Table of Contents

Abstract

Lawyers and legal professionals carry out research to build strong cases, provide accurate legal advice and stay informed with legal changes. Digital databases of legal information have allowed for improved efficiency and accessibility with quicker searches and remote access. However, the vast amount of data can be overwhelming, requiring strong research skills to find relevant references. This paper proposes an application to mitigate this issue, by providing enhanced assistance in legal research through data enrichment with High Performance Computing Cluster (HPCC) Systems. The proposed application leverages Natural Language Processing (NLP) to extract keywords from a case abstract entered by the user by using Named Entity Recognition (NER). The NER model, trained on an open-source dataset curated by data scientists and legal experts at

OpenNyAI, can recognize up to 12 different legal entities in any given abstract, which are used to search for case statements from a main cases dataset. The model achieves state-of-the-art performance with precision, recall and F1 scores of 0.92, 0.90 and 0.91. The keywords and phrases are then used to search the Indian Legal Documents Corpus (IDLC) dataset, which is sprayed on a HPCC Systems cluster. HPCC Systems is leveraged here to improve the efficiency of the searching process due to its distributed architecture, allowing for parallelism. The user interacts with the application through a web interface, and after search and retrieval, the relevant references are displayed to the user through the same.

Keywords

Big Data, Natural Language Processing, Named Entity Recognition, Distributed Computing, HPCC Systems, Legal Research, Web Application

Introduction

The legal profession is fundamentally based on comprehensive and precise research as a backbone of effective practice. For lawyers, judges, researchers, and scholars, finding case documents relevant to their present cases is essential. Such research, which in the previous times involved a careful study of physical legal texts and records, may take weeks or even months, thereby delaying the pace of case preparation. The Indian legal system is one of the biggest jurisdictions around the world and poses an especial challenge, as over 30 million cases are pending in courts across the country, which creates an unprecedented amount of legal data. This massive collection of case law, growing every year by tens of thousands of judgments from the Supreme Court, High Courts and District Courts, poses a challenge to legal practitioners in finding case documents as soon as possible with maximum possible accuracy.

The digital revolution has affected legal practice in many ways, but the move from physical to digital records has created issues in itself [1,2]. Increase of digital records in law adds a paradoxical twist: though more accessible, it also creates a challenge for the retrieval of documents. Lawyers have to wade through massive digital repositories to retrieve only that case precedent relevant to them. The Indian context is more pronounced in this challenge, due to the sheer volume of case law, complexity of legal language, and diversity of jurisdictions, making traditional search methods inefficient and time-consuming. The repercussions add more complications in terms of document retrieval regarding the intricacy of legal terminology and judicial nuances.

Big Data has become a central technology within the legal sector, offering sophisticated capabilities to address the complexities inherent in managing, analyzing, and deriving actionable insights from extensive legal datasets [7]. Defined as the processing and analysis of massive, high-dimensional data that exceeds the capacity of traditional

computational methods, Big Data is instrumental in handling the scale, speed, and diversity of legal documentation and case records, especially in the Indian context. By employing advanced methodologies in artificial intelligence, data mining, machine learning, and predictive analytics, Big Data enables a systematic exploration of case patterns, enhances predictive accuracy, and fosters deeper insights, thereby improving precision, operational efficiency, and accessibility in legal research and informed decision-making.

Natural Language Processing (NLP) is a subarea of artificial intelligence that has become an important technology that can be used to overcome challenges in document retrieval and analysis in the legal sector [3]. NLP refers to computational methods that allow machines to process, analyze, and understand human language in meaningful ways. In the legal domain, NLP techniques are particularly valuable because they can handle the complexity and nuances of legal text. These methods employ sophisticated algorithms capable of outscanning the legal documents more effectively for matching user queries than searching with keys and words.

In this paper, we propose an application that leverages the capabilities of Big Data solutions and NLP to allow users to search for legal documents more efficiently. The purpose of the application is to allow users to find legal references based on a case abstract that is pertinent to them, providing quicker search and retrieval times. The application employs NLP to extract and tag key words/phrases from a case abstract entered by the user. The words and phrases are then used to search for similar references with the Indian Legal Documents Corpus (IDLC) dataset. To make the process more efficient, we utilize the distributed processing capabilities of High Performance Computing Cluster (HPCC) Systems for the search and retrieval process.

Named Entity Recognition (NER) is an applied part of the solution, employed to perform the key word/phrase extraction. This is a technique based on NLP and the application uses a pre-trained spaCy model fine-tuned over a huge legal corpora. This is done to make the key word/phrase extraction more domain specific and prevent non-legal terms from being identified as keywords. HPCC Systems is the computational backbone of the application. It is an open-source, data-intensive computing system platform with several key advantages for processing large-scale legal data. HPCC Systems uses a distributed architecture where massive datasets can be parallelly processed across multiple nodes and reduces the time taken for document retrieval. The Enterprise Control Language (ECL) is the platform's programming language developed to process big data in an efficient manner, which enables the manipulation and search operation on data. The complex processing of data is managed by HPCC's Thor cluster and rapid delivery of data to satisfy real-time queries for quick document retrieval is offered by its Roxie cluster.

The application addresses the problem of tedious searching and document retrieval by making novel use of these technologies. The system begins with processing

user-submitted case descriptions through a specialized NER model, which identifies and extracts relevant legal entities. These entities are then used to construct sophisticated search queries that leverage HPCC Systems' distributed computing capabilities. The application employs a multi-stage processing pipeline: first, the NLP components analyze and structure the input; then, HPCC Systems executes parallel searches across its distributed database; finally, the results are ranked using a relevance algorithm that prioritizes the most pertinent documents. The users interact with a React-based web application, and the NER model is run as an independent service used for inferencing. The use of HPCC Systems is achieved using a custom API layer, optimizing query execution and retrieving results.

