Project Proposal

1. Title: “Small-Scale Transformer vs LLM Response Comparison”

2. Team Members: Junyao Liu, Hanxiao Wang

3. Motivation:

Large Language Models (LLMs) such as GPT-4 and Claude have demonstrated remarkable capabilities in open-domain dialogue and reasoning. However, these models require enormous computational resources, incur high inference costs, and are often deployed as closed-source APIs, making them difficult to analyze. In contrast, small-scale Transformer models like DialoGPT are lightweight, open, and easily fine-tuned on task-specific data, but they are limited in conversation quality.

Understanding how small Transformer models differ from LLMs in fluency, coherence, and reasoning can shed light on the scaling behavior of Transformer architectures and inform future research on efficient conversational AI. Our project aims to systematically compare small Transformers and LLMs across multiple dialogue scenarios to identify which linguistic and behavioral gaps persist despite sharing the same underlying Transformer architecture.

4. Research Question:

(1)Quality Difference: How do small Transformers compare to LLMs in terms of fluency, relevance, coherence, and empathy across different dialogue scenarios (casual, empathetic, and instruction-following)?

(2)Task Generalization: To what extent can small Transformers follow user instructions or maintain factual consistency without extensive instruction-tuning?

(3)Efficiency Trade-offs – What are the computational and cost advantages of small models, and how do these trade-offs affect dialogue quality?

5. Existing Research:

Transformer-based language models have been studied for conversational AI. Early approaches such as BlenderBot[1] fine-tuned GPT-like architectures on large conversation corpora, achieving fluent but sometimes shallow responses. Later research introduced instruction-tuned LLMs,

including InstructGPT[2], Alpaca[3], and Llama-2/3-Instruct[4], demonstrating strong instruction-following and reasoning capabilities through large-scale supervised fine-tuning.

However, most prior studies either focus exclusively on large models or on task-specific fine-tuning of smaller models without a systematic comparison across scales. The relationship between model size, training objectives, and dialogue quality remains underexplored. In particular, there is limited empirical analysis of how small-scale Transformers behave relative to modern LLMs in terms of fluency, coherence, empathy, and instruction adherence under the same dialogue prompts.

Our work addresses this gap by conducting a controlled evaluation of small Transformer models (DialoGPT-small) and instruction-tuned LLMs (e.g., GPT-3.5, Llama-3-Instruct) across multiple dialogue scenarios. This comparison aims to find which aspects of conversational quality are primarily driven by scaling versus alignment strategies, providing insight into efficient conversational AI design.

References

[1]Shuster, Kurt, et al. "Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage." arXiv preprint arXiv:2208.03188 (2022).

[2]Ouyang, Long, et al. "Training language models to follow instructions with human feedback." Advances in neural information processing systems 35 (2022): 27730-27744.

[3]Maeng, Kiwan, Alexei Colin, and Brandon Lucia. "Alpaca: Intermittent execution without checkpoints." Proceedings of the ACM on Programming Languages 1.OOPSLA (2017): 1-30.

[4]Touvron, Hugo, et al. "Llama: Open and efficient foundation language models." arXiv preprint arXiv:2302.13971 (2023).

6. Technical:

(1)Model: We use two models for comparison.

-Small Transformers (trainable): microsoft/DialoGPT-small (≈124M), which is a decoder-only Transformers with masked self-attention.

-LLM baselines (inference-only): GPT-3.5 / Llama-3-Instruct via API for zero-shot/few-shot prompting under fixed decoding settings.

(2)Parameter-Efficient Fine-Tuning (PEFT).

To fit our computational constraints, we apply LoRA/QLoRA on attention projections:

$W \approx W0 + \alpha AB$, $A \in R^{d \times r}$, $B \in R^{r \times d}$, $r \ll d$,

freezing $W0$ and updating only (A,B). We use 8-bit (or 4-bit) weight quantization for the frozen backbone to reduce memory, with trainable LoRA ranks $r \in \{4,8,16\}$.

