MRI Impression Assistant

By

VRAJKUMAR NANDWANA (92200133018)

Under the guidance of

Prof. C.D. Parmar

A Project Submitted to

Marwadi University in Partial Fulfillment of the Requirements for the subject

Capstone Project (01CT0715)

September 2025

MARWADI UNIVERSITY
Rajkot-Morbi Road, At & Po. Gauridad,
Rajkot-360003, Gujarat, India

Abstract

Medical reports, especially MRI findings, often contain detailed technical descriptions that are difficult for patients and sometimes time-consuming for doctors to interpret. This project focuses on developing an automated system for **MRI report impression generation** using **BART (Bidirectional and Auto-Regressive Transformer)**, a transformer-based sequence-to-sequence model. The system processes raw MRI report text through a preprocessing pipeline (cleaning, tokenization, and formatting) and then generates concise impressions that summarize the essential clinical findings.

The proposed tool aims to assist radiologists by reducing repetitive documentation effort and improving efficiency, while also making reports more readable for non-specialists. Implementation involves text preprocessing, model fine-tuning with medical datasets, and integration into an interactive interface for testing. Initial results indicate that the approach can generate accurate, coherent, and clinically meaningful summaries, highlighting its potential to support medical professionals in routine practice.

Index

CHAPTER 1

1.1 Introduction

Medical imaging plays a vital role in the diagnosis and treatment of various health conditions. Among these, **Magnetic Resonance Imaging (MRI)** reports provide detailed observations of anatomical and pathological findings. However, such reports are often lengthy, highly technical, and written in complex medical terminology. This creates challenges for both radiologists, who spend significant time documenting findings, and patients, who may struggle to interpret the information.

To address these challenges, **automated impression summarization** has emerged as an important application of Natural Language Processing (NLP). The *impression* section of a radiology report condenses critical findings into a concise summary, highlighting the most relevant diagnostic details. By automating this process, the burden on radiologists can be reduced, consistency in reporting can be improved, and patients can gain clearer insights into their medical condition.

This project applies the **BART (Bidirectional and Auto-Regressive Transformer)** model, a state-of-the-art sequence-to-sequence architecture designed for text generation tasks such as summarization. The system processes raw MRI findings, applies preprocessing steps (text cleaning, tokenization, formatting), and generates structured impressions. Through this approach, the project demonstrates the potential of AI-powered summarization tools to enhance efficiency, reduce errors, and improve accessibility in the medical reporting workflow.

1.2 Problem Statement

Magnetic Resonance Imaging (MRI) is one of the most widely used diagnostic techniques in modern healthcare. After each scan, radiologists must prepare an **impression** — a concise summary of the key abnormalities and clinical relevance. However, this process comes with several challenges:

- **High Workload**: Radiologists are required to handle a large number of cases daily, which increases pressure and reduces time available for each report.
- **Complexity of Findings**: MRI reports are often lengthy and technical, making it difficult to summarize important details quickly.
- **Inconsistency**: Impressions can vary between radiologists, which sometimes causes miscommunication or delays in diagnosis.
- **Shortage of Radiologists**: In many countries, including India, there is a low ratio of radiologists to patients, adding to the reporting burden.

These challenges highlight the need for a **technological solution** that can assist radiologists in summarizing MRI findings more efficiently and consistently. With recent progress in **Artificial Intelligence (AI), Natural Language Processing (NLP), and Generative AI**, it is possible to build systems that automatically generate clinically accurate and professional impressions. Such a system can save time, reduce variability, and support radiologists in improving diagnostic outcomes.

1.3 Features

1. Input Methods

- **Direct Text Input:** Allows users to simply copy and paste MRI findings directly into a text area for quick processing.
- **PDF Report Upload:** Users can upload complete MRI report PDFs. The system automatically identifies and extracts only the relevant "Findings" section, streamlining the workflow.

2. Smart Input Validation

- Before processing, the application uses a quick AI check to validate that the provided text is relevant to clinical MRI findings, preventing errors and misuse with irrelevant data.

3. User-Configurable Model Parameters

- Provides an "Advanced Model Settings" section where users can adjust key generation parameters for the BioBART model, such as **Beam Size**, **Minimum Length**, and **Maximum Length** of the output.

4. Automated Impression Generation (BioBART Core)

- Utilizes a fine-tuned BioBART model, specialized in biomedical text, to automatically generate a concise "Raw Impression" from detailed MRI findings. This provides a rapid, AI-driven first draft.

5. AI-Powered Expert Refinement (GPT-4 Integration)

- Employs an advanced GPT-4 model to act as an "expert reviewer." It synthesizes the original findings and the raw BioBART draft to produce a comprehensive, professionally formatted, and clinically nuanced "Enhanced Impression."

6. Side-by-Side Comparison

- The user interface is structured with separate tabs, allowing for a clear, side-by-side comparison between the initial "Raw Impression" and the final "Enhanced Impression."

7. One-Click Report Download

- The final "Enhanced Impression" can be instantly downloaded as a .txt file, making it easy to save, share, or integrate into other clinical documentation systems.

