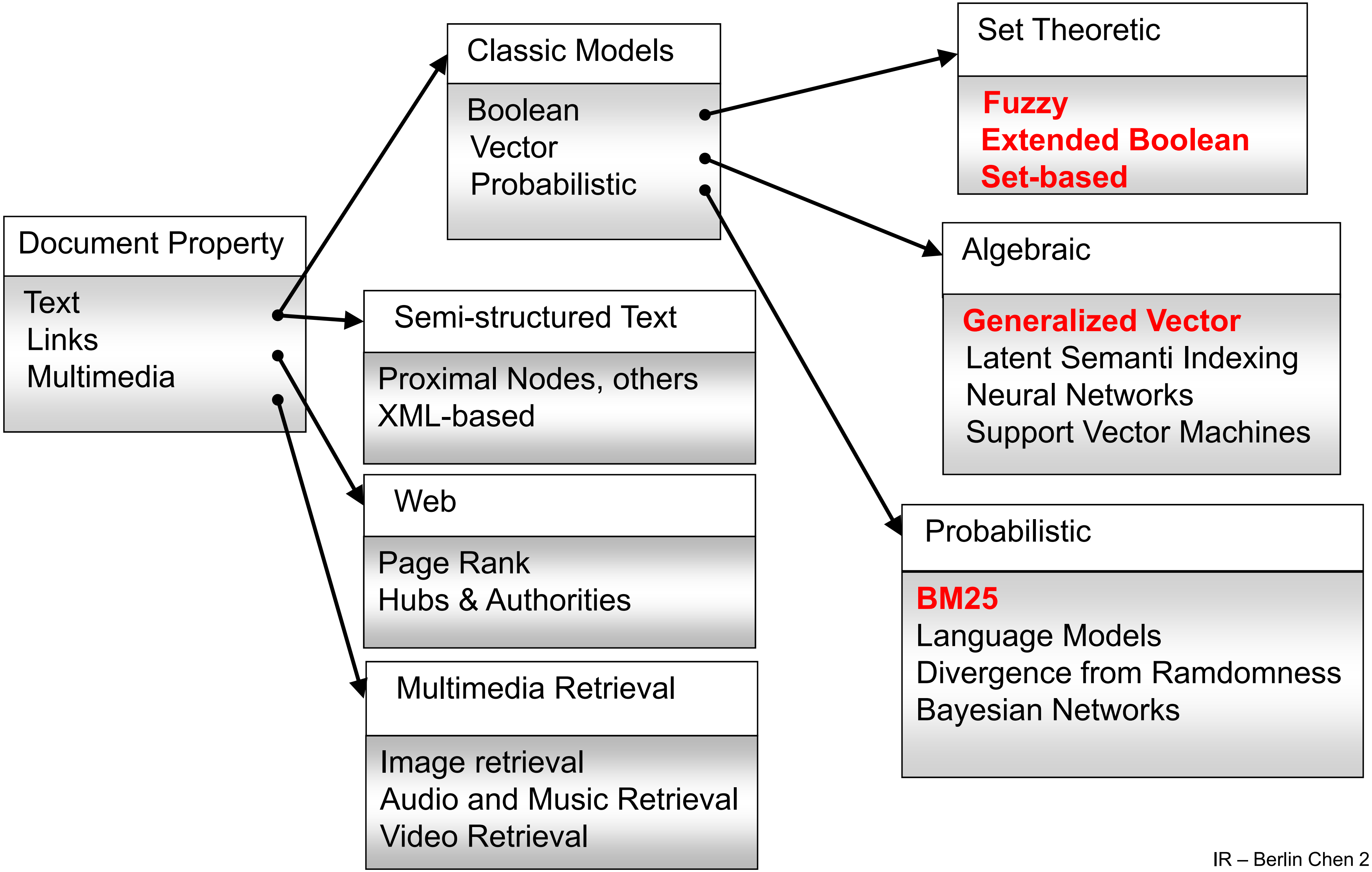
Modeling in Information Retrieval

- Fuzzy Set, Extended Boolean, Generalized Vector Space, Set-based Models, and Best Match Models

Berlin Chen
Department of Computer Science & Information Engineering
National Taiwan Normal University

References:
1. *Modern Information Retrieval*, Chapter 3 & Teaching material
2. *Language Modeling for Information Retrieval*, Chapter 3

Taxonomy of Classic IR Models

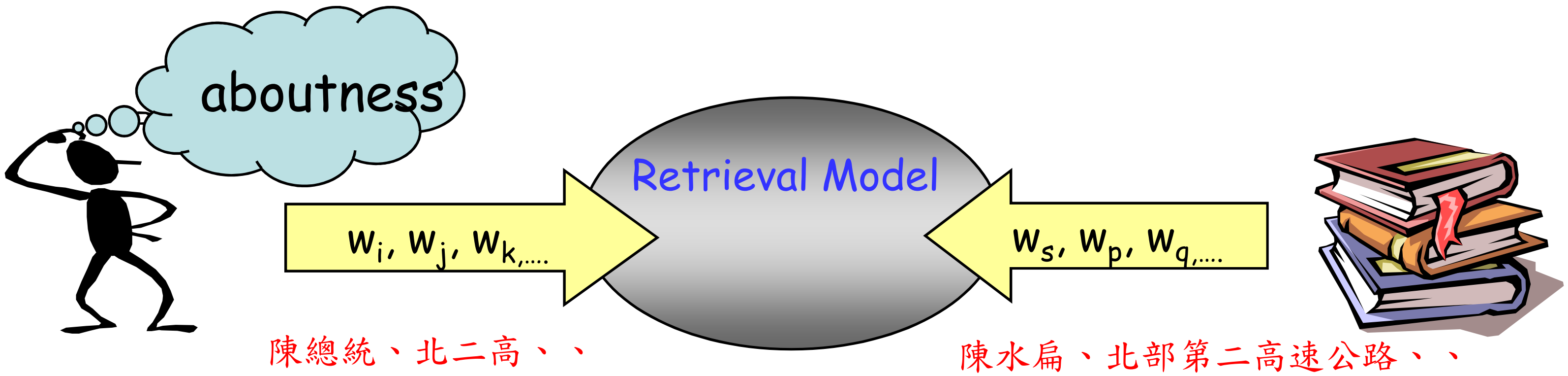


Outline

- Alternative Set Theoretic Models
  - Fuzzy Set Model (Fuzzy Information Retrieval)
  - Extended Boolean Model
  - Set-based Model
- Alternative Algebraic Model
  - Generalized Vector Space Model
- Alternative Probabilistic Models
  - Best Match Models (BM1, BM15, BM11 & BM 25)

Fuzzy Set Model

- Premises
  - Docs and queries are represented through sets of keywords, therefore the matching between them is vague
    - Keywords cannot completely describe the user's information need and the doc's main theme



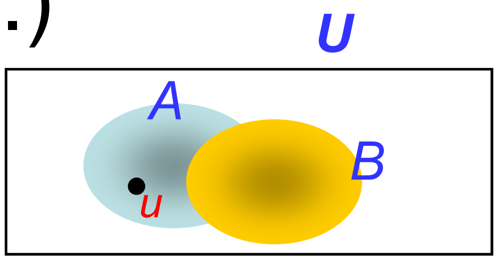
  - For each query term (keyword)
    - Define a fuzzy set and that each doc has a degree of membership (0~1) in the set

Fuzzy Set Model (cont.)

- Fuzzy Set Theory
  - Framework for representing classes (sets) whose boundaries are not well defined
  - Key idea is to introduce the notion of a *degree of membership* associated with the elements of a set
  - This degree of membership varies from 0 to 1 and allows modeling the notion of *marginal* membership
    - 0 →no membership
    - 1 →full membership
  - Thus, membership is now a gradual instead of abrupt
    - Not as conventional Boolean logic

Here we will define a fuzzy set for each query (or index) term, thus each doc has a degree of membership in this set.

Fuzzy Set Model (cont.)



