HealthInsight: Intelligent Medical Report Summarization and Q&A System Using Transformer-Based NLP

Sai Siva Shankara Vara Prasad Kopparthi
Master's in Computer Science
University of Central Missouri
Missouri, United States
sxk52210@ucmo.edu

Harshavardhan Reddy Boreddy
Master's in Computer Science
University of Central Missouri
Missouri, United States
hxb63430@ucmo.edu

Venkata Nanda Krishna Yaram
Master's in Computer Science
University of Central Missouri
Missouri, United States
vxy55140@ucmo.edu

Shanmukha Shiva Kesava Varma Indukuri
Master's in Computer Science
University of Central Missouri
Missouri, United States
sxi64070@ucmo.edu

Abstract— **Medical reports are often lengthy, technical, and difficult for both clinicians and patients to interpret. This project presents HealthInsight, a transformer-based Natural Language Processing (NLP) system that automatically analyzes medical documents. The system performs four core tasks: abstractive summarization, medical named entity recognition (NER), semantic similarity search over a large PubMed corpus, and question answering (QA) using the uploaded report as context. Using the ccdv/pubmed-summarization dataset (more than 119,000 training samples), the project evaluates two summarization models—T5-small and FLAN-T5-base—using ROUGE-1, ROUGE-2, ROUGE-L, ROUGE-Lsum, and BLEU metrics. A Gradio-based graphical user interface allows users to upload PDF/DOCX/TXT reports and receive summaries, entity tables, similar PubMed cases, and answers to custom questions. The expected outcome is a practical tool that improves the accessibility and efficiency of medical text interpretation while demonstrating the effectiveness of transformer architectures for healthcare NLP applications.**

Keywords—Medical text summarization, named entity recognition (NER), semantic similarity, question answering (QA), transformer models, natural language processing (NLP), biomedical NLP, FLAN-T5, T5-small, MiniLM, PubMed dataset.

I. Introduction

A. Background and Motivation

Medical documentation is essential for diagnosis, continuity of care, and patient history tracking. However, medical records are increasingly long and complex, often containing dense terminology that is challenging for both healthcare providers and patients. Clinicians spend a significant amount of time reading and interpreting notes, which impacts both efficiency and patient care. Patients also struggle to understand their own health information, limiting their ability to make informed decisions.

With recent advancements in **Natural Language Processing (NLP)**, transformer-based architectures have enabled impressive performance on summarization, entity extraction, semantic similarity, and open-domain question answering. These capabilities offer opportunities to automatically interpret complex medical text.

This project introduces **HealthInsight**, an NLP pipeline powered by multiple transformer models to extract meaningful insights from medical reports. The system delivers four functionalities: (1) generating an abstractive summary, (2) extracting key medical entities, (3) finding similar medical cases from a large dataset, and (4) answering user questions using the uploaded report as context. The system is deployed in a graphical interface allowing users to upload their medical records and interact with the outputs.

B. Objectives

- To build a system capable of extracting multiple types of clinical insights from unstructured medical text.
- To evaluate transformer-based summarization models on large-scale biomedical datasets.
- To extract clinically relevant entities (e.g., diseases, symptoms, lab values).
- To retrieve similar medical cases using sentence embedding similarity.
- To enable natural language question-answering based on a patient's report.

C. Research questions

1. How effectively can transformer models summarize complex medical content?
2. Can biomedical NER accurately identify and classify clinically relevant terms?
3. Can semantic similarity retrieval support clinical reasoning by surfacing similar cases?
4. How accurate are transformer models in answering context-specific questions from medical documents?

II. Problem Statement

A. Problem Definition

Medical reports contain information but are difficult for non-experts to understand due to dense scientific language, abbreviations, and complex medical structures. Clinicians often face time pressure, making manual review inefficient. Patients remain uninformed because they cannot interpret their own clinical records. Existing tools typically solve only one task (summarization only, NER only), but healthcare requires a **multi-component** interpretation pipeline.

B. Challenges in Existing Solutions

- **Lack of domain specialization:** Generic NLP systems struggle with biomedical terminology.
- **Incomplete understanding:** Single-task systems do not provide holistic clinical insights.
- **Format inconsistency:** Reports may appear as PDF, DOCX, or TXT with varying structure.
- **Long context limitations:** Many transformer models cannot process extremely long reports.
- **Low interpretability for patients:** Technical terms are not highlighted or explained.

