Problem 1 (Result diversity).

Consider the query "hosts of Academy Awards" with its unranked results in Table 1.

(a) Assuming relevancy (all results answer the query correctly), rank by popularity of results, which we define as the Σ of popularity of proper nouns per result.

Show top-10 results.

(b) Are the top-5 results diverse enough? What about the top-3 results? What else can we call a non-diverse set of results?

(c) To ensure diversity, apply MMR (Maximal Marginal Relevance) to re-rank the results.

i Compute the relevance of each ranked result, from part a), as follows.
\[ \text{sim}(q, d_i) = (1 - \frac{\text{position of } d_i}{|d|}) \]
- so the result at position 1 will get a relevance score of 0.9.

ii \( \lambda = 0.5 \)

iii Measure the similarity between two documents (results) as the size of the intersection of their terms divided by the size of the union of their terms - disregard stop word(s).

Show the top-5 re-ranked results. Are the top-5 results diverse enough? top-3 results?

Note: you must include all calculation details in your answers.

(d) What does it mean to set \( \lambda \) to 0.5? \( \lambda = 1? \lambda = 0? \)

Solution.

(a) Table 2.

(b) no and no. Homogeneous result set/ set with duplicates/ near-duplicates results (any of these).

(c) Table 3 shows the results ranked by relevancy \( \text{sim}(q, d_i) \).

The similarity computations between documents:

\[ \text{sim}(d_1, d_2) = 0.6, \text{sim}(d_1, d_3) = 0.14, \text{sim}(d_1, d_4) = 0.11, \text{sim}(d_1, d_5) = 0.14, \text{sim}(d_1, d_6) = 0.14, \text{sim}(d_1, d_7) = 0.14, \text{sim}(d_1, d_8) = 0.14, \text{sim}(d_1, d_9) = 0.14, \text{sim}(d_1, d_{10}) = 0.14, \text{sim}(d_2, d_3) = 0.14, \text{sim}(d_2, d_4) = 0.11, \text{sim}(d_2, d_5) = 0.14, \text{sim}(d_2, d_6) = 0.14, \]

\[ \text{sim}(d_3, d_4) = 0.14, \text{sim}(d_3, d_5) = 0.14, \text{sim}(d_3, d_6) = 0.14, \text{sim}(d_3, d_7) = 0.14, \text{sim}(d_3, d_8) = 0.14, \text{sim}(d_3, d_9) = 0.14, \text{sim}(d_3, d_{10}) = 0.14, \text{sim}(d_4, d_5) = 0.14, \text{sim}(d_4, d_6) = 0.14, \text{sim}(d_4, d_{10}) = 0.14, \text{sim}(d_5, d_6) = 0.14, \text{sim}(d_5, d_{10}) = 0.14, \text{sim}(d_6, d_{10}) = 0.14, \text{sim}(d_7, d_{10}) = 0.14, \text{sim}(d_8, d_{10}) = 0.14, \text{sim}(d_9, d_{10}) = 0.14.\]

1 Popularity = number of hits when submitting the proper noun as a query to Google. Example: score("Bill Gates was born in Seattle") = hits(Bill Gates) + hits(Seattle) = 1,479 million

2 Initialize the diversified results by adding the result with the highest popularity score.
Step 1: we move the result with the highest relevance score to the final set (re-ranked/diversified).

1 Chris Rock hosted in 2005. \([d_3]\)

Step 2: We now need to re-score the 9 results left using the equation in Lecture 18, slide 8.

\[
\begin{align*}
score(d_1) &= 0.5 \times 0.2 - 0.5 \times 0.14 = 0.03, \\
score(d_2) &= 0.5 \times 0.1 - 0.5 \times 0.14 = -0.02, \\
score(d_4) &= 0.5 \times 0.1 - 0.5 \times 0.11 = 0.145, \\
score(d_5) &= 0.5 \times 0.3 - 0.5 \times 0.14 = 0.08, \\
score(d_6) &= 0.5 \times 0.8 - 0.5 \times 0.6 = 0.1, \\
score(d_7) &= 0.5 \times 0.7 - 0.5 \times 0.14 = 0.28, \\
score(d_8) &= 0.5 \times 0.6 - 0.5 \times 0.14 = 0.23, \\
score(d_9) &= 0.5 \times 0.5 - 0.5 \times 0.14 = 0.18, \\
score(d_{10}) &= 0.5 \times 0.5 - 0.5 \times 0.14 = 0.07.
\end{align*}
\]

Move the one with the maximum score to the final set.

