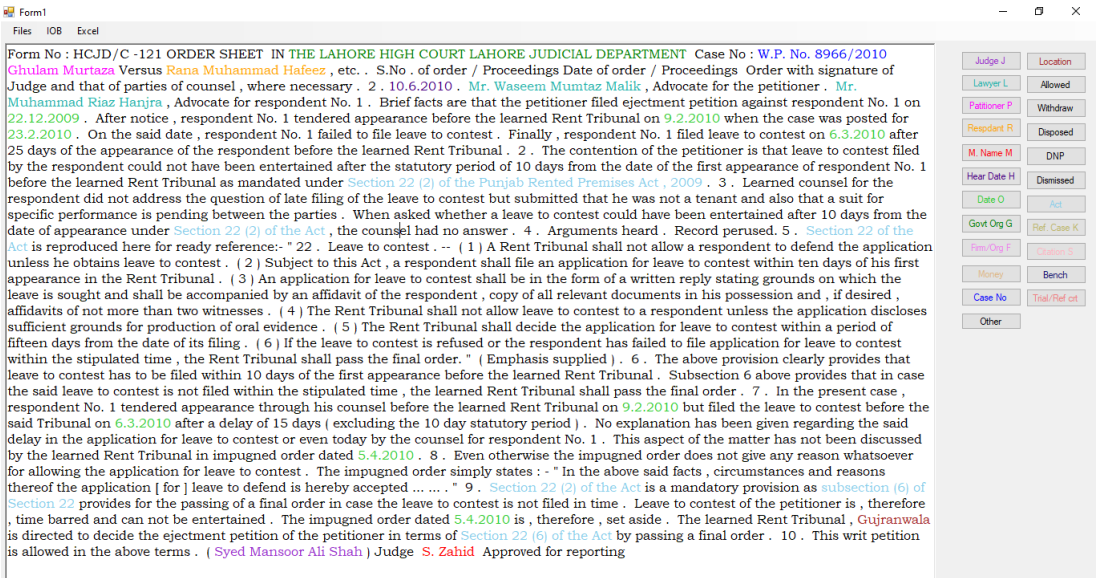

Figure 0-1 Annotation utility

Where multiple annotators are available, inter-annotator agreement can be calculated on a shared subset to quantify guideline clarity. If a single annotator is used, a second-pass review, spot checks, and rule-based validation help reduce systematic mistakes.

IOB file generation and gold JSON construction

After annotation, each judgment is transformed into a token-level IOB representation that serves as the gold reference for evaluation. The IOB scheme follows the widely used BIO/IOB convention adopted in benchmark NER datasets [15]. Each token is assigned exactly one tag: B-TYPE for the first token of an entity span, I-TYPE for subsequent tokens inside the same span, and O for tokens outside any entity.

The gold IOB output is stored as a plain text file per document, where each line contains a token and its gold label (and optional sentence boundary markers). This format is intentionally simple because it supports repeatable processing and reduces ambiguity when reconstructing entity spans during scoring.

For scoring and auditability, the gold IOB files are also converted into a structured gold JSON representation. The conversion step reconstructs entity spans by grouping contiguous B/I labels of the same type and then records: (i) entity_type, (ii) start and end token indices, (iii) start and end character offsets in the cleaned text, and (iv) the exact extracted surface form. Character offsets are preserved so that evaluation can be performed deterministically even when models operate over chunked inputs.

Entity type	Definition (summary)	Example (illustrative)
CASE_NUMBER	Case or petition identifier (including type and number)	Civil Revision No. 364-24
CASE_NAME	Short case title (party v party)	Muhammad Akram vs Sarfraz
COURT	Court name or bench	Lahore High Court
JUDGE	Judge or bench member	Mr. Justice [Judge Name]
DATE	Date reference (judgment date, hearing date)	12-03-2024

PARTY_PETITIONER	Petitioner/appellant name span	Muhammad Akram
PARTY_RESPONDENT	Respondent/defendant name span	Sarfraz
ADVOCATE	Counsel name span	Learned counsel for [Party]
ACT	Statute or act name	Pakistan Penal Code, 1860
LAW_SECTION	Section, clause, article references	Section 302
CITATION	Law report citation string	PLD 2019 SC 123
PRECEDENT_CASE	Name of cited case	Ghulam Hussain v Bahadar
ORGANIZATION	Institution or organization name	State Bank of Pakistan
LOCATION	Place names in case context	Lahore
MONEY	Monetary amounts	Rs. 500,000
CRIME_OFFENCE	Offence label when explicitly stated	Murder
FIR_NUMBER	FIR identifier when present	FIR No. 12/2020
POLICE_STATION	Police station reference	PS Model Town
PROCEDURAL_TERM	Procedural items where used as entities	writ petition
LAWYER_ROLE	Role markers for counsel where separable	Deputy Attorney General
GOV_ENTITY	Government department bodies	Punjab Police
DOCUMENT_TYPE	Judgment, order, petition, etc	Judgment

Table 0-3: Entity schema used for judicial NER in this thesis (22 types)

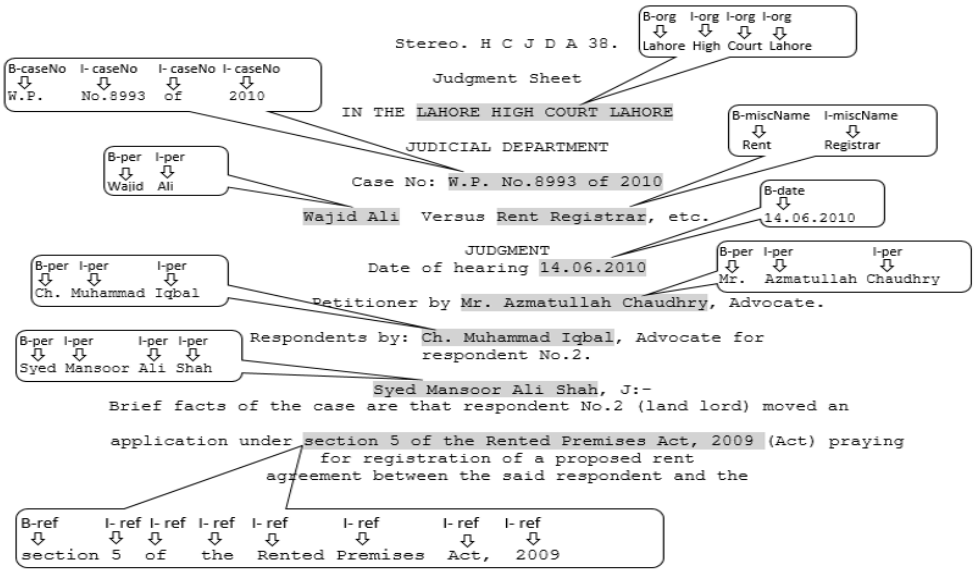

Figure 0-2 Judgment IOB

3.3 LLM Systems Under Evaluation

Four LLM families are evaluated. Each model is accessed through its publicly available interface (web application or API), and each is instructed using the same prompt and output specification.

