D = [2 0 5]          Binary form          D = [1 0 1]
1 2 0               keep terms that appears
1 3 2               at least twice

For words, we may use **normalization** rather than using words as they are use their stems.

- Stem vs. Lemma

**Word**

Better → Stemmer (first 5 letters) → Bette

Lemmatizer → Good

**Inverted File**

<table>
<thead>
<tr>
<th>talent</th>
<th>is</th>
<th>overrated</th>
<th>else</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>d1</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Main memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>posting list</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>d_no</th>
<th>Term weight</th>
</tr>
</thead>
</table>
Retrieval

Boolean information
(AMD Retrieval)

Ranking-based uses similarity

Example: Search Engine
Classic

STAIRS: Storage and Information Retrieval System (it is a software by IBM)

<table>
<thead>
<tr>
<th>word</th>
<th>Synonym pointer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>word</th>
<th>Synonym pointer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No of docs</th>
<th>No of occurrences</th>
<th>Occ 1</th>
<th>Occ 2</th>
<th>............</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Doc. no</th>
<th>Paragraph no</th>
<th>Sentence no</th>
<th>Word no</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Privacy code  Formatted field  Paragraph 1 header  Text  Paragraph 2 header  Text

<table>
<thead>
<tr>
<th>Author name</th>
<th>Journal no</th>
<th>Page no</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

STAIRS has 2 programs

1. Utility for database creation and maintenance
2. Query and retrieval systems (AQUARIUS)

Modes of Operations

1. SEARCH mode: for textual retrieval
2. SELECT mode: for structured information retrieval
Search example:

HEART

HEART or DISEASE

HEARTS

DISEASE$3

HEART and DISEASE: in the same sentence

HEART SAME DISEASE: in the same paragraph

**Ranking**

After doing SEARCH how to rank?

a: The frequency of the term in the document

b: The frequency of the term in the retrieval set

c: The number of documents in the retrieval set in which the term appears

Value of a term$^f = \frac{a \times b}{c}$

Score of a document = $\sum$ value of all query terms which appear in this document

**Example:**

a: 16 appears a times in this document

b: 1,247 appears b times in all of the retrieved docs

c: 152 appears c many of distinct documents

value = $\frac{(16 \times 1,247)}{152} = 131.26$
**Similarity Calculation**

Vector Space Model

Q **c**

Calculate similarity and rank

D **c**

<table>
<thead>
<tr>
<th>Measure</th>
<th>Binary</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inner Product</td>
<td>![Math Equation](<a href="https://latex.codecogs.com/svg.latex?">https://latex.codecogs.com/svg.latex?</a></td>
<td>X \cap Y</td>
</tr>
<tr>
<td>Dice Coefficient</td>
<td>![Math Equation](<a href="https://latex.codecogs.com/svg.latex?%5Cfrac%7B2">https://latex.codecogs.com/svg.latex?\frac{2</a></td>
<td>X \cap Y</td>
</tr>
<tr>
<td>Cosine Coefficient</td>
<td>![Math Equation](<a href="https://latex.codecogs.com/svg.latex?%5Cfrac%7B">https://latex.codecogs.com/svg.latex?\frac{</a></td>
<td>X \cap Y</td>
</tr>
<tr>
<td>Jaccard Coefficient</td>
<td>![Math Equation](<a href="https://latex.codecogs.com/svg.latex?%5Cfrac%7B">https://latex.codecogs.com/svg.latex?\frac{</a></td>
<td>X \cap Y</td>
</tr>
</tbody>
</table>

**Example:**

X = (1 0 1 1 1)  | |x| = 4

Y = (1 1 0 1 0)  | |y| = 3

Inner Product = 2

Dice Coefficient = (2 x 2) / (4 + 3) = 4 / 7

Cosine Coefficient = 2 / (4^(1/2) x 3^(1/2))
X = (2 0 1 3 2)  
Y = (1 0 2 1 5)  

Dice Coefficient = \( \frac{2(2x1 + 1x2 + 3x1 + 2x5)}{4+1+9+4} + \frac{(1+4+1+25)}{18.31} \approx 0.69 \)

Cosine Coefficient = \( \frac{2(2x1 + 1x2 + 3x1 + 2x5)}{(18.31)^{1/2}} = 0.72 \)

Construction of Similarity (Proximity) Matrices

\[
D = \begin{bmatrix}
d_{11} & 0 & 0 & 1 & 0 \\
d_{21} & 0 & 1 & 1 & 0 \\
0 & 0 & d_{30} & 0 & 1 \\
0 & 0 & d_{40} & 0 & 1 \\
0 & 0 & d_{50} & 0 & 1
\end{bmatrix}
\]

\[
S = \begin{bmatrix}
1.00 & S_{12} & S_{13} & S_{14} & S_{15} & 4 \\
X & 1.00 & S_{23} & S_{24} & S_{25} & 3 \\
X & X & 1.00 & S_{34} & S_{35} & 2 \\
X & XX & 1.00 & S_{45} & 1 \\
X & XXX & 1.00 & & &
\end{bmatrix}
\]

\[S_{ij} = S_{ji}\]

\[4 + 3 + 2 + 1 = 10\]

\[(N-1) + (N-2) + \ldots + 2 + 1 = \sum_{i=1}^{n} i\]

n (n-1) / 2 \(\rightarrow\) \(O(n^2)\)

Similarity Calculation Methods

1. **Brute Force**: Calculate every similarity value

   \[m = \text{number of documents}\]

   \[\text{for } i = 1 \text{ to } m-1\]
   \[\text{for } j = i + 1 \text{ to } m\]
   \[\text{find } S_{ij}\]
   \[\text{end for}\]
   \[\text{end for}\]
2. Using the knowledge of term distributions in documents.