- Definition
  - A fuzzy subset A of a universal of discourse U is characterized by a membership function $\mu_A: U \rightarrow [0,1]$
    - Which associates with each element u of U a number $\mu_A(u)$ in the interval [0,1]
  - Let A and B be two fuzzy subsets of U. Also, let $\overline{A}$ be the complement of A. Then,
    - Complement $\mu_{\overline{A}}(u) = 1 - \mu_A(u)$
    - Union $\mu_{A \cup B}(u) = \max(\mu_A(u), \mu_B(u))$
    - Intersection $\mu_{A \cap B}(u) = \min(\mu_A(u), \mu_B(u))$

Fuzzy Set Model (cont.)

- Fuzzy information retrieval

  Defining term relationship

  - Fuzzy sets are modeled based on a **thesaurus**
  - This thesaurus can be constructed by a term-term correlation matrix (or called keyword connection matrix)
    - $\vec{c}$: a term-term correlation matrix
    - $c_{i,l}$: a normalized correlation factor for terms k_i and k_l

$$c_{i,l} = \frac{n_{i,l}}{n_i + n_l - n_{i,l}}$$

ranged from 0 to 1

n_i	: no of docs that contain k_i
$n_{i,l}$	: no of docs that contain both k_i and k_l

docs, paragraphs, sentences, ..

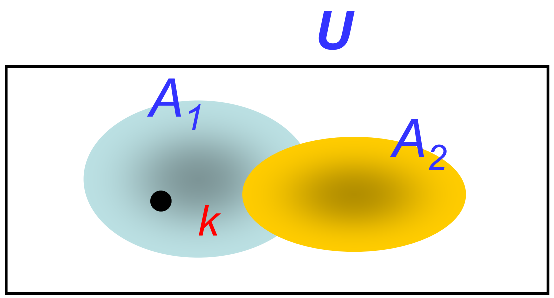
- We now have the notion of proximity among index terms

  - The relationship is symmetric !

$$\mu_{k_i}(k_l) = c_{i,l} = c_{l,i} = \mu_{k_l}(k_i)$$

Fuzzy Set Model (cont.)

- The union and intersection operations are modified here



$$ab+\overline{a}b+a\overline{b}$$
$$=ab+(1-a)b+a(1-b)$$
$$=ab+b-ab+a-ab$$
$$=1-(1-a-b+ab)$$
$$=1-(1-a)(1-b)$$

  - Union: algebraic sum (instead of max)

$$\mu_{A_1\cup A_2}(k)=\mu_{A_1}(k)\mu_{A_2}(k)+\mu_{\overline{A_1}}(k)\mu_{A_2}(k)+\mu_{A_1}(k)\mu_{\overline{A_2}}(k)$$
$$=1-\prod_{j=1}^{2}\left(1-\mu_{A_j}(k)\right)$$

a negative algebraic product

$$\Rightarrow \quad \mu_{A_1\cup A_2\cdots\cup A_n}(k)=\mu_{\cup_j A_j}(k)=1-\prod_{j=1}^{n}\left(1-\mu_{A_j}(k)\right)$$

  - Intersection: algebraic product (instead of min)

$$\mu_{A_1\cap A_2}(k)=\mu_{A_1}(k)\mu_{A_2}(k)$$

$$\Rightarrow \quad \mu_{A_1\cap A_2\cdots\cap A_n}(k)=\prod_{j=1}^{n}\mu_{A_j}(k)$$

Fuzzy Set Model (cont.)

– The degree of membership between a doc d_j and an index term k_i

algebraic sum (a doc is a union of index terms)

	k_a	k_b
k_i	$c_{i,a}$	$c_{i,b}$
	$1-c_{i,a}$	$1-c_{i,b}$

$$\mu_{k_i}(d_j) = \boxed{\mu_{d_j}(k_i) = \mu_{\underset{k_l \in d_j}{\cup} k_l}(k_i)}$$

$$= 1 - \prod_{k_l \in d_j}\left(1 - \mu_{k_l}(k_i)\right) = 1 - \prod_{k_l \in d_j}\left(1 - c_{i,l}\right)$$

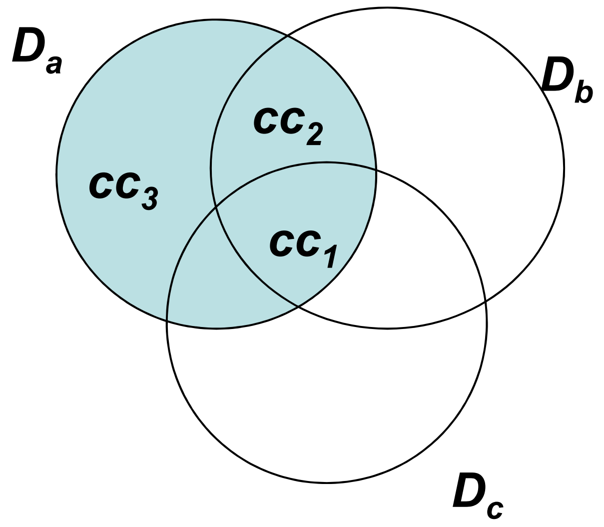
- Computes an **algebraic sum** over all terms in the doc d_j
  - Implemented as the complement of a negative algebraic product
  - A doc d_j belongs to the fuzzy set associated to the term k_i if its own terms are related to k_i
- If there is at least one index term k_l of d_j which is strongly related to the index k_i ($c_{i,l} \sim 1$) then $\mu_{k_i,d_j} \sim 1$
  - k_i is a good fuzzy index for doc d_j
  - And vice versa

Fuzzy Set Model (cont.)

- Example:
  - Query $q = k_a \wedge (k_b \vee \neg k_c)$

    disjunctive normal form

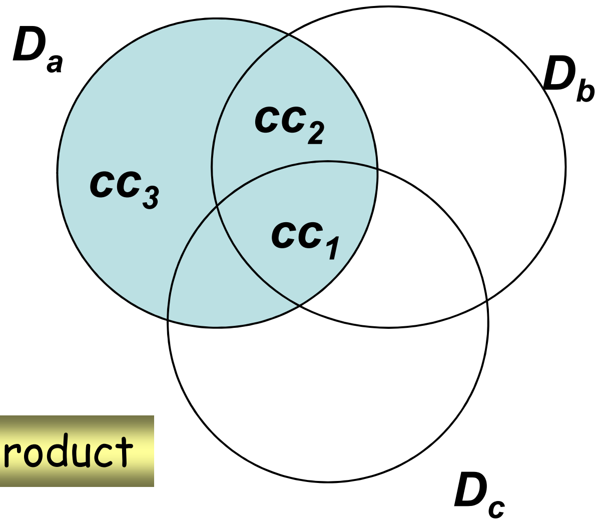
    $\vec{q}_{dnf} = (k_a \wedge k_b \wedge k_c) \vee (k_a \wedge k_b \wedge \neg k_c) \vee (k_a \wedge \neg k_b \wedge \neg k_c)$
    $= cc_1 + cc_2 + cc_3$ conjunctive component



  - D_a is the fuzzy set of docs associated to the term k_a
  - Degree of membership ?

Fuzzy Set Model (cont.)

- Degree of membership



algebraic sum

$$\mu_q(d_j) = \mu_{cc_1 \cup cc_2 \cup cc_3}(d_j)$$

for a doc d_j in the fuzzy answer set D_q

negative algebraic product

$$= 1 - \prod_{i=1}^{3}\left(1 - \mu_{cc_i}(d_j)\right)$$

$$= 1 - \left(1 - \overbrace{\mu_{a \cap b \cap c}(d_j)}^{cc_1}\right)\left(1 - \overbrace{\mu_{a \cap b \cap \bar{c}}(d_j)}^{cc_2}\right)\left(1 - \overbrace{\mu_{a \cap \bar{b} \cap \bar{c}}(d_j)}^{cc_3}\right)$$

algebraic product

$$= 1 - \left(1 - \mu_a(d_j)\mu_b(d_j)\mu_c(d_j)\right)$$
$$\times \left(1 - \mu_a(d_j)\mu_b(d_j)(1 - \mu_c(d_j))\right) \times \left(1 - \mu_a(d_j)(1 - \mu_b(d_j))(1 - \mu_c(d_j))\right)$$

More on Fuzzy Set Model

- Advantages
  - The correlations among index terms are considered
  - Degree of relevance between queries and docs can be achieved

- Disadvantages
  - Fuzzy IR models have been discussed mainly in the literature associated with fuzzy theory
  - Experiments with standard test collections are not available
  - Do not consider the frequecny (or counts) of a term in a document or a query

Extended Boolean Model

Salton et al., 1983

- Motive
  - Extend the Boolean model with the functionality of partial matching and term weighting 陳水扁 及 呂秀蓮
    - E.g.: in Boolean model, for the qery $q=k_x \wedge k_y$, a doc contains either k_x or k_y is as irrelevant as another doc which contains neither of them
    - How about the disjunctive query $q=k_x \vee k_y$ 陳水扁 或 呂秀蓮
  - Combine Boolean query formulations with characteristics of the vector model
    - Term weighting
    - Algebraic distances for similarity measures

a ranking can be obtained

Extended Boolean Model (cont.)

