
Question Answering — Task-Specific Models vs. LLMs

Zongyan Yao

Department of Electrical and Computer Engineering
University of Toronto
zongyan.yao@mail.utoronto.ca

Zhengyang Li

Department of Electrical and Computer Engineering
University of Toronto
zhengyang.li@mail.utoronto.ca

Qiwen Lin

Department of Electrical and Computer Engineering
University of Toronto
qw.lin@mail.utoronto.ca

Abstract

Large Language Models (LLMs) have shown strong performance on open-domain question answering, but they often lack the domain-specific precision required for specialized tasks. This project investigates whether fine-tuning smaller pretrained transformer models can outperform general-purpose LLMs on a narrow recipe-focused benchmark. Using the Recipe-MPR dataset of 500 multi-perspective cooking questions, we fine-tuned several encoder-based models (BERT, DistilBERT, RoBERTa) as well as a mid-sized decoder model (Qwen2.5–7B with LoRA). Our goal is to exceed the 65% accuracy threshold and to analyze the trade-offs between accuracy, efficiency, and model capacity. Initial results show that all fine-tuned encoder models exceed the target, with BERT-large achieving 91.4% accuracy and DistilBERT providing strong performance at lower computational cost. The LoRA-tuned Qwen2.5–7B model achieves 100% accuracy, demonstrating the potential of parameter-efficient fine-tuning on modern architectures. These findings highlight the benefits of domain adaptation and provide a basis for a deeper comparison between specialized and general-purpose models in the final report.

1 Introduction

Large Language Models (LLMs) such as GPT and BERT have demonstrated impressive performance on open-domain question answering tasks. However, their general-purpose design means that they may overlook domain-specific nuances, especially in specialized areas such as culinary instructions or recipe-based reasoning. Fine-tuning smaller and mid-sized transformer models provides a practical method to improve interpretability, efficiency, and task performance.

The Recipe-MPR dataset consists of 500 queries, each paired with five answer variations. This dataset is designed for multi-perspective question answering within the recipe domain and represents a challenging, domain-specific benchmark. The objective of this project is to fine-tune several pretrained transformer models on this dataset, surpass the accuracy baseline, and compare their performance against a general-purpose LLM prompted directly with the same queries.

2 Preliminaries and Problem Formulation

2.1 Problem Definition

Given a recipe-related query and five candidate response variations, the task is formulated as a **five-class text classification problem**. Each model must predict the correct label corresponding to the appropriate answer variation.

Formally, the model receives an input query q and outputs a class label $y \in \{1, 2, 3, 4, 5\}$.

2.2 Objective

The primary goals of the project include:

- Fine-tune multiple pretrained transformers (BERT, DistilBERT, RoBERTa, Qwen2.5-7B).
- Achieve at least **65% accuracy** and target performance above **75%**.
- Evaluate models using accuracy and macro F1-score.
- Compare fine-tuned models with a state-of-the-art LLM prompted directly.
- Analyze specialization vs. generalization trade-offs.

2.3 Relevant Background Concepts

Transformer Architecture All models used are transformer-based and rely on self-attention to capture contextual relationships. Key components include multi-head attention, positional embeddings, feed-forward layers, and layer normalization.

Fine-Tuning Fine-tuning adapts pretrained weights to a task-specific dataset by updating all or part of the model parameters. Compared with knowledge distillation, fine-tuning is simpler and more feasible for this project.

GPT-3 Embedding GPT-3 embeddings turn text into vectors (lists of numbers) that represent meaning. Texts with similar meaning get vectors that are close together. We could use them for semantic search, clustering. They capture context beyond keywords, and the vector size is fixed so it's easy to store and compare.

Evaluation Metrics Accuracy serves as the primary evaluation metric, with macro F1-score used to assess class-balanced performance.

3 Solution via Deep Learning

3.1 Dataset

The Recipe-MPR dataset includes 500 queries, each paired with five human-written answer variations and a label. We apply a 80/10/10 split for training, validation, and testing respectively. Preprocessing includes lowercasing, tokenization using model-specific tokenizers, and padding or truncation to a fixed sequence length.

3.2 Models Used

We fine-tuned several transformer families:

- BERT-base (standard and aggressive fine-tuning)
- BERT-large
- DistilBERT
- RoBERTa-base (standard and aggressive variants)
- Qwen2.5-7B with LoRA fine-tuning

- Llama3.2-3B and Llama3.2-1B with LoRA fine-tuning

Each model outputs logits over five classes.