2 documents can have a similarity value > 0 if they have one or more common terms.

\[ t_1 \rightarrow d_1, d_2 \]
\[ t_2 \rightarrow d_1, d_2 \]
\[ t_3 \rightarrow d_4, d_5 \]
\[ t_4 \rightarrow d_2, d_5 \]
\[ t_5 \rightarrow d_1, d_2 \]
\[ t_6 \rightarrow d_3, d_4, d_5 \]

Consider \( d_1 \): contains \( t_1, t_2 \) and \( t_5 \)

\[ d_1, d_2 \cup d_1, d_2 = d_1, d_2 \]

Computed \( S_{12} \)

Consider \( d_2 \): contains \( t_1, t_2, t_4 \) and \( t_5 \)

\[ d_1, d_2 \cup d_1, d_2 \cup d_2, d_5 = d_1, d_2, d_5 \]

Computed \( S_{25} \)

Consider \( d_3 \): contains \( t_6 \)

\[ d_3, d_4, d_5 \]

Computed \( S_{34}, S_{35} \)

Consider \( d_4 \): \( S_{45} \)

Consider \( d_5 \): anything

Total number of similar items we calculate = 5
There is 50% saving

The knowledge of terms distribution is good, however we still have some unnecessary computations during term matching.
Consider finding number of common entries between two vectors.

\[ x_{\text{lng}} = \text{number of elements in } x \]

\[ X: \begin{bmatrix} 5 & 7 & 8 \end{bmatrix} \]

\[ Y: \begin{bmatrix} 5 & 7 & 8 \end{bmatrix} \]

\[ x_{\text{ndx}} \leftarrow \]

\[ y_{\text{ndx}} \leftarrow \]

\[ \text{sum} = 0 \]

while (\( x_{\text{ndx}} \leqslant x_{\text{lng}} \) and \( y_{\text{ndx}} \leqslant y_{\text{lng}} \))

\[ x(x_{\text{ndx}}) = y(y_{\text{ndx}}); \text{sum} = \text{sum} + 1; \]

\[ x_{\text{ndx}} ++; \]

\[ y_{\text{ndx}} ++; \]

\[ x(x_{\text{ndx}}) < y(y_{\text{ndx}}); x_{\text{ndx}} ++ \]

3. Using an inverted index file for terms

\[ t_1 \rightarrow 1,1<2,1> \]
\[ t_2 \rightarrow 1,1<2,1> \]
\[ t_3 \rightarrow 4,1<5,1> \]
\[ t_4 \rightarrow 2,1<5,1> \]
\[ t_5 \rightarrow 1,1<2,1> \]
\[ t_6 \rightarrow 3,1<4,1><5,1> \]

posting list
Document Length Info

<table>
<thead>
<tr>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>d4</th>
<th>d5</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

Consider d1: contains t1, t2 and t5

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Due to t1

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>t2</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>3</th>
<th>2</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>t5</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Final Similarity Value  

<table>
<thead>
<tr>
<th></th>
<th>(2 x 3) / (3+4)</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
</table>

\[ \text{xd} = \text{depth of indexing average number of terms / documents} \]

\[ 3 + 4 + 1 + 2 + 3 / 5 = t, \text{ where } 5 = m \]

\[ \text{tg} = \text{term generality average number of documents / term} = t / n, \text{ where } n = 6 \]

\[ \text{xd .tg} + \text{xd .tg} + \ldots + \text{xd .tg} \text{ (except last one)} \]

\[ (m-1) \cdot \text{xd .tg} = m \cdot \text{xd .tg} \text{ (since m is a large number)} \]
S = 

\[
\begin{pmatrix}
1.00 & S_{12} & S_{13} & S_{14} & S_{15} \\
X & 1.00 & S_{23} & S_{24} & S_{25} \\
X & X & 1.00 & S_{34} & S_{35} \\
X & XX & 1.00 & S_{45} \\
X & XXX & 1.00 & & \\
\end{pmatrix}
\]

During the computations go over the posting list from right to left (from higher document to lower document no.)

\[
x_d \cdot \text{tg} \cdot \frac{(m-1)}{m} + x_d \cdot \text{tg} \cdot \frac{(m-2)}{m} + \ldots + x_d \cdot \text{tg} \cdot \frac{1}{m} = (x_d \cdot \text{tg} / m) \cdot \left( (m-1)+(m-2)+\ldots+1 \right)
\]

\[
= (x_d \cdot \text{tg} / m) \cdot (m \cdot (m-1) / 2)
\]

\[
\sim m \cdot x_d \cdot \text{tg} / 2
\]

Performance Evaluation

**Effectiveness and Efficiency**

**Precision**

time: response time

**Recall**

space

Precision = number of relevant documents / total number of documents retrieved

Recall = number of relevant documents / total number of relevant documents in collection

**Cranfield approach to Information Retrieval Experiments**

We work in a lab environment.