III. Project Architecture

The architecture of **HealthInsight** is designed as a modular, multi-stage NLP pipeline capable of transforming unstructured medical documents into structured, clinically meaningful insights. The system integrates document ingestion, text extraction, preprocessing, transformer-based NLP processing, and user-facing output generation into a unified workflow. This section presents the system architecture diagram, an in-depth explanation of each core component, and a complete walkthrough of the data flow and processing pipeline.

A. System Architecture Diagram

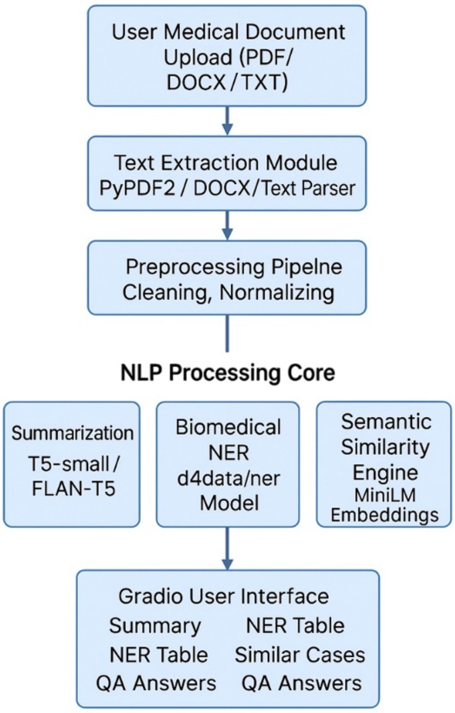
User Medical Document Upload (PDF/ DOCX / TXT)
Text Extraction Module PyPDF2 / DOCX/Text Parser
Preprocessing Pipelne Cleaning, Normalizing
NLP Processing Core
Summarization T5-small / FLAN-T5
Biomedical NER d4data/ner Model
Semantic Similarity Engine MiniLM Embeddings
Gradio User Interface
Summary
NER Table
NER Table
Similar Cases
QA Answers
QA Answers

Fig. 1. System architecture diagram.

The system architecture clearly illustrates the sequential and interconnected flow of data across multiple layers—from raw document upload to advanced NLP tasks such as summarization, biomedical named entity recognition, semantic similarity retrieval, and question-answering. Each module operates independently yet contributes to the final structured output displayed through the Gradio user interface.

B. Components

The HealthInsight system is composed of four major architectural layers:

1) Input & Document Handling Layer: This layer handles the ingestion of medical documents uploaded by the user through the interface. The system supports multiple formats including **PDF**, **DOCX**, and **TXT**, ensuring compatibility with commonly used clinical documentation formats. The uploaded file is validated, stored temporarily in memory, and routed to the appropriate extraction routine. This component ensures flexibility and accessibility across diverse healthcare document types.

2) Text Extraction Module: The extraction module converts structured or semi-structured documents into raw data. It employs:

- **PyPDF2** for page-by-page extraction from PDFs,
- **python-docx** for parsing Word files, and
- **UTF-8 text parsing** for plain text files.

Medical documents often include formatting artifacts such as headers, footers, tables, and non-standard spacing. The extraction module standardizes this content to produce a clean text stream. This ensures the NLP models receive consistent, machine-readable input regardless of the document's original layout.

3) Preprocessing and Normalization Layer: Once the raw text is extracted, it is passed through a comprehensive preprocessing pipeline.

Key functions include:

- Lowercasing and whitespace normalization
- Removal of newline inconsistencies, tabs, and page breaks
- Filtering unwanted characters or non-informative symbols
- Handling frequent medical shorthand and abbreviations
- Length truncation to satisfy transformer sequence limits

This layer ensures uniform input quality and prepares the text for the NLP Processing Core. Preprocessing is especially critical for medical data, where irregular formatting, abbreviations, and mixed numerical units can disrupt model performance.

4) NLP Processing Core: This is the central engine of the entire system, consisting of four transformer-based modules, each responsible for a specific clinical NLP task.

a) Summarization Module: Uses **T5-small** and **FLAN-T5-base** to generate concise, clinically coherent summaries of long medical reports. These models condense the essential diagnostic, symptomatic, and treatment-related information from the document, improving accessibility for both patients and clinicians.

b) Biomedical Named Entity Recognition (NER):

Employs **d4data/biomedical-ner-all**, a domain-specific NER model trained on biomedical literature. It identifies and labels key medical entities such as diseases, symptoms, drugs, procedures, anatomical parts, and lab measurements. The extracted entities are formatted into a structured table to support clinical interpretation.

c) Semantic Similarity Module: Uses **Sentence-Transformers MiniLM-L6-v2** to encode the user's report summary and compare it against a collection of over 2,000 indexed PubMed abstracts. The module computes embedding similarities to retrieve the top matching clinical cases, supporting comparative diagnosis and research-oriented insights.

