# SI 650 / EECS 549: Homework 2 – Document Ranking

Due: Wednesday, October 14, 4:30pm

## Introduction

Do you ever have questions like "how long does coronavirus survive on surfaces," "what alcohol sanitizer kills coronavirus" or about"coronavirus immunity"? If so, then this assignment is for *you*. In Homework 2, you'll building a vertical search engine (of sorts) by designing document ranking functions specifically for coronavirus-related queries to match against documents containing those answers or related information. In fact, those three initial queries are direct examples of the 50 you'll use in this homework!

The goal of this assignment is to understand how a real retrieval system works. The major task in building an IR system is to implement a basic retrieval algorithm. This assignment has two parts. In the first, you'll re-implement the BM25 ranking function and Pivoted Normalization method for ranking. This algorithm is simple yet effective and should help you familiarize yourself with the basic features and data you'll likely use for part two. In part two, you'll implement *your own new ranking function*. Part two is where you will likely spend most of your time

To evaluate correctness, we'll be using Kaggle to have you submit your document rankings for each query. Your rankings will be evaluated using NDCG@10. The BM25 method will server simple-yet-effective method will serve as the baseline for how well retrieval could work.

To participate, you must join the Kaggle competition linked here: `https://www.kaggle.com/t/92a8757a190943b287cc2c7bb34f164f`. Please choose a username that we can identify as you or specify your Kaggle username in your homework submission itself.

This assignment is language agnostic and you are welcome to use any language (within reason). However, to get you started, we've put together a Jupyter notebook using the `metapy` package, which is introduced in the textbook, and to experiment with the algorithms on some sample text data. You can download the notebook from Canvas. You are welcome to run it locally or to run it on Google colab (https://colab.research.google.com), which is particularly recommended if you're running this on Windows.

## Implement the Pivoted Normalization and BM25 Retrieval Functions (50 points)

In general, you need a function that computes a score of every document by aggregating the weight of every word that occurs in both the document and the query:

$$s(q,d) = g[f(w_1,q,d) + ... + f(w_N,q,d), q, d],$$

where $w_1, w_2, ..., w_N$ are terms in the query $q$ that appeared in the documents $d$. That is, the score of a document $d$ given a query $q$ is a function $g$ of the accumulated weight $f$ for each matched term.

We have implemented a simple retrieval function called `MyBM25Reimplementation` based on `metapy` in one of the cell in the Jupyter notebook to help get you started.

You can find the formulas for Pivoted Normalization and BM25 in the lecture slides. In the Jupyter notebook, you will need to change the methods in the classes to implement them. Please include your code for this retrieval function in your submission (`.ipynb` file). Each method's implementation is worth 25 points overall.

**Note**: If you want to check whether your retrieval function runs normally, you can test the code in the end of the notebook to illustrate search results for your queries and compare them against the `metapy` BM25 implementation.

# Design Your Own Scoring Function [50 points]

Implement at least one retrieval function *different* from BM25, Dirichlet Prior, and Pivoted Normalization. You will be graded based on your best performing function. You'll get full credit if your retrieval function can beat the provided baseline in the dataset. By "beat," we mean that your implemented function and your choice of parameters should reach higher NDCG@10 than the baseline on Kaggle for our dataset, which you can check at any time. Report this information in your submission: the code to implement the retrieval function, the parameter you used that achieved the best performance, and the best performance. In addition, ***explain*** **what you have explored and why you decide to try those**. You will lose points if you cannot explain why your function can reach a higher performance. You can include your explanations in the end of the submitted notebook.

**Note**: Simply varying the value of parameters in Okapi/BM25, Dirichlet Prior or Pivoted Normalization does not count as a new retrieval function.

# What if I don't do python?

If you program in some other language other than python, you are still welcomed to do this homework in that language. The core part of the homework can be done with a bit of extra effort to built the inverted index and the dataset is sufficiently small that you could even do this manually. If there are existing packages you want to use, you are welcome to build upon them to implement Parts 1 or 2. If you have questions on concerns, please reach out to the instructors ASAP.

# What to submit?

You need to submit three things:

1. Please submit your code in a runnable format to Canvas; .ipynb files are acceptable. If you use a language other than python (which is fine), please include a note on how to run your code.

2. Please submit the rankings for Part 2 to Kaggle. Be sure the join using the link above.

3. Please submit a text (pdf/Word) file that describes your choices and implementation for Part 2. We will need to understand this to make sense of your code. **Be sure to include your Kaggle username in this file** so we can figure out which score is yours.

Everything needs to be submitted to Canvas.

# Late Policy

Throughout the semester, you have three free late days total. These are counted as whole days, so 1 minute past deadline result sin 1 late day used (Canvas tracks this so it's easier/fair). However, if you have known issues (interviews, conference, etc.) let us know at least 24 hours in advance and we can work something out. **Special Covid Times™ Policy**: If you are dealing with Big Life Stuff®, let the instructor know and we'll figure out a path forward (family/health should take priority over this course). Once the late days are used up, the homework cannot be submitted for a second, though speak with the instructor if you think this is actually a possibility before actually not submitting.

# Academic Honesty Policy

Unless otherwise specified in an assignment all submitted work must be your own, original work. Any excerpts, statements, or phrases from the work of others must be clearly identified as a quotation, and a proper citation provided. Any violation of the University's policies on Academic and Professional Integrity may result in serious penalties, which might range from failing an assignment, to failing a course, to being expelled from the program. Violations of academic and professional integrity will be reported to Student Affairs. Consequences impacting assignment or course grades are determined by the faculty instructor; additional sanctions may be imposed.

Please be aware, that we know that many implementations of BM25 and Pivoted Indexing exist out there. We have collected many (many) of these and are able to check your implementation against theirs, as well as against other students. Please do you own work and do not submit code you found online (or anywhere else).