(3)Optimization.

We fine-tune the small Transformer using the causal language modeling objective with AdamW optimization and a cosine learning-rate schedule. Dropout and early stopping improve generalization, while mixed-precision and gradient accumulation enable efficient training on Google Colab's GPU resources.

(4)Data.

We employ two publicly available dialogue corpora to ensure both linguistic diversity and emotional coverage.

-DailyDialog: an open-domain dataset of multi-turn daily conversations used for fine-tuning and evaluation of general conversational fluency and coherence.

-EmpatheticDialogues: a set of emotionally grounded dialogues covering 32 emotion categories, used to assess empathy and contextual understanding.

This combination allows us to evaluate model performance across both general and emotion-rich conversational contexts while keeping the overall data volume feasible for training on Google Colab.

(5)Prompt Templates.

Chit-chat/Empathy: [User]: ... <eos> [System]:

Instruction: You are a helpful assistant. Task: ... <eos> Answer:

(6)Evaluation Metrics.

To quantitatively assess generation quality, we compute three automatic metrics on the held-out test sets:

-BERTScore to measure semantic similarity between generated and reference responses at the embedding level;

-BLEU to evaluate n-gram lexical overlap and surface accuracy;

-Distinct-n (Distinct-1 and Distinct-2) to estimate response diversity and mitigate repetitive patterns.

These metrics provide a comprehensive evaluation of dialogue quality for both the fine-tuned small Transformer and the LLM baselines.

7. Timeline/Milestones:

Timeline	Milestones	Expected Outcomes
Oct 13 - 17	Conduct literature review on Transformer-based dialogue models; Finalize project scope, dataset, and model setup; Set up Google Colab environment and verify LLM API access	Clear project direction and scope; Configured Colab environment and confirmed dataset/model accessibility
Oct 18 - 24	Preprocess datasets (cleaning, tokenization, and prompt formatting); Design unified dialogue prompt templates for both models; Define automatic evaluation metrics	Cleaned and formatted datasets; Unified input format; Defined evaluation plan
Oct 24 - 31	Parallel model development: – Fine-tune DialoGPT-small with LoRA and mixed-precision on Colab – Generate LLM (GPT-3.5 / Llama-3) responses for the same prompts; Monitor training loss, perplexity, and record sample outputs	Fine-tuned small Transformer model; LLM response dataset collected; Stable training and comparable outputs
Nov 1 - 7	Compute automatic metrics for both models; Compare output quality and efficiency (inference latency, token length); Refine prompt or evaluation scripts based on results	Quantitative comparison between small Transformer and LLM; Adjusted evaluation setup for consistency
Nov 8 - 14	Conduct evaluation; Visualize results with radar and bar charts	Obtained evaluation dataset; Visualized results and qualitative analysis
Nov 15 - 19	Summarize findings and analyze trade-offs between both models; Discuss limitations and future improvements; Finalize report, slides, and presentation demo	Completed report and presentation; Finalized conclusions and deliverables

8. Team Member Roles:

Team Member	Main Responsibilities
Hanxiao Wang	• Collaboratively prepare and preprocess datasets (DailyDialog and EmpatheticDialogues) • Fine-tune DialoGPT-small using LoRA and mixed-precision on Colab • Optimize training with AdamW and cosine learning-rate schedule • Compute automatic metrics (BERTScore, BLEU, Distinct-n) for small Transformer outputs • Co-author discussion, analysis, and presentation materials
Junyao Liu	• Collaboratively prepare and preprocess datasets (DailyDialog and EmpatheticDialogues) • Generate LLM (GPT-3.5 / Llama-3) responses for the same inputs • Compute automatic metrics (BERTScore, BLEU, Distinct-n) and record latency / token cost • Perform qualitative error analysis and interpret metric results to identify model strengths / weaknesses • Co-author discussion, analysis, and presentation materials