**Intelligent Document Analysis and Comprehension: A System for Automated Keyword Extraction, Question Answering, and Question Generation from PDF Files: A Comprehensive Analysis**

Chirag Vats, Ria Chandra, Aarjav Jain

Vellore Institute of Technology

Vellore, India

**Abstract**

In an era characterized by the proliferation of digital documents, the need for efficient and intelligent tools for document analysis and comprehension has become paramount. This paper presents the development and implementation of a novel product that facilitates document understanding through automated keyword extraction, question answering, and question generation functionalities. The system allows users to upload PDF files, from which relevant keywords are extracted to provide a concise summary of the document's content. Additionally, users can pose inquiries about the uploaded document, to which the system responds with accurate answers derived from the document's context. Furthermore, the system employs advanced natural language processing techniques to generate probable questions based on the document's content, fostering deeper engagement and comprehension. The product's architecture integrates PDF parsing, keyword extraction algorithms, question answering models, and question generation techniques into a cohesive user interface, ensuring seamless

interaction and usability. Through comprehensive testing and evaluation, the system demonstrates robust performance in document analysis tasks, offering valuable insights into the document's content while empowering users with enhanced comprehension capabilities. This paper contributes to the advancement of document analysis tools and sets the stage for further research and development in the field of intelligent document processing.

**Introduction**

In recent years, the proliferation of digital content has led to an exponential increase in the volume of information available across various domains [1]. Amidst this wealth of information, the ability to efficiently comprehend and extract meaningful insights from textual documents has become increasingly essential. Traditional methods of document analysis and comprehension often rely on manual effort and expertise, making them time-consuming and resource-intensive. However, advancements in deep learning and natural language processing (NLP) have paved the way for the development of automated systems capable of analyzing and understanding textual data with remarkable accuracy and efficiency [2].

The advent of deep learning techniques, coupled with the availability of large-scale datasets and powerful computational resources, has revolutionized the field of document analysis and comprehension. Researchers have leveraged deep learning models such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformers to develop sophisticated systems for tasks such as question generation, question answering, and information retrieval [3]. These systems exhibit remarkable capabilities in extracting semantic meaning, identifying patterns, and generating human-like responses, thereby enabling a wide range of applications across domains such as education, healthcare, finance, and more.

In this paper, we present a comprehensive study on the development and implementation of an automated system for question generation and answering from textual documents. Drawing inspiration from recent advancements in deep learning and NLP, our system employs state-of-the-art techniques to analyze the content of textual documents, extract relevant information, and generate coherent and contextually relevant questions and answers. We explore the use of deep learning models such as T5 transformer, convolutional neural networks, and bidirectional encoder representations from transformers (BERT) for tasks such as question generation, question answering, and information retrieval [4].

The contributions of this paper are twofold. Firstly, we propose a novel framework for automated question generation and answering from textual documents, leveraging the latest advancements in deep learning and NLP. Secondly, we conduct a comprehensive empirical evaluation of our system using benchmark datasets and performance metrics, demonstrating its effectiveness and scalability across various domains and document types [5]. Through this study, we aim to advance the state-of-the-art in automated document analysis and comprehension, offering valuable insights and paving the way for future research in this rapidly evolving field.

**Literature Review**

In recent years, the fields of question generation and question answering have garnered significant attention, particularly with advancements in natural language processing (NLP) and deep learning. Various methodologies have been explored to enhance the capabilities of systems designed to automatically generate questions and provide accurate answers based on a given text or set of documents.

Sun, Zhou, and Fang (2021) proposed a neural question generation system guided by question type. Their approach leverages measurement, computational modeling, and semantics to generate questions that adhere to specific types, thereby improving the relevance and contextual accuracy of the generated questions. This method integrates multitasking and metric learning to enhance question generation processes, demonstrating improved performance over baseline models.

Virani et al. (2023) introduced an automatic question-answer generation system using the T5 transformer model and various NLP techniques. Their system focuses on generating multiple types of questions, such as multiple-choice questions (MCQs), fill-in-the-blanks, true/false, and short answers. Key NLP tools such as part of speech (POS) tagging and named entity recognition (NER) are utilized to enrich the quality of the questions and answers. The system is tailored for scalability and is designed to support educational applications and other industries requiring robust question-answering mechanisms.

