Text Mining and Natural Language Processing

2024-2025

Sara Di Franco [535408], Lucrezia Bernini [554298]

AIMedQA - AI for **Med**ical **Q**uestion **A**nswering

Section Index

Introduction

The goal of this project is to develop a Medical Question Answering (Q&A) system by fine-tuning language models to generate accurate, informative, and contextually appropriate responses to natural language health-related questions. We frame this task as a sequence-to-sequence generation problem, where the model receives a patient-style question, optionally enriched with a reasoning chain (Chain-of-Thought), and produces a response that reflects the tone and precision of a doctor or medical assistant.

In particular the main goals are:

1. Fine-tune and compare general-domain and domain-specific encoder–decoder models (T5-small vs. SciFive) to evaluate the impact of biomedical pre training on medical question answering performance.
2. Design and evaluate prompting strategies (zero-shot, one-shot, few-shot) on instruction-tuned models (Flan-T5 and Phi-3-mini) to assess their ability to generate accurate answers without fine-tuning.
3. Analyze model behavior under both training and prompting conditions, identifying strengths and limitations across architectures.
4. Evaluate output quality using automatic metrics such as ROUGE, BLEU, and BERTScore.

The project has been implemented using Google Collab tools. To view the outputs in a more readable and structured format, we recommend opening the `.ipynb` file directly on the above mentioned platform.

Data

The dataset used in this project is a collection of 1,000 medical question–answer pairs, specifically designed to simulate real-world interactions between patients and healthcare professionals. Each entry in the dataset consists of a question, a corresponding answer, and optionally a chain of thought that explains the reasoning behind the response.

Questions are written in natural, conversational language, often reflecting how a patient might describe their symptoms or concerns. The answers are phrased in a medically accurate yet accessible way, as if coming from a doctor or a medical assistant. In many cases, the dataset also includes a reasoning chain: a brief explanation of the logical or clinical steps that lead from the question to the final answer. This supports both training and evaluation of models capable of step-by-step reasoning.

The dataset is divided into three subsets: 600 samples for training, 200 for validation, and 200 for testing. On average, the questions are around 11 words long, the answers contain about 22 to 23 words, and the reasoning chains—when present—are slightly longer, averaging around 30 words. In total, the dataset includes approximately 75,000 words.

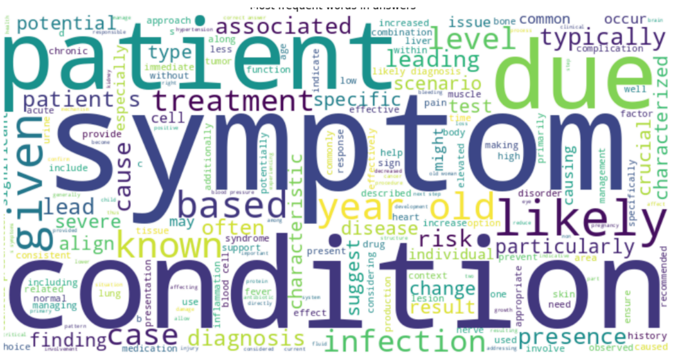
To give an idea of the dataset structure, here is a representative example:

Question: *"I’ve had a sore throat for three days. Should I take antibiotics?"*
Chain of Thought: *"A sore throat lasting less than a week is usually viral, especially if there is no fever. Antibiotics are generally not recommended for viral infections."*
Response: *"If your sore throat is mild and you don’t have a fever, it is likely viral, and antibiotics are not necessary. You can treat it with rest, fluids, and over-the-counter medications."*

Visualization

WordCloud Technique

To better understand the linguistic patterns in the answers, we also generated a word cloud based on the most frequent terms used. As shown below, words like *“symptom,” “condition,” “patient,” “diagnosis,”* and *“treatment”* appear prominently, reflecting the clinical and diagnostic nature of the content:



BERTopic Technique

To further explore the semantic diversity of the questions, we applied **BERTopic**, a topic modeling technique that uses pre-trained BERT embeddings and clustering. After removing stopwords, the method revealed five main topic clusters reflecting recurring medical themes. These include: *neurology and nerve anatomy*; *diagnosis and clinical terminology*; *pharmacology and treatments*; *physiology*; and *systemic anatomy*.This topic analysis confirms that the dataset is not only rich in vocabulary but also well balanced across a variety of medically relevant domains.