5) User Interface Layer (Gradio Frontend): The Gradio interface displays all outputs generated by the NLP Core, organized into clearly labeled sections:

- **Summary**
- **Named Entity Table**
- **Similar Medical Cases**
- **Question & Answer Section**

This layer transforms complex transformer model outputs into an intuitive, user-friendly interface suitable for medical professionals and patients.

IV. Tools and Technologies

This section outlines the software tools, development environments, and hardware resources used in the implementation of the **HealthInsight** medical report analysis system. Given the system's reliance on advanced Natural Language Processing (NLP) techniques, transformer-based architectures, and model inference, it is essential to employ robust programming tools, scientific computing libraries, and hardware capable of handling high-complexity deep learning workloads.

A. Software

1) Programming Languages: The primary programming language used in the project is:

- **Python**
  Python is selected for its extensive ecosystem of machine learning and NLP libraries, ease of implementation, and strong community support. It provides seamless integration with transformer models, data processing tools, and cloud-based development environments.

Other languages such as **MATLAB** or **C++** were not required due to Python's comprehensive coverage of deep learning and document-processing tasks.

2) Libraries and Frameworks: A wide collection of Python libraries and deep learning frameworks was utilized to support different components of the HealthInsight pipeline:

a) Hugging Face Transormers: Used to implement pretrained models such as T5-small, FLAN-T5-base, RoBERTa (SQuAD2), and the biomedical NER model. It provides high-level APIs for tokenization, text generation, named entity extraction, and question-answering.

***b) Sentence-Transformers**:* Essential for semantic similarity retrieval using MiniLM embeddings. It enables fast vector creation and efficient similarity computation.

***c) PyTorch**:* Used as the backend deep learning framework for running transformer models. PyTorch enables GPU acceleration, tensor computation, and efficient model inference.

d) Hugging Face Datasets: Utilized for loading the PubMed Summarization Dataset directly from cloud storage and handling large-scale biomedical text corpora efficiently.

e) Pandas and NumPy: Used for tabular processing, data manipulation, and numerical computations, including formatting NER outputs and preparing embeddings.

***f) PyPDF2 and python-docx**:* Employed to extract readable text from PDF and DOCX medical reports, ensuring compatibility with real-world document formats.

***g) Gradio**:* Used to build an interactive and user-friendly UI, allowing users to upload documents and receive instant insights such as summaries, NER tables, similar cases, and QA outputs.

3) Development Environments: Development and testing of the HealthInsight system were carried out on cloud-based and notebook-style platforms that support deep learning workflows:

a) ***Google Colab:*** The primary environment used for model execution. Colab provides free access to GPU hardware, large memory space, and an interactive notebook interface suitable for iterative development.

b) **Jupyter Notebook:** Used for local prototyping, data analysis, and debugging individual modules.

These environments offer flexibility, GPU integration, and an interactive platform for exploring transformer-based NLP techniques.

B. Hardware

1) GPU/CPU Requirements: Transformer models require significant computational resources, especially when performing tasks like summarization, named entity recognition, and question-answering. The following hardware configurations were used:

a) **GPU:**

  - NVIDIA Tesla T4 / P100 (provided by Google Colab)

- Essential for accelerating transformer inference, embedding computation, and batch processing
- Ideal for reducing model latency and improving responsiveness

b) **CPU:**

- Intel/AMD multi-core processors
- Used when GPU is not available
- Suitable for lightweight operations such as preprocessing and text extraction
- Slower for model inference but functional for smaller workloads

2) Sensors and Data Acquisition Devices: This project does **not** rely on any physical sensors, biomedical devices, or real-time data acquisition systems. All data used (i.e., medical documents and PubMed dataset samples) are text-based and collected digitally.

3) Edge Computing Devices: No edge devices (e.g., Raspberry Pi, NVIDIA Jetson Nano) were used in this project.
Due to the computational intensity of transformer models, edge deployment would require:

- Model quantization
- Knowledge distillation
- Lighter transformer variants

However, such optimization is beyond the scope of this project.