1.4 System Architecture & Workflow

The overall architecture of the proposed system is designed to ensure smooth data flow from input to output, with well-defined processing stages. The workflow can be divided into the following steps:

1. **Data Acquisition** :Input can be provided in two ways:
    - **Text Input**: Radiologist pastes MRI findings directly into the interface.
    - **PDF Upload**: A report file is uploaded, and the system extracts relevant content.
2. **Preprocessing & Configuration**
    - Extracted findings are cleaned and formatted.
    - Users can adjust impression generation parameters (beam size, min/max length) through the sidebar.
3. **Draft Impression Generation (BioBART)**
    - The findings are passed to the fine tuned BioBART-based model.
    - The model generates a **draft impression** summarizing the key points.
4. **Refinement (GPT Integration)**
    - GPT model takes the draft impression and original findings.
    - Performs refinement:
    1. Ensures completeness of critical details.
    2. Improves clarity, readability, and professional tone.
    3. Outputs a structured, numbered **enhanced impression**.
5. **Output & Download:** Results are displayed in two tabs:
    - **Raw Impression (BioBART)**
    - **Enhanced Impression (GPT)**
        - Users can download the enhanced impression as a .txt file for clinical or record-keeping purposes.

1.5 AI Techniques Used

1.Generative AI:

- **Models:**
    - BioBART (fine-tuned)
    - GPT-4 (via Azure)
- **Purpose:**
    - To generate new, original clinical text (both the raw and enhanced impressions) by synthesizing information from the detailed findings.

2.Transfer Learning & Fine-Tuning:

- **Model:**
  - BioBART
- **Purpose:**
  - To adapt a powerful, pre-trained biomedical language model to the highly specialized task of creating MRI impressions from findings, saving significant training time and data.

3.Natural Language Generation (NLG):

- **Models:**
  - BioBART
  - GPT-4
- **Purpose:**
  - BioBART generates the initial draft impression, while GPT-4 generates the final, polished, and comprehensive report.

4.Text Classification:

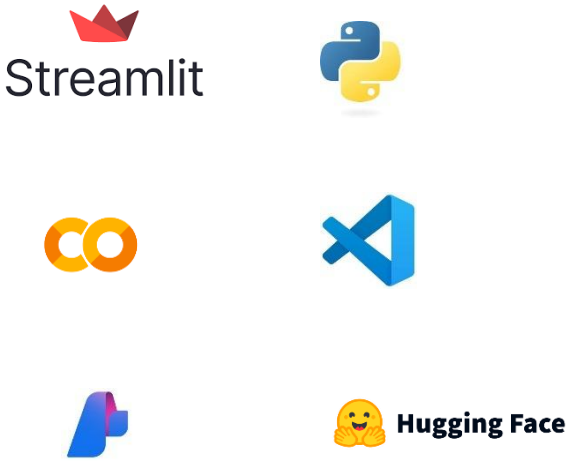
- **Model:**
  - GPT-4
- **Purpose:**
  - To classify input text as either "MRI Findings" or "Not MRI Findings," acting as a smart filter to ensure data relevance.

5.Information Extraction:

- **Tool:**
  - unstructured.io library
- **Purpose:**
  - To intelligently parse PDF documents and extract only the relevant "Findings" section, separating it from other parts of the report.

1.6 Technology

- **Frontend Technology**
  - **Streamlit**
- **Backend Technology**
  - **Python**
  - **Unstructured.io**
- **Tools**
  - **VScode**
  - **Colab**
  - **Github**
  - **Azure Ai Foundary**
  - **Hugging Face**



1.7 Deployment And Operation

Platform Selection

For the live deployment of the MRI Impression Assistant, we chose Hugging Face Spaces as our hosting platform. This decision was made for a few key reasons. First, its integration with the Hugging Face Hub, where our fine-tuned BioBART model is stored, is seamless.

- Second, Spaces offers native support for Streamlit applications, which meant we wouldn't have to manage complex server configurations.
- Finally, its generous free tier provided a perfect, cost-effective environment for hosting and sharing our academic project. This allowed us to focus on the application's functionality rather than on complex cloud infrastructure.

Deployment Architecture

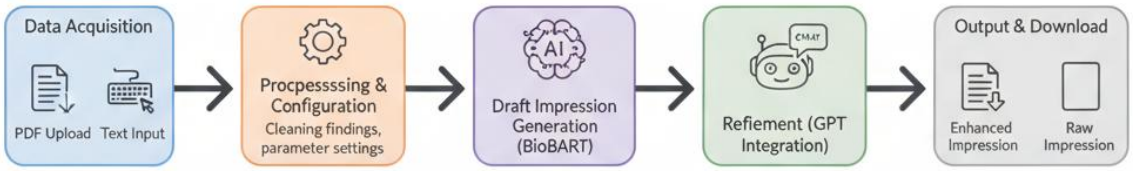
Our deployment strategy was built on the principle of separating the large model from the application code.

- Model : Our fine-tuned BioBART model, which is approximately 675 MB, is hosted on its own Hugging Face Model Repository. This keeps the model's files organized and version-controlled.
- Code : Our application code, including the app.py script and the requirements.txt file, is hosted in a separate Hugging Face Space.

When the application starts, it dynamically downloads the model from the Hub. This hybrid architecture ensures that our application is lightweight and that the large model is managed efficiently.

CHAPTER 2

2.1 Flow chart



2.2 User Manual

User Manual: MRI Impression Assistant

Welcome! This guide will walk you through the features and steps to effectively use the AI-Powered MRI Impression Assistant. This tool is designed to help you quickly and accurately generate radiological impressions from detailed MRI findings.

1. Getting Started

To access the application, simply open your web browser and navigate to the provided URL. The application runs entirely in the cloud, so no installation is required.

