# Medical Question Answering System

## Technical Deep Dive

### NLP Project - Medical Meadow Medical Flashcards

**Authors:** Rishabh Tiwari, Dan Lionis, Federica Vinciguerra, Felipe Bagni, Erfan Amidi

**Date:** February 25, 2026

# Table of Contents

# 1. Project Overview & Motivation

## Problem Statement

Building an intelligent medical question-answering system that provides accurate, contextually relevant answers to medical queries using state-of-the-art NLP models.

## Key Objectives

- Analyze and understand medical text data comprehensively
- Fine-tune multiple language models for medical domain adaptation
- Compare different architectures (T5 vs GPT) and training approaches
- Evaluate performance using multiple metrics (semantic similarity, BLEU, etc.)

## Why This Matters

Healthcare professionals need quick access to medical knowledge. This project addresses critical needs in medical information retrieval, requiring high accuracy and reliability. By comparing different model architectures, we identify optimal solutions for domain-specific tasks.

# 2. Dataset Analysis

## Primary Dataset: Medical Meadow Medical Flashcards

**Source:** Hugging Face `medalpaca/medical_meadow_medical_flashcards`

### Dataset Statistics

| Metric | Value |
|---|---|
| Total Samples | 33,955 Q&A pairs |
| Training Set | 27,164 (80%) |
| Validation Set | 3,395 (10%) |
| Test Set | 3,396 (10%) |
| Random Seed | 42 (reproducibility) |

### Data Structure Example

```
{"input": "What is the relationship between very low Mg2+ levels and PTH levels?",
"output": "Very low Mg2+ levels correspond to low PTH levels...",
"instruction": "Answer this question truthfully"}
```

## Secondary Dataset: General Knowledge Q&A;

**Source:** `GokulWork/QuestionAnswer_MCQ` (205 questions)

Used for cross-domain evaluation to test model generalization beyond medical topics (physics, chemistry, biology, history).

# 3. Exploratory Data Analysis

## Statistical Analysis

Comprehensive analysis of document length, vocabulary size, and text complexity using Flesch Reading Ease scores to assess medical text characteristics.

## Word Embeddings (Word2Vec)

- Vector dimensions: 100
- Window size: 5
- Minimum word count: 1
- Algorithm: Skip-gram
- Document embeddings: Created by averaging word vectors

## Clustering & Visualization

**K-Means Clustering:** Applied to document embeddings to identify natural groupings in medical topics.

**t-SNE Visualization:** Reduced dimensionality from 100 to 2 dimensions, revealing distinct semantic groupings of medical concepts.

## Keyword Analysis

**Most Common Medical Terms:**

Treatment | Symptoms | Patient | Blood | Cause | Condition | Risk

**Inverted Index:** Implemented for efficient keyword-based document retrieval.

## Linguistic Feature Extraction

- **POS Tagging:** Analyzed grammatical structure, identified noun-heavy patterns
- **NER (spaCy):** Extracted medical conditions, procedures, medications
- **N-gram Analysis:** Bigrams and trigrams revealed common medical phrase patterns
- **Readability:** Flesch scores showed technical complexity for medical professionals

# 4. Model Architecture & Training

## 4.1 T5-Small with LoRA Adapters

### Model Specifications

| Specification | Value |
|---|---|
| Base Model | Google T5-Small |
| Total Parameters | 61,096,448 (61M) |
| Model Size | 242 MB |
| Architecture | Encoder-Decoder Transformer |

### 4-bit Quantization Strategy

**BitsAndBytes Configuration:** NF4 (Normal Float 4-bit) quantization reduces memory footprint by 8x while maintaining performance.

**Benefits:**

- Memory reduction: 8x less than 32-bit
- Computational efficiency: Faster inference
- Performance preservation: Minimal accuracy loss with NF4

### LoRA (Low-Rank Adaptation)

**Why LoRA?** Parameter-efficient fine-tuning updates only 0.97% of parameters (589,824 out of 61M), maintaining base model performance while adapting to medical domain.

| Parameter | Value | Description |
|---|---|---|
| r (rank) | 16 | Rank of low-rank matrices |
| lora_alpha | 32 | Scaling factor |
| target_modules | q, v | Query & Value projections |
| lora_dropout | 0.05 | Regularization |
| task_type | SEQ_2_SEQ_LM | Sequence-to-sequence |