1 Chris Rock hosted in 2005. \([d_3]\)
2 Steve Martin hosted in 2001 \([d_5]\)

Step 3: We again re-score the 8 results left (now that we have 2 results in the final set to compare with novelty-wise) instead of 1).

\[
\begin{align*}
score(d_1) &= 0.5 \times 0.2 - 0.5 \times 0.14 = 0.03, \\
score(d_2) &= 0.5 \times 0.1 - 0.5 \times 0.14 = -0.02, \\
score(d_4) &= 0.5 \times 0.1 - 0.5 \times 0.11 = 0.145, \\
score(d_5) &= 0.5 \times 0.3 - 0.5 \times 0.14 = 0.08, \\
score(d_6) &= 0.5 \times 0.8 - 0.5 \times 0.6 = 0.1, \\
score(d_7) &= 0.5 \times 0.7 - 0.5 \times 0.14 = 0.28, \\
score(d_8) &= 0.5 \times 0.6 - 0.5 \times 0.14 = 0.23, \\
score(d_9) &= 0.5 \times 0.5 - 0.5 \times 0.14 = 0.18, \\
score(d_{10}) &= 0.5 \times 0.5 - 0.5 \times 0.14 = 0.07.
\end{align*}
\]

Move the one with the maximum score to the final set.

1 Chris Rock hosted in 2005. \([d_3]\)
2 Steve Martin hosted in 2001 \([d_5]\)
3 Ann Hathaway and James Franco hosted in 2011 \([d_4]\)

Step 4: again with the 7 results left.

\[
\begin{align*}
score(d_1) &= 0.5 \times 0.2 - 0.5 \times 0.14 = 0.03, \\
score(d_2) &= 0.5 \times 0.1 - 0.5 \times 0.14 = -0.02, \\
score(d_4) &= 0.5 \times 0.1 - 0.5 \times 0.11 = 0.145, \\
score(d_5) &= 0.5 \times 0.3 - 0.5 \times 0.14 = 0.08, \\
score(d_6) &= 0.5 \times 0.8 - 0.5 \times 0.6 = 0.1, \\
score(d_7) &= 0.5 \times 0.7 - 0.5 \times 0.14 = 0.28, \\
score(d_8) &= 0.5 \times 0.6 - 0.5 \times 0.14 = 0.23, \\
score(d_9) &= 0.5 \times 0.5 - 0.5 \times 0.14 = 0.18, \\
score(d_{10}) &= 0.5 \times 0.5 - 0.5 \times 0.14 = 0.07.
\end{align*}
\]

Move the one with the maximum score to the final set.

1 Chris Rock hosted in 2005. \([d_3]\)
2 Steve Martin hosted in 2001 \([d_5]\)
3 Ann Hathaway and James Franco hosted in 2011 \([d_4]\)
4 Chris Rock hosted in 2016 \([d_4]\)

Step 5: again with the 6 results left.

\[
\begin{align*}
score(d_1) &= 0.5 \times 0.2 - 0.5 \times 0.14 = 0.03, \\
score(d_2) &= 0.5 \times 0.1 - 0.5 \times 0.14 = -0.02, \\
score(d_5) &= 0.5 \times 0.3 - 0.5 \times 0.14 = 0.08, \\
score(d_8) &= 0.5 \times 0.6 - 0.5 \times 0.14 = 0, \\
score(d_9) &= 0.5 \times 0.5 - 0.5 \times 0.14 = -0.05, \\
score(d_{10}) &= 0.5 \times 0.5 - 0.5 \times 0.14 = -0.07.
\end{align*}
\]

Move the one with the maximum score to the final set. We stop. Top-5 diversified results are:

1 Chris Rock hosted in 2005. \([d_3]\)
2 Steve Martin hosted in 2001 \([d_5]\)
3 Ann Hathaway and James Franco hosted in 2011 \([d_4]\)
4 Chris Rock hosted in 2016 \([d_4]\)
5 Jon Stewart hosted in 2006 \([d_5]\)

Top-3 and top-5 are now diverse enough.
Table 2: ranked-by-popularity results

<table>
<thead>
<tr>
<th>id</th>
<th>Results</th>
<th>Score(million)</th>
</tr>
</thead>
<tbody>
<tr>
<td>d3</td>
<td>Chris Rock hosted in 2005</td>
<td>868</td>
</tr>
<tr>
<td>d6</td>
<td>Chris Rock hosted in 2016</td>
<td>868</td>
</tr>
<tr>
<td>d7</td>
<td>Steve Martin hosted in 2001</td>
<td>640</td>
</tr>
<tr>
<td>d8</td>
<td>Steve Martin hosted in 2003</td>
<td>640</td>
</tr>
<tr>
<td>d9</td>
<td>Steve Martin hosted in 2010</td>
<td>640</td>
</tr>
<tr>
<td>d4</td>
<td>Ann Hathaway and James Franco hosted in 2011</td>
<td>500</td>
</tr>
<tr>
<td>d5</td>
<td>Jon Stewart hosted in 2006</td>
<td>112</td>
</tr>
<tr>
<td>d1</td>
<td>Jimmy Kimmel hosted in 2017</td>
<td>84</td>
</tr>
<tr>
<td>d2</td>
<td>Jimmy Kimmel hosted in 2018</td>
<td>84</td>
</tr>
<tr>
<td>d10</td>
<td>Alan Alda hosted in 1986</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 3: ranked results - by relevance