- Term weighting
  - The weight for the term k_x in a doc d_j is

$$w_{x,j} = tf_{x,j} \times \boxed{\frac{idf_x}{\max_i idf_i}}$$

normalized frequency

Normalized *idf*

ranged from 0 to 1

    - $w_{x,j}$ is normalized to lie between 0 and 1

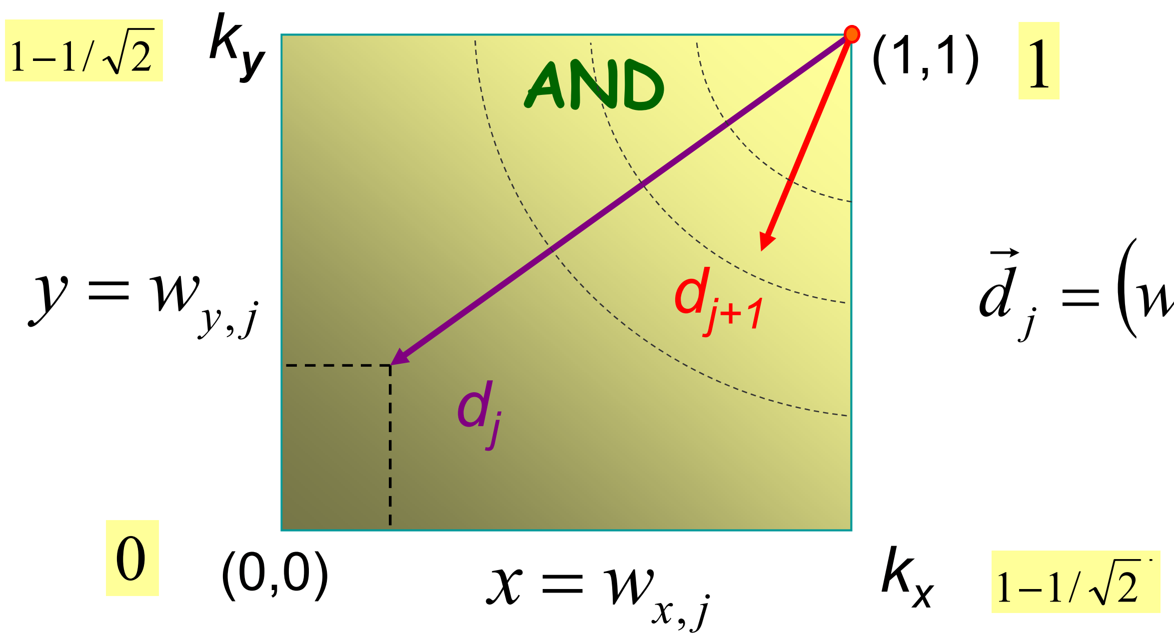
- Assume two index terms k_x and k_y were used
  - Let x denote the weight $w_{x,j}$ of term k_x on doc d_j
  - Let y denote the weight $w_{y,j}$ of term k_y on doc d_j
  - The doc vector $\vec{d}_j = (w_{x,j}, w_{y,j})$ is represented as $d_j = (x, y)$
  - Queries and docs can be plotted in a two-dimensional map

Extended Boolean Model (cont.)

- If the query is $q = k_x \wedge k_y$ (conjunctive query)
  - -The docs near the point (1,1) are preferred
  - -The **similarity measure** is defined as

$$sim(q_{and}, d) = 1 - \sqrt{\frac{(1-x)^2 + (1-y)^2}{2}}$$

2-norm model
(Euclidean distance)



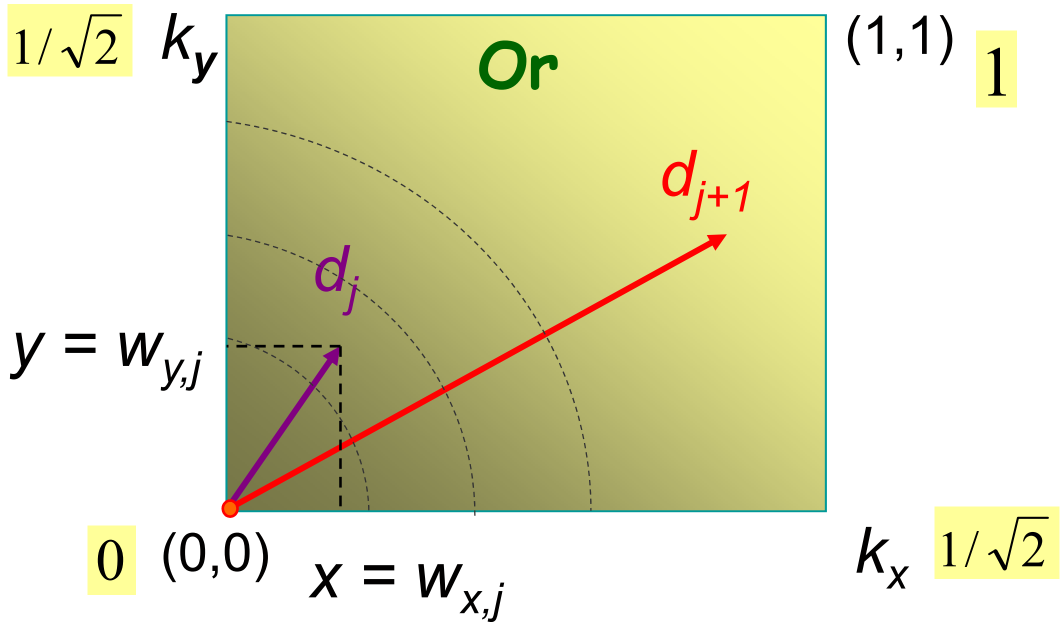
$$\vec{d}_j = (w_{x,j}, w_{y,j})$$

Extended Boolean Model (cont.)

- If the query is $q = k_x \vee k_y$ (disjunctive query)
  - The docs far from the point (0,0) are preferred
  - The similarity measure is defined as

$$sim(q_{or}, d) = \sqrt{\frac{x^2 + y^2}{2}}$$

2-norm model
(Euclidean distance)



Extended Boolean Model (cont.)

- The similarity measures $sim\left(q_{or}, d\right)$ and $sim\left(q_{and}, d\right)$ also lie between 0 and 1

Extended Boolean Model (cont.)

- Generalization
  - *t* index terms are used → *t*-dimensional space
  - *p*-norm model, $1 \leq p \leq \infty$

$$q_{and} = k_1 \wedge^p k_2 \wedge^p \ldots \wedge^p k_m \Rightarrow sim(q_{and}, d) = 1 - \left(\frac{(1-x_1)^p + (1-x_2)^p + \ldots + (1-x_m)^p}{m} \right)^{\frac{1}{p}}$$

$$q_{or} = k_1 \vee^p k_2 \vee^p \ldots v^p k_m \Rightarrow sim(q_{or}, d) = \left(\frac{x_1^p + x_2^p + \ldots + x_m^p}{m} \right)^{\frac{1}{p}}$$

  - Some interesting properties

Similar to vector space model

- $p=1$ $\Rightarrow$ $sim(q_{and}, d) = sim(q_{or}, d) = \dfrac{x_1 + x_2 + \ldots + x_m}{m}$
- $p=\infty$ $\Rightarrow$ $sim(q_{and}, d) \approx \min(x_i)$

$$sim(q_{or}, d) \approx \max(x_i)$$

just like the formula of fuzzy logic

Extended Boolean Model (cont.)