Because providers update models over time, the experiment logs the model name, access method, and the date of execution for reproducibility.

ChatGPT (OpenAI)

ChatGPT represents OpenAI's instruction-following LLM family. The evaluation treats ChatGPT as a black-box service and focuses on extraction behavior under strict formatting constraints. The GPT-4 technical report is used as a primary reference for the underlying model family capabilities and alignment objectives.

Gemini (Google)

Gemini is Google's family of multimodal and text-capable LLMs. This thesis evaluates Gemini for its ability to follow extraction instructions and produce consistent token-level tags when faced with judicial citations and long-form judgment text.

Grok (xAI)

Grok is an LLM product line developed by xAI. This thesis evaluates Grok for judicial NER under the same prompting interface as other models. Public documentation about Grok-1 and its open release provides background on model scale and training configuration.

DeepSeek (DeepSeek-AI)

DeepSeek is a family of language models developed by DeepSeek-AI. DeepSeek-V2 is described as a mixture-of-experts model optimized for efficient inference. This thesis evaluates DeepSeek for judicial NER using the shared prompt and the shared evaluation protocol.

Prompt Design and Inference Protocol

To ensure fair comparison, a single prompt template is used across models. The prompt defines the entity label set, provides concise definitions, and specifies strict IOB constraints: one tag per token, valid labels only, and no hallucinated entities. The models are instructed to preserve token order and to avoid introducing new tokens.

Because output variability can affect structured extraction, decoding settings are chosen to be as deterministic as the interface allows (for example low temperature or equivalent). When deterministic settings are not exposed, multiple runs can be performed and majority voting can be used, but the primary protocol aims to minimize randomness.

Handling long judgments: chunking and aggregation

Judgments can exceed practical context limits or can degrade model performance due to long-range distractions. The pipeline therefore chunks each judgment into overlapping segments. For each chunk, the model returns IOB tags. Chunks are then merged using an overlap-resolution rule that prioritizes consistent spans and reduces boundary loss at chunk edges.

Chunk size and overlap are treated as controlled parameters and are fixed during evaluation after development tuning.

Output validation and post-processing

Model outputs are validated to ensure they can be scored. Validation includes:

A. Enforcing that every token has exactly one tag.
B. Mapping invalid or unknown tags to O.
C. Repairing illegal IOB transitions (for example an I-TYPE tag that starts an entity without a preceding B-TYPE is converted to B-TYPE).
D. Reconstructing entity spans from token tags for span-level scoring.
E. Post-processing does not add entities. It only normalizes outputs to a consistent representation for evaluation.

Controlled factor	Control strategy
Input texts	Same LHC judgments and same preprocessing pipeline are used for all models.

Entity schema	Same 22-type schema and same definitions across models.
Output format	Strict token-level IOB tags with one label per token.
Prompt	Same prompt template and constraints, fixed before final evaluation.
Chunking	Same chunk size and overlap, fixed during test runs.
Scoring	Same strict span-level micro precision, recall, and F1 for all outputs.
Logging	Model name, access method, execution date, and prompt version recorded.

Table 0-4: Experimental controls used to ensure comparability across models

Batch inference application and standardized JSON output

To run inference at scale, a dedicated batch inference application was used to call each provider API with the same prompt template and the same preprocessing pipeline. The application loads the cleaned text of each judgment, applies the selected chunking strategy for long documents, and then submits requests to the model endpoint. In this study, the batch runner processed 500 judgment files end-to-end, producing outputs in a consistent structure for every model.

The batch runner enforces a single JSON schema for all model outputs. Each model response is parsed and converted into JSON records that mirror the gold JSON format. At minimum, each record stores: document_id, model_name, chunk_id (if chunking is used), entity_type, start/end character offsets, and extracted text. Storing offsets and text together supports both strict scoring and manual inspection of failures. The application also logs execution metadata (run date, prompt version, and preprocessing parameters) to support reproducibility and regression testing.