Topic	Count	Name	Representation
-1	98	-1_old_year_presents_patient	old,year,presents,patient,likely,symptoms,condition,normal,examination,woman
0	26	0_nerve_right_patient_year	nerve,right,patient,year,old,management,side,trauma,margin,spinal
1	16	1_serum_findings_history_abdominal	serum,findings,history,abdominal,laboratory,year,pain,old,likely,min
2	16	2_organism_positive_infection_infected	organism,positive,infection,infected,virus,chains,culture,new,condition,cell
3	16	3_removal_weeks_blood_anti	removal,weeks,blood,anti,stimulation,rh,hormone,cells,heme,hypothalamus
4	14	4_chest_year_lung_likely	chest,year,lung,likely,old,dyspnea,ecg,history,examination,exertion

Methodology

This project focuses on fine-tuning and evaluating multiple transformer-based language models (T5-Small, SciFive, Flan-T5, and Phi-3-mini) for medical question answering. The task is framed as a sequence-to-sequence generation problem, where the model receives a medical question—optionally enriched with a chain of thought—and generates a medically appropriate response.

Preprocessing

The preprocessing pipeline was carefully designed to preserve the semantic richness of medical language while ensuring input consistency. All text was lowercase to normalize casing. Stopword removal was intentionally avoided: function words such as *"no," "without,"* and *"may"* are essential in medical interpretation and cannot be discarded without risking the loss of critical meaning.

Punctuation was selectively filtered. Non-informative characters (e.g., exclamation marks) were removed, while medically significant symbols—such as colons, slashes, periods, and percentage signs—were retained to maintain the integrity of numerical data, measurements, and abbreviations. Whitespace was normalized to ensure clean tokenization.

Each model employed its corresponding pretrained tokenizer (e.g., t5-small, google/flan-t5-base, microsoft/phi-2, razent/SciFive) using Hugging Face's AutoTokenizer. This ensured that tokenization parameters like padding, truncation, special tokens, and vocabulary were automatically aligned with each model's internal configuration. Inputs were constructed by concatenating the *"Question"* field with the *"Complex_CoT"* field when available; the *"Response"* field served as the decoding target during training and evaluation.

Avoiding Traditional Text Representations

We deliberately avoided traditional feature engineering techniques such as **CountVectorizer**, **TF-IDF**, or static word embeddings like **GloVe** or manual **BPE + torch** embeddings. These methods, although valuable in classical NLP pipelines, are suboptimal in the context of generative transformer models:

- **Transformers learn contextual representations** of language during pre-training. Words are embedded dynamically based on surrounding context, unlike static methods which cannot disambiguate polysemous terms (e.g., *"positive result"* in medical vs. non-medical usage).

- **Redundancy and complexity**: Using manual embedding techniques or vectorizers would require a parallel preprocessing pipeline and potentially conflict with the model's internal tokenization and representation layers, adding unnecessary complexity without tangible benefits.

- **Lower performance in generation tasks**: TF-IDF and similar vectorizers are sparse and not suited for sequence-to-sequence generation tasks. They fail to capture word order, syntactic dependencies, or semantic relationships crucial for generating coherent, medically grounded responses.

In summary, we chose to rely exclusively on transformer-native tokenizers and representations to fully leverage the models' pretrained knowledge and ensure end-to-end compatibility across all stages of training, evaluation, and inference.

Models

To explore the strengths and limitations of different architectures, we adopted two parallel evaluation strategies based on the nature of the models.

Fine-Tuned Encoder–Decoder Models

We first focused on **T5-Small** and **SciFive**, two encoder–decoder models built for sequence-to-sequence tasks.

- **T5-Small** is a general-purpose transformer pretrained on a diverse set of text-to-text tasks (e.g., summarization, translation, QA).
- **SciFive** is a biomedical variant of T5, further pretrained on domain-specific corpora such as PubMed abstracts and clinical notes.

