experiments conducted/ planned for LLM Blending work

Experiment 1 (This was done by Vishal and his team already)

- **Aim:** To observe the performance of different LLMs against the LLM-BLENDER framework
- **What was done:** Questions from the **ConvQuestions and Atlas-Conversation datasets** were passed through the member LLMs to generate the output, and scores were noted based on reference answers. The LLM generated answers were then passed to LLM-BLENDER framework. The framework generated responses were then evaluated similarily.
- **Dataset and setup:** The datasets used were 2,000 conversations each from **ConvQuestions and Atlas-Conversation**, which are conversational question-answering datasets. ConvQuestions has 5 questions in each conversation, Atlas has varying length of conversation but on average it is like 10 questions.
- **Expected output:** Scores of member LLMs and the LLMBlender
- **Summary/Usage:** The experiment yields a bar chart, y-axis as score and LLMs as different bars **LLMBlender is better than any single LLM**

Experiment 2 (New and included in our draft)

- **Aim:** To observe the ranking of different LLMs inside the LLM-BLENDER framework
- **What was done:** The LLM generated answers in experiment 1 were then passed to LLM-BLENDER framework. The ranker module generates ranks as intermediate step, those ranks were printed for analysis
- **Setup:** The generated list of ranks for each question for each conversation was obtained, we were checking is it the case that for all questions only one LLM is ranked within top K rank and only its answer contributes. The answer was NO, depending on the topic and question different LLMs rank differently
- **Expected output:** Counts of number of times a LLM gets Rank 1, then similarly Rank 2, and so on
- **Summary/Usage:** a bar chart, that shows Distribution of rankings across LLMs for both datasets. Different bars represent different ranks from 1 to 10 as left to right. Different color pieces represeent the different LLMs. The length of the colored pieces represent the percentage of times of getting that particular rank by the LLM-Blender

Experiment 3

- **Aim:** To study the percentage of time taken by each module component in the LLMBlender

- **What was done**: Fix a particular N (number of LLMs) and observe the time in generation, ranking and fusion.
- **Expected output**: As the N grows, the ranking becomes a bottleneck, it grows to 30-40 seconds and takes significant percentage of the total time, acting as bottleneck
- **Summary of output**: As stacked bar chart with time of different component in particular N, and as the fixed length bar chart with percentage

Experiment 4

- **Aim:** To study the trade-offs between response quality and speedup when avoiding full ranking, comparing different strategies with a full ranking baseline.
- **What was done**: Four different ranking strategies were compared : **Full Ranking, Fixed-Interval Elimination, Dynamic Conversation-Specific Elimination, and Combining Interval based and Conversation Specific - Alternate Ranking**.
- **Dataset and setup:** The experiments used the **ConvQuestions and Atlas-Conversation datasets**. The setup included implementing and comparing different ranking strategies within the LLM-BLENDER framework.
- **Expected output**: The expected output was to measure the performance of each ranking strategy with respect to quality of responses and latency.
- **Summary of output**: **Full Ranking** provided the highest quality at the highest latency. **Fixed-Interval Elimination** offered a balance of speedup and moderate quality degradation. **Dynamic Conversation-Specific Elimination** achieved near-optimal quality with significant latency reductions. The **Alternate Ranking** strategy maintained a good quality with less latency.

Experiment 5

- **Aim:** To understand the impact of the parameter K on the quality of final answers produced by fusing responses from multiple LLMs.
- **What was done**: Two main experiments were conducted: **Random Fusion of (− 1) LLMs** where a number of LLMs were randomly selected and their responses fused. **Varying K in Fusion**, where K was systematically varied from 1 to N.
- **Dataset and setup:** The study used the **ConvQuestions and Atlas-Conversation datasets**. The setup involved the LLM-Blender framework.
- **Expected output**: The expected output was to observe how varying the number of models used in the fusion process affects both quality and computational efficiency.
- **Summary of output:** The random fusion approach resulted in a lower quality output than the baseline approach. For the varying K experiment, the study found that quality increased with K up to a threshold, after which the quality degraded, also showing that ranked fusion is better than random fusion. Latency was shown to increase linearly with K.

Experiment 6

- **Aim:** To understand the tradeoff of time and quality
- **What was done**: plotting a final graph that shows the goodness of the tradeoff with a point. We take the full ranking blender, and the approximated ranking strategy - policy 1, policy 2 and policy 3.