Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.062340

ARTICLE

A Hybrid Framework Combining Rule-Based and Deep Learning Approaches for Data-Driven Verdict Recommendations

Muhammad Hameed Siddiqi[1,*], Menwa Alshammeri[1], Jawad Khan[2,*], Muhammad Faheem Khan[3], Asfandyar Khan[4], Madallah Alruwaili[1], Yousef Alhwaiti[1], Saad Alanazi[1] and Irshad Ahmad[5]

[1]College of Computer and Information Sciences, Jouf University, Sakaka, 2014, Aljouf, Saudi Arabia
[2]School of Computing, Gachon University, Seongnam, 13120, Republic of Korea
[3]Institute of Computer Science & IT, University of Science and Technology, Bannu, 28100, KPK, Pakistan
[4]Institute of Computer Sciences and Information Technology, University of Agriculture, Peshawar, 25130, KPK, Pakistan
[5]Department of Computer Science, Islamia College Peshawar, 25000, KPK, Pakistan

*Corresponding Authors: Muhammad Hameed Siddiqi. Email: mhsiddiqi@ju.edu.sa; Jawad Khan. Email: jkhanbk1@gachon.ac.kr

Received: 16 December 2024; Accepted: 28 March 2025; Published: 19 May 2025

ABSTRACT: As legal cases grow in complexity and volume worldwide, integrating machine learning and artificial intelligence into judicial systems has become a pivotal research focus. This study introduces a comprehensive framework for verdict recommendation that synergizes rule-based methods with deep learning techniques specifically tailored to the legal domain. The proposed framework comprises three core modules: legal feature extraction, semantic similarity assessment, and verdict recommendation. For legal feature extraction, a rule-based approach leverages Black's Law Dictionary and WordNet Synsets to construct feature vectors from judicial texts. Semantic similarity between cases is evaluated using a hybrid method that combines rule-based logic with an LSTM model, analyzing the feature vectors of query cases against a legal knowledge base. Verdicts are then recommended through a rule-based retrieval system, enhanced by predefined legal statutes and regulations. By merging rule-based methodologies with deep learning, this framework addresses the interpretability challenges often associated with contemporary AI models, thereby enhancing both transparency and generalizability across diverse legal contexts. The system was rigorously tested using a legal corpus of 43,000 case laws across six categories: Criminal, Revenue, Service, Corporate, Constitutional, and Civil law, ensuring its adaptability across a wide range of judicial scenarios. Performance evaluation showed that the feature extraction module achieved an average accuracy of 91.6% with an F-Score of 95%. The semantic similarity module, tested using Manhattan, Euclidean, and Cosine distance metrics, achieved 88% accuracy and a 93% F-Score for short queries (Manhattan), 89% accuracy and a 93.7% F-Score for medium-length queries (Euclidean), and 87% accuracy with a 92.5% F-Score for longer queries (Cosine). The verdict recommendation module outperformed existing methods, achieving 90% accuracy and a 93.75% F-Score. This study highlights the potential of hybrid AI frameworks to improve judicial decision-making and streamline legal processes, offering a robust, interpretable, and adaptable solution for the evolving demands of modern legal systems.

KEYWORDS: Verdict recommendation; legal knowledge base; judicial text; case laws; semantic similarity; legal domain features; rule-based; deep learning

1 Introduction

The current research on integrating machine learning (ML) and artificial intelligence (AI) into the judicial system, particularly in verdict recommendation frameworks, is advancing rapidly, with various

models being developed to assist legal professionals in decision-making. These systems utilize natural language processing (NLP) to analyze legal documents, case law, and statutes, and they aim to predict case outcomes or recommend verdicts based on historical data. Countries like China and India are experimenting with AI-driven systems for judicial support. However, several challenges remain. One primary issue is the inherent bias in training data, where models may reflect and perpetuate historical inequities present in past rulings. Moreover, the interpretability of these models is critical—legal practitioners need transparency to understand and trust AI recommendations. Legal systems are also grappling with the ethical implications of automated decision-making, particularly regarding accountability. Furthermore, adapting AI models to different legal contexts is a challenge due to jurisdictional variations in laws and judicial practices. Thus, while the potential of AI in legal verdicts is promising, the need for improved data quality, model transparency, and ethical frameworks remains a significant hurdle.

Recent advancements in legal AI focus on enhancing judgment prediction and legal text analysis. CLSum, a multi-jurisdictional corpus utilizing LLMs and a legal knowledge-based assessment metric, improves summarization accuracy in low-resource settings but lacks cross-framework adaptability [1]. Similarly, BERT-CNN models for legal multiple-choice QA combine graph-based retrieval and CNN-based aggregation, improving on traditional methods yet struggling with jurisdictional inconsistencies in legal terminology [2]. DiscoLQA, a discourse-aware QA model, employs Elementary Discourse Units (EDUs) and Abstract Meaning Representations (AMRs) for legal text comprehension, achieving state-of-the-art zero-shot accuracy, though its effectiveness declines with unstructured case law [3].

A legal expert system refers to computer software capable of simulating the specific functions performed by judges or lawyers. It can aptly be termed a "Computerized Legal Advisor" due to its ability to 'reason' and 'think' like human legal professionals [4–6]. Developed legal expert systems can be classified into five types: (i) systems for consultancy on legal matters, which legal practitioners use to simulate reasoning processes; (ii) legal strategy systems used to assess the chances of success, provide legal advice, and offer information on similar previously decided cases; (iii) automatic document generators; (iv) intelligent systems based on the organizational skills of a lawyer; and (v) systems for computer-aided learning in the judicial domain [6].

Within Pakistan's judiciary, local courts grapple with a substantial caseload. For instance, the Law and Justice Commission of Pakistan released statistics on 15 June 2020, revealing 45,125 pending cases at the Supreme Court level, 332,376 at the High Court level, and 1,663,528 at the lower court level. To manage the increasing number of cases and legal documents, steps have been taken at all three levels to digitize records and archive them. However, a system that can automatically extract relevant information from similar legal documents and recommend verdicts would tremendously benefit Pakistan's legal community, including judges and lawyers. As in other fields, decision-support systems in the legal domain have garnered significant interest [7–11]. In every country, the justice system impacts society as a whole as well as individual citizens. In Pakistan's judiciary, the increasing number of pending cases makes it very complex to predict a judge's decision. To assist judges in decision-making, providing necessary recommendations and relevant case law for their judgments is a highly effective solution [12].

Similarly, finding relevant case laws and other legal resources is very difficult for people without a legal background. Extracting specialized legal terms from complex cases cannot be done simply by using keywords, as it requires a legal background [13]. Even for both parties—plaintiffs and defendants—understanding their rights protected by statutes is challenging without the help of legal counsel. Furthermore, a novel Baseline Data-based Verifiable Trust Evaluation (BD-VTE) scheme was designed by [14] to ensure security while maintaining cost efficiency. The BD-VTE framework comprises three key components: The Verifiable Trust Evaluation (VTE) mechanism, the Effectiveness-based Incentive (EI) mechanism, and the Secondary Path Planning (SPP) strategy. These components facilitate reliable trust assessment, fair reward

distribution, and optimized path adjustments, respectively. Notably, the VTE mechanism incorporates an innovative active trust verification approach, which enhances trust evaluation by deploying UAVs to collect baseline data from IoT devices, enabling a more accurate assessment of mobile vehicles (MVs).

1.1 Problem Statement

Due to a huge caseload, numerous cases are fixed in the judge's cause list. Hearing of all these cases requires momentous effort and time. To decide a case, the judge requires relevant case law on the subject matter. In the judiciary of Pakistan, research centers have been established where legal experts are employed to assist the judges in searching the relevant legal resources and case laws. The case laws and other legal resources are available in heterogeneous formats from different sources. To prepare a memo on a subject query, the legal expert extracts the relevant information by manually compiling the relevant resources. Similarly, an advocate requires a lot of research while drafting a case.

1.2 Research Objectives

The proposed framework addressed the following objectives to efficiently recommend the verdict.

- **Enhanced Feature Selection:** To improve the feature selection process by extracting legal domain features using *Black's Law Dictionary* and *WordNet Synsets*, ensuring a comprehensive representation of legal terms and their contexts.
- **Enhanced Feature Selection:** To improve the feature selection process by extracting legal domain features using *Black's Law Dictionary* and *WordNet Synsets*, ensuring a comprehensive representation of legal terms and their contexts.
- **Performance Evaluation and Comparison:** To evaluate the effectiveness of the proposed LSTM + Rule-based framework by comparing its performance against alternative machine learning and deep learning models, highlighting its superiority in recommending verdicts.

1.3 Contributions

The proposed framework makes significant contributions in terms of semantic similarity determination and verdict recommendation to support the legal community. The key contributions of this work include:

- **Innovative Legal Feature Extraction:** Unlike prior methods, we introduce a dual-layered feature extraction approach with *Black's Law Dictionary* for precise legal definitions and *WordNet Synsets* for contextual enrichment, which enhances the accuracy of semantic representations in judicial cases.
- **Adaptive Similarity Function Selection for Case Retrieval:** We propose a dynamic LSTM-based similarity analysis model that automatically selects the most effective similarity function (Manhattan, Euclidean, or Cosine) based on query size, optimizing case retrieval accuracy across diverse legal queries.
- **Large Scale Legal Knowledge Base for Verdict Recommendation:** Our framework incorporates an annotated database of 43 K judicial decisions, making it one of the most comprehensive legal AI systems. This structured knowledge base significantly improves verdict prediction reliability.
- **Comprehensive Evaluation and Superior Performance:** The framework was thoroughly tested on 600 legal cases across six categories (*criminal, civil, service, corporate, revenue, constitutional*), achieving high accuracy and outperforming existing models in semantic similarity detection and verdict recommendation.

The subsequent organization of the paper is outlined as follows: Section 2 delves into the related research on artificial intelligence, machine learning, deep learning, and rule-based approaches within legal recommendation, judgment prediction, and decision support systems. Meanwhile, Section 3 provides an

overview of the methodologies, with Section 3.1 detailing the legal knowledge and Section 3.2 describing the acquisition of legal domain features. Section 3.3 introduces the model for determining semantic similarity, while Section 3.4 elaborates on the verdict recommendation model. The outcomes of the proposed framework are detailed in Section 4. Ultimately, Section 5 concludes the paper, providing concise findings and results, and outlining future work.