Singh, Suraksha, and Nirmala (2021) developed a question-answering chatbot leveraging deep learning and NLP. Their system employs BM25 for information retrieval, Stanford CorefAnnotator for coreference resolution, and word embeddings for semantic understanding. The chatbot is trained on extensive datasets and uses convolutional neural networks (CNNs) to improve the accuracy and relevance of the responses, showing promising results in various domains.

Srivastava et al. (2020) introduced "Questionator," an automated question generation system using deep learning. This system combines natural language processing techniques with deep learning models to generate questions and options. The focus on image captioning and automation highlights its versatility in various applications, including education and market research. Their methodology emphasizes pipeline efficiency and the integration of multiple data sources to enhance the quality of generated questions.

Malhar et al. (2022) developed a deep learning-based system for answering questions using the T5 transformer model. Their structured question generation system leverages WordNet to improve the semantic understanding of text and generate more accurate answers. The system shows enhanced performance in generating questions and answers from structured and unstructured data sources, making it applicable in diverse fields such as economics and biological system modeling.

Kumar et al. (2021) explored automatic question-answer pair generation using deep learning techniques. Their approach employs transformers like BERT and SQuAD datasets to train the models. The integration of the Natural Language Toolkit (NLTK) and the use of BLEU scores for evaluation emphasize the system's ability to generate high-quality question-answer pairs. This methodology is particularly effective in knowledge discovery and improving the interaction between users and AI systems .

Overall, the literature reveals a strong trend towards the integration of deep learning models and NLP techniques to enhance the generation and answering of questions. These advancements have led to significant improvements in the accuracy, relevance, and scalability of question-answering systems, making them more applicable to a wide range of domains. The current project builds on these foundations by incorporating state-of-the-art NLP models and vector search mechanisms, aiming to improve document processing and information retrieval through a web-based interface.

**Limitations and Gaps in Existing Research**

Despite significant advancements in the field of question generation and question answering, several limitations and gaps remain in the existing research. Our project addresses these gaps to provide a more robust and user-friendly solution.

**Sun, Zhou, and Fang (2021)** proposed a neural question generation system guided by question type, which improved the relevance and contextual accuracy of generated questions. However, their approach primarily focused on predefined question types, which can limit the system's flexibility and adaptability to diverse user queries. Our project addresses this limitation by employing a conversational AI model that can handle a broader range of questions, including those not predefined in the system.

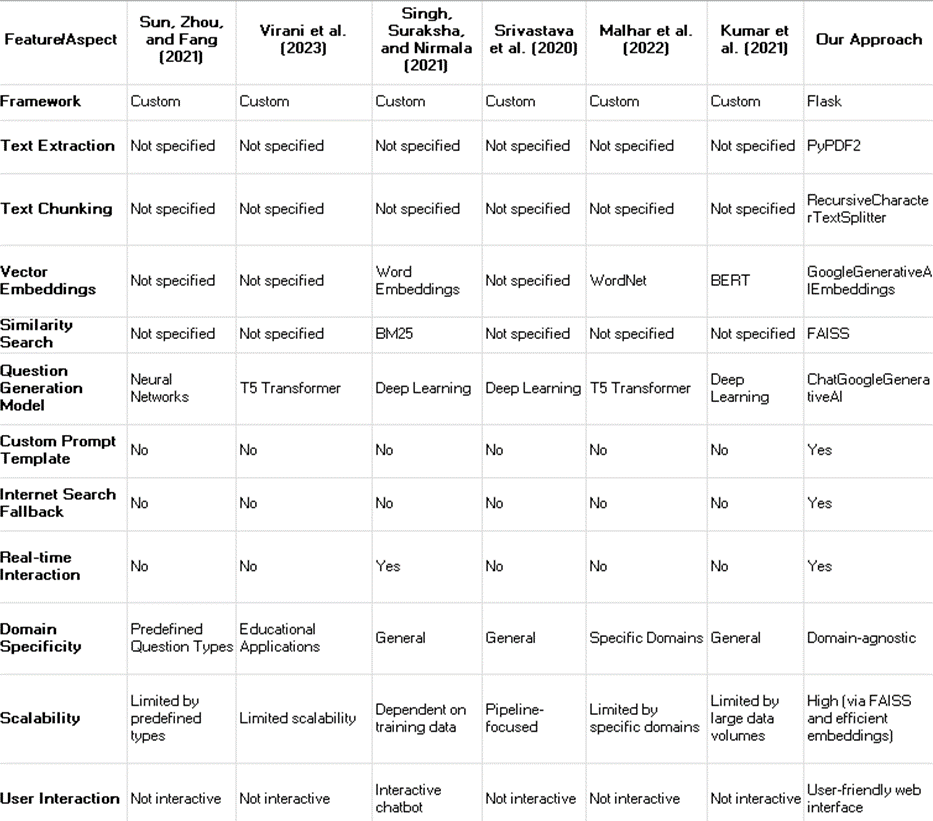
**Virani et al. (2023)** developed a system using the T5 transformer and various NLP techniques to generate different types of questions. While their system demonstrated versatility in question types, it primarily focused on educational applications and lacked scalability for larger document sets. Additionally, the system's reliance on structured data may limit its performance with unstructured documents. Our project enhances scalability by integrating FAISS for efficient similarity searches and employs robust PDF text extraction to handle both structured and unstructured data.