### Training Configuration

- Learning rate: 3e-4
- Batch size: 3 (training), 4 (validation)
- Epochs: 10
- Weight decay: 0.01 (L2 regularization)
- Optimizer: AdamW
- Evaluation: End of each epoch

### Training Results

| Metric | Value |
|---|---|
| Final Training Loss | 0.3955 |
| Training Time | ~8.3 hours |
| Training Steps | 90,550 |
| Throughput | 9.1 samples/sec |
| Total FLOPs | 2.6 petaFLOPs |

## 4.2 OpenAI GPT Models Fine-Tuning

### davinci-002 Fine-Tuning

**Data Preparation:**

- Original samples: 27,503
- After duplicate removal: 27,164
- Format: JSONL with prompt-completion pairs

**Data Format:** `{"prompt": "<medical question>", "completion": "<ideal answer>"}`

| Model | Training Samples | Model ID | Status |
|---|---|---|---|
| davinci-002 (full) | 27,164 | ft:davinci-002::9KLi6nKN | Succeeded |
| davinci-002 (dummy) | 12 | ft:davinci-002::9Smjsxqf | Succeeded |

### gpt-3.5-turbo Fine-Tuning

**Model:** gpt-3.5-turbo-0125 | **Fine-tuned ID:** ft:gpt-3.5-turbo-0125::9So7gDaT

**Data Format:** Chat-based (conversational) with system, user, and assistant roles

```
{"messages": [
{"role": "system", "content": "Answer this question truthfully"},
{"role": "user", "content": "<question>"},
{"role": "assistant", "content": "<answer>"}
]}
```

**Key Differences from davinci-002:**

- Chat format allows better instruction-following
- More recent architecture (2023 vs 2022)
- Better handling of conversational context

# 5. Evaluation Methodology

## Models Evaluated

| ID | Model Name | Description | Type |
|---|---|---|---|
| m0 | davinci-002 (dummy) | FT on 12 samples | GPT Baseline |
| m1 | davinci-002 (full) | FT on 27,164 samples | GPT Fine-tuned |
| m2 | gpt-3.5-turbo (base) | No fine-tuning | GPT Base |
| m3 | gpt-3.5-turbo (FT) | FT on 27,164 samples | GPT Fine-tuned |
| mf0 | T5-Small (base) | No fine-tuning | T5 Baseline |
| mf1 | T5-Small (LoRA) | FT with LoRA | T5 Fine-tuned |

## Evaluation Metrics

### Metric 1: Similarity Rate

Percentage of responses where the most similar answer (by cosine similarity) matches the ground truth. Measures how often the model's answer is semantically closest to correct.

### Metric 2: Average Cosine Similarity

Semantic similarity between predicted and reference embeddings using SentenceTransformer. Range: -1 (opposite) to 1 (identical). Measures semantic meaning alignment independent of wording.

### Metric 3: BLEU Score

Precision-based metric comparing n-grams between prediction and reference. Uses NLTK tokenization with smoothing (Method 1). Considers 1-gram through 4-gram matches. Measures token-level overlap and fluency.

## Evaluation Process

- **Test Set:** 100 randomly selected questions ensuring diverse coverage
- **Answer Generation:** Each model generates response (max length 20-100 tokens)
- **Similarity Computation:** Predictions compared against ground truth
- **Result Storage:** Per-sample results saved to CSV with summary statistics

# 6. Results & Comparison

## Quantitative Results (100 Test Samples)

| Model | Similarity Rate | Cosine Sim | BLEU Score |
|---|---|---|---|
| m0 (davinci-002 dummy) | 85% | 0.824 | 0.122 |
| m1 (davinci-002 full) | 78% | 0.789 | 0.115 |
| m2 (gpt-3.5-turbo base) | 82% | 0.838 | 0.112 |
| m3 (gpt-3.5-turbo FT) | 83% ★ | 0.831 ★ | 0.163 ★ |
| mf0 (T5 base) | 76% | 0.627 | 0.017 |
| mf1 (T5 LoRA) | 54% | 0.543 | 0.027 |

★ = Best performance

## Key Findings

### Best Overall: gpt-3.5-turbo Fine-tuned (m3)

- Highest BLEU score: 0.163 (45% better than closest GPT)
- Strong cosine similarity: 0.831
- Consistent similarity rate: 83%
- Balanced across all metrics

### Surprising Result: davinci-002 Dummy (m0)

- Highest similarity rate: 85% with only 12 training samples!
- Strong cosine similarity: 0.824
- Interpretation: Inherent medical knowledge from pre-training

