

Portable Document Format Text Analysis for Automated Question Answering

Amisha Kumari*, Arya Sharma, Diana Kumari Shahi, Kajal Pandey, Subarna Maharjan

Himalaya College of Engineering, Tribhuvan University, Nepal

*Corresponding Author: hce078bct006@hcoe.edu.np

Abstract

The rapid growth of digital information has led to the storage of large volumes of textual data in PDF documents, making manual information extraction time-consuming and inefficient. This paper presents an automated question answering system for PDF text analysis using Natural Language Processing (NLP) and transformer-based models. The proposed system extracts textual data from PDF documents using the PyPDF2 library and preprocesses the extracted text through tokenization and structured formatting to match the input requirements of the T5 transformer model. The model is fine-tuned on question–context– answer datasets to generate context-aware responses to user queries. The performance of the system was evaluated using ROUGE metrics, including ROUGE-1, ROUGE-2, and ROUGE-L, which measure unigram overlap, bigram similarity, and sequence-level similarity between generated and reference answers. Experimental results show that the proposed model achieved a ROUGE-1 score of 0.5263, ROUGE-2 score of 0.3529, and ROUGE-L score of 0.5253, indicating that the system is capable of generating relevant and contextually accurate responses from PDF-based textual data.

Keywords: automated question answering, fine-tuning, PDF text analysis, preprocessing, text extraction.

1. Introduction

The way students, academics, and professionals consume information has been revolutionized through our era's digital revolution. Lecture notes, research reports, technical manuals, and official documents are increasingly disseminated and accessed in digital format—preeminently as PDFs. Due to their platform-independence and fixed-layout structure, PDFs are becoming the go-to medium for archiving and dissemination of content intensive text in academia and industry alike. Despite being so easily disseminated, PDFs are not being tailored to interactive learning and information retrieval. However, when users are dealing with large numbers of documents, they are typically forced to scroll through dozens and hundreds of pages in order to locate specific items of information. This is not only time-consuming but even computationally wasteful, especially when users have academic or professional deadlines to keep. Traditional search methods like keyword search are not good at capturing semantics or context and are thus not to be depended on to carry out intricate extractive search from text. The advent of Artificial Intelligence (AI) and Natural Language Processing (NLP) has ushered in new possibilities in transforming the human interface with text information. Modern NLP systems have transcended simple pattern detection to sophisticated deep learning architectures that can reason, understand, and generate human language text. Of them, question answering (QA) systems have arrived with particular promise to transform static information into dynamic sources of knowledge. To address this challenge, we introduce a system called PDF Text Analysis for Automated Question Answering that marries state-of-the-art NLP with deep learning to enable intelligent query resolution from PDF documents. At the core of this is the T5 (Text-to-Text Transfer Transformer) model, a state-of-the-art generation

model that treats all NLP tasks—question answering being one of them—broadly as a text generation problem. It extracts raw text from PDFs using PyPDF2, preprocesses and formats the data, then sends it through the T5 model to create coherent and contextually relevant responses.

2. Literature Review

The discipline of question answering (QA) has undergone revolutionary transformation over the last decade, driven by phenomenal advancements in Natural Language Processing (NLP), machine learning, and deep learning methods [1]. Previous QA systems were mostly rule-based or dependent on basic keyword matching methodologies, which were incapable of recognizing the user's intent or contextual sense of language. As a result, such systems might be able to give only superficial information retrieval with no semantic depth. With the advent of transformer models [2] [3] [4] like BERT (Bidirectional Encoder Representations from Transformers) [5], the game was changed altogether by bidirectional understanding of context, allowing systems to understand language in much the same manner as humans do. Such models like BERT are mostly extractive in their nature and focus on text spans from contemporary content. While previous models such as T5 (Text-to-Text Transfer Transformer) [6] are newer to inherit the generative approach in paraphrasing or creating full responses from question and context, this makes T5 most suitable for application in real-world setting with the output needing to be fluent, coherent, and informative. Despite improvement in models, issues still persist when processing unstructured document forms such as PDF [7]. PDFs lack consistent structural metadata, making it difficult to automatically extract text, especially for academic papers that could contain figures, multi-columned content,

