DevRev PS Midterm updates

Resource constraints

- Resource constraint for the final evaluation remains the same i.e., google collab free tier notebook without any hardware accelerator.
- Training and synthetic data generation techniques should complete within 12 hrs as mentioned earlier with above resource limits.

Latency constraints

- Latency constraint has been revised from **200ms to 1sec.** Sample evaluation notebook has been updated accordingly. Please note that mid term report evaluation is done on the basis of 200ms constraint. So, you would have earned points if you tried to keep it around 200ms.
- So, if it goes above 1 sec., your score would be scaled by 1/(inference time(sec)).
- If you keep it below 1 sec, no boost in metric scores would be there. But definite bonus points in the other sections if you do the work to bring down the latency 🙂.

Evaluation rounds

- In the original problem statement, it was mentioned that there would be two rounds of evaluation. First one where you can fine tune on the basis of new paragraphs and the next one, where you can also take in account the given QA pairs to further fine tune your model.
  Going through mid term reports, we realized teams haven't worked on the second part yet which can potentially lead to huge score improvements. To make it concise and clear, we are combining the **two rounds of evaluation into one.** Problem statement remains the same, only the evaluation mechanism has been combined into one to ease the process.
- On 5th February (tentative evaluation day), you would be given 30 new themes and their paragraphs. Also, for each theme, you may receive a few question answer pairs, i.e., for a question, a paragraph which answers it and an answer text from that paragraph.
  At this point, you are supposed to fine tune models if you'd like to for each theme. After **6 hours**, you'd receive questions for these new 30 themes. You have to run your notebook and submit your predictions for the same. We too would run your notebook on a random sample and it must match your submitted predictions. You'd be given **4 hours** to make the predictions for ~10k questions.
- Each theme would have its own weightage which would remain hidden.
- A sample inference for a question could look like this:
  - Check if a similar question has been answered before from the given QA pairs. If it's the case, you can just return the same paragraph_id and answer text.
  - If the above case is not valid, retrieve the paragraph from that theme which can answer the question. Here, you can have theme specific models to assist you better in paragraph retrieval.
  - Once the paragraph is retrieved, use your QA module to get the answer text. Here again, you can have a theme specific model to assist you better in getting the answer text.
- In the final report, template of which will be shared soon, we would also expect you to help us understand the economics of using a fine tuned model for a theme *vs* using a generic model for all themes.