<table>
<thead>
<tr>
<th>id</th>
<th>Results</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>d3</td>
<td>Chris Rock hosted in 2005</td>
<td>0.9</td>
</tr>
<tr>
<td>d6</td>
<td>Chris Rock hosted in 2016</td>
<td>0.8</td>
</tr>
<tr>
<td>d7</td>
<td>Steve Martin hosted in 2001</td>
<td>0.7</td>
</tr>
<tr>
<td>d8</td>
<td>Steve Martin hosted in 2003</td>
<td>0.6</td>
</tr>
<tr>
<td>d9</td>
<td>Steve Martin hosted in 2010</td>
<td>0.5</td>
</tr>
<tr>
<td>d4</td>
<td>Ann Hathaway and James Franco hosted in 2011</td>
<td>0.4</td>
</tr>
<tr>
<td>d5</td>
<td>Jon Stewart hosted in 2006</td>
<td>0.3</td>
</tr>
<tr>
<td>d1</td>
<td>Jimmy Kimmel hosted in 2017</td>
<td>0.2</td>
</tr>
<tr>
<td>d2</td>
<td>Jimmy Kimmel hosted in 2018</td>
<td>0.1</td>
</tr>
<tr>
<td>d10</td>
<td>Alan Alda hosted in 1986</td>
<td>0.0</td>
</tr>
</tbody>
</table>

(d) \( \lambda \) to 0.5: 50-50 importance to relevance and novelty. \( \lambda = 1 \): we only rank by relevance. \( \lambda=0 \): only rank by novelty.

**Problem 2** (Indexing and query processing).

(a) Create an inverted index for the corpus in Table 4. Include documents ids and frequencies(no need for positions).

(b) Process the following 2 queries using TAAT(Term-at-a-time) and return top-1 result:
   Q1: apple stores in paris.
   Q2: apple songs.
   i Are both answers relevant to the queries? What’s the problem?
   ii Briefly, propose a solution.

**Solution.**

1) Disregard stop words. ii) Stem words using the Porter Stemmer: [https://text-processing.com/demo/stem/](https://text-processing.com/demo/stem/)
With an apple I will astonish Paris.

I am still a very big fan of Fiona Apple.

I love making apple strudel.

Apple has done a great job on the iPod.

<table>
<thead>
<tr>
<th>document id</th>
<th>text</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_1$</td>
<td>.....With an apple I will astonish Paris....</td>
</tr>
<tr>
<td>$d_2$</td>
<td>...I am still a very big fan of Fiona Apple....</td>
</tr>
<tr>
<td>$d_3$</td>
<td>....I love making apple strudel....</td>
</tr>
<tr>
<td>$d_4$</td>
<td>....Apple has done a great job on the IPod....</td>
</tr>
</tbody>
</table>

Table 4

(a) appl — d1,1 d2,1 d3,1 d4,1
  astonish — d1,1
  big — d2,1
  done — d4,1
  fan — d2,1
  fiona — d2,1
  great — d4,1
  ipod — d4,1
  job — d4,1
  love — d3,1
  make — d3,1
  pari — d1,1
  strudel — d3,1

(b) Q1: apple stores in paris.
Top-1 result is $d_1$.
Accumulators: d1:2,d2:1,d3:1,d4:1.
Q2: apple songs.
Top-1 is any document picked randomly (to break ties).
Accumulators: d1:1,d2:1,d3:1,d4:1.

i. Ambiguity, 'apple' according to this corpus can mean the company, the painting, a singer, or a fruit.
ii. Any of these are acceptable: increase k to allow more aspects; diversifying results to allow novelty; query expansion.

**Problem 3** (Compression).
Compute variable-byte codes for the postings lists: 200, 5630, 18000, 22500.
Use gaps instead of doc ids where possible. Write binary in 8-bit blocks.

**Solution.** Gaps encoding: 200, 5430, 12370, 4500.
Variable-byte code: 00000001 11001000 00101010 10110110 01100000 11010010 00100011 10010100