- **URL:** [https://huggingface.co/spaces/Vrajk/mri]

You will be greeted by the main interface, ready to accept your input.

2. How to Generate an Impression

The process is designed to be simple and is broken down into three steps on the main screen.

Step 1: Choose Your Input Method

First, you need to tell the assistant how you want to provide the MRI findings. You have two options:

- **Text Input:** Choose this if you have the findings text copied and ready to paste.
- **PDF Upload:** Choose this if you have the findings in a standard PDF report file.

Step 2: Provide the Findings

- **If you chose "Text Input":** A large text box will be visible. Simply paste the entire "Findings" section of your MRI report into this box.
- **If you chose "PDF Upload":** A file uploader will appear. Click "Browse files" to select the PDF report from your computer, or simply drag and drop the file onto the uploader. The system will automatically extract the relevant text.

Step 3: Generate the Report

Once your text is in the input box, click the **" Generate & Enhance Impressions"** button.

The application will now perform its multi-step AI process. You will see status messages like "Validating input..." and "Generating..." appear. Please be patient, as this can take a minute or two.

3. Reviewing the Results

After the generation is complete, the results will appear on the same page, organized into two columns for easy comparison.

- 🔹 **Raw Impression (BioBART):** This column shows the initial draft generated by the specialized BioBART model. It is a concise summary of the most critical findings.
- 🤖 **Enhanced Impression (GPT):** This column shows the final, polished report. This version has been reviewed and completed by the expert GPT model to be more comprehensive, accurate, and professionally formatted.

CHAPTER 3

3.1 Code Snippets and screenshots

```
# app.py
import streamlit as st
from transformers import AutoTokenizer, AutoModelForSeq2SeqLM
import torch
from dotenv import load_dotenv
import os
from openai import AzureOpenAI
from unstructured.partition.pdf import partition_pdf
import tempfile
import re

# ---- Load environment variables ----
load_dotenv()
AZURE_OPENAI_API_KEY = os.getenv("AZURE_OPENAI_API_KEY")
AZURE_OPENAI_ENDPOINT = os.getenv("AZURE_OPENAI_ENDPOINT")
AZURE_OPENAI_API_VERSION = os.getenv("AZURE_OPENAI_API_VERSION")
AZURE_OPENAI_DEPLOYMENT_NAME = os.getenv("AZURE_OPENAI_DEPLOYMENT_NAME")

# ---- Session State Initialization ----
if "raw_impression" not in st.session_state:
    st.session_state.raw_impression = ""
if "enhanced_impression" not in st.session_state:
```

```
    st.session_state.enhanced_impression = ""

# ---- FINAL VERSION: Model config for deployment ----
#MODEL_DIR = "Vrajk/mri-impressions"
MODEL_DIR = "./biobart-mri"
DEVICE = "cpu"

@st.cache_resource(show_spinner="Loading MRI Impression Generation model...")
def load_model():
    tokenizer = AutoTokenizer.from_pretrained(MODEL_DIR)
    model = AutoModelForSeq2SeqLM.from_pretrained(
        MODEL_DIR,
        device_map=None,
        dtype=torch.float32
    )
    model.to(DEVICE)
    return tokenizer, model

tokenizer, model = load_model()


# --- FINAL VERSION: validate input text ---
def is_valid_mri_findings(text: str) -> bool:
    """Uses GPT to quickly check if the text is relevant."""
    if not AZURE_OPENAI_API_KEY:
        return True # Skip check if API is not configured

    client = AzureOpenAI(
        api_key=AZURE_OPENAI_API_KEY,
        api_version=AZURE_OPENAI_API_VERSION,
        azure_endpoint=AZURE_OPENAI_ENDPOINT
    )
     prompt = f'Is the following text a clinical description of MRI findings? Respond with only "YES" or "NO".\n\nTEXT:
"{text[:1000]}"'
    try:
        response = client.chat.completions.create(
            model=AZURE_OPENAI_DEPLOYMENT_NAME,
            messages=[{"role": "user", "content": prompt}],
            temperature=0.0,
            max_tokens=5
        )
        answer = response.choices[0].message.content.strip().upper()
        return "YES" in answer
    except Exception:
        return True # Default to true if the validation check fails

def generate_impression(text, min_len, max_len, beams):
    inputs = tokenizer(text, return_tensors="pt", max_length=1024, truncation=True)
    inputs = {k: v.to(DEVICE) for k, v in inputs.items()}
    summary_ids = model.generate(
        inputs["input_ids"],
        attention_mask=inputs["attention_mask"],
        max_length=max_len,
        min_length=min_len,
        num_beams=beams,
        early_stopping=True
    )
    return tokenizer.decode(summary_ids[0], skip_special_tokens=True)

# --- FINAL VERSION: Upgraded GPT enhancement function ---
def enhance_with_gpt(raw_impression: str, original_findings: str):
    client = AzureOpenAI(
        api_key=AZURE_OPENAI_API_KEY,
        api_version=AZURE_OPENAI_API_VERSION,
```

```
        azure_endpoint=AZURE_OPENAI_ENDPOINT
    )
    prompt = f"""
    You are an expert radiologist writing a final impression for an MRI report.
    You have the full "FINDINGS" section and a "DRAFT IMPRESSION" from a junior AI.
    Your task is to create a comprehensive final impression.
    - Review the FULL FINDINGS carefully.
    - Use the DRAFT IMPRESSION as a guide, but you MUST add any clinically significant details from the FULL FINDINGS that the draft missed.
    - The final output should be a concise, numbered list, which is standard for radiological reports.
    - **CRITICAL:** Return ONLY the final, numbered impression text, without any conversational text or explanation.

    **FULL FINDINGS:**
    ---
    {original_findings}
    ---

    **DRAFT IMPRESSION:**
    ---
    {raw_impression}
    ---

    **FINAL IMPRESSION:**
    """
    response = client.chat.completions.create(
        model=AZURE_OPENAI_DEPLOYMENT_NAME,
        messages=[{"role": "user", "content": prompt}],
        temperature=0.2,
        max_tokens=300
    )
    return response.choices[0].message.content.strip()

def extract_findings_from_pdf(pdf_file):
    with tempfile.NamedTemporaryFile(delete=False, suffix=".pdf") as tmp_file:
        tmp_file.write(pdf_file.getvalue())
        tmp_path = tmp_file.name
    elements = partition_pdf(filename=tmp_path)
    os.remove(tmp_path)
    full_text = "\n".join([el.text for el in elements if el.text])
    impression_keywords = ["Impression", "IMPRESSION"]
    findings_start_patterns = ["SEQUENCES:", "HISTORY:", "FINDINGS:"]
    start_idx = 0
    for pat in findings_start_patterns:
        idx = full_text.find(pat)
        if idx != -1:
            start_idx = idx + len(pat)
            break
    end_idx = len(full_text)
    for kw in impression_keywords:
        idx = full_text.find(kw)
        if idx != -1:
            end_idx = min(end_idx, idx)
    findings_text = full_text[start_idx:end_idx].strip()
    return re.sub(r"(\n\s*)+\n+", "\n", findings_text).strip()

# ---- Streamlit UI ----
st.set_page_config(page_title="MRI Impression Assistant", layout="wide")
st.markdown("<h1 style='text-align:center; color:#4B0082;'>🧠 Magnetic Resonance Imaging Impression Assistant</h1>", unsafe_allow_html=True)
st.markdown("---")
with st.expander("🧠 Know About MRI"):
    st.markdown("""#### What is an MRI?
```