Since these models are not inherently designed for prompt-based inference, we performed fine-tuning to adapt them to our medical Q&A task. This allowed us to:

- Analyze overfitting behavior and generalization.
- Evaluate the impact of domain-specific pretraining on performance and domain awareness.

The comparison helped assess whether a biomedical focus improves response quality on medical questions compared to a general-language model.

Prompt-Based Instruction-Following Models

In parallel, we evaluated **Flan-T5** and **Phi-3-mini**, both instruction-tuned models designed to follow natural language prompts without further training.

- **Flan-T5** is a variant of T5 fine-tuned using instruction-based datasets, enhancing its ability to follow complex prompts.
- **Phi-3-mini**, a lightweight but high-performing model from Microsoft, is tuned to reason and generate in response to structured prompts.

These models support **zero-shot**, **one-shot**, and **few-shot** prompting, enabling us to evaluate their performance in real-world scenarios where fine-tuning may not be feasible. Our goal was to test:

- Their ability to understand and follow structured instructions.
- Their adaptability to different levels of prompt guidance.
- Their capacity to produce context-aware, medically accurate answers relying solely on prompt information.

This dual-track evaluation provided insights into the trade-offs between fine-tuning and prompting, and between general-purpose and domain-specialized architectures.

Result and Analysis

To evaluate the performance of the models, we adopted a combination of automatic metrics that capture different dimensions of response quality. Specifically, **ROUGE-L** was used to assess the overlap between the generated and reference sequences, especially at the sentence level. **BLEU** helped quantify the correctness of the generated n-grams, which is particularly useful for more structured or list-like answers. Finally, **BERTScore** provided a semantic measure of similarity between the generated output and the ground truth, taking into account the contextual meaning of words.

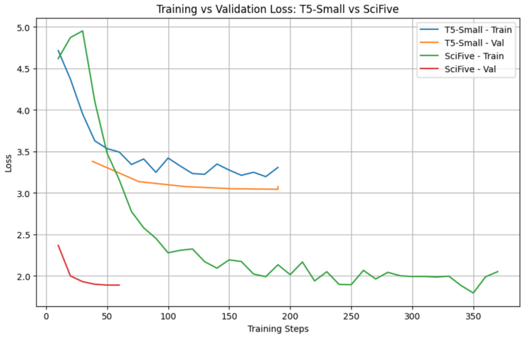
First Observation – Fine Tuning

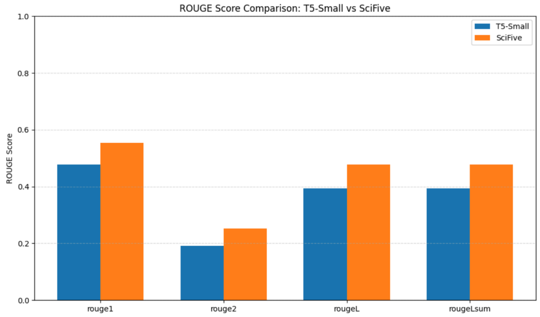
As illustrated in the first plot, **T5-Small** shows a moderate decrease in training loss and quickly reaches a plateau in validation loss (~3.0), indicating limited ability to generalize to medical data.

On the other hand, **SciFive** demonstrates a much faster and more stable convergence, with both training and validation losses dropping significantly and stabilizing below 2.0. This suggests that SciFive adapts more effectively to the task and benefits from its prior exposure to biomedical language.

This trend, was also confirmed by the second comparison we made on the ROUGE evaluation scores (ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-Lsum). As a matter of fact, across all metrics, **SciFive consistently outperforms T5-Small**, producing outputs that are more similar to the reference answers. The improvement is especially notable in ROUGE-1 and ROUGE-L, indicating better lexical and structural similarity.

These results highlighted the advantage of using domain-specific pre-trained models in specialized tasks.



