**Company:**

Ernst & Young LLP   
Technology Consulting, Data & Analytics   
UB City, 'Canberra Block', #24, Vittal Mallya Road, Bengaluru, , Karnataka, 560001, India

**Team Members:**

**Rohit Gavval: Manager,** Data & Analytics

**Onkar Pawar: Data Scientist,** Data & Analytics

**Varsha: Associate Data Scientist,** Data & Analytics

**Abhishek: Data Scientist,** Data & Analytics (+91 9632579892)

**Problem Statement B:**

Development of an artificial intelligence-based model for conversational use case chatbot in English and scheduled languages of the Constitution of India, 1950, to answer queries about case-related information, summarisation of judgments, court documents, etc.

**POC Architecture:**

A screenshot of a computer

Description automatically generated

We used multiple preprocessing modules to prepare data, used effective advance summarisation method with map reduce to extract summary, leveraged skillsets to enrich exiting legal documents data and stored in the index database

A screenshot of a computer

Description automatically generated

During Retrieval stage we analyse user question, we analyse the source language and perform question translation to English if required, perform embedding generation, using vectors and question we perform hybrid search(sematic + vector search) with reranking Fusion in build algorithm extract the top relevant chunks to send it to LLM to answer question based on factual evidence, if use require in regional language we perform translation from English to regional language to get the response

**POC Results:**

**QNA in regional language:**

A screenshot of a computer

Description automatically generated

**QNA in English**

A screenshot of a computer

Description automatically generated

**QNA in cross languages : ex: question in English response in regional language**

A screenshot of a computer

Description automatically generated

**Sample evidence for above response correctness:**

**s**A screenshot of a document

Description automatically generated

**Summary Generation In English**

A screenshot of a computer

Description automatically generated

**Summary Generation In Regional Language:**

A screenshot of a computer

Description automatically generated

**Understanding & Background Research On Problem Statement:**

**Language support required for – AI Chat Bot**

The Eighth Schedule of the Indian Constitution originally defined 14 languages in 1950, reflecting the linguistic diversity of the country. These languages included: Assamese, Bengali, Gujarati, Hindi, Kannada, Kashmiri, Malayalam, Marathi, Oriya, Punjabi, Sanskrit, Tamil, Telugu, Urdu

**Public datasets by the Supreme Court Of India for POC – AI Chat Bot**

Under the digitization initiative led by the Chief Justice of India, Dr. Justice Dhananjaya Y Chandrachud, the Supreme Court has launched the Electronic Supreme Court Reports (e-SCR) project on 2nd January 2023. This project is part of the broader objective of digitizing the Indian judiciary.

**Electronic Supreme Court Reports (eSCR) Portal:** The eSCR is a digitized version of the Supreme Court Reports, the official law report of India. The portal allows users to search for judgments using various criteria, such as keywords, judge names, acts or sections, and party names. Judgments are available for free download in PDF format via the portal: eSCR Portal.

As of 30th August 2024, the eSCR portal contains:

* 36,803 Supreme Court judgments
* 14,494,999 High Court judgments

For our Proof of Concept (POC) during the hackathon, we will focus on analysing all judgments from January 2024 to August 2024 where the Chief Justice of India was part of the judges' panel. This selection includes approximately 30 judgments and will serve as the primary dataset for our AI-based legal research solution.

A screenshot of a computer

Description automatically generated

Sample metadata:

A screen shot of a computer

Description automatically generated

Supreme Court of India: Information Technology Services

The Supreme Court of India, through its e-Committee, has implemented various information technology services to enhance judicial processes. These services include live streaming, live court hearing transcriptions and scrutiny, e-Courts Services Mobile App, e-Courts Fee Payment, Touch Screen Kiosks, e-Courts Services Portal, Virtual Courts, e-Seva Kendra, High Court Services, National Judicial Data Grid, e-Filing. These will be helpful to enable deep AI technology connection multiple existing services

**Data Collection Flow For AI Chat Bot :**

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

Above diagram shocases the backend flow of indexing escr judements files.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Problem Statement A:**

Development of an artificial intelligence-based model for extraction of data including metadata, data fields such as name of parties, address, Act, section – legal provisions, subject categories, identification of formats of petitions such as special leave petition – Form 28, Supreme Court Rules 2013, statutory appeals, etc., to facilitate in scrutiny of cases, removal of defects.