MRI stands for **Magnetic Resonance Imaging**. It's a safe and powerful medical imaging technique that uses a strong magnet and radio waves to create incredibly detailed pictures of the inside of the body. Unlike X-rays or CT scans, it does not use any radiation. It is especially good at showing soft tissues like the brain, muscles, ligaments, and organs.

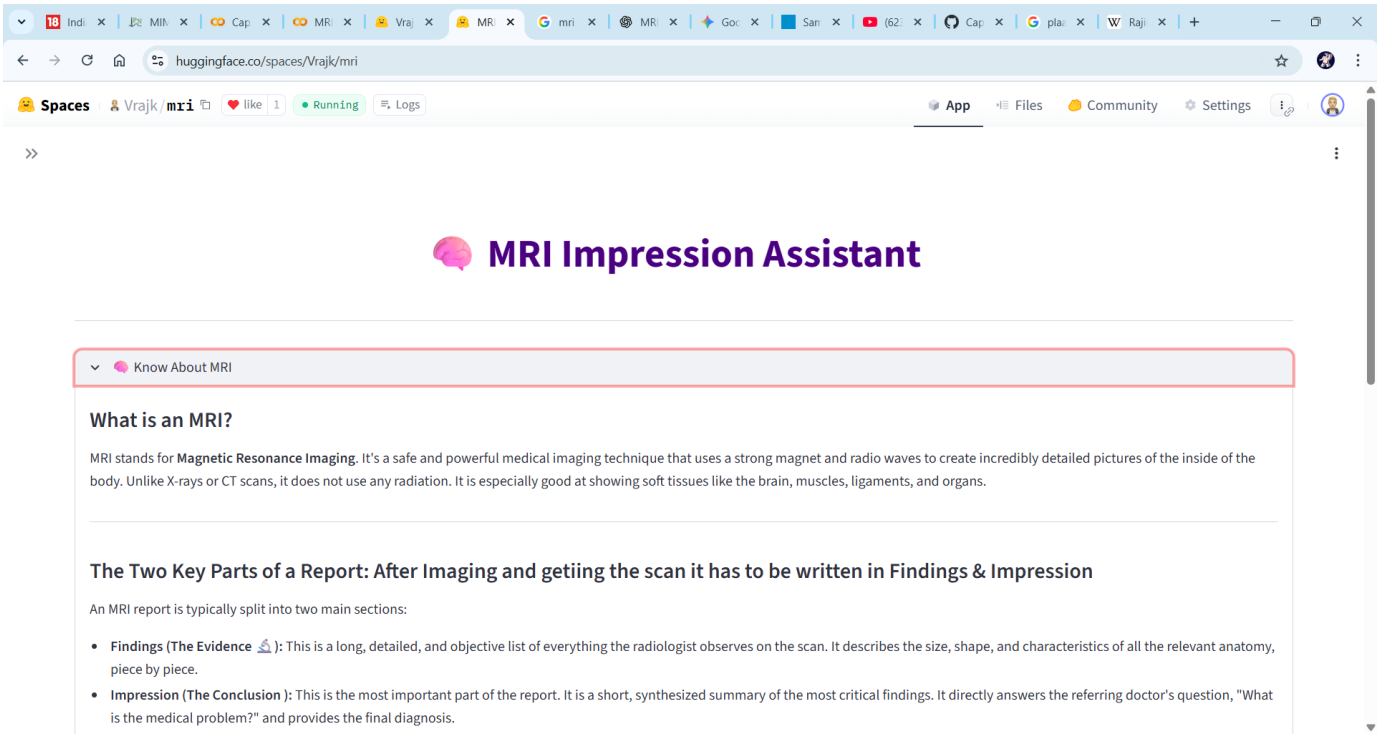
The Two Key Parts of a Report: After Imaging and getiing the scan it has to be written in Findings & Impression
An MRI report is typically split into two main sections:

