Turn Off Your Cell Phone. Use of any electronic device during the test is prohibited. As previously noted, you are allowed notes: Up to four sheets of 8.5x11 or A4 paper, single-sided (or two sheets double-sided).

Time will be tight. If you spend more than the recommended time on any question, go on to the next one. If you can’t answer it in the recommended time, you are either giving too much detail or the question is material you don’t know well. You can skip one or two parts and still demonstrate what I believe to be an A-level understanding of the material.

Note: It is okay to abbreviate in your answers, as long as the abbreviations are unambiguous and reasonably obvious.

In all cases, it is important that you give some idea of how you derived the answer, not simply give an answer. Setting up the derivation correctly, even if you don’t carry out the calculations to get the final answer, is good for nearly full credit.

1 Query Expansion (7 minutes, 5 points)

Dictionary-based query expansion typically includes synonyms or related concepts to words in the original query. How would you expect this to impact precision and recall? Explain your answer

A. Precision:

This will result in additional documents being retrieved, increasing the denominator in \( \frac{\#relevantretrieved}{\#retrieved} \). If the synonyms are a better match than the original query, the relevant retrieved may go up faster, but in most cases, precision would go down.

Scoring: 1 for decrease, 1 for solid reason why showing understanding of math.

B. Recall:

This will result in additional documents being retrieved, if any of them are relevant then the numerator in \( \frac{\#relevantretrieved}{\#relevant} \) will increase, and recall will increase. (The number of relevant documents is based on the information need, not the query or results of the query, so won’t change.)

Scoring: 1 for increase, 1 for solid reason why showing understanding of math.

What if the word in question had multiple senses (e.g., Java could mean coffee, a programming language, or an island), and you couldn’t determine which word sense was meant? Would this change your answers?

This would likely decrease precision, as it may get synonyms of the wrong word sense.

Scoring: 1 for no or would make precision worse.

2 Politeness (5 minutes, 2 points)

The robots.txt file allows specifying which crawlers are allowed to index which documents. It would seem like a good idea to use it to enforce politeness as well, for example a “frequency: 100” entry could mean that only one crawler index request is allowed every 100 milliseconds. Explain why this would be unlikely to work well (remember that the goal is to keep total web crawler traffic to one request every 100 milliseconds.)

Even if all crawlers respects the robots.txt file and only makes a request every 100ms, requests from multiple crawlers could results in more than 1 request per 100 ms. Addressing this would require coordination between crawlers.

Scoring: 1 for multiple crawlers, 1 for requires coordination or noting that the robots.txt file doesn’t support any sort of coordination / dynamic handling of times.
3 Freshness/Age (5 minutes, 3 points)

Suppose we crawl a web page every two days, and the expected age of the page when we crawl it is 24 hours. If we crawled it every day, how would the expected age change? You do not need to calculate the number, but for full credit, set up the equations.

Need to typeset formula

Scoring: 1 for decrease, 1 for non-zero, 1 for integration. Note that setting up and showing relation between 1-day and 2-day equations captures all of these.

4 Mercator method (5 minutes, 3 points)

The Mercator method assists in both politeness and freshness using front and back queues. Suppose we are using the Mercator method, and we realize that the Purdue CS events web page is not getting crawled frequently enough (the index is out of date). What would we do to correct this within the Mercator method?

The CS events page should be directed to a higher priority front queue.

Scoring: 3 for put this in a different front queue, 2 for some change in priority, 1 for knowing this has to do with the front queue.

5 Web Crawling: Duplicate Elimination (12 minutes, 6 points)

One issue with web crawling is avoiding indexing duplicates or near-duplicates. One method we discussed was using shingles and sketches:

- Shingle: word n-gram from the document, e.g., “One issue with”, “issue with web”, “with web crawling”.
- Sketch: collection of shingles

The sketch method discussed in class allowed us to efficiently estimate the percentage of shingles that overlap between two documents. This just determines if they contain the same shingles (n-grams), not if they are in the same position.

A. How many 3-gram shingles would there be in a document of length \( d \)?

\[ d - 2 \]. Every word is the start of a new shingle, except the last two.

Scoring: 2 for \( d - 2 \), 1 for \( O(d) \)

B. Suppose I have documents \( D_1 \) and \( D_2 \) that match on 90% of 1-gram shingles (90% of the words are the same), and \( D_1 \) and \( D_3 \) have 90% of 3-gram shingles matching. Which do you think is closer to a duplicate, \( D_1 \) and \( D_2 \), or \( D_1 \) and \( D_3 \)? Explain your reasoning. Hint: Think what would happen if I had \( k \)-grams, where \( k \) were close to the length of a document.

\( D_3 \) would be closer to a duplicate, as it is not only the same words, but a lot are in the same order. One way to see this is to imagine a \( k \)-gram, where \( k \) is the length of the document - matching implies exact duplicate.

Scoring: 1 for \( D_3 \), 1 for some idea that 3-grams capture order.

C. What factors would you want to consider to determine if a near-duplicate detection scheme were effective? (I.e., if you were designing an evaluation metric to compare duplicate detection schemes.)

I’d want to measure false positives, as false positives would result in a failure to index documents. False negatives are worth measuring as well, but they would result in duplicates rather than failing to index documents, so is less critical.

