Project Proposal: Question Answering — Task-Specific Models vs. LLMs

Course Project

Code of Honor. All external resources used in the project, including research papers, open-source repositories, datasets, and any content or code generated using AI tools, e.g., ChatGPT, GitHub Copilot, Claude, Gemini, must be *clearly cited* in the final submission. The final report must also include *a clear breakdown of individual group member contributions.* Any lack of transparency in the use of external resources or in reporting group contributions will be considered academic dishonesty and will significantly impact the final evaluation.

Topic	Question Answering — Task-Specific Models vs. LLMs
Dataset	Recipe-MPR multi-perspective recipe QA

Background

Large language models (LLMs) perform strongly on open-domain question answering, yet their general-purpose nature can miss task-specific nuances. In specialized domains, fine-tuning a focused model often yields gains in efficiency, interpretability, and reliability. The Recipe-MPR dataset, containing 500 user queries with five answer variations apiece, offers a benchmark for multi-perspective responses in the recipe domain and enables a grounded comparison between specialized and general-purpose approaches.

Objective

Fine-tune several task-specific transformer models on Recipe-MPR to surpass the 65% baseline accuracy while benchmarking performance, robustness, and efficiency against prompted LLM outputs. Then analyze tradeoffs among the models.

Motivation

According to researchers [1], nowadays, the rise of large language models (LLMs) has improved a lot in dealing question answering (QA). However, most LLMs need large amounts of data to build their own corpus and also require tons of computational resources. Therefore, how to reduce hardware and dataset threshold while maintaining the performance with original LLMs is a hot topic for researchers.

Currently, there are two main streams to reach this goal: Knowledge Distillation and Fine-Tuning[2]. As noted by Hinton et al.(2015)[3], Knowledge distillation using teacher-student models to transfer knowledge. When it comes to Fine-Tuning technology, it is based on a pre-training model (e.g. BERT, GPT). According to the view of Devlin et al.(2019)[4] , Fine-Tuning technology is a standard method to apply BERT to fit for other specific tasks, especially for focusing on several small majors. As for process, what we need to do is just to use a small training dataset to tune some learnable parameters for per-trained models in order to match our needs.

Now comparing these two methods, from our perspective, although Knowledge Distillation could transfer the features of dataset to small model, it would be complex for us, since it requires a well-designed transfer process and also needs a good enough teacher to generate Soft Labels so it can pass enough critical information to student model. Furthermore, according to Fei et al.(2021)[5], to make student model performance effectively, more teachers should be employed with a special combination method to weight all output from teachers. In contrast, Fine-Tuning has lower entry threshold because the pre-training model has a lot of general knowledge, so the only thing to do is to provide different datasets which match our objective.

By contrasting several domain-adapted models with a general-purpose LLM, we aim to surface the trade-offs between specialization and broad capability. Understanding when a dedicated model outperforms or complements a large foundation model can inform deployment decisions in domains that require efficiency, transparency, or domain fidelity.

Requirements

The final submission should address the following requirements while the details can be freely decided by the group members.

1. **Model Training:** Fine-tune several pretrained transformers (BERT, Llama 7B, Qwen) on Recipe-MPR to capture recipe-domain subtleties.

2. **Evaluation:** Report accuracy and F1-score on a held-out test split to measure each task-specific model performance.

3. **Comparison:** Two stage comparison. First prompt a state-of-the-art LLM with identical queries, score its responses using the same metrics, and compare with the fine-tuned model. Next comparing the performance among all models to find which is best.

4. **Analysis:** Provide qualitative and quantitative analyses to highlight scenarios where the specialized model excels or complements the LLM baseline. Also analysis the advantages and shortages for all fine-tuned models.

Milestones

The following milestones are to be accomplished through semester.

1. **Weeks 1:** Conduct literature review on recipe QA, finalize dataset handling, and prepare the development environment.

2. **Weeks 2:** Data Processing (e.g. Cleaning and preparing the Recipe-MPR dataset) and word embedding.

3. **Weeks 3–4:** Fine-tune the selected transformer models variant on Recipe-MPR.

4. **Week 5:** Evaluate all the task-specific models and benchmark against LLM-generated answers.

5. **Week 6:** Perform comparative analysis and draft key findings.

6. **Final Week:** Complete the written report and prepare the project submission materials.

Expected Outcomes

We anticipate the fine-tuned model achieving accuracy comparable to or exceeding that of prompted LLMs within the recipe domain. The project should demonstrate the efficiency, domain adaptability, and interpretability gains of specialized models while clarifying the circumstances in which reliance on general-purpose LLMs remains advantageous.

Submission Guidelines

The main body of work is submitted through Git. In addition, each group submits a final paper and gives a presentation. In this respect, please follow these steps.

- Each group must maintain a Git repository, e.g., GitHub or GitLab, for the project. By the time of final submission, the repository should have
  - Well-documented codebase
  - Clear `README.md` with setup and usage instructions
  - A `requirements.txt` file listing all required packages or an `environment.yaml` file with a reproducible environment setup
  - Demo script or notebook showing sample input-output
  - *If applicable,* a `/doc` folder with extended documentation
- A final report (maximum *5 pages*) must be submitted in a PDF format. The report should be written in the provided formal style, including an abstract, introduction, method, experiments, results, and conclusion.
  Important: Submissions that do not use template are considered *incomplete.*
- A 5-minute presentation (maximum *5 slides including the title slide*) is given on the internal seminar on Week 15, i.e., *Dec 8 to Dec 12,* by the group. For presentation, any template can be used.

Final Notes

While planning for the milestones please consider the following points.

1. You are encouraged to explore innovative approaches to conditioning or generation as long as the core objectives are met.
2. While computational resources are limited, carefully chosen datasets and training setups can make even diffusion models feasible. Trade-offs, e.g., resolution, training steps, are expected and should be justified.
3. Teams are expected to manage their computing needs and are advised to perform early tests to estimate runtime and training feasibility. As graduate students, team members can use facilities provided by the university, e.g., ECE Facility. Teams are expected to inform themselves about the limitations of the available computing resources and design the model accordingly.

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. Advances in neural information processing systems, 33:1877–1901, 2020.

[2] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. Journal of Machine Learning Research, 20(55):1–21, 2019.

[3] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proc. Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4171–4186, 2019.

[5] Fei Yuan, Linjun Shou, Jian Pei, Wutao Lin, Ming Gong, Yan Fu, and Daxin Jiang. Reinforced multi-teacher selection for knowledge distillation. In Proceedings of the AAAI conference on artificial intelligence, volume 35, pages 14284–14291, 2021.