

## Introduction to IR Models

# Introduction to Information Retrieval Models

## Information Retrieval vs. Data Retrieval

### Data collections

Well structured collections of related items; items are usually atomic with a well-defined interpretation. Data retrieval involves the selection of a fixed set of data based on a well-defined query (e.g SQL, OQL).

### Information collections

Information, on the other hand, is usually semi-structured or unstructured. Information retrieval (IR) involves the retrieval of documents of natural language which is typically not structured and may be semantically ambiguous.

## Information Retrieval vs. Information Filtering

### The main differences are:

- the nature of the information need
- the nature of the document set

### Similarities

Other than these two differences, the same models are used. Documents and queries are represented using the same set of techniques and similar comparison algorithms are also used.

## User Role

In traditional IR, the user role was pretty well-defined in that a user:

- formulated a query
- viewed the results
- potentially offered feedback
- possibly reformulated query and repeated steps

## User Role

In more recent systems, with the increasing popularity of the hypertext paradigm, users usually intersperses browsing with the traditional querying.  
Raises many new difficulties and challenges

## Pre-processing

## Document Pre-processing

Application of a set of well-known techniques to the documents and queries prior to any comparison. Includes, among others:

- stemming
- stop-word removal
- thesaurus construction

## Stemming

Stemming refers to the reduction of words to a potentially common root.

## Example

*Computerisation, computing, computers* could all be stemmed to common form *comput*.

Stemming involves the reduction of similar words to a common root form. Lovin's and Porter's algorithms are the most common.



## Stop word removal

- This involves the removal of very frequent terms from documents.
- These terms add little to the semantics or meaning of the document.

## Thesaurus construction

Thesauri used to try to identify synonyms within the documents. Manually or automatically created.

## Representation

Representation and comparison technique depends on the information retrieval model chosen. The choice of feedback techniques is also dependent on the model chosen.

## Models

## IR Models

Retrieval models can be broadly categorised as:

- Boolean
  - Classical Boolean
  - Fuzzy Set approach
  - Extended Boolean
- Vector
  - Vector space approach
  - Latent Semantic Indexing
  - Neural Networks
- Probabilistic
  - Inference Network
  - Belief Network

## IR model

Can view any IR model as comprising:

- $D$  is the set of logical representations of the documents.
- $Q$  is the the set of logical representations of the user information needs (queries).
- $F$  is a framework for modelling these representations ( $D$  and  $Q$ ) and the relationship between  $D$  and  $Q$ .
- $R$  is a ranking function which defines an ordering among the documents with regard to any query  $q_i$ .

## IR models

We have a set of index terms:

$$t_1 \dots t_n$$

A weight  $w_{i,j}$  is assigned to each term  $t_i$  occurring in document  $d_j$

Can view a document or query as a vector of weights:

$$\vec{d_j} = (w_1, w_2, w_3 \dots)$$

## Boolean Model



## Boolean Model

- Based on set theory and the Boolean algebra.
- A query is viewed as a Boolean expression.
- The model also assumes terms are present or absent, hence term weights  $w_{i,j}$  are binary and discrete, i.e.,  $w_{i,j}$  is an element of  $\{0, 1\}$
- Suffers from quite a few shortcomings:
  - people often have difficulty formulating expressions
  - documents are considered either relevant or irrelevant; no partial matching allowed

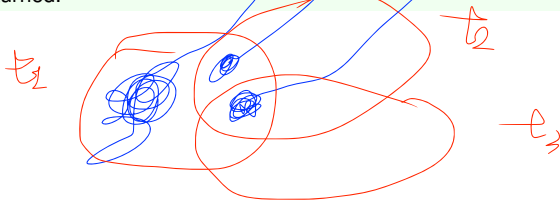
## Example

$$q = t_1 \text{ AND } (t_2 \text{ OR } (\text{NOT } t_3))$$

This can be mapped to what is termed disjunctive normal form, where we have a series of disjunctions (or logical ORs) of conjunctions.

$$q = 100 \vee 110 \vee 111$$

If a document 'satisfies' any of the components, the document is deemed relevant and returned.



## Advantages

Clean formalism popular, widespread relatively simple

## Disadvantages

- Not very good performance.
- Suffers badly from natural language effects of synonymy etc.
- No ranking of results.
- Harbours some difficulty in use.
- Terms in a documents are considered independent of each other.

## Vector Space model

## Vector space model

Attempts to improve upon the Boolean model by removing the limitation of binary weights for index terms.

Terms can have a non-binary weights in both queries and documents.

Hence we can represent documents and query as n-dimensional vectors.

$$\vec{d}_j = (w_{1,j}, w_{2,j} \dots w_{n,j})$$

$$\vec{q} = (w_{1,q}, w_{2,q} \dots w_{n,q})$$

## Similarity

We can calculate the similarity between a document and a query by calculating the similarity between the vector representations of the document and query.

We can measure this similarity by measuring the cosine of the angle between the two vectors.

$$\vec{a} \cdot \vec{b} = |\vec{a}| |\vec{b}| \cos(\vec{a}, \vec{b})$$

$$\Rightarrow \cos(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| |\vec{b}|}$$

We can therefore calculate similarity between document and query as:

$$\text{sim}(q, d) = \cos(\vec{q}, \vec{d}) = \frac{\vec{q} \cdot \vec{d}}{|\vec{q}| |\vec{d}|}$$

Considering term weights on query and document, we can calculate similarity between document and query as:

$$\text{sim}(q, d) = \frac{\sum_{i=1}^N (w_{i,q} \times w_{i,d})}{\sqrt{\sum_{i=1}^N (w_{i,q})^2} \times \sqrt{\sum_{i=1}^N (w_{i,d})^2}}$$



## Weighting schemes

- We need a means to calculate the term weights in the document and query vector representations.
- A term's frequency within a document quantifies how well a term describes a document. The more frequent a term occurs in a document, the better it is at describing that document and vice-versa.
- This frequency is known as the term frequency or *tf* factor.

## Weighting schemes

- If a term occurs frequently across all the documents, that term does little to distinguish one document from another. This factor is known as the inverse document frequency (*idf*-frequency).
- Traditionally, the most commonly used weighting schemes are known as *tf-idf* weighting schemes.

For all terms in a document, the weight assigned can be calculated as:

$$w_{i,j} = f_{i,j} \times \log\left(\frac{N}{N_i}\right)$$

where

- $f_{i,j}$  is the (possibly normalised) frequency of term  $t_i$  in document  $d_j$
- $N$  is the number of documents in the collection
- $N_i$  is the number of documents that contain term  $t_i$ .

## Advantages

Improved performance over the Boolean model due to weighting schemes Partial matching allowed which gives a natural ranking

## Disadvantages

Terms are considered to be mutually independent