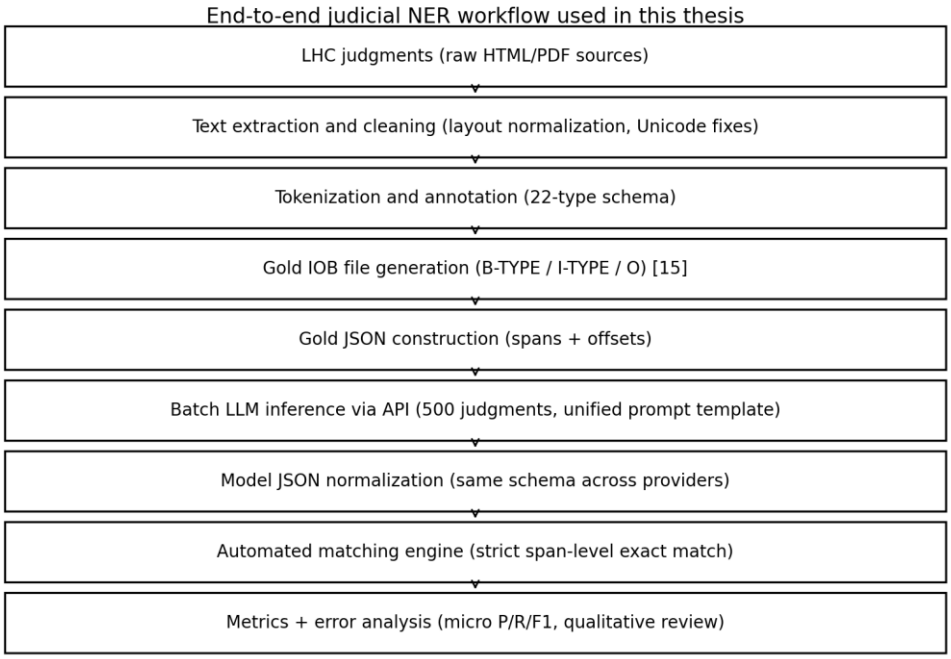

Figure 0-3 End-to-end workflow from judgments to evaluation outputs

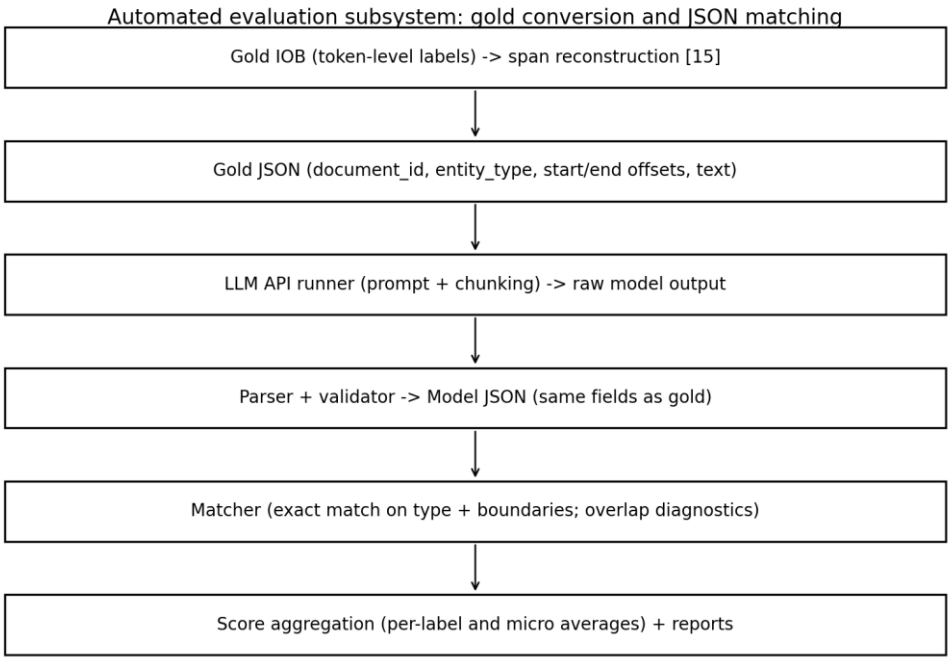

Figure 0-4: Evaluation subsystem: gold IOB conversion and JSON matching

Evaluation Methodology

Evaluation follows the standard NER practice of computing precision, recall, and F1 at the entity-span level. A predicted entity is counted as correct only if its type and span boundaries exactly match the gold annotation. Micro-averaging is used to aggregate counts across all entity types and documents, which is appropriate when entity frequency is imbalanced.

In addition to overall micro scores, the study reports per-entity-type performance to identify which legal entities are most difficult. A confusion analysis highlights label confusions (for example CITATION versus PRECEDENT_CASE, or ACT versus LAW_SECTION).

Metric definitions

Let TP be the number of correctly predicted entity spans, FP be the number of predicted spans that are incorrect, and FN be the number of gold spans missed by the model. Precision, recall, and F1 are computed as:

Precision = TP / (TP + FP)

Recall = TP / (TP + FN)

F1 = 2 * Precision * Recall / (Precision + Recall)

These metrics are reported for the full dataset and for individual entity types.

Error analysis protocol

Quantitative scores are complemented with qualitative error analysis. Errors are categorized into:

A. Boundary errors: predicted span is close to the gold span but includes extra tokens or misses tokens.
B. Type confusions: correct span but incorrect label, or label swaps between related categories.
C. Missed entities: entity present in gold but not predicted, often due to long-context effects or uncommon formatting.
D. Spurious entities: predicted entities not present in gold, often linked to over-generalization.
E. A sample of errors is inspected for each model to identify recurring patterns and to propose mitigation strategies.

Automated JSON matching and scoring application

A separate evaluation application performs automated matching between the model JSON output and the gold JSON derived from the IOB files. The evaluator reconstructs entity spans for both sides and compares them under strict span-level exact match rules. A prediction is counted as correct only if (i) the entity_type matches and (ii) the predicted span boundaries match the gold boundaries exactly, using character offsets to avoid ambiguity when chunking is applied.

The matching stage produces true positives, false positives, and false negatives at the entity level, then aggregates these counts to compute precision, recall, and F1. In addition to strict scoring, the evaluator logs diagnostic categories such as partial overlap, boundary drift, and type confusion. These logs support the qualitative error analysis reported later in the thesis and allow targeted improvement of prompts, validation rules, or segmentation strategy.

Validity, Reliability, and Reproducibility

Internal validity is addressed by holding constant the input data, preprocessing, prompt template, chunking strategy, and scoring rules. Construct validity is supported by using a judicially motivated entity schema and strict span-level scoring aligned with real indexing and citation-linking needs.

Reliability is improved through deterministic preprocessing, output validation, and complete experiment logging. Because model services can change, the thesis records the evaluation date and model identifiers and recommends re-running the benchmark when major provider updates occur.

Ethical Considerations and Data Management

The judgments used in this thesis are publicly available documents, which reduces privacy risk. Nevertheless, data handling follows responsible research practice. The dataset is used only for academic evaluation, and the thesis avoids unnecessary exposure of sensitive personal information beyond what appears in publicly reported judgments.

Annotated data, prompts, and evaluation scripts are stored with version control to support reproducibility. If the dataset is redistributed, licensing and portal terms should be checked and citations should be preserved to maintain attribution to the original sources.

Summary

This chapter presented the research design and methods used to compare four LLM families for judicial NER on LHC judgments. The next chapter reports the experimental results using strict span-level metrics and provides detailed error analysis across entity types and models.

Findings (Results)

For clarity, the key experimental conditions are summarized here. All four models received the same prompt template, the same label set (22 entity types), and the same output format requirements (token-level IOB tags). Inputs were produced using the same preprocessing and chunking strategy. Outputs were normalized using the same validation rules, and then scored using strict span-level exact match, where an entity counts as correct only if both its type and boundaries match the gold reference exactly.

Because the models were not fine-tuned, results should be interpreted as prompt-only extraction performance. This is important for practical adoption, since organizations often begin with prompt-based approaches before investing in supervised training or domain adaptation.