- Example query 1: $q = (k_1 \wedge^p k_2) \vee^p k_3$
  - Processed by grouping the operators in a predefined order

$$sim(q, d) = \left(\frac{\left(1 - \left(\frac{(1-x_1)^p + (1-x_2)^p}{2} \right)^{\frac{1}{p}} \right)^p + x_3^p}{2} \right)^{\frac{1}{p}}$$

- Example query 2: $q = (k_1 \vee^2 k_2) \wedge^\infty k_3$
  - Combination of different algebraic distances

$$sim(q, d) = \min \left(\left(\frac{x_1^2 + x_2^2}{2} \right)^{\frac{1}{2}}, x_3 \right)$$

More on Extended Boolean Model

- Advantages
  - A hybrid model including properties of both the set theoretic models and the algebraic models
  - That is, relax the Boolean algebra by interpreting Boolean operations in terms of algebraic distances
    - By varying the parameter *p* between 1 and infinity, we can vary the *p* -norm ranking behavior from that of a vector-like ranking to that of a fuzzy logic-like ranking
    - Have the possibility of using combinations of different values of the parameter *p* in the same query request

More on Extended Boolean Model (cont.)

- Disadvantages
  - Distributive operation does not hold for ranking computation
    - E.g.:

$$q_1 = \left(k_1 \wedge^2 k_2\right) \vee^2 k_3, q_2 = \left(k_1 \vee^2 k_3\right) \wedge^2 \left(k_2 \vee^2 k_3\right)$$

$$\left(\frac{\left(1-\left(\frac{(1-x_1)^2+(1-x_2)^2}{2}\right)^{\frac{1}{2}}\right)^2 + x_3^2}{2}\right)^{\frac{1}{2}}$$

$$sim\left(q_1, d\right) \neq sim\left(q_2, d\right)$$

$$1-\left(\frac{\left(1-\left(\frac{x_1^2+x_2^2}{2}\right)\right)^2+\left(1-\left(\frac{x_2^2+x_3^2}{2}\right)\right)^2}{2}\right)^{\frac{1}{2}}$$

  - Assumes mutual independence of index terms

Generalized Vector Model

Wong et al., 1985

- Premise
  - Classic models enforce independence of index terms
  - For the **Vector model**
    - Set of term vectors $\{\vec{k}_1, \vec{k}_1, \ldots, \vec{k}_t\}$ are linearly independent and form a basis for the subspace of interest
    - Frequently, it means pairwise orthogonality

      $\forall i,j \Rightarrow \vec{k}_i \bullet \vec{k}_j = 0$ (in a more restrictive sense)
- Wong et al. proposed an interpretation
  - An alternative intepretation: The index term vectors are linearly independent, but not pairwise orthogonal
    - Generalized Vector Model

Generalized Vector Model (cont.)

- **Key idea**
  - Index term vectors form the basis of the space are not orthogonal and are represented in terms of smaller components (minterms)
- **Notations**
  - $\{k_1, k_2, ..., k_t\}$: the set of all terms
  - $w_{i,j}$: the weight associated with $[k_i, d_j]$
  - **Minterms**:binary indicators (0 or 1) of all patterns of occurrence of terms within documents
    - Each represents one kind of co-occurrence of index terms in a specific document

Generalized Vector Model (cont.)

- Representations of **minterms**

$m_1=(0,0,\ldots.,0)$
$m_2=(1,0,\ldots.,0)$
$m_3=(0,1,\ldots.,0)$
$m_4=(1,1,\ldots.,0)$
$m_5=(0,0,1,..,0)$
…
$m_{2^t}=(1,1,1,..,1)$

2^t minterms

Points to the docs where only index terms k_1 and k_2 co-occur and the other index terms disappear

Point to the docs containing all the index terms

$\vec{m}_1=(1,0,0,0,0,\ldots.,0)$
$\vec{m}_2=(0,1,0,0,0,\ldots.,0)$
$\vec{m}_3=(0,0,1,0,0,\ldots.,0)$
$\vec{m}_4=(0,0,0,1,0,\ldots.,0)$
$\vec{m}_5=(0,0,0,0,1,\ldots.,0)$
…
$\vec{m}_{2^t}=(0,0,0,0,0,\ldots.,1)$

2^t minterm vectors

Pairwise orthogonal vectors $\vec{m}_i$ associated with minterms m_i as the basis for the **generalized vector space**

Generalized Vector Model (cont.)

- Minterm vectors are **pairwise orthogonal**. But, this does not mean that the index terms are independent

  – Each **minterm** specifies a kind of dependence among index terms

  – That is, the co-occurrence of index terms inside docs in the collection induces dependencies among these index terms

Generalized Vector Model (cont.)

- The vector associated with the term k_i is represented by **summing** up all minterms containing it and **normalizing**

$$\vec{k}_i = \frac{\sum_{\forall r, g_i(m_r)=1} c_{i,r} \vec{m}_r}{\sqrt{\sum_{\forall r, g_i(m_r)=1} c_{i,r}^2}} = \sum_{\forall r, g_i(m_r)=1} \hat{c}_{i,r} \vec{m}_r$$

$$\text{where} \quad \hat{c}_{i,r} = \frac{c_{i,r}}{\sqrt{\sum_{\forall r, g_i(m_r)=1} c_{i,r}^2}}$$

$$c_{i,r} = \sum_{\boxed{d_j | g_l(\vec{d}_j) = g_l(m_r), \text{ for all } l}} w_{i,j}$$

All the docs whose term co-occurrence relation (pattern) can be represented as (exactly coincide with that of) minterm m_r

- The weight associated with the pair [k_i, m_r] sums up the weights of the term k_i in all the docs which have a term occurrence pattern given by m_r.
- Notice that for a collection of size N, only N minterms affect the ranking (and not 2^N)

$g_i(m_r)$ Indicates the index term k_i is in the minterm m_r

Generalized Vector Model (cont.)

- The similarity between the query and doc is calculated in the space of minterm vectors

$$\vec{d}_j = \sum_i w_{i,j} \vec{k}_i \quad \Rightarrow \quad = \sum_r s_{j,r} \vec{m}_r$$

$$\vec{q}_j = \sum_i w_{i,q} \vec{k}_i \quad \Rightarrow \quad = \sum_r s_{q,r} \vec{m}_r$$

t-dimensional 2^t-dimensional

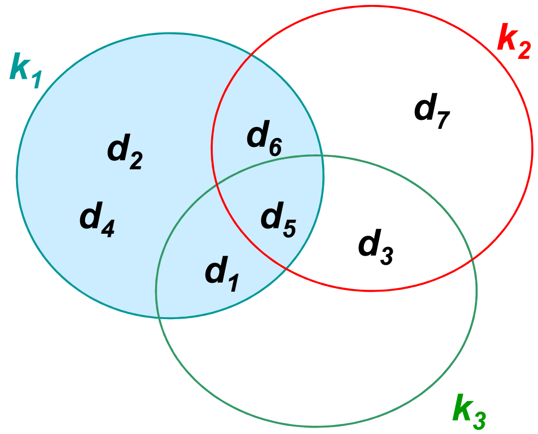
$$sim\left(\vec{q}_j, \vec{d}_j\right) = \frac{\sum_i w_{i,q} \cdot w_{i,j}}{\sqrt{\sum_i w_{i,q}} \sqrt{\sum_i w_{i,q}}}$$

$$sim\left(\vec{q}_j, \vec{d}_j\right) = \frac{\sum_r s_{q,r} \cdot s_{d,r}}{\sqrt{\sum_r s_{q,r}} \sqrt{\sum_r s_{d,r}}}$$

Generalized Vector Model (cont.)