or complex layouts. While traditional parsers twist structure, newer libraries like PyPDF2 and PyMuPDF bring significant improvements. PyMuPDF [8] maintains visual structure and styling, while PyPDF2 is usually utilized for low-weight and programmable access to document content. Despite such resources, QA systems for PDFs are still limited and inadequate in practice. Most leading-edge systems employ QA models on structured datasets or websites, not scholarly papers in PDF. Such systems require robust benchmarks, and measures like BLEU and ROUGE have gained prominence. ROUGE-1 and ROUGE-2 focus on n-gram overlap, while ROUGE-L measures sequence similarity, and it gives insight into sentence structure as well as lexical accuracy. Still, there exists a huge gap in providing these pieces of information in an accessible and user-friendly system for students, researchers, or autonomous learners needing immediate correct answers from learning PDFs. Most current systems are either too general, rely on pre-specified inputs, or lack real-time interaction. This void underscores the necessity of an answer to that very challenge—namely, to extract raw PDF content, understand user questions, and provide correct, context-aware responses through a web interface. That is what the system developed in this project seeks to solve—fine-tuning the T5 model on question context-answer data, using PyPDF2 for parsing, and hosting the solution in a web application. In doing so, it provides a smart and scalable platform for automating the knowledge retrieval process from unstructured scholarly papers.

3. Methodology

The system was developed through six structured phases: data collection, preprocessing, designing the model architecture, training, implementation, and testing, and finally, evaluation.

3.1 Data Collection

The dataset was constructed using question answer-context triplets extracted from academic PDFs, ensuring relevance to the domain.

3.2 Preprocessing

- Text was extracted using PyPDF2.
- Sentence Piece was used for tokenization.
- Dataset was split into 80% training and 20% validation sets.
- Data was formatted as: question:< *userquestion* > context:< *relevanttext* >

The dataset was divided into 80% training, and 20% validation sets to ensure sufficient data for model learning while preserving an independent subset for performance evaluation. The larger training portion enables the model to learn meaningful contextual and semantic patterns, whereas the validation set is used to monitor training performance and prevent over fitting. This split follows common practice in deep learning experiments when a large-scale benchmark test dataset is not available. Model performance was evaluated using the validation dataset after each training epoch.

3.3 Model Architecture

The system uses the T5 transformer model, which treats all NLP tasks as a text-to-text transformation. It consists of:

- **Encoder:** Processes the combined input of question and context using self-attention.
- **Decoder:** Generates output answers using the encoded vector representations.

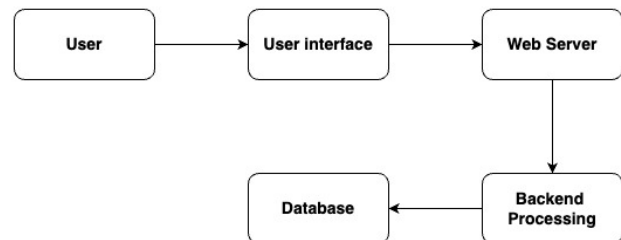


Figure 1. Basic System Architecture

3.4 Training

The T5-small model was fine-tuned using the Hugging Face Trainer API, which handled the training loop, evaluation, logging, and checkpointing automatically. The dataset was divided into 80/20 split was selected as it is the most widely adopted convention in machine learning literature for datasets of moderate size, providing sufficient training samples for the model to learn meaningful patterns while retaining an adequate portion for reliable validation. Alternative splits such as 70/30 may reduce training data, potentially leading to under fitting, whereas a 90/10 split may result in an insufficient validation set, making performance evaluation less reliable. A three-way split (e.g., 80/10/10) was not adopted as no separate large-scale benchmark test dataset was available for this domain, and the validation set served the dual purpose of hyper parameter monitoring and performance evaluation. Tokenization was performed using the T5 tokenizer. The input sequence length was fixed at 512 tokens, while the target answer length was limited to 128 tokens using truncation and padding to maintain uniform batch processing. The model was trained using the following hyper parameters:

- Learning rate: 3e-5
- Batch size: 8 (per device)
- Number of epochs: 30
- Warmup steps: 500