Overall Model Performance

Table 5 reports strict span-level micro-averaged precision, recall, and F1 for each model. Micro-averaging aggregates true positives, false positives, and false negatives across the entire evaluation set, which is appropriate when some entity types appear more frequently than others.

Across the four models, Grok achieves the highest F1 (0.6854), followed closely by Gemini (0.6820) and ChatGPT (0.6783). DeepSeek shows lower overall performance (0.5790). The results suggest that, under this protocol, Grok and Gemini provide slightly stronger overall

balance between precision and recall, while ChatGPT is the most conservative model (highest precision).

Model	Precision	Recall	F1-score
ChatGPT	0.7366	0.6286	0.6783
Gemini	0.7131	0.6534	0.6820
Grok	0.7097	0.6628	0.6854
DeepSeek	0.5975	0.5617	0.5790

Table 0-5: Overall strict span-level micro-averaged results for judicial NER

Ranking and differences in operating style

Although the top three models are close in overall F1, their operating styles differ in a way that matters for deployment. ChatGPT produces the highest precision (0.7366), meaning it tends to generate fewer false positives. This behavior is beneficial when extracted entities feed directly into a legal index or a citation graph, because false positives can introduce noisy links or misleading metadata.

Grok and Gemini achieve higher recall than ChatGPT, which increases coverage of entity mentions and improves overall F1. This can be preferable when missing entities is costly, for example when the goal is comprehensive retrieval of cited cases or statutory sections. DeepSeek, in this evaluation, produces both lower precision and lower recall, indicating that it both misses more entities and introduces more incorrect spans relative to the other systems.

Precision-Recall Trade-offs and Practical Interpretation

In judicial information extraction, the cost of an error depends on the use case. In a legal search engine, precision errors can degrade trust, because users may click into incorrect results caused by spurious entities. In contrast, in exploratory legal research, recall errors can be more damaging, because missing citations or parties can hide relevant decisions.

The findings suggest a practical interpretation:

A. If the primary goal is clean structured metadata for indexing and linking, ChatGPT's higher precision may reduce downstream cleaning effort.
B. If the primary goal is broader coverage of entities with fewer misses, Grok or Gemini may be preferable due to higher recall.
C. For production systems, a hybrid design can be considered, for example using a higher-recall model for candidate extraction and a high-precision model or rule-based validator for confirmation.

This thesis does not claim that one model is universally best. Instead, it shows measurable trade-offs that can guide model selection based on operational requirements.

Qualitative Error Analysis

Strict span-level evaluation highlights errors that are especially important in legal pipelines, because legal entities often have long spans and specialized formats. Across models, the most common error families are summarized below. The categories are consistent with the error analysis protocol defined in Chapter 3.

Boundary drift on long legal spans

Boundary drift occurs when a predicted entity overlaps with the correct entity but does not match the exact start and end positions. This error is frequent for multi-token party names, long statute titles, and citations that include reporter abbreviations, years, and page numbers. Even small boundary mistakes can break citation linking, because downstream systems typically require exact string matches or normalized patterns.

Boundary errors were observed in all models, but they have different causes. Some boundary errors appear to be caused by line-break artifacts in judgment headings, while others arise when

the model includes role words (for example, "petitioner") inside the entity span even when guidelines treat the role word as outside the name span.

Type confusion among closely related legal labels

Type confusion refers to predicting the correct span but assigning the wrong entity type. In judicial text, several labels are naturally close:

A. CITATION versus PRECEDENT_CASE: citations may appear alone (reporter strings) or beside case titles, and models may not consistently separate the two.
B. ACT versus LAW_SECTION: a provision reference may include both a statute title and a section number; models sometimes label the entire string as a single type.
C. PARTY roles: party names may be tagged correctly but mapped to the wrong role label (petitioner versus respondent) when headings are ambiguous or when the narrative refers to parties by description rather than by explicit role blocks.

These confusions indicate that prompt definitions and annotation guidelines must be tightly aligned, and that adding short role-specific examples in prompts can help reduce label swapping.

Missed entities due to long-context effects

Missed entities (false negatives) occur when entities present in the gold reference are not predicted. In long judgments, important entities may occur in earlier headings and then later reappear in reasoning sections. When a model focuses heavily on a local paragraph, it may miss entities mentioned earlier or later, especially if the evaluation pipeline uses chunking and the entity appears near a chunk boundary.

A practical mitigation is to use overlapping chunks and to merge predictions conservatively. Another mitigation is to run a lightweight rule-based pass for highly structured patterns, such as case number formats or section references, and then reconcile with model predictions.

Spurious entities and hallucination risk

Spurious entities (false positives) occur when the model labels a span that should be O under the annotation guidelines. In legal text, this can happen when the model generalizes from common judicial templates and labels a word as an entity even if the entity is not explicitly present as a named span. This issue is especially concerning in legal settings because it can create incorrect records in an index.

ChatGPT's higher precision suggests fewer spurious entities overall in this evaluation. However, no model is immune to hallucination risk, especially when prompts are permissive or when the text is noisy. Therefore, output validation and simple cross-checking rules are recommended for deployment.

Summary of Findings

This chapter presented the comparative results of prompt-only judicial NER across ChatGPT, Gemini, Grok, and DeepSeek. Grok achieved the highest overall F1, with Gemini and ChatGPT close behind, while DeepSeek performed lower under the same protocol. The qualitative analysis highlighted that most errors are driven by boundary drift on long legal spans, confusion among closely related legal labels, misses caused by long-context effects, and occasional spurious predictions

Discussion

The core empirical result is that the three leading systems are closely clustered in strict micro F1, with Grok slightly ahead, Gemini and ChatGPT close behind, and DeepSeek lower under the same protocol. Because legal extraction pipelines often have different tolerance for false positives and false negatives, the practical significance of these differences depends on the intended downstream use (indexing, citation linking, or exploratory search).

Discussion of RQ1: Comparative Accuracy Across LLMs

RQ1 asked how accurately ChatGPT, Gemini, Grok, and DeepSeek identify judicial entities in LHC judgments under strict span-level evaluation. The results show the following ordering by F1: Grok (0.6854), Gemini (0.6820), ChatGPT (0.6783), and DeepSeek (0.5790). While the top three are close, their precision and recall profiles differ in a meaningful way for deployment.