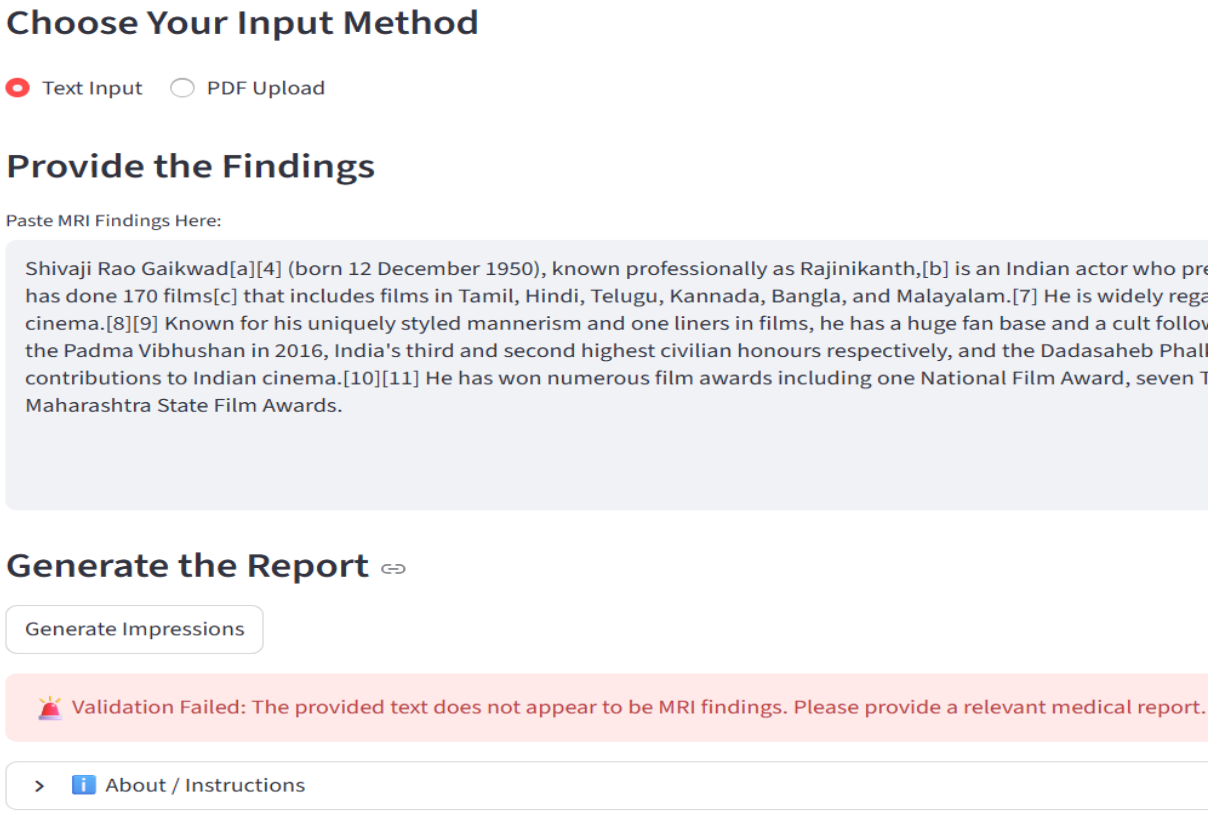
* **Findings (The Evidence 🔬):** This is a long, detailed, and objective list of everything the radiologist observes on the scan. It describes the size, shape, and characteristics of all the relevant anatomy, piece by piece.
* **Impression (The Conclusion):** This is the most important part of the report. It is a short, synthesized summary of the most critical findings. It directly answers the referring doctor's question, "What is the medical problem?" and provides the final diagnosis.
"""")

```
with st.sidebar:

    with st.expander("⚙️ Advanced Model Settings"):
        beam_size = st.slider("Beam Size", min_value=4, max_value=10, value=6)
        min_len = st.number_input("Min Impression Length", min_value=80, max_value=180, value=120)
        max_len = st.number_input("Max Impression Length", min_value=110, max_value=400, value=300)

tab1, tab2, tab3 = st.tabs(["💡 Input", "🔹 Raw Impression", "🤖 Enhanced Impression"])

with tab1:
    st.header("Input MRI Findings")
    st.info("Follow the steps below to generate a report.", icon="ℹ️")
    st.subheader("Choose Your Input Method")
    input_option = st.radio("Select Input Type:", ["Text Input", "PDF Upload"], horizontal=True, label_visibility="collapsed")

    st.subheader("Provide the Findings")
    text_input = ""
    if input_option == "Text Input":
        text_input = st.text_area("Paste MRI Findings Here:", height=250, placeholder="Enter MRI findings...")
    elif input_option == "PDF Upload":
        pdf_file = st.file_uploader("Upload an MRI Report PDF", type=["pdf"])
        if pdf_file:
            with st.spinner("Extracting text from PDF..."):
                extracted_text = extract_findings_from_pdf(pdf_file)
                st.success("✅ Findings extracted from PDF")
                text_input = st.text_area("Edit or Add to the Extracted Findings:", value=extracted_text, height=250)

    st.subheader("Generate the Report")
    if st.button("Generate Impressions", key="generate_btn"):
        if text_input and text_input.strip():
            with st.spinner("Validating input text..."):
                if is_valid_mri_findings(text_input):
                    st.toast("Input is valid. Generating...", icon="✅")
                    with st.spinner("Generating raw impression..."):
                            st.session_state.raw_impression = generate_impression(text_input, min_len=min_len, max_len=max_len,
beams=beam_size)

                    if AZURE_OPENAI_API_KEY:
                        with st.spinner("Enhancing impression with GPT..."):
                            st.session_state.enhanced_impression = enhance_with_gpt(st.session_state.raw_impression, text_input)

                    st.toast("Impressions generated successfully! see the raw impressions", icon="🎉")
                else:
                    st.error("Validation Failed: The provided text does not appear to be MRI findings. Please provide a relevant medical
report.", icon="🚨")
        else:
            st.warning("Please provide MRI findings before generating.")

with tab2:
```

```
    st.header("Raw Impression From Findings")
    if st.session_state.raw_impression:
                                    st.markdown(f"<div          style='background-color:#E6E6FA;          padding:15px;          border-
radius:10px;'>{st.session_state.raw_impression}</div>", unsafe_allow_html=True)
    else:
        st.info("The raw impression generated by our model will appear here.")

with tab3:
    st.header("Enhanced Impression (GPT)")
    if st.session_state.enhanced_impression:
                                    st.markdown(f"<div          style='background-color:#D8F6CE;          padding:15px;          border-
radius:10px;'>{st.session_state.enhanced_impression}</div>", unsafe_allow_html=True)
            st.download_button(label="📥 Download Enhanced Impression", data=st.session_state.enhanced_impression,
file_name="enhanced_mri_impression.txt", mime="text/plain")
    elif not AZURE_OPENAI_API_KEY:
         st.warning("Azure OpenAI not configured. Add secrets to enable enhancement.")
    else:
        st.info("The final, enhanced impression from GPT will appear here after generation.")

with st.expander("ℹ️ About / Instructions"):
    st.markdown("""
    - **Step 1:** Provide MRI findings by pasting text or uploading a PDF report.
    - **Step 2:** The app validates the input. If it's a valid report, our model model fine tuned on mimic 4 radiology mri textual
clinical data with biobart model generates a raw impression draft.
     - **Step 3:** An expert AI (GPT) refines this generated impression draft using the original findings to create a complete,
professional mri radiology report.
    - Use the tabs to navigate between the different stages of the output.
    """)
```

