International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

IndicMind: Multilingual Named Entity Recognition System for Indian Languages

[1]**Dr. Sangram S. Dandge,** Assistant Professor, CSE Dept, PRMIT&R, Badnera,
[2]**Radha K. Waghmare,** [3]**Rutuja U. Bhabutkar,** [4]**Sujit A. More,** [5]**Yash P. Pokle,** [6] **Zaina Fatema Irfan Ahmed**
[23456]Final Year, Computer Science & Engineering Dept, PRMIT&R,Badnera-Amravati MAHARASHTRA

Abstract - IndicMind is developed to simplify the extraction and retrieval of information from documents containing Indian-language content. The application brings document processing, named entity identification, and document-based question answering together through a web interface. Users can submit supported document formats and examine the information identified by the system. The application uses a Flask backend to coordinate text extraction and the configured NLP and AI processing components, including spaCy, Gemini API, and Ollama. It also provides interactive facilities such as document queries and speech-based interaction where supported by the application. The work explores how these components can be integrated to make multilingual document information easier to locate and examine.

Keywords - Multilingual Named Entity Recognition, Indian Languages, Natural Language Processing (NLP), Document Analysis, spaCy, Gemini API, Ollama, Retrieval-Augmented Generation (RAG), Information Extraction, Document-Based Question Answering.

I. INTRODUCTION

Understanding and analysing multilingual documents is an important requirement in domains such as government administration, education, healthcare, finance, and legal services. In India, documents are created in several languages and writing scripts, making information extraction a challenging task. The presence of unstructured content, scanned pages, and different document formats further increases the difficulty of locating specific information. Named Entity Recognition (NER) provides a method for identifying important information, including person names, organizations, locations, and dates, from textual content [10].

Research on Indian-language NER has explored different approaches, including multilingual language models, cross-lingual learning, and language-specific recognition techniques [2], [6]–[9]. Other studies have investigated document analysis and OCR-based extraction to process information from digital and image-based sources [3], [4]. However, these individual approaches address different aspects of document understanding, leaving scope for applications that combine document processing, entity identification, and interactive information retrieval.

To address this requirement, this paper presents IndicMind, a Flask-based multilingual document analysis system for selected Indian languages. The application integrates document text extraction, NER, AI-assisted processing, and document-based question answering. It uses spaCy and configured language models through the Gemini API and Ollama to support entity identification and document interaction. The system also provides a web interface for displaying extracted entities and supports voice-based interaction through STT and TTS
components. The main objective is to simplify the process of identifying and retrieving relevant information from multilingual documents.

II. LITERATURE REVIEW

The development of Natural Language Processing (NLP) has improved the ability of computer systems to process, analyze, and extract useful information from unstructured text. Techniques such as Named Entity Recognition (NER), Optical Character Recognition (OCR), and multilingual language models help identify important information from documents written in different languages. Recent research has focused on improving entity recognition for low-resource languages, extracting text from scanned documents, and developing systems that can understand document content more effectively. However, differences in language structure, writing scripts, document formats, and the availability of annotated datasets continue to create challenges for accurate information extraction.

Research in multilingual Named Entity Recognition has focused on developing methods for identifying entities across languages with different linguistic structures and levels of

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

resource availability. Several studies have investigated multilingual datasets, cross-lingual transfer learning, and language-specific models to improve recognition performance for Indian languages [1], [2], [6]–[10]. In addition, research on document understanding and OCR-based processing has explored techniques for extracting information from scanned documents and combining textual and visual information [3], [4].

