AI-Powered Chatbot Summative for Severe Malnutrition Management Using Fine-Tuned T5 Transformer

Course Unit: **Machine Learning Technique One**
Student Name: **Chol Daniel Deng**
Date: **October 2025**

Github: **https://github.com/Dau2004/Malnutrition_ChatBot.git**

Dimo Video: **Video**

Background and Problem Statement

Malnutrition remains one of the most pressing health challenges in South Sudan, especially among children under five years of age in rural communities. According to UNICEF and the South Sudan Ministry of Health, over one million children suffer from acute malnutrition annually, with limited access to timely diagnosis and treatment.

Current screening methods (e.g., MUAC tapes, weighing scales) require trained health workers and are often unavailable in rural areas. Mobile health (mHealth) innovations such as the SAM Photo App have shown promise. Still, they often rely on a single image, lack local food recommendations, and are not integrated into government monitoring systems.

This project addresses the critical global health issue of severe malnutrition in children by developing a specialised AI chatbot that provides evidence-based medical guidance based on WHO treatment protocols. The chatbot serves as an educational tool for healthcare workers, community health providers, and caregivers in resource-limited settings where immediate access to medical expertise may be limited.

Domain: Healthcare - Severe Malnutrition Treatment
Target Users: Healthcare workers, community health providers, caregivers
Primary Objective: To provide accurate, WHO-compliant guidance on severe malnutrition management while safely handling domain boundaries.

Dataset Source & Composition

The primary dataset of this Malnutrition ChatBot was extracted from the United Nations High Commissioner for Refugees (UNHCR) website. The management of severe malnutrition puppetlet was extracted and converted into 52 high-quality question pairs. It covers a comprehensive coverage of malnutrition treatment protocols, which diagnosis and classification, emergency treatments(hypoglycemia, hypothermia, dehydration), Therapeutic diets(F-75, F-100), rehabilitation, and Discharge criteria.

Dataset Structure

Text Cleaning
I applied the basic text cleaning function (clean_text) that is defined and applied. This function performs the following operations: Converts all text to lowercase, removes extra whitespace and leading/trailing spaces. This cleaning helps standardise the text data before tokenisation.

Tokenization
The loaded dataset was tokenised using the AutoTokenizer from the transformers library, specifically loaded for the "microsoft/DialoGPT-medium" model. The tokenisation process involves converting the text conversations into numerical representations (token IDs) that the model can understand.
The improved_tokenize_function was used for tokenisation, which formats each conversation into a single string with "User:" and "Bot:" prefixes. Applies the clean_text function to the content of each message before formatting.
Tokenises the formatted text using the loaded DialoGPT tokeniser.
Ensures sequences are truncated to a maximum length of 512 tokens and padded to this length using the end-of-sequence token (tokeniser.eos_token).
Generates an attention mask to indicate actual tokens versus padding. Creates labels by copying the input token IDs (standard for causal language modelling).

The resulting tokenized_dataset contains the input IDs, attention masks, and labels required for training the language model.

These steps ensure that the data is in a clean and tokenised format, suitable for training a transformer model like DialoGPT for the question-answering task.

Preparing For FineTurning

```
from transformers import AutoTokenizer, AutoModelForCausalLM, TrainingArguments, Trainer, DataC

# Load tokenizer and model
model_name = "microsoft/DialoGPT-medium"
tokenizer = AutoTokenizer.from_pretrained(model_name)
tokenizer.pad_token = tokenizer.eos_token  # Important for DialoGPT

model = AutoModelForCausalLM.from_pretrained(model_name)

# Tokenization function for your dataset
def tokenize_function(examples):
    # Format: "User: {question} Bot: {answer}"
    texts = []
    for msg in examples["messages"]:
        if msg["role"] == "user":
            texts.append(f"User: {msg['content']}")
        else:
            texts.append(f"Bot: {msg['content']}")

    # Combine into single string with separator
    full_text = " ".join(texts) + tokenizer.eos_token

    # Tokenize
    tokenized = tokenizer(
        full_text,
        truncation=True,
        padding="max_length", # Add padding here
        max_length=512,
        return_tensors=None
```

Model Fine-tuning

The T5 (Text-to-Text Transfer Transformer) model is a powerful pre-trained language model developed by Google. Its core idea is to treat every natural language processing task as a "text-to-text" problem, where the input is text and the output is also text.