### T5 Model Challenges

- Both T5 models underperformed GPT models significantly
- Smaller model size (61M vs GPT's larger architecture)
- 4-bit quantization impact on precision
- LoRA updating only 0.97% may be insufficient
- Architecture may be less suited for this task

## Fine-Tuning Impact Analysis

**davinci-002:** Minimal improvement (slight decrease)

- Dummy (m0): 85% → Full (m1): 78%
- Conclusion: Model already had strong medical knowledge

**gpt-3.5-turbo:** Significant improvement

- Base (m2): 0.112 BLEU → Fine-tuned (m3): 0.163 BLEU (+45%)
- Conclusion: Fine-tuning effectively specialized the model

**T5:** Performance decreased after fine-tuning

- Base (mf0): 76% similarity → LoRA (mf1): 54% similarity
- Conclusion: Needs more aggressive tuning or different hyperparameters

## Qualitative Analysis - Sample Prediction

**Question:** "What is the relationship between very low Mg2+ levels, PTH levels, and Ca2+ levels?"

**Ground Truth:** "Very low Mg2+ levels correspond to low PTH levels which in turn results in low Ca2+ levels."

| Model | Response | Quality |
|---|---|---|
| m3 (gpt-3.5-turbo FT) | Low Mg2+ suppresses PTH secretion, leading to low Ca2+ | ✓ Accurate |
| m1 (davinci-002 full) | Mg required for PTH; low Mg→low PTH→low Ca2+ | ✓ Detailed |
| mf1 (T5 LoRA) | Low PTH levels correspond to low calcium | ✗ Incomplete |

## Error Analysis - Common Error Types

- **Incomplete Answers (T5 models):** Missing key relationships, over-simplification
- **Hallucination (rare):** Introducing facts not in ground truth (more in base models)
- **Over-verbosity (GPT models):** More detail than necessary, affects BLEU
- **Medical Terminology Errors:** Imprecise vocabulary (T5 > GPT)

# 7. Technical Challenges & Solutions

## Challenge 1: Memory Constraints

**Problem:** Training large language models requires significant GPU memory.

**Solutions:**

- 4-bit Quantization (BitsAndBytes): 8x memory reduction using NF4
- LoRA Adapters: Train only 0.97% of parameters
- Gradient Accumulation: Small batch sizes (3-4) accumulated over steps
- Result: Successfully trained T5-Small on consumer-grade GPUs

## Challenge 2: Data Quality & Preprocessing

**Problem:** Medical text contains complex terminology and varied answer lengths.

**Solutions:**

- Duplicate Removal: 339 duplicates removed from 27,503 samples
- Length Handling: Input max 128 tokens, output max 512 tokens
- Prompt Engineering: Added prefix for T5, system message for gpt-3.5-turbo
- Consistent formatting across all models

## Challenge 3: Evaluation Methodology

**Problem:** Single metrics don't capture full model performance.

**Solutions:**

- Multi-metric evaluation: Cosine similarity + BLEU + Similarity rate
- Sample-based testing: 100 random questions (computational constraints)
- Parallelized metric computation for efficiency
- Cached embeddings to reduce computation

## Challenge 4: Cross-Domain Generalization

**Problem:** Medical fine-tuning might destroy general knowledge.

**Solutions:**

- Tested on 205 general knowledge questions (physics, chemistry, biology)
- Monitored performance on non-medical topics
- Used appropriate learning rates to prevent catastrophic forgetting
- Finding: GPT models maintained general knowledge better than T5

# 8. Key Learnings & Future Work

## Key Learnings

### 1. Pre-trained Knowledge Matters

davinci-002 with minimal tuning (12 samples) achieved 85% similarity, indicating strong inherent medical knowledge.

### 2. Model Architecture Selection Critical

GPT models significantly outperformed T5 due to larger capacity and better pre-training on medical text.

### 3. Fine-Tuning Strategy Varies by Model

gpt-3.5-turbo: +45% BLEU improvement | davinci-002: minimal change | T5: performance degradation

### 4. Parameter-Efficient Methods Have Limits

LoRA (0.97% trainable) may be too restrictive for T5-Small. May need higher rank or more modules.

### 5. Quantization-Performance Trade-off

4-bit quantization enabled training on limited hardware but may have impacted T5 performance.

### 6. Multiple Metrics Essential

BLEU, cosine similarity, and similarity rate each provide different insights. Need qualitative analysis too.

