## Information retrieval models

- Documents and queries are characterized by a number of index terms
  - Based on a query (representation of an information problem), guess the relevance of each document
  - Rank documents in the order of relevance
  - Return the most relevant documents
- The effectiveness of an IR system depends on the ability of the document representation to capture the “meaning” of the documents with respect to the users’ needs



## Query methods

- Browsing
- Adhoc retrieval
  - Document collection remains stable, users try to find relevant documents using adhoc queries
- Filtering
  - User queries remain stable as “profiles”
  - As new documents are added they are sent to users who might be interested in these documents
  - Profiles can be constructed on keyword queries, terms occurring in documents retrieved by users

## Information retrieval model

- An information retrieval model is a quadruple <D,Q,F,R($q_i$, $d_j$)> where
  - D is a set composed of logical views (or representations) for the documents in the collection
  - Q is a set composed of logical views (or representations) for the user information needs called “queries”
  - F is a framework for modeling document representations, queries and their relationships
  - R($q_i$, $d_j$) is a ranking function which associates a real number with a query $q_i$ in Q and a document representation $d_j$ in D.



## Documents

- A document is a collection of words
- An index term is an “important” word that
  - Possess a meaning, such as a noun and has been simplified (stop words, stemming)
  - Distinguishes the document from the others
- The set of all index terms for a document collection is given by {$k_1$,…,$k_t$}
- A document $d_j$ in IR is usually given by a vector:

  $d_j$ = <$w_{1,j}$, …, $w_{t,j}$> where $w_{i,j}$ is the weight of term $k_i$ in document $d_j$.

## Documents

- Assumption:
  - The occurrence of a term $t_1$ in a document is completely independent of the occurrence of another term $t_2$ in the same document
  - Not true in general, but does not appear to have a big impact on the retrieval effectiveness



## Boolean model for retrieval

- A Boolean query contains query terms connected by logical connectives and, or, not.
- A Boolean query is interpreted as a set membership function.
- Query:
  - Q = “UFO” return documents that contain the word “UFO”
  - Q = “UFO Sightings” AND “Albany” return documents that contain the phrase “UFO Sightings” and the word “Albany”

# Boolean model for retrieval

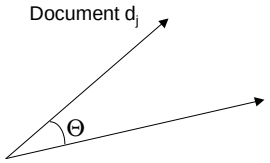
- Q = $k_a$ and ($k_b$ or not $k_c$) return documents
  - that contain the word $k_a$ and
  - either contain $k_b$ or does not contain $k_c$
- In the boolean model, each document either
  - satisfies the query, then we return 1 (relevant)
  - does not satisfy the query, then we return 0 (irrelevant)
- Documents can be represented as a vector of 0s and 1s
  - 1 if a term appears and 0 if it does not appear



# Vector model

- In the vector model, both queries and documents are weighted vectors
- The relevance of a document to a query is given by the "cosine of the angle" between a document vector and a query vector

$$Sim(d_j,q) = sum_{i=1..t}(w_{i,j} \cdot w_{iq}) / sqrt( sum_{i=1..t} (w_{i,j}^2) \cdot sum_{i=1..t} (w_{i,q}^2) )$$




$Cos(\Theta) = Sim(d_j,q)$

# Vector model

- The importance of a term in a document depends on:
  - How important it is for identifying the content of this document (term frequency)

    $f_{i,j} = freq_{i,j} / (max_l\ freq_{l,j})$

    frequency of term $k_i$ in document $d_j$, versus the highest frequency of a term in the same document
  - How important it is for identifying the document from the others (document frequency)

    $idf_i = \log N/n_i$ total number of documents versus total number of documents containing this term

    The term weight is given by $f_{i,j} * idf_i$



# Vector model

- A user query consists of a number of terms
- How do we assign weights to query terms:

$$w_{i,q} = (.5 + (.5\ freq_{i,q} / max_l\ freq_{l,q})) \cdot \text{Log } N/n_i$$

# Fuzzy set model

- A fuzzy set has a membership function, $\mu_A(u)$, that returns a real number $0 <= \mu(A) <= 1$.
  - If $\mu_A(u) = 1$, then A is definitely a member
  - If $\mu_A(u) = 0$, then A is definitely not a member
- Fuzzy sets use a number of pre-set functions to determine the meaning of various connectives
  - $\mu_{not\ A}(u) = 1 - \mu_A(u)$
  - $\mu_{A\ or\ B}(u) = \max \{\mu_A(u), \mu_B(u)\}$ or $\mu_A(u) + \mu_B(u)$
  - $\mu_{A\ and\ B}(u) = \min \{\mu_A(u), \mu_B(u)\}$ or $\mu_A(u) * \mu_B(u)$



# Fuzzy set model

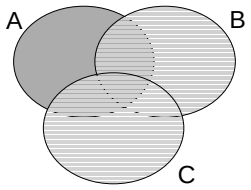
- Determine the term-to-term correlation in a collection of documents between terms $k_i$ and $k_l$

$c_{i,l} = n_{i,l} / (n_i + n_l - n_{i,l})$ where $n_x$ is the number of documents containing term $k_x$

Then, compute $\mu_{i,j} = 1 - ( \text{product}_{kl\ in\ dj} (1 - c_{i,l}))$
the degree of membership of document $d_j$ to term $k_i$

# Fuzzy queries

- Given a query $q=k_i$ then similarity of a document $d_j$ to q is given by $\mu_{i,j}$
- Given a query $q= k_i$ AND $k_l$, the similarity of a document $d_j$ to query q is given by $\mu_{i,j} * \mu_{l,j}$ (or using any appropriate operator for AND)
- Similarly for OR (use + or max)
- Given a complex query: (A and (not B)) or (C),





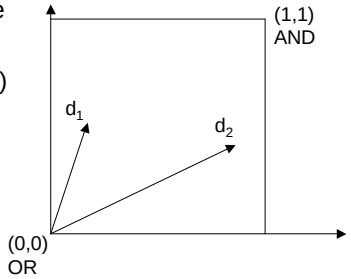

# Extended Boolean Model

- Suppose, you are given a query containing keywords $k_x$ and $k_y$
- Assume, the weight of terms $k_x$ and $k_y$ in document $d_j$ are given by $(x_1,y_1)$
- Given query "$k_x$ OR $k_y$", we would like

to be as far away from (0,0) as possible
hence maximize distance((0,0), $(x_1,y_1)$)

- Given query "$k_x$ AND $k_y$", we would like

to be as close to (1,1) as possible
hence maximize 1 - distance((1,1), $(x_1,y_1)$)



# Extended Boolean Model

- Under this model:
  - Sim(or-query, d) = sqrt( (x^2+y^2)/2)
  - Sim(or-query, d) = 1 - sqrt( ((1-x)^2+(1-y)^2)/2)
- Suppose now connectives and/or have a degree "p"
  - I.e. or-query: $k_1$ $OR^p$ $k_2$ $OR^o$ … $OR^p$ $k_m$
  - sim(or-query, d) = power((x1^p+x2^p+…+xm^p)/m), 1/p)
  - I.e. and-query: $k_1$ $AND^p$ $k_2$ $AND^o$ … $AND^p$ $k_m$
  - sim(and-query, d) = 1 - power(((1-x1)^p+(1-x2)^p+…+(1-xm)^p)/m), 1/p)



# Extended Boolean Model

- Given p-norms, we have the following properties:
  - If p = 1, then sim(or-query)=sim(and-query)= (x1+…+xm)/m
  - Reduces to arithmetic mean
  - If p = ∞, then sim(or-query)= min(xk) and sim(and-query) = max(xk)