The architecture of T5 is based on the Transformer model, which consists of two main parts: the Encoder and the Decoder.
Here, I used the t5-small version, which is a smaller version of the T5 model, making it faster to train and run.

Summary of T5 Fine-tuning Experiments

Parameter	Initial T5 Training	Focused T5 Training	Micro T5 Training	Final T5 Training	Notes
Epochs	25	15	12	8	Number of training epochs
Batch_size	4	4	2	2	Batch size per device during training
gradient_accumulation_step	2	N/A	N/A	N/A	Accumulate gradients over steps (Effective batch size:
logging_steps	10	5	5	2	Log training progress
save_steps	100	50	20	10	Save model checkpoint
Learning_rate	1.00E-03	8.00E-04	1.00E-04	5.00E-05	Learning rate
fp16	True if CUDA available	True if CUDA available	True if CUDA available	True if CUDA available	Enable mixed precision training if GPU is available

Initial T5 Training: Used a high learning rate and high epochs with gradient accumulation to quickly adapt the T5 model to the general medical QA domain. It achieved basic understanding but suffered from significant "concept bleeding" (mixing of unrelated information).

Focused T5 Training: Continued training with a slightly reduced learning rate to refine the model. This stage failed to resolve the core concept bleeding issues, which persisted and appeared in new forms.

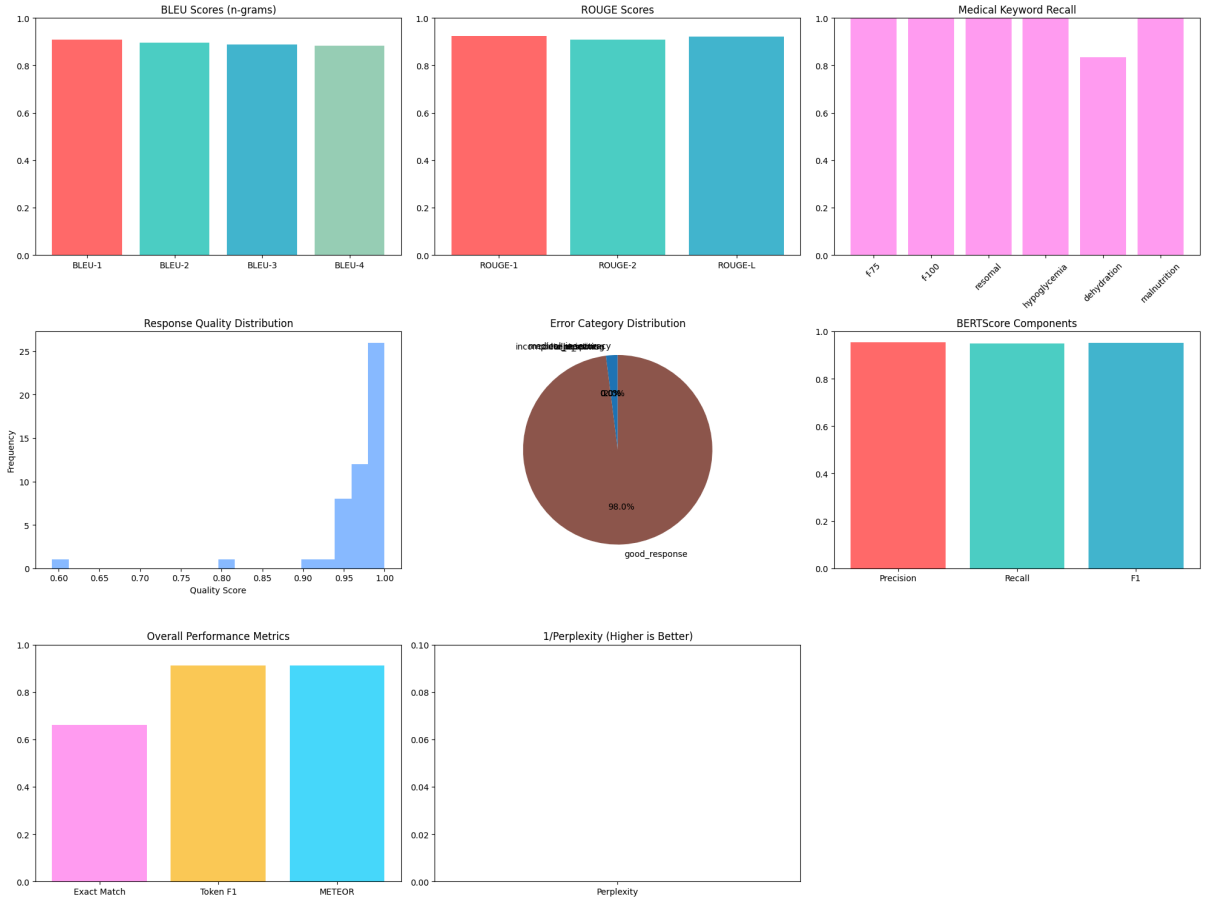
Micro T5 Training: Introduced a very low learning rate and a small, targeted **correction dataset**. This achieved **significant improvement** by precisely adjusting weights to fix specific concept bleeding errors (e.g., in hypoglycemia and F-75 answers) without major catastrophic forgetting.