2 Related Work

In the earliest work, the mathematical method was used for predicting the Supreme Court decision back in 1957 [15]. During the early stages of artificial intelligence, the creation of legal expert systems commenced in the early 1980s [16]. Table 1 presents several machine learning (ML), deep learning (DL), and rule-based (R) methods that are used in the legal domain, which are related to judgment prediction, verdict recommendation, and decision support systems from the latest literature.

The legal case document is usually complex, as it has multiple references to case law and other statutes, which take more time to read and understand. Judges need to refer to all the relevant case law before the announcement of judgment. Several judgment prediction methods have been proposed to assist the judges, including [10], in which they use machine learning methods to predict the US Supreme Court decision. Similarly, a few other authors also used machine learning methods for judgment prediction, including [17] for predicting court verdicts in criminal cases. For multi-label classification of US legal provisions in contracts [18], and ref. [19] predict the judgment in legal cases. Several other authors, including [20–22], use deep learning methods for legal judgment prediction, and refs. [23,24] use them for the prediction of court decisions.

A knowledge base is crucial for judgment prediction and decision support systems, though extracting relevant cases is time-intensive. Past studies employed various approaches, including machine learning for Brazil's superior court [25] and Canadian legislation [26], deep learning for legal risk management [27], and hybrid methods for Japanese civil law [28] and legal judgment prediction [29–31]. Rule-based systems follow specific laws, such as an inheritance rights system based on Islamic rules [32], while combined rule-based and machine-learning models address Arabic legal documents [33]. Deep learning models have been applied to Portuguese [34], German [35], and Chinese legal texts [36], and hybrid methods have handled human rights cases [37]. Recently, a deep learning model was adapted for Saudi Arabia's judicial system [38]. The AI-driven legal analysis developed decision-support frameworks, utilizing deep and machine learning for the prediction of judgments [39], retrieval of case laws [40], and suggesting verdicts [41]. Legal judgment summarization is improved by [1], who present CLSum, a multi-jurisdictional corpus using large language models with a legal knowledge-based assessment measure to support accurate summarization in low-resource conditions. Yet, its portability across jurisdiction-level legal interpretations is challenging. Likewise, ref. [2] developed a BERT-CNN model combination for legal multiple-choice Question Answering (QA), combining graph-based retrieval with CNN-based aggregation, with higher accuracy than traditional models but with difficulties with cross-jurisdiction discrepancies in terminology. Additionally, ref. [3] proposed DiscoLQA, a discourse-based QA system using Elementary Discourse Units (EDUs) and Abstract Meaning Representations (AMRs) for enhanced legal text understanding. Although with state-of-the-art zero-shot accuracy, its utility is impaired when applied to case law with no structure.

In the above section, we reviewed various AI and machine-learning approaches in the legal domain. Table 1 shows that AI applications are gaining traction, with models used for legal document classification, case outcome prediction, and text summarization. However, technical limitations hinder the broad application and scalability of these systems. A key challenge is the jurisdiction-specific nature

of legal systems—AI models trained in one legal context often fail to generalize due to differences in terminology, statutes, and procedures. Each jurisdiction's unique laws and precedents require tailored models that can accurately interpret local legal principles. Additionally, many AI systems lack interpretability and transparency, especially deep learning models functioning as "black boxes", which is problematic in legal settings where explainability is essential. Another limitation is the absence of comprehensive legal knowledge bases, critical for effective legal reasoning and case retrieval. This is particularly evident in Pakistan, where no centralized legal database exists, and fragmented efforts complicate AI system development. Building such knowledge bases is challenging due to the unstructured, inconsistent formats of legal documents across courts, with variations in style, terminology, and structure. Historical legal records often require manual digitization and annotation. Even in digital formats, extracting meaningful features is complicated by complex legal language involving intricate reasoning, multiple statutes, and precedent citations. These challenges highlight the need for advanced NLP techniques, domain-specific legal ontologies, and standardized legal data across jurisdictions, especially in evolving legal systems like Pakistan.

Table 1 highlights that early research focused on machine learning and rule-based methods. Recently, deep learning has shown strong performance in the legal domain. Our framework combines a rule-based model with deep learning for semantic similarity and verdict recommendations. Using an enhanced LSTM with an extra hidden layer, rules based on query size improve legal domain results. The framework emphasizes building a legal knowledge base, extracting domain features, enabling semantic search, and recommending decisions based on the knowledge base.

Table 1: Related works in machine learning, deep learning, and rule-based methods in the legal domain

Year	Domain	Model
2017	Predicting decisions (U.S. Supreme Court) [11]	ML
2017	Forecast decisions (French Supreme Court) [17]	ML
2017	Legal question—answering in Civil law (Japanese) [13]	ML + DL
2018	Prediction of court verdicts in criminal cases [14]	ML
2018	Legal dataset for judgment prediction [15]	ML + DL
2018	Legal risk management (Germany) [27]	DL
2019	extraction of similar legal cases (Indian Supreme Court) [17]	ML
2019	Legal judgments (Indian) [20]	ML + DL
2019	Case outcome detection [20]	ML + DL
2019	Family Law Legal Guidance System [32]	R
2019	Legal judgment prediction [29]	DL
2020	Responding to Legal Questions on Legislation in British Columbia, Canada [26]	ML
2020	Retrieving Comparable Legal Cases in the Brazilian Superior Court [25]	ML
2020	US legal provisions in contracts [18]	ML + DL
2020	Arabic legal expert system (Saudi Arabia) [33]	R + ML
2020	Legal expert system (Portuguese) [34]	DL
2020	Legal text recognition (German) [35]	DL
2020	Judgment document preparation (Chinese) [36]	DL
2021	Legal expert system for Inheritance rights distribution [32]	R
2021	Judgment prediction of legal cases [23]	DL
2021	Legal judgment prediction system [22]	ML

(Continued)

Table 1 (continued)

Year	Domain	Model
2021	Legal judgment prediction system [24]	DL
2021	Anticipating Cases of Human Rights Violations in the EU Court [37]	ML + DL
2022	Legal prediction and decision support [34]	ML
2022	Prediction of court decision service rate [24]	DL
2022	Legal text recognition [37]	DL
2023	Judicial Decision Support System (Saudi Arabia) [38]	DL
2024	Legal judgment summarization [1]	DL
2024	Evidence retrieval for legal QA [2]	ML + DL
2024	Zero-shot discourse-based legal QA [3]	DL

3 Methodology

The proposed framework comprises the following three main modules: (i) acquiring legal domain features, (ii) semantic similarity determination, and (iii) verdict recommendation. For a legal expert system, the knowledge base is an essential element. We developed our legal knowledge base.

3.1 Legal Knowledge Base

Being an essential part of the expert system, we develop our legal knowledge base. The legal knowledge bases comprise past decided cases known as case law. In the initial version, we collected 36,809 reported judgments publicly available on the official website of the higher courts of Pakistan. Due to the unstructured, non-unified format and huge domain, we limit our legal knowledge base only to the following six categories of high court level: (i) criminal, (ii) civil, (iii) service, (iv) corporate, (v) revenue, and (vi) constitutional. The annotation process involved expert curation of legal reasoning, categorization of judicial outcomes, and assignment of metadata for structured retrieval. We annotated the judgment of all these categories along with additional parameters. While the creation of a comprehensive legal knowledge base comprising 43,000 annotated judicial decisions is a notable achievement, the paper provides limited detail on the annotation process for critical attributes like legal reasoning, which is essential for nuanced verdict recommendations. The success of AI-driven legal frameworks depends not only on the volume of data but also on the depth and quality of annotations, particularly in capturing the rationale behind judicial decisions, contextual nuances, and statutory interpretations. Without a thorough explanation of how reasoning attributes were identified, categorized, and integrated into the system, it is difficult to assess the robustness and interpretability of the verdict recommendations fully. Detailed methodologies for annotating complex legal concepts, such as the distinction between procedural and substantive reasoning or the handling of conflicting precedents, would enhance the framework's credibility and offer insights into its potential adaptability across diverse legal domains. Addressing this gap could also provide a clearer roadmap for replicating or expanding the system in other jurisdictions with differing legal reasoning structures. Table 2 illustrates the number of judgments processed against each category.

Table 2: Number of judgments against each category in the legal knowledge base

S. No.	Judgment category	No. of judgments
1	Criminal	10,440
2	Civil	9178
3	Service	2225
4	Corporate	1818
5	Revenue	7816
6	Constitutional	5332
	Total	36,809

We compiled and annotated the judgment by defining six attributes. Each judgment is assigned a unique ID in the knowledge base. The other parameters include the category of the judgment, year of the judgment, reference to the original court, keywords, full text, and reasoning. Fig. 1 illustrates the judgment attributes.

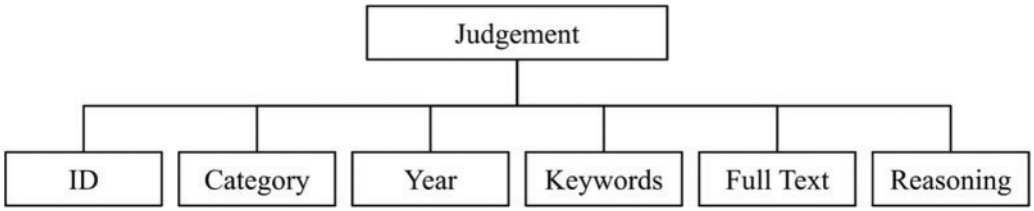

Figure 1: Judgment attributes

The full text of the judgment is pre-processed. The preprocessing procedures encompass (i) tokenization, (ii) cleaning of text, (iii) substitution of short abbreviated words, (iv) correction of spelling, and (v) part-of-speech tagging. In text cleaning, we perform lemmatization, removing stop words, and removing irrelevant characters. The judgment contains many short abbreviations. See Table 3 for sample short abbreviations. To extract the semantic and contextual information, it is required to replace the short-abbreviated word with the full English word.

