2025 International Conference on Advances in Modern Age Technologies for Health and Engineering Science (AMATHE) | 979-8-3315-0103-7/25/$31.00 ©2025 IEEE | DOI: 10.1109/AMATHE65477.2025.11080869

AI-Driven Legal Assistance: Optimizing Document Retrieval and Case Classification with NLP Techniques

Mahima Raje singh, Dr.Sartaj Ahmad, Khushi Sharma, Prakriti Sharma, Pranjali Kaushik

KIET Group of Institutions, Delhi NCR, India

{mahima.2125it1109, sartaj.ahmad,khushi.2125it1192, prakriti.2125it1028, pranjali.2125it1008}@kiet.edu

***Abstract*— Access to legal information is difficult, particularly for those who are unfamiliar with legal terminology. This research introduces an AI legal chatbot to improve document retrieval and case classification. The chatbot uses TF-IDF and cosine similarity to rank documents in order of relevance to user requests. It will classify instances so that users can navigate to the appropriate legal environment. The solution applies core NLP techniques, such as keyword extraction, to Elasticsearch, resulting in scalability and increased performance. That strategy produces better search results with a faster reaction time and could replace the old method of contacting lawyers. This study proposes constructing a scalable, simple, and low-complexity technique for easy access to legal resources.**

***Keywords:* Legal chatbot, document retrieval, case categorization, TF-IDF, cosine similarity, Elasticsearch, natural language processing (NLP)**

I. Introduction

Access to credible legal information is another key difficulty, particularly for people unfamiliar with the legal jargon. There is a lot of legal material available, and without some assistance, it would be difficult to discover the right document or case in a timely and cost-effective manner [1]. New developments in artificial intelligence (AI) and natural language processing (NLP) open up new opportunities for overcoming this challenge [2], [3]. AI-powered systems can now retrieve, categorize, and show legal content exactly, allowing customers to obtain relevant resources without relying on traditional sources [4].

This study focuses on the creation of an AI-powered legal chatbot that can improve case categorization and legal document retrieval. The Chatbot uses NLP-based approaches such as Term Frequency-Inverse Document Frequency, as well as cosine similarity, to give context-specific answers that are free of inaccuracies or bias [5], [2]. Additionally, it allows for the categorization of legal issues [6]. Elasticsearch is utilized for performance and scalability since it enables faster search and retrieval from large databases [23]. This strategy offers a more efficient and accessible alternative to typical legal research approaches. The current study will address the democratization of legal services by attempting to establish a low-cost platform for accessing credible legal information, thereby ensuring equal access to legal aid for all [7].

II. Literature Review

The increasing complexity of tasks in textual similarity and information retrieval, especially in areas such as legal document analysis, has motivated the development of various methodologies. This paper addresses significant developments in the use of TF-IDF and cosine similarity, as well as complementary approaches to handle document retrieval, classification, and summarization difficulties.

Hybrid models have seen significant improvements in retrieval precision and scalability. Traditional approaches such as BM25 are suited for keyword-based retrieval but cannot understand the context and semantics of legal language. [8] used BM25 with BERT for legal document retrieval. BM25's keyword precision is augmented by BERT's ability to capture deep semantic meaning [9] advances the approach by integrating not only BM25 and BERT but also a Semantic Thesaurus that addresses domain-specific legal terminology. It performs well on smaller datasets scaling might become impractical without significant computational resources. In [8] authors present a hybrid model combining BM25 with neural semantic matching. This combination can introduce integration challenges because it might be difficult to get a proper balance between the precision of BM25 based solely on keywords and a better semantic understanding of neural networks. Without proper tuning, this combination might lead to less accurate retrieval results. In [23] author uses BM25 within Elasticsearch in ranking the documents using terms' relevance within the document, fine-tuned parameter settings to achieve better scores, and the full use of capabilities of Elasticsearch to efficiently retrieve information.Research in [10] presents BERT-based encoding as well as cosine similarity in improving the retrieval of cases. The research in [9] presents TF-IDF, Word2Vec, Law2Vec, and BERT for computing similarity among legal cases. The work in [11] shows TF-IDF, BM25, word embeddings, and Sentence-BERT for automatically retrieving relevant prior cases. The authors in [[12]], [9], and [14] highlight very similar challenges like having to work with small-sized and imbalanced datasets as well as the necessity to train models like BERT in domain-specific areas, and the diversity as well as coverage could improve by increasing the datasets.