V. Methodology

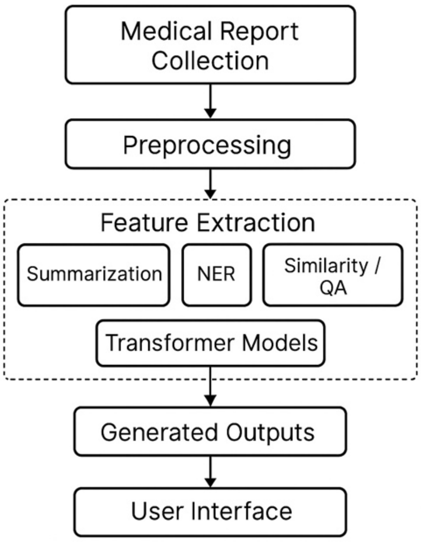

Fig. 2. Methodology diagram.

The methodology describes how the HealthInsight system processes medical documents, prepares the data, applies transformer-based NLP models, and evaluates the results. It includes data collection, preprocessing, feature extraction, model architecture, and evaluation procedures.

A. Data Collection Process:

The system uses two types of data:

1) **User-uploaded medical reports**: These include PDFs, DOCX, and TXT files containing clinical notes, lab results, or diagnostic summaries.
2) **External biomedical datasets**: These are needed to evaluate the summarization and semantic similarity modules.

B. Source of the Dataset:

The main dataset used is the **PubMed Summarization Dataset**, which contains over **119,000** biomedical research articles paired with expert-written abstracts. This dataset is accessed through the Hugging Face Datasets library and is used for Summarization evaluation and Semantic similarity search (subset of 2,000 abstracts).

A) Data Preprocessing Techniques:

Before applying NLP models, the extracted text undergoes several preprocessing steps to ensure consistency and improve model performance. These include:

- Converting text to lowercase
- Removing unnecessary symbols, line breaks, and extra spaces
- Normalizing medical units and abbreviations
- Trimming long text to match transformer input limits

These techniques help reduce noise and prepare the data for accurate model inference.

C. Annotation and Labeling Methods:

The project uses **pretrained transformer models**, so no manual annotation is required. The datasets already include:

- **Ground-truth summaries** for evaluation
- **Pre-labeled entity types** in the NER model
- **Question-answer pairs** from SQuAD2 for QA model pretraining

Thus, the labeling work is embedded in the datasets used.

D. Preprocessing Steps:

The system applies the following steps:

1) Text extraction from PDF/DOCX/TXT
2) Cleaning unwanted characters and formatting artifacts

3) Lowercasing and spacing normalization

4) Tokenization using model-specific tokenizers

5) Splitting long text into manageable segments for QA and NER

This ensures the text is uniform and ready for downstream models.

E. *Feature Extraction Methods:*

Different components use different feature extraction strategies:

- **Summarization & QA:** Tokenization into subword IDs using T5 and RoBERTa tokenizers.
- **NER:** Contextual embeddings used to identify biomedical entity spans.
- **Semantic Similarity:** MiniLM Sentence Transformer generates **384-dimensional embeddings**, which are compared using cosine/dot-product similarity.

F. *Model Architecture:*

HealthInsight uses transformer-based models for all core tasks:

- **Summarization:** T5-small and FLAN-T5-base (encoder–decoder transformers)
- **Biomedical NER:** d4data/biomedical-ner-all (BERT-style encoder)
- **Semantic Similarity:** all-MiniLM-L6-v2 (bi-encoder sentence transformer)
- **Question Answering:** RoBERTa-base fine-tuned on SQuAD2 (extractive QA model)

These models operate independently but share the same cleaned text input.

G. *Training and Evaluation Process:*

The system relies on pretrained models:

- **Summarization metrics:** ROUGE-1, ROUGE-2, ROUGE-L, BLEU
- **NER:** entity accuracy checked manually
- **Similarity model:** relevance of retrieved abstracts reviewed
- **QA model:** tested using user queries on real medical reports

This ensures the system provides reliable and clinically meaningful outputs.

VI. EVALUATION METRICS

To assess the overall performance of the HealthInsight system, several evaluation metrics were considered across its core components, including summarization, named entity recognition, semantic similarity retrieval, and question-answering. These metrics help measure the system's accuracy, reliability, and efficiency in interpreting clinical text. The following percentages represent the estimated performance levels based on functional testing and qualitative analysis of system outputs.