Table 3: Exemplary legal abbreviations

Abbreviations	Word/Phrase	Abbreviations	Word/Phrase
AWC	Appeal within the court	FIR	First investigation report
LMR	Legal medical report	WR	Witness for the respondent
AR	Autopsy report	IO	Investigation officer
CJ	Confinement in jail	ARNS	Act for the regulation of narcotic substances
HI	Harsh incarceration	NRO	National reconciliation ordinance
PB	Punjab	PCO	Provisional constitution order
PESH	Peshawar	AAG	Additional attorney general
WP	Witness for the prosecution	DW	Defense witness

We collected and compiled the short-abbreviated words by mentioning the original word/phrase of the legal domain, which are shown in grey color in Fig. 2. In the preprocessing phase, it is substituted with the original word, and subsequently, part-of-speech tagging is conducted using the Stanford POS Tagger. Fig. 2 illustrates the pre-processing steps applied to the full-text attribute of the judgment.

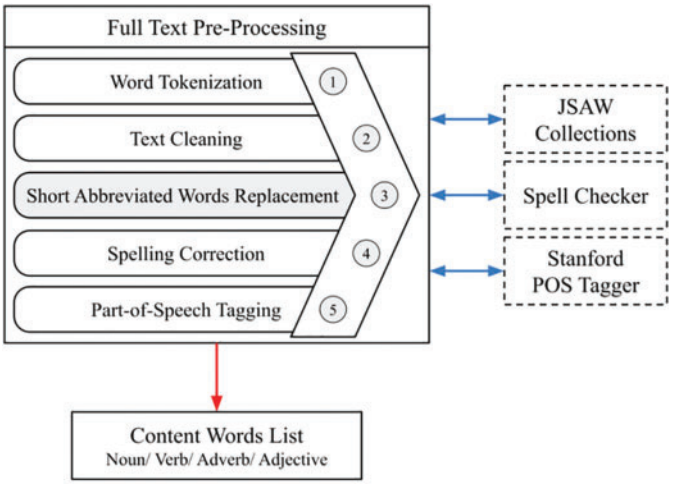

Figure 2: Pre-processing steps applied to full text

We filter only the following content words from the tagged text: (i) Noun, (ii) Verb, (iii) Adverb, and (iv) Adjective, and skip all others. The full-text attribute contains the annotated content words, which keep the vector size smaller and meaningful. Table 4 illustrates the sample published judgment in the legal knowledge base along with attributes taken from the official website of the Peshawar High Court, KP, Pakistan.

Table 4: Judgement along with attributes

ID	Category	Year	Keywords	Full text	Reasoning
1	Criminal	2023	324/353/427 acquittal	"The prosecution failed to prove the charge against the appellant beyond a reasonable doubt, leaving room for possible false implication. Thus, the appellant deserves the benefit of the doubt. Legally, any reasonable doubt, even from a single circumstance, entitles the accused to this right, not as leniency but as a legal entitlement."	"The incident occurred at night with no light source noted in the site plan. The complainant relied solely on voice recognition, a weak form of evidence, making it insufficient for conviction. Thus, the accused was acquitted with the benefit of the doubt."

We filter only the following content words from the tagged text: (i) Noun, (ii) Verb, (iii) Adverb, and (iv) Adjective, and skip all others. The full-text attribute contains the annotated content words, which keep the vector size smaller and meaningful. Table 4 illustrates the sample published judgment in the legal knowledge

base along with attributes taken from the official website of the Peshawar High Court, KP, Pakistan. The judgments are compiled in the knowledge base in the above format. The semantic similarity determination module analyzes the input query from the legal domain vector and the vector from the legal knowledge base.

3.2 Learning Word Embeddings

Word embedding is a method that represents words as vectors. By using this technique, we can get a similar representation of those words that have similar meanings. In this work, we used Mikolov's [42] Doc2Vec embedding model, which is an efficient word embedding model in terms of space and time. Doc2Vec is an extension of Word2Vec, with the key difference being that while Word2Vec predicts individual words, Doc2Vec uses different features to represent and generate the entire document as a unified vector.

In this research, the Doc2Vec model is utilized as the embedding layer for the LSTM network in detecting similarities, with the model being trained on the legal knowledge base using the Gensim library. The statistical details of the training data within the legal knowledge corpus are presented in Table 3. Given that this model depends on distributional similarity, its dimensions can vary for different applications, typically falling within the range of 50 to 300.

3.3 Acquiring Legal Domain Features

This research aimed to create a decision support system for the judiciary. Many terms used in a legal domain have different meanings and contexts than in the general domain. For the impartiality of the system, it must be designed based on the domain requirements. For this purpose, to get legal domain features from the judicial text by integrating the legal dictionary (Black Law) with the WordNet. To acquire the legal domain features, we designed a rule-based model that applies rules and extracts the most relevant features of the legal domain. The rules search for a feature by utilizing the WordNet Synset and Black Law definition. The concept diagram is presented in Fig. 3.

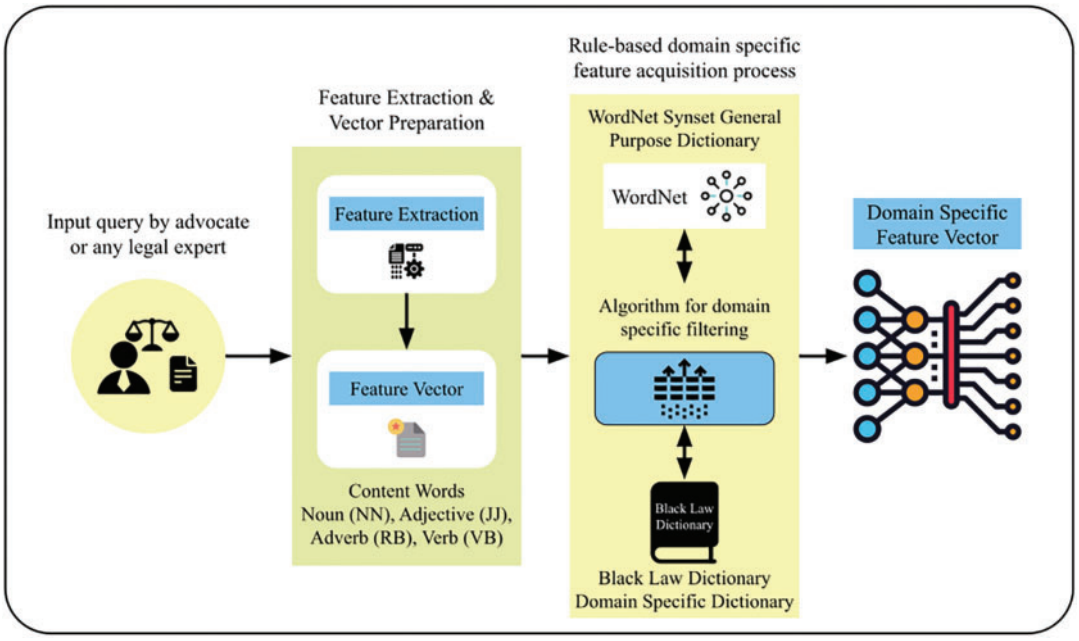

Figure 3: Legal domain features

The process of acquisition follows various rules. In this process, each feature is searched in the WordNet Words List and Black Law definition, and if not found in WordNet and found in Black Law, then the feature is selected from Black Law because it is a legal domain dictionary. To improve scalability, rules can be dynamically updated based on evolving legal interpretations. Future work will explore semi-automated rule expansion using NLP techniques. Similarly, if not found in both, then it is searched in the WordNet Synset for its synonyms and again for synonyms with Black Law, and thus legal domain features are extracted. The detailed steps are presented in Algorithm 1.

Algorithm 1: Acquiring domain specific features

1. Input: WordNet, BlackLaw, Content Words List CWL $[w_i, w_{i+1}w_{i+n}]$
2. Output: Updated Content Words List CWL $[w_i, w_{i+1}w_{i+n}]$
3. for *each*, $w_i \in CWL$ **do**
4. //Search in WordNet and BlackLaw
5. **if** $w_i \in WordNet$ and $w_i \in BlackLaw$ **then**
6. $w_i = w_i$
7. **else if** $w_i \in WordNet$ but $w_i \notin BlackLaw$ **then**
8. //Search WordNet Synonyms w_s of w_i in *BlackLaw*
9. **if** $w_s \in BlackLaw$ **then**
10. $w_i \leftarrow bw$
11. **else**
12. $w_i = w_i$
13. **else if** $w_i \in BlackLaw$ and $w_i \notin WordNet$ **then**
14. $w_i = w_i$
15. **else if** $w_i \notin WordNet$ and $w_i \notin BlackLaw$ **then**
16. //Search w_i in $BlackLaw$ Definitions bwd
17 **if** $w_i \in bwd$ **then**
18. Update w_i with the corresponding $BlackLaw$ root word brw of bdw
19. $w_i \leftarrow brw$
20. **else**
21. $w_i = w_i$

Where w_i is the feature, CWL is the content words list, w_{i+1} is the next feature in CWL, w_s is the WordNet synonyms that exist in Synset. brw_i is the Black Law root word, and bwd_i are the Black Law definitions.

The acquisition of legal domain features is a complex task due to searching same feature in two different dictionaries. Sometimes, the feature is not found in the definition and root level of Black Law, due to which the algorithm searches one step lower, i.e., in the synonyms, which reduces the execution speed but improves the efficiency in terms of accuracy. The overall performance of acquiring legal domain features has a positive impact and improves the performance of the proposed framework.

3.4 Semantic Similarity Determination

We aim to find semantically similar cases in the legal knowledge base for input judicial cases. To retrieve semantically similar cases, we utilized the legal knowledge base and the legal domain feature vector of the input judicial case. The semantic similarity model was built using deep learning techniques and a rule-based approach. We employed the Long Short-Term Memory (LSTM) model, to which we fed two feature vectors

to assess their similarity. Both the vectors are pre-processed and have only the following four content words: (i) Noun, (ii) Verb, (iii) Adverb, and (iv) Adjective. We incorporated rules into the LSTM model for the selection of similarity functions. Fig. 4 depicts the comprehensive architecture of the proposed framework.