- **Example** (a system with three index terms)

minterm	k_1	k_2	k_3
m_1	0	0	0
m_2	1	0	0
m_3	0	1	0
m_4	1	1	0
m_5	0	0	1
m_6	1	0	1
m_7	0	1	1
m_8	1	1	1



$$\vec{k}_1 = \frac{c_{1,2}\vec{m}_2 + c_{1,4}\vec{m}_4 + c_{1,6}\vec{m}_6 + c_{1,8}\vec{m}_8}{\sqrt{{c_{1,2}}^2 + {c_{1,4}}^2 + {c_{1,6}}^2 + {c_{1,8}}^2}}$$

$$\vec{k}_2 = \frac{c_{2,3}\vec{m}_3 + c_{2,4}\vec{m}_4 + c_{2,7}\vec{m}_7 + c_{2,8}\vec{m}_8}{\sqrt{{c_{2,3}}^2 + {c_{2,4}}^2 + {c_{2,7}}^2 + {c_{2,8}}^2}}$$

$$\vec{k}_3 = \frac{c_{3,5}\vec{m}_5 + c_{3,6}\vec{m}_6 + c_{3,7}\vec{m}_7 + c_{3,8}\vec{m}_8}{\sqrt{{c_{3,5}}^2 + {c_{3,6}}^2 + {c_{3,7}}^2 + {c_{3,8}}^2}}$$

	k_1	k_2	k_3	minterm
d_1	2	0	1	m_6
d_2	1	0	0	m_2
d_3	0	1	3	m_7
d_4	2	0	0	m_2
d_5	1	2	4	m_8
d_6	1	2	0	m_4
d_7	0	5	0	m_3
q	1	2	3	

$$c_{1,2} = w_{1.2} + w_{1.4} = 1 + 2 = 3$$
$$c_{1,4} = w_{1.6} = 1$$
$$c_{1,6} = w_{1,1} = 2$$
$$c_{1,8} = w_{1,5} = 1$$

$$\vec{k}_1 = \frac{3\vec{m}_2 + 1\vec{m}_4 + 2\vec{m}_6 + 1\vec{m}_8}{\sqrt{3^2 + 1^2 + 2^2 + 1^2}}$$

$$c_{2,3} = w_{2,7} = 5$$
$$c_{2,4} = w_{2,6} = 2$$
$$c_{2,7} = w_{2,3} = 1$$
$$c_{2,8} = w_{2,5} = 2$$

$$\vec{k}_2 = \frac{5\vec{m}_3 + 2\vec{m}_4 + 1\vec{m}_7 + 2\vec{m}_8}{\sqrt{5^2 + 2^2 + 1^2 + 2^2}}$$

$$c_{3,5} = 0$$
$$c_{3,6} = w_{3,1} = 1$$
$$c_{3,7} = w_{3,3} = 3$$
$$c_{3,8} = w_{3,5} = 4$$

$$\vec{k}_3 = \frac{0\vec{m}_5 + 1\vec{m}_6 + 3\vec{m}_7 + 4\vec{m}_8}{\sqrt{0^2 + 1^2 + 3^2 + 4^2}}$$

Generalized Vector Model (cont.)

- **Example**: Ranking

$$\vec{k}_1 = \frac{3\vec{m}_2 + 1\vec{m}_4 + 2\vec{m}_6 + 1\vec{m}_8}{\sqrt{3^2 + 1^2 + 2^2 + 1^2}} = \frac{3\vec{m}_2 + 1\vec{m}_4 + 2\vec{m}_6 + 1\vec{m}_8}{\sqrt{15}}$$

$$\vec{k}_2 = \frac{5\vec{m}_3 + 2\vec{m}_4 + 1\vec{m}_7 + 2\vec{m}_8}{\sqrt{5^2 + 2^2 + 1^2 + 2^2}} = \frac{5\vec{m}_3 + 2\vec{m}_4 + 1\vec{m}_7 + 2\vec{m}_8}{\sqrt{34}} \qquad \vec{k}_3 = \frac{0\vec{m}_5 + 1\vec{m}_6 + 3\vec{m}_7 + 4\vec{m}_8}{\sqrt{0^2 + 1^2 + 3^2 + 4^2}} = \frac{1\vec{m}_6 + 3\vec{m}_7 + 4\vec{m}_8}{\sqrt{26}}$$

$$\vec{d}_1 = 2\vec{k}_1 + 1\vec{k}_3$$

$$= \underbrace{\frac{2 \cdot 3}{\sqrt{15}}}_{s_{d1,2}} \vec{m}_2 + \underbrace{\frac{2 \cdot 1}{\sqrt{15}}}_{s_{d1,4}} \vec{m}_4 + \underbrace{\left(\frac{2 \cdot 2}{\sqrt{15}} + \frac{1 \cdot 1}{\sqrt{26}} \right)}_{s_{d1,6}} \vec{m}_6 + \underbrace{\frac{1 \cdot 3}{\sqrt{26}}}_{s_{d1,7}} \vec{m}_7 + \underbrace{\left(\frac{2 \cdot 1}{\sqrt{15}} + \frac{1 \cdot 4}{\sqrt{26}} \right)}_{s_{d1,8}} \vec{m}_8$$

$$\vec{q} = 1\vec{k}_1 + 2\vec{k}_2 + 3\vec{k}_3$$

$$= \underbrace{\frac{1 \cdot 3}{\sqrt{15}}}_{s_{q,2}} \vec{m}_2 + \underbrace{\frac{2 \cdot 5}{\sqrt{34}}}_{s_{q,3}} \vec{m}_3 + \underbrace{\left(\frac{1 \cdot 1}{\sqrt{15}} + \frac{2 \cdot 2}{\sqrt{34}} \right)}_{s_{q,4}} \vec{m}_4 + \underbrace{\left(\frac{1 \cdot 2}{\sqrt{15}} + \frac{3 \cdot 1}{\sqrt{26}} \right)}_{s_{q,6}} \vec{m}_6 + \underbrace{\left(\frac{2 \cdot 1}{\sqrt{34}} + \frac{3 \cdot 3}{\sqrt{26}} \right)}_{s_{q,7}} \vec{m}_7 + \underbrace{\left(\frac{1 \cdot 1}{\sqrt{15}} + \frac{2 \cdot 2}{\sqrt{34}} + \frac{3 \cdot 4}{\sqrt{26}} \right)}_{s_{q,8}} \vec{m}_8$$

$$sim(q,d) = \text{consine}(q,d) = \frac{\sum_{r | s_{q,r} \neq 0 \wedge s_{d,r} \neq 0} s_{q,r} \cdot s_{d,r}}{\sqrt{\sum_{r | s_{q,r} \neq 0 \wedge s_{d,r} \neq 0} s_{q,r}^2} \sqrt{\sum_{r | s_{q,r} \neq 0 \wedge s_{d,r} \neq 0} s_{d,r}^2}}$$

The similarity between the query and doc is calculated in the space of minterm vectors

$$sim(q,d_1) = \frac{s_{q,2}s_{d_1,2} + s_{q,4}s_{d_1,4} + s_{q,6}s_{d_1,6} + s_{q,7}s_{d_1,7} + s_{q,8}s_{d_1,8}}{\sqrt{s_{q,2}^2 + s_{q,3}^2 + s_{q,4}^2 + s_{q,6}^2 + s_{q,7}^2 + s_{q,8}^2}\sqrt{s_{d_1,2}^2 + s_{d_1,4}^2 + s_{d_1,6}^2 + s_{d_1,7}^2 + s_{d_1,8}^2}}$$

Generalized Vector Model (cont.)

- Term Correlation
  - The degree of correlation between the terms k_i and k_j can now be computed as

$$\vec{k}_i \bullet \vec{k}_j = \sum_{\forall r \mid g_i(m_r)=1 \wedge g_j(m_r)=1} \hat{c}_{i,r} \times \hat{c}_{j,r}$$

    - Do not need to be normalized? (because we have done it before! See p26)

More on Generalized Vector Model

- Advantages
  - Model considers correlations among index terms
  - Model does introduce interesting new ideas

- Disadvantages
  - Not clear in which situations it is superior to the standard vector model
  - Computation cost is fairly high with large collections
    - Since the number of “active” minterms might be proportional to the number of documents in the collection

Despite these drawbacks, the generalized vector model does introduce new ideas which are of importance from a theoretical point of view.

