# TDT4117 Information Retrieval - Autumn 2021

## Assignment 1

## By Thomas Bjerke and Trym Grande

### Task 1: Basic Definitions

**Explain the main differences between:**

**1. Information Retrieval vs Data Retrieval**

- Information retrieval uses search, where the goal is making knowledge accessible to the user (i.e. google.com). Data retrieval systems work in the form of a database management system.
- IR deals with unstructured/semi-structured data (usually text) that satisfies an information need from within large collections (usually stored on computers), while data retrieval is similar, but uses structured data with well-defined semantics.
- IR systems return multiple results with ranking - partial match is allowed here. Querying in data retrieval returns exact results, or no results if no exact match is found.

**2. Structured Data vs Unstructured Data**

- Structured data can be used by the average IT student/employee by querying, while unstructured data can requires the use of a search engine, or the work of a data scinetist to get actual data out.
- Structured data only has a select amount of data types defined, while unstructured can have many, intertwined and cross-referenced.
- Structured data is schema-on-write, meaning that the data is structured when writing (e.g. creating a db table), and will also be when reading. Unstructured data is schema-on-read (e.g. client parses html document), meaning that the data will be stored without schema, and the schema will not be created until it is read.
- Structured data is usually stored in big data warehouses, while unstructured data will be stored in smaller data centers.
- Structured data has a predefined format that can be read easily by machines, while unstructured data has no specific format.

### Task 2: Term Weighting

**Explain:**

**1. Term Frequency (tf)**

Describes how many times a given term occurs in a given set (e.g. a document).

**2. Document Frequency (df)**

Describes how many documents containing the given term occurs within a collection. The document frequency of a term is always less than its collection frequency.

**3. Inverse Document Frequency (idf)**

This is similar to Document Frequency, but uses higher weights for more specific words. This means that "birch tree" will score higher than "tree" as the term "tree" is more generalized, and expected to appear more frequently. A commonly used model for weighing out a distribution of terms in a collection is "Zipf's law":

$$df_i \propto \frac{i}{j}$$

**4. Why idf is important for term weighting**

IDF provides a foundation for modern term weighting schemes and is used for ranking in almost all IR systems. The reason for this is that it corrects the skew of general vs. specific terms.

## Task 3: IR Models

**Given the following document collection containing words from the set O = {Big, Cat,Small, Dog }, answer the questions in subtasks 3.1 and 3.2.**

**doc1={Big Cat Small Dog}**

**doc2={Dog}**

**doc3={Cat Dog}**

**doc4={Big Cat Big Small Cat Dog}**

**doc5={Big Small}**

**doc6={Small Cat Dog Big }**

**doc7={Big Big Big}**

**doc8={Dog Cat Cat }**

**doc9={Cat Small }**

**doc10={Small Small Big Dog}**

### SubTask 3.1: Boolean Model and Vector Space Model

**Given the following queries:**

**q1 = "Cat AND Dog"**

**q2 = "Cat AND Small"**

**q3 = "Dog OR Big"**

**q4 = "Dog AND NOT Small"**

**q5 = "Cat"**

**1. Which of the documents will be returned as the result for the above queries using the Boolean model? Explain your answers and draw a figure to illustrate.**

Term Document Matrix:

| | **w1** | **w2** | **w3** | **w4** |
|---|---|---|---|---|
| d1 | 1 | 1 | 1 | 1 |
| d2 | 0 | 0 | 0 | 1 |
| d3 | 0 | 1 | 0 | 1 |
| d4 | 1 | 1 | 1 | 1 |
| d5 | 1 | 0 | 1 | 0 |
| d6 | 1 | 1 | 1 | 1 |
| d7 | 1 | 0 | 0 | 0 |
| d8 | 0 | 1 | 0 | 1 |
| d9 | 0 | 1 | 1 | 0 |
| d10 | 1 | 0 | 1 | 1 |

Term vector for w1 = "big" = <1,0,0,1,1,1,1,0,0,1>
Term vector for w2 = "cat" = <1,0,1,1,0,1,0,1,1,0>
Term vector for w3 = "small" = <1,0,0,1,1,1,0,0,1,1>
Term vector for w4 = "dog" = <1,1,1,1,0,1,0,1,0,1>

Query vector for q1 = "Cat AND Dog" = w2 ∧ w4 = <1,0,1,1,0,1,0,1,0,0>
Query vector for q2 = "Cat AND Small" = w2 ∧ w3 = <1,0,0,1,0,1,0,0,1,0>
Query vector for q3 = "Dog OR Big" = w4 ∨ w1 = <1,1,1,1,1,1,1,1,0,1>

Query vector for q4 = "Dog AND NOT Small" = w4 ∧ !w3 = <0,1,1,0,0,0,0,1,0,0>

Query vector for q5 = "Cat" = w2 = <1,0,1,1,0,1,0,1,1,0>

Each query will return the respective documents listed below according to their term vectors:

q1 will return the following documents: doc1,doc3,doc4,doc6,doc8

q2 will return the following documents: doc1,doc4,doc6,doc9

q3 will return the following documents: doc1,doc2,doc3,doc4,doc5,doc6,doc7,doc8,doc10

q4 will return the following documents: doc2,doc3,doc8

q5 will return the following documents: doc1,doc3,doc4,doc6,doc8,doc9

**2. What is the dimension of the vector space representing this document collection when you use the vector model and how is it obtained?**

D = document collection

d = document: w1, w2... $\epsilon$ V, occurs in D

V = vocabulary: w1, w2...

|D| = 10

|V| = 4

The dimensions of the vector space is defined as:

$$|D| \times |V| = 10 \times 4$$

**3. Calculate the weights for the documents and the terms using tf and idf weighting.Put these values into a document-term-matrix.**

Note: we used log base 10 instead of 2 by accident, and did not notice until just before the deadline.