The paper describes the procedure and results of development of the described application and is divided into three sections. Section I is a literature survey, describing the research works and applications reviewed. Section II focuses on the methodology and features of building the application. Section III focuses on results and discussions and lastly Section IV provides the conclusions and future work.

Literature Survey

Legal research involves finding pertinent legal information to strengthen arguments and rulings in court cases. Lawyers have traditionally carried out this process by hand, which can be laborious and prone to human error. By automating some processes and improving the speed and accuracy of the results, artificial intelligence (AI)—particularly machine learning (ML) and natural language processing (NLP)—has the potential to revolutionize legal research.

Bhupatiraju et al. [1] discussed how the digitization of legal information in India, initiated by organizations like IndianKanoon.org significantly enhanced the capabilities of Indian lawyers. Platforms having a centralized and searchable database of court cases have streamlined legal research, enabling lawyers to find relevant precedents more efficiently.

Maheshwari [2] explained that digital records of data have significantly improved accessibility and efficiency, as it eliminates the conventional barriers separating the public from judicial records. Digital records increase public confidence in the legal system while improving security and having a smaller environmental impact. One advancement in the modernization of the legal system is the incorporation of AI and big data technologies.

Tung [3] explores the ways in ML is improving decision-making processes and revolutionizing the legal profession. The paper emphasizes how ML can analyze large datasets, identify patterns, and forecast case outcomes more accurately than with conventional techniques. In addition, issues like bias, data privacy, and the requirement for algorithmic transparency are covered in the paper. It highlights how crucial it is to incorporate ML tools in order to enhance legal research, expedite case management, and support well-informed legal strategies—all of which will maximize client outcomes.

Vinay et al. [4] analyzes the potential impact of using NLP in the field of legality. The article also brings up the subject of Indian languages and their impact on legal documents and NLP, as every court might follow more than one language in their documents. It also discusses how the progress made in the field of NLP would help the process of drafting and interpretation of legal documents with high speeds, precision and volume.

Modi et al. [5] explores the impact of natural language processing (NLP) on legal research and how it extends from simple document classification and case summary to predictive analysis. Further, in the context of an increasingly complex legal data universe, it is seen that NLP assists in evaluating legal issues many times faster and more accurately. However, many challenges were also identified including bias, shortage of benchmark data and privacy of user data. NLP was said to have the potential to redefine provision of legal services in a radical way but it also presented fears of job redundancy and traditional billing mechanisms. It was concluded that the sector is headed towards embracing technology so as to be more effective and more efficient in delivering justice.

Caballero et al. [6] addresses the growing demand of NLP for legal services due to adaptability to increased complexity and volume of legal texts. The review classifies studies into categories such as Multiclass Classification, Summarization, and Question answering further explaining the present challenges faced and emphasizing the importance of properly sourced and ethical datasets. It offers immensely useful and vital information to the advancement of Legal NLP studies and its utilization in the legal environment.

While ML techniques can be useful to analyze, understand and interpret legal texts, the massive volume of data available can still pose significant challenges to legal research, primarily related to the management and searching of such large repositories. Thillaieswari et al. [7] has articulated the challenges and technologies relevant in managing such large quantities of data. It described big data characteristics such as volume, variety, velocity, value, and veracity and further explains the architectural framework of big data systems that can be used to manage this data, with particular interest in the role of Hadoop, HDFS, and MapReduce. Finally, the study discusses a range of technologies implemented for handling big data: Hive, NoSQL, and HPCC. Security and integrity of data are especially underlined in big data management within the paper.

Devins et al. [8] discussed how big data can provide a scientific and evidence approach to law, and can contribute to behavioral optimization and the personalization of law where analysis and prediction using data can generate a more client-tailored legal experience. The article also, however, cautions against how big data and its methods fundamentally differ from the current principles of legal practice and that one should be cautioned against its over-optimistic use.

Fina [9] addresses the ways in which big data analytics can improve the legal sector, specifically in the area of litigation. The benefits of employing big data tools to track cases and predict the possibility of success in appeals is highlighted, empowering clients to decide whether to pursue or drop a case with greater knowledge. It also emphasizes how big data can enhance legal research and case preparation, which could result in better client outcomes. Ződi [10] examines the ways in which legal practice and research are being altered by big data. The integration of data analytics into legal processes is covered, including how to improve decision-making, forecast case outcomes, and optimize legal tactics. The study also discusses obstacles, such as the need for new legal frameworks to regulate the use of big data in the legal system and worries about data privacy and ethics. The potential of big data to enhance legal transparency, efficiency, and access to justice is discussed.

Thus, it can be concluded that the use of NLP and Big Data techniques can significantly aid and enhance research methods and processes in the legal domain. One of the fundamental challenges of legal research involves automating the process of finding important legal terms in a large body of text. This could be useful for summarization, document analysis and reference finding. One of the fundamental challenges of legal research involves automating the process of finding important legal terms in a large body of text. This could be useful for summarization, document analysis, and reference finding. NLP can help in this regard by employing techniques such as keyword extraction and Named Entity Recognition (NER), which can identify and categorize relevant legal entities, concepts, and themes. Such automation not only accelerates legal research but also reduces the risk of overlooking critical information, ultimately enabling legal professionals to make more informed decisions based on comprehensive and systematically analyzed data

Keyword extraction is a class of NLP techniques, where algorithms are utilized to classify words in a text into primarily two categories - keyword and non-keyword. Keywords hold more importance as compared to other words in the text and can convey the gist of what the text contains in its entirety. There are several different methods for keyword extraction but the approach primarily focuses on two algorithms - supervised and unsupervised machine learning algorithms.