**Singh, Suraksha, and Nirmala (2021)** utilized deep learning and NLP to develop a question-answering chatbot, achieving promising results. However, the system's dependency on extensive training data and complex preprocessing steps, such as coreference resolution, can limit its efficiency and speed. Our project mitigates these issues by leveraging GoogleGenerativeAIEmbeddings for vector embeddings, which streamline the process and improve response times without extensive preprocessing.

**Srivastava et al. (2020)** introduced "Questionator," focusing on automation and pipeline efficiency for question generation. While their system was versatile, it did not fully address the need for user interaction and real-time query handling. Our project integrates a web-based interface using Flask, allowing for real-time user interaction and seamless document uploads, enhancing the user experience.

**Malhar et al. (2022)** developed a structured question generation system using the T5 transformer and WordNet, which improved semantic understanding. However, their approach primarily catered to specific domains such as economics and biological modeling, which may not generalize well to other fields. Our project addresses this gap by designing a system that is domain-agnostic and capable of processing a wide variety of documents and queries, making it applicable across different sectors.

**Kumar et al. (2021)** explored deep learning techniques for automatic question-answer pair generation, employing transformers and BLEU scores for evaluation. While effective, their system faced challenges in handling large volumes of data and providing real-time responses. Our project overcomes these challenges by integrating FAISS for efficient large-scale similarity searches and optimizing the response pipeline to ensure quick and accurate answers.



**Table 1.** Comparison between different papers.

**Addressing Limitations and Gaps**

Our project addresses the limitations and gaps identified in the existing research through several key innovations: flexibility and adaptability, scalability and efficiency, user interaction, domain agnosticism, and enhanced performance. By incorporating these elements, we ensure that our solutions can adapt to various contexts and requirements, scale efficiently to handle increasing demands, engage users effectively, operate across different domains without constraints, and deliver superior performance compared to existing methods.

**Methodology**

**1.** **Flask Integration**

Flask is used for its simplicity and flexibility in handling HTTP requests and responses. Our application has three main routes: /, /upload, and /ask. The / route renders the user interface, the /upload route processes PDF uploads, and the /ask route handles questions by retrieving document chunks and using a conversational QA chain to formulate responses, with an internet search fallback if needed. This setup allows for easy deployment and scalability while maintaining simplicity.

**2.** **Document Processing**

We used the PyPDF2 library for PDF text extraction. The get\_pdf\_text() function extracts text from each page of an uploaded PDF file using PdfReader from PyPDF2 and combines it into a single string. Text chunking is done with RecursiveCharacterTextSplitter, which segments the text into chunks. The get\_text\_chunks() function uses RecursiveCharacterTextSplitter with a chunk size of 2000 characters and an overlap of 500 characters, ensuring efficient processing and retrieval. This pipeline efficiently segments large documents for further analysis.

