CS47300 Fall 2019 Midterm 1, October 15, 2019
Prof. Chris Clifton

Turn Off Your Cell Phone. Use of any electronic device during the test is prohibited. As previously noted, you are allowed notes: Up to two sheets of 8.5x11 or A4 paper, single-sided (or one sheet 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 Stemming (5 minutes, 3 points)

<table>
<thead>
<tr>
<th>Word</th>
<th>Stemmed word</th>
</tr>
</thead>
<tbody>
<tr>
<td>community</td>
<td>commun</td>
</tr>
<tr>
<td>group</td>
<td>group</td>
</tr>
<tr>
<td>living</td>
<td>live</td>
</tr>
<tr>
<td>things</td>
<td>thing</td>
</tr>
<tr>
<td>commonality</td>
<td>common</td>
</tr>
<tr>
<td>norms</td>
<td>norm</td>
</tr>
<tr>
<td>values</td>
<td>valu</td>
</tr>
<tr>
<td>customs</td>
<td>custom</td>
</tr>
<tr>
<td>communities</td>
<td>commun</td>
</tr>
<tr>
<td>share</td>
<td>share</td>
</tr>
<tr>
<td>sense</td>
<td>sens</td>
</tr>
<tr>
<td>place</td>
<td>place</td>
</tr>
<tr>
<td>country</td>
<td>countri</td>
</tr>
<tr>
<td>village</td>
<td>villag</td>
</tr>
<tr>
<td>virtual</td>
<td>virtual</td>
</tr>
<tr>
<td>communication</td>
<td>commun</td>
</tr>
<tr>
<td>platforms</td>
<td>platform</td>
</tr>
</tbody>
</table>

Given words and their stems at right, for the query: real time communication networks explain how stemming could hurt precision. For full credit your explanation should be in terms of the mathematical definition of precision.

Precision is #relevant retrieved / #retrieved. Stemming will lead to documents containing community and communities being retrieved (would “real time community networks” be where we get flash mobs?), which would increase the number retrieved, but probably not relevant retrieved. The increase in denominator while the numerator stays the same would cause precision to decrease.

Scoring: 1 for understanding multiple words go to same stem, 1 for showing understanding of precision, 1 for proof.
2 Boolean Retrieval (5 minutes, 3 points)

Exact Match Retrieval Models are still used in fields such as law and healthcare. Why is exact match necessary in those fields?

There are a lot of reasons. One is a need for completeness, Boolean returns everything that matches. Another is expertise on the part of the users, Boolean allows for complex queries. A third would be unambiguity, with Boolean we know exactly why items are returned. This is by no means a complete list.

Scoring: 1 for an answer showing understanding of boolean retrieval, 1 for reason for need, 1 for why boolean answers this

3 Term Weighting and Cosine Similarity (10 minutes, 10 points)

Given the documents at right, and assuming no further stemming or stopword removal will be done:

D1 to be
D2 or not to be
D3 that is the question
D4 is nobler

A. Give the vector space representation for document D1. Use TF*IDF term weights, with raw term counts for term frequency, and \( \log_2(N/df) \) for inverse document frequency.

Scoring: 1 for including all terms, 1 for term counts, 1 for IDF

B. Complete the index below (fill in the underscored blanks), which uses \( \log_2(tf) + 1 \) for term frequency and \( \log_2(N/df) \) for inverse document frequency.

<table>
<thead>
<tr>
<th>be</th>
<th>D1: 1 ; D2: 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>is</td>
<td></td>
</tr>
<tr>
<td>nobler</td>
<td>D4: 2</td>
</tr>
<tr>
<td>not</td>
<td>D2: 2</td>
</tr>
<tr>
<td>question</td>
<td>D3: 2</td>
</tr>
<tr>
<td>that</td>
<td>D3: 2</td>
</tr>
<tr>
<td>the</td>
<td>D3: 2</td>
</tr>
</tbody>
</table>

Scoring: 1 for D3 and D4, 1 for 1 ; 1 for “to” or “or”, 1 for one (to) or 2 (or)

C. Using the above index, compute the cosine similarity between each document and the query nobler question.
Scoring: 1 for formula, 1 for correct numerator, 1 for correct denominator

