1 Information Retrieval

1.1 TF-IDF

Given is the following document collection containing two documents:

\( D_1: \) Game of Thrones is an American fantasy drama television series.
\( D_2: \) Game of Thrones has an ensemble cast estimated to be the largest on television.

1. Create an inverted index for the following document collection. Include TF and DF values at a suitable position in the index.

- **Tokenization rules**: word wise, case-folding, ignore punctuation.
- **Stop list**: an, has, is, of, on, the, to.

2. Which search results can be obtained from this index for the following queries?

\( Q_1: \) Game Thrones
\( Q_2: \) Television Series

Compute the relevance scores for each query and search result using the following formula:

\[
W_{Q,d} = \sum_{q \in Q} \log(1 + TF_{q,d}) \cdot IDF_q
\]

Explain the results!

1.2 Precision and Recall

A collection of documents contains 10 documents that are relevant for the query \( q \). For this query, the search engines \( S_1 \) and \( S_2 \) return the following relevant (R) and non-relevant (N) documents:

\( S_1: \) NNRRR  RNRNR
\( S_2: \) RRRNN  NRRRN

Draw a precision-recall diagram for the both search results and compare the quality of the search results based on the interpolated precision at 45% recall.

<table>
<thead>
<tr>
<th>Recall(%)</th>
<th>P(( S_1 ))</th>
<th>P(( S_2 ))</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sergej Wildemann - wildemann@L3S.de
2 Query Optimization and Tolerant Retrieval

1. A document collection with 50,000 documents contains weather forecasts. Given is the following query:

\[(\text{snow}) \text{ AND } (\text{sun OR wind}) \text{ AND NOT } (\text{rain OR thunderstorm})\]

Specify the most efficient order of execution for this query that can be determined from the following table:

<table>
<thead>
<tr>
<th>Term</th>
<th>DF</th>
</tr>
</thead>
<tbody>
<tr>
<td>snow</td>
<td>25,000</td>
</tr>
<tr>
<td>sun</td>
<td>15,000</td>
</tr>
<tr>
<td>wind</td>
<td>3,000</td>
</tr>
<tr>
<td>rain</td>
<td>20,000</td>
</tr>
<tr>
<td>thunderstorm</td>
<td>10,000</td>
</tr>
</tbody>
</table>

Describe a possible term distribution for which the order you proposed is not optimal.

2. Given is a wildcard query “w*a*n*t*e*r* f* e*l” (winterfell).
   a) Describe a trigram index structure.
   b) For this query, create queries for a trigram index and a permuterm index.

3. Compute the Levenshtein distance and the bigram based similarity between the terms “storm” and “swords”.

4. Explain the exact edit operations that are needed to transform “swords” into “storm” and how they can be derived from the Levenshtein distance calculation table.

3 Text Classification and Clustering

1. The figure below shows a state of the k-means algorithm with k=3. The squares represent centroids and circles represent the data points. The color encoding corresponds to the current cluster assignment.

   a) What phase of the algorithm has just finished and what phase is going to follow next?
   b) Sketch the changes that will be performed by the k-means algorithm in the next step.
2. Given is a model of a Naive-Bayes-Classifier with two classes C1 and C2:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(C1)</td>
<td>0.4</td>
</tr>
<tr>
<td>P(C2)</td>
<td>0.5</td>
</tr>
<tr>
<td>P(game</td>
<td>C1)</td>
</tr>
<tr>
<td>P(game</td>
<td>C2)</td>
</tr>
<tr>
<td>P(t</td>
<td>C1), t ≠ game</td>
</tr>
<tr>
<td>P(t</td>
<td>C2), t ≠ game</td>
</tr>
</tbody>
</table>

Classify the following document using this classification model:

*Game of Thrones has been the most pirated TV series since 2012.*

4 **Link Analysis with PageRank**

Given is the PageRank formula:

$$\tilde{x}_{k+1} = (1 - c)\tilde{x}_k A + \frac{c}{N^\top e}$$

and the following graph:
A

B

D

C

Figure 2: Web graph

1. Create the link matrix $A'$ with teleportation for this graph. Use the teleportation probability of 10%.

2. $\vec{c}$ is $\vec{1}$. In $\vec{x}_0$ all random surfers are on node C. Compute the vector $\vec{x}$ for the first four iterations of the PageRank formula ($k = 0, 1, 2, 3$) for this graph. Round to 2 decimal places!