### 7. Medical Domain Complexity

Average BLEU scores (0.02-0.16) lower than typical NLU tasks. Medical answers require precision and completeness.

## Limitations

- **Model:** T5-Small (61M) insufficient; 4-bit precision loss; LoRA 0.97% limiting; 128 input tokens may truncate
- **Data:** Flashcard-style only; single ground truth per question; only 205 cross-domain samples
- **Evaluation:** 100 test samples (compute constraints); limited manual review; metrics don't capture clinical correctness
- **Resources:** Limited GPU memory; OpenAI API costs; training took ~8 hours

## Future Work

- **T5 Enhancement:** Larger variants (Base, Large); 8-bit quantization; increase LoRA rank (r=32, 64)
- **Advanced Fine-Tuning:** Other PEFT methods (Prefix Tuning, Adapters); curriculum learning; data augmentation
- **Expanded Evaluation:** Human expert evaluation; task-specific metrics; adversarial testing

### Medium-Term Extensions:

- Multi-modal integration (medical images, X-rays, MRIs)
- Retrieval-Augmented Generation (RAG) with medical literature database
- Domain-specific pre-training on PubMed corpus
- Conversational context support (multi-turn dialogue)

### Long-Term Vision:

- Clinical deployment with HIPAA compliance
- Specialized models per medical domain (radiology, cardiology, etc.)
- Explainability & safety (attention viz, uncertainty quantification)
- Continuous learning from new medical research

# Conclusion

## Project Summary

- **Comprehensive EDA:** Word2Vec, clustering, linguistic features, keyword indexing
- **Multiple Architectures:** 6 models evaluated (T5 + GPT families)
- **Advanced Techniques:** LoRA, 4-bit quantization, parameter-efficient fine-tuning
- **Rigorous Evaluation:** Multi-metric assessment on 100 test samples
- **Clear Winner:** gpt-3.5-turbo fine-tuned (83% similarity, 0.831 cosine, 0.163 BLEU)
- **Valuable Insights:** Pre-trained knowledge importance, architecture criticality, tuning variability

## Technical Contributions

- Parameter-efficient training demonstration (LoRA potential and limitations)
- Quantization trade-offs documentation (4-bit impact analysis)
- Model comparison framework for diverse architectures
- Cross-domain testing validation methodology

## Practical Impact

This project provides a foundation for medical AI assistants that could support healthcare professionals with rapid knowledge retrieval, enhance medical education through interactive Q&A;, improve patient understanding, and accelerate research through automated literature analysis.

## Final Thoughts

While gpt-3.5-turbo fine-tuned achieved strong performance, medical AI requires continuous improvement to meet clinical standards. The combination of thorough data analysis, multiple model comparison, and rigorous evaluation provides a solid framework for future medical NLP systems.

# Questions & Discussion

## Thank you for your attention!

**Ready to discuss:**

- Technical implementation details
- Alternative approaches and trade-offs
- Model architectures and training strategies
- Evaluation methodology and metric selection
- Future directions and potential collaborations
- Ethical considerations in medical AI
- Deployment challenges and solutions

# Appendix: Technical Specifications

## Repository Structure

```
Final_Submission/
■■■ pre_analysis.ipynb (EDA)
■■■ linear_classifier_&_evaluation_of_the_models.ipynb (Comparison)
■■■ t5_adapters+metrics.ipynb (T5 fine-tuning)
■■■ GPT-FT-merged-for-presentation.ipynb (GPT fine-tuning)
■■■ GPT-FT-TESTING-merged-for-presentation.ipynb (GPT evaluation)
■■■ model_results/ (CSV results for all models)
■■■ images/ (Visualization outputs)
```

## Key Dependencies

transformers >= 4.30.0 | datasets >= 2.12.0 | peft >= 0.4.0 | bitsandbytes >= 0.39.0 | torch >= 2.0.0 | evaluate >= 0.4.0

sentence-transformers >= 2.2.0 | openai == 0.28 | nltk >= 3.8 | spacy >= 3.5 | scikit-learn >= 1.2.0

## Reproducibility Notes

- Random Seeds: Set to 42 throughout
- Data Split: Stratified 80-10-10 with seed 42
- Model Checkpoints: Available upon request
- OpenAI Models: Model IDs provided for exact reproduction

## Contact Information

**Authors:** Rishabh Tiwari, Dan Lionis, Federica Vinciguerra, Felipe Bagni, Erfan Amidi

**Project Repository:** https://github.com/Icon1cc/NLP-Polimi-Project