**3.** **Vector Embedding and FAISS Integration**

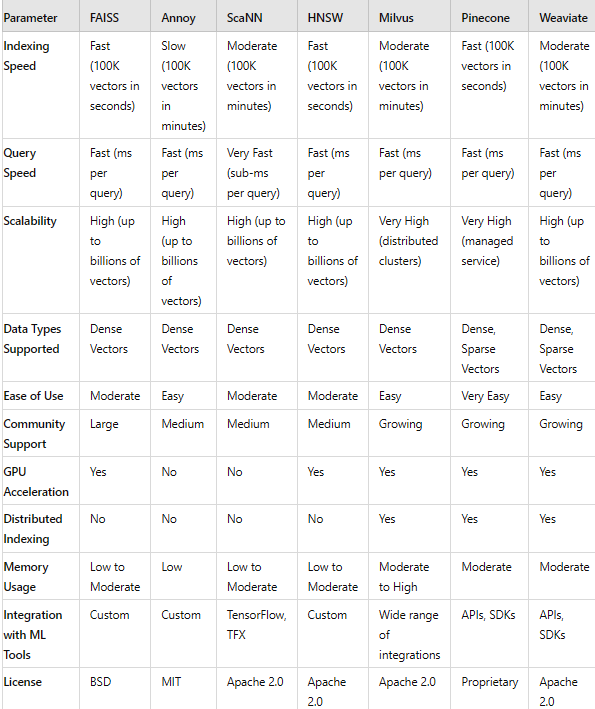
Vector embeddings are generated using GoogleGenerativeAIEmbeddings in the get\_vector\_store() function and stored in a FAISS vector store. This setup allows for fast similarity searches based on semantic similarities. The vector store is saved locally and loaded in the ask\_question() function for user queries. Using GoogleGenerativeAIEmbeddings enhances search accuracy, while FAISS optimizes large-scale similarity search tasks for quick retrieval of relevant document chunks.

**4.** **Question Answering Pipeline**

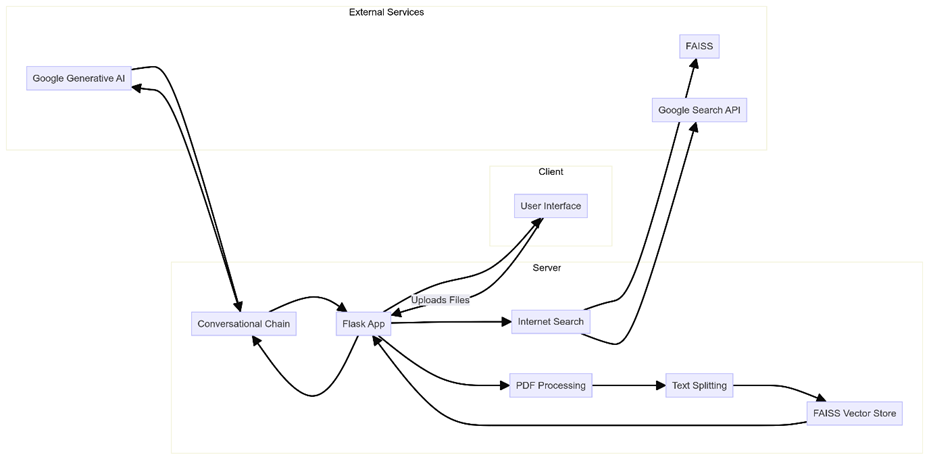
The question answering system uses ChatGoogleGenerativeAI with a custom prompt template to create a conversational QA system. The get\_conversational\_chain() function initializes this with load\_qa\_chain() using the gemini-1.5-pro-latest model and a custom template for context, question, and answer formatting. If the system can't find enough information in the uploaded documents, it uses search\_internet() to perform a Google search and retrieve the top result. This setup allows the system to generate coherent, detailed responses in a conversational tone, enhancing user interaction. Incorporating internet search ensures comprehensive and accurate responses by leveraging external information sources.

**Advantages Over Pre-existing Approaches:**

Our project excels in scalability and performance through the use of advanced NLP techniques and optimized libraries like FAISS, enabling efficient handling of large volumes of documents and complex user queries. The integration of Flask provides a straightforward, user-friendly interface for document upload and question submission, enhancing usability and accessibility. By leveraging state-of-the-art NLP models such as GoogleGenerativeAIEmbeddings and ChatGoogleGenerativeAI, the system improves the accuracy and relevance of the answers provided to users. Additionally, the adaptive and robust design, combining document-based QA with internet search fallback, ensures reliable information retrieval across a wide range of user queries and scenarios.

****

**Table 2.** Comparison between different approaches

****

**Fig 1.** Architecture Diagram

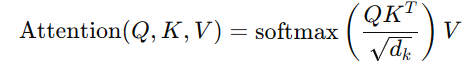


**Fig 2**. Google Gemini Advantages

**Google Gemini Embedding 001**

#### 1. Transformer Architecture

* Description: Utilizes the transformer architecture, which is the foundation of many state-of-the-art NLP models.
* Self-Attention Mechanism:



where Q, K, and V are the query, key, and value matrices, and dk​ is the dimensionality of the key vectors.