3.2 Screenshots





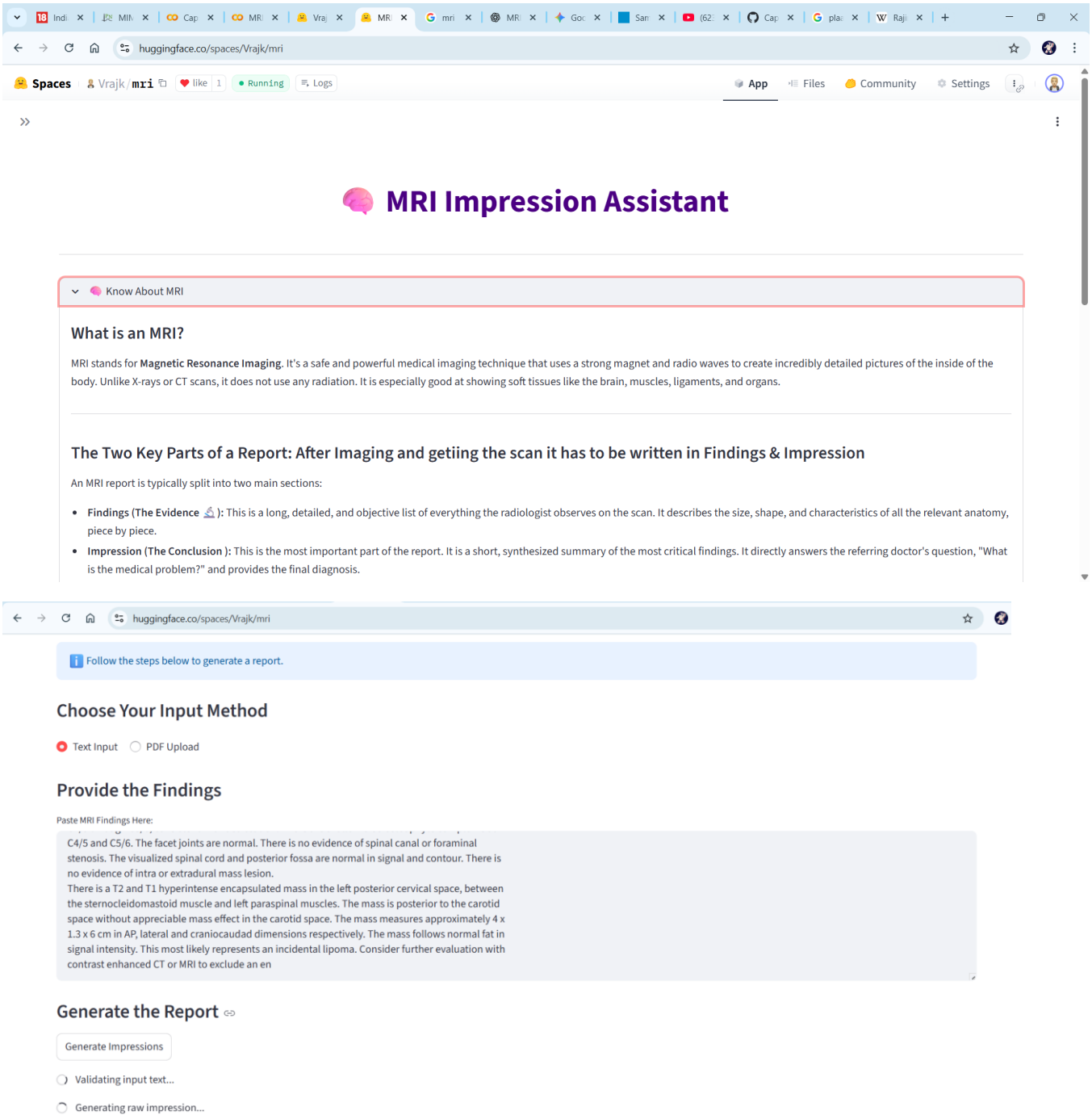
So this is a not valid as it says abour Rajnikant out of context so the validation will invalidate it.