Author / Reference	Methodology	Contribution / Focus	Limitation / Scope
Kaushik and Anand [1]	SampurNER dataset and fine-grained NER for 22 Indian languages	Provides a dataset for studying and evaluating Indian-language NER	Focuses on dataset development and entity recognition rather than a complete document-analysis application
Yadav et al. [2]	Multilingual BERT and cross-lingual transfer learning	Studies Hindi–Nepali NER for low-resource language settings	Focuses on a particular language pair and NER task
Kumar et al. [3]	Automated multilingual document understanding and analysis	Explores document processing and multilingual information extraction	The paper's scope is document analysis rather than the complete combination of NER, voice interaction, and document Q&A
Rai et al. [4]	OCR-extracted text, ResNet-101, TF-IDF, and XGBoost	Combines textual and visual features for meme-based hate-speech classification	Designed for meme classification rather than general-purpose multilingual document analysis
Kiruthika et al. [5]	Integrated Speech-to-Text and Text-to-Speech	Explores real-time cross-language communication using speech technologies	Focuses on communication rather than document-based entity extraction
Dhamaskar and Ransing [6]	TriNER models for Hindi, Bengali, and Marathi	Focuses on NER for three Indian languages	Limited to the languages and entity-recognition scope addressed in the study
Bahad et al. [7]	Fine-tuning pretrained NER models	Investigates adaptation of pretrained models for Indian-language entity recognition	Focuses on model fine-tuning rather than an integrated document application
Mohan et al. [8]	Fine-tuned multilingual BERT-based model	Studies entity recognition in native Indian languages	Focuses on model-based NER
Sharma et al. [9]	Neural language model and Conditional Random Field (CRF)	Investigates Hindi-language NER	Focuses on Hindi entity recognition
Vijayvergia et al. [10]	Survey of NER approaches for Indian languages	Reviews different approaches used for Indian-language NER	Provides a survey rather than an implemented document-processing platform

The reviewed literature indicates that multilingual NER, OCR-based document processing, and speech technologies have been investigated through different approaches. However, these capabilities are often addressed as separate research problems. There is an opportunity to integrate document text extraction, entity identification, and document-based question answering into a single accessible application. IndicMind is designed to address this integration requirement by providing a common platform for analysing selected Indian-language documents, displaying recognized entities, and supporting user interaction through text and voice-based queries.

II. METHODOLOGY

The proposed IndicMind AI system is a Flask-based multilingual document analysis platform designed to extract named entities and support document-based question answering. The system accepts supported digital documents and image-based inputs through a web interface. Text is extracted from digital documents using the application's document-handling mechanism, while image inputs are sent to the Google Gemini API for multimodal text extraction.

The extracted text is prepared for subsequent processing. The backend uses spaCy for named entity recognition when its configured NER pipeline is invoked. Depending on the implemented workflow, language-model-based processing

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

using Gemini or Ollama may also support document understanding and question answering. Gemini provides cloud-based model inference, while Ollama supports local model inference and can serve as a fallback when the application invokes that mechanism.

For document-based question answering, the system uses the document content and the user's query to generate a response. Where a retrieval pipeline is implemented, relevant document segments are retrieved and supplied to the selected language model as context. The resulting entities and answers are presented through the web interface, enabling users to inspect extracted information and query the processed document. Voice input and speech output are included where the corresponding speech-processing components are enabled.

A. Backend Entity Identification Mechanism

Named Entity Recognition (NER) is an NLP task that identifies real-world entities in text and classifies them into categories such as Person (PER), Organization (ORG), Location (LOC), and Date (DATE). In an IndicMind document analysis system, entity identification generally involves the following technical stages.

First, the input document is converted into machine-readable text. The text is then tokenized into words or subword units and processed by an NLP model. SpaCy provides NLP processing and entity recognition, while Gemini or Ollama can be used for additional extraction, depending on the implemented backend workflow.

In sequence-based entity recognition, the input text is represented as an ordered sequence of tokens. The recognition process associates each token with a label that indicates whether it belongs to an entity and its position within that entity. One commonly used representation is the BIO tagging scheme [12]. In this scheme, B indicates the first token of an entity, I represents a token that continues an existing entity, and O identifies a token that does not belong to any recognized entity.

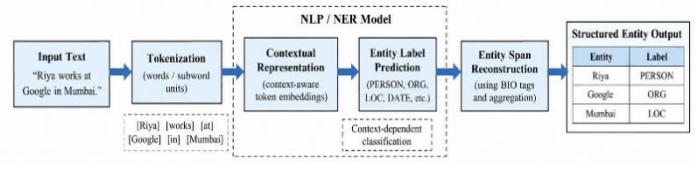


Fig. 1 Sequence-Based Entity Identification and Classification process.

B. Text Processing and Entity Identification Using spaCy

Within IndicMind, spaCy acts as one of the NLP components responsible for processing textual input and identifying entity spans. When the backend invokes the configured spaCy pipeline, the input string is converted into a document object. The tokenizer separates the text into tokens, after which the enabled pipeline components perform their respective processing operations. If a trained NER component is available and invoked, the detected entities can be accessed through the Doc.ents attribute. Each entity contains its text and associated category, which can subsequently be converted into a structured representation for display in the application.