The research by [13] addressed the scalability issues. It uses a pre-learned domain-specific word embeddings like Law2Vec, LeGlove for semantic representation with distributed computing frameworks like MapReduce and Spark to ensure scalability. By integrating these methods, the system balances semantic understanding with computational efficiency. Legal document retrieval requires both precision and contextual understanding. In[14] The authors proposed a two-phase system combining TF-IDF, LDA, and Ranking SVM, enhanced with syntactic and semantic measures of similarity, to answer legal questions. Similarly, [15] integrated TF-IDF, LDA, and Ranking SVM to recognize personal injury cases, emphasizing semantic relevance. Both approaches emphasize the fact that hybrid models are inevitable.

In [16],[17] researchers reinforce the potential of hybrid systems further by analyzing their roles in enhancing retrieval capabilities in large-scale datasets. In [24] author compare a range of methods for legal text summarization including LexRank, TextRank, and domain-specific models such as Legal Pegasus. It utilizes cosine similarity to compute the relevance between it makes possible the extraction of semantic relations among documents [18].

In [19] authors introduce research which improves semantic meaning through the application of cosine similarity using TF-IDF, word embeddings, and soft cosine similarity. While these increase precision, the potential for improvement beyond this in Doc2Vec is uncharted territory. The work in [6] improves retrieval for legal documents by employing TF-IDF, cosine similarity, and advanced word embeddings such as GloVe and Doc2Vec in the representation of legal texts as vectors. While the method successfully ranks relevant documents, a gap exists in the exploration of more sophisticated models, such as LEGAL-BERT, for higher accuracy and the handling of complex legal queries or even using Law2Vec, a variant of Word2Vec or similar word embedding models specially tailored for the legal domain. The combination of TF-IDF and cosine similarity has been successful in most applications, ranging from legal document retrieval [14] to classification [7],[3].

Our research builds upon these foundations, aiming to bridge gaps in document similarity analysis and classification efficiency.

III. METHODOLOGY

A. Model Overview

The Legal Chatbot utilizes artificial intelligence and natural language processing to automate legal research [9]. It uses Elasticsearch for faster retrieval [23], cosine similarity for case categorization [5], and TF-IDF for document ranking [2],[20] to provide the user with relevant and contextually accurate legal information on family, criminal, and property law. This scalable technique boosts the chatbot's ability to provide timely and accurate legal assistance [16]. Fig. 1 depicts the workflow of the Legal Chatbot, which uses the hybrid TF-IDF technique with Elasticsearch to process queries, classify situations, and retrieve documents [8].

B. Dataset Description

The chatbot is trained on a dataset of 100 legal papers gathered from public legal sources, including a variety of legal subjects such as family, criminal, and property law. These papers include statutes, court decisions, case rulings, and legal pleadings, all of which are formatted using legal terminology, headings, and paragraph numbers [6]. This dataset is intended to closely resemble real-world legal papers, allowing the chatbot to handle genuine legal queries. The dataset replicates real-world user conditions, which improves the chatbot's retrieval accuracy and context relevance [7].

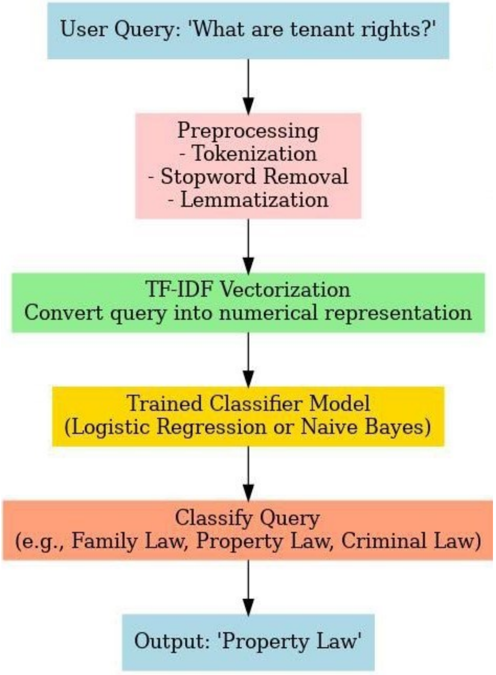
User Query: 'What are tenant rights?'
Preprocessing
- Tokenization
- Stopword Removal
- Lemmatization
TF-IDF Vectorization
Convert query into numerical representation
Trained Classifier Model
(Logistic Regression or Naive Bayes)
Classify Query
(e.g., Family Law, Property Law, Criminal Law)
Output: 'Property Law'