Stage 1: Classification/Identification of Document Type Format

The first stage focuses on classifying and identifying various legal document types, such as special leave petitions (Form 28), Supreme Court Rules 2013, and statutory appeals. This classification is essential for applying the correct extraction models to each document type.

Solution Flow:  
A computer screen shot of a computer

Description automatically generated

Methodology

1. Utilization of Azure AI Document Intelligence:

This cloud-based service analyses documents to extract and detect various content, layout, and semantic elements. Custom classification models, which combine layout and language features, can accurately identify document types.

2. Training Custom Classification Models:

A minimum of \*\*two distinct classes\*\* with at least \*\*five document samples per class\*\* is required for training. The model classifies documents one page at a time and can identify multiple document types within a single file Or Training of Custom entity recognition model

**Implementation Steps**

1. Define document classes, including:

- Special Leave Petition – Form 28

- Supreme Court Rules 2013

- Statutory Appeals

2. Collect and prepare training samples for each class.

3. Train the classification model using Azure Document Intelligence APIs.

4. Validate the model's performance and adjust the training dataset as necessary.

By following this methodology, the AI-based model will effectively classify and identify various legal document types, setting the stage for accurate data extraction in the next phase.

Stage 2: Extraction of Data Fields and Metadata Fields

Feasibility:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Overview

The second stage focuses on extracting specific data fields and metadata from the identified document types. This involves defining key-value pairs or query fields to capture the required information.

Conclusion

By following these structured stages, the development of an AI-based model for data extraction will be streamlined, ensuring efficient classification and accurate data retrieval from legal documents. This approach leverages Azure AI Document Intelligence's robust capabilities, facilitating the scrutiny of cases and the removal of defects in legal documentation processes.

The extraction process can be integrated with Azure’s REST API to automate the retrieval of key-value pairs and query fields from processed documents.

It is essential to ensure that the model is trained on layouts that include the necessary fields to optimize extraction accuracy.

For scenarios where the fields to be extracted are not predefined or exceed a certain number (greater than 20), query fields can be used. This allows for a flexible schema that can adapt to various document structures.

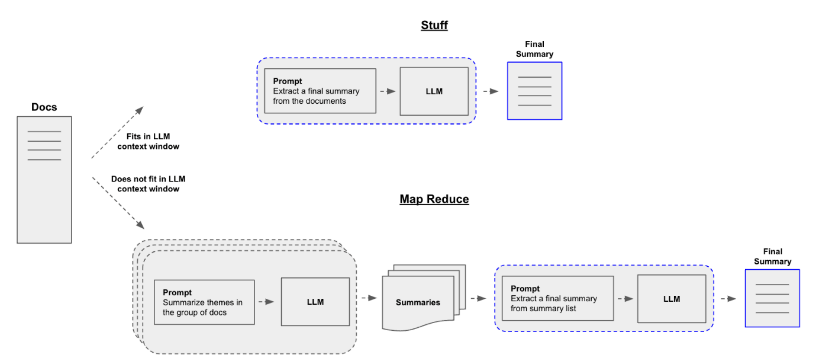
Query fields enable the extraction of specific values based on defined keys, which is useful for capturing unique information such as contract dates.

Key-value pairs are utilized for extracting known fields directly from documents. This method is effective when the fields are predictable and have consistent naming conventions (e.g., "First Name" vs. "Given Name").

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**INNOVATION AND CONCEPTS ALIGNMENT SPECIFIC TO Supreme COURT AI CHATBOT:**

**ONE TIME SUMMARY GENRATION TO SAVE COST:**



**Method 1: Single-Pass Summarization**

A single LLM processes the entire document to generate a summary. Suitable for smaller documents but may struggle with larger ones due to context window limitations.