Baruni et. al [11] investigates the use of Rapid Automatic Keyphrase Extraction (RAKE) in extracting keyphrases from a document. It is discussed that RAKE, an unsupervised algorithm, uses co-occurrence graphs to track word associations within candidate phrases, and assign scores to each word in the text. The top third of the phrases are then displayed as keywords. It is seen that RAKE outperforms other algorithms such as the TextRank algorithm. Issa et. al [12] discussed keyBERT, another unsupervised technique, for keyword extraction. The study also performed a comparative study between the different embedding models in keyBERT and found that Paraphrase-mpnet-base-v2 model provides the best results among all other models.

Yao et. al [13] discussed a keyword extraction approach using TF-IDF and TextRank to extract keywords by constructing a word graph model and counting word frequency and inverse document frequency. The results indicated that integrating the TF-IDF and TextRank algorithms yields significantly superior performance and extraction effectiveness compared to traditional methods.

Another method for identifying important terms in text is Named Entity Recognition (NER). NER is formally defined as a component or subtask under the vaste umbrella of NLP algorithms and it is used to identify objects in a text that belong to a certain set of predefined categories. Jehangir et al. [14] conducted an analysis of different methods for implementing a NER system, with methods ranging from unsupervised, rule-based, supervised and deep learning methods. The study discussed various toolkits and packages that are used for NER such as SpaCy, Natural Language Toolkit (NLTK) and PyTorch as well as approaches to implementing an NER model. The possible limitations of NER were also highlighted, such as the difficulty of dataset preparation for domain-specific NER and noisy and misspelled input text leading to incorrect outputs.

Schmitt et al. [15] performed a comparative study between different NER toolkits such as NLTK and Stanford NLP. As a result of the comparative study, Stanford NLP was found to outperform all other methods including NLTK in most regards, but the degree varied drastically across different datasets and even across different tags. The study, however, did not cover performance on the basis of time taken for classification or ease of implementation.

Shelar et al. [16] performed a similar comparison between several NER approaches such as spaCy, TensorFlow and Apache OpenNLP for customizing a pre-built NER model which these libraries provide. The study concluded that training accuracies, loss F1-score and prediction probability of spaCy and OpenNLP outperformed those of TensorFlow, with spaCy holding a small advantage over OpenNLP as well in terms of prediction probability. The spaCy model also took less time as compared to the OpenNLP model for prediction but the model size was significantly larger. Kalamkar et. al [17] introduced a corpus of a corpus of 14444 Indian court judgment sentences and 2126 judgment preambles annotated with 14 legal named entities, in which they used spaCy to predict legal named entities where spaCy defined entities were mapped to pre-defined legal named entities. A transformer-based legal NER model using spaCy was also proposed and was made available as a spaCy pipeline.

Akbik et al. [18] proposed FLAIR, an NLP framework designed to simplify training and deployment of language models, specifically for tasks like sequence labeling and text classification. FLAIR enables combining classic word embeddings (like GloVe) with contextual embeddings, facilitating the creation of high-performing NER models. Its "model zoo" includes pre-trained NER models, making it easy for users to apply or fine-tune these models for language-specific or multilingual entity recognition tasks, streamlining the NER process significantly. ElDin et al. [19] demonstrated the use of FLAIR in Med-Flair, an NER tagger application focused on identification of entity types,

medications and diseases in Electronic Health Records (EHR). It achieved state-of-the-art performance on 4 benchmark datasets.

Following the key word/phrase extraction, the next key challenge lies in the searching process. The searching process must be designed so that it is quick and efficient and yet thorough so that no valid and important references are missed. Glavic et al. [20] explores the formidable challenge of indexing and searching unstructured data in distributed systems and presents FusionDex, a system that could achieve much better performance and scalability without global coordinators and with minimal data movement. The proposal further introduces the SCANNS framework, which is designed for high-performance indexing in many-core architectures, and also explores the integration of machine learning for metadata generation in scientific data search.

Locke et al. [21] discussed how tags are used to improve the retrieval of relevant content within legal documents, focusing particularly on relevance and precision. The paper suggests that tags can help highlight key paragraphs or segments, enabling retrieval at a more granular level (i.e., paragraph-level) rather than retrieving entire documents. The tags can also support advanced retrieval models, such as learning-to-rank algorithms and neural retrieval systems, by reducing the influence of irrelevant text that typically surrounds critical content in lengthy documents. By isolating relevant segments, tags can assist in "intra-document weighting," where tagged sections receive higher importance, enhancing the model's accuracy in ranking results. This refinement helps the ranking function prioritize tagged content over non-tagged sections, thus improving the efficiency and relevance of legal document retrieval.

Xiao et al. [22] describe Similar Case Matching (SCM) as an approach that applies semantic text matching to assess similarity between legal case documents, a crucial function in legal information retrieval. Semantic text matching, central to various NLP tasks like question answering, information retrieval, and natural language inference, enables the model to gauge the semantic similarity between a query and database documents. In SCM, this helps in determining which cases are most relevant by comparing the semantic content of cases, a process that has gained considerable interest among researchers for its applications across multiple domains.