Fig. 1. Workflow of chatbot

Table 1 lists key parameters from the dataset used to train the Legal Chatbot. It specifies the quantity of documents, length, and legal jurisdictions covered.

TABLE I. Dataset description

Attribute	Value
Number of documents	100
Average lines per document	100-500
Average words per document	2,000 to 10,000
Legal domains covered	Family, Criminal, Property

C. Model Planning

The Hybrid Model was chosen for the Legal Chatbot because it needed to strike a balance between precision and efficiency, both of which are critical requirements for retrieving legal documents and classifying cases [5]. We tested two key approaches during the pilot phase:

TF-IDF and Cosine Similarity: This technique turns documents into vectors using phrase frequencies [21]. The advantage is that it can discover key phrases in smaller datasets. The downside of employing TF-IDF is its inability to handle vast amounts of diverse legal data because it is based on term frequencies and document vectors, making it difficult to analyze due to the complexity of legal texts [5].

Elasticsearch: This search engine-based solution is ideal for quick document retrieval since it ranks documents based on term frequency and relevance utilising an inverted index and the BM25 algorithm [23], [22] Although Elasticsearch has a fast retrieval speed, it lacks the deeper semantic understanding required for accurate case categorization.

To address the shortcomings of the two techniques, we created a Hybrid Model [8], which combines the rich text representation ability of TF-IDF with the quick search capabilities of Elasticsearch.

Thus, it improves document categorization and retrieval efficiency. This increases the possibilities of accurately categorizing legal papers and faster retrieval. We aimed to develop a system that could efficiently handle legal enquiries, classify cases, and retrieve relevant documents rapidly; this is critical in the legal area, where information must be timely and correct. The next section will detail the Model Development procedure that resulted in the creation of this Hybrid Model.

D. Model Development

a) Query Processing and Case Categorization

The legal text is cleaned and tokenized to help process the query. It removes special characters, punctuation, and stop words while retaining the essential words [7]. The question is vectorized using TF-IDF, which highlights significant terms from both the query and the documents [5]. The document's relevance is judged by its cosine similarity and TF-IDF.

Steps of the process:

1. The query is tokenized meaningfully.
2. TF-IDF is applied to represent both the query and documents as vectors.
3. Cosine similarity of the query vector and all the document vectors are calculated to rank the documents.

Cosine similarity estimates the angle between vectors. Values nearer to one indicate a greater resemblance. Thus, utilising this, the chatbot may fetch the most relevant documents to the user's question [9].

Cosine Similarity can be calculated as follows

$$\mathrm{Cos}(\theta) = \frac{A \cdot B}{\|A\| \, \|B\|}$$

Where A and B are the vectors, and cosine similarity ranges from 0 (no similarity) to 1 (perfect similarity).

$$\text{Cosine Similarity} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Code 1: For TF-IDF and Cosine Similarity Calculation

```
from sklearn.feature_extraction.text
import TfidfVectorizer
from sklearn.metrics.pairwise import
cosine_similarity

# Sample documents and a query
documents = ["Property disputes in
court.", "Laws regarding family
disputes.", "Criminal law covers theft."]
query = ["Laws on property disputes."]

# Vectorize the documents and query
vectorizer = TfidfVectorizer()
tfidf_matrix =
vectorizer.fit_transform(documents +
query)

# Calculate cosine similarity
cos_sim =
cosine_similarity(tfidf_matrix[-1],
tfidf_matrix[:-1])
```

Code 1 demonstrates how to compute cosine similarity by vectorizing the query and documents using TF-IDF and then evaluating relevance based on the cosine similarity between their vectors.