Set-Based Model

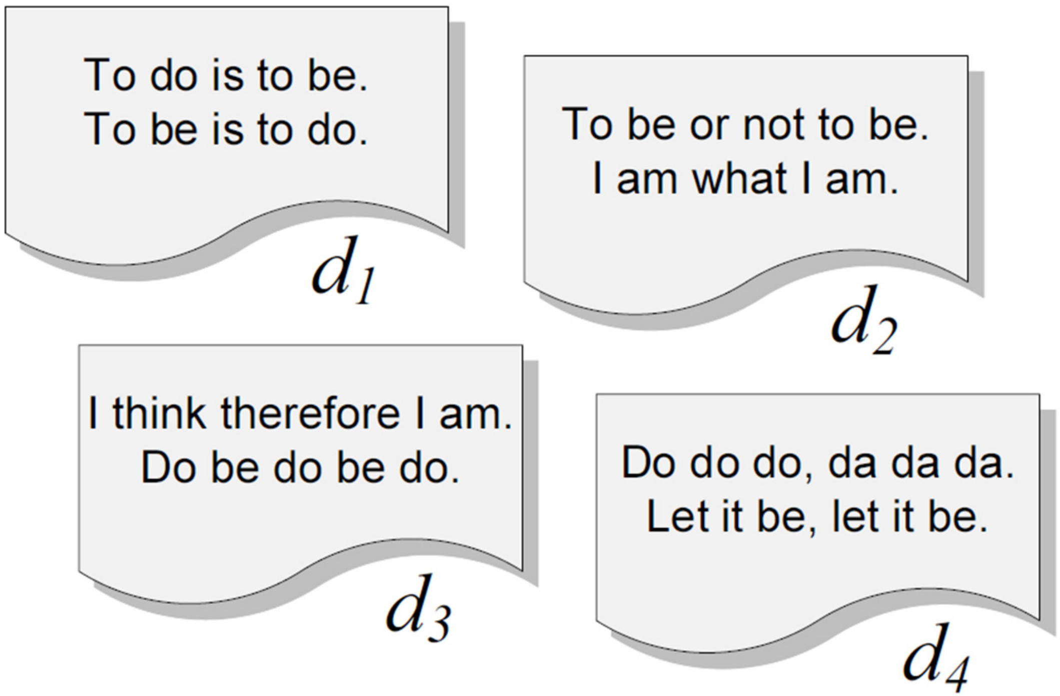
- This is a more recent approach (2005) that combines set theory with a vectorial ranking
- The fundamental idea is to use **mutual dependencies** among index terms to improve results
- Term dependencies are captured through **termsets**, which are sets of correlated terms
- The approach, which leads to improved results with various collections, constitutes the first IR model that effectively took advantage of term dependence with general collections

Set-Based Model: Termsets

- **Termset** is a concept used in place of the index terms
  - A termset $S_i = \{k_a, k_b, ..., k_n\}$ is a subset of the terms in the collection
  - If all index terms in S_i occur in a document d_j then we say that the termset S_i occurs in d_j
- There are 2^t termsets that might occur in the documents of a collection, where t is the vocabulary size
  - However, most combinations of terms have no semantic meaning
  - Thus, the actual number of termsets in a collection is far smaller than 2^t

Set-Based Model: Termsets (cont.)

- Let *t* be the number of terms of the collection
  - Then, the set $V_S = \{S_1, S_2, ..., S_{2^t}\}$ is the **vocabulary-set** of the collection
- To illustrate, consider the document collection below

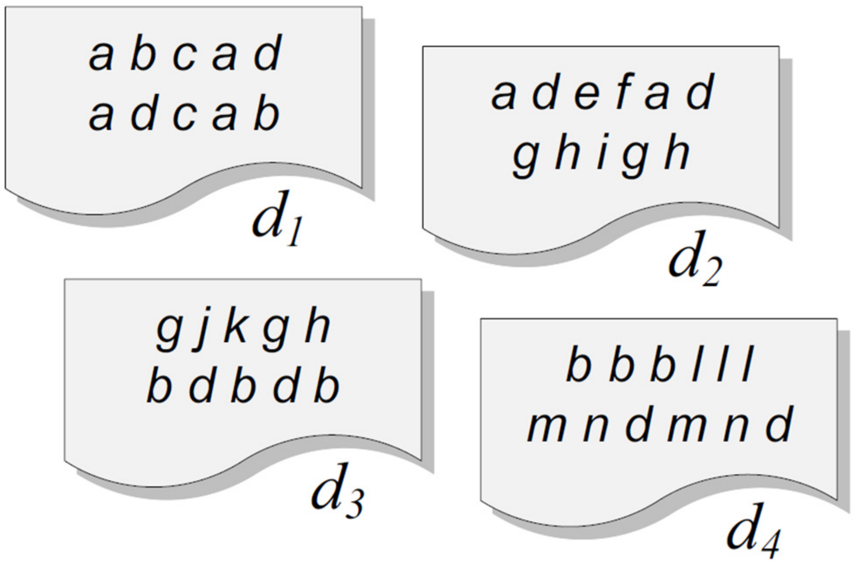


Set-Based Model: Termsets (cont.)

- To simplify notation, let us define

k_a = to	k_d = be	k_g = I	k_j = think	k_m = let
k_b = do	k_e = or	k_h = am	k_k = therefore	k_n = it
k_c = is	k_f = not	k_i = what	k_l = da	

- Further, let the letters *a*...*n* refer to the index terms *ka*...*kn* , respectively



Set-Based Model: Termsets (cont.)

- Consider the query q as "to do be it", i.e. $q = \{a, b, d, n\}$
- For this query, the vocabulary-set is as below

Termset	Set of Terms	Documents
S_a	$\{a\}$	$\{d_1, d_2\}$
S_b	$\{b\}$	$\{d_1, d_3, d_4\}$
S_d	$\{d\}$	$\{d_1, d_2, d_3, d_4\}$
S_n	$\{n\}$	$\{d_4\}$
S_{ab}	$\{a, b\}$	$\{d_1\}$
S_{ad}	$\{a, d\}$	$\{d_1, d_2\}$
S_{bd}	$\{b, d\}$	$\{d_1, d_3, d_4\}$
S_{bn}	$\{b, n\}$	$\{d_4\}$
S_{abd}	$\{a, b, d\}$	$\{d_1\}$
S_{bdn}	$\{b, d, n\}$	$\{d_4\}$

Notice that there are 11 termsets that occur in our collection, out of the maximum of 15 termsets that can be formed with the terms in q

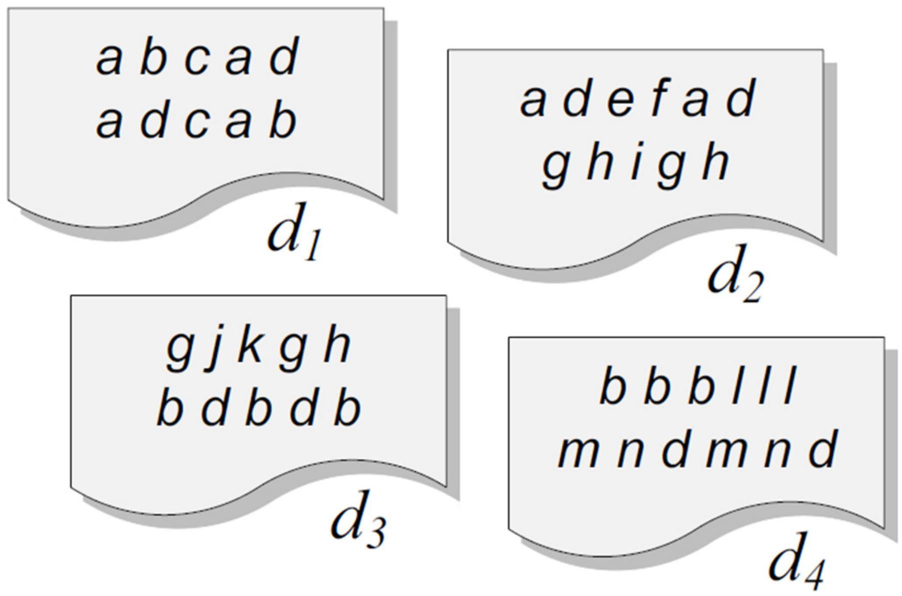
Set-Based Model: Termsets (cont.)