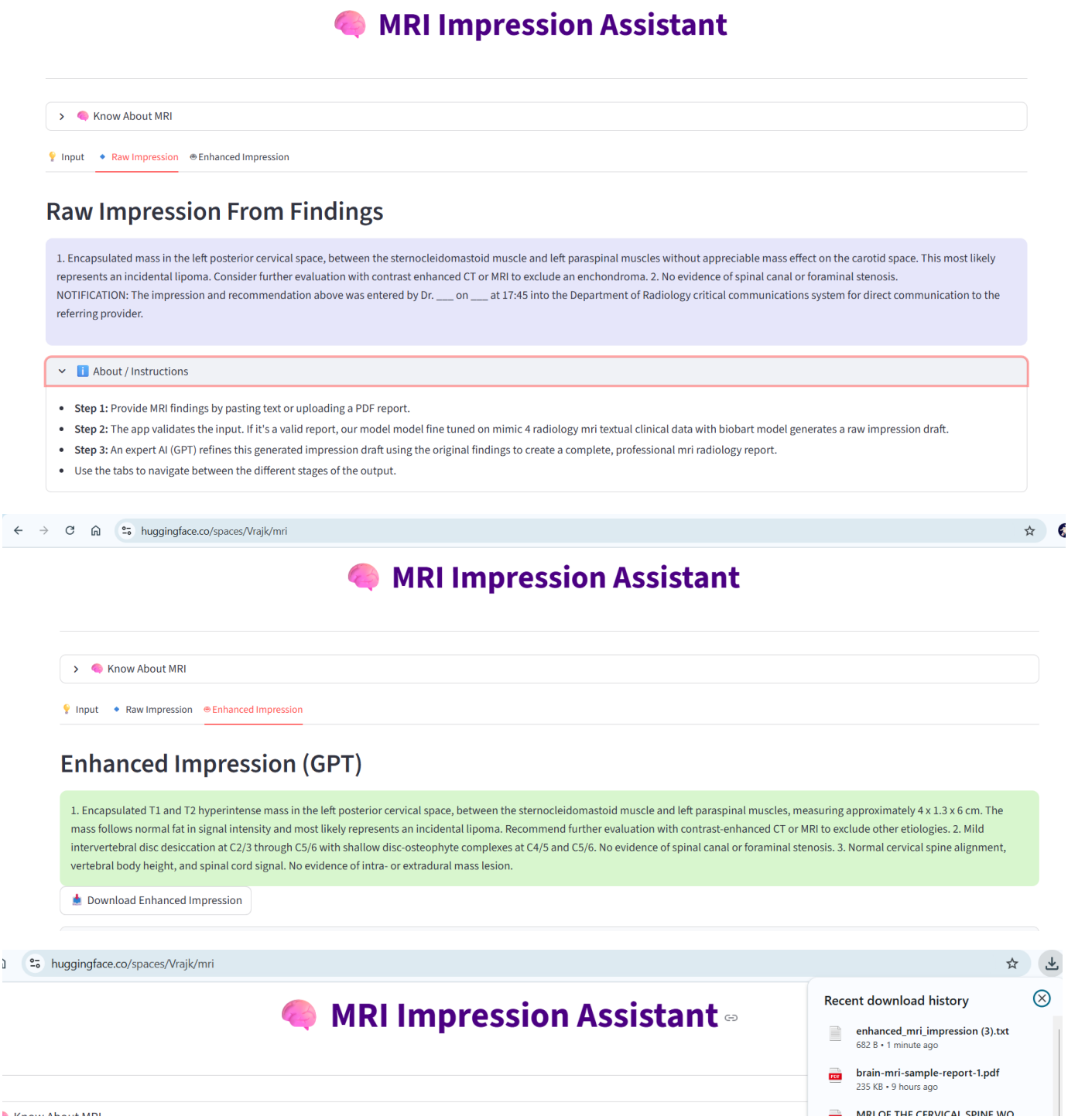
cinema.[8][9] Known for his uniquely styled mannerism and one liners in films, he has a huge fan base and a cult following. The Government of India hon
the Padma Vibhushan in 2016, India's third and second highest civilian honours respectively, and the Dadasaheb Phalke Award in 2019, the highest Indian
contributions to Indian cinema.[10][11] He has won numerous film awards including one National Film Award, seven Tamil Nadu State Film Awards, a Na
Maharashtra State Film Awards.