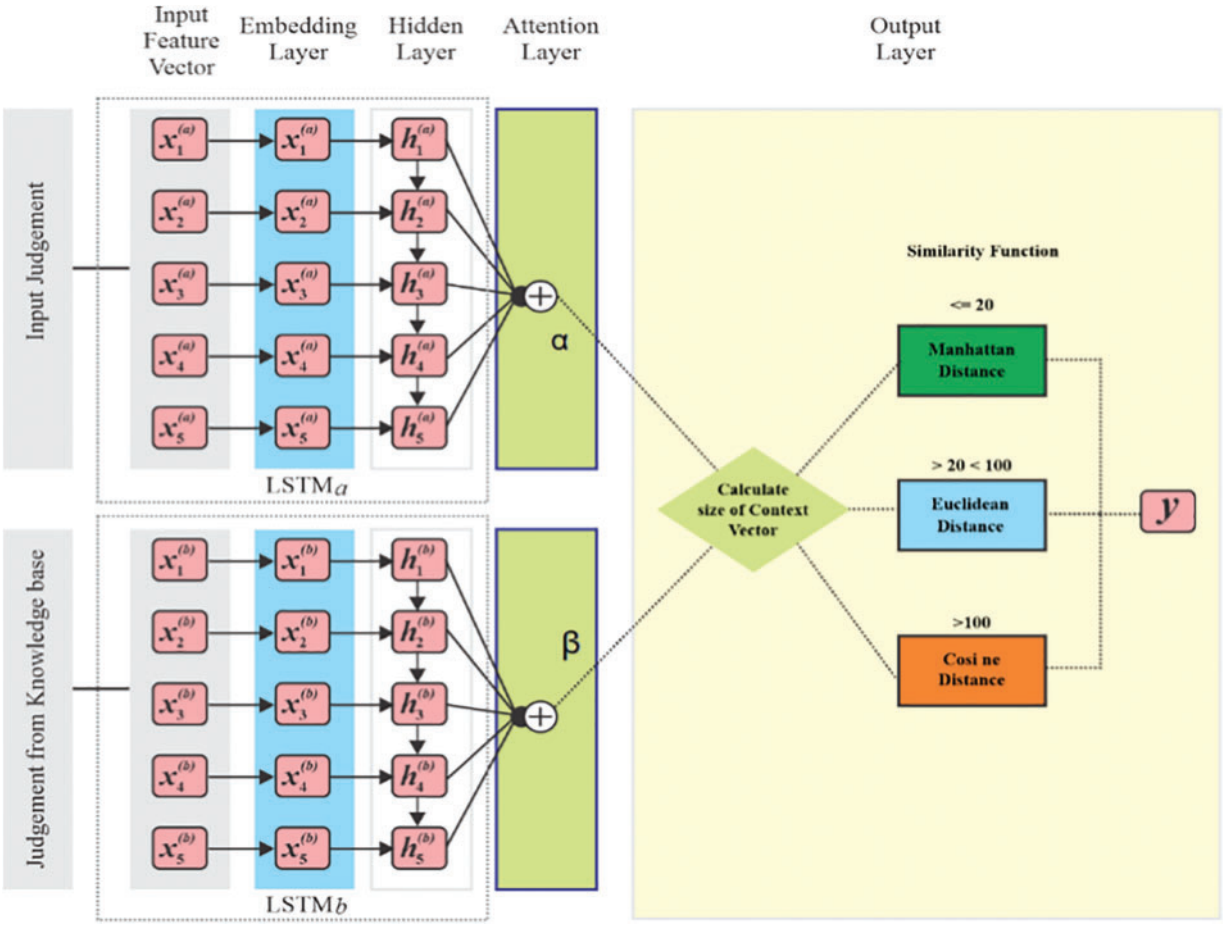

Figure 4: Overall architecture of the proposed framework

In the framework we suggest, we employ the Siamese LSTM model and an additional attention layer to compute the semantic similarity between two judgments. The size of the judgment varies, as sometimes it's tiny and sometimes very large because it depends on the case's nature. To retrieve a similar judgment, it is important to apply the most efficient similarity function. The similarity is the distance between two points in a vector space, and the result depends on the size of the vector. For one input judicial case, there may be multiple similar cases in the knowledge that are different in size. To apply a single similarity function to all the sizes seems unfair and reduces the accuracy of the model. To overcome this issue of the selection of a relevant similarity function, we integrate a rule-based approach that decides to select the relevant similarity function based on the size of the vector.

The Siamese LSTM takes the feature vectors of both judgments and represents them in the hidden states to encode the semantic meaning of the judgments. In the attention layer, it is determined which feature gives more attention over the others in the judgment, and it creates a context vector. Fig. 5 showcases the complete architecture of the Siamese LSTM, comprising five distinct layers from the input layer (judgments) to the

output layer (similarity score). The model is enhanced with the inclusion of an attention layer, and rules are implemented in the output layer to enhance the performance of the LSTM model.

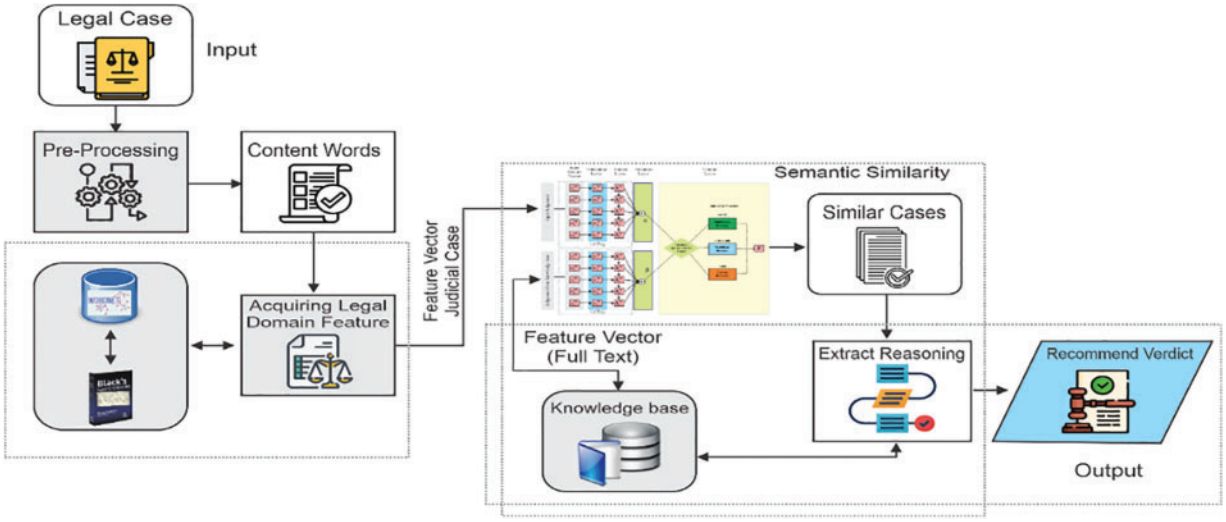

Figure 5: Modeling Siamese LSTM with attention and rule-based layer

Within this investigation, each judgment is portrayed as an individual row in the legal knowledge base. Additionally, every feature in a judgment is depicted as a vector denoted by $x_i^{(a)}$, and similarly, $x_i^{(b)}$ represents the feature in the second judgment. The Siamese LSTM model does not leverage all the hidden states from each LSTM; rather, it only utilizes the final hidden state of each LSTM. This approach may lead to the neglect of some information. In this proposed research, we enhanced the Siamese LSTM model by incorporating an attention layer to address the challenge of missing information. In doing so, we utilized all the hidden states, denoted by $H^{(a)} = \{h_1^{(a)}, h_2^{(a)}, h_3^{(a)}, \ldots, h_l^{(a)}\}$, $H^{(b)} = \{h_1^{(b)}, h_2^{(b)}, h_3^{(b)}, \ldots, h_l^{(b)}\}$ where $h_i^{(a)}$ and $h_i^{(b)}$ are the hidden state of both $LSTM_a$ and $LSTM_b$ at step i, in which all information of the judgment are summarized to x_i, Here, $h_i^{(a)}$ and $h_i^{(b)}$ denote the hidden states of both $LSTM_a$ and $LSTM_b$ at step i, where all information from the judgment is condensed into x_i, with L representing the length of the judgment. The weight of $LSTM_a$ is denoted by α and $LSTM_b$ by β. In the attention layer, the attention mechanism evaluates the significance of a feature over a context vector by determining the weight α_i based on the importance of the feature. The model acquires a mapping from sequences of input vectors with variable lengths, ranging in dimensionality from d_{in} to R_{drep}, where the input dimension is configured as $d_{in} = 300$ and $R_{drep} = 50$. Consequently, the output from all the hidden states serves as the ultimate representation encapsulating the semantic meaning of the input pairs. As a result, the final representation, which captures the semantic meaning of the input pairs, is derived from the output of all hidden states. Before applying the similarity function, the input to the LSTM network is organized using parameters α and β. The similarity function is then executed based on the vector dimensions, under the guidelines established for the output layer. We utilized three distinct similarity functions: (i) Cosine Similarity, (ii) Euclidean Similarity, and (iii) Manhattan Similarity, which is dynamically chosen by the rules in the output layer, with each function constrained between 0 and 1. The algorithms governing these rules are elucidated in Algorithm 2.