Scoring: 1 for false positive or something comparable, 1 for false negative or something comparable.
6  Pagerank (11 minutes, 8 points)

Given the graph at right, answer the following questions on PageRank:

A. Pagerank can be expressed as a vector $r$. Assuming A is the top entry in the vector, and D is the bottom, give the equation that would hold if you knew $r$. $r$ can be left as an unknown variable, but but you should show values for the rest of the equation. If there are any other unknowns, please explain what you would need to specify them.

$$M = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & .5 & .5 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$r = B^T r$$

Scoring: 1 having the form $r = B \times r$ or noting it is the eigenvector of a matrix $B$, 1 for setting up $B$ with links, 1 for correctly representing as row of incoming links (row/column done right), 1 for normalizing columns to 1, 1 for correctly including/explaining damping, maximum 4 points of 5 possible.

B. Which document has the lowest Pagerank? Explain how you came up with this answer (there are multiple ways.)

Pagerank of parents is split among children, so B will end up with lower pagerank than others.

Scoring: 1 for B, 1 for explanation.

C. Which document has the highest Pagerank? Again, explain how you came up with this answer.

The graph is largely symmetrical, but as C has two parents, and as such will accumulate pagerank from both sources, it is likely to be higher.

Scoring: 1 for C, 1 for explanation. 1 for A or D with explanation.

7  Using PageRank (6 minutes, 3 points)

You are given a query “Information Retrieval jobs”, and asked to order documents using PageRank and give the top 20 results for the query. Which of these three approaches is the most appropriate? For the inappropriate approaches, briefly explain why they are not the right way to use PageRank.

A. Calculate the Pagerank of each document using the entire corpus, sort by Pagerank, and return the highest 20 documents.

This won’t work, as it doesn’t take the query into account.

B. Calculate the Pagerank of each document using the entire corpus, run the query “Information Retrieval jobs” using a boolean retrieval model, sort the returned documents by their Pagerank, and return the highest 20 documents.

This would generally be the most appropriate approach.
C. Run the query “Information Retrieval jobs” using a boolean retrieval model, build a graph using only the documents returned by the query, calculate the Pagerank using that graph, sort the documents by the Pagerank, and return the highest 20 documents.

This is likely to result in a graph with few links (since other documents pointing to, or pointed to by, the information retrieval documents may not be in the graph.) Pagerank won’t work well with a disconnected graph. Furthermore, this requires rerunning pagerank each query, the other methods are more efficient as pagerank only needs to be run once.

Scoring: 1 for yes to second, 1 for reason for others.

8 Relevance Feedback (4 minutes, 2 points)

The Rocchio formula for pseudo-relevance feedback is nice in theory, but if used with TF*IDF and cosine similarity, it would lead to big increases in computational cost if used as is. Explain why.

The Rocchio formula results in a query that contains all the terms in the relevant and irrelevant documents, a much greater number than the original query. This results in a big increase in index lookups - from $O(|q|)$ to $O(|d|)$. Many noted that it requires computing everything twice. While true, this is much smaller than the impact of the increase in query length.

Scoring: 1 for increase in terms, 1 for index lookups.

9 Text Categorization (8 minutes, 6 points)

Typical text categorization approaches place each document into a single category. As we have seen in Project 2, sometimes documents really do fit multiple categories. Assume you have a very good text categorization scheme that could do either. Explain what would happen to precision and recall if you went from placing each document in a single category to possibly placing it in multiple categories. State both what change you would expect, and why.

A. Precision would:

   Precision would probably see little change, or go down. If a document belonged in category A and B, putting it in only one would reduce both the relevant retrieved and total retrieved for the other category.

B. Recall would:

   Recall would improve, as it would be possible to place a document that belonged in multiple categories in all of them, increasing the number of relevant retrieved in each of the categories it belongs to.

C. Name a third measure for evaluating text categorization, and how you would expect it to change.

   Many choices; accuracy is one. If placing an article in a single category is viewed as a false drop from the other categories it belongs in, then false negatives would be significantly reduced by having multiple clusters, resulting in an increase in accuracy.

Scoring: 1 for Precision unchanged or goes down, 1 for recall improves, 1 for naming another measure, 1 each for explanation showing understanding of mathematics of measure.

10 Text Categorization (4 minutes, 2 points)

K-Nearest Neighbor classification simply finds the closest points based on a similarity metric, it doesn’t really learn how to weight different words. Surprisingly, when we use k-NN with TF*IDF and cosine similarity as the similarity metric, it performs comparably to methods that do learn how to weight words differently (e.g., Naïve Bayes).
Why do you think it works well? Is it because all words are equally important for text categorization? Is $k$-NN really learning to weight words differently? Or some other reason? Explain briefly (one to two sentences should be enough.)

TF*IDF weights words differently, and is apparently quite effective at choosing weights that are appropriate for text categorization.

Scoring: 1 for TF*IDF weighting, 1 for idea that weights are different in way that works.

11 Text Clustering (4 minutes, 2 points)

Partitional clustering methods, such as $k$-means, requires that every document be placed in one and only one cluster. Give an example of why this might work poorly for text clustering and explain why.

The key challenge is that documents may be about multiple topics, for example an article on sports injuries could belong in both healthcare and sports topics. This makes it hard to create appropriate clusters, as the distinction between the healthcare and sports clusters becomes blurred. Ambiguous words (e.g., java) can be one source of this issue, but really isn’t as good an example, as this is really a failure of word sense disambiguation rather than the clustering.

Scoring: 1 for any reasonable example, 1 for explanation matching example.