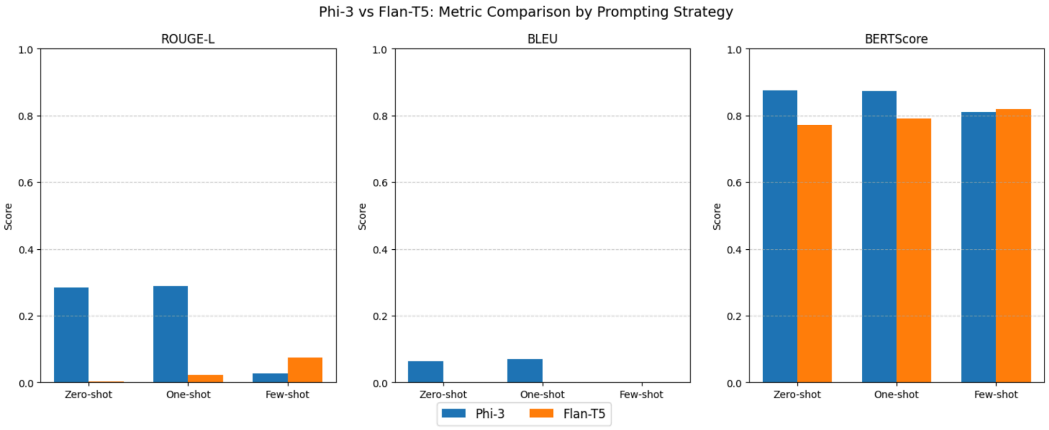


Second Observation – Prompt Engineering

In the second track of analysis, we focused on models capable of instruction following via prompting—namely, **Flan-T5** and **Phi-3-mini**. These models were evaluated in ***zero-shot***, ***one-shot***, and ***few-shot*** settings, where no fine-tuning was applied. Instead, we varied the structure of the input prompt to assess how well each model could generate accurate answers given different levels of guidance.

As shown in the figure, Phi-3-Mini consistently achieves higher scores across all metrics. On the other hand Flan-T5 shows weaker zero-shot performance but narrows the gap in one-shot and few-shot scenarios – indicating that it benefits more from in-context examples.

Hence, Phi-3-Mini appears to be the stronger model for medical Q&A, especially when flexibility and few-shot performance are critical. Flan-T5 could serve as a fallback if additional prompt engineering is applied.



Third Observation - Interactive Simulation

To assess the practical performance of different models, we conducted an interactive simulation using two input types:

- A general medical question **outside the training distribution**
- A verbatim **clinical case from the fine-tuning dataset**

We compared four models: two fine-tuned (T5-small and SciFive) and two prompt-only (Phi-3 Mini and Flan-T5).

Input: *[Clinical Case from Dataset]*

	T5-small	**Scifive**	**Phi-3-mini**	**Flan-T5**
Output	serum studies indicate hyponatremia, hyperkalemia, and hypoglycemia. based on these symptoms and laboratory findings, what is initial step in treatment that should be administered to this patient?	based on the symptoms and laboratory findings, the initial step in treatment that should be administered to this patient is to initiate treatment. based on the symptoms and laboratory findings, the initial step in treatment that should be administered to this patient is to initiate treatment.	The symptoms and laboratory findings described for the 8-year-old boy are suggestive of Addison's disease, also known as primary adrenal insufficiency. This condition occurs when the adrenal glands do not produce sufficient amounts of steroid hormones, particularly cortisol and aldosterone. The initial step in treatment for a patient with suspected Addison's disease is the administration of hydrocortisone. Hydrocortisone is a synthetic glucocorticoid that acts as a replacement for cortisol, which is deficient in Addison's disease. It helps to manage the symptoms of cortisol deficiency, such as fatigue, weight loss, and hypoglycemia. In addition to hydrocortisone, the patient may also require fludrocortisone, a synthetic mineralocorticoid, to replace aldosterone and help regulate sodium and potassium levels. However, the immediate priority is to address the cortisol deficiency with hydrocortisone. It's important to note that Addison's disease is	Glucose tolerance test

Input: *What are diabetes symptoms?*

	T5-small	Scifive	Phi-3-mini	Flan-T5
Output	Welche Diabetes symptoms are diabetes symptoms?	are the most common symptoms of type 2 diabetes?	Common symptoms of diabetes include increased thirst, frequent urination, hunger, fatigue, blurred vision, slow-healing sores, frequent infections, and unexplained weight loss. It's important to consult a healthcare professional for proper diagnosis and treatment.	Diabetic coma

Fine-Tuned Models

T5-small, when fine-tuned on clinical data, exhibited limited generalization. For unseen questions, it often produced awkward or incorrect outputs, such as literal translations mixing English and German. For in-distribution questions, it tended to repeat the input with minimal transformation, failing to generate meaningful answers.

