# COMPSCI546_132815_FA20-Applied Information Retrieval Fall 2020

Moodle home / My courses / COMPSCI546_132815_FA20 / Beyond Bag of Words / Programming: Query Independent Features

## Programming: Query Independent Features

# Query Independent Features

The purpose of this project is to explore query independent features on a small collection of documents. You will use the collection, your index, and your retrieval APIs from previous projects.

## Implement a prior operator for your retrieval API

A prior is a document dependent feature, represented by a value that can be interpreted as a probability. Your Index will need to provide a way to store and access the values. The operator will need to access the values for composition with other values in your scoring function.

### Evaluation

Test your operator by creating  two prior probability values for the shakespeare collection. The first should be uniformly distributed. Name it uniform. The second should be randomly distributed. Name it random. Save each set of document probabilities in a separate text file to turn in with the assignment. Evaluate the query: "the king queen royalty" using the uniform prior and the random prior. Save the results of each in trecrun format, as per previous projects.

Use belief_AND as your evidence combination method, eg, in galago's query language, the two queries would be

#combine(#prior(uniform) the king queen royalty)

#combine(#prior(random) the king queen royalty)

### Grading Rubric

## (5%) Submission is in the correct format.

A ***single*** archive file (zip, tar.gz) was submitted to Moodle and it contains at least the following contents:

- ***report.pdf*** - your report (see below)
- ***src/**** - your source code
- ***README***
  - It has instructions for downloading dependencies the code.
  - It has instructions for building the code.
  - It has instructions for running the code.
- Query output files: random.trecrun, uniform,trecrun
- Prior probability files: random.prior, uniform.prior

### (50%) Source code that implements retrieval with query independent features.

Please be aware that you may lose points for problems in your code even if it appears to run correctly. We expect to see code used for all parts of the assignment in your submission.

## (45%) The report discusses your work.

- (5%) 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.
- (20%) What is the difference between your two query runs? Why would it be that way? Be specific.
- (20%) How should the priors be stored in the index? Raw probabilities? Log probabilities? Some other value? What should drive your choice? Be specific.

## Submission status

| | |
|---|---|
| Submission status | No attempt |
| Grading status | Not graded |
| Due date | Friday, November 20, 2020, 11:59 PM |
| Time remaining | 9 days 8 hours |
| Last modified | - |
| Submission comments | ⊞ Comments (0) |

Add submission

You have not made a submission yet

◀ Evaluation

Jump to...

Linear Feature-Based Models ▶