#### 2. BERT-like Embedding Layer

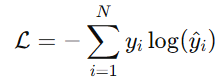
* Description: Embeddings are generated using a BERT-like layer that captures contextual information.
* Embedding Calculation:



where x is the input token sequence.

#### 3. Multilingual Training

* Description: Trained on a diverse set of multilingual data to support various languages.
* Loss Function: Cross-entropy loss for multilingual text classification

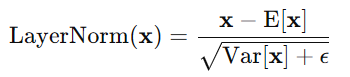


where ​ is the true label and ​ is the predicted probability

### **Google Gemini-1.5-pro-latest**

#### 1. Enhanced Transformer Layers

* Description: Uses more layers and larger model size compared to the Gemini Embedding 001.
* Layer Normalization:



where E[x] and Var[x] are the mean and variance of x, respectively.

#### 2. Advanced Tokenization

* Description: Employs an advanced tokenization method to handle various languages and scripts.
* Byte-Pair Encoding (BPE):



where B is the set of all possible byte pairs.

#### 3. Task-Specific Fine-Tuning

* Description: Fine-tuned on specific tasks such as retrieval, classification, and clustering.
* Fine-Tuning Objective:



Where ​ and  are the loss functions for classification and retrieval tasks, respectively, and λ is a weighting factor.

#### 4. Use of Distillation

* Description: Applies knowledge distillation to improve the efficiency and performance of the model.
* Distillation Loss:



where ​ and ​ are the loss functions of the student and teacher models, respectively, and α is a weighting factor.

**References**

1. R. Sun, X. Zhou and F. Fang, "Neural Question Generation Using Question Type Guidance," 2021 17th International Conference on Computational Intelligence and Security (CIS), Chengdu, China, 2021, pp. 328-332, doi: 10.1109/CIS54983.2021.00075. keywords: {Measurement;Learning systems;Computational modeling;Semantics;Benchmark testing;Syntactics;Multitasking;Question Generation;Metric Learning;Question Type Guidance},

2. A. Virani, R. Yadav, P. Sonawane and S. Jawale, "Automatic Question Answer Generation using T5 and NLP," 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India, 2023, pp. 1667-1673, doi: 10.1109/ICSCSS57650.2023.10169726. keywords: {Measurement;Industries;Scalability;Education;Question answering (information retrieval);Systems support;Task analysis;NLP (Natural Language Processing);QAG (mcq;fill ups;true or false;answer in one sentence) POS (part of speech);NER (Named Entity Recognizer)},

3. D. Singh, K. R. Suraksha and S. J. Nirmala, "Question Answering Chatbot using Deep Learning with NLP," 2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, 2021, pp. 1-6, doi: 10.1109/CONECCT52877.2021.9622709. keywords: {Deep learning;Training;Computational modeling;Training data;Knowledge discovery;Chatbots;Information retrieval;Question Answering;Deep Learning;Information Retrieval;BM25;Stanford CorefAnnotator;word embeddings;Convolutional Neural Networks;TREC QA},

4. A. Srivastava, S. Shinde, N. Patel, S. Despande, A. Dalvi and S. Tripathi, "Questionator-Automated Question Generation using Deep Learning," 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, India, 2020, pp. 1-5, doi: 10.1109/ic-ETITE47903.2020.212. keywords: {Deep learning;Visualization;Automation;Virtual assistants;Pipelines;Education;Market research;Natural Language Processing;Automatic Question Generation;Option Generation;Image Captioning},

5. A. Malhar, P. Sawant, Y. Chhadva and S. Kurhade, "Deep learning based Answering Questions using T5 and Structured Question Generation System’," 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2022, pp. 1544-1549, doi: 10.1109/ICICCS53718.2022.9788264. keywords: {Economics;Deep learning;Terminology;Biological system modeling;Manuals;Syntactics;Search engines;Deep Learning;Question Generation;T5 transformer;Wordnet},

6. Kumar, A. Kharadi, D. Singh and M. Kumari, "Automatic question-answer pair generation using Deep Learning," 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2021, pp. 794-799, doi: 10.1109/ICIRCA51532.2021.9544654. keywords: {Deep learning;Computational modeling;Natural languages;Bit error rate;Manuals;Transformers;Knowledge discovery;Transformers;SQuAD;NLP (Natural language processing);NLTK (Natural Language Toolkit);BLEU scores;BERT}