Vacek et al. [23] demonstrated that their ensemble of machine learning models, combining SVM and deep learning methods like CNN-GRU and Nested-GRU, is highly effective for data retrieval within the legal domain. By targeting case outcome classification, these models allow for more accurate extraction of relevant case information, supporting precise retrieval of legal outcomes essential for tasks in litigation analysis and legal research. Custom word embeddings, based on Google’s word2vec, further refine these retrieval capabilities by enhancing the model’s understanding of legal text nuances, making the approach particularly valuable for focused legal data retrieval.

In summary, NLP and Big Data holds immense potential to be used for building applications in the legal domain. While previous studies have extensively examined the

issues with legal research and the use of NLP and Big Data in legal research separately, there remains a major scope for the cross-integration of these technologies to create a more robust solution to make legal research easier and more efficient. This paper seeks to complement this by proposing an application that utilizes these technologies to create an efficient legal research application, the key objective of which is to allow users to find legal references quickly. The following section outlines the methodology and work.

Proposed Methodology and Work

The workflow of the proposed application consists of three primary phases. Firstly, the user enters a case abstract from which relevant legal key words/phrases are extracted using an NER model. Secondly, these words/phrases are used to query a database of legal documents to find references based on matching keywords within the documents. Lastly, the documents are ranked based on number of keywords matched and relevancy of keywords matched, which are then displayed to the user. To allow for more accurate key word/phrase extraction, NLP techniques were employed. As the corpus of data being searched can be very large, HPCC Systems has been used to allow for parallel processing of the data, leading to more efficient searching and retrieval. The features of the application have been discussed below.

Extracting Important Legal Terms from Text

The first step in the application workflow is to extract important legal terms from a case abstract entered by the user, which is to be used during the subsequent searching process. Two different techniques were tested for the same - keyphrase extraction and NER. Different approaches and models for both techniques were compared to conclude on the most suitable approach for implementation. In order to ensure consistent evaluation conditions, the same dataset was used across all methods. The dataset used was developed by legal experts and data scientists at OpenNyAI [17], of which only the training split was used, and has 9442 data entries in JSON format. Each entry has a unique ID, the text on which extraction is performed and the extracted keywords along with its associated entity. If the method required training, the original training dataset itself was split in an 80-20 ratio for training and testing. For different methods, the dataset format requirements may have differed, and the suitable conversions were done on the dataset before testing.

- **Keyphrase Extraction**

Keyword extraction is an NLP technique dedicated to the systematic identification and extraction of important terms or phrases that represent the primary content within unstructured textual data. This methodology can be advantageous in specialized fields such as legal research, where identifying important legal terms can allow for better case analysis, reference extraction, and efficient information retrieval by emphasizing critical legal concepts and references. Two unsupervised techniques, namely Rapid Automatic

Keyword Extraction (RAKE) and keyBERT and two supervised techniques, namely Term Frequency-Inverse Document Frequency (TF-IDF) with logistic regression and a Bidirectional Long Short-Term Memory (BiLSTM) neural network model, were tested on the dataset.

RAKE

It is a straightforward, unsupervised algorithm that is very domain-independent and automatically attempts to discover keyphrases in a body of text. In contrast to other keyword extraction techniques, RAKE doesn't require any training. It is very helpful in applications involving text mining, summarization, and information retrieval when it is crucial to quickly identify important concepts. RAKE-NLTK is a Python implementation that enables keyword extraction in Python-based projects by leveraging the Natural Language Toolkit.

On testing it on the dataset, RAKE only picked a few legal keywords because it is not a dedicated legal keyword extractor.It did however, extract a lot of other keywords, most of which are related to the topic of the paper but not relevant to its legal context, and hence has low precision and total score.

Key-BERT

The second unsupervised method tried out during this study was Key-BERT. KeyBERT is a robust and context-aware keyword extraction technique and applies BERT (Bidirectional Encoder Representations from Transformers) embeddings to extract the best possible keywords in contexts from text. It uses transformer-based models giving KeyBERT a strong edge over older techniques that were based on frequency and co-occurrence instead of understanding the relationships between words in context. For each phrase, the similarity score is calculated against the document embedding; that is, how well the phrase captures the essence of content. Keywords with higher scores are more relevant since these words are closer to themes within the text. Unlike the other traditional algorithms, which would extract general or irrelevant keywords, KeyBERT returns phrases closer in meaning to the actual document, therefore relevant.

TF-IDF with Logistic Regression

In this approach, we used a combination of TF-IDF vectorization and logistic regression classification for identifying a keyword within a document. Data preprocessing had to be done, where the dataset was parsed to tokenize the text input and assigned binary labels based on keyword annotations (1 for keywords and 0 for non-keywords) provided in the dataset. Following preprocessing, the document was transformed into TF-IDF vectors, in which the importance of each term relative to the document and corpus is found by calculating the frequency of

terms within the document against their distribution across all documents. To capture word-level classification features, the TF-IDF representation was extended by incorporating document-level features and position-based information, providing additional context for each word in the document. For model training, logistic regression was used on these features to classify words as either keywords or non-keywords.

This method significantly outperformed the unsupervised techniques and was able to distinguish between important legal terms in the text as compared to other words that may have held importance in a more general context.

BiLSTM Model

We also tested out a deep learning approach using a BiLSTM model. A similar data preprocessing step was carried out as in the TF-IDF approach. To ensure a consistent input format, tokenized sequences were padded to a fixed length and converted to word embeddings. These embeddings were fed into a BiLSTM model, which processes the text in both forward and backward directions, allowing the network to learn contextual relationships between words within the sequence. The BiLSTM model outputs a probability score for each word, indicating its likelihood of being a keyword. A binary threshold is applied to these probabilities to classify each word. Since the purpose was only binary classification, the threshold was set to 0.5. The model was trained using binary cross-entropy loss, with early stopping and dropout layers to prevent overfitting.