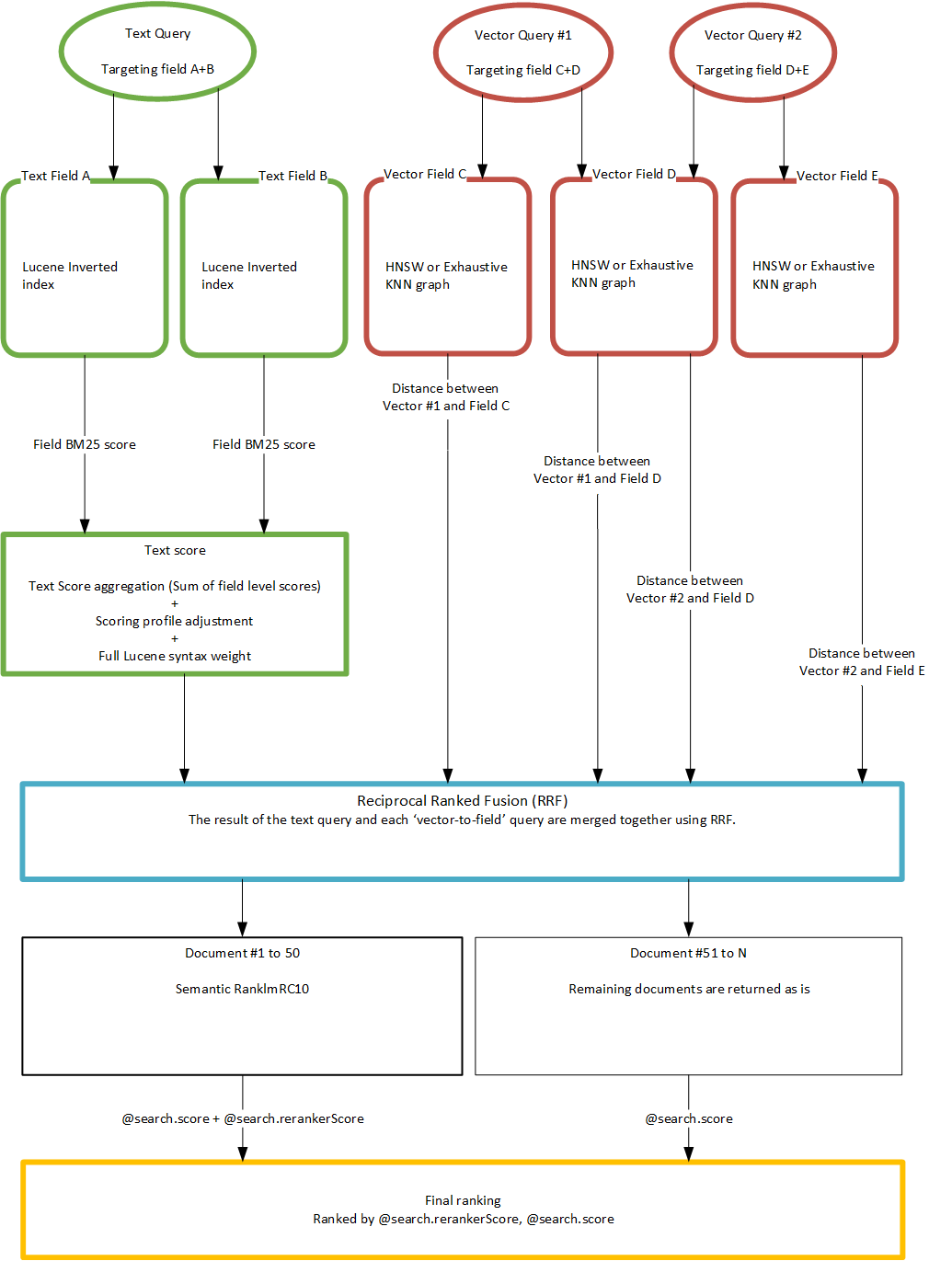
**Method 2: MapReduce Summarization**

The document is divided into chunks, each summarized by a separate LLM. The summaries are then combined using MapReduce to generate a final summary. Suitable for large documents, as it avoids context window limitations and leverages multiple LLMs.