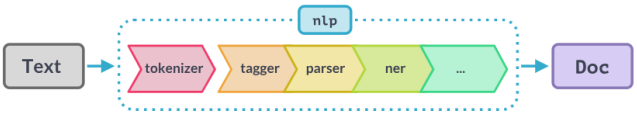


Fig. 2 spaCy Processing Pipeline (Tokenizer, Tagger, Parser, NER)

Algorithmic interpretation

For an input document D, spaCy processes the text using its configured language pipeline:

Each detected $D \xrightarrow{\text{Tokenizer}} \text{Doc} \xrightarrow{\text{NER}} \{e_1, e_2, \ldots, e_m\}$

$$e_i = (\text{text}_i, \text{label}_i, \text{start}_i, \text{end}_i)$$

Here, text is the extracted entity, label is its predicted category, and start and end denote its character boundaries in the document.

C. Transition-based named entity recognition

The Entity Recognizer component in spaCy uses a transition-based approach to identify labelled entity spans in text. Rather than treating each token as an independent classification result, the component predicts structured actions that determine the boundaries and labels of recognized entities. The resulting entity spans are stored in the Doc.ents attribute, while token-level entity information is represented through the corresponding entity-type and IOB attributes [11].

The transition-based method is suitable for conventional named entity recognition tasks involving non-overlapping entity spans. However, its performance depends on the language model, training data, and characteristics of the input text. In IndicMind, the actual entity-recognition behaviour depends on the spaCy pipeline and model configuration loaded by the backend.

D. NER Architecture Considerations

NER systems can employ different model architectures, including transition-based structured prediction and Transformer-based token classification. In a transition-based

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

approach, the model predicts actions that construct labeled entity spans. In Transformer-based approaches, contextual token representations are used to predict entity labels, which can subsequently be grouped into complete spans.

These approaches are not identical. IndicMind AI's implemented entity-recognition behavior is determined by the model and pipeline actually invoked by the backend. Transformer-based architectures such as BERT or XLM-R are discussed here as alternative approaches and should not be considered part of the implementation unless the corresponding models are explicitly integrated.

E. Optical Character Recognition

In IndicMind, image-based text extraction is performed by passing the input image to the Gemini API, which supports multimodal image understanding and text generation. An image can be represented as:

$$\text{Image} = H \times W \times C$$

where H represents the image height in pixels, W represents the image width in pixels, and C represents the number of color channels.

In IndicMind, image-based inputs are submitted to the Gemini API for multimodal processing. The input image contains visual information represented through its height, width, and colour channels. The model analyses this visual input and generates a textual response containing the recognized document content. The resulting text is then supplied to the subsequent processing stages of the application. This approach allows image-based information to enter the same document-analysis workflow as text extracted from digitally encoded files.

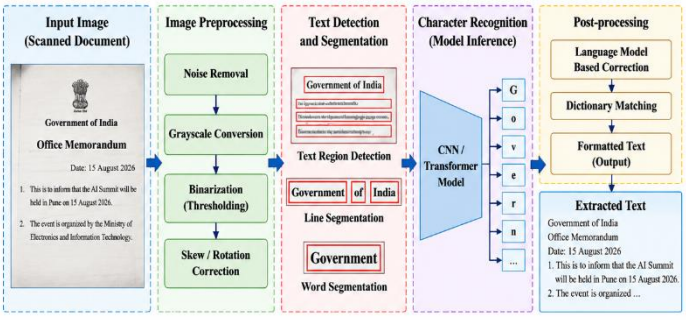


Fig. 3 Optical Character Recognition Process for Extracting Text from Scanned or Image-Based Documents

III. WORKING

A. System Overview

IndicMind AI is a web-based multilingual document analysis and knowledge-extraction system designed to identify named entities from user-provided content and facilitate context-aware interaction with document information. The system is implemented using a Python-based Flask backend, with spaCy, the Gemini API, and an Ollama-hosted local language model serving as the principal AI-related components described in the implementation. The application provides a unified interface for document submission, entity extraction, visualization of recognized information, and document-based question answering.