ChatGPT achieved the highest precision (0.7366) but lower recall (0.6286). This pattern is consistent with a conservative extraction style, where the model produces fewer predicted entities and therefore fewer false positives, but also misses a higher share of gold spans. Gemini and Grok showed higher recall (0.6534 and 0.6628), indicating broader coverage of entities at the cost of slightly lower precision. DeepSeek produced both lower precision and recall in this evaluation, suggesting that, under the same prompt and formatting constraints, it was less reliable both in finding entities and in labeling them accurately.

These outcomes align with a general observation in legal NER research: extraction quality depends not only on language understanding, but also on disciplined boundary control and stable label selection. Legal texts include long, irregular spans such as statute titles and citations, and strict evaluation penalizes even small deviations. Prior legal-domain studies show that models trained on general-domain NER can degrade substantially when applied to legal text, which supports the need for domain-specific evaluation rather than relying on general claims of capability. [26], [15]

What the precision and recall patterns imply in practice

The measured precision and recall differences can be interpreted as different operating points. For indexing and citation linking, false positives can be expensive because they create incorrect metadata or invalid links. Under such conditions, ChatGPT's higher precision may reduce downstream cleaning workload, even if recall is slightly lower. For exploratory search and evidence discovery tasks, recall is often more valuable because missing a citation or a key party can hide relevant judgments. In that setting, Grok and Gemini may be preferable due to higher coverage.

A practical strategy for production systems is to treat these models as complementary. For example, a higher-recall model can generate candidate entities, then a validation step (rule-based normalization for citations, or a second model pass) can confirm or reject uncertain outputs. This approach is consistent with how legal NLP toolkits often combine statistical extraction with rule-based normalization. [22], [45]

Discussion of RQ2: Difficult Entity Types and Error Patterns

RQ2 asked which entity categories are most difficult across models and what patterns explain the difficulty. While this thesis reports headline micro scores, error inspection indicates that most failures cluster around four recurring challenges: boundary drift, label confusion among closely related legal types, context loss in long judgments, and spurious predictions due to template generalization. These error families are widely reported in legal IE and prompt-based extraction literature, especially when texts are long and semi-structured. [13], [10]

Boundary drift remains a primary bottleneck under strict scoring

Boundary drift was frequently observed for multi-token party names, statute titles, and citation strings. Under strict exact-match scoring, even a single extra token (for example adding a role term such as petitioner) or missing punctuation in a citation span causes a full error. This is particularly impactful in legal NER because many entities are long and include format markers, such as reporter abbreviations and years.

The literature helps explain why boundary control is hard. In legal datasets, entity taxonomies are often fine-grained and include overlapping expressions (for example, a provision reference includes both a statute and a section). Without explicit instruction on span segmentation, models may choose plausible but non-compliant boundaries. Dataset efforts in legal NER emphasize the

importance of detailed annotation guidelines and consistent span rules, which supports the methodology used in this thesis. [4], [11]

Label confusion occurs when legal surface forms overlap

A second major error source is type confusion among closely related labels, particularly: (i) CITATION versus PRECEDENT_CASE, and (ii) ACT versus LAW_SECTION. The surface form of a citation may appear adjacent to a case name, and a provision reference may include both a statute title and section number. When the prompt does not illustrate separation rules, models may label the combined string as a single type or swap labels.

This confusion mirrors what is seen in legal-domain toolkits and benchmarks, where legal labels are often defined to support downstream linking and normalization. For instance, benchmarks such as LexGLUE highlight that legal language understanding is not a single task, and that domain-adapted evaluation must consider the precise target output required by an application. [21]

Long-context effects and chunk boundaries contribute to misses

Missed entities were often linked to long judgments and chunk boundaries. Even with overlap, entities that begin near the end of a chunk or depend on context from a prior heading can be missed. This is consistent with the view that legal extraction requires careful handling of document structure and segmentation, not only a powerful model. Tools such as LexNLP and Blackstone emphasize segmentation and normalization as key components of legal text processing, which supports the thesis decision to use deterministic preprocessing and validation. [22], [45]

Spurious predictions and the risk of hallucination

Spurious predictions remain a critical concern in judicial pipelines. In legal settings, hallucinated entities can create incorrect records, undermine trust, and mislead users. The higher precision observed for ChatGPT suggests fewer spurious entities under this prompt, but all models can still produce false positives when judgment layouts resemble common templates. In the generative IE literature, this risk is one reason researchers advocate schema-based constraints, verification steps, and stricter output interfaces rather than free-form generation. [13], [31]

Model	Observed profile in this study	When it is a good fit	Main risk to manage
ChatGPT	Highest precision, lower recall	Clean indexing, citation graphs, metadata extraction where false positives are costly	More missed entities; improve recall via chunk overlap or a second extraction pass
Gemini	Balanced, higher recall than ChatGPT	Search and analytics where coverage matters, and moderate noise is acceptable	Boundary drift on long spans; require normalization and validation
Grok	Highest F1, highest recall among top group	Broad coverage extraction for discovery and retrieval tasks	Spurious entities and label confusion; add verification and schema checks
DeepSeek	Lower precision and recall in this evaluation	Exploratory baseline comparisons or when cost constraints dominate	Higher error rates; stronger post-processing and human review needed

Table 0-6: Model selection guidance based on observed precision and recall profiles

Discussion of RQ3: Role of Prompts, Constraints, and Validation

RQ3 asked how prompt constraints and output validation influence reliability and what practices improve reproducibility. This thesis used a prompt-only approach and enforced token-level IOB tagging with a fixed label set. Two conclusions follow from the results and the observed error patterns.