Following fig. 2 shows an example of query processing that leads to case classification. The system examines and ranks legal documents based on the user's inquiry.

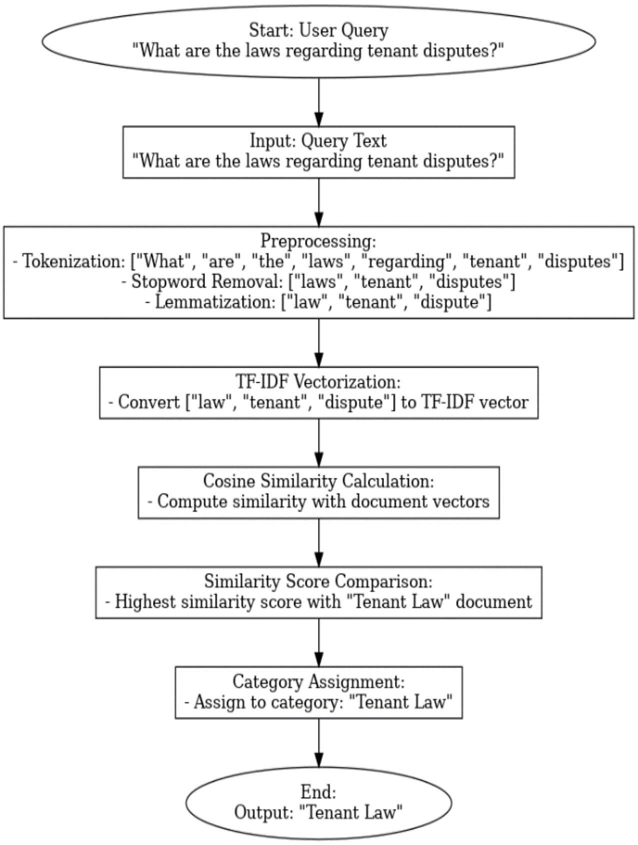
Start: User Query
"What are the laws regarding tenant disputes?"
Input: Query Text
"What are the laws regarding tenant disputes?"
Preprocessing:
- Tokenization: ["What", "are", "the", "laws", "regarding", "tenant", "disputes"]
- Stopword Removal: ["laws", "tenant", "disputes"]
- Lemmatization: ["law", "tenant", "dispute"]
TF-IDF Vectorization:
- Convert ["law", "tenant", "dispute"] to TF-IDF vector
Cosine Similarity Calculation:
- Compute similarity with document vectors
Similarity Score Comparison:
- Highest similarity score with "Tenant Law" document
Category Assignment:
- Assign to category: "Tenant Law"
End:
Output: "Tenant Law"

Fig. 2. Flowchart for Query Processing and Case Categorization

b) Document Retrieval using Elasticsearch

Following categorization, Elasticsearch employs an inverted index to get the most relevant documents by swiftly matching query terms to those in the indexed documents [23].

The BM25 method ranks papers according on phrase frequency and inverse document frequency, with the more relevant and rarer phrases appearing higher [23].

For example, an Elasticsearch query for "tenant rights" ranks documents according on their relevance to the query, which is fine-tuned using the BM25 algorithm.

Code 2: Document retrieval using Elasticsearch Query

```
from elasticsearch import Elasticsearch
# Connect to Elasticsearch
es = Elasticsearch()
# Search query
query = {"query": {"match": {"content":
"tenant rights"}}}
# Execute and print results
response =
es.search(index="legal_documents",
body=query)
for hit in response['hits']['hits']:
    print(hit["_source"]["content"])
```

Code 2 displays a code snippet that demonstrates how Elasticsearch executes a query and retrieves relevant documents from the indexed collection.
Figure 3 depicts the flow of query processing and document retrieval using Elasticsearch in the Legal Chatbot.