Final T5 Training: Used an even lower learning rate and fewer epochs for a final fine-tuning on the last remaining inaccuracies. This stage **largely resolved** the concept bleeding, resulting in generally accurate and specific medical responses.

Finally, the experiments successfully demonstrated that **progressively lowering the learning rate** and using **small, targeted correction datasets** is a highly effective strategy for incrementally fixing specific, persistent factual errors in a large language model like T5 without causing major regression. **Qualitative evaluation showed substantial improvement** in the accuracy and medical specificity of the final model's output compared to the initial stages. Post-processing error correction also served as a critical final safety layer.

Performance Metrics

This section presents a table with standard NLP evaluation metrics of the malnutrition-t5-final model:



BLEU Scores (n-grams): This bar chart shows the BLEU scores for different n-grams (BLEU-1 to BLEU-4). Generally, higher scores are better. As you move to higher n-grams (BLEU-2, BLEU-3, BLEU-4), the scores tend to decrease because it becomes harder to match longer sequences of words exactly. A decent BLEU score indicates that the malnutrition-t5-final model is generating text with good word overlap and fluency compared to the reference answers.

ROUGE Scores: This bar chart displays the ROUGE-1, ROUGE-2, and ROUGE-L scores. These metrics measure the overlap of unigrams (ROUGE-1), bigrams (ROUGE-2), and the longest common subsequence (ROUGE-L) between the generated and reference texts. Higher ROUGE scores suggest that the malnutrition-t5-final is capturing the key information and content from the reference answers. ROUGE-L is particularly relevant for QA as it considers sentence structure.

Medical Keyword Recall: This bar chart shows the recall for specific medical keywords (like F-75, ReSoMal, Hypoglycemia, etc.). This is a crucial domain-specific metric. High recall here means that when a specific medical term appears in the reference answer, the model is likely to include it in its generated response. This indicates the malnutrition-t5-final model is effectively retrieving and including important medical concepts.