This approach also significantly outperformed the unsupervised methods, but did not perform as well as the TF-IDF approach.

In order to evaluate and compare each model, the macro average precision, recall and F1-scores of each model was calculated. These scores are shown in Table 1. The supervised methods outperformed the unsupervised techniques by a large margin. While supervised keyword extraction methods did provide somewhat satisfactory results, it still did not work with the level of accuracy our application required. Moreover, as described later in this section, the use of just keywords without any entity tagging made searching and ranking more tedious.

Method	Precision	Recall	F1-score
RAKE	0.18	0.11	0.07
KeyBERT	0.05	0.22	0.07
TF-IDF with Logistic Regression	0.58	0.71	0.56
BiLSTM	0.51	0.44	0.47

Table I

- **Named Entity Recognition (NER)**

A branch of NLP, NER is concerned with finding and categorizing important textual data into pre-established groups, such as names of individuals, companies, places, dates, and context-relevant terms. By labeling these named entities after parsing unstructured text data, NER systems convert unstructured textual data into structured information. This procedure is especially useful in a variety of fields, including legal research, where tasks like data analysis and information retrieval depend on the identification of entities such as statutes, judicial bodies, and case names. Three packages were tested and compared, namely spaCy, FLAIR and NLTK.

SpaCy

SpaCy is an NLP package with various capabilities including NER, and can process large amounts of text in a relatively shorter amount of time. SpaCy has pre-trained models that inherently support the identification of 18 different tags in NER-based use cases. These models can also be fine-tuned on data organized in a specified format, so as to make its use more domain-specific. Several models fine-tuned using spaCy are also available as a part of the spaCy pipeline, and we tested one such model as described in [17].

FLAIR

FLAIR is another NER package built on top of PyTorch and trained on a very large corpora of unlabelled text. Like spaCy, the pre-trained model can identify up to 18 different tags and can also be finetuned on domain specific data. Result of FLAIR testing (accuracy and training time)

NLTK

NLTK refers to a set of libraries used in Python, and can be used to perform NER by following a two-step methodology - part-of-speech tagging and chunking. Unlike spaCy and FLAIR, NLTK can inherently identify only 3 tags. Two model types were tested, namely Naive-Bayes and Decision Tree models. For both, the dataset was divided into 80% training data and 20% testing data. The Naive Bayes classifier achieved a testing accuracy of 78.85% whereas the Decision Tree classifier achieved an accuracy of 89.40% on the test set.

The precision, recall, F1-score and inference time was computed for each package, which are shown in Table II.

Package	Precision	Recall	F1-score
NLTK (Naive-Bayes)	0.90	0.79	0.83
NLTK (Decision Tree)	0.89	0.89	0.89
FLAIR			
spaCy	0.92	0.90	0.91

Table II

The spaCy model outperformed all other NER techniques, showing higher precision, recall and F1-scores.

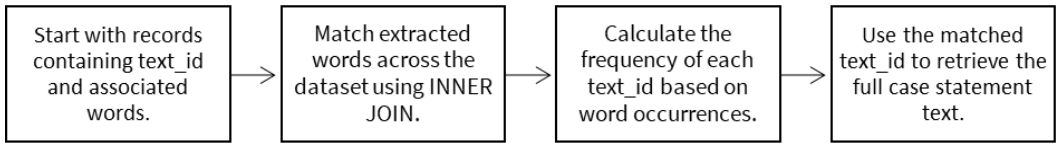
Searching for References using Retrieved Keywords

The next step in the workflow is to utilize the keywords and phrases to search a legal corpus for references, using HPCC Systems to expedite the search. The dataset that is being searched for references is the Indian Legal Documents Corpus (IDLC) dataset. The IDLC dataset is a comprehensive collection of legal documents from various courts and legal institutions across India, designed to facilitate research and development in legal NLP. This dataset consists of a wide range of legal texts, including case judgments, legal statutes, and contracts, making it a valuable resource for the legal tech community. Its primary purpose is to enable the development of NLP models that can analyze, process, and extract relevant information from legal texts. Given the complex nature of legal language and the volume of legal data, the IDLC dataset plays a crucial role in advancing tasks such as legal document summarization, named entity recognition, case outcome prediction, and legal information retrieval. It is reliable, well-structured and open source. The dataset is limited to cases related to the Supreme Court of India and this shortcoming can easily be rectified by using more diverse datasets. For the purpose of investigating and establishing a stable and usable workflow, we have, however, limited our use to the IDLC dataset due to its structured nature. The dataset is sprayed onto the HPCC Systems cluster and is used to search for references.

Data storage in HPCC Systems is primarily managed through its Thor Cluster, which distributes data across multiple nodes for efficient processing. Data is typically stored in flat files on a distributed file system, such as HDFS, and can be formatted in various ways, including CSV, JSON, or binary. This flexibility allows users to handle different types of data seamlessly. Retrieving data in HPCC Systems is streamlined through ECL queries, which enables users to specify what data they need without worrying about the actual mechanisms of data acquisition.

Within HPCC Systems, several search techniques were explored, including Sequential Searching, Dictionary-Based Searching, and Indexed Searching, each with its unique advantages and limitations in handling large-scale legal data. The keywords are first extracted from the main cases dataset. These keywords, along with their respective case statements, are stored in a separate dataset comprising around 1,200,000 keywords, using which the searching and mapping is carried out.