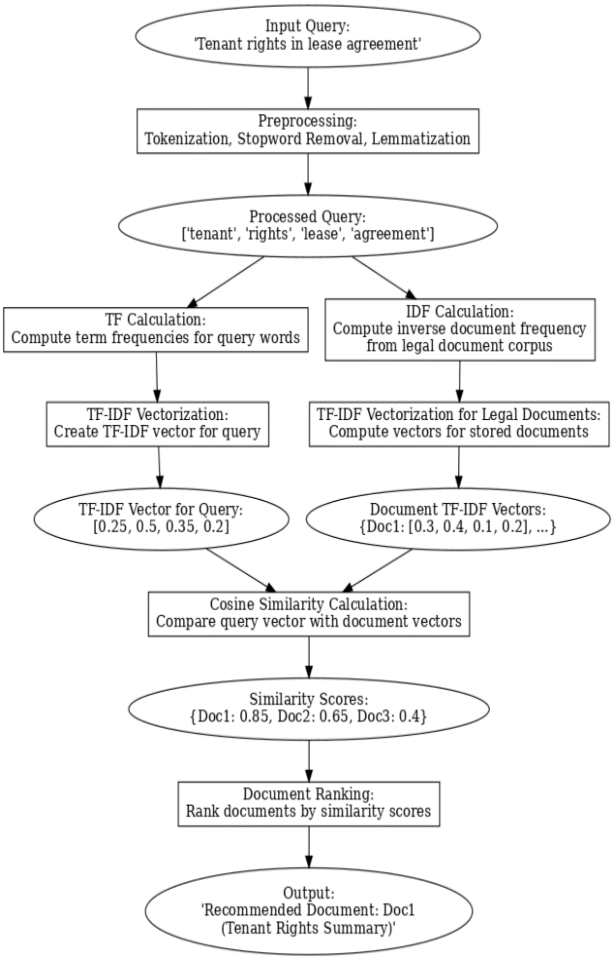
Input Query:
'Tenant rights in lease agreement'
Preprocessing:
Tokenization, Stopword Removal, Lemmatization
Processed Query:
['tenant', 'rights', 'lease', 'agreement']
TF Calculation:
Compute term frequencies for query words
IDF Calculation:
Compute inverse document frequency
from legal document corpus
TF-IDF Vectorization:
Create TF-IDF vector for query
TF-IDF Vectorization for Legal Documents:
Compute vectors for stored documents
TF-IDF Vector for Query:
[0.25, 0.5, 0.35, 0.2]
Document TF-IDF Vectors:
{Doc1: [0.3, 0.4, 0.1, 0.2], ...}
Cosine Similarity Calculation:
Compare query vector with document vectors
Similarity Scores:
{Doc1: 0.85, Doc2: 0.65, Doc3: 0.4}
Document Ranking:
Rank documents by similarity scores
Output:
'Recommended Document: Doc1
(Tenant Rights Summary)'

Fig 3. Flowchart for Query Processing and Document Retrieval

With the approach laid out, it is critical to assess the performance of the Legal Chatbot. The following part delves deeper into the assessment of how the chatbot retrieves and categorizes legal knowledge, comparing it to other top legal tools [9], [24].

Further paper shows how well the system performs its tasks by looking at some of the key metrics such as accuracy, precision, recall, and F1-Score [5], [8], and then discusses the advantages and disadvantages of the approach.

IV. RESULTS AND DISCUSSION

This section discusses the performance of Legal Chatbot in detail with an analysis of specific key evaluation criteria, which are accuracy, precision, recall, and F1-score [5], [8]. These metrics measure the effectiveness of the system to retrieve relevant documents and classify legal cases.

A. *Evaluation Metrics*

The chatbot's performance is evaluated using accuracy, precision, recall, and F1-score criteria. These metrics assess how effectively the chatbot gets relevant documents and classifies occurrences.

a) Accuracy: Calculates the percentage of correctly predicted documents over total predictions. It shows how often the chatbot is correct in its suggestions.

$$\boxed{\text{Accuracy} = \frac{True\ Positives + True\ Negatives}{Total\ Predictions}}$$

b) Precision: This is the ratio of correctly predicted relevant documents by the chatbot to the number of documents it labeled as relevant. It determines how relevant the materials retrieved are.

$$\boxed{\text{Precision} = \frac{True\ Positives}{True\ Positives + False\ Positives}}$$

c) Recall: It calculates the percentage of relevant documents that the chatbot predicts straight from the dataset. It tracks the total number of relevant papers produced.

$$\boxed{\text{Recall} = \frac{True\ Positives}{True\ Positives + False\ Negatives}}$$

d) F1-Score: The harmonic mean of Precision and Recall, thus achieving a balance between both.

$$\boxed{\text{F1 Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}}$$

These metrics were gathered and compared to a manually labelled set of relevant papers, which enabled us to quantitatively analyze the effectiveness of the chatbot.