4 Latent Semantic Indexing (15 minutes, 8 points)

Given the following $U \times S \times V^T$ singular value decomposition of a Term/Document matrix:

$$
U = \begin{pmatrix}
    \text{human} & 0.22 & -0.11 \\
    \text{interface} & 0.20 & -0.07 \\
    \text{computer} & 0.24 & 0.04 \\
    \text{user} & 0.40 & 0.06 \\
    \text{system} & 0.64 & -0.17 \\
    \text{response} & 0.27 & 0.11 \\
    \text{time} & 0.27 & 0.11 \\
    \text{EPS} & 0.30 & -0.14 \\
    \text{survey} & 0.21 & 0.26 \\
    \text{trees} & 0.01 & 0.49 \\
    \text{graph} & 0.04 & 0.62 \\
    \text{minors} & 0.03 & 0.45
\end{pmatrix},
\quad S = \begin{pmatrix}
    3.3 \\
    2.5
\end{pmatrix},
\quad V^T = \begin{pmatrix}
    D1 & 0.20 & 0.61 & 0.46 & 0.54 & 0.28 & 0.00 & 0.02 & 0.02 & 0.08 \\
    D2 & -0.06 & 0.17 & -0.13 & -0.23 & 0.11 & 0.19 & 0.44 & 0.62 & 0.53
\end{pmatrix}

(I’ll also give you $S^{-1} = \begin{pmatrix}
    0.3 \\
    0.4
\end{pmatrix}$.)

For the query: trees graph

A. Show the score computation for the similarity between document D1 and the query.
   You don’t need to actually perform the arithmetic, as long as you set up the problem properly.

Scoring: 1 for transforming query, 1 for transform using $q^T \times U \times S^{-1}$, 1 for cosine similarity, 1 for similarity with D1 entry in $V^T$

B. Give the top 3 documents and their ranking. You can do this by calculating the scores
   (and if you do, list them), but if you can determine the order without calculating the scores,
   please explain your reasoning.

Computed as described above - this is a bit tricky, as my (by hand) calculations gave $D8 \approx .448$, $D6 \approx .445$, and $D7 \approx .445$. The shortcut is noticing
that the query is almost entirely column/concept 2 (which is obvious from looking at the words trees and graph in $U$), so this leads us to having comparatively high values in the second row of $V^T$, and low in the first row. The final thing is remembering that this is cosine similarity, so it is really the ratio of first and second row that matter, not the magnitude (as they are normalized). So $D9$ is not as good a choice, since the transformed query of $[0.015, 0.444]$ has much less weight on the first entry.

Scoring: 1 for understanding it is matching the vector, 1 for noting column 2 / row 2 is most important, 1 for identifying proper documents based on this reasoning, 1 for catching the normalization, not just magnitude.

5 Evaluating ad-hoc information retrieval (10 minutes, 10 points)

There are nine documents and three queries represented below, with relevant documents listed at right. Assume a 0 similarity means the document is not returned, and a higher score means the document is a better match for the query.

<table>
<thead>
<tr>
<th>Document</th>
<th>Q1 similarity</th>
<th>Q2 similarity</th>
<th>Q3 similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Doc1</td>
<td>0</td>
<td>.8</td>
<td>0</td>
</tr>
<tr>
<td>Doc2</td>
<td>.8</td>
<td>0</td>
<td>1.5</td>
</tr>
<tr>
<td>Doc3</td>
<td>1.2</td>
<td>0</td>
<td>1.2</td>
</tr>
<tr>
<td>Doc4</td>
<td>.4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Doc5</td>
<td>.6</td>
<td>.9</td>
<td>.8</td>
</tr>
<tr>
<td>Doc6</td>
<td>.9</td>
<td>.6</td>
<td>0</td>
</tr>
<tr>
<td>Doc7</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Doc8</td>
<td>.4</td>
<td>.6</td>
<td>1.1</td>
</tr>
<tr>
<td>Doc9</td>
<td>.6</td>
<td>.5</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Q1 relevant documents: Doc2 Doc5 Doc6

Q2 relevant documents: Doc1 Doc5 Doc7

Q3 relevant documents: Doc1 Doc2 Doc3 Doc5

Answer the following questions on evaluating the effectiveness of the ad-hoc retrieval system on the given queries.

A. Compute precision for Q3

\[
\frac{\#\text{relevant retrieved}}{\#\text{retrieved}} = \frac{3}{5}
\]

Scoring: 1 for precision formula, 1 for correct use.

B. Compute recall for Q3

\[
\frac{\#\text{relevant retrieved}}{\#\text{relevant}} = \frac{3}{4}
\]

Scoring: 1 for recall formula, 1 correct use.

C. Compute reciprocal rank for Q1

Reciprocal Rank is $1/$rank of first relevant document. Based on the similarity scores, Doc3 is rank 1, Doc6 is rank 2, Doc2 is rank 3, etc. Doc6 is the first relevant document, so Reciprocal Rank $= \frac{1}{2}$.
Scoring: 1 for showing understanding that it measures order, 1 for correct formula or correct use.