First, structure matters. Prompt-based NER can be flexible, but reliability depends on how tightly the output is constrained. Methods such as PromptNER highlight that providing explicit entity definitions and a precise output format can improve extraction in few-shot settings. Similarly, ChatIE shows that multi-turn prompting can decompose extraction and improve control, although at a higher interaction cost. These ideas support the thesis emphasis on strict output interfaces and deterministic preprocessing. [9], [43]

Second, validation is not optional for legal extraction. The study's post-processing did not add entities, but it repaired invalid IOB transitions and enforced one-tag-per-token, which is necessary to score and to integrate outputs into downstream systems. The generative IE literature supports the broader principle that schema enforcement and constrained generation reduce format errors and improve usability. In practical deployments, a validation layer can also reject outputs that contain spans not present in the source text, which helps limit hallucination. [13]

Reproducibility challenges for provider models

A key reproducibility issue for commercial LLMs is model drift: providers update models and system behaviors over time. Even when the same prompt is used, performance may shift due to changes in alignment policies, training data, or decoding defaults. For this reason, a comparative thesis should report evaluation dates and model identifiers, and should preserve prompt versions and preprocessing scripts. This recommendation aligns with broader best practices in benchmarking large models, where experimental context is necessary for interpretation. [12], [1]

Positioning the Findings Within Legal NLP and Pakistani Judicial Text Mining

The findings contribute evidence for a jurisdiction that is underrepresented in public legal NLP benchmarks. LHC judgments are publicly accessible through the court's reporting portals, but they are not accompanied by standardized structured metadata suitable for large-scale analytics. [46]

Earlier work on LHC criminal judgments demonstrated that extracting structured information from Pakistani judgments is feasible and useful for information access. This thesis extends that direction by evaluating modern instruction-tuned LLMs under a strict judicial NER protocol and by focusing on cross-model comparison. The results support two broader points. First, even strong general-purpose models still face boundary and label challenges on legal text, reinforcing the continued importance of domain-specific resources and careful schema design. Second, prompt-only extraction can produce competitive performance without supervised fine-tuning, which is valuable when labeled data is limited and annotation is expensive. [5]

Implications for Deployment and System Design

The results suggest several design implications for practical legal systems in Pakistan:

A. Use-case driven model selection: choose a higher-precision model for clean metadata pipelines, and a higher-recall model for discovery workflows.
B. Build a hybrid pipeline: combine an LLM with deterministic normalization for citations and provisions (for example regex-based canonicalization and case-insensitive matching) to reduce boundary sensitivity.
C. Treat headings as high-value signals: party roles and bench information are often concentrated in the preamble; extracting the preamble separately can improve role accuracy.

D. Add verification steps: confirm that each predicted span is a contiguous substring of the source text and that each citation matches expected reporter patterns before persisting to an index.
E. Preserve auditability: store the extracted span, its character offsets, and the surrounding context to support later review and correction.

Error family	Why it matters in legal workflows	Mitigation step(s)
Boundary drift	Breaks citation linking and exact search matches	Character-offset extraction, stricter span rules in prompts, normalization rules for citations and sections
Type confusion	Mislabels can mislead filtering and analytics	Add short label-specific examples, separate ACT from LAW_SECTION, separate CITATION from PRECEDENT_CASE
Chunk boundary misses	Entities near boundaries are missed, reducing recall	Overlapping chunks, preamble-first extraction, aggregation with overlap resolution
Spurious entities	Creates incorrect metadata and harms trust	Span verification (must appear in text), conservative thresholds, second-pass confirmation
Format non-compliance	Blocks scoring and downstream ingestion	Schema validation, IOB repair, reject invalid outputs rather than guessing

Table 0-7: Observed error families and practical mitigation steps for LHC judicial NER

Threats to Validity and Study Limitations

Several limitations should be considered when interpreting the discussion.

First, the evaluation is prompt-only and does not include fine-tuning or retrieval-augmented strategies. It therefore measures a particular, practical scenario, but it does not represent the maximum possible performance of any model. Second, strict span-level scoring is demanding and may understate usefulness in applications where partial matches can be normalized. Third, the dataset is limited to LHC judgments in English; results may differ for other Pakistani courts, for Urdu judgments, or for documents with heavier OCR noise.

Fourth, provider models can change over time. Even with the same protocol, later evaluations may produce different outcomes. For transparency, the evaluation date and model identifiers should be recorded alongside the reported scores. Finally, qualitative error analysis depends on the inspected sample and may not capture all rare failure modes. It is best interpreted as a targeted explanation of common patterns observed in the evaluated outputs.

Recommendations and Future Work

This thesis motivates several future research directions.

Entity-type level benchmarking: report per-label precision, recall, and F1 to identify which of the 22 categories drive most errors.

A. Normalization-aware scoring: complement strict exact-match F1 with a second metric that accounts for normalized citations and sections, to reflect downstream use.

B. Hybrid extraction: combine a legal-domain tagger (for example a domain-adapted transformer) with an LLM that resolves ambiguous roles and cross-paragraph references, leveraging the strengths of both approaches. Legal-domain pretraining results in the literature suggest that domain adaptation can improve performance on legal tasks.
C. Multi-turn verification prompting: explore interactive pipelines similar to ChatIE to improve reliability on long documents and reduce hallucinations, while measuring added cost. [43]

Expansion to multilingual legal text: extend the schema and pipeline to handle Urdu names and mixed-language segments that appear within Pakistani judgments.

Summary

This chapter discussed why the evaluated LLMs behave differently under strict judicial NER scoring and how the observed precision and recall profiles translate into practical choices for legal systems. It also explained common error families and linked them to known challenges in legal information extraction. The next chapter concludes the thesis by summarizing contributions, answering the research questions succinctly, and outlining an actionable roadmap for future improvements.

Conclusion And Future Directions

Judicial entity schema of 22 types aligned with the information needs of LHC judgments and prepared a gold-standard dataset using token-level IOB tagging. A prompt-and-parse pipeline was designed to enforce a consistent output interface across different model providers. All models were evaluated on the same texts under the same prompts, chunking strategy, validation rules, and strict span-level exact match scoring. Results were reported with micro-averaged precision, recall, and F1, and were complemented with qualitative error analysis that summarized recurring boundary errors, label confusions, missing entities, and spurious predictions.

The central comparative finding is that Grok achieved the highest overall micro F1 (0.6854), followed closely by Gemini (0.6820) and ChatGPT (0.6783), while DeepSeek performed lower (0.5790) under the same protocol. The top three models were close in F1 but differed in precision and recall behavior. ChatGPT achieved the highest precision (0.7366), while Grok and Gemini provided higher recall (0.6628 and 0.6534). These differences translate into different deployment choices depending on whether the system prioritizes clean indexing with minimal noise or prioritizes high coverage for discovery and retrieval.

Theoretical Contributions

First, it reinforces that judicial NER should be treated as a jurisdiction-specific problem. Entity taxonomies, formatting conventions, and citation patterns vary by legal system, so performance should be measured on local data rather than assumed transfer from foreign benchmarks.

Second, it contributes to the understanding that prompt-based NER is partly an interface design problem. In legal settings, boundary rules and label definitions embedded in prompts and guidelines strongly influence extraction quality because strict scoring and downstream systems depend on consistent spans and stable label selection.