1. Sequential Searching

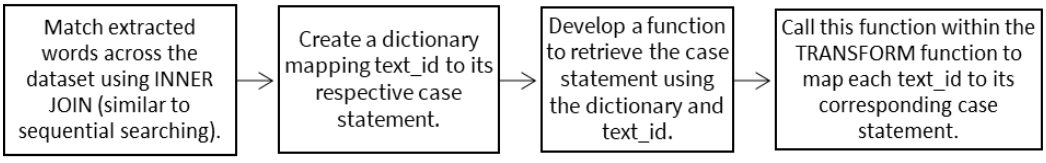


Sequential searching involves going through the record in the dataset in order starting from the first records and comparing with the search criteria. The process for this method is outlined as follows:

- This data set comprises text identifiers which are also called key texts in the given table (text_id) and the keywords related to them.
- An inner JOIN was employed to match the extracted keywords across the entire dataset.
- The frequency of each text_id is computed based on how often the corresponding words appear.
- Finally, the matched text_ids are used to retrieve the original case statements.

Although this method is easy to understand conceptually, it experiences vast inefficiencies especially when using very large data in the cases. Even with optimized query engines such as the HPCC Systems' ROXIE, each query takes time to scan the entire corpus and therefore incurs considerable time overheads. Hence it is not advisable to use this approach to solve large scale problems.

2. Dictionary-Based Searching

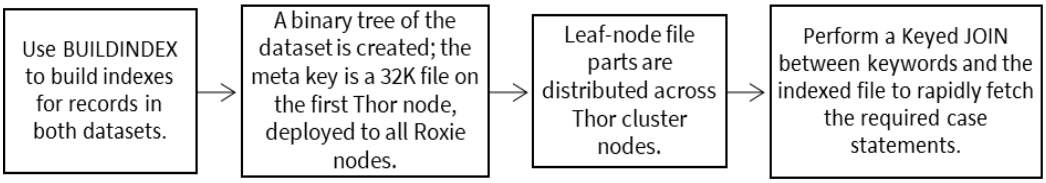


The dictionary-based search method enhances the search relevancy and effectiveness with the help of a key-value pair search method where the key is the ‘text_id’ and the value is the corresponding case statement.
Steps involved in dictionary searching include:

- Matching extracted keywords across the dataset, in the same manner as in the sequential method.
- The creation of the dictionary that will map each of the text_id to its corresponding case statement.
- Then implementing a function that would allow us to retrieve the case statement from the dictionary using the text_id.
- Finally employing the TRANSFORM function for mapping each text_ids to their respective case statements while querying to fetch all the relevant cases.

This method takes less time for lookup once the dictionary is created because case statements can be easily retrieved using text_id keys. However, re-creating the dictionary each time when a ROXIE query is executed consumes considerable time, particularly when dealing with large data. As a result, the dictionary-based method, while faster than sequential searching, is not suitable for large-scale dataset querying due to its time overhead.

3. Indexed Searching (Current Method)



To address the problems associated with the use of sequential and dictionary search methods, we used the Indexed Searching method. This method leverages indexed datasets to significantly reduce search times, making it more suitable for large-scale corpus querying. The indexed search process involved the following steps -

- Index Creation - HPCC Systems’ BUILDINDEX function is employed, which builds indexes for each record in the dataset created. The good thing about these indexes is that they build up a binary tree, thus making it easier to search for the information. Index files are further compressed to bring down their storage requirement in the disk system.
- Index Deployment - The binary-tree meta key, which is a 32K file part, is located on the first node in the Thor cluster and replicated on all nodes in

the Roxie cluster. This distribution ensures rapid query execution, so that the query is executed as fast as possible on the nodes involved.

- Keyed Join - These indexed datasets are then used in a keyed JOIN operation in order to extract the case statements that were linked to its index. When a dataset has been indexed, it allows for rapid access to the records associated with a specific key. Instead of scanning through the entire dataset for matching records (as is done in a standard JOIN operation), the system can quickly look up the relevant entries using the index. This significantly reduces the time complexity.

Searching Method	Time Taken (in seconds)
Sequential	44.225
Dictionary	19.660
Indexed	1.296

Table III

By utilizing the indexed search method, the search process was made much faster and more efficient. This method offered significant advantages over sequential and dictionary-based searches, making it the most suitable for large-scale data retrieval in HPCC Systems. The ten most relevant references are extracted using a latency of about 1.7 seconds. The reference titles, along with a short summary for each, is sent as a JSON response back to the proxy server.

Integration with Web Application

For building the web application, we used the React framework. React is a JavaScript library for building dynamic user interfaces with reusable components. Its virtual (Document Object Model) DOM efficiently updates only the changed parts of the user interface, making it fast and ideal for single-page applications. The frontend has an easy to use interface where the user can enter in a case abstract that's relevant to them. Important legal terms are then extracted from the text using the NER model. These key words/phrases are used to search for references, and after retrieval, the references are displayed to the user.

The backend consists of the Application Programming Interface (API) based on Flask and is equipped with NLP functions and connects to the Roxie server through a proxy server. The backend is organized into three primary entities. The first is the NLP endpoint which is written in Python and based on the Flask web framework, which takes user-entered text as input and extracts corresponding key words/phrases. The second is

the proxy server which acts as a bridge between the front end and the Roxie server. After key words/phrases are obtained from the front end, the proxy server receives them and transmits them over to the Roxie endpoint. The proxy server also receives the JSON response from Roxie and forwards the response data to the front end. The third is the Roxie server interaction. Through the use of the keywords, the Roxie server is able to query the relevant legal cases. These cases are processed, ranked and returned to the user in JSON format. are most similar to the keywords and phrases in the case are retrieved and ranked. These cases are ranked in order of importance starting with the one that has the highest similarity of the key words to the case in question, which is then displayed to the user.