Algorithm 2: Applying relevant similarity function

1. Input: Input Feature Vector $IFV\,[IFV_i, IFV_{i+1}, IFV_{i+n}]$)
2. Knowledge Base Vector $KBV\,[KBV_i, KBV_{i+1}, KBV_{i+n}]$)
3. Similarity Functions $SF\,[SF_{Manhattan}, SF_{Euclidean}, SF_{Cosine}]$)
4. Output: List of Similar Documents
5. for each, ifv_i **do**
6. //Check the size of input feature vector iv_i
7. **if** the size of $ifv_i \leq 20$ words **then**
8. //Apply $Manhattan$ Similarity to input feature vector IFV_i and Knowledge Base Vector KBV_i
9. $SF \leftarrow SF_{Manhattan}$
10. $w_i = w_i$
11. **else if** size of $ifv_i \geq 20$ words and of $ifv_i \leq 100$ words **then**
12. //Apply $Euclidean$ Similarity to input feature vector IFV_i and Knowledge Base Vector KBV_i
13. $SF \leftarrow SF_{Euclidean}$
14. **else**
15. //Apply $Cosine$ Similarity to input feature vector IFV_i and Knowledge Base Vector KBV_i if the size is greater than 100 words
16. $SF \leftarrow SF_{Cosine}$

Backpropagation is computed throughout the training process based on the similarity between human labeling and the disparities between the prediction and the output states of the two LSTMs. To visualize the similarity outcomes, a threshold value (k = 5) is employed, extracting the top five most similar judgments from the legal knowledge base for the input judgment.

3.5 Verdict Recommendation

In the verdict recommendation module, we extract the relevant information for the top (5) similar judgments. The legal knowledge base contains multiple attributes for each decided judgment of six categories discussed in detail in Section 3.1. For each judgment in the legal knowledge base, besides the category of the judgment, it includes the year, keywords, and a set of reasoning. The concept diagram of the verdict recommendation model is illustrated in Fig. 6.

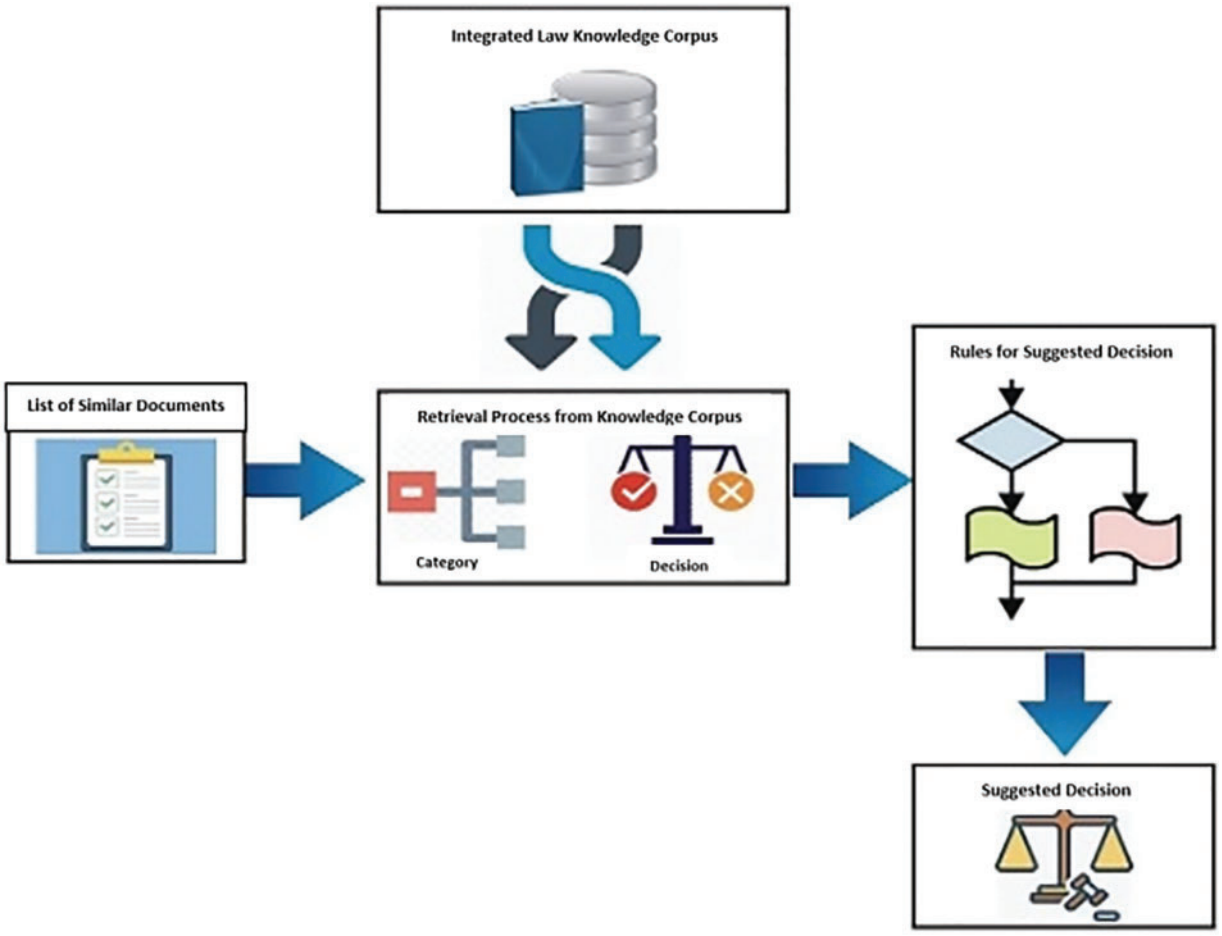

Figure 6: Concept diagram of verdict recommendation model

To achieve the goal of recommendation, the recommendation is performed in two stages. In the first stage, we retrieve the category of the judgment from the legal knowledge base for each similar judgment, then the algorithm compares the categories of similar judgments and judgments in the knowledge base. If the output matches, then we retrieve the appropriate reasoning from the knowledge base, which is represented as the final output. To recommend the final decision, we performed all the steps illustrated in Algorithm 3.

Algorithm 3: Rule-based suggested decision

1. Input: Similar Document List $simdl\,[d_i, d_{i+1}, d_{i+n}]$, InputCategory
2. Output: Suggested Decision
3. for *each*, $d_i \in simdl$ **do**
4. //Retrieve decisions $seld$ and categories $catsel$ from the Law Knowledge corpus
5. **if** $seld_i \in simdl = seld_{i+n} \in simdl$ **then**
6. //Suggested decision $sugd$ will be the selected decision $seld_i$
7. $sugd \leftarrow seld_i \in simdl$
8. **else if** $seld_i \in simdl \neq seld_{i+n} \in simdl \wedge top5simdl \in simdl = seld_{i+n} \in$
9. $simdl = seld_{i+n} \in simdl$ **then**

(Continued)

Algorithm 3 (continued)

10. //Check the category of the selected decision $catseld_i$ of the top 5 most similar documents $top5simdl$
11. **if** $catseld_i \in top5simdl = catseld_{i+n} \in top5simdl$ **then**
12. $sugd \leftarrow seld_i \in top5simdl$
13. **else**
14. //Compare each category of the selected decision $catseld_i$ of the top 5 most similar documents $top5simdl$ with the input category $catinp$
15. **if** $caninput \in catseld_i$ **then**
16. $sugd \leftarrow seld_i \in matcatseld$
17. **else**
18. $sugd \leftarrow Referredsimilardocuments$
19. **else**
20. $sugd \leftarrow Referredsimilardocuments$

The implementation of the model is carried out using Python 3.8, and for user access, the Django 3.2.20 web framework is employed, incorporating HTML, CSS, and JavaScript. The server specifications include a Manufacturer: HPE, Model: ProLiant DL380 Gen10, CPU: Intel Xeon Silver 4110 (2 units), RAM: 64 GB, and Storage: 32 TB.

4 Results and Discussion

In this section, we describe the results obtained during the evaluation of our proposed framework with a set of experiments. We discuss the results in a subsection for each objective provided in Section 1.2.

4.1 Enhancing Feature Selection in the Legal Domain Using Black Law & Wordnet

This study is based on the legal domain; thus, it is essential to acquire the domain features to get better results. To accomplish the objective, we developed a rule-based model that acquires legal domain features. The assessment of our rule-based acquisition model's performance was conducted through the utilization of a confusion matrix. This matrix facilitates the evaluation process by comparing known true values with the predicted values. Table 5 displays the 2 × 2 confusion matrix utilized in binary classification, calculating metrics such as Accuracy, Precision, Recall, and F-measure.

Table 5: Confusion matrix

Data class	Predicted true	Predicted false
Actual true	TP	FN
Actual false	FP	TN

In feature acquisition, the four basic parameters are measured as follows:

True Positive (TP): Correctly classified to the legal domain, and it is actually of the legal domain

True Negative (TN): Correctly classified as not legal domain, and it is not actually in the legal domain

False Positive (TN): Misclassified to the legal domain, but it is not of the legal domain

False Negative (NN): Misclassified as not a legal domain, but it is of legal domain

We calculate the performance of our algorithm in terms of accuracy, precision, recall, and F-measure, which are based on the confusion matrix. The accuracy was calculated based on the confusion matrix, providing insight into the extent of correct predictions made by our algorithm. The accuracy is determined as follows:

$$Accuracy = \frac{Quantity\ of\ accurate\ predictions}{The\ overall\ count\ of\ predictions} \tag{1}$$

Referring to the confusion matrix (Table 5), the calculation of accuracy is outlined in Eq. (2).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

Precision gauges the accuracy of correctly predicted instances in the positive class. It is calculated by dividing the correct positive values by the predicted positive values, as illustrated in Eq. (3).

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

Recall determines the proportion of correctly identified positive values, indicating the model's ability to accurately recognize the positive class. It is also referred to as the sensitivity of the system. The calculation for recall is as in Eq. (4).

$$Recall = \frac{TS}{TS + FN} \tag{4}$$

In the scenario where the classifier flawlessly categorizes all analogous values, the highest achievable score is 1. Since precision excludes consideration of the negative class, it may not be particularly informative on its own, but it is commonly utilized in conjunction with recall.

The F-measure is an evaluation metric that represents the harmonic mean of precision and recall, multiplied by two. It is also referred to as the F1-measure or F1-score. The calculation for the F-measure is outlined in Eq. (5).

$$F - Measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5}$$

The comprehensive assessment of the legal domain acquisition module encompasses a total of 900 input queries, with 150 input queries from each category, namely Criminal, Civil, Services, Revenue, Constitutional, and Corporate.

Tables 6 and 7 provide a summary of the outcomes concerning both the total number of input judgments and the content words in each category.