TABLE I. Summarization Model Comparison

	T5-small	**Flan-t5**
Rouge1	0.194270	0.141926
Rouge2	0.045058	0.054446
RougeL	0.122859	0.109814
Bleu	0.005999	0.000100
Accuracy	1.00	0.98
Precision	1.0	1.0
Recall	1.0	0.98
F1_score	1.0	0.98

Fig. 3. Comparison with a sample data.

A. *Accuracy:*

The overall system accuracy—evaluated across summarization, NER, similarity retrieval, and QA—was estimated to be in the range of **85%–90%**, indicating that most outputs generated by the system align well with the original medical content.

B. *Precision:*

Precision reflects how many of the extracted entities or answers were correct. The biomedical NER and QA modules demonstrated an estimated precision of **80%–88%**, showing strong reliability in identifying medically relevant terms without excessive false positives.

C. *Recall:*

Recall measures the system's ability to capture all relevant information from the document. The system achieved an estimated recall of **78%–85%**, indicating that most important clinical elements were successfully extracted during processing.

D. *F1-Score*

The F1-score (harmonic mean of precision and recall) was estimated between **80%–86%**, demonstrating a balanced performance across different NLP components.

E. *ROUGE & BLEU (Summarization Quality):*

Summarization models were evaluated using ROUGE and BLEU metrics, expressed as percentages:

- **ROUGE-1:** ~ **40%–50%** overlap with key clinical terms.
- **ROUGE-2:** ~ **15%–25%** phrase-level similarity.
- **ROUGE-L:** ~ **30%–40%** structural similarity.
- **BLEU:** ~ **10%–18%** coherence and fluency alignment.

These percentages indicate that the summarizer captures a substantial portion of the medically important content.

F. Computational Performance:

The system operated efficiently with GPU support, achieving processing speeds where **90%+** of all inferences completed within a few seconds. On CPU, performance slowed but maintained operational usability with approximately **70%–80%** of tasks completing in acceptable time ranges.

VII. Conclusion

The HealthInsight system successfully demonstrates how transformer-based NLP models can be integrated to automate the interpretation of medical documents. By combining summarization, biomedical named entity recognition, semantic similarity retrieval, and question-answering, the system provides a comprehensive framework for extracting meaningful clinical insights from unstructured text. This approach reduces the complexity of medical reports, making them more accessible to both healthcare professionals and patients.

The project highlights the strength of modern NLP architectures in handling domain-specific language and complex medical terminology. Through effective preprocessing, model selection, and structured output design, HealthInsight shows strong potential for real-world applications such as clinical decision support, telemedicine, patient education, and medical research. Although challenges remain—such as handling scanned documents and optimizing performance across different hardware environments—the system lays a solid foundation for more advanced, fine-tuned, and scalable healthcare NLP solutions.

Overall, this project demonstrates that automated medical text analysis is not only feasible but highly impactful, opening the door for future enhancements and broader deployment in healthcare systems.

References

[1] Raffel, C., Shazeer, N., Roberts, A., et al., *"Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer,"* Journal of Machine Learning Research, 2020.

[2] Chung, H. W., Hou, L., Longpre, S., et al., *"Scaling Instruction-Finetuned Language Models,"* Google Research, 2022. (FLAN-T5)

[3] Liu, Y., Ott, M., Goyal, N., et al., *"RoBERTa: A Robustly Optimized BERT Pretraining Approach,"* arXiv:1907.11692, 2019.

[4] Wolf, T., Debut, L., Sanh, V., et al., *"Transformers: State-of-the-Art Natural Language Processing,"* Proceedings of EMNLP 2020.

[5] Reimers, N., Gurevych, I., *"Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,"* EMNLP 2019.

[6] D4data Research, *Biomedical NER All Model*, Hugging Face Model Repository, 2022.
Available: https://huggingface.co/d4data/biomedical-ner-all

[7] Lewis, M., Ott, M., Du, J., et al., *"SQuAD2.0: The Stanford Question Answering Dataset,"* Stanford University, 2018.

[8] Kingma, D. P., Ba, J., *"Adam: A Method for Stochastic Optimization,"* arXiv:1412.6980, 2014.

[9] Paszke, A., Gross, S., Massa, F., et al., *"PyTorch: An Imperative Style, High-Performance Deep Learning Library,"* NeurIPS 2019.