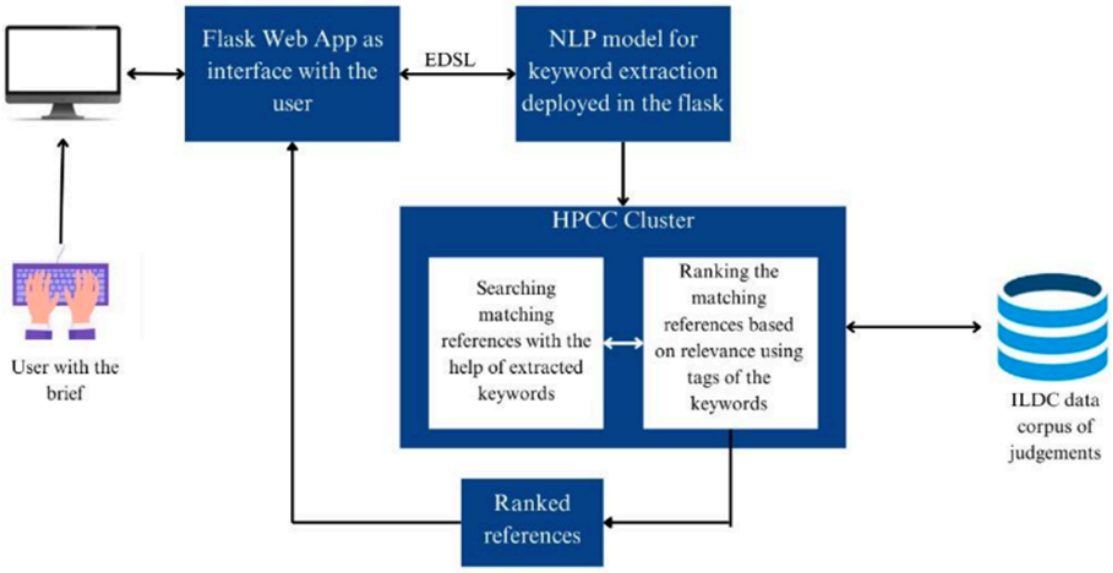


Figure 1

Results and Discussions

The application detailed above can be utilized to simplify legal research, with the major advantage of faster searching times for references based on case abstracts pertinent to the user. The entire application workflow is shown in Figure 1.

The application is developed in such a way that it helps to search and retrieve legal cases by entering some pieces of text or specific keywords. The application includes two basic user input entry points, namely raw text entry, where users can enter text directly and submit it for keyword extraction and keyword entry, where users can input specific keywords to initiate a targeted search directly. If the user chooses to enter in text, the keywords are extracted using NER and are to the user, who has the choice to delete or add more keywords as per their discretion, as displayed in Figure 2. The user

is then provided with a ‘Send Request to Roxie’ option that allows the keywords to be sent to the backend Roxie server to retrieve relevant cases. After the searching, only a maximum of 10 cases that Every search result consists of a headline and an abbreviated statement containing the essence of the case providing an interactive output view. For those users who require more information, a ‘View More’ option is provided to access the extended version of the case completing the details,as displayed in Figure 3.

We also performed a comparative analysis between HPCC Systems, Hadoop and a custom multithreaded Python application to analyse and ratify our approach. Hadoop and HPCC Systems are designed to process huge amounts of data, but they come with different conceptions. Hadoop has been designed using the MapReduce programming style and executes in distributed clusters of commodity hardware, which makes the platform highly scalable and fault-tolerant. It splits data into blocks and then executes them parallel amongst the nodes. Therefore, it could be highly effective for those jobs which involve bulk storage and processing of data. However, since Hadoop relies on disk I/O to store intermediate data, this could cause delay if the datasets are non-indexed.

By contrast, HPCC Systems is designed to handle data in real time. The architecture of HPCC Systems is based on in-memory processing, which reduces latency and increases speed-most particularly with complex queries and very large datasets. More specialized than the general-purpose Hadoop, HPCC Systems includes customized frameworks for data processing, such as ECL which caters to efficient querying and analysis.

When these two frameworks are compared against each other in real situations, Hadoop took 6 seconds for the delivery of results from a non-indexed dataset, whereas HPCC could do it within 1.7 seconds. The multithreaded Python application was made to search for in a smaller dataset of 300 records and took 12 seconds to return results, showing that its use for larger datasets would be impractical. Such a difference underlines the advantage of HPCC on grounds of high speed and efficiency, particularly when rapid retrieval and processing of data becomes necessary. Thus, the application provides highly satisfactory results in terms of search times and reference relevance.

Thus, the developed application significantly streamlines legal research by offering a fast, user-friendly solution for reference finding based on case abstracts. Comparative analysis has demonstrated the efficiency of HPCC Systems over Hadoop and a multithreaded Python application, with HPCC Systems showing clear advantages in speed and suitability for large, real-time data processing tasks. The application's dual input options and interactive results display enhance its usability, allowing users to perform targeted legal searches quickly and effectively. These features make it an impactful tool for legal professionals who need rapid access to relevant case information.