D. Name one additional metric for rank (other than reciprocal rank) and compute for Q1.

Several good ones, an easy one is precision@n. Precision@5=\(\frac{3}{5}\), since 3 of the 5 retrieved documents are relevant.

Scoring: 1 for naming a ranking metric, 1 for showing how to use it or using it correctly.

E. Compute mean reciprocal rank for the entire set of queries.

Part C has Q1. The top-ranked documents are relevant for Q2 and Q3, so RR=1/1. The main is 1+1+.5/3 = 5/6.

Scoring: 1 for understanding this measures average across all queries, 1 for understanding how to compute or computing it correctly.

6 Retrieval Models (6 minutes, 4 points)

Given the documents at right, and assuming no stemming has already been done and any stop words have already been removed:

D1 to be
D2 or not to be
D3 that is the question
D4 is nobler

A. For the one word query not, what documents would be returned using TF*IDF and cosine similarity? (You may find it helpful to refer to Question 3, if the answer isn’t obvious.)

TF*IDF and cosine similarity typically assumes 0 in the vector for terms not present, so we only get non-zero if the term is present in both. This is only true for D2.

Scoring: 1 for D2, no explanation needed

B. Of the retrieval model approaches we have discussed, name two that you would expect to return documents other than just D2, and explain why you think it may give additional documents. (If you can only think of one, please list and explain that - it is only one additional point for getting two.)

Latent Semantic Indexing is the obvious answer. Because of the overlap between D1 and D2 (to be), they would likely have a lot over concept overlap, so a query that leads to that concept (which not would, because of the similarity with D2) would also pick up D1.

Pseudo-relevance feedback is an obvious second. Some other reasonable answers were Pagerank, controlled vocabulary mechanisms that might make “not” and other terms similar, or probabilistic mechanisms that treated the
probability of all words not in a query/document as non-zero. BIM is NOT an example of such a mechanism, it only is non-zero for matching terms.

Scoring: 1 for 1st, 1 for explanation, 1 for 2nd. 1 point total for naming and showing understanding of a model, even if it doesn’t address the problem.

7 Probabilistic Retrieval (8 minutes, 4 points)

The binary independence model tends to favor long documents. Explain briefly why this is true. For full credit, this should relate to why this is true in terms of the mathematics of the model.

BIM computes the $c_i$ independent of the particular document we are comparing with (they are based on the corpus, and possibly query). So the only difference is what terms get included in $\sum_{x_i=q_i=1} c_i$. The more matching terms, the more $c_i$ are added in, and longer documents have a higher chance of having matching terms.

Scoring: 1 for based purely on presence/absence, so longer documents likely to have more words. 1 for some reasonable mathematical basis.

One solution to this is to normalize by the length of the documents, so that terms in a long document are given less weight than terms in a short document. With TF*IDF using cosine similarity, such normalization is unnecessary. Explain why. (Again, for full credit the explanation should be in terms of the mathematics behind the method.)

The key is that cosine similarity measures the angle between vectors, so length doesn’t matter. Note that taking the logarithm of term frequency reduces the impact of having many occurrences of a term, but the impact is still there. And logarithm doesn’t help with having many different terms, which results in more matching terms and a larger dot product with the query. It is the division by $||d||$ that really prevents larger documents from having an advantage.

Scoring: 1 for cosine similarity already normalizes, 1 for some reasoning why (angle, or numerator)

8 Stemming (12 minutes, 4 points)

Assume a boolean retrieval mechanism that uses only conjunction (and), i.e., all terms in the query must appear in the document for it to be a match. Prove that stemming could improve, but could not hurt, recall.

Recall is $\frac{\text{Relevant Retrieved}}{\text{Relevant}}$. Relevant does not change, so the question reduces to Can stemming reduce the number of relevant documents retrieved? Assume the query contains the term $A$, then every retrieved document must contain $A$. If the term $A$ stems to $A'$, then every retrieved document must contain the stemmed term $A'$. But since every original retrieved document contains $A$, each of those when stemmed will contain $A'$. Thus every document retrieved originally will be retrieved after stemming. Thus stemming could not hurt recall.
If there is a relevant document containing a term $A_2$ but not $A$, then it would not be in the relevant retrieved documents without stemming. But if $A_2$ also stems to $A'$, the document would be in the relevant retrieved set, so it is possible that stemming could improve recall.

Scoring: 1 for mathematical understanding of recall, 1 for idea of showing does not hurt, 1 for idea of showing it can help, 1 for reasonably solid proof.