Generate the Report

Generate Impressions

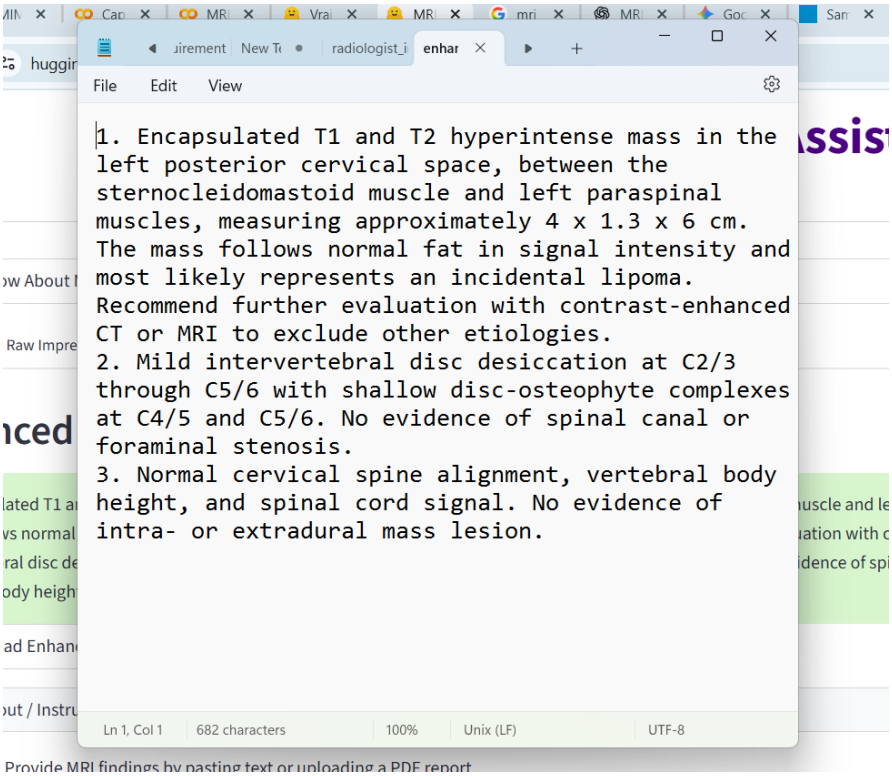
Validation Failed: The provided text does not appear to be MRI findings. Please provide a relevant medical report.




MRI Impression Assistant
Know About MRI
Input
Raw Impression
Enhanced Impression
Raw Impression From Findings
1. Encapsulated mass in the left posterior cervical space, between the sternocleidomastoid muscle and left paraspinal muscles without appreciable mass effect on the carotid space. This most likely represents an incidental lipoma. Consider further evaluation with contrast enhanced CT or MRI to exclude an enchondroma. 2. No evidence of spinal canal or foraminal stenosis.
NOTIFICATION: The impression and recommendation above was entered by Dr. ___ on ___ at 17:45 into the Department of Radiology critical communications system for direct communication to the referring provider.
About / Instructions
Step 1: Provide MRI findings by pasting text or uploading a PDF report.
Step 2: The app validates the input. If it's a valid report, our model model fine tuned on mimic 4 radiology mri textual clinical data with biobart model generates a raw impression draft.
Step 3: An expert AI (GPT) refines this generated impression draft using the original findings to create a complete, professional mri radiology report.
Use the tabs to navigate between the different stages of the output.

huggingface.co/spaces/Vrajk/mri
MRI Impression Assistant
Know About MRI
Input
Raw Impression
Enhanced Impression
Enhanced Impression (GPT)
1. Encapsulated T1 and T2 hyperintense mass in the left posterior cervical space, between the sternocleidomastoid muscle and left paraspinal muscles, measuring approximately 4 x 1.3 x 6 cm. The mass follows normal fat in signal intensity and most likely represents an incidental lipoma. Recommend further evaluation with contrast-enhanced CT or MRI to exclude other etiologies. 2. Mild intervertebral disc desiccation at C2/3 through C5/6 with shallow disc-osteophyte complexes at C4/5 and C5/6. No evidence of spinal canal or foraminal stenosis. 3. Normal cervical spine alignment, vertebral body height, and spinal cord signal. No evidence of intra- or extradural mass lesion.
Download Enhanced Impression

huggingface.co/spaces/Vrajk/mri
MRI Impression Assistant
Recent download history
enhanced_mri_impression (3).txt
682 B • 1 minute ago
brain-mri-sample-report-1.pdf
235 KB • 9 hours ago



3.3 Future Enhancements:

- Work with images of mri to generate findings as well as impressions both.
- Expanding to other medical imaging modalities (e.g., CT, X-ray).
- Incorporating user feedback loops for continuous model improvement.
- Discussion With radiologist suggested that it needs more details so that it can give you impressions close to what radioloist give and aslo suggested to work on patient history as well the impression are base on patient history , the location form where they belong to any medication being consumed .

The MRI Impression Assistant is successfully deployed and publicly accessible.

- Live URL: https://huggingface.co/spaces/Vrajk/mri
- Gitub : https://github.com/Vrajnandwana/CapstoneProject