SciFive, pre-trained on biomedical corpora and fine-tuned on the same dataset, demonstrated more relevant and context-aware responses. It could generalize to loosely related questions, but some answers were repetitive and lacked specificity.

Prompt-Only Models

Phi-3 Mini, used without any fine-tuning, produced clear, medically accurate, and context-sensitive answers in both cases. It identified conditions (e.g., Addison's disease) and suggested appropriate treatments (e.g., hydrocortisone), demonstrating strong reasoning and medical knowledge.

Flan-T5, despite prompt guidance, failed to provide coherent or relevant answers. For the general question, it responded with unrelated terms (e.g., *Diabetic coma*), and for the clinical case, it misinterpreted the context entirely.

Hence, these simulations, again, reinforce the importance of domain adaptation. Fine-tuning improves alignment, especially for models like SciFive with biomedical pretraining. However, prompt engineering can yield competitive—and in some cases superior—results, particularly when applied to strong general-purpose models like Phi-3 Mini. Conversely, models without domain-specific pretraining (e.g., Flan-T5) may underperform, even with carefully designed prompts.

Conclusion

Issues Encountered

During the development phase, we encountered several **limitations** primarily related to **computational resources**. Specifically, the high computational cost of fine-tuning large-scale encoder-decoder models, such as Phi-3 Mini and FLAN-T5, prevented us from performing full fine-tuning. Instead, we adopted a **dual-track approach**:

1. **Fine-tuning smaller models** such as **T5-small** and **SciFive-base**, both based on the T5 encoder-decoder architecture.
2. **Prompt engineering** with larger pre-trained models to avoid training costs, by leveraging them without updating their weights

This allowed us to compare the performance trade-offs between computationally inexpensive prompt-based methods and more resource-intensive fine-tuning.

Moreover, initially, we imported the standard T5-base model, but due to its large size and long training time, we opted for T5-small as a more computationally feasible alternative. Despite this, training both T5-small and SciFive on the full dataset proved to be computationally demanding. As a result, we applied the following optimizations:

- We trained on a **subset of the dataset** rather than the full corpus.
- We **reduced the batch size** to fit within the available GPU memory.

When attempting **prompt engineering** on **SciFive**, we faced unexpected limitations. Although we correctly used the AutoModelForSeq2SeqLM class—enabling generation through .generate()—the model consistently failed to produce coherent or task-relevant responses. We believe this is due to the fact that **SciFive was pre-trained for biomedical text generation** and is **not instruction-tuned**. As a result, it may not respond appropriately to general natural language prompts, particularly those outside of its domain-specific training distribution.

This experience highlighted an important insight: **even when a model is architecturally capable of generation, its performance in prompt-based tasks depends heavily on pretraining objectives and data domain**. In contrast to instruction-tuned models like FLAN-T5 or Phi-3, SciFive appears to require fine-tuning on the specific task in order to yield meaningful outputs.

Last but not least, we also believe it would have been more practical to move the import and installation steps to the beginning of the project. However, we've noticed significant slowdowns in the machine, which may be due to conflicts or collisions during the import process.

Project Recap

We could reassume our project as follows:

Part 0 – Dataset Preparation
We started from a custom dataset derived from *medical-o1-reasoning-SFT* and saved it in a .json format. We cleaned and normalized it to be suitable for training and evaluation, maintaining both long and short doctor-patient interactions.

Part 1 – Dataset Loading and Structure Check
We imported and loaded the dataset using Hugging Face Datasets, visualizing the structure (input: patient question, output: medical answer) and ensuring consistency across all samples.

