546 - assignment 6
Report
Jaideep Rao

- **Description of the system, design tradeoffs, questions you had and how you resolved them, etc. List the software libraries you used, and for what purpose.**

The design of the system remained largely unchanged from the inference network project. The only addition was a new PriorNode class that also implemented the querynode interface. I also modified the code for indexBuilder to create the prior probabilities map at the same time as the creation of the index to save additional computation. I chose to write it to disk and read only the required prior on demand rather than loading it back up into memory along with the other components of the index. I used the (docId-1)*8 as the index for the value as Double would take up 8 bytes. For the random prior I just used random values ranging between 0.0 and 1.0. I did not need to use any new libraries for this project over the ones already being used in the inference network (mainly the json-simple jar)

- **What is the difference between your two query runs? Why would it be that way? Be specific.**

The difference between the two query runs is that the produced ranked list for the uniform prior matches that of the one produced by the and node in the inference network project, whereas the list produced by the random prior does not. This would be because the uniform prior adds the same score value for each document, effectively not changing any document scores relative to each other. The random prior on the other hand unequally influences all document scores in a random manner, thus it changes the ranking of the documents. In addition to this, The uniform prior always adds log(1/documentCount) to each document's score, whereas the random prior might supply much larger values, causing the random prior scores to generally be higher than the ones found in the uniform prior ranked list

- **How should the priors be stored in the index? Raw probabilities? Log probabilities? Some other value? What should drive your choice? Be specific.**

I stored raw probabilities in my index. I think this makes the index more generalizable and these prior values can be used by anyone using any kind of system as they can always modify the values appropriately for their use. Storing log probabilities or any other kind of specialized form of probabilities directly into the index would either force everyone to conform to that paradigm or force them to go out of their way to get raw probabilities and use them some other way (assuming they know what form the values are stored in the

index in the first place). Thus, it made the most sense to me to go with raw probabilities and convert them to log space in my priorNode implementation as this computation was easy and inexpensive to perform