Document-term-matrix:

| | big | cat | small | dog |
|---|---|---|---|---|
| d1 | 0.222 | 0.222 | 0.222 | 0.155 |
| d2 | 0 | 0 | 0 | 0.155 |
| d3 | 0 | 0.222 | 0 | 0.155 |
| d4 | 0.289 | 0.289 | 0.222 | 0.155 |
| d5 | 0.222 | 0 | 0.222 | 0 |
| d6 | 0.222 | 0.222 | 0.222 | 0.155 |
| d7 | 0.328 | 0 | 0 | 0 |

| | big | cat | small | dog |
|---|---|---|---|---|
| d8 | 0 | 0.289 | 0 | 0.155 |
| d9 | 0 | 0.222 | 0.222 | 0 |
| d10 | 0.222 | 0 | 0.289 | 0.155 |

**4. Study the documents 2, 3, 5 and 7 and compare them to document 9. Calculate the similarity between document 9 and these four documents according to Euclidean distance. (Use tf-idf weights for your computations).**

Euclidean distance between 2 and 9:

$$\sqrt{(0-0)^2+(0.222-0)^2+(0.222-0)^2+(0-0.155)^2}=0.35$$

Euclidean distance between 3 and 9:

$$\sqrt{(0-0)^2+(0.222-0.222)^2+(0.222-0)^2+(0-0.155)^2}=0.27$$

Euclidean distance between 5 and 9:

$$\sqrt{(0-0.222)^2+(0.222-0)^2+(0.222-0.222)^2+(0-0)^2}=0.31$$

Euclidean distance between 7 and 9:

$$\sqrt{(0-0.328)^2+(0.222-0)^2+(0.222-0)^2+(0-0)^2}=0.45$$

**5. Rank the documents for query q5 using cosine similarity.**

The cosine similarity between d1 and q5 is:

$$sim(d1,q5)=\frac{(1*1)+(1*0)+(1*0)+(1*0)}{\sqrt{1^2+1^2+1^2+1^2}*\sqrt{1^2+0^2+0^2+0^2}}=0.5$$

And for the rest:

| | q5 |
|---|---|
| d1 | 0.5 |
| d2 | 0 |
| d3 | 0.71 |
| d4 | 0.63 |

|  | q5 |
|---|---|
| d5 | 0 |
| d6 | 0.5 |
| d7 | 0 |
| d8 | 0.89 |
| d9 | 0.71 |
| d10 | 0 |

In ranked order: d8, (d3,d9), d4, (d1,d6), (d2,d5,d7,d10)

**SubTask 3.2: Probabilistic Models**

**Given the following queries:**

**q1 = "Cat Dog"**

**q2 = "Small"**

**1. What are the main differences between BM25 model and the probabilistic model introduced by Robertson-Jones?**

Probabilistic model:
Advantages:

- Documents are ranked in decreasing order of theri probability of being relevant.
  Disadvantages:
- Need to guess initial separation of documents into relevant and non-relevant sets.
- Index terms are assumed to be occuring independent from each other in the document.
- No accounting for term frequency, meaning all weights are binary.
- Lacking document length normalization.
- As frequency of queries found in a document increase, the score increase linearly. Meaning the score doubles when the hits are doubled, which is a bit too agressive.

BM25 model:
Advantages:

- Supports document and query length normalization.
- Accounts for query term frequencies.

**2. Rank the documents using the BM25 model. Set the parameters to k = 1.2 and b = 0.75. (Here we assume relevance information is not provided.)**

**Hint: To avoid getting negative numbers, you need to use idf = log [ N/dft] in the BM25 model.**

$f_{i,j}$ = frequency of a term for a document

TF:

$$tf_{i,j} = 1 + log(f_{i,j}) \text{ if } (f_{i,j} > 0); 0 \text{ otherwise}$$

IDF:

$$idf_j = log(\frac{N}{df_t})$$

$$B_{i,j} = \frac{(k_1 + 1) \cdot tf_{i,j}}{k1 \cdot [(1-b) + b \cdot \frac{len(d_i)}{avgDoclen}] + tf_{i,j}}$$

Using $k1 = 1.2$ and $b = 0.75$:

$$B_{i,j} = \frac{(1.2 + 1) \cdot tf_{i,j}}{1.2 \cdot [(1-0.75) + b \cdot \frac{len(d_i)}{avgDoclen}] + tf_{i,j}}$$

Got stuck on this problem.