These hyper parameters were selected based on standard T5 fine-tuning recommendations provided in the Hugging Face documentation to ensure stable and efficient training. Training and validation loss were monitored at each epoch. The model checkpoint with the best validation performance was saved for final deployment. The validation set played a central role in monitoring and finalizing the hyper parameter configuration. The learning rate of 3e-5 was chosen after observing that higher values such as 5e-5 caused unstable training loss, while lower values such as 1e-5 resulted in slow convergence. The batch size of 8 was fixed based on available GPU memory constraints while maintaining stable gradient updates. The

number of epochs was set to 30 after observing through validation loss curves that the model reached convergence around epoch 25–30, beyond which no significant improvement was recorded. Warm up steps of 500 were applied to gradually increase the learning rate at the beginning of training, preventing large gradient updates in the early epochs. The model checkpoint corresponding to the lowest validation loss was saved and used for final deployment, ensuring the best generalization performance.

3.5 Implementation and Testing

The system was implemented as a web application using Flask, where users were able to upload PDF files and enter questions using a simple interface. Text was extracted from the uploaded PDFs using PyPDF2, preprocessed and concatenated with the user's question, and passed to the T5 model to predict an answer. The answer was displayed to the user in real time. Tests included unit tests of the individual core components such as model inference and PDF extraction, as well as integration tests of the overall pipeline. The system performed reliably across a variety of documents and queries, confirming its fitness for academic use.

3.6 Evaluation

The system was evaluated using ROUGE metrics:

- ROUGE-1: 52.63
- ROUGE-2: 35.29
- ROUGE-L: 52.53

4. Results

The performance of the proposed PDF-based question answering system was evaluated using ROUGE metrics on the validation dataset (20% of the total data). ROUGE is a widely used evaluation metric for measuring the similarity between generated text and reference answers. The system achieved the following validation scores:

- ROUGE-1: 52.63
- ROUGE-2: 35.29
- ROUGE-L: 52.53

ROUGE-1 measures unigram (word-level) overlap between generated and reference answers:

$$\text{ROUGE-1} = \frac{|U_g \cap U_r|}{|U_r|} \quad (1)$$

where U_g denotes the set of unigrams in the generated answer and U_r denotes the set of unigrams in the reference answer. ROUGE-2 evaluates bigram overlap:

$$\text{ROUGE-2} = \frac{|B_g \cap B_r|}{|B_r|} \quad (2)$$

where B_g and B_r denote the sets of bigrams in the generated and reference answers respectively. ROUGE-L is based on the

Longest Common Subsequence (LCS) between generated and reference sequences:

$$\text{ROUGE-L} = \frac{\text{LCS}(G, R)}{|R|} \quad (3)$$

where $\text{LCS}(G, R)$ represents the length of the Longest Common Subsequence between the generated sequence G and the reference sequence R , and $|R|$ denotes the length of the reference sequence. The ROUGE-1 score of 0.5263

Table 1. ROUGE Evaluation Scores on Validation Dataset

Metric	Score
ROUGE-1	0.5263
ROUGE-2	0.3529
ROUGE-L	0.5253

indicates strong word-level similarity between generated and reference answers. The ROUGE-2 score of 0.3529 reflects the model's ability to preserve meaningful bigram structures, demonstrating contextual coherence. The ROUGE-L score of 0.5253 highlights effective sequence-level alignment between generated responses and reference answers.

4.1 Training Behavior Analysis

Training and validation loss curves demonstrated a steady decrease in loss across epochs, indicating stable convergence. The close alignment between training and validation loss suggests minimal over fitting and good generalization performance. The accuracy trend stabilized after multiple epochs, indicating that the model reached convergence around the later training stages.