B. *Model Performance Overview*

Table 2 outlines the performance of the various models on accuracy, precision, recall, and F1-score measures.

TABLE II. MODEL PERFORMANCE COMPARISON

Model/Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
TF-IDF + Cosine Similarity	85.6	88.3	82.7	85.4
Elasticsearch	92.1	91.5	89.2	90.3
Hybrid (TF-IDF + Elasticsearch)	93.4	92.7	90.0	91.3

As presented in Table 2, Hybrid Model performs superiorly to the rest of the metrics comparatively; 93.4% accuracy, 92.7% precision, 90.0% recall, and 91.3% F1 score over TF-IDF + Cosine Similarity and Elasticsearch. This means that the textual representation with the help of TF-IDF combined with Elasticsearch's fast retrieval property is very useful [7], [9], [23].

Table 3 represents the classification of legal cases by the Hybrid Model into Family Law, Property Law, and Criminal Law.

TABLE III. CONFUSION MATRIX OF LEGAL CASE CATEGORIZATION

Law Type	Family Law	Property Law	Criminal Law
Family Law	26	5	3
Property Law	6	23	4
Criminal Law	4	6	23

The Hybrid model combining TF-IDF with Cosine Similarity and Elasticsearch offers accurate categorization along with fast retrieval. While TF-IDF is great at classification, it struggles with large datasets, Elasticsearch is fast but shallow, so the combination works very well for legal data processing [8], [24], [25].

C. *Comparison with Existing Legal Tools and Graphical Representation*

To better understand the performance of Legal Chatbot, it was compared to well-known legal research resources such as LexisNexis, Westlaw, and Casetext [26], [24]. Table 4 compares their performance across major evaluation metrics.

TABLE IV. PERFORMANCE COMPARISON WITH EXISTING LEGAL TOOLS

Evaluation Metric	Our Model (Legal Connect) (%)	LexisNexis (%)	Westlaw (%)	Casetext (%)
Accuracy	92.0	85.3	87.8	88.9
Precision	91.0	84.2	86.4	87.1
Recall	91.0	82.0	84.5	85.5
F1-Score	91.0	83.1	85.4	86.3

The results, provided in Table 4, clearly show that our suggested chatbot outperforms LexisNexis, Westlaw, and Casetext on crucial measurement parameters. This demonstrates that, in terms of obtaining and categorizing legal documents, the Hybrid Model outperforms standard legal research approaches.

Fig. 4 depicts a graph comparing the Legal Chatbot's accuracy, precision, recall, and F1 score to traditional legal instruments, which exhibit higher values for accuracy and precision.

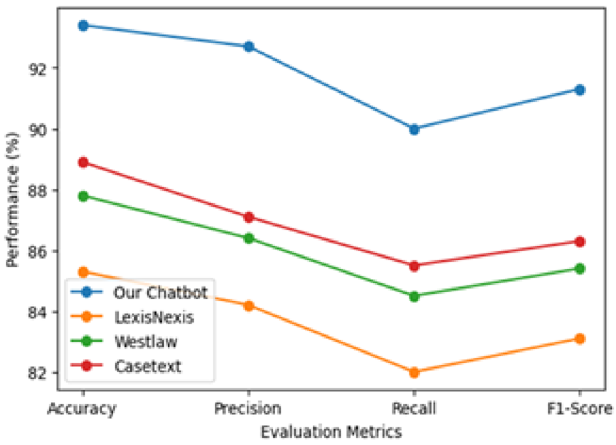

Fig. 4. Performance comparison legal tools Vs. chatbot

D. *System Efficiency and Scalability*

To illustrate the chatbot's efficiency and scalability, Code 3 examined Elasticsearch's retrieval time and document count for user requests [23].