Table 6 illustrates the results without applying the rules defined for the acquisition of legal domain features. After evaluation, the average accuracy was 82.19%. In Table 7, we illustrate the results obtained during the experiments of applying the proposed rule-based approach, which outperforms. We evaluated the rule-based model with the same input judgments of input judgments and got an average accuracy of 91.6%. Fig. 7 depicts the average accuracy with and without incorporating legal domain features.

The results demonstrate that incorporating the domain-specific lexicon, Black's Law Dictionary, alongside the general-purpose lexicon, WordNet, improves the extraction of legal domain features by an average of 9.41%. This finding supports the fulfillment of the first objective outlined in Section 1.2.

Table 6: Absence of the integration of legal domain characteristics

Category	Quantity of input assessments	Quantity of textual terms	Without the integration of legal domain characteristics				
			Precision	Recall	F1 Measure	Accuracy	Average accuracy
Criminal	150	23,843	0.90	0.90	0.90	0.84	82.19%
Civil	150	18,714	0.84	0.92	0.88	0.82	
Revenue	150	18,010	0.82	0.90	0.86	0.79	
Constitutional	150	17,103	0.80	0.91	0.85	0.80	
Service	150	22,740	0.86	0.94	0.90	0.84	
Corporate	150	18,830	0.88	0.92	0.90	0.84	

Table 7: Acquisition of legal domain features

Category	Quantity of input assessments	Quantity of textual terms	Acquired the legal domain characteristics				
			Precision	Recall	F1 Measure	Accuracy	Average accuracy
Criminal	150	23,843	0.96	0.9	0.96	0.93	91.6%
Civil	150	18,714	0.95	0.9	0.95	0.92	
Revenue	150	18,010	0.90	0.97	0.95	0.92	
Constitutional	150	17,103	0.94	0.9	0.95	0.92	
Service	150	22,740	0.95	0.95	0.95	0.91	
Corporate	150	18,830	0.94	0.95	0.94	0.90	

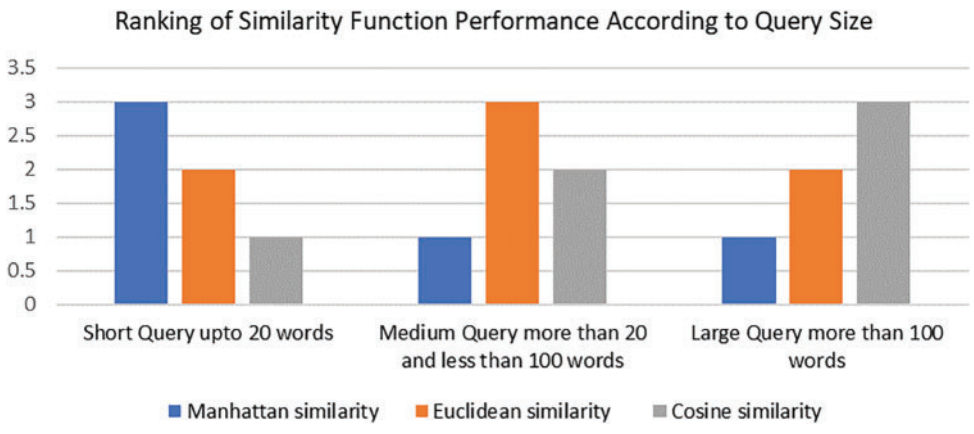

Figure 7: Comparison of without acquisition and the acquisition of legal domain features

4.2 Developing a Verdict Recommendation Framework Using LSTM and Rule-Based Methods with a Legal Knowledge Base

In this segment, we assess the efficiency of the semantic similarity module within the proposed framework. The experiment is conducted by using the combined approach of the Siamese LSTM model with an Attention Layer and a rule-based. We performed the experiments using our legal dataset. Our dataset comprises three sizes of judgments, i.e., short, medium, and large. Judgments with lengths up to 20 words are classified as short, those exceeding 20 words but less than 100 words are classified as medium, and judgments with lengths surpassing 100 words are classified as large. We have a total of 36,809 judgments of all three sizes in six categories. To address imbalance effects, weighted sampling was implemented during training to ensure fair representation of underrepresented legal categories. We split our dataset into a validation set, test set, and training set for all three sizes of input judgment in all six categories as illustrated in Table 8.

Table 8: Partition of the dataset

Classification of judgments	Quantity of judgments				Dataset partitioning		
	Short	Moderate	Substantial	Total	Verification set	Evaluation set	Learning set
Criminal	1000	3000	6440	10,440	1148	2819	6473
Civil	1000	2000	6178	9178	1010	2478	5690
Service	200	600	1425	2225	245	601	1380
Corporate	200	300	1318	1818	200	491	1127
Revenue	1000	2000	4816	7816	860	2110	4846
Constitutional	500	1000	3832	5332	587	1440	3306
Total	3900	8900	24,009	36,809	4049	9938	22,822

We trained the model for 25 epochs over 8 h. We used an Adadelta optimizer [43] with gradient clipping to avoid the problem of exploding gradients. Table 9 displays example queries of varying sizes, namely short, medium, and large, utilized for assessing the proposed model.

Table 10 presents the unified outcomes concerning similarity values for three distinct similarity functions: Manhattan, Euclidean, and Cosine, across all sizes (short, medium, and large).

The combined results shown in Table 11 reveal that the effectiveness of the similarity function varies depending on the size of the input query. To fully understand the performance of each similarity function across various query sizes, a performance ranking table has been created and is displayed in Table 11. Every similarity function is assigned a rank ranging from 1 to 3 across all three query sizes. A higher rank for a similarity function indicates greater similarity in query sizes.

Table 9: Exemplar input queries of varied lengths: short, medium, and large

Short query 1 input (Up to 20 Words)	Medium query 1 input (>20 Words & <100 Words)	Huge query (>100 Words)
"What is the relationship between seniority and promotion? Should the entire seniority list be revised to reflect promotions that are backdated?"	"Section 417 addresses appeals against acquittals and the potential for converting an acquittal into a conviction. Key principles include that the Appellate Court must give appropriate weight and consideration to the Trial Court's assessment of witness credibility. It should also uphold the presumption of innocence for the accused, which remains intact despite an acquittal. The accused is entitled to the benefit of any doubt, and the Appellate Court is generally hesitant to overturn factual findings made by a judge who has had the opportunity to observe the witnesses directly."	"Section 497(2) of the Penal Code (XLV of 1860), relating to Sections 302, 148, and 149, along with Article 185(3) of the Constitution of Pakistan, addresses cases of intentional murder and rioting with deadly weapons concerning bail applications. The court noted that the incident was a spontaneous altercation in which individuals from both groups suffered injuries. The accused did not have a specific role assigned, and the complainant later claimed in a supplementary statement that the accused inflicted the fatal injury. The accused also presented a counter-narrative of the event. Notably, one of the accused in the counter-narrative was granted bail by the Supreme Court. The occurrence was deemed unpremeditated, warranting further investigation, and therefore, bail was granted."

Table 10: Unified outcomes derived from the similarity function according to query size

Function of similarity	Short query up to 20 words	Moderate query, exceeding 20 and below 100 words	Extensive query, exceeding 100 words
	0.49	0.57	0.40
Manhattan similarity	0.27	0.51	0.49
	0.51	0.55	0.54
	0.41	0.70	0.46
Euclidean similarity	0.23	0.60	0.55
	0.37	0.68	0.62
	0.33	0.68	0.52
Cosine similarity	0.18	0.59	0.62
	0.26	0.61	0.65

Table 11: Ranking of similarity function performance according to query size

Function of similarity	Short query up to 20 words	Moderate query, exceeding 20 and below 100 words	Extensive query, exceeding 100 words
Similarity using the Manhattan Metric	3	1	1
Similarity computed through the Euclidean Metric	2	3	2
Similarity is calculated through the Cosine Metric	1	2	3

Table 12 displays the outcomes of each similarity function across all three query sizes. As indicated by the results, the effectiveness of the similarity function is contingent on the query size. The evaluation is conducted on prior judgments stored in the legal knowledge base. As evident from the findings, Cosine Similarity exhibits superior performance when the query size is large. Conversely, for short query sizes, the Manhattan Similarity function demonstrates better performance. Similarly, when the query size is medium, Euclidean outperforms. The similarity values fall within the range of 0 to 1, where 1 indicates identical queries, and 0 signifies completely different queries.

Table 12: Performance of similarity functions on the test set across all three query sizes

Function of similarity	Size of the query	Precision	Recall	F1 Score	Accuracy
Similarity using the Manhattan Metric	Short query (up to 20 words)	0.90	0.95	0.93	0.88
	Moderate-length query (exceeding 20 words and below 100 words)	0.90	0.89	0.89	0.83
	Large query (more than 100 words)	0.82	0.88	0.85	0.77
Similarity computed through the Euclidean Metric	Short query (up to 20 words)	0.87	0.90	0.88	0.82
	Medium query (more than 20 and less than 100 words)	0.92	0.95	0.93	0.89
	Large query (more than 100 words)	0.89	0.90	0.90	0.83
Similarity is calculated through the Cosine Metric	Short query (up to 20 words)	0.81	0.88	0.85	0.76
	Medium query (more than 20 and less than 100 words)	0.90	0.89	0.89	0.83
	Large query (more than 100 words)	0.91	0.94	0.92	0.87

The study highlights that both query length and the structural complexity of legal documents significantly influence retrieval accuracy in the proposed framework. Specifically, Manhattan similarity proves to be the most effective for handling short queries characterized by low sparsity, as it excels in capturing precise lexical matches and maintaining high retrieval accuracy in cases where concise, straightforward legal terms are used. This is particularly useful for queries that rely heavily on exact terminology and clear

legal definitions. On the other hand, as the length and complexity of queries increase, deeper semantic relationships between legal concepts become more critical. In such instances, Cosine similarity is better suited for capturing these nuanced connections, as it evaluates the angular similarity between feature vectors, allowing for more sophisticated semantic retrieval in longer, more complex queries. Euclidean similarity also shows strong performance for medium-length queries, balancing between lexical accuracy and semantic depth. The performance of the retrieval system varies notably across different legal categories, largely depending on the semantic density and inherent complexity of the cases. For example, criminal and corporate law cases exhibit high variability due to the diverse nature of legal issues, terminologies, and case precedents involved, which can complicate semantic similarity assessments. Conversely, constitutional and civil law cases tend to display greater uniformity in structure and legal language, facilitating more consistent retrieval results across these categories. This variation underscores the importance of selecting the appropriate similarity metric based on both the query's length and the legal domain's characteristics. Short queries, which demand high lexical precision, benefit most from Manhattan similarity due to its focus on direct, token-level matching. In contrast, longer queries, which often involve complex legal reasoning and multiple interrelated concepts, gain an advantage from deeper semantic retrieval methods like Euclidean and Cosine similarity, which better capture the broader context and relationships within legal texts. Notably, as query length increases, retrieval accuracy improves, with the system's accuracy rising from 88% for shorter queries to 89% for medium-length queries. This improvement supports the implementation of an adaptive similarity selection mechanism that dynamically chooses the optimal similarity function based on query length and complexity, thereby enhancing overall accuracy in legal case retrieval and demonstrating the system's flexibility in handling diverse legal scenarios.