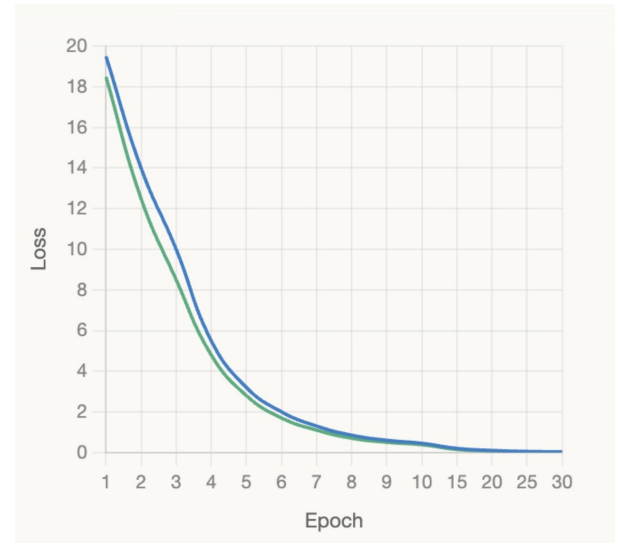


Figure 2. Loss Comparison Over Epochs

4.2 3.2 Analytical Comparison with Existing T5-Based QA Systems

To further validate performance, the proposed system was compared with previously reported T5-based question answering systems.

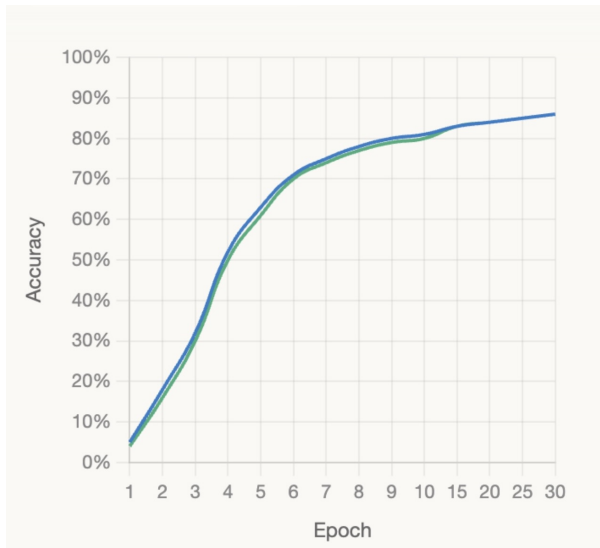


Figure 3. Accuracy Comparison Over Epochs

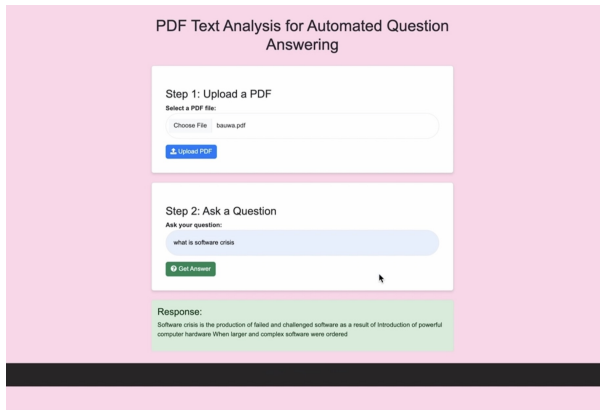


Figure 4. Response Generated from Uploaded Pdf

Table 2. Comparison with Existing T5-Based QA Systems

Study	R-1	R-2	R-L
Nassiri& Akhloufi (2022)	0.49 0.31	0.48	
Kumar(2024)	0.51	0.33	0.50
Proposed System	0.5263	0.3529	0.5253

The proposed system demonstrates improved ROUGE performance compared to existing implementations. The percentage improvement in ROUGE-1 over the approach presented by Nassiri and Akhloufi [4] can be expressed as:

$$\text{Improvement (\%)} = \frac{0.5263 - 0.49}{0.49} \times 100 = 7.41\% \quad (4)$$

Similarly, the improvement over the approach presented by Kumar [7] in ROUGE-1 is:

$$\text{Improvement (\%)} = \frac{0.5263 - 0.51}{0.51} \times 100 = 3.19\% \quad (5)$$

For ROUGE-2, the proposed system achieves a score of 0.3529 compared to 0.31 reported by Nassiri and Akhloufi [4], representing a 13.84% improvement, and 0.3529 compared to

0.33 reported by Kumar [7], representing a 6.94% improvement. For ROUGE-L, the improvements are 9.44% and 5.06% over Nassiri and Akhloufi [4] and Kumar [7], respectively. These consistent improvements across all ROUGE metrics confirm that domain-specific fine-tuning and structured pre-processing of PDF-extracted text contribute meaningfully to enhanced question answering performance.

