PA3 Search Rank Report

Dhruv Amin
James Webb

System Design

We mostly followed the architecture provided in the skeleton code for PA3. Given a ranking method as a command-line argument, Rank.java would create an instance of either CosineSimilarityScorer.java, BM25Scorer.java, or SmallestWindowScorer.java, each of which extended the AScorer.java abstract class. Document information was maintained as instances of the Document.java class, which included instance variables containing information for the five document fields. Query information, namely the words in the query string, were maintained in the Query.java class. We implemented a LoadHandler.java class that used a simplified BSBI to generate idf scores for each document in our corpus. For each of the scorer tasks, document vectors were maintained as mappings from Strings (query term) to doubles (score for that term in the doc). These mappings were populated and normalized for each of the different tasks, and would finally be used to return a net score for the instance of the scorer class.

Parameter Values

Task 1 Cosine

task1_W_url : -1.5	task1_W_title : -4.4
task1_W_body : -1	task1_W_header : -1.1
task1_W_anchor : -10.5	

Explanation: For cosine scoring, the different fields are given different weights when calculating the net impact of a field in a document. We reasoned that some fields are more telling than others of the relevance of a certain document to a query. We generated some hypotheses and empirically tuned the parameters to verify these hypotheses.

- URLs often have many erroneous symbols that do not help in determining the relevance of a document. However, if a term match can be made in a URL, for us that seemed to imply that the URL is referring to a document that is very relevant to the matched term.
- Titles often embody the topic of the document, so we hypothesized and verified that if term matches are found in the title, then the document is more likely to be relevant.
- For all parameter tuning across all tasks, we held the body weight constantly at -1. The body text seems to suggest the most basic information about a document, thus we decided we would determine the relative importance of every other field to the body.
- Headers, similar to titles, denote the subject of certain parts of the document. While this is not as useful as the document title in determining relevancy, we found that header term matches are more suggestive of relevance than plain body text.
- Anchors by far were the most telling field when determining relevancy. We hypothesized that if there is a term match with an anchor (and correspondingly a high anchor count for that text), then many pages are saying that the current document is very relevant to those anchor terms. Thus if the anchor terms match the query, the current document is very relevant to the query.

Task 2 BM25

task2_W_url : -1.5	task2_W_title : -4.4
task2_W_body : -1	task2_W_header : -1.1

task2_W_anchor : -10.5	task2_B_url : -0.2
task2_B_title : -0.5	task2_B_body : -1
task2_B_header : -1.1	task2_B_anchor : -20
K_1 : -1.01	λ : -1
λ' : 300	Vj : page_rank / (λ' + page_rank)

Explanation: For comparison purposes, we held our field weights the same as they were in the cosine scorer task. This way, we could see how much difference can be made using the BM25-specific parameters. Also, the same intuition for field weighting applies in the BM25 case, so the same arguments as presented in cosine can be made. As for the B-weights, there seemed to be an inverse correlation of field significance to weighting, which makes sense as B-weights are in the denominator of the score function for BM25. Thus, the more important a field, the lower magnitude the B-weight should be.

- Contrary to our hypothesis, the title field empirically performed better when it was weighted less significantly than URLs (although still more so than the body).
- The anchor field is the greatest exception to our intuition about the relation between B-weights and significance. We attributed this to possibly the average length of anchor texts often being much larger than that for other fields (anchor count * anchor text), which would shrink the denominator for BM25 term frequency score.
- We hypothesizes that we would want a low K_1 value (in the prescribed 1.1-2 window) in order to allow for early saturation. In other words, we wanted to heavily damped the effect of term frequency in our calculations.
- We empirically tested and determined the our data set worked best with the page_rank / (λ' + page_rank) Vj function. As discussed in lecture, this formula best estimates the 2-Poisson model. λ = -1 was empirically determined to produce the best accuracy.
- In order to make increases in page_rank produce high contributions to Vj, we hypothesized λ' would need to be fairly large. Empirically we determined the best value to be 300.

Task 3 Smallest Window (Cosine)

task3_W_url : -1.5	task3_W_title : -5
task3_W_body : -1	task3_W_header : -5
task3_W_anchor : -30	B : 155

Explanation: The explanations for smallest window cosine follow the same intuitions as Task 1 Cosine, with some minor exceptions (anchor and header weight are even more important). B is a scaling factor for the importance of the small window; it's value was determined empirically.

Report Questions

1. What was the reasoning behind giving the weights to the url, title, body, header and anchor fields for the three tasks? Were there any particular properties about the documents that allowed a higher weight to be given to one field as opposed to another?

The reasoning behind giving different weights to different fields is that finding query terms in different fields have differing amounts of information relevance. We describe the reasons for giving different fields different weights above.

2. What other metrics, not used in this assignment, could be used to get a better scoring function from the document? The metrics could either be static (query-independent, e.g. document length) or dynamic (querydependent, e.g. smallest window).

There are a number of other features & metrics we could have used as inputs to a scoring function in order to determine better relevance:

- The html tags around matched text / formatting could get different weights for more prominent formatting (h1 vs. h2, bold vs. italic vs. normal)
- Keyword order (same order as the query, more relevance)
- # of synonyms to the query terms in the doc (more synonyms, more relevance)
- Query terms in the domain vs. subdomains of a URL
- If a query term is the most frequently used term in the doc
- document length -- longer documents most likely contain more information & are preferable
- Recency of document content publishing
- Grammar & spelling correctness of the document itself (better language is more likely a sign of a better document)
- Where in the document the query terms appear (top vs. middle vs. bottom)
- The amount of duplicate content as an inverse signal

3. In BM25F, in addition to the weights given to the fields, there are 8 other parameters, Burl, Btitle, Bheader, Bbody, Banchor, λ, λ' and K1. How do these parameters affect the ranking function?

The B-weights are used in tandem with the extra information available in BM25, namely the average field lengths and document length. B-weights are given to the ratio of these values, thus B-weights work similarly to the normal W-weights in that they denote the significance of one field over another when determining relevance. However, because B-weights are applied in the denominator of the score function, they are inversely related to their field's significance. λ and λ' are used with the Vj function. Although subject to the choice of Vj, in our case λ' was used as a saturation factor when determining the importance of page_rank to the BM25 field weight. K_1 similarly is used as a saturation factor for determining the contribution of each field weight to the overall BM25 rank. λ scales the impact of Vj and thus the importance of page_rank.

4. In BM25F, why did you select a particular Vj function?

Although in practice log(page_rank) tends to be a good Vj function, our results concluded that for this data set, page_rank / (λ' + page_rank) worked best. We knew we wanted our field-independent parameter, page_rank, to impact the BM25 score, following a non-linear curve with a horizontal asymptote (i.e. for very high page_rank value, the contributions would diminish to nearly zero). However, we did not want the contributions to diminish too rapidly (saturation), so we selected a large λ' value.

5. For a function that includes the smallest window as one component, how does varying B and the boost function change the performance of the ranking algorithm?

Increasing the initial value of B determines the absolute scale to which having a small window is important relative to other signals. The rate of decrease of the ranking function with increased query size determines how important having a smaller window size is relative to longer window sizes. After experimenting with various functions, we found the exponential function 1/x, as described in the handout, was the best rate of decrease