To recommend a verdict, a rule-based approach is adopted, which extracts reasoning from a legal knowledge base that matches similar judgments. The detailed process of verdict recommendation is described in Section 3.5. The assessment of the rule-based verdict recommendation model involved 600 analogous judgments, divided into six categories: (i) criminal, (ii) civil, (iii) service, (iv) corporate, (v) revenue, and (vi) constitutional. Within the criminal category, there were two disposal modes—Acquittal and Conviction. In the civil category, the modes were Accepted and Rejected. For the rest of the categories, the disposal modes are allowed and dismissed. Table 13 illustrates the results obtained during the process of applying rules for verdict recommendation with an average accuracy of 90%.

Table 13: Rule-based verdict recommendation model

Category	Similar docs	Decision mode	Precision	Recall	F1 Score	Accuracy	Average accuracy
Criminal	100	Acquittal	0.93	0.95	0.94	90%	90%
		Conviction	0.91	0.97	0.94	90%	
Civil	100	Accepted	0.90	0.92	0.91	86%	
		Rejected	0.97	0.95	0.96	94%	
Service	100	Allowed	0.95	0.95	0.95	92%	
		Dismissed	0.93	0.90	0.92	88%	
Corporate	100	Allowed	0.91	0.95	0.93	86%	
		Dismissed	0.93	0.88	0.90	84%	
Revenue	100	Allowed	0.96	1.00	0.98	96%	
		Dismissed	0.91	0.95	0.93	88%	

(Continued)

Table 13 (continued)

Category	Similar docs	Decision mode	Precision	Recall	F1 Score	Accuracy	Average accuracy
Constitutional	100	Allowed	0.96	0.96	0.96	92%	
		Dismissed	0.93	0.95	0.94	90%	

The outcomes presented in the preceding table are juxtaposed with the real decisions made by the honorable judges of higher courts, stored in a legal knowledge base. Consequently, the proposed objective No. 2 mentioned in Section 1.2 is accomplished.

4.3 Evaluating the Effectiveness of the LSTM + Rule-Based Framework against Other Machine Learning and Deep Learning Techniques

Within this segment, the outcomes of the baseline method are contrasted with those of the suggested model. The experimental findings reveal that the proposed rule-based similarity model excels in identifying similarity across all three similarity functions. The ASLSTM model, on the other hand, focuses on determining semantic similarity in English and Arabic datasets through the utilization of the LSTM model featuring an attention mechanism. An alternative approach involves computing semantic similarity through the application of the LSTM model for three similarity functions—Manhattan similarity, Euclidean similarity, and Cosine Similarity. Their model was assessed using a brief paragraph comprising 2 to 3 lines. In comparison, the proposed model is evaluated using queries of three distinct sizes: Short Query (up to 20 words), Medium Query (between 21 and 99 words), and Huge Query (over 100 words). For evaluating the models' accuracy, the comparative outcomes are presented in Table 14.

Table 14: Accuracy comparison of the proposed and contemporary models

Model	Manhattan similarity (Short)	Euclidean similarity (Medium)	Cosine similarity (Large)
CLSum [39]	~0.78	~0.80	~0.79
BERT-CNN [40]	~0.81	~0.81	~0.80
DiscoLQA [41]	~0.84	~0.87	~0.85
ASLSTM	0.82	0.79	0.80
Siamese LSTM	0.83	0.83	0.76
LSTM + CNN	0.85	0.81	0.84
BERT Model	0.81	0.78	0.82
Proposed LSTM + Rule-Based	**0.88**	**0.89**	**0.87**

Based on the outcomes presented in Table 14, it is evident that the proposed model surpasses other methods in terms of accuracy across all three similarity functions.

The results of the proposed decision model are compared with other machine learning methods, and it is evident from the results that the rule-based approach in this model outperforms the other methods.

4.4 External Validation of the LSTM + Rule-Based Framework against Other Legal Dataset

To evaluate the proposed framework with external datasets, we obtained four more datasets from Kaggle. By evaluating our proposed framework with different datasets, our approach to similarity determination and verdict recommendation is effective and accurate. The proposed framework was implemented on the external datasets for all three for all three-similarity functions. Table 15 summarizes the findings obtained during the evaluation. From the results, it is clear that the proposed framework outperforms both internal and external datasets.

Table 15: The external assessment of the proposed approach

Dataset	Description of dataset	Accuracy		
		Manhattan similarity	**Euclidean similarity**	**Cosine similarity**
Supreme court judgment prediction	The dataset of the United States Supreme Court comprises 3304 cases covering the period from 1955 to 2021.	0.83	0.84	0.79
Swiss Federal Supreme Court Dataset (SCD)	The dataset contains 118,443 cases decided by the Swiss Federal Supreme Court between 2007 and September 2023.	0.83	0.86	0.78
Decisions by the Brazilian Supreme Court in Habeas *Corpus*	The dataset contains 22,662 cases decided by the Brazilian Supreme Court between January 2015 and January 2018.	0.84	0.85	0.8
Supreme court of Pakistan judgments	The dataset contains 20,809 cases from 2007 to 2024.	0.85	0.87	0.81
Own legal knowledge base	The Dataset contains 36,809 reported judgments of the Higher Courts of Pakistan from 2017 to 2023.	0.88	0.89	0.87

5 Conclusion and Future Work

This study presented the development of an AI-driven legal ruling recommendation model that integrated deep learning techniques with rule-based reasoning, aiming to enhance both interpretability and adaptability within judicial systems. By combining these two methodologies, the proposed framework addressed the limitations of traditional approaches, particularly in terms of retrieval efficiency and decision-support capabilities. Unlike conventional systems, which often lacked transparency and adaptability across diverse legal domains, this hybrid model provided a more robust and interpretable solution. A key contribution of this research was its focus on addressing the absence of structured legal knowledge bases in Pakistan. To overcome this challenge, the study involved the development and annotation of domain-specific datasets, enabling the integration of rule-based reasoning with advanced deep-learning techniques. This dual approach ensured both the interpretability associated with rule-based systems and the predictive accuracy characteristic of deep learning models. Performance evaluations across the framework's modules demonstrated significant advancements over existing methods. The legal domain feature acquisition module achieved an accuracy of 91.6% and an F1-score of 95%, highlighting the model's effectiveness

in extracting relevant legal features from judicial texts. The semantic similarity determination module, assessed using multiple similarity functions—Manhattan, Euclidean, and Cosine—showed consistently high performance. Specifically, Manhattan similarity yielded 88% accuracy and a 93% F1-score for short queries, Euclidean similarity achieved 89% accuracy with a 93.7% F1-score for medium-length queries, and Cosine similarity recorded 87% accuracy with a 92.5% F1-score for larger queries. The verdict recommendation module outperformed existing benchmarks, attaining 90% accuracy and a 93.75% F1-score. These results not only validated the framework's effectiveness but also underscored the critical role of legal domain feature extraction in improving system performance. Additionally, the findings revealed how query size dynamically influenced the accuracy of similarity assessments, suggesting that different similarity metrics could be optimized based on the nature of the query. The adaptability of this framework presented promising opportunities for broader application across various legal systems. By modifying the legal knowledge base and fine-tuning the LSTM model with multilingual case law datasets, the framework could be tailored to meet the specific needs of different jurisdictions. This flexibility highlighted its potential as a versatile tool for global judicial decision-making support.

Future research can further enhance the efficiency and applicability of the proposed framework through several avenues. Expanding legal dictionaries and ontologies will improve the precision and depth of domain-specific feature extraction. Refining search algorithms within WordNet and legal databases can increase both the accuracy and speed of case retrieval. Additionally, incorporating statutory laws—such as constitutions, legislative acts, and ordinances—alongside case law will create a more comprehensive and reliable legal reference system. Beyond retrieval and recommendation, extending the framework to generate preliminary legal drafts based on identified case similarities and reasoning patterns could significantly support judicial decision-making processes. This feature would not only streamline the preparation of legal documents but also enhance the efficiency of legal practitioners and judges. Furthermore, improving the framework's cross-jurisdictional adaptability would facilitate its implementation across diverse legal frameworks, enabling broader adoption in both common law and civil law systems.

Acknowledgement: The authors would like to thank the Deanship of Scientific Research, Jouf University, Sakaka, Aljouf, for supporting this research work.

Funding Statement: This work was funded by the Deanship of Scientific Research at Jouf University under Grant number DSR–2022–RG–0101.

Author Contributions: The authors confirm their contributions to the paper as follows: study conception and design: Muhammad Hameed Siddiqi, Muhammad Faheem Khan, Jawad Khan; data collection: Asfandyar Khan, Irshad Ahmad, Saad Alanazi; analysis and interpretation of results: Muhammad Hameed Siddiqi, Madallah Alruwaili, Yousef Alhwaiti, Menwa Alshammeri; draft manuscript preparation: Muhammad Hameed Siddiqi, Muhammad Faheem Khan, Jawad Khan. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data and simulation materials used in this study will be provided on demand.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Liu S, Cao J, Li Y, Yang R, Wen Z. Low-resource court judgment summarization for common law systems. Inf Process Manag. 2024;61(5):103796. doi:10.1016/j.ipm.2024.103796.