The working process is organized into two complementary processing paths: (i) named entity extraction and structured result generation, and (ii) retrieval-augmented document question answering. The first path identifies and categorizes relevant information, whereas the second retrieves contextually relevant document content to answer user queries. The Flask backend coordinates these operations and returns the resulting information to the frontend.

The processing pipeline begins when a user submits a supported document, enters plain text, or provides a voice-based query through the application interface. The document-processing workflow accepts formats such as PDF, DOCX, TXT, and supported image files, subject to the application's upload and file-handling restrictions.

After receiving a document, the backend identifies the applicable content-extraction mechanism. For digitally encoded documents, textual content can be obtained through a suitable document parser. For scanned or image-based documents, an OCR component is required to convert image content into machine-readable text. OCR should be considered part of the implemented workflow only when the application actually invokes an OCR-capable library, service, or multimodal model for that purpose.

The extracted text is passed to the subsequent analysis stage. Depending on the document format and the operations supported by the application, this stage may include whitespace normalization, removal of extraction artefacts, and preparation of the text for model inference. These operations are intended to preserve the original semantic content while making the input suitable for entity recognition and document analysis.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

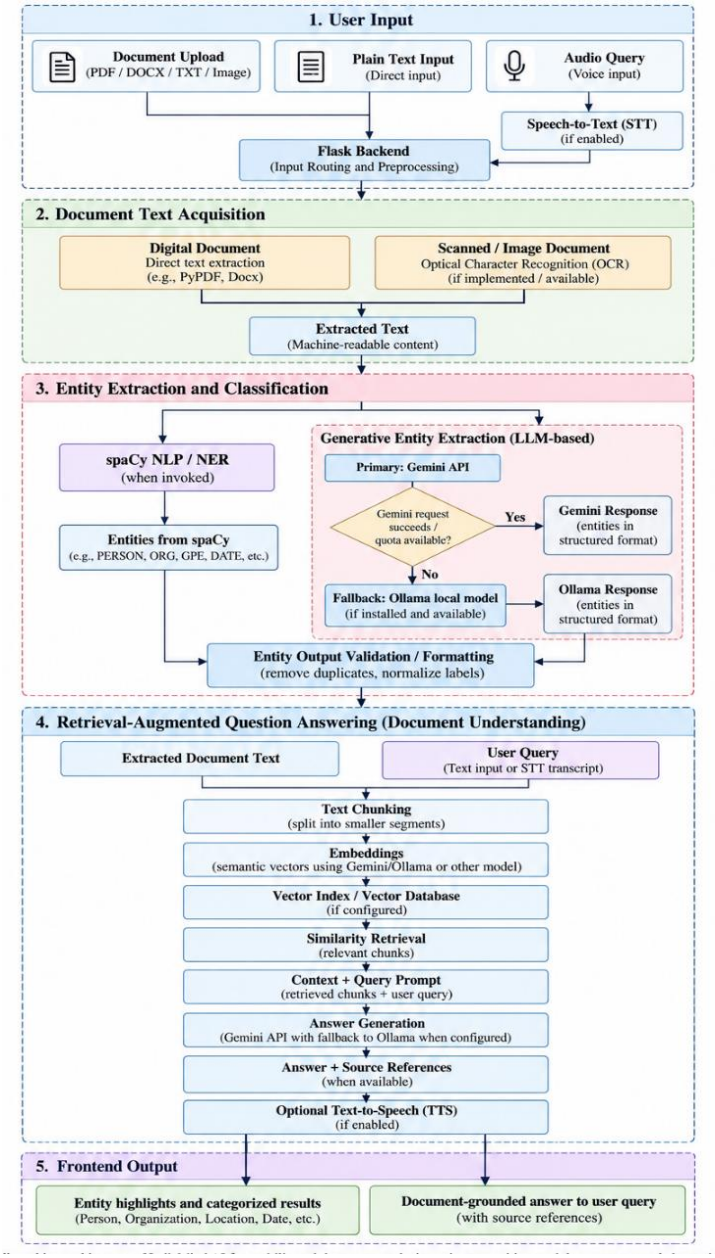


Fig. 4 Overall Working Architecture of IndicMind for Multilingual Document Analysis, Entity Recognition, and Document-Grounded Question Answering

B. Named Entity Recognition and Classification

The principal information-extraction stage identifies meaningful entity mentions and associates them with semantic categories. IndicMind AI uses spaCy as an NLP and entity-recognition component when the corresponding pipeline is invoked by the backend. The application also integrates the Gemini API and an Ollama-based local model for generative AI operations, including entity extraction where that processing path is selected.