Third, it provides comparative evidence across four modern LLM families under a controlled protocol, clarifying how different LLM services behave on long legal documents when required to generate structured outputs.

Practical Contributions

This thesis provides practical contributions that can support legal technology work in Pakistan.

A. A 22-type judicial entity schema with span rules aligned to LHC judgment structure. A gold-standard annotated dataset in token-level IOB format for repeatable evaluation.
B. A prompt-and-parse protocol that enables fair comparison across different LLM providers without fine-tuning.

C. Deployment guidance based on precision and recall profiles, including recommendations for chunking and output validation.
D. An error taxonomy that can guide quality assurance, targeted mitigation, and regression testing.

Contribution	What it provides	Value for research and deployment
Entity schema and span rules	22-type schema aligned with LHC judgments	Defines consistent extraction targets and evaluation scope
Gold annotation in IOB	Token-level labeled judicial text	Enables repeatable benchmarking and later supervised training
Prompt-and-parse protocol	Standard prompt plus validation rules	Supports cross-model comparison and robust ingestion
Error taxonomy	Structured categories of failures	Guides targeted mitigation and quality assurance
Model trade-off insight	Precision and recall profiles	Informs model selection for indexing versus discovery tasks

Table 0-8: Summary of contributions and their practical value

Limitations

First, the evaluation is prompt-only and does not include fine-tuning, retrieval-augmented prompting, or domain-adaptive pretraining. Therefore, results reflect a practical baseline rather than the maximum achievable performance.

Second, strict span-level exact match scoring is intentionally demanding. It can understate usefulness in settings where citations and provisions are normalized downstream or where partial matches can still support search. A second normalized scoring view could better represent application utility.

Third, the dataset focuses on English-language LHC judgments. Results may differ for other Pakistani courts, other legal document types, and Urdu or mixed-language judgments. The schema may also need extension for national coverage.

Fourth, provider models can change over time, which can affect reproducibility. Logging model identifiers, evaluation dates, prompt versions, and preprocessing rules reduces this risk but cannot remove it entirely.

Fifth, qualitative error analysis depends on inspected samples and may not capture rare failure modes. A larger dataset and per-entity performance reporting would strengthen conclusions about the most difficult labels.

Future Directions

Methodological improvements

Future work can improve methodology in several ways:

A. Per-entity evaluation: report precision, recall, and F1 per label and group labels into difficulty bands.
B. Normalization-aware metrics: complement strict F1 with a metric that canonicalizes citations, case numbers, and section references before matching.

C. Hybrid extraction: combine rule-based recognizers for highly structured patterns (case numbers, dates, sections) with LLM prompting for role-sensitive and cross-paragraph entities.
D. Verification and self-consistency: use multi-pass extraction where the model proposes candidates and then verifies each candidate against text evidence, rejecting unsupported spans.
E. Long-document strategies: test heading-aware segmentation, preamble-first extraction, and overlap rules that reduce boundary loss at chunk edges.

Data and resource expansion

Future work can expand resources for Pakistani legal NLP:

A. Broader court coverage: include other Pakistani courts and tribunals to test generalization across drafting styles.
B. Multilingual legal NER: extend the pipeline to Urdu and mixed-language segments, focusing on transliteration, spelling variation, and localized legal terminology.
C. Annotation scaling: add inter-annotator agreement on a shared subset, refine guidelines from disagreements, and increase dataset size to enable supervised baselines.

Practical deployment directions

Several applied directions can translate these findings into usable systems:

A. Search enhancement: expose extracted entities as structured filters (judge, act, section, case number) for faster legal retrieval.
B. Citation linking and networks: build citation graphs from extracted precedent cases and citations to support navigation and analytics.
C. Workflow integration: embed extraction into document intake with an audit trail that stores extracted spans and character offsets for review.
D. Risk controls: treat LLM outputs as suggestions and verify before committing to official records, especially in high-stakes workflows.

Closing Remarks

This thesis provided a controlled, dataset-driven comparison of four modern LLM families for judicial NER on Lahore High Court judgments. It demonstrated that prompt-only extraction can deliver usable performance, while also revealing recurring legal-specific errors that must be managed for reliable deployment. The results emphasize that successful legal extraction requires careful schema design, strict structured outputs, and robust validation.

With expanded datasets, normalization-aware evaluation, and hybrid pipelines that combine deterministic legal pattern matching with LLM flexibility, judicial NER for Pakistani case law can become a reliable component of legal information systems. This can support faster legal research, improved retrieval, and more transparent analysis of precedent and statutory usage.

References

[1] Lahore High Court, "Judgments Approved for Reporting (Reported Judgments Portal)," https://data.lhc.gov.pk/reported_judgments/judgments_approved_for_reporting.
[2] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The Muppets straight out of Law School," Findings of EMNLP, 2020.
[3] I. Chalkidis et al., "LexGLUE: A Benchmark Dataset for Legal Language Understanding in English," ACL, 2022.
[4] P. Kalamkar et al., "Named Entity Recognition in Indian court judgments," Natural Legal Language Processing Workshop, 2022.