We have a test collection.

a set of documents

a set of queries

relevant documents for each query

Pooling approach for determining relevant documents for each query

**Pooling Approach**

\[ Q \rightarrow IRS1 \quad IRS2 \quad IRS3 \quad \ldots \quad IRS10 \]

**TREC queries**

A few words: ad – hoc

A narrative

\[
F = 2 \times \text{Precision} \times \text{Recall} / \text{Precision} + \text{Recall} \quad \rightarrow \quad \text{harmonic mean of R & P}
\]

This measure can be found in Van Rijsbergen Information Retrieval in the Evaluation part.
User’s Perspective in terms of Evaluation

- Database / collection coverage
- Live links (alive pages)
- Response time
- More recent ones are provided at the beginning (for a football match showing recent results are more important)
- Duplicates are eliminated (near duplicates)
- Personalized results
- Query suggestion / correction (query completion)
- Query diversification
- Context based results (according to region)

<table>
<thead>
<tr>
<th></th>
<th>Relevant</th>
<th>Not Relevant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieved</td>
<td>a (TP)</td>
<td>b (FP)</td>
</tr>
<tr>
<td>Not retrieved</td>
<td>c (FN)</td>
<td>d (TN)</td>
</tr>
</tbody>
</table>

(P) Precision = a / a + b
(R) Recall = a / a + c

TP = True Positive
TN = True Negative
FP = False Positive
FN = False Negative

True means correct decision about its relevance.

Inverted files for text search engines, J. Zobel, A. Moffat, which is on the course web site, is a good source for this topic.
Questions

1. Search engines have some similarities with database systems. However, in some points they are different from each other. What are these main differences? Write 3 of them.

2. To stem and casefold is removing variant ending from words and converting to lowercase. Also, stopping is the process of removing the stop words such as and, in, the, had, etc. According to the below example, apply the casefolding, stemming and stopping methods respectively.

   1. The old night keeper keeps the keep in the town.
   2. In the big old house in the big old gown.
   3. The house in the town had the big old keep.
   4. Where the old night keeper never did sleep.
   5. The night keeper keeps the keep in the night.
   6. And keeps in the dark and sleeps in the light.

   **Fig. 1. The Keeper database. It consists of six one-line documents.**

3. All current search engines use ranking to identify potential answers. In a ranked query, a statistical similarity heuristic or similarity measure is used to assess the closeness of each document to the textual query. A typical older formulation that is effective in practice calculates the cosine of the angle in $n$-dimensional space between a query vector $<wq,t>$ and a document vector $<wd,t>$. Write down these cosine formulas.

4. Suppose that we have an information retrieval system that includes 5 relevant documents, and 10 nonrelevant documents. There are 25 relevant documents in the collection. What is the precision and recall of the system?

Answers

1. 

<table>
<thead>
<tr>
<th><strong>Database Systems</strong></th>
<th><strong>Search Engines</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>To contend with arbitrarily complex queries.</td>
<td>Most of the queries are lists of terms and phrases.</td>
</tr>
<tr>
<td>A match is a record that meets a specified logical condition.</td>
<td>A match is a document that is appropriate to the query according to statistical heuristics and may not even contain all of the query terms.</td>
</tr>
<tr>
<td>They return all matching records.</td>
<td>They return a fixed number of matches which are ranked by their statistical similarity.</td>
</tr>
</tbody>
</table>
2. With casefolding method:
   and big dark did gown had house in keep keeper keeps light
   never night old sleep sleeps the town where
   [2]

   After casefolding, with stemming method:
   and big dark did gown had house in keep light never night old
   sleep the town where
   [2]

   After stemming, with stopping method:
   big dark gown house keep light night old sleep town
   [2]

3. 

\[
\begin{align*}
  w_{q,t} &= \ln \left( 1 + \frac{N}{f_t} \right) & w_{d,t} &= 1 + \ln f_{d,t} \\
  W_d &= \sqrt{\sum_t w_{d,t}^2} & W_q &= \sqrt{\sum_t w_{q,t}^2} \\
  S_{q,d} &= \frac{\sum_t w_{d,t} \cdot w_{q,t}}{W_d \cdot W_q}.
\end{align*}
\]

[2]

---

\(f_{d,t}\): the frequency of term \(t\) in document \(d\)

\(f\): the number of documents containing one or more occurrences of term \(t\)

\(N\): the number of documents in the collection

4. Precision = # relevant items retrieved / # retrieved items = 5/15
   Recall = # relevant items retrieved / # relevant items = 5/25

[1], [2], [3]: Inverted files for text search engines, J. Zobel, A. Moffat.