In a spaCy-based processing path, the input text is processed by the configured language pipeline. The pipeline tokenizes the text and passes the resulting document representation through its enabled components. When a trained named entity recognition component is present and invoked, the recognized entity spans and their labels are available through the document's entity collection, Doc.ents. The exact labels and recognition capabilities depend on the model and configuration loaded by the application.

The output of this stage is represented as a collection of structured entity records:

$E = \{e_1, e_2, .., e_m\}$

where an entity record can be represented as

$e_i = (s_i, c_i)$

Here, (s_i) denotes the extracted entity text and (c_i) denotes its assigned category. Character offsets, confidence scores, and provenance information may also be retained if the selected model and application implementation provide them.The extracted information may include person names, organizations, geographical locations, dates, monetary amounts, and other application-defined categories. The resulting records are passed to the presentation layer, where entities can be displayed using colour-coded highlights and category-wise result panels. This structured representation makes important information easier to inspect without requiring the user to read the complete document manually.

C. Gemini API and Ollama-Based Fallback Mechanism

IndicMind AI incorporates a cloud-based inference path using the Gemini API and a local inference path using Ollama. This arrangement provides an alternative processing option when the cloud-based service cannot be used, provided that the fallback behaviour is implemented in the backend. Under normal operation, the application attempts to process the relevant request using the selected Gemini model. If the request succeeds, the returned response is parsed and forwarded to the next processing stage. If the request fails because of a handled condition, such as a quota or rate-limit error, a network failure, or a service-availability issue, the backend may invoke the configured Ollama model as a fallback.

The use of Ollama enables inference through a locally available model and can reduce dependence on cloud availability. However, local inference requires the relevant model to be installed and the Ollama service to be running. Its language coverage, processing speed, and extraction quality depend on the model selected and the available hardware. Therefore, the fallback mechanism provides an alternative

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

execution path rather than guaranteeing identical output from both models

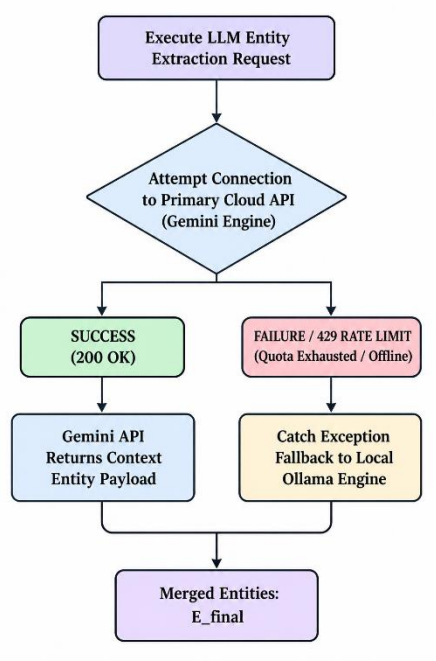


Fig. 5 Gemini API to Ollama Fallback Mechanism for Entity Extraction

D. Retrieval-Augmented Generation (RAG) & Vector Database Operations

When the user enters a specific query (Q) regarding an uploaded document, IndicMind executes a Retrieval-Augmented Generation (RAG) workflow to prevent hallucination and provide verified, grounded answers backed by source clause citations.