**ENHANCED RETRIVAL OF CHUNKS USING RRF ALGORITHM:**

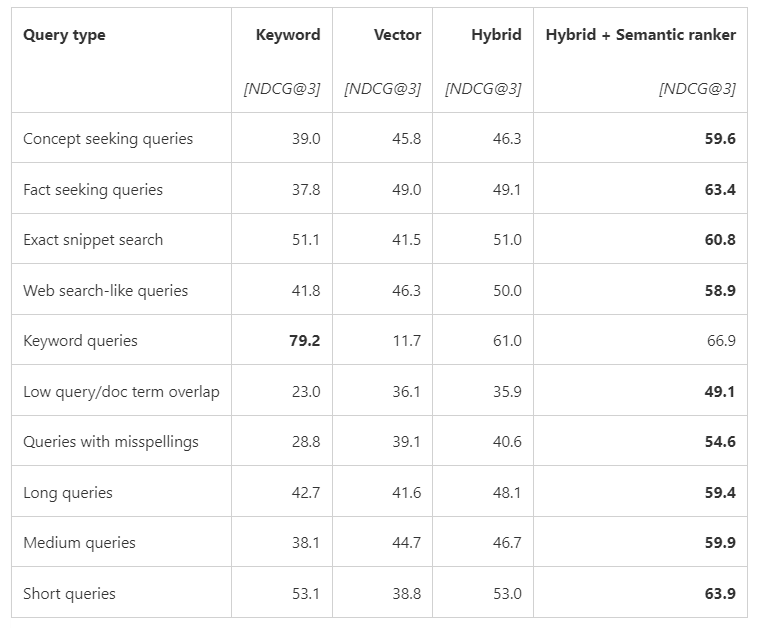
Research : [Azure AI Search: Outperforming vector search with hybrid retrieval and ranking capabilities - Microsoft Community Hub](https://techcommunity.microsoft.com/t5/ai-azure-ai-services-blog/azure-ai-search-outperforming-vector-search-with-hybrid/ba-p/3929167)

RRF (Reciprocal Rank Fusion) is a ranking algorithm that combines multiple ranking lists into a single ranked list. It is often used in hybrid search systems to merge the results from different search engines or ranking algorithms.



The diagram illustrates a hybrid search approach that combines text-based and vector-based queries to rerank search results. Text queries are processed using Lucene inverted indexes, while vector queries are processed using HNSW or exhaustive KNN graphs. The results from both types of queries are merged using RRF, and the final ranking is determined by a combination of text score and reranker score.

**RESEACH OUTCOME OF AN HYBRID SEARCH FOR FACT SEEKING QUERY:**



Generative AI scenarios typically use the top 3 to 5 results as their grounding context to prioritize the most important results. AI Search applications work best with a calibrated relevance score that can be used to filter out low quality results.

The semantic ranker runs the query and documents text simultaneously though transformer models that utilize the cross-attention mechanism to produce a ranker score. The query and document chunk score is calibrated to a range that is consistent across all indexes and queries. A score of 0 represents a very irrelevant chunk, and a score of 4 represents an excellent one. In the chart below, Hybrid + Semantic ranking finds the best content for the LLM at each result set size.

thumbnail image 1 of blog post titled 
 
 
  
 
 
 
    
  
   
    
      
       Azure AI Search: Outperforming vector search with hybrid retrieval and ranking capabilities
       
      
     
   
  
 
   
 
 
 
 
 


Percentage of queries where high-quality chunks are found in the top 1 to 5 results, compared across search configurations. All retrieval modes used the same set of customer query/document benchmark. Document chunks were 512 tokens with 25% overlap. Vector and hybrid retrieval used Ada-002 embeddings