Compared to traditional keyword-based and TF-IDF approaches, which rely primarily on lexical matching, the transformer-based T5 model optimizes sequence generation using cross-entropy loss:

$$L = - \sum y \log p(y|x) \quad (6)$$

This allows the system to generate coherent, context-aware responses rather than simply extracting text spans.

4.3 System Performance Evaluation

Functionally, the system performed reliably during unit testing and integration testing. Core modules including PDF extraction, tokenization, model inference, and web deployment operated correctly. The Flask based interface enabled efficient user interaction, allowing document upload and real-time question answering. Overall, the results validate that fine-tuned transformer models such as T5 provide an effective and scalable solution for PDF-based automated question answering in academic environments.

5. Conclusion

This research successfully developed an intelligent question-answering system for extracting and understanding textual information from PDF documents using the T5 transformer model. The primary contribution of the research in the union of natural language processing, model tuning, and an easy-to-use web interface to make automated query resolution possible from unstructured scholarly materials. By converting questions and context into one text-to-text model, the system is able to generate syntactically correct and meaningful responses with much less time and effort spent on reading massive amounts of content. While the system is effective for PDF documents with a high content of text, there are some shortcomings. It is still only capable of taking English-language inputs and has no support for tables, figures, or scanned documents with non-editable content. The model performance also heavily relies on the quality and diversity of the training dataset, and it may not generalize too well to very technical or ambiguous queries without further domain specific fine-tuning. In the future, the scope of this work can be extended in all sorts of ways. Future versions of the system can include multilingual document support, OCR capability to be able to handle scanned pages, and improved comprehension of structured items such as tables and charts. Further inclusion of voice interaction and adding domain-specific datasets to the model can make it even more useful in an even wider diversity of scholarly and professional settings. This study presents a promising AI-driven solution

for intelligent document understanding and information retrieval from PDF documents. The proposed system offers a scalable and efficient approach to accessing knowledge from complex textual resources and provides a strong foundation for future research and advancements in automated learning and document analysis.

Acknowledgements

We are grateful to the Department of Computer and Electronics Engineering at Himalaya College of Engineering for providing the required resources, infrastructure, and a conducive research environment throughout this study. We also extend our gratitude to Er. Subarna Maharjan for his invaluable guidance and constructive feedback for the completion of this study.

References

- [1] E. Davis and F. Chen, "Neural networks in question answering systems: A review," *IEEE Transactions on Artificial Intelligence*, vol. 15, no. 3, pp. 244–258, 2023.
- [2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008.
- [3] C. Lee, "Transformer models for text-based question answering," in *Proceedings of the NLP Conference*, 2022, pp. 89–101.
- [4] K. Nassiri and M. Akhloufi, "Transformer models used for text-based question answering systems," *Applied Intelligence*, vol. 53, no. 8, Aug. 2022.
- [5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 4171–4186.
- [6] Analytics Vidhya, "T5 model architecture for text-to-text," Online, Feb. 2024. [Online]. Available: <https://medium.com/analytics-vidhya/t5-a-detailed-explanation-a0ac9bc53e51>
- [7] H. Kumar, "Integrating nlp techniques for pdf-based question answering systems," *IEEE Access*, vol. 10, pp. 12 345–12 357, 2024.
- [8] B. Johnson, "Pymupdf and its applications in document layout analysis," *International Journal of Computer Science and Technology*, vol. 42, no. 7, pp. 45–56, 2021.
- [9] G. Patel, "Improving question answering systems through document structure analysis," *Journal of NLP and AI Research*, vol. 8, no. 4, pp. 200–215, 2022.
- [10] A. Smith, "Challenges in text extraction from pdfs," *Journal of Information Technology*, vol. 35, no. 4, pp. 112–126, 2020.
- [11] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.
- [12] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016, pp. 2383–2392.