Programming Assignment 1
Query Likelihood Model and LMs

*Tentative Deadline: **10th Sep**

In probabilistic IR, one way to express document relevance is expressed by Query Likelihood p(q|d), that is the likelihood that the document generated the query. Its is easy to see that p(d|q) is proportional to p(q|d) by Bayes rule.

In this assignment we are going to explore how is the performance gained or lost by formulating p(q|d) in different ways. The task is a standard document re-ranking task, where you are given a query and your job is to retrieve the gold documents from a small set of documents.
For both of the datasets below, the training set contains only the gold documents. To mimic a reranking task, you will need to add negatives (in proper proportion)

Datasets:

- **HotpotQA**
  - Training set does not contain negatives, make sure to add them
  - Test on the Dev(distractor) dataset
- **WikiNQ**
  - Training set contains query and its positive and negative documents
  - In the query likelihood setting, you need to extract negatives by yourself, in the same way as above
  - In the document likelihood setting, negatives are already provided
  - Evaluate on the test set

For all the parts, assume the Input is fed to a BERT-like LM as [CLS] [Doc] [Query] [Sep], with the output logits of query words [q1, q2, q3….qn] as o1, o2, o3….on, with them being n1, n2, n3… after normalisation (usually softmax).

1) **QueryLikelihood: Sum of log of normalised-logits over whole query**

- If p(q|d) is modelled as the sum of log(n_i) where n_i is the ith query logit (normalised).
- The sum is only over the query logits
- Input format to bert can be: [CLS; document; SEP; query]

2) **QueryLikelihood: Direct Modelling via FFL over CLS readout**

- [CLS; document; SEP; query]→ [CLS] → FF → score, that is instead of reading out the query logits, we directly read the CLS logit and apply a non-linear(possibly) transform via FFL to input a score.
- Experiment with varying depths of FFL. Don't forget to make use of tricks like Batch-Norm, dropout, etc.

3) **DocumentLikelihood: Sum of log of normalised-logits over whole document**

- If p(d|q) is modelled as the sum of log(n_i) where n_i is the ith document logit (normalised).
- Similar to part 1 except the logits considered are of documents and not the query
- The sum is only over the document logits
- Observe that the negatives to be fetched are now negative documents and **not** queries, as has been explained in the tutorial

4) **DocumentLikelihood: Direct Modelling via FFL over CLS readout**

- Same as part 3
- Observe that the negatives to be fetched are now negative documents and **not** queries, as has been explained in the tutorial
- Input format to bert can be: [CLS; document; SEP; query]

Intermediate representations (Extra credit, Might consume a lot of GPU) :
For each of the above parts, experiment with how predictions are affected when you take intermediate layer logits instead of taking the last layer logits. For example, for a model like BERT having 12 layers, test how predictions vary when logits are taken from the 6th layer instead of the 12th layer.

Model Evaluation procedure:
For each dataset, do the following -

- Explore the following strategies for adding negatives: Inbatch, Random from entire Corpus
- Report Precision@1, Precision@10, MRR, MAP on the test data of each dataset.

General Hints:
You may use either Google Collab or Kaggle. However Kaggle has a more generous 30 hr per week GPU quota, so please make use of it. Don't forget to terminate GPU usage when you don't need it.

Submission Protocol:
Moodle will be used for the submission of the assignment.

- All the findings need to be put down into a report file.
- Zip all the files used for building/training/testing of the models along with the report into a single file and then submit it on Moodle. Name the compressed file as [Roll1_roll2.(zip/tar.gz)]

Note - Only one member of the team should submit the assignment on Moodle.

In-class notes:

1. measure p(d|q) using BERT, expect that it's worse than the usual p(q|d)
2. Contextualization vs layers - reduce number of layers and see how much worse the prediction gets