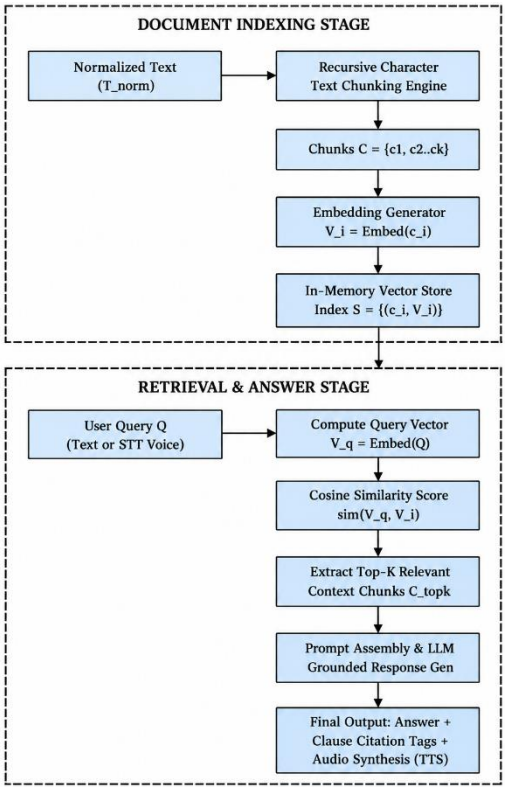


Fig. 6 RAG Workflow: Document Indexing Stage and Retrieval and Answer Stage

IV. SYSTEM REQUIREMENT

SOFTWARE REQUIREMENT

- **Frontend:** HTML, CSS
- **Backend:** Python (Flask Framework)

V. IMPLEMENTATION & RESULT

IMPLEMENTATION

A. Document Upload Interface

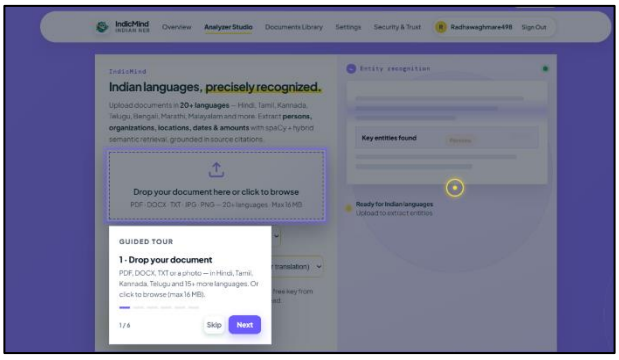


Fig. 7 shows the Analyzer Studio showing the document upload area and guided instructions for initiating multilingual document analysis

The Analyzer Studio provides the main working interface for submitting documents to the MNER System. As shown in the figure, the document upload section is highlighted by an on-

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

screen guide that directs users to drag and drop a file or select one from their device. The interface accepts PDF, DOCX, TXT, JPG, and PNG files up to 16 MB and supports documents written in multiple Indian languages. The guided tour also includes navigation controls to help users understand the workspace before starting the analysis. This screen establishes the initial stage of the document processing workflow, where users provide the input required for subsequent text extraction and named entity recognition.

B. Named Entity Recognition

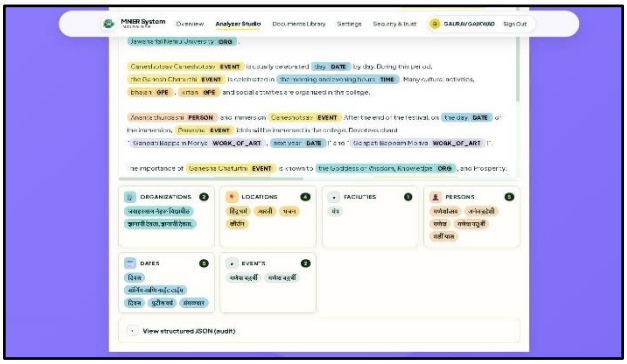


Fig. 8 shows the Colour-coded entity highlighting in the processed document showing the recognition of people, organizations, locations, and dates

The figure illustrates how the IndicMind System presents recognized entities directly within the processed document. Important names and references, such as Claude Shannon, Frank Rosenblatt, Arthur Samuel, IBM, Dartmouth, and the years 1955, 1956, and 1959, are highlighted with different colours and assigned corresponding entity labels.

C. Document-Based Question Answering Interface

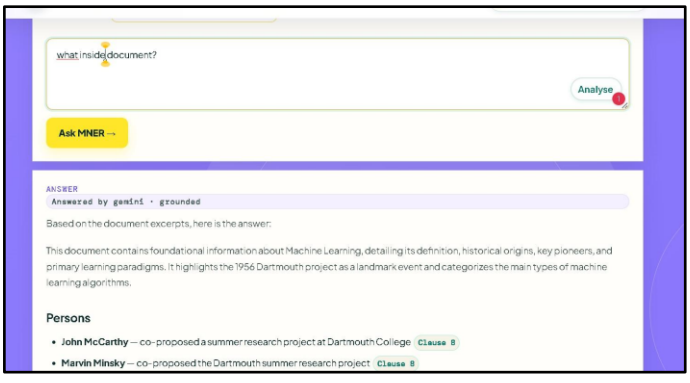


Fig. 9 shows the MNER question-answering results displaying a document-grounded response with extracted entities and source clause references

The figure shows the response generated after a user submits the question “What inside document?” through the MNER question-answering interface. The system displays an answer based on the processed document, summarizing its content related to machine learning, its historical development, key contributors, and learning paradigms.