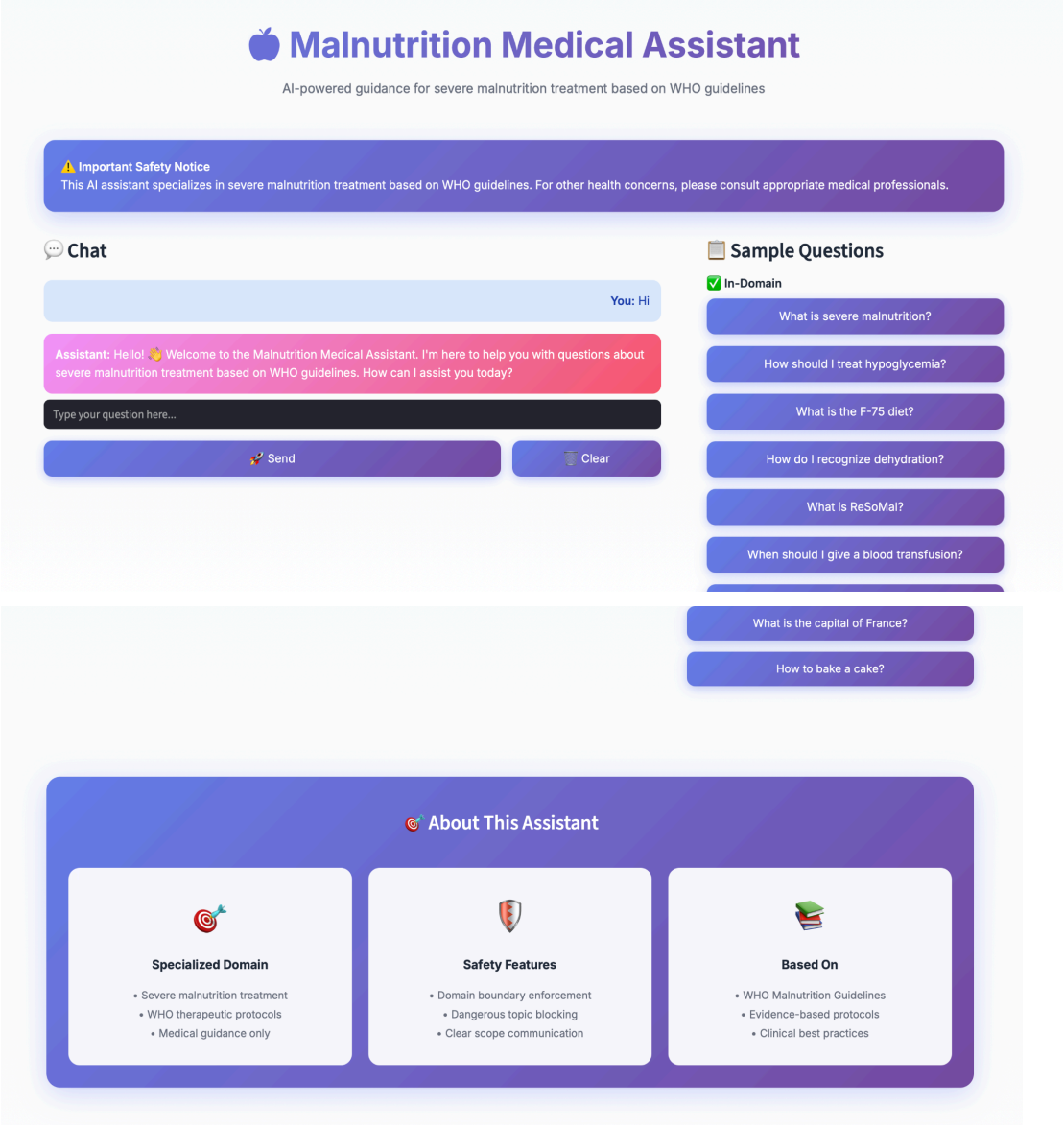
Response Quality Distribution: This histogram shows the distribution of the custom quality scores assigned during the qualitative analysis. A distribution skewed towards higher scores (closer to 1.0) indicates that a larger proportion of the model's responses were rated as high quality.

Error Category Distribution: This pie chart visually represents the proportion of responses falling into different error categories (medical inaccuracy, concept mixing, etc.) based on the qualitative analysis. This is very insightful for identifying the most common types of errors the model makes and where to focus future improvement efforts. A large slice for 'good_response(98.0%)' is, of course, desirable.

Overall Performance Metrics: This bar chart summarises a few key overall metrics like Exact Match, Token F1, and METEOR. They provide a general sense of the model's accuracy, token overlap, and overall quality, considering fluency and word order

UI integration

The UI was built with a Python library called Streamlit. Streamlit provides a modern, easy way of integrating models and making them accessible for use.