- At query processing time, only the termsets generated by the query need to be considered
  - A termset composed of *n* terms is called an *n*-termset
  - Let N_i be the number of documents in which S_i occurs
- An *n*-termset S_i is said to be **frequent** if N_i is greater than or equal to a given threshold
  - This implies that an *n*-termset is frequent if and only if all of its ($n - 1$)-termsets are also frequent
  - **Frequent termsets** can be used to reduce the number of termsets to consider with long queries

Set-Based Model: Termsets (cont.)

- Let the threshold on the frequency of termsets be 2
- To compute all frequent termsets for the query
- $q = \{a, b, d, n\}$ we proceed as follows

1. Compute the frequent 1-termsets and their inverted lists:
   - $S_a = \{d_1, d_2\}$
   - $S_b = \{d_1, d_3, d_4\}$
   - $S_d = \{d_1, d_2, d_3, d_4\}$



2. Combine the inverted lists to compute frequent 2-termsets:
   - $S_{ad} = \{d_1, d_2\}$
   - $S_{bd} = \{d_1, d_3, d_4\}$

3. Since there are no frequent 3-termsets, stop

Set-Based Model: Termsets (cont.)

- Notice that there are only 5 *frequent* termsets in our collection
- Inverted lists for frequent n-termsets can be computed by starting with the inverted lists of frequent 1-termsets
  - Thus, the only indices required are the standard inverted lists used by any IR system
- This is reasonably fast for short queries up to 4-5 terms

Set-Based Model: Ranking Computation

- The ranking computation is based on the vector model, but adopts termsets instead of index terms
- Given a query q,
  - let $\{S_1, S_2, \ldots\}$ be the set of all termsets originated from q
  - N_i be the number of documents in which termset S_i occurs
  - N be the total number of documents in the collection
  - $F_{i,j}$ be the frequency of termset S_i in document d_j
- For each pair $[S_i, d_j]$ we compute a weight $W_{i,j}$ given by

$$\mathcal{W}_{i,j} = \begin{cases} (1 + \log \mathcal{F}_{i,j}) \log(1 + \frac{N}{N_i}) & \texttt{if} \ \ \mathcal{F}_{i,j} > 0 \\ 0 & \mathcal{F}_{i,j} = 0 \end{cases}$$

- We also compute a $W_{i,q}$ value for each pair $[S_i, q]$

Set-Based Model: Ranking Computation (cont.)

- Consider
  - query $q = \{a, b, d, n\}$
  - document d_1 = "*a b c a d a d c a b*"

Termset	Weight	
S_a	$\mathcal{W}_{a,1}$	$(1+\log 4) * \log(1+4/2) = 4.75$
S_b	$\mathcal{W}_{b,1}$	$(1+\log 2) * \log(1+4/3) = 2.44$
S_d	$\mathcal{W}_{d,1}$	$(1+\log 2) * \log(1+4/4) = 2.00$
S_n	$\mathcal{W}_{n,1}$	$0 * \log(1+4/1) = 0.00$
S_{ab}	$\mathcal{W}_{ab,1}$	$(1+\log 2) * \log(1+4/1) = 4.64$
S_{ad}	$\mathcal{W}_{ad,1}$	$(1+\log 2) * \log(1+4/2) = 3.17$
S_{bd}	$\mathcal{W}_{bd,1}$	$(1+\log 2) * \log(1+4/3) = 2.44$
S_{bn}	$\mathcal{W}_{bn,1}$	$0 * \log(1+4/1) = 0.00$
S_{dn}	$\mathcal{W}_{dn,1}$	$0 * \log(1+4/1) = 0.00$
S_{abd}	$\mathcal{W}_{abd,1}$	$(1+\log 2) * \log(1+4/1) = 4.64$
S_{bdn}	$\mathcal{W}_{bdn,1}$	$0 * \log(1+4/1) = 0.00$

Assume here a minimum threshold frequency of 1.

Set-Based Model: Ranking Computation (cont.)

- A document d_j and a query q are represented as vectors in a 2^t-dimensional space of termsets

$$
\begin{aligned}
\vec{d}_j &= (\mathcal{W}_{1,j}, \mathcal{W}_{2,j}, \ldots, \mathcal{W}_{2^t,j}) \\
\vec{q} &= (\mathcal{W}_{1,q}, \mathcal{W}_{2,q}, \ldots, \mathcal{W}_{2^t,q})
\end{aligned}
$$

- The rank of d_j to the query q is computed as follows

$$
sim(d_j, q) = \frac{\vec{d}_j \bullet \vec{q}}{|\vec{d}_j| \times |\vec{q}|} = \frac{\sum_{S_i} \mathcal{W}_{i,j} \times \mathcal{W}_{i,q}}{|\vec{d}_j| \times |\vec{q}|}
$$

- For termsets that are not in the query q, $W_{i,q} = 0$

Set-Based Model: Ranking Computation (cont.)

- The document norm $|\vec{d_j}|$ is hard to compute in the space of termsets
- Thus, its computation is restricted to 1-termsets
- Let again $q = \{a, b, d, n\}$ and d_1
- The document norm in terms of 1-termsets is given by

$$
\begin{aligned}
|\vec{d}_1| &= \sqrt{\mathcal{W}_{a,1}^2 + \mathcal{W}_{b,1}^2 + \mathcal{W}_{c,1}^2 + \mathcal{W}_{d,1}^2} \\
&= \sqrt{4.75^2 + 2.44^2 + 4.64^2 + 2.00^2} \\
&= 7.35
\end{aligned}
$$

Set-Based Model: Ranking Computation (cont.)

- To compute the rank of d_1, we need to consider the seven termsets S_a, S_b, S_d, S_{ab}, S_{ad}, S_{bd}, and S_{abd}
- The rank of d_1 is then given by

$$
\begin{aligned}
sim(d_1, q) &= (\mathcal{W}_{a,1} * \mathcal{W}_{a,q} + \mathcal{W}_{b,1} * \mathcal{W}_{b,q} + \mathcal{W}_{d,1} * \mathcal{W}_{d,q} + \\
&\quad \mathcal{W}_{ab,1} * \mathcal{W}_{ab,q} + \mathcal{W}_{ad,1} * \mathcal{W}_{ad,q} + \mathcal{W}_{bd,1} * \mathcal{W}_{bd,q} + \\
&\quad \mathcal{W}_{abd,1} * \mathcal{W}_{abd,q}) \,/ |\vec{d_1}| \\
&= (4.75 * 1.58 + 2.44 * 1.22 + 2.00 * 1.00 + \\
&\quad 4.64 * 2.32 + 3.17 * 1.58 + 2.44 * 1.22 + \\
&\quad 4.64 * 2.32)/7.35 \\
&= 5.71
\end{aligned}
$$

BM25 (Best Match 25)

- BM25 was created as the result of a series of experiments on variations of the **probabilistic model**
- A good term weighting is based on three principles
  - **Inverse document frequency**
  - Term frequency
  - Document length normalization
- The classic probabilistic model covers only the first of these principles
- This reasoning led to a series of experiments with the Okapi system, which led to the BM25 ranking formula

BM1, BM11 and BM15 Formulas

- At first, the Okapi system used the Equation below as ranking formula

$$sim(d_j, q) \sim \sum_{k_i \in q \wedge k_i \in d_j} \log\left(\frac{N - n_i + 0.5}{n_i + 0.5}\right)$$

  - which is just the equation used in the probabilistic model, when no relevance information is provided
- It was referred to as the BM1 formula (*Best Match 1*)

BM1, BM11 and BM15 Formulas (cont.)