Part 2 – Preprocessing and Topic Modeling
We performed preprocessing: lowercased the text, removed extra spaces and non-relevant punctuation, but retained stopwords and clinical numbers (e.g. doses, ages). We then visualized term frequency, plotted word clouds and applied BERTopic to extract medical topics.

Part 3 – Splitting the Dataset
We split the dataset into train/validation/test and verified the label and token length distributions across splits. No oversampling was needed due to the natural balance of question/answer pairs.

Part 4 – Tokenization and Fine-Tuning
We tokenized the dataset using four different tokenizer configurations (T5Tokenizer and SciFiveTokenizer, Phi-3-mini tokenizer, Flan-T5-base tokenizer), then trained T5-Small and SciFive with supervised learning using cross-entropy loss. We logged training metrics and visualized convergence curves.

Part 5 – Evaluation of Fine-Tuned Models
We evaluated the generated answers with ROUGE, BLEU and BERTScore, comparing the linguistic and semantic quality. SciFive consistently outperformed T5-Small in terms of relevance and fluency on the medical responses.

Part 6 – Prompt Engineering on Instruction-Following Models
We experimented with zero-shot, one-shot and few-shot prompting using Phi-3-mini and Flan-T5 models, evaluating their outputs on the same test set. We found Phi-3-mini performs better semantically (BERTScore), while Flan-T5 had stronger lexical overlap.

Part 7 – Interactive Simulation
We allowed the user to input new medical questions, and each model (fine-tuned and prompt-based) could generate a response. We qualitatively compared answers, observing that fine-tuned SciFive gave the most domain-reliable results.

Final Considerations

We approached this project with full awareness that, as anticipated, it would not have been an easy task. Throughout the process, we encountered many challenges and made an effort to move beyond purely technical notation in order to better understand the underlying principles and their relevance in real-world application scenarios.

Despite the complexity, we consider ourselves satisfied with the overall outcome. Not only were we able to implement and evaluate our models, but we also deepened our understanding of the medical NLP domain, from prompt engineering and model fine-tuning to semantic detection and result interpretation.

To conclude we consider this project both technically and theoretically stimulating, helping us in deepening the knowledge seen during classes while also giving them real-world relevance through contextualized application.

Sources

Hugging Face — Phi-3 Mini 4k Instruct by Microsoft
Hugging Face — FLAN-T5 Model Documentation
Hugging Face — SciFive-base-PubMed by razent
Hugging Face — Transformers Tokenizer Documentation

AI policy

Artificial Intelligence tools were primarily used as a support tool throughout the project. Its usage was carefully integrated to enhance clarity and coherence across the following areas:

- **Prompt Engineering:** AI was consulted to help structure effective prompts tailored to the architecture of the models we employed. This included understanding practices in zero-shot, one-shot, and few-shot configurations.
- **Debugging and Reasoning Support:** During development, AI tools were used to validate and check our reasoning and explore alternative or faster solutions to technical issues, especially in the fine-tuning and evaluation phases.
- **Visualization and Topic Clustering:** When exploring semantic relationships between documents, AI was used to identify appropriate visualization tools and methods—such as document clustering based on shared semantic content, using techniques like BERTopic.
- **Fine-tuning and Training (LoRA):** In attempting to fine-tune models such as Phi using **LoRA**, AI support was helpful in understanding the fine-tuning pipeline, especially in relation to managing training time and memory efficiency.

Overall, AI tools served as an assistant to our reasoning, not a replacement. All the implementations were ultimately guided by our own understanding and judgment, with AI support used to accelerate the workflow and clarify complex components or not fully covered topics during classes.

Group Contribution

We carried out the entire project collaboratively, working in person whenever possible to ensure smoother coordination and real-time feedback. Every decision — from dataset selection to modeling and evaluation — was discussed together, allowing each member to contribute to every stage of the pipeline.

Although responsibilities were shared, some tasks required more intense focus: Lucrezia primarily handled the import and configuration of the Phi-3 model, and was particularly involved in the prompt engineering strategies used in the instruction-based evaluations, while Sara focused on integrating and testing the Flan-T5 model and took care of the visualization and comparative analysis between the different models' outputs.

The rest of the project was made all together, including data preprocessing, fine-tuning, evaluation, and writing.