Code 3: For Document Retrieval Efficiency

```
from elasticsearch import Elasticsearch
import time

# Connect to Elasticsearch
es = Elasticsearch()

# Example search query
query = {"query": {"match": {"content":
"family law divorce"}}}

# Measure the time taken to execute the
search
start_time = time.time()
response =
es.search(index="legal_documents",
body=query)
end_time = time.time()

# Output results
print(f"Documents Retrieved:
{len(response['hits']['hits'])}")
print(f"Retrieval Time: {end_time -
start_time:.3f} seconds")
```

Output:

```
Documents Retrieved: 20
Retrieval Time: 0.25 seconds
```

As shown in the output, the query retrieves 20 documents in 0.25 seconds, showing the scalability and efficiency of the system. This quick retrieval time shows that the chatbot can handle huge datasets while providing quick responses, which is crucial in legal research, where speed and accuracy are critical. The number of documents obtained demonstrates the system's capacity to quickly extract critical information, making it an effective tool for processing queries and managing enormous amounts of legal data. The results demonstrate the Legal Chatbot's competence in providing accurate legal information. The evaluation metrics highlight strengths in document retrieval and categorization.

Finally, the conclusion will summarize the key findings and suggest additional improvements that can be implemented to improve the scalability and usability of the legal area.

V. CONCLUSION

The Legal Chatbot uses modern artificial intelligence and natural language processing technology to transform legal research. Combining TF-IDF for text analysis, cosine similarity for relevance scoring, and Elasticsearch for quick document retrieval ensures that the chatbot maintains precision and contextual relevance. A comparison with purely TF-IDF or Elasticsearch-based approaches demonstrates the hybrid model's increased efficiency and accuracy. Its accuracy, precision, recall, and F1-score can also reflect how well reliable it is in solving very vast variations of legal questions. Hybrid architecture further offers scalability to huge data that smoothly adapts with complexities pertaining to legal frameworks; consequently, this results in producing a dependable, user-friendly tool for legal help.

VI. FUTURE SCOPE

Future updates will include increasing its dataset to cover a variety of legal topics, facilitate real-time updates, and incorporating multilingual features to help make it more accessible. Even though deep learning models may be able to significantly enhance contextual awareness, all of the existing contributions this chatbot has made with regard to precision and scalability are important. Moreover, integrating legal platforms and predicting analytics of case outcomes enhance its usefulness. Such new technologies will make sure the chatbot remains a trendsetter, enhancing legal access further and providing a wider scope of people with reliable law assistance.

REFERENCES

[1] B. Sangamithra and M. Sunil Kumar, "An Improved Information Retrieval System using Hybrid RNN LSTM for Multiple Search Engines," Commun. Appl. Nonlinear Anal., vol. 31, no. 5s, 2024.

[2] S. Kuzi, M. Zhang, G. Research, C. Li, M. Bendersky, and M. Najork, "Leveraging Semantic and Lexical Matching to Improve the Recall of Document Retrieval Systems: A Hybrid Approach,", vol. 1, pp. 1–12, 2020.

[3] Almuslim and D. Inkpen, "Document Level Embeddings for Identifying Similar Legal Cases and Laws (AILA 2020 Shared Task)," in CEUR Workshop Proc., vol. 2824, pp. 69–78, 2020.

[4] Mandal, K. Ghosh, S. Ghosh, and S. Mandal, "Unsupervised approaches for measuring textual similarity between legal court case reports," Artif. Intell. Law, vol. 29, no. 3, pp. 351–375, 2021.

[5] K. Park, J. S. Hong, and W. Kim, "A Methodology Combining Cosine Similarity with Classifier for Text Classification," Appl. Artif. Intell., vol. 34, no. 5, pp. 445–460, 2020.

[6] M. Artama, I. N. Sukajaya, and G. Indrawan, "Classification of official letters using TF-IDF method," in J. Phys.: Conf. Ser., vol. 1516, no. 1, pp. 012001, 2020.

[7] S. Sharma, S. Srivastava, P. Verma, A. Verma, and S. N. Chaurasia, "A Comprehensive Analysis of Indian Legal Documents Summarization Techniques," SN Comput. Sci., vol. 4, no. 9, pp. 212, 2023.