References

1. Impact of free legal search on rule of law: Evidence from Indian Kanoon, Sandeep Bhupatiraju, Daniel L. Chen, Shareen Joshi, Peter Neis, May 20, 2024
2. Vipul Maheshwari, “Facilitating Legal Access through Digitalization of Supreme Court and High Court Records,” Bar And Bench - Indian Legal News, Feb. 05, 2024. https://www.barandbench.com/law-firms/view-point/facilitating-legal-access-digitalization-of-supreme-court-high-court-records
3. K. Tung, “AI, the internet of legal things, and lawyers,” *Journal of Management Analytics*, vol. 6, no. 4, pp. 390–403, Oct. 2019, doi: 10.1080/23270012.2019.1671242.
4. Vinay, S & Pub, Iaeme. (2024). NATURAL LANGUAGE PROCESSING FOR LEGAL DOCUMENTATION IN INDIAN LANGUAGES. 1. 1-11.
5. Modi, Hiral, Leveraging Natural Language Processing for Legal Research: Trends and Future Directions (July 18, 2023). Available at SSRN: https://ssrn.com/abstract=4514036 or http://dx.doi.org/10.2139/ssrn.4514036
6. Quevedo Caballero, Ernesto & Černý, Tom & Rodriguez, Alejandro & Rivas, Pablo & Yero Salazar, Jorge & Sooksatra, Korn & Zhakubayev, Alibek & Taibi, Davide. (2023). Legal Natural Language Processing from 2015-2022: A Comprehensive Systematic Mapping Study of Advances and Applications. IEEE Access.
7. B. Thillaieswari and M. Phil, “Comparative Study on Tools and Techniques of Big Data Analysis.” Available: https://www.ijana.in/Special%20Issue/TPID15.pdf
8. Devins, Caryn and Felin, Teppo and Koppl, Roger and Koppl, Roger and Kauffman, Stuart, The Law and Big Data (March 15, 2017). Cornell Journal of Law and Public Policy, Vol. 27, No. 2, 2017, Available at SSRN: https://ssrn.com/abstract=4389815
9. S. Fina, “Big Data & Litigation: Analyzing The Expectation of Lawyers to Provide Big Data Predictions when Advising Clients,” *Indian Journal of Law and Technology*, vol. 13, no. 1, Jan. 2017, doi: 10.55496/mnpz6794.
10. Z. Ződi, “Law and Legal Science in the Age of Big Data,” *Intersections*, vol. 3, no. 2, Jun. 2017, doi: 10.17356/ieejsp.v3i2.324.
11. Baruni JS, Sathiaseelan JG. Keyphrase extraction from document using RAKE and TextRank algorithms. Int. J. Comput. Sci. Mob. Comput. 2020 Oct;9(9):83-93.
12. Issa, B., Jasser, M. B., Chua, H. N., & Hamzah, M. (2023, October). A comparative study on embedding models for keyword extraction using KeyBERT method. In 2023 IEEE 13th International Conference on System Engineering and Technology (ICSET) (pp. 40-45). IEEE.
13. Yao, L., Pengzhou, Z., & Chi, Z. (2019, June). Research on news keyword extraction technology based on TF-IDF and TextRank. In 2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS) (pp. 452-455). IEEE.
14. Jehangir B, Radhakrishnan S, Agarwal R. A survey on Named Entity Recognition—datasets, tools, and methodologies. Natural Language Processing Journal. 2023 June 1;3:100017.
15. Schmitt X, Kubler S, Robert J, Papadakis M, LeTraon Y. A replicable comparison study of NER software: StanfordNLP, NLTK, OpenNLP, SpaCy, Gate. In2019 sixth international conference on social networks analysis, management and security (SNAMS) 2019 Oct 22 (pp. 338-343). IEEE.

16. Shelar H, Kaur G, Heda N, Agrawal P. Named entity recognition approaches and their comparison for custom ner model. Science & Technology Libraries. 2020 Jul 2;39(3):324-37.
17. Kalamkar, Prathamesh, Astha Agarwal, Aman Tiwari, Smita Gupta, Saurabh Karn, and Vivek Raghavan. "Named entity recognition in Indian court judgments." arXiv preprint arXiv:2211.03442 (2022).
18. Akbik A, Bergmann T, Blythe D, Rasul K, Schweter S, Vollgraf R. FLAIR: An easy-to-use framework for state-of-the-art NLP. InProceedings of the 2019 conference of the North American chapter of the association for computational linguistics (demonstrations) 2019 Jun (pp. 54-59).
19. ElDin, Heba Gamal, Mustafa AbdulRazek, Muhammad Abdelshafi, and Ahmed T. Sahlol. "Med-Flair: medical named entity recognition for diseases and medications based on Flair embedding." Procedia Computer Science 189 (2021): 67-75.
20. B. Glavic, K. Hale, J. Wang, K. Chard, L. Ramakrishnan, and I. Raicu, “DISTRIBUTED INDEXING AND SEARCH IN LARGE-SCALE STORAGE SYSTEMS” Available: http://216.47.155.57/publications/2022_IIT_PhD-proposal_Alexandru-Orhean.pdf
21. Locke, D., & Zuccon, G. (2018). A Test Collection for Evaluating Legal Case Law Search. The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval - SIGIR ’18. doi:10.1145/3209978.3210161
22. Xiao, C., Zhong, H., Guo, Z., Tu, C., Liu, Z., Sun, M., Zhang, T., Han, X., Hu, Z., Wang, H., & Xu, J. (2019). CAIL2019-SCM: A Dataset of Similar Case Matching in Legal Domain. Proceedings of the 2019 Conference on Natural Language Processing and Chinese Computing.
23. Vacek, T., Teo, R., Song, D., Cowling, C., Schilder, F., & Nugent, T. (2019). Litigation Analytics: Case Outcomes Extracted from US Federal Court Dockets. In Proceedings of NAACL-HLT.