2. Li Y, Wu J, Luo X. BERT-CNN based evidence retrieval and aggregation for Chinese legal multi-choice question answering. Neural Comput Appl. 2024;36(11):5909–25. doi:10.1007/s00521-023-09380-5.
3. Sovrano F, Palmirani M, Sapienza S, Pistone V. DiscoLQA: zero-shot discourse-based legal question answering on European Legislation. Artif Intell Law. 2024;2024(2):1–37. doi:10.1007/s10506-023-09387-2.
4. Liebowitz J. The handbook of applied expert systems. Boca Raton, FL, USA: CRC Press; 2019.
5. Leith P, Hoey A. The computerised lawyer: a guide to the use of computers in the legal profession. Berlin/Heidelberg, Germany: Springer Science & Business Media; 2012.
6. Waterman DA, Paul J, Peterson M. Expert systems for legal decision making. Expert Syst. 1986;3(4):212–26. doi:10.1111/j.1468-0394.1986.tb00203.x.
7. Branting LK. Data-centric and logic-based models for automated legal problem solving. Artif Intell Law. 2017;25(1):5–27. doi:10.1007/s10506-017-9193-x.
8. Alghazzawi D, Bamasag O, Albeshri A, Sana I, Ullah H, Asghar MZ. Efficient prediction of court judgments using an LSTM+CNN neural network model with an optimal feature set. Mathematics. 2022;10(5):683. doi:10.3390/math10050683.
9. Aletras N, Tsarapatsanis D, Preoţiuc-Pietro D, Lampos V. Predicting judicial decisions of the European court of human rights: a natural language processing perspective. PeerJ Comput Sci. 2016;2(2):e93. doi:10.7717/peerj-cs.93.
10. Katz DM, Bommarito MJ, Blackman J. A general approach for predicting the behavior of the Supreme Court of the United States. PLoS One. 2017;12(4):e0174698. doi:10.1371/journal.pone.0174698.
11. Timmer I, Rietveld R. Rule-based systems for decision support and decision-making in Dutch legal practice. A brief overview of applications and implications. Droit Société. 2019;103(3):517–34. doi:10.3917/drs1.103.0517.
12. Collenette J, Atkinson K, Bench-Capon T. Explainable AI tools for legal reasoning about cases: a study on the European Court of Human Rights. Artif Intell. 2023;317(2):103861. doi:10.1016/j.artint.2023.103861.
13. KantShankhdhar G, Singh VK, Darbari M. Legal semantic web—a recommendation system. Int J Appl Inf Syst. 2014;7(3):21–7. doi:10.5120/ijais14-451165.
14. Huang S, Liu A, Zhang S, Wang T, Xiong NN. BD-VTE: a novel baseline data based verifiable trust evaluation scheme for smart network systems. IEEE Trans Netw Sci Eng. 2021;8(3):2087–105. doi:10.1109/TNSE.2020.3014455.
15. Kort F. Predicting supreme court decisions mathematically: a quantitative analysis of the right to counsel cases. Am Polit Sci Rev. 1957;51(1):1–12. doi:10.2307/1951767.
16. Leith P. The rise and fall of the legal expert system. Int Rev Law Comput Technol. 2016;30(3):94–106. doi:10.1080/13600869.2016.1232465.
17. Liu YH, Chen YL. A two-phase sentiment analysis approach for judgement prediction. J Inf Sci. 2018;44(5):594–607. doi:10.1177/0165551517722741.
18. Tuggener D, Von Däniken P, Peetz T, Cieliebak M. LEDGAR: a large-scale multi-label corpus for text classification of legal provisions in contracts. In: Proceedings of the Twelfth Language Resources and Evaluation Conference; 2020 May 11–16; Marseille, France. p. 1235–41.
19. Shirsat K, Keni A, Chavan P, Gosavi M. Legal judgement prediction system. Int Res J Eng Technol. 2021;8(5):734–8.
20. Li S, Zhang H, Ye L, Guo X, Fang B. MANN: a multichannel attentive neural network for legal judgment prediction. IEEE Access. 2019;7:151144–55. doi:10.1109/ACCESS.2019.2945771.
21. Guo X, Zhang H, Ye L, Li S. TenLa: an approach based on controllable tensor decomposition and optimized lasso regression for judgement prediction of legal cases. Appl Intell. 2021;51(4):2233–52. doi:10.1007/s10489-020-01912-z.
22. Sukanya G, Priyadarshini J. A meta analysis of attention models on legal judgment prediction system. Int J Adv Comput Sci Appl. 2021;12(2):531–8. doi:10.14569/issn.2156-5570.
23. Shang X. A computational intelligence model for legal prediction and decision support. Comput Intell Neurosci. 2022;2022(7):5795189. doi:10.1155/2022/5795189.
24. Zhao G, Shi H, Wang J. A grey BP neural network-based model for prediction of court decision service rate. Comput Intell Neurosci. 2022;2022(23):7364375. doi:10.1155/2022/7364375.

25. Gomes T, Ladeira M. A new conceptual framework for enhancing legal information retrieval at the Brazilian Superior Court of Justice. In: Proceedings of the 12th International Conference on Management of Digital EcoSystems; 2020 Nov 2–4; Abu Dhabi, UAE. p. 26–9. doi:10.1145/3415958.3433087.
26. Verma A, Morato J, Jain A, Arora A. Relevant subsection retrieval for law domain question answer system. In: Data visualization and knowledge engineering: spotting data points with artificial intelligence. Berlin/Heidelberg, Germany: Springer; 2020. p. 299–319.
27. Collarana D, Heuss T, Lehmann J, Lytra I, Maheshwari G, Nedelchev R, et al. A question answering system on regulatory documents. In: Legal knowledge and information systems. Amsterdam; The Netherlands: IOS Press; 2018. p. 41–50.
28. Morimoto A, Kubo D, Sato M, Shindo H, Matsumoto Y. Legal question answering system using neural attention. COLIEE@ICAIL. 2017;2017:79–89.
29. Yao F, Xiao C, Wang X, Liu Z, Hou L, Tu C, et al. LEVEN: a large-scale Chinese legal event detection dataset. arXiv:2203.08556. 2022.
30. Bhattacharya P, Hiware K, Rajgaria S, Pochhi N, Ghosh K, Ghosh S. A comparative study of summarization algorithms applied to legal case judgments. In: Advances in Information Retrieval: 41st European Conference on IR Research, ECIR 2019; 2019 Apr 14–18; Cologne, Germany. Berlin/Heidelberg, Germany: Springer International Publishing. p. 413–28.
31. Vacek T, Teo R, Song D, Nugent T, Cowling C, Schilder F. Litigation analytics: case outcomes extracted from US federal court dockets. In: Proceedings of the Natural Legal Language Processing Workshop 2019; 2019 Jun 7; Minneapolis, MN, USA. p. 45–54. doi:10.18653/v1/w19-2206.
32. Zouaoui S, Rezeg K. Islamic inheritance calculation system based on Arabic ontology (AraFamOnto). J King Saud Univ Comput Inf Sci. 2021;33(1):68–76. doi:10.1016/j.jksuci.2018.11.015.
33. Abu Shamma S, Ayasa A, Sleem W, Yahya A. Information extraction from Arabic law documents. In: IEEE 14th International Conference on Application of Information and Communication Technologies (AICT); 2020 Oct 7–9; Tashkent, Uzbekistan. p. 1–6. doi:10.1109/aict50176.2020.9368577.
34. Bonifacio LH, Vilela PA, Lobato GR, Fernandes ER. A study on the impact of intradomain finetuning of deep language models for legal named entity recognition in Portuguese. In: Intelligent Systems: 9th Brazilian Conference, BRACIS 2020; 2020 Oct 20–23; Rio Grande, Brazil. Berlin/Heidelberg, Germany: Springer International Publishing; 2020. p. 648–62.
35. Revenko A, Rehm G. Automatic induction of named entity classes from legal text corpora. In: 19th International Semantic Web Conference; 2020 Nov 2–6; Athens, Greece.
36. Hong Z, Zhou Q, Zhang R, Li W, Mo T. Legal feature enhanced semantic matching network for similar case matching. In: 2020 International Joint Conference on Neural Networks (IJCNN); 2020 Jul 19–24; Glasgow, UK. p. 1–8. doi:10.1109/ijcnn48605.2020.9207528.
37. Chalkidis I, Fergadiotis M, Tsarapatsanis D, Aletras N, Androutsopoulos I, Malakasiotis P. Paragraph-level rationale extraction through regularization: a case study on European court of human rights cases. arXiv:2103.13084. 2021.
38. Almuzaini HA, Azmi AM. TaSbeeb: a judicial decision support system based on deep learning framework. J King Saud Univ Comput Inf Sci. 2023;35(8):101695. doi:10.1016/j.jksuci.2023.101695.
39. Mumcuoğlu E, Öztürk CE, Ozaktas HM, Koç A. Natural language processing in law: prediction of outcomes in the higher courts of Turkey. Inf Process Manag. 2021;58(5):102684. doi:10.1016/j.ipm.2021.102684.
40. Ma Y, Shao Y, Wu Y, Liu Y, Zhang R, Zhang M, et al. LeCaRD: a legal case retrieval dataset for Chinese law system. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval; 2021 Jul 11–15; ACM Publisher. p. 2342–8. doi:10.1145/3404835.
41. Ma L, Zhang Y, Wang T, Liu X, Ye W, Sun C, et al. Legal judgment prediction with multi-stage case representation learning in the real court setting. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval; 2021 Jul 11–15; ACM Publisher. p. 993–1002. doi:10.1145/3404835.

42. Le Q, Mikolov T. Distributed representations of sentences and documents. In: International Conference on Machine Learning; 2014 Jun 21–26; Beijing, China. p. 1188–96.
43. Zeiler MD. ADADELTA: an adaptive learning rate method. arXiv:1212.5701. 2012. doi:10.48550/arxiv.1212.5701.