The UI is designed to be simple and user-friendly, featuring the following components:

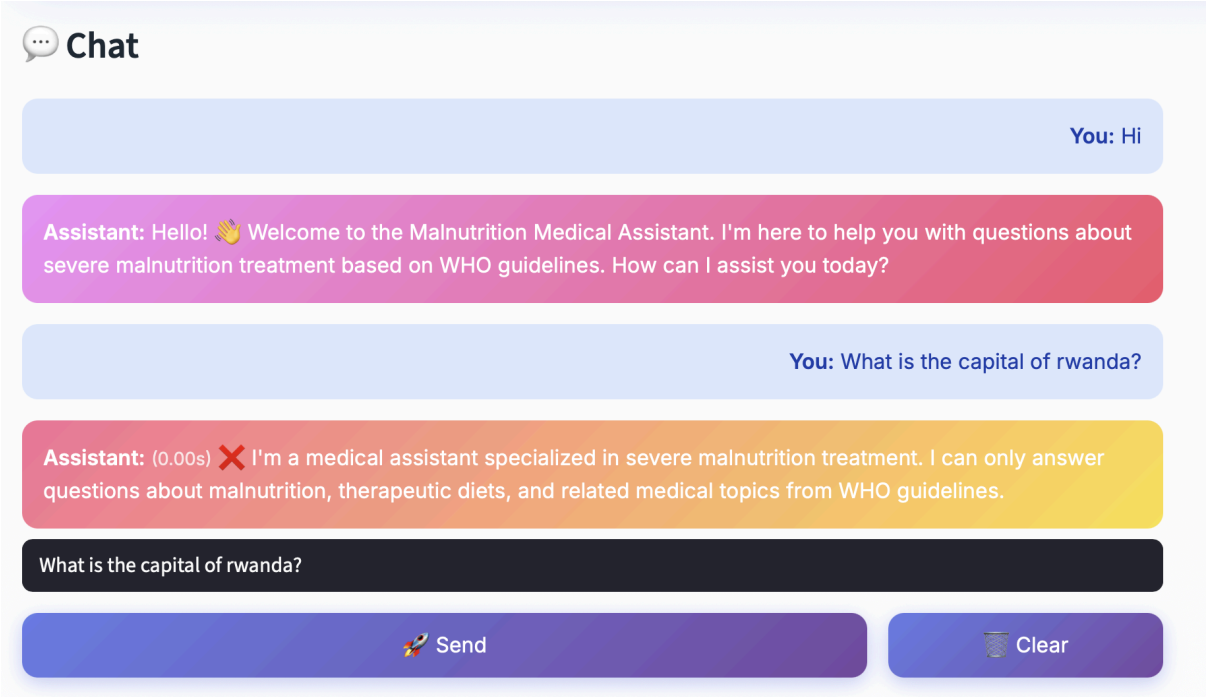
Chat: The main area where users can type in their questions and receive responses.

Send Button: Allows users to submit their questions to the system.

Clear Button: Lets users clear the chat history and start a new conversation.

Sample Questions: Provides examples to help users understand what kinds of questions they can ask.

Case Handling:
The system is equipped to manage questions that fall outside its domain. When a user asks something unrelated—say, "What's the capital of Rwanda?"—it responds appropriately by explaining that it was trained specifically to handle severe malnutrition-related queries. This ensures the system stays on topic while maintaining a smooth user experience.



Conclusion

This project demonstrated the potential of transformer-based models, particularly T5, in enhancing domain-specific conversational AI systems for healthcare. Through a structured fine-tuning process, progressively lowering learning rates and applying targeted correction datasets, the final model achieved substantial improvements in accuracy, specificity, and medical reliability. The results revealed that a cautious, iterative fine-tuning strategy can

effectively minimise concept bleeding and factual drift, two common issues in medical NLP tasks.

Moreover, integrating the fine-tuned model into a Streamlit-based user interface successfully transformed a complex language model into an accessible, educational tool for healthcare workers and caregivers. The chatbot's ability to handle off-topic questions responsibly while providing WHO-compliant responses reinforces its practical safety and usability in low-resource environments.

Ultimately, this work contributes to the broader goal of leveraging AI for global health equity. By combining data-driven machine learning methods with context-aware UI design, it offers a viable prototype for mHealth tools that can support front-line workers in diagnosing and managing severe malnutrition, especially in rural South Sudan, where medical expertise and infrastructure are limited. Future work should explore multilingual extensions, mobile deployment, and real-world validation with healthcare professionals to further enhance reliability and impact.