3.3 Training Procedure

We fine-tuned each model on the Recipe-MPR training split using appropriate hyperparameters such as learning rate, batch size, number of epochs, and gradient accumulation steps. All models were evaluated on the 50-item test set. Qwen2.5-7B, Llama3.2-3B, Llama3.2-1B models were fine-tuned using LoRA to reduce memory requirements during training. We also compared the performance of models before and after training to see the improvement of fine-tuning. Finally, we tried to make a contrast with our llama models vs. GPT3-Embedding, which shows our exploration of a new technique, and the difference between fine-tuning a model under a specific dataset and a general Large Language Model.

3.4 Testing and Early Results

3.4.1 Qwen2.5-7B (LoRA Fine-Tuned)

The Qwen2.5-7B model demonstrates exceptional specialization on the Recipe-MPR task. After applying LoRA fine-tuning and training on a 32GB GPU, the model achieved a perfect score with no mistakes across 50 test examples. Table 1 compares the fine-tuned Qwen model with the base Qwen model and the top-performing BERT variants. Despite Qwen’s much larger parameter count, LoRA fine-tuning remains computationally tractable and yields perfect task performance.

Table 1: Performance Comparison: Qwen vs. BERT Models

Model	Parameters	Accuracy	Training Time	VRAM	Architecture
Fine-tuned Qwen	7B	100.00%	~15 min	20 GB	Decoder-only
Base Qwen	7B	79.20%	0 min	20 GB	Decoder-only
BERT-large	340M	91.4%	~93 sec	16 GB	Encoder-only
DistilBERT	66M	82.4%	~35 sec	6 GB	Encoder-only

3.4.2 BERT-Family Model Results

Training on a 32GB GPU, all BERT-family models meeting the 65% threshold are summarized in Table 2.

Model	Accuracy	Above Goal?
BERT-large	91.4%	Yes
DistilBERT	82.4%	Yes
BERT-base (aggressive)	67.6%	Yes
BERT-base (standard)	65.6%	Yes
RoBERTa-base (aggressive)	65.8%	Yes
RoBERTa-base (standard)	49.8%	No
DistilBERT (over-trained)	36.8%	No

Table 2: Performance of BERT-family models on the Recipe-MPR dataset.

3.4.3 Llama Models Results Vs. GPT3 Embedding

We fine-tuned Llama-3.2-3B, Llama-3.2-1B with LoRA to do recipe recommendation. LoRA lets us train only a tiny part of the model . We trained 5 epochs on 400 samples with 4-bit quantization on an 8GB GPU.

Results are clear. The 3B LoRA model got 84.00% in ~14 min. The 1B LoRA model got 78.00% in about 7 min. Both beat the 65% goal. The 3B zero-shot model reached 73.00% (meets goal), while

1B zero-shot was 58.00%. The GPT-3 Embedding baseline was 54.55%. Fine-tuning gave +11 points over 3B zero-shot, and LLM reasoning beat the embedding method by +18.45 points, for a total +29.45 points over the baseline.

All evaluated models with their accuracy and whether they meet the 65% threshold are summarized in Table 3.

Table 3: Performance of Llama models vs. GPT3 on the Recipe-MPR test set.

Model	Parameters	Accuracy	Training Time	Above Goal?
Fine-tuned Llama-3.2-3B	3.2B	84.00%	14 min	Yes
Base Llama-3.2-3B	3.2B	73.00%	0 min	Yes
Fine-tuned Llama-1B	1B	78.00%	~7 min	Yes
Base Llama-1B	1B	58.00%	0 min	No
GPT-3 Embedding	N/A	54.55%	N/A	No

3.5 Progress Summary

The following tasks have been completed:

- Dataset preprocessing and splitting
- Implementation of all fine-tuning pipelines
- Full training of BERT-family models
- LoRA fine-tuning of Qwen2.5-7B
- LoRA fine-tuning of Llama3.2-3B, Llama3.2-1B
- Compare Llama to GPT3 embedding
- Full testing and evaluation
- Cross-model comparison
- Initial performance analysis

The project is on schedule, with all core components completed.

References

Include all references here. It's important to have your references cited.

- [1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609–616. Cambridge, MA: MIT Press.
- [2] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GENeral NEural SImulation System*. New York: TELOS/Springer-Verlag.
- [3] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.

Appendix

Any descriptions about supplementary materials go here.