RESULT

The application was tested using documents containing text in different Indian languages. During testing, the uploaded content was passed through the available extraction and entity-identification components. The recognized information was presented through category-wise results and highlighted sections in the document interface. The question-answering interface was also examined using queries related to the uploaded content. These tests provided an opportunity to observe the functioning of the different modules and identify areas requiring further improvement.

VI. CONCLUSION

The Multilingual Named Entity Recognition System for Indian Languages provides a useful approach to simplifying the analysis of documents written in different Indian languages. By combining OCR, NLP, Named Entity Recognition, and Retrieval-Augmented Generation, the system helps users extract important information and find relevant details without having to read entire documents manually. The integration of text and voice-based queries, along with speech output, makes the platform more convenient and accessible for different types of users. It also offers practical value in areas such as government administration, education, healthcare, legal services, and finance, where information is frequently stored in different formats and languages. Although the system may face challenges related to OCR quality, language variations, and the accuracy of generated responses, these limitations can be addressed through further testing and improvement. Overall, the paper demonstrates how multilingual language processing technologies can make document information easier to retrieve, understand, and use in real-world situations.

VII. REFERENCE

[1] Kaushik, Prachuryya, and Ashish Anand. "SampurNER: Fine-Grained Named Entity Recognition Dataset for 22 Indian Languages." In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 40, no. 37, pp. 31410-31418. 2026.

[2] Yadav, Dipendra, Sumaiya Suravee, Tobias Strauß, and Kristina Yordanova. "Cross-lingual named entity recognition for low-resource languages: A Hindi-Nepali case study using multilingual BERT models." In Proceedings of the Fourth Workshop on Multilingual Representation Learning (MRL 2024), pp. 167-174. 2024.

[3] Kumar, M. Prasanna, G. Sri Tulasi, and T. Hema Varshini. "Automated Multilingual Document Understanding and Analysis Platform." In 2026 Contemporary Computing Innovations Conference (CCIC), pp. 1-5. IEEE, 2026.

[4] Rai, Subham Kumar, Kunal Goswami, Gunjan Kumar, Sangita Singh,

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

and Jyoti Prakash Singh. "Multilingual Hate Speech Classification in Memes Using OCR-Extracted Text and Visual Features." In FIRE (Working Notes), pp. 666-677. 2025.

[5] Kiruthika, M., Balaraman Sundarambal, C. Selvaganesan, and Vishali Muralidharan. "Real-Time Cross-Language Communication with Integrated STT and TTS." In 2025 International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 475-483. IEEE, 2025.

[6] Mohammed Amaan Dhamaskar, Rasika Ransing, “TriNER: A Series of Named Entity Recognition Models For Hindi, Bengali & Marathi”, (ArXiv), 2025.

[7] Sankalp Bahad, Pruthwik Mishra, Karunesh Arora, Rakesh Chandra Balabantaray, Dipti Misra Sharma, Parameswari Krishnamurthy, “Fine-tuning Pre-trained Named Entity Recognition Models For Indian Languages”, (arXiv), May 2025.

[8] G Bharathi Mohan, R Prasanna Kumar, Elakkiya R, Mukhtesh Venkata Sri Sai Pendem, “Fine-Tuned BERT based multilingual model for named entity recognition in Native Indian languages”, (International Conference on Evolutionary Algorithms and Soft Computing Techniques), 2023.

[9] Richa Sharma, Sudha Morwal, Basant Agarwal, “Named entity recognition using neural language model and CRF for Hindi language”, (Computer Speech & Language), 2022.

[10] Rekha Vijayvergia, Bharti Nathani, Nisheeth Joshi, Rekha Jain, “A Survey on Various Approaches Used in Named Entity Recognition for Indian Languages”, (ICIMMI), 2022.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)