[5] A. Iftikhar, S. W. U. Q. Jaffry, and M. K. Malik, "Information Mining From Criminal Judgments of Lahore High Court," IEEE Access, 2019, doi:10.1109/ACCESS.2019.2915352.
[6] S. Sharafat, Z. Nasar, and S. W. Jaffry, "Data mining for smart legal systems," Computers and Electrical Engineering, 2019, doi:10.1016/j.compeleceng.2019.07.017.
[7] M. K. Malik, "Urdu Named Entity Recognition and Classification System Using Artificial Neural Network," ACM Transactions on Asian and Low-Resource Language Information Processing, 2017. https://dl.acm.org/doi/abs/10.1145/3129290
[8] D. Xu et al., "Large language models for generative information extraction: a survey," Frontiers of Computer Science, 2024. https://link.springer.com/article/10.1007/s11704-024-40555-y
[9] D. Ashok and Z. C. Lipton, "PromptNER: Prompting for Named Entity Recognition," arXiv:2305.15444, 2023. https://arxiv.org/abs/2305.15444
[10] OpenAI, "GPT-4 Technical Report," arXiv:2303.08774, 2023. https://arxiv.org/abs/2303.08774
[11] Gemini Team, "Gemini: A Family of Highly Capable Multimodal Models," arXiv:2312.11805, 2023. https://arxiv.org/abs/2312.11805
[12] xAI, "Open Release of Grok-1," 2024. https://x.ai/news/grok-os
[13] DeepSeek-AI, "DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model," arXiv:2405.04434, 2024. https://arxiv.org/abs/2405.04434
[14] N. Chinchor and P. Robinson, "Appendix E: MUC-7 Named Entity Task Definition (version 3.5)," in Seventh Message Understanding Conference (MUC-7), 1998. Available: https://aclanthology.org/M98-1028/
[15] E. F. Tjong Kim Sang and F. De Meulder, "Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition," in CoNLL 2003, 2003. Available: https://aclanthology.org/W03-0419/
[16] J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," in ICML, 2001. Available: https://repository.upenn.edu/cgi/viewcontent.cgi?article=1162&context=cis_papers
[17] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF Models for Sequence Tagging," arXiv:1508.01991, 2015. Available: https://arxiv.org/abs/1508.01991
[18] A. Vaswani et al., "Attention Is All You Need," in NeurIPS, 2017. Available: https://arxiv.org/abs/1706.03762
[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv:1810.04805, 2018. Available: https://arxiv.org/abs/1810.04805
[20] I. Chalkidis et al., "LEGAL-BERT: The Muppets straight out of Law School," in Findings of EMNLP, 2020. Available: https://arxiv.org/abs/2010.02559
[21] I. Chalkidis et al., "LexGLUE: A Benchmark Dataset for Legal Language Understanding in English," in ACL, 2022. Available: https://arxiv.org/abs/2110.00976
[22] M. J. Bommarito II, D. M. Katz, and E. M. Detterman, "LexNLP: Natural Language Processing and Information Extraction for Legal and Regulatory Texts," arXiv:1806.03688, 2018. Available: https://arxiv.org/abs/1806.03688
[23] ICLR Research, "Blackstone: A spaCy pipeline for legal NLP," 2019. Available: https://research.iclr.co.uk/blackstone
[24] D. Hendrycks, C. Burns, A. Chen, and S. Ball, "CUAD: An Expert-Annotated NLP Dataset for Legal Contract Review," arXiv:2103.06268, 2021. Available: https://arxiv.org/abs/2103.06268

[25] D. Tuggener et al., "LEDGAR: A Large-Scale Multi-label Corpus for Text Classification of Legal Provisions in Contracts," in LREC, 2020. Available: https://aclanthology.org/2020.lrec-1.155/
[26] T. W. T. Au, I. J. Cox, and V. Lampos, "E-NER: An Annotated Named Entity Recognition Corpus of Legal Text," arXiv:2212.09306, 2022. Available: https://arxiv.org/abs/2212.09306
[27] E. Leitner, G. Rehm, and J. Moreno-Schneider, "A Dataset of German Legal Documents for Named Entity Recognition," in LREC, 2020. Available: https://aclanthology.org/2020.lrec-1.551/
[28] R. Kalamkar et al., "InLegalNER: A Named Entity Recognition Dataset for Legal Domain," arXiv:2203.04253, 2022. Available: https://arxiv.org/abs/2203.04253
[29] J. Wei, S. Lin, and Z. Chen, "ChatIE: Zero-Shot Information Extraction via Chatting with ChatGPT," arXiv:2305.09551, 2023. Available: https://arxiv.org/abs/2305.09551
[30] J. Wang et al., "InstructUIE: Multi-task Instruction Tuning for Unified Information Extraction," arXiv:2304.08085, 2023. Available: https://arxiv.org/abs/2304.08085
[31] Y. Xu et al., "Generative Information Extraction with Large Language Models: A Survey," arXiv:2312.17617, 2023. Available: https://arxiv.org/abs/2312.17617
[32] S. Geng, M. Josifoski, M. Peyrard, and R. West, "Grammar-Constrained Decoding for Structured NLP Tasks without Finetuning," in EMNLP, 2023. Available: https://dlab.epfl.ch/people/west/pub/Geng-Josifoski-Peyrard-West_EMNLP-23.pdf
[33] T. Minaee et al., "Large Language Models: A Survey," arXiv:2402.06196, 2024. Available: https://arxiv.org/abs/2402.06196
[34] T. B. Brown et al., "Language Models are Few-Shot Learners," in NeurIPS, 2020. Available: https://arxiv.org/abs/2005.14165
[35] xAI, "Open-sourcing Grok-1," 2024. Available: https://x.ai/blog/grok-os
[36] P. Premasiri et al., "Legal Information Extraction: A Survey of Tasks, Datasets, Models, and Challenges," ACM Computing Surveys, 2025. Available: https://dl.acm.org/doi/10.1145/3777009
[37] D. M. Katz et al., "Natural Language Processing in the Legal Domain," arXiv:2302.12039, 2023. Available: https://arxiv.org/abs/2302.12039
[38] M. A. Iftikhar et al., "Information Mining from Criminal Judgments of Lahore High Court," IEEE Access, 2019. Available: https://ieeexplore.ieee.org/document/8891597
[39] S. Sharafat, Z. Nasar, and S. W. U. Q. Jaffry, "Data Mining for Smart Legal Systems," in 2019 22nd International Multitopic Conference (INMIC), 2019. Available: https://ieeexplore.ieee.org/document/9022770
[40] Lahore High Court, "Judgments Approved for Reporting," accessed 2026. Available: https://lhc.gov.pk/judgments-approved-for-reporting
[41] A. Daud et al., "A Named Entity Dataset for Urdu NER Task," in Conference on Language and Technology (CLT), 2016. Available: https://aclanthology.org/W16-3804/
[42] A. Anam et al., "Urdu Named Entity Recognition: A Deep Learning Approach," PLOS ONE, 2024. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0295988
[43] X. Wei et al., "ChatIE: Zero-Shot Information Extraction via Chatting with ChatGPT," arXiv:2302.10205, 2023. Available: https://arxiv.org/abs/2302.10205
[44] S. Geng et al., "Grammar-Constrained Decoding for Structured NLP Tasks without Finetuning," EMNLP, 2023. Available: https://aclanthology.org/2023.emnlp-main.674/
[45] ICLR&D, "Blackstone: A spaCy pipeline for legal NLP," project repository, 2019. Available: https://github.com/ICLRandD/Blackstone
[46] Lahore High Court, "Reported Judgments," available: https://www.lhc.gov.pk/reported_judgments (accessed January 10, 2026).