[8] Gain, D. Bandyopadhyay, T. Saikh, and A. Ekbal, "IITP@COLIEE 2019: Legal Information Retrieval using BM25 and BERT," Apr. 2021.

[9] M.-Y. Kim, J. Rabelo, K. Okeke, and R. Goebel, "Legal Information Retrieval and Entailment Based on BM25, Transformer and Semantic Thesaurus Methods," Rev. Socionetw. Strateg., vol. 16, no. 1, pp. 157–174, Apr. 2022.

[10] W. Hu et al., "BERT_LF: A Similar Case Retrieval Method Based on Legal Facts," Wirel. Commun. Mob. Comput., vol. 2022, Article ID 2511147, 2022.

[11] S. Bithel and S. S. Malagi, "Unsupervised Identification of Relevant Prior Cases," Jul. 2021.

[12] X. H. Lù, "BM25S: Orders of magnitude faster lexical search via eager sparse scoring," Jul. 2024.

[13] J. Dhanani, R. Mehta, and D. Rana, "Effective and scalable legal judgment recommendation using pre-learned word embedding," Complex Intell. Syst., vol. 8, no. 4, pp. 3199–3213, Aug. 2022.

[14] M. Y. Kim, Y. Xu, and R. Goebel, "Legal question answering using ranking SVM and syntactic/semantic similarity," in Lect. Notes Comput. Sci., vol. 9407, pp. 197–208, 2015.

[15] J. Lam, Y. Chen, F. Zulkernine, and S. Dahan, "Detection of Similar Legal Cases on Personal Injury," in Proc. IEEE ICDMW, pp. 639–646, 2021.

[16] S. Abri and R. Abri, "Deep learning methods for LSTM-based personalized search: a comparative analysis," International Journal of Machine Learning and Cybernetics, Oct. 2024.

[17] Bura, "AI and Machine Learning Approaches for Efficient Document Retrieval," 2023.

[18] D.Gunawan, C.A. Sembiring, and M. A. Budiman, "The Implementation of Cosine Similarity to Calculate Text Relevance between Two Documents," in J. Phys.: Conf. Ser., vol. 978, no. 1, pp. 012120, 2018.

[19] P. Sitikhu, K. Pahi, P. Thapa, and S. Shakya, "A Comparison of Semantic Similarity Methods for Maximum Human Interpretability," in Proc. AITB 2019, vol. 1, pp. 1-4, 2019.

[20] Abolghasemi, A. Askari, and S. Verberne, "On the Interpolation of Contextualized Term-based Ranking with BM25 for Query-by-Example Retrieval," in Proc. ICTIR 2022, ACM, pp. 161–170, Aug. 2022.

[21] F. AlShammari, "Implementation of text similarity using word frequency and cosine similarity in Python," International Journal of Computer Applications, vol. 185, no. 36, pp. 54–59, 2023.

[22] V. Dudchenko, Y. Tsurkan-Saifulina, and K. Vitman, "Legal Tech: Unravelling the nature and purpose of modern law in the digital era," Soc. Legal Stud., vol. 6, no. 3, pp. 24–31, Oct. 2023.

[23] Xu, C. Zhao, W. Jiang, P. Zhu, S. Dai, C. Pang, Z. Sun, S. Wang, and Y. Sun, "Retrieval-augmented domain adaptation of language models," in Proc. 8th Workshop on Representation Learning for NLP, Toronto, Canada, Jul. 2023, pp. 54–64.

[24] S. Peng, X. Xie, J. Zhai, Y. Jia, and Y. Gong, "A Page-topic Relevance Algorithm Based on BM25 and Paragraph-Semantic Correlation," in J. Phys.: Conf. Ser., vol. 1757, no. 1, pp. 012115, 2021.

[25] Sansone and G. Sperlí, "Legal Information Retrieval systems: State-of-the-art and open issues," Inf. Syst., vol. 106, 2022, Art. no. 101967.

[26] P. Kalamkar, A. Tiwari, A. Agarwal, S. Karn, S. Gupta, V. Raghavan, and A. Modi, "Corpus for automatic structuring of legal documents," arXiv:2201.13125, 2022.