- The first idea for improving the ranking was to introduce a **term-frequency** factor $F_{i,j}$ in the BM1 formula
- This factor, after some changes, evolved to become

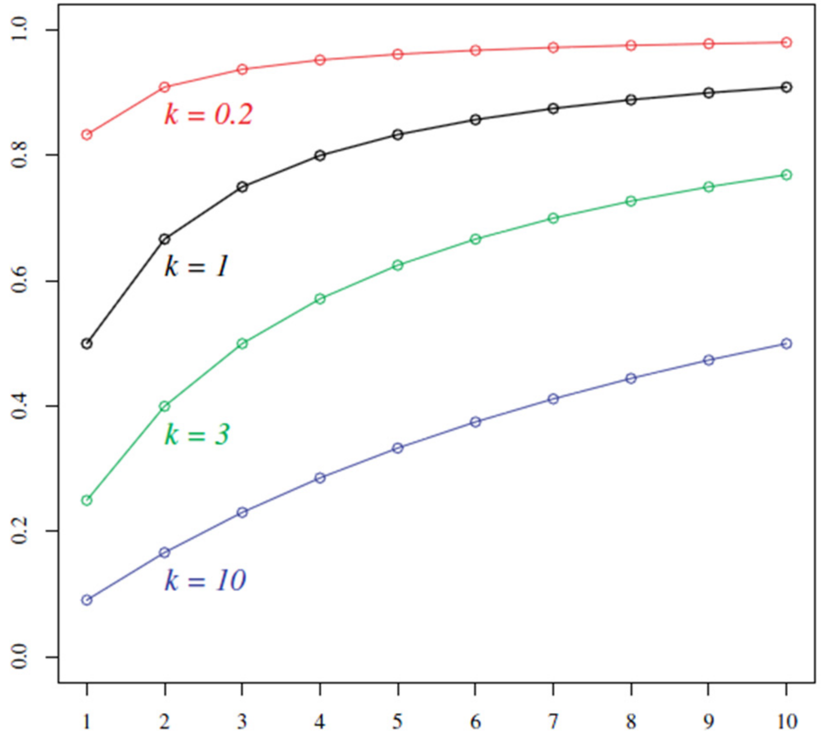
$$F_{i,j} = S_1 \times \frac{f_{i,j}}{K_1 + f_{i,j}}$$

  - Where
    - $f_{i,j}$ is the frequency of term k_i within document d_j
    - K_1 is a constant setup experimentally for each collection
    - S_1 is a scaling constant, normally set to $S_1 = (K_1 + 1)$
- If $K_1 = 0$, this whole factor becomes equal to 1 and bears no effect in the ranking

BM1, BM11 and BM15 Formulas (cont.)

- $\dfrac{f_{i,j}}{K+f_{i,j}}$ can be viewed as a saturation function

$$\frac{f_{i,j}}{K+f_{i,j}}$$



$$f_{i,j}$$

BM1, BM11 and BM15 Formulas (cont.)

- The next step was to modify the *Fi,j* factor by adding **document length normalization** to it, as follows:

$$F'_{i,j} = S_1 \times \frac{f_{i,j}}{\frac{K_1 \times len(d_j)}{avg\_doclen} + f_{i,j}}$$

  – Where
    - *len*(*dj*) is the length of document dj (computed, for instance, as the number of terms in the document)
    - *avg_doclen* is the average document length for the collection

BM1, BM11 and BM15 Formulas (cont.)

- Next, a correction factor $G_{j,q}$ dependent on the document and query lengths was added

$$G_{j,q} = K_2 \times len(q) \times \frac{avg\_doclen - len(d_j)}{avg\_doclen + len(d_j)}$$

  - Where
    - $len(q)$ is the query length (number of terms in the query)
    - K_2 is a constant

BM1, BM11 and BM15 Formulas (cont.)

- A third additional factor, aimed at taking into account term frequencies within queries, was defined as

$$F_{i,q} = S_3 \times \frac{f_{i,q}}{K_3 + f_{i,q}}$$

  – where
    - $f_{i,q}$ is the frequency of term k_i within query q
    - K_3 is a constant
    - S_3 is an scaling constant related to K_3, normally set to $S_3 = (K_3 + 1)$

BM1, BM11 and BM15 Formulas (cont.)

- Introduction of these three factors led to various BM (Best Matching) formulas, as follows:

$$sim_{BM1}(d_j,q) \sim \sum_{k_i \in q \wedge k_i \in d_j} \log\left(\frac{N-n_i+0.5}{n_i+0.5}\right)$$

$$sim_{BM15}(d_j,q) \sim G_{j,q} + \sum_{k_i \in q \wedge k_i \in d_j} F_{i,j} \times F_{i,q} \times \log\left(\frac{N-n_i+0.5}{n_i+0.5}\right)$$

$$sim_{BM11}(d_j,q) \sim G_{j,q} + \sum_{k_i \in q \wedge k_i \in d_j} F'_{i,j} \times F_{i,q} \times \log\left(\frac{N-n_i+0.5}{n_i+0.5}\right)$$

BM1, BM11 and BM15 Formulas (cont.)

- Experiments using TREC data have shown that BM11 outperforms BM15 (due to additional document length normalization)
- Further, empirical considerations can be used to simplify the previous equations, as follows:
  - Empirical evidence suggests that a best value of K_2 is 0, which eliminates the $G_{j,q}$ factor from these equations (i.e., BM15 and BM12)
  - Further, good estimates for the scaling constants S_1 and S_3 are $K_1 + 1$ and $K_3 + 1$, respectively
  - Empirical evidence also suggests that making K_3 very large is better. As a result, the $F_{i,q}$ factor is reduced simply to $f_{i,q}$
  - For short queries, we can assume that $f_{i,q}$ is 1 for all terms

BM1, BM11 and BM15 Formulas (cont.)

- These considerations lead to simpler equations as follows

$$sim_{BM1}(d_j,q) \sim \sum_{k_i \in q \wedge k_i \in d_j} \log\left(\frac{N-n_i+0.5}{n_i+0.5}\right)$$

$$sim_{BM15}(d_j,q) \sim \sum_{k_i \in q \wedge k_i \in d_j} \frac{(K_1+1)f_{i,j}}{K_1+f_{i,j}} \times \log\left(\frac{N-n_i+0.5}{n_i+0.5}\right)$$

$$sim_{BM11}(d_j,q) \sim \sum_{k_i \in q \wedge k_i \in d_j} \frac{(K_1+1)f_{i,j}}{\frac{K_1 \times len(d_j)}{avg\_doclen}+f_{i,j}} \times \log\left(\frac{N-n_i+0.5}{n_i+0.5}\right)$$

BM25 Ranking Formula

- BM25: combination of the BM11 and BM15
- The motivation was to **combine the BM11 and BM25 term frequency factors** as follows

$$B_{i,j} = S_1 \times \frac{(K_1+1) f_{i,j}}{K_1\left[(1-b)+b\frac{len(d_j)}{avg\_doclen}\right]+f_{i,j}}$$

  - Where *b* is a constant with values in the interval [0, 1]
    - If *b* = 0, it reduces to the BM15 term frequency factor
    - If *b* = 1, it reduces to the BM11 term frequency factor
    - For values of *b* between 0 and 1, the equation provides a combination of BM11 with BM15

BM25 Ranking Formula (cont.)

- The ranking equation for the BM25 model can then be written as

$$sim_{BM25}(d_j, q) \sim \sum_{k_i \in q \wedge k_i \in d_j} B_{i,j} \times \log\left(\frac{N - n_i + 0.5}{n_i + 0.5}\right)$$

  – Where K_1 and b are empirical constants
    - $K_1 = 1$ works well with real collections
    - *b* should be kept closer to 1 to emphasize the document length normalization effect present in the BM11 formula
    - For instance, $b = 0.75$ is a reasonable assumption
    - Constants values can be fine tuned for particular collections through proper experimentation

BM25 Ranking Formula (cont.)

- Unlike the probabilistic model, the BM25 formula can be computed without relevance information
- There is consensus that BM25 outperforms the classic vector model for general collections
- Thus, it has been used as